Delaunay-based Derivative-free Optimization via Global Surrogates (Δ-DOGS)

Pooriya Beyhaghi Daniele Cavaglieri Thomas Bewley

June 4, 2013

Abstract. Nonsmooth CFD-based optimization problems are difficult, due both to the nonconvexity of the cost function and to the high cost of function evaluations. In this work, we develop a derivative-free optimization scheme which makes maximum use of each function evaluation, improving on the efficiency of the existing methods that have recently been applied to this class of problems (genetic algorithms, SMF, orthoMADS, etc). At each optimization step, the algorithm proposed creates a Delaunay triangulation based on the existing evaluation points. In each simplex so created, the algorithm optimizes a cost function based on a polyharmonic spline interpolant. This interpolation strategy behaves appropriately even when the evaluation points are clustered in particular regions of interest in the parameter space, in contrast with the Kriging interpolation strategy used in existing GPS/SMF algorithms. At each optimization step, an appropriately-modeled error function is combined with the interpolant, weighted with a tuning parameter governing the trade-off between local refinement and global exploration. An effective rule is introduced to dynamically adapt this tuning parameter at each step. The resulting algorithm is demonstrated on representative test functions.

1 Introduction

In this paper, a new algorithm is presented to

$$\text{minimize } f(x) \text{ subject to } x \in B = \{x | a_i \leq x_i \leq b_i\}, \tag{1}$$

where $x, a, b \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ (we consider simple "box" constraints on the parameters here; more complicated constraints will be considered in future work). Derivative-free algorithms are well suited for such problems even if neither the derivative of $f(x)$ nor its accurate numerical approximation is available. This issue is common in situations in which the cost function is derived from either experiment or simulation, especially if the metric of interest is a finite-time-average approximation of an infinite-time-average statistic of a turbulent flow.

An important class of derivative-free algorithms is Direct Search Methods, developed in the 1960s, as reviewed in [10]. The first type of methods in this class is Pattern Search Methods, which are characterized by a series of exploratory moves on a pattern of points that lie on a rational lattice. Exhaustive Polling and Generalized Pattern Search (GPS) are typical examples. The efficiency and convergence property of these algorithms has been studied by [21] and [23]. Another type of direct search algorithm is the Nelder-Mead simplex algorithm, which is widely implemented in numerical software packages. This method has been studied in [19]. The last type of direct search algorithms is the adaptive direction search algorithm, which includes the Rosenbrock [17] and Powell [15] methods. Direct search algorithms are used to find a local minima whose cost functions are less than the initial guess; note that global convergence is an issue of significant interest when the cost function of interest is nonconvex.

Another class of optimization is Response Surface Methods. In these algorithms, an underlying model for optimization is designed in order to approximate the trend of the cost function at any arbitrary feasible point. Trust Region method is one of the first optimization algorithms appearing in the literature which uses a model for cost function for optimization; however, this model does not used all the information employed in previous steps.

The Response Surface Methods usually uses all previous information in order to find the best model (a surrogate function) for the cost function. An exhaustive review of global optimization methods based on surrogate functions can be found in [9]. Although the majority of them relies on basis function interpolation, Kriging method is the most popular one, since an estimate of the predicted value $\hat{f}(x)$ and the error $\sigma(x)$ that the interpolant is derived simultaneously.

The expected improvement algorithm [18] is a method which uses Kriging interpolation in order to model the cost function in the whole feasible domain. The main advantage of this interpolation is that the value of uncertainty at each point inside the domain is approximated. In other words, with this interpolation, for each point in the domain, the cost function value is considered as a random Gaussian variable which its mean value and its covariance are known. This interpretation for the cost function is used in order to find a point which has a maximum probability of the target value for minimization. A particularly efficient algorithm for global optimization is the Surrogate Management Framework (SMF) which combines the expected improvement algorithm with general pattern search [3]. This algorithm has been extended in [1] where the general pattern search has been employed on a non-Cartesian grid in order to develop a Lattice-based Derivative-free Optimization via Global Surrogates (LABDOGS) highly leveraging dense sphere packing theory with the improving the uniformity of the discretization process.

Nevertheless, Kriging interpolation has its own drawbacks. One of the problems is its numerical inaccuracy when data are clustered in a narrow region or when the number of parameters of the interpolant grows. Furthermore, finding the interpolation and minimizing the search function are non-convex optimization problems which have to be solved with another global optimization algorithm. Even though, these optimization problems are less expensive compared to the original problem, they carry this disadvantage of being nonsmooth and nonconvex; thus, deriving their global minimum is still an issue.

In order to overcome the limits of Kriging interpolation, we seek another interpolation which leads to an easier minimization. However, we have to remark that in this way we automatically lose the estimation of uncertainty provided by Kriging. For interpolation with radial basis functions, an appropriate uncertainty function has been presented by [8] which is related to the properties of these interpolations.

In this paper, we will introduced an artificial function quantifying the uncertainty based on the distance from the evaluation points. This uncertainty function relies on the concept of Delaunay triangulation which is independent from our interpolation; hence, any smooth interpolation could be used in this algorithm instead of the one here proposed. The structure of the paper is the following. In Section 2, we will describe in detail all the elements that compose the algorithm. Proofs of convergence and properties of the algorithm will be also provided. In Section 3, the results of the application of our global optimization algorithm to a selected number of test functions will be presented and discussed.

2 Description of the algorithm

Algorithm 1. *In this algorithm, we will find the global minimum of a nonconvex function $f(x)$ inside of a rectangular domain defined by the user. The steps that constitute the algorithm are the following:*

1. *Define the domain as a box $X_{1_i} < x_i < X_{2_i}$ with upper and lower bounds for each variable.*

2. *Define a set of initial evaluation points and add the vertices of the box to it.*

3. *Calculate an interpolation function $p(x)$ among the set of evaluation points.*

4. *Perform a Delaunay triangulation among the points.*

5. *For each simplex S_i:*

- *Calculate its circumcenter x_C and the radius of the circumsphere R.*
- *Define an uncertainty function $e_i(x)$ such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.*
- *Define a search function as $c_i(x) = p(x) - K\, e_i(x)$*
- *Minimize the search function in the simplex.*

6. *Take the minimum of the result of the minimization performed in each simplex and add it to the set of evaluation points.*
7. *Repeat steps 3 to 6 until convergence.*

In this framework, $p^n(x)$ is an interpolating function for unstructured data at step n. In the present work, we adopted polyharmonic spline interpolation due to its capability of handling points very close to each other and points widely distant without causing unwanted bumps which would originate nonexistent local minima in the minimization process. The function $e^n(x)$ is the uncertainty function, which artificially defines the amount of interest to be put in to the unexplored zone between the evaluation points. The value of this function is zero at the evaluation points and positive elsewhere. Finally, K is a tuning parameter which defines the trade-off between the global exploration versus the local refinement. This is the only user-defined parameter in the whole algorithm. Later in the present work, a strategy will be introduced to dynamically choose the optimum value of K. In this way, we can define a search function $c^n(x) = p^n(x) - K e^n(x)$ that will be used in order to find the best candidate for the next evaluation point.

This algorithm allows us prove the following

Lemma 1. *Assume that $f(x)$ and $p^n(x)$ are uniformly continuous functions and define x^* as the global minimum of $f(x)$. If for all n, a point $\tilde{x}$ exists with $f(x^*) > c^n(\tilde{x})$, then for any $\varepsilon > 0$ there exists a δ such that*

$$\begin{gathered} \textit{if} \quad \|x_i - x_j\| < \delta, \quad j > i \\ \Rightarrow 0 \le f(x_j) - f(x^*) \le \varepsilon \end{gathered}$$

Proof. Since $f(x)$ and $c^j(x)$ are uniformly continuous functions, then there exist δ_1 and δ_2 such that

$$\text{if} \quad \|x_i - x_j\| < \delta_1 \Rightarrow \|f(x_j) - f(x_i)\| < \varepsilon/2$$

and

$$\text{if} \quad \|x_i - x_j\| < \delta_2 \Rightarrow \|c^j(x_j) - c^j(x_i)\| < \varepsilon/2$$

Now define $\delta = \min(\delta_1, \delta_2)$. We have

$$\begin{gathered} \text{if} \quad \|x_i - x_j\| < \delta, \quad j > i \\ \Rightarrow \|f(x_j) - f(x_i)\| < \varepsilon/2 \quad \text{and} \\ \|c^j(x_i) - c^j(x_j)\| < \varepsilon/2. \end{gathered}$$

Since x_i is one of the evaluation points at the j-th step, the value of the uncertainty function at this point is zero and the values of the interpolant $p_j(x_i)$ and function $f(x_i)$ are equal. Therefore, we have:

$$c^j(x_i) = p^j(x_i) - K\, e^j(x_i) = f(x_i). \tag{2}$$

Since, by definition, x_j is the global minimum of $c_j(x)$, we have

$$c^j(x_j) \le c^j(\tilde{x}) \le f(x^*)$$

hence

$$\begin{aligned} &c^j(x_j) > c^j(x_i) - \varepsilon/2 = f(x_i) - \varepsilon/2 \\ &f(x_i) > f(x_j) - \varepsilon/2 \\ &f(x_j) - \varepsilon < f(x^*) \end{aligned}$$

Finally, since $f(x^*) \le f(x_j)$

$$0 \le f(x_j) - f(x^*) \le \varepsilon \tag{3}$$

□

Remark 1. *In the aforementioned algorithm, we have a set of evaluation points to which we will add a point at each step. Then, if the new point is close enough to one of the existing evaluation points, i.e. $\|x_i - x_j\| < \delta$, the algorithm will stop and the last point will become the best candidate for the global minimum. According to Lemma 1, the cost function evaluated at this point is within the ε neighborhood of the global minimum provided that $f(x^*) - c^n(x^*)$ is positive for all n.*

This preliminary result, together with the following Lemma, will allow us to show that if we choose K large enough, the algorithm will converge to the global minimum.

Lemma 2. *If a function $G(x)$ defined in $\mathbb{R}^d$ is strictly convex inside a simplex S, then the maximum value of $G(x)$ is located at one of its vertices.*

The proof of this lemma is relatively easy, and will be presented in the full version of this paper. This result allows us to prove the following

Theorem 1. *The algorithm 1 will converge to the global minimum inside the domain, provided K satisfies the following inequality*

$$K \geq \lambda_{\max}(\nabla^2 f(x) - \nabla^2 p_n(x))/2 \tag{4}$$

for all x located in the simplex including the global minimum x^*, where $\lambda_{\max}$ represents the maximum eigenvalue.

Proof. Since the uncertainty function inside of each simplex is defined as

$$e(x) = R^2 - (x - x_C)^T (x - x_C) \tag{5}$$

its Hessian is simply

$$\nabla^2 e(x) = -2\, I$$

Now, let us define a function $G(x)$ as follows:

$$G(x) = p_n(x) - K\, e_n(x) - f(x), \tag{6}$$

hence

$$\nabla^2 G(x) = (\nabla^2 p_n(x) - \nabla^2 f(x)) + 2\, K\, I \tag{7}$$

By choosing K according to (4), function $G(x)$ is strictly convex inside of the simplex including x^*. Moreover, the value of $G(x)$ at the vertices of this simplex is zero, thus, according to Lemma 2 $G(x^*) < 0$. Then, by using Lemma 1 for arbitrarily small ε, we can conclude that the present algorithm will converge to the global minimum. $\square$

What follows is a detailed description of the main parts which together constitute the above-described algorithm.

2.1 Polyharmonic spline interpolation

The polyharmonic spline interpolation $p(x)$ of a function $f(x)$ in $\mathbb{R}^d$ is defined as the sum of the weighted summation of radial basis functions $\varphi(r)$ and a linear combination of the N given evaluation points x_i, i.e.

$$p(x) = \sum_{i=1}^{N} w_i\, \varphi(r) + v^T \begin{bmatrix} 1 \\ x \end{bmatrix}, \tag{8}$$
$$\text{where} \quad \varphi(r) = r^3 \quad \text{and} \quad r = \|x - x_i\|.$$

Weights w_i and v_i represents N and $d+1$ unknowns, respectively, to be determined through appropriate conditions. First of all, we want to match the values $f(x_i)$ at the evaluation points x_i with the values of the interpolant, i.e. $p(x_i) = f(x_i)$. This gives N conditions. Then, we impose the orthogonality conditions $\sum_i w_i = 0$ and $\sum_i w_i x_{ij} = 0, \quad j = 1, 2, \ldots, d$. This gives the other $d+1$ conditions. Thus we can rewrite the whole problem as

$$\begin{bmatrix} F & V^T \\ V & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \begin{bmatrix} f(x_i) \\ 0 \end{bmatrix} \quad \text{where}$$
$$F_{ij} = \varphi(\|x_i - x_j\|) \quad \text{and} \tag{9}$$
$$V = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix}.$$

The smoothness and computational cost of this intepolation has been studied in [5] and [22] in detail.

2.2 Uncertainty function

By using polyharmonic splines, we lose the advantage of Kriging interpolation which internally provides an estimation of uncertainty related to the stochastic nature of such approach; hence, the need to build an artificial function representing the uncertainty among the evaluation points. Assuming zero uncertainty at the evaluation points, and grow positively amongst them, such function must go to zero at the interpolation points.The uncertainty function here developed is based on the concept of circumsphere of a simplex.

According to the definition of simplex in [4, p. 32], suppose we have $d+1$ points $V_0, V_1, \ldots, V_d \in R^d$ such that $V_0 - V_1, V_0 - V_2, \ldots, V_0 - V_d$ are linearly independent, then the simplex defined by these points represents the convex hull of such points. Consider a simplex that pass through V_0, V_1, V_2, ..., V_d. Assume x_C as its circumcenter and R the radius of the associated circumsphere; then we can define the related uncertainty function $e(x)$ in each point in the simplex as

$$e(x) = R^2 - (x - x_C)^T(x - x_C) \tag{10}$$

This function is zero at the vertices and positive inside the simplex. In this way, the uncertainty function is defined piecewise in each simplex, but the particular construction assures that the function is at least C_0, i.e.

Lemma 3. *The uncertainty function defined in* (10) *is continuous.*

Proof. Consider x as a point on the boundary shared between two different simplices S_1 and S_2. Assume x_{C_1} and x_{C_2} are the circumcenters and $e_1(x)$ and $e_2(x)$ are the values of the error functions of S_1 and S_2, respectively, evaluated at x. Since the triangulation fully covers the domain with simplices, the interface between S_1 and S_2 is another simplex of lower dimension S_3. The projection of x_{C_1} and x_{C_2} on the simplex S_3 is by construction its circumcenter x_{C_3}. Therefore, $x_{C_1}x_{C_3}$ and $x_{C_2}x_{C_3}$ are perpendicular to the simplex S_3. Now consider x_{S_3} as one of the vertices of simplex S_3. Some trivial considerations on triangles $x_{C_1}xx_{C_3}$ and $x_{C_1}x_{S_3}x_{C_3}$ lead to

$$\begin{aligned} e_1(x) &= \|x_{C_1}x_{S_3}\|^2 - \|x_{C_1}x\|^2 \\ \|x_{C_1}x_{S_3}\|^2 &= \|x_{C_1}x_{C_3}\|^2 + \|x_{C_3}x_{S_3}\|^2 \\ \|x_{C_1}x\|^2 &= \|x_{C_1}x_{C_3}\|^2 + \|x_{C_3}x\|^2 \end{aligned} \tag{11}$$

By combining the equations in (11), we get

$$e_1(x) = \|x_{C_3}x_{S_3}\|^2 - \|x_{C_3}x\|^2 \tag{12}$$

With a similar reasoning, we can obtain

$$e_2(x) = \|x_{C_3}x_{S_3}\|^2 - \|x_{C_3}x\|^2 \tag{13}$$

Hence, the value of the uncertainty function at x is equal at the interface of two neighboring simplices. □

Remark 2. *Since the uncertainty function is continuous, and the domain is compact; therefore, it is uniformly continuous.*

2.3 Delaunay triangulation

Since the uncertainty function has been defined piecewise in each simplex obtained by connecting the evaluation points and the vertices of the domain, an algorithm for the triangulation of such points is required.

By definition, if S is a set of points in $\mathbb{R}^d$, the triangulation of S is a set of simplices whose vertices are elements of S. A triangulation is defined as valid if the following conditions hold:

- Every point of S is a vertex of a simplex in the triangulation
- The union of these simplices fully covers the convex hull of S
- The intersection of two different simplices is a k-simplex such that $k = -1, 0, \ldots, d-1$. For example, in a three-dimensional problem the intersection of two tetrahedra must be an empty set, a vertex, an edge or a triangle.

Delaunay triangulation is a valid triangulation such that the intersection of the open circumsphere around each simplex with S is empty. This specific triangulation, with respect to all the possible triangulations, has the following properties:

- The maximum circumradius among the simplices is minimum;
- The sum of the squares of the edge lengths weighted by the sum of the volumes of the elements sharing these edges is minimal

The main advantage of using Delauney triangulation is to improve the numerical accuracy of the algorithm. Remind the definition of the uncertainty function in (10), if the ratio between the circumradius and the maximum distance between two edges of a simplex is a large number, the problem of finding the circumcenter is an ill-posed problem. As a consequence, numerical errors can arise and undermine the evaluation of the uncertainty function. In order to avoid this, it is necessary to minimize the maximum circumradius of the simplices in the triangulation, and this goal is achieved by Delaunay triangulation.

Finding a Delaunay triangulation has been a challenging problem for a long time in computational geometry and a great number of algorithms have been proposed. For the interested reader an exhaustive review can be found in [6] and [7]. In the present work, a Delaunay triangulation must be performed over a set of initial evaluation points to which a new point is added at each iteration. Hence, the incremental method (see [7], chapter 2) is particularly appropriate for our goal. There are a number of other algorithms which perform better as for computational cost and memory storage, yet in this framework, because of the specific structure of the problem, the incremental method represents the best choice.

After constructing a triangulation for the domain, we have to find the minimum of the search function in each simplex. This is the most expensive part of the algorithm, and its computational cost is proportional to the number of simplices M obtained by the Delaunay triangulation. Thus, the number of simplices plays a key role in our optimization computation cost. The expression for the number of triangles for the two-dimensional problem as given in [16] is:

$$M = N_B + 2\,N_I - 2 \tag{14}$$

where N_B is the number of vertices on the boundary and N_I is the number of interior vertices. Hence, in a two-dimensional problem, the number of triangles is $O(N)$ where N is the number of vertices. Unfortunately, a similar formula does not exist for more that two dimensions. In other words, different triangulations for higher dimensional problems have a different number of simplices. However, an upper bound for the number of simplices given by the Delaunay triangulation has been derived in [12], i.e.

$$M \leq O(\lceil N^{\frac{d}{2}} \rceil) \tag{15}$$

where d is, as usual, the dimension of the problem. Once the triangulation has been completed, it is possible to perform the optimization in each simplex, as described in the following subsection.

2.4 Optimization of the search function

As previously stated, at iteration n, for each simplex i, it is required to minimize the following search function

$$c_i^n(x) = p^n(x) - K\,e_i^n(x). \tag{16}$$

Since $e(x)$ is defined piecewise, it is required to solve an optimization problem in each simplex which is a nonconvex optimization problem with linear constraints. The result of all the minimizations is the global minimum of the search function $c^n(x)$. Having information about Hessian and gradient plays a key role in local minimization of the search function. Fortunately, for the search function we have an analytical expression for both its gradient and Hessian. The search function is a linear combination of the uncertainty function and polyharmonic spline interpolation. The uncertainty function is a quadratic function whose gradient and Hessian can be derived as following:

$$\nabla(e_i(x)) = -2\,(x - x_{c_i}) \tag{17}$$

$$\nabla^2(e_i(x)) = -2\,I \tag{18}$$

For the polyharmonic spline as defined in 8, the gradient and Hessian formula is

$$\begin{aligned} \nabla p(x) &= \nabla \left(\sum_{i=1}^{N} w_i \|x - x_i\|^3 + v^T \begin{bmatrix} 1 \\ x \end{bmatrix} \right) = \\ &= 3 \sum_{i=1}^{N} w_i \|x - x_i\| (x - x_i) + \bar{v}, \end{aligned} \tag{19}$$

where $\bar{v} = [v_2, v_3, ..., v_{d+1}]^T$.

$$\begin{aligned} \nabla^2 p(x) &= \nabla^2 \left(\sum_{i=1}^{N} w_i \|x - x_i\|^3 + v^T \begin{bmatrix} 1 \\ x \end{bmatrix} \right) = \\ &= 3 \sum_{i=0}^{N} w_i \left(\frac{(x - x_i)(x - x_i)^T}{\|x - x_i\|} + \|x - x_i\| I_{d \times d} \right) \end{aligned} \tag{20}$$

In order to minimize the search function in each simplex, a good initial guess of the solution is required. For this reason, we decided to take the result of the optimization problem obtained after replacing the polyharmonic spline interpolant with a linear piecewise interpolation passing through the vertices of the simplex as a initial guess for global minimum of search function. In this way, we rewrite the coordinates of a point inside the simplex as a linear combination of the vertices of the simplex, i.e.

$$x = X_i \, w$$

where X_i is a $d \times (d+1)$ matrix whose columns are the coordinates of the $d+1$ vertices of the simplex S_i and w is the $(d+1)$-vector of weights w_j. In this way we can represent any point inside the simplex provided the following conditions are satisfied:

$$\begin{aligned} &\sum_{j=1}^{d+1} w_j = 1 \\ &w_j \geq 0 \quad j = 1, 2, \ldots, d+1 \end{aligned} \tag{21}$$

In this fashion, at each iteration n, in each simplex S_i we are required to minimize a new function $c_i'^n(w)$ defined as

$$c_i'^n(w) = Y_i \, w - K \, \left[R_i^2 - (X_i \, w - x_C)^T (X_i \, w - x_C)\right] \tag{22}$$

which can be rewritten in quadratic form as

$$\begin{aligned} c_i'^n(x) = K \, w^T \, X_i^T X_i \, w + (Y_i - 2K x_C^T X_i) \, w + \\ + K \, (x_C^T x_C - R_i^2). \end{aligned} \tag{23}$$

In a similar fashion, we can rewrite the constraints in (21) as

$$\begin{aligned} \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix} w = 1 \\ -Iw \leq 0 \end{aligned} \tag{24}$$

Minimization of (23) can be performed using convex quadratic programming, provided the constraints in (24) are implemented. The optimization gives the vector of weights w_0, which defines the initial point for the minimization of the search function. Since we have a good initial guess for the search function minimization in each simplex, the minimum of the search function in each simplex can be obtained by a local optimization method; furthermore, the gradient and Hessian of the search function has been derived analytically; thus, Newton's based method is a good algorithm for this local minimization.

Newton's method is a line search algorithm whose descent direction has been derived based on both Hessian and gradient of the cost function, yet because of the nonconvexity of the problem; the Hessian modification is required. The Hessian modification that has been used in our algorithm is the modified Cholesky factorization, and the line search algorithm is backtracking line search algorithm (Algorithm 3.1 [14]). The convergence of this algorithm to a local minimum has been proved in [14].

2.5 Optimization With Adaptive K

Since K is the only tuning parameter and the outcome of the algorithm turns out to be strongly affected by it, we introduced an optimal way to dynamically adjust K at each iteration according to the outcome of the optimization performed at the previous iteration.

The strategy directly relies on Lemma 1. According to this lemma, the global convergence can be assured if there exists an x which satisfied $f(x^*) \geq p(x) - K \, e(x)$, where x^* is the point that globally minimizes the cost function. Now assume that y_0 is a lower bound for the cost function $f(x)$. In this way, if we choose K such that there exists x such that $y_0 \geq p(x) - K \, e(x)$, then the global convergence can be assured. Thus, we are looking for smallest value of K which has this property. This value can be derived as following:

$$K = \min \frac{p(x) - y_0}{e(x)}, \tag{25}$$

and the x which minimize above the expression will minimize the correspondent search function $p(x) - K \, e(x)$ too. In this way, It is possible to have a negative value of K; however, in that situation, we forced K to be zero, and the search function in that step would be $c(x) = p(x)$ in the whole domain in order to find the next candidate point for the global minimum.

Like the previous search function minimization, in order to minimize $\frac{p(x)-y_0}{e(x)}$ for each simplex, a good initial guess is required. In each simplex, this point must have a large value of $e(x)$. We will consider a point in the simplex which has a minimum distance from its circumcenter. This point is the circum center of the simplex; however, simplex does not include its circumcenter necessary. In this way, we will consider the projection of this point on the simplex as an initial point.

With a good initial guess for minimum of $\frac{p(x)-y0}{e(x)}$ in each simplex, we can find its global minimum by using an iterative derivative-based method. For this part, we use the Newton's method with Cholescy modification too, since the Hessian and derivative of the function $\frac{p(x)-y0}{e(x)}$ can be derived from Hessian and gradient of function $p(x)$ and $e(x)$.

If the value of y_0 is not available, we can choose some initial value for it, and adapt it during the optimization process. Note that the value of K that we use by this method is the smallest value as possible, if this value became so large during the process; then, the approximate value of y_0 is conservative for the problem.

The algorithm for optimization with Adoptive K can be formalized in the following expressions:

Algorithm 2. *In this algorithm, we will express an algorithm which can find the global minimum of a cost function $f(x)$ with adaptive K, and y_0 assumed as a lower bound for the cost function $f(x)$.*

1. *Define the domain as a box $X_{1_i} < x_i < X_{2_i}$ with upper and lower bounds for each variable.*
2. *Define a set of initial evaluation points and add the vertices of the box to it.*
3. *Calculate an interpolation function $p(x)$ among the set of evaluation points.*
4. *Perform a Delaunay triangulation among the points.*
5. *For each simplex S_i:*
    - *Calculate its circumcenter x_C and the radius of the circumsphere R.*
    - *Define an uncertainty function $e_i(x)$ such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.*
    - *Put $x_1 = x_C$ if x_C is inside of the simplex; else put x_1 as a projection of x_C on the simplex.*
    - *Use Newton's method in order to minimize $(p(x) - y0)/e(x)$ with the initial point x_1 in the circumcenter of the simplex. If during the Newton's algorithm we reach a point x_m such that $p(x_m) \leq y_0$; then we will minimize $p(x)$ in all simplices instead of $\frac{p(x)-y_0}{e(x)}$.*
6. *Take the minimum of the result of the minimization performed in each simplex and add it to the set of evaluation points;*
7. *Repeat steps 3 to 6 until convergence.*

3 Results

In order to evaluate the performance of our optimization algorithms, we applied them to the minimization of the two optimization test functions. The functions we tested are:

- *Rosenbrock function:*

$$f(x,\, y) = (1 - x^2) + 100\,(y - x^2)^2 \qquad (26)$$

- *Rastrigin function:*

$$f(x,\, y) = 20 + x^2 + y^2 + \\ - 10\cos(2\pi\, x) - 10\cos 2\pi\, y \qquad (27)$$

For each simulation, we used Latin hypercube sampling as first introduced in [11] in order to define a set of N_i initial points. According to this statistical approach, we divided the rectangular domain with a grid having N_i intervals for each dimension. In this way the domain is divided into d^{N_i} subintervals. Then we place one evaluation point in each subinterval so that all the other subintervals that have one dimension in common are left empty. In this way we are able to place exactly N_i points, no matter the dimension of the problem. For our purpose, we defined a number of initial evaluation points N_i equal to 2^d for each test, unless otherwise specified, in order to account for the higher exploration required by the increased domain.

The fist test function is Rosenbrock function which has the global minimum in (1, 1) over the domain $x_i \in [-2\,2]$, but the function is nearly flat along the curve $y = x^2$ where the global minimum lies, thus leading all the derivative-based methods to converge extremely slowly. We have implemented our algorithms for this test function. When we use algorithm 1a high value of the tuning parameter $K = 100$ for the optimization (Figure 1) allows a wide exploration all over the domain, with the candidate minimum points concentrating inside the valley of the function. A small value of tuning parameter $K = 3$ (Figure 2), instead, limits the research to the very bottom of the valley, approaching from the very beginning the global minimum of the function; however, the convergence to the global minimum cannot be achieved. An adaptive K (Figure 3) offers a sufficient degree of exploration at the beginning, then, once the valley has been located, the exploration is confined to the curve $y = x^2$ where all the minima lie.

Rastrigin function is a non-convex test function example which has global minimum at [0, 0] in the domain $x_i \in [-2,\, 2]$ presents a higher number of local minima and a unique global minimum at the origin. The results for this test function is presented in Figures (4), (5),(6). It can be observed that the global minimum has been derived when we use a large value of K for optimization, yet a lot of function evaluation has been used. Unfortunately the small value for K has been stocked in a local minimum of the test function. However, the result for adaptive K algorithm is acceptable since the global minimum has been found with reasonable amount of cost function evaluation.

4 Conclusions

In this paper, we have developed a derivative-free optimization algorithm with surrogate functions which leverages the concept of Delaunay triangulation. We have achieved this goal by using polyharmonic spline interpolation with some initial feasible points for the cost function, then an appropriate uncertainty func-

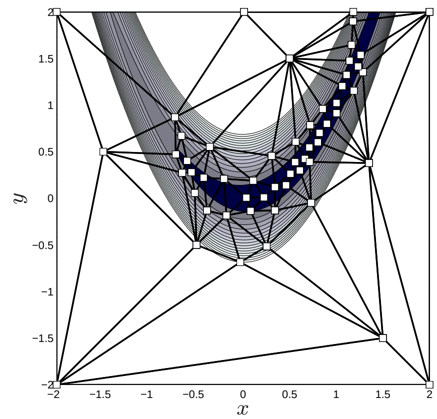

Figure 1: Rosenbrock function for $K = 100$

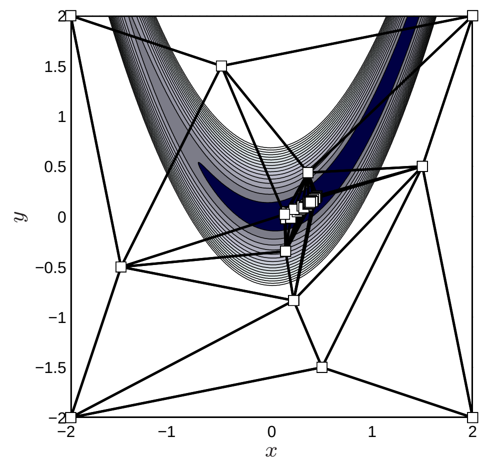

Figure 2: Rosenbrock function for $K = 3$

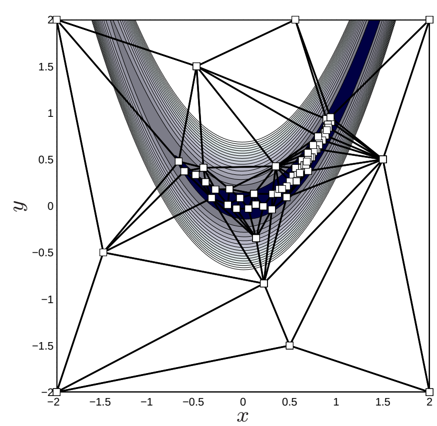

Figure 3: Rosenbrock function for dynamic K

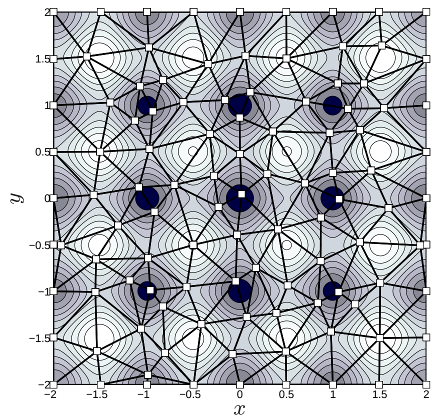

Figure 4: Rastrigin function for small tuning parameter $K = 300$

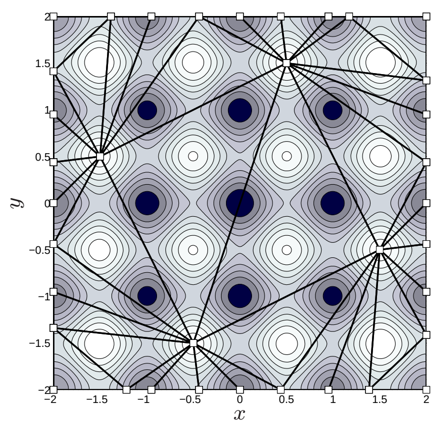

Figure 5: Rastrigin function for large tuning parameter $K = 20$

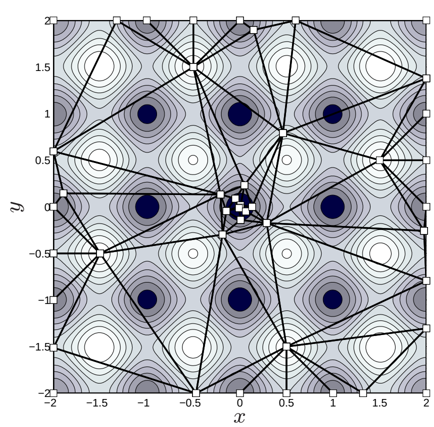

Figure 6: Rastrigin function for adaptive K

tion has been designed for the data points in each step based on Delaunay triangulation; afterwards a search function has been defined for each feasible point. The candidate point for the global minimum of the cost function in each iteration is defined as a point which globally minimizes the search function.

We have introduced a tuning parameter K in the definition of the search function which under some assumption for this parameter, the global convergence has been assured; then we have developed another algorithm which could find the optimum value for the tuning parameter K which can globally minimize the cost function with the minimum number of function evaluations.

The main advantage of our method compared to expected improvement algorithm [18] is related to the smoothness of the polyharmonic spline interpolation for unstructured data. Moreover, the search function which has been developed in this paper has analytic expression for its derivative and Hessian; thus, its minimum could be found with Newton method, provided a good initial guess of the solution is given.

Another advantage of our algorithm is that it is insensitive to minimization of the search function. In other words, the global convergence can be achieved for the cost function, even if we locally minimize the search function in each step. This is an important property which shows the robustness of the algorithm.

Although this methods works well for the test functions that has been presented in this paper, it is not practical yet for high dimensional problems due to the exponential growth of the number of simplices with the dimension. This is an important limitation for our optimization algorithm; however, this issue can be solved by changing the feasible domain to a simplex instead of a hypercube.

In future work, it is our intent to modify our algorithm in order to deal with constrained optimization problems, and also combine this method with a Direct Search algorithm in order to find a more efficient method, since the speed of convergence for Direct Search methods is higher; however, the global minimum may be not achieved. The Hybrid methods can be used in order to find the global minimum with high rate of convergence.

References

[1] P. Belitz, *Applications on Multi-Dimensional Sphere Packings: Derivative-Free Optimization*, PhD Dissertation. University of California, San Diego, 2011.

[2] P. Belitz and T. Bewley, *New Horizons in Sphere Packing Theory, Chapter II: Lattice Based Derivative Free Optimization via Global Surrogates*. SIAM Journal on Optimization, submitted, 2011.

[3] A.J. Booker and J.E. Deniss, P.D. Frank, D.B. Serafini, V.Torczon and M.W.Trosset, *A Rigorous framework for optimization of expensive function by surrogates*. Springer-Verlag, 1999.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[5] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans, *Reconstruction and Representation of 3D Objects with Radial Basis Functions*. ACM SIGGRAPH, 2001.

[6] R.A. Dwyer, *A Faster Divide-and-Conquer Algorithm for Constructing Delaunay Triangulation*. Algorithmica, 1987.

[7] P.L. George and H. Borouchaki, *Delauney triangulation and meshing: Application to Finite element*. Paris: Herms, 1998.

[8] H.M. Gutmann *A Radial Basis function method for Global Optimization*. Journal of Global Optimization 21: 345-383, 2001.

[9] D. Jones, *A Taxonomy of Global Optimization Methods Based on Response Surfaces*. Journal of Global Optimization 21: 345-383, 2001.

[10] R. M. Lewis, V. Torczon and M. W. Trosset *Direct Search Method: Then and Now*. NASA/CR-2000-210125, ICASE Report No.2000-26, 2000.

[11] M.D. McKay, R.J. Beckman and W.J. Conover, *A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code*. Technometrics 21(2): 239-245, 1979

[12] P. McMullen, The maximum numbers of faces of a convex polytope. Mathematika 17: 179-184, 1971.

[13] M. Molga and C. Smutnicki, *Test functions for optimization needs*, 2005.

[14] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[15] M.J.D Powell,*An efficient method for finding the minimum for function of several variables without calculating derivatives*, The Computer Journal, 7, 1964.

[16] F.P. Preparata and M.I. Shamos, *Computational Geometry-an Introduction*. Springer, 1985.

[17] H.H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*. The Computer Journal, 1960.

[18] M. Schonlau, W.J. Welch, and D.R. Jones, *A Data-Analytic Approach to Bayesian Global Optimization*, 1997.

[19] W. Spendley, G.R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometric, 1962.

[20] A. Iske, *Multiresolution Methods in Scattered Data Modeling*, Springer, 2004.

[21] V. Torczon, Multi-Directional Search *A Direct Search Algorithm for Parallel Machines* , PHD thesis, Rice University, Houstin, TX, 1989,

[22] G. Wahba, *Spline Models for Observational Data*, Number 59 in CBMS-NSF, 1990. Regional Conference Series in Applied Math. SIAM, 1990.

[23] V. Torczon,*On The Convergence Of Pattern Search Algorithms*, SIAM J.OPTIM, Vol.7, No.1, 1997.

[24] S. Xu, R. M.Freund and J. Sun, *Solution Methodologies for the Smallest Enclosing Circle Problem*. Kluwer Academic Publisher, Netherlands, 2003.

[25] A. Torn and A. Zilinkas, *Global Optimization*. Springer, Berlin, 1987.