2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

An active subspace method for accelerating convergence in Delaunay-based optimization via dimension reduction

Muhan Zhao ⋆ Shahrouz Ryan Alimo† Thomas R. Bewley⋆

Abstract— **Delaunay-based derivative-free optimization (Δ-DOGS) is an efficient and provably-convergent global optimization algorithm for problems with computationally-expensive function evaluations, including cases for which analytical expressions for the objective function may not be available. Δ-DOGS belongs to the family of response surface methods (RSMs), and suffers from the typical "curse of dimensionality", with the computational cost increasing quickly as the number of design parameters increases. As a result, the number of design parameters n in Δ-DOGS is typically limited to $n \lesssim 10$. To improve performance for higher-dimensional problems, this paper proposes a combination of derivative-free optimization, seeking the global minimizer of a successively-refined surrogate model of the objective function, and an active subspace method, detecting and exploring preferentially the directions of most variability of the objective function. The contribution of other directions to the objective function is bounded by a small constant. This new algorithm iteratively applies Δ-DOGS to seek the minimizer on the d-dimensional active subspace that has most function variation. Inverse mapping is used to project data from the active subspace back to full-model for evaluating function values. This task is accomplished by solving a related inequality constrained problem. Test results indicate that the resulting strategy is highly effective on a handful of model optimization problems.**

I. Introduction

In this paper, we consider a nonconvex optimization problem as follows:

$$\text{minimize } f(x) \text{ with } \ x \in B = \{x | a \leq x \leq b\}, \qquad (1)$$

where a and b are two vectors in $\mathbb{R}^n$ such that $a < b$, and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is an expensive-to-compute function which varies most along a few (d) directions, while the other ($n-d$) directions contribute only weakly (up to a small constant γ) to the cost function value. We seek a point $x \in B$ such that $f(x) \leq f_0$. Solving an optimization problem of the form (1) is difficult and, for general functions, convergence can only be guaranteed if the function evaluation set becomes dense over the entire search domain, B, in the limit of an infinite number of function evaluations [1]. The rapid identification of the d directions of maximum variation of the objective function is a key question facilitating more rapid approximate solution of the optimization problem in practical application, and is discussed in detail in §III-A.

Optimization of nonconvex objective functions is a problem of intense interest in many practical engineering problems, such as hydrofoil design optimization [12], which represents a typical challenge problem for this effort, as its objective function $f(x)$, which characterizes the lift/drag ratio of the foil, may be expresses as a function of 10 or more variables, though some of these variables are much more significant in the optimization problem than others (though this is not known a priori). The ultimate goal in such practical problems is to approximate the global minimum of (1) using as few function evaluations as possible.

There are a variety of established techniques for dimension reduction in the optimization setting. Sensitivity analysis [2] is a well-known method by ranking the input parameters due to the measure of their contribution to the objective function. However, some functions may have the most variable directions that are not aligned with the coordinate. Principle component analysis (PCA) [4] is another popular method for dimension reduction by creating new artificial coordinates that are linear combination of the observed variables. PCA could only maintain the largest variation of data points instead of identifying the direction that has most variation of the objective function, which possibly passes the global minimum region. Locally linear embedding [5] identifies the low-dimensional subspace when the high-dimensional data lie on a manifold that embedded in high dimensional space. However, the sub-region of original parameter space is explored by the function values, which indicates that there is a must to design a strategy to project the data from low-dimensional embedding back to original parameter space. In Section III-B we propose a new re-transformation strategy by solving a constrained minimization problem.

Under appropriate assumptions, it is guaranteed that derivative-free methods could converge to a global optimum, but in general they are computationally inefficient since it is a NP-hard problem and many more function evaluations are required. Response surface methods (RSMs) are the most efficient globally-convergent derivative-free optimization methods available today, which iteratively minimize a search function using an interpolant of existing data points, known as the "surrogate", and a model of the "uncertainty" of this surrogate which goes to zero at the evaluated data points. The modern examples of RSMs include Efficient global optimization (EGO) [6], optimization by radial basis function interpolation in trust-regions (ORBIT) [7], the Surrogate-Management-Framework (SMF) [8], and Delaunay-based derivative-free optimization via global surrogates (Δ-DOGS) [9], [10].

The derivative-free scheme upon which the present work

⋆ Dept of MAE, UC San Diego, muz021,bewley@ucsd.edu
⋆⋆ California Institute of Technology, Jet Propulsion Laboratory sralimo@jpl.nasa.gov

is based on is Δ-DOGS, which is a broad family of computationally-efficient RSMs developed to optimize low-dimensional and black-box functions that are both non-convex and computationally intractable. There are already a handful schemes in this family, including schemes designed specifically linear constraints [11], and nonconvex constraints [10].

This paper combines the dimensionality reduction scheme together with Δ-DOGS algorithm to minimize the high-dimensional objective function that has most variation along at most d-directions. We first apply gradient sampling on the response surface to obtain the active subspace. Then Δ-DOGS is applied on the active subspace to identify a low-dimensional minimizer which is potentially close to the global minimum after projection. Lastly a new inverse mapping scheme is proposed to transform the minimizer back into original parameter space by solving an inequality constrained minimization problem.

The paper is structured as follows: Section II briefly reviews the essential ideas of [11], [18], which accelerates a Δ-DOGS search by coordinating it with a Cartesian grid which is successively refined as convergence is approached. Section III explains the new optimization scheme, which combines an active subspace method with our derivative-free optimization scheme (Δ-DOGS). Section IV briefly analyzes the global convergence property of the new algorithm under appropriate assumptions. In Section V, the new algorithm is applied to synthetic optimization problems to illustrate its competitive performance. Conclusions are presented in Section VI.

II. A Brief Review of Δ-DOGS

In this section we briefly review the essential ideas of Δ-DOGS [11], [18]. This paper focuses on the variation of this core algorithms by leveraging the active subspace method, in order to identify (and, preferentially explore) the directions in parameter space with the greatest variability of the objective function. Note that other variants of Δ-DOGS, such as those implementing Cartesian grids to accelerate the convergence rate, as discussed in [11], and those leveraging multivariate adaptive polyharmonic splines (MAPS), as discussed in [12], may also be considered in the present dimension-reduced setting.

The Δ-DOGS algorithm successively determines the location within B with the highest probability to achieve a function value less than or equal to the prescribed target f_0. This approach is realized by minimizing a synthetic (and, cheap-to-evaluate) surrogate model $s_c(x)$, constructed via polyharmonic splines [13] $p(x)$, and the uncertainty function $e(x)$. The approach is akin to the expected improvement and Bayesian optimization algorithms [14], [6].

The local uncertainty function, defined at each iteration as a piecewise quadratic "bump" function within each simplex of a Delaunay triangulation of the evaluated datapoints, reaches its maxima at the circumcenter of each simplex. This uncertainty function has several important foundational properties, such as Lipschitz continuity and twice-differentiability within each simplex.

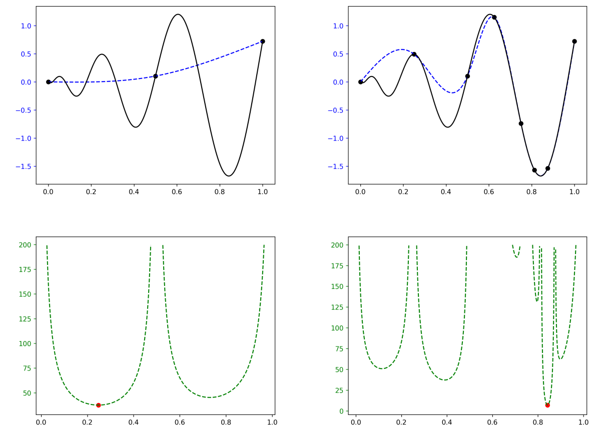

Fig. 1: The essential elements of Δ-DOGS algorithm in different iterations for 1D Schwefel function(17). The upper figures contain: The solid black line indicates the truth function $f(x)$, the blue dotted line indicates interpolant function $p(x)$; The lower figures contain: The green dotted line represents the continuous search function $s_c(x)$, as defined in equation (2). The red circles are the minimizer of $s_c(x)$ as known as the next data point to evaluate.

Definition 1: Consider a set of N datapoints $S=\{x_i\}_{i=1}^N$ over the feasible domain B. The continuous search function $s_c(x)$ is defined as follows:

$$s_c(x) = \begin{cases} \frac{p(x)-f_0}{e(x)} & \text{if} \quad p(x) \geq f_0, \\ p(x)-f_0 & \text{otherwise,} \end{cases} \tag{2}$$

where $p(x)$ is some smooth interpolating function such that $p(x_i)=f(x_i), \forall i \in \{1,\dots,N\}$.

The interpolation $p(x)$, the truth function $f(x)$, and the continuous search function $s_c(x)$ are illustrated in Fig. 1. There are two possible termination scenarios for Δ-DOGS: either the target value f_0 is achievable and Δ-DOGS identifies a point x with function value $f(x) \leq f_0$, or Δ-DOGS conducts infinite number of mesh refinement iterations, ultimately with data points becoming dense over the entire feasible domain.

Definition 2: The Cartesian grid of level L for the feasible domain $B=\{x|a \leq x \leq b\}$, denoted B_L, is defined as

$$B_L = \left\{ x | x_L = a_L + \frac{1}{N}(b_L - a_L)\cdot z_L, \quad z_L \in \{0,1,\dots,2^L\} \right\}$$

The point $x \in B$ quantized onto the grid B_L, denoted x_q, is the nearest gridpoint to the original point x. The quantizer x_q is not necessarily unique. The maximum quantization error $\forall x \in B$ to the mesh grid B_L is simply

$$\delta_L(x) = \max_{x_q \in B_L} |x - x_q| \tag{3}$$

III. Dimension reduction based on the active subspace method

A. Active Subspace method

We now discuss briefly the theory of active subspaces [15]. The general approach is similar to that of principle component analysis (PCA), which implements a proper orthogonal decomposition (POD) on the covariance matrix of the evaluated data to acquire the components with the highest variability. The active subspace method, in contrast, performs this POD on the covariance matrix of the gradient of the objective function.

Consider the scalar function f of the n-dimensional column vector x, whose variability is concentrated in d directions. The gradient $\nabla f(x)$ is also reshaped as a column vector. Denoting the evaluated data point set as $S = (x_1, x_2, \dots, x_N)$, we have

$$f_i = f(x_i),\ \nabla f(x_i) \in \mathbb{R}^n,\ x_i = [0,1]^n. \tag{4}$$

The task is to identify the "active" directions, representing the highest variability of f. A few comments are in order before we review the active subspace method. First, the POD of the gradient outer-product involves an integral with respect to the original coordinates n, which is difficult to solve. To address this, one could use a Monte Carlo simulation to estimate this outer-product. In [15], it is established that the required number of Monte Carlo samples is $M = \alpha \cdot d \cdot \log(n)$, where α is an adjustable parameter.

However, the primary challenge is to approximate the gradient function itself, which is unavailable in derivative-free approaches. [15] provides an error estimate for this approximation using the gradient of the RSMs under the assumption that only a few directions have significant variability. We randomly sampled the gradient of the RSMs $\nabla \hat{f}$ built by Δ-DOGS, subject to a uniform probability density function, to construct an estimate of the covariance matrix in parameter space using a Monte Carlo method. The estimated covariance matrix is

$$C \approx \hat{C} = \frac{1}{M} \sum_{i=1}^{M} \nabla \hat{f}(x_i) \nabla \hat{f}(x_i)^T. \tag{5}$$

The "active" directions of parameter space are then determined by performing a spectral decomposition on this covariance matrix, which is symmetric positive semidefinite, and may thus be decomposed as

$$C = W \Lambda W^T, \tag{6}$$

where $W \in \mathbb{R}^{n\times n}$ and Λ is a diagonal matrix of descending eigenvalues. Since we consider an objective function varying primarily along d directions, the first d orthogonal eigenvectors from W are selected to define the *active subspace*. Note that the corresponding eigenvalues are relatively large, which means there is increased variability along the directions indicated by those eigenvectors. We then identify a significant "gap" in the eigenvalues in Λ, and partition W and Λ accordingly [15] (often capturing a certain minimum "degree of variabiility" with the eigenvalues retained in the active subspace - e.g., 90% or more) such that

$$W = \begin{bmatrix} \underbrace{W_1}_{d \text{ columns}} & \underbrace{W_2}_{n-d \text{ columns}} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \tag{7}$$

Once the spectral decomposition of matrix C is achieved, the original parameter x may then be written as $x = W_1 y + W_2 z$, where y represents the coordinates within the "active" subspace, and z represents the coordinates within the "inactive" subspace (with relatively little variation of f). We then define the search domain within the active subspace as follows

Definition 3: Suppose B denotes the domain of original parameter space. For $\forall x \in B,\ x = W_1 y + W_2 z$. Then the domain $\bar{B}$ of the *active(reduced) subspace* is defined as

$$\bar{B} = \{ y \mid y = W_1^T x,\ \forall x \in B \}. \tag{8}$$

We may then construct the interpolation g in the reduced subspace to approximate the function f. This interpolation is constructed by conditional expectation, as discussed in §III-B, and results in

$$f(x) \approx g(W_1^T x). \tag{9}$$

B. Δ-DOGS with Active Subspace Method

In this section, we consider the problem of identifying a location in the feasible domain B with function value less than or equal to f_0. The objective function is assumed to vary primarily within d directions in the feasible domain.

Definition 4: Assume the function $f(x)$ has at most $d < n$ "active" directions. Suppose x_D is the d-dimensional column vector that represents the value of the input x. For a sufficiently small γ and $\forall x, y \in \mathbb{R}^n$ there exists δ such that, if $\|x_D - y_D\| < \delta$, then $|f(x) - f(y)| \leq \gamma$.

Δ-DOGS with active subspace method algorithm may then be presented in three phases as follows.

1. **First Phase: Active Subspace Method (ASM).** We apply the active subspace method to determine the d directions that have the most variability of the objective function. The original n-dimensional dataset S is mapped to the d-dimensional active subspace $\bar{B}$. Each coordinate of the active subspace $\bar{B}$ is a linear combination of original parameters in B.
2. **Second Phase: Δ-DOGS.** In the second phase, a regular Δ-DOGS optimization is performed on the active subspace $\bar{B}$, to approximate the minimizer $y_r \in \mathbb{R}^d$. This step aims to establish knowledge about which (lower-dimensional) subregion within the original parameter space B is most likely to contain the global minimum.
3. **Third Phase: Δ-DOGS with ASM.** An inverse mapping is then developed to transform the best point from the d-dimensional optimization in the second phase back to the full n-dimensional problem. This inverse mapping requires another response surface constructed with data points in B.

For the second phase, we first construct a new interpolation that is needed for the Δ-DOGS optimization in the active subspace $\bar{B}$. The approximate value $\hat{f}(x_r)$ at the image of evaluated points after mapping is calculated based on conditional expectation of the interpolant value.

Definition 5: The value of the interpolant in the reduced subspace is defined as

$$\begin{gathered} P_r(y) = \hat{f}(y) = \frac{1}{M}\sum_{i=1}^{M} f(x_{e_i}) \\ \text{subject to } W_1^T x_{e_i} = y, \quad y \in \bar{B},\ x_{e_i} \in B. \end{gathered} \tag{10}$$

where $M = \alpha\, d \log(n)$, $\alpha \approx 10$ is an oversampling factor, and the M points x_{e_i} are sampled in a uniformly random way over the inactive subspace in order to approximate the value of f in the original search domain B.

Note that the reduced model $\bar{B}$ is also a manifold, thus another mesh grid scheme is applied in the reduced subspace. Each time the mesh grid L in original parameter space B gets refined, the mesh grid ℓ in the reduced subspace $\bar{B}$ is also refined to accelerate the convergence of Δ-DOGS to the global minimum in reduced model $\bar{B}$.

By implementing Δ-DOGS optimization, we approximate the minimizer y_r of the continuous function $s_c(x)$ in the reduced subspace, which functions as an estimate to indicate which subregion of the full parameter space most likely has the global minimizer that is sought.

As the function evaluations are performed in the original parameter space B, it is necessary to approximate y_r in the original parameter space B. We propose an inverse mapping that transforms y_r into the original parameter space B based on the goal of minimizing the surrogate of objective function in B [16]. This inverse mapping is constructed by solving an inequality constrained minimization described below.

In inequality constrained optimization, the objective function is defined as discrete search function $s_d(x)$ with similar structure to the continuous search function. $s_d(x)$ is constructed by the interpolant function $P(x)$ in B, and a new distance-uncertainty function $u(x)$. The uncertainty function $u(x)$ is the distance of x to its nearest neighbor in the evaluated points set S defined as follow.

Definition 6: Suppose $S = \{x_1, x_2, \dots, x_N\}$ denotes the evaluated points set in original parameter space. For $\forall x \in B$ the uncertainty function $u(x)$ is defined as

$$u(x) = \text{dist}(x, S) = \min_{z \in S} \|x - z\| \tag{11}$$

Then the search function $s_d(x)$ is defined as

$$s_d(x) = \frac{P(x) - f_0}{\text{dist}(x, S)} = \frac{P(x) - f_0}{\min_{z \in S} \|x - z\|} \tag{12}$$

It is obvious that the distance-uncertainty function $u(x)$ is continuous and differentiable inside the Voronoi cell of every evaluated point $x \in S$ [18]. The key properties of $u(x)$ are: 1) $u(x) \geq 0\ \forall x \in B$, and $u(x_i) = 0\ \forall x_i \in S,\ i = \{1, \dots, N\}$; 2) Since the point-wise distance in B is bounded, and $\max u(x)$ is achieved on the boundaries of box domain B, thus $u(x)$ is Lipschitz continuous with Lipschitz constant L_u.

$$\|u(x) - u(x')\| \leq L_u \|x - x'\|,\ \forall x, x' \in B \tag{13}$$

Algorithm 1 Strawman of Δ-DOGS with ASM

0. Initialize $k = 0$, L, ℓ and the initial set of datapoints S^0, and calculate $f(x_i)$ for all $x_i \in S_0$.
1. Calculate or update the interpolating function $p^k(x)$ for all the points in S^k.
2. By equation (5), calculate or update the uncentered covariance-like matrix C and the coordinate transformation matrix W_1^k.
3. By Definition 5, establish the interpolating function $P_r^k(x)$ in reduced model, minimize the continuous search function (2) to obtain y_r^k as a minimizer in reduced model.
4. Solve the inequality constrained minimization (14) to obtain x^k as a minimizer of the response surface.
5. Determine z^k as the quantization of x^k on B_{L_k}. If $z_k \notin S^k$, $S^{k+1} = S^k \cup z_k$; otherwise, refine the mesh by increasing $L_k = L_k + 1$ and $\ell_k = \ell_k + 1$. Increase $k = k + 1$.
6. Repeat steps 1-5 until a point x is found with $f(x) \leq f_0$.

Definition 7: Determine the minimizer of Δ-DOGS y_r, and establish the discrete search function as stated in Definition 6. Given a slack tolerance variable ε, the inequality constrained minimization is defined as follow

$$\begin{gathered} \min\ s_d(x) = \frac{P(x) - f_0}{\text{dist}(x, S)} \\ \text{with } \|W_1^T x - x_r\| \leq \varepsilon \end{gathered} \tag{14}$$

This inequality constrained optimization is solved by sequential least-square quadratic programming. The initial guess is defined by $x_0 = W_1 y_r$. The slack variable ε is a user-defined variable that how much amount of variation that we could tolerate. The larger ε indicates that we allow searching more globally in B.

We have now presented all of the essential elements of the new algorithm. The result is summarized as Algorithm 1.

IV. Convergence Analysis

All the convergence proofs of Theorems, Lemma related to Algorithm 1 can be found in `http://fccr.ucsd.edu/pubs/zab18.pdf`. In this section, we analyze the convergence properties of Algorithm 1. Under the appropriate assumptions, we will establish the following property:

Target achievability: If the target is achievable, the algorithm will either: (a) find the feasible point with objective function equal or less than the target f_0 in a finite number of iterations, or (b) generate an infinite sequence of points that contain a point with function value equal to f_0. We first establish the following theorem based on Definition 4.

Theorem 1: Suppose the perturbation of the dominant directions is small, $\|x_D - x'_D\| \leq \delta$. Let $x = W_1 y + W_2 z$ and

$x' = W_1 y' + W_2 z'$. Then the interpolant in active subspace is Lipschitz with constant L_{P_r}.

$$|P_r(y) - P_r(y')| \leq L_{P_r} \|y - y'\| \tag{15}$$

where $L_{P_r} = \frac{2C_1(1+N^{-\frac{1}{2}})\varepsilon_0 + \gamma}{\delta_0}$.

The uncertainty function $u(x)$ also keeps the properties such as continuous and twice-differentiable as needed to prove the target achievability in [9]. It is established in [11] that Δ-DOGS is capable to converge to the point with the target value f_0. The results is shown in the Theorem 2.

Theorem 2: Suppose the Definition 4 holds and the Lipschitz continuous reduced interpolant $P_r(x)$ is built up as stated in Definition 5. Algorithm 1 will converge to the global minimum in the feasible domain $\bar{B}$.

In active subspace $\bar{B}$, Δ-DOGS could approach the target value f_0, i.e. for sufficiently many iterations k, we have

$$|P_r(x_r^k) - f_0| < \varepsilon_2 \tag{16}$$

We may now prove that Algorithm 1 is also target achievable in the original parameter space B under Def 4.

Theorem 3: Suppose the objective function has most variability along d directions. The target value f_0 and box domain B are given. The Algorithm 1 is target achievable if the reduced interpolant $P_r(x)$ is Lipschitz continuous.

For sufficiently small ε_0 and ε_2, the Algorithm 1 globally converges to the target value with finite iterations.

V. Results

In this section, Δ-DOGS with ASM algorithm has been applied to the following synthetic functions. The initial size of the Cartesian grid for each coordinate is set to be 8. The Algorithm 1 will continue the search until 4 times of mesh refinement have been performed. We initialize $\varepsilon = 0.2$ and reduce it to zero by decreasing 0.001 for each iteration.

The performance of the Δ-DOGS with ASM algorithm is measured by the number of function evaluations and the relative error defined as follows. Suppose the best minimum point obtained until iteration k is defined as the candidate point at iteration k. Let f_{min} denotes the function value of the candidate point and f_0 denotes the global optimum value. The relative error is defined as the ratio of the subtraction of best minimum obtained and the global minimum to the global minimum.

The initial data points in S^0 are constructed with $3n+3$ points that are uniformly drawn from the parameter space B. The number of dimension of active subspace is set to be one. To approximate the first d eigenvectors of C, [15] recommended to have M samples of gradient sampling on the response surface subject to uniform distribution. Here $M = \alpha d \log(n)$ and α is an oversampling factor which is fixed as 10, the other parameters are set as $d = 1$ and $n = 10$. Thus we have As we have mesh grid refined, we would increase the gradient samples as there are more grid points in the parameter space. Thus when the finest grid is achieved, the number of gradient sampling is computed as $M = \alpha d \log(n)$, $L_k = 1280$.

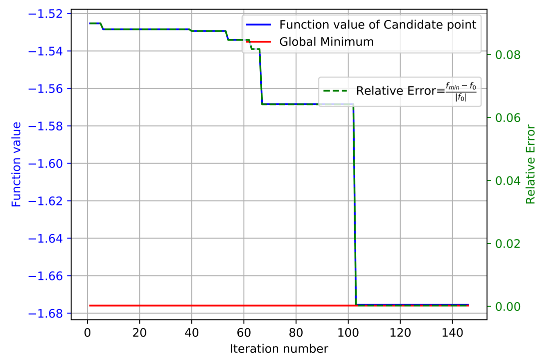

Fig. 2: $f_1(x)$ on Candidate point.

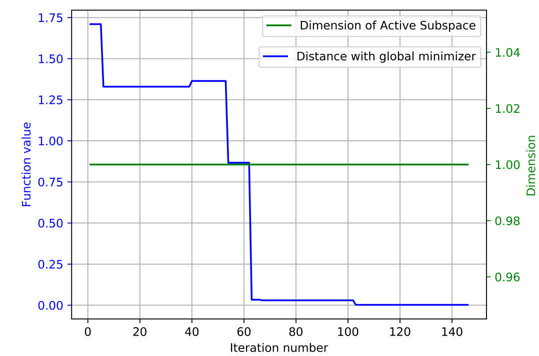

Fig. 3: Distance of candidate points to the global minimum.

The first test function(17) is constructed mainly by Schwefel function and the quadratic function which only makes small contribution to the objective function.

$$f_1(x) = -\frac{x_1}{2}\sin(500|x_1|) + \sum_{i=2}^{10}(0.001)\cdot i \cdot x_i^2 \tag{17}$$

It has several properties: (a) The coordinate x_1 has most variability; (b) It is continuous, nonconvex and twice-differentiable in the box domain $x \in [0,1]^{10}$; (c) The minimizer is $x^* = [0.8419, 0, \ldots, 0]$ and the target value is $f_0 = -1.675936$. We applied Algorithm 1 on the first test function with $n = 10$. The target value $f_0 = -1.6759$ is achieved by 103 iterations with relative error 0.025%. The distance of candidate point to the global minimizer also converges to zero as shown in Fig. 2 and Fig. 3.

The second test function(18) is a combination of exponential function, Rosenbrock function and quadratic function.

$$f_2(x) = e^{0.2x_1} + e^{0.2x_2} + 10(x_2 - x_1^2)^2 + (x_1 - 1)^2 + 0.001 \cdot \sum_{i=3}^{10}(x_i - 0.1\cdot i)^2 \tag{18}$$

The second test function has several properties:(a) The most variability direction is along the combination of two coordinates x_1 and x_2; (b) It is continuous, convex and differentiable; (c) The minimizer is $x^* = [0.512, 0.723, 0.3, 0.4, \ldots, 1]$ and the target value $f_0 = 2.34128$. The target value f_0 is achieved by 413 iterations with relative error 0.0532%.

Table I shows the results of applying the Δ-DOGS with ASM to the above two test problems.

Test function	No. param.	stopping criterion	final error (ε)	No. Eval. for 1% error
(17)	10	146	0.025%	52
(18)	10	465	0.0532%	246

TABLE I: Experiment results of Algorithm 1.

VI. Conclusions

In this paper we have extended the active subspace method to the Delaunay-based derivative-free optimization algorithm scheme, Δ-DOGS, to handle high-dimensional nonconvex problems. Δ-DOGS has been performed on the active subspace to enable low-dimensional surrogate search. Once the minimizer is obtained, we proposed an inverse mapping scheme by solving an inequality optimization problem. This approach, Algorithm 1, has two main modifications as compared with the original Δ-DOGS algorithm:

- Previously Δ-DOGS is restricted by the number of design parameters because of the unaffordable computational cost to construct the Delaunay triangulation in high-dimensional parameter space. Δ-DOGS with ASM avoids the huge cost by transforming all data points Under appropriate assumptions on the objective function, Algorithm 1 is asymptotically convergence provable to identify where the objective function achieves the target value.
- We proposed a new inverse mapping method that projects the points in active subspace back into the original parameter space. This process is achieved by solving an optimization problem constructed on the global surrogate with inequality constraints. Our results are compelling compared with the normal linear mapping routines.
- The objective function could have different variance along different coordinates. However, Δ-DOGS scales poorly with the dimension of the objective function because it treats each coordinate with the same importance in their contribution to the objective function. Δ-DOGS with ASM could mitigate this effect by projecting the original and leave out the directions that have small contribution to the function.

In future work, this framework will be applied to engineering-based problems such as the design of hydrofoil in high-speed boats and localization of aerial vehicle in the GPS denied environment. Most of the engineering-based problems where the objective function with its most variability along more than one direction. Moreover, the dimension of the active subspace could be increased such that more complicated objective functions will be handled by Δ-DOGS with ASM. The vertices of active subspace $\bar{B}$ could be found by algorithm provided in [17] subject to linear mapping. Also, the presented Delaunay-based optimization scheme with active subspace method will be incorporated into Δ-DOGS family of schemes [9], [11] to enable such optimization method to handle problems with more variables.

Acknowledgement

We thank Pooriya Beyhaghi and Clark Briggs for their critical insights, and AFOSR FA 9550-12-1-0046 and Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration in support of this work.

References

[1] Stephens, C. P., & Baritompa, W. (1998). Global optimization requires global information. Journal of Optimization Theory and Applications, 96(3), 575-588.

[2] Saltelli, Andrea, Karen Chan, and E. Marian Scott, eds. Sensitivity analysis. Vol. 1. New York: Wiley, 2000.

[3] Adjengue, L., Audet, C., & Yahia, I. B. (2014). A variance-based method to rank input variables of the mesh adaptive direct search algorithm. Optimization Letters, 8(5), 1599-1610.

[4] Wold, Svante, Kim Esbensen, and Paul Geladi. Principal component analysis. Chemometrics and intelligent laboratory systems 2.1-3 (1987): 37-52.

[5] Roweis, Sam T., and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. science 290.5500 (2000): 2323-2326.

[6] Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.

[7] Wild, S. M., Regis, R. G., & Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions. SIAM Journal on Scientific Computing, 30(6), 3197-3219.

[8] Audet, C., Kokkolaras, M., Le Digabel, S., & Talgorn, B. (2018). Order-based error for managing ensembles of surrogates in mesh adaptive direct search. Journal of Global Optimization, 70(3), 645-675.

[9] Beyhaghi, P., Cavaglieri, D., & Bewley, T. (2015). Delaunay-based derivative-free optimization via global surrogates, part I: linear constraints. Journal of Global Optimization.

[10] Alimo, S. R., Beyhaghi, P., & Bewley, T. R. (2019). Delaunay-Based Global Optimization in Nonconvex Domains Defined by Hidden Constraints. In Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems (pp. 261-271). Springer, Cham.

[11] Beyhaghi. P., & Bewley. T.: Implementation of Cartesian grids to accelerate Delaunay-based derivative-free optimization. Journal of Global Optimization, 69(4), 927-949.

[12] Alimo, S., Beyhaghi, P., Meneghello, G., & Bewley, T. (2017): Delaunay-based optimization in CFD leveraging multivariate adaptive polyharmonic splines (MAPS). In 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (p. 0129).

[13] Wahba, Grace. Spline models for observational data. Society for industrial and applied mathematics, 1990.

[14] Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. Journal of the American Statistical Association, 103(483), 1119-1130.

[15] Constantine, P. G., Dow, E., & Wang, Q. (2014). Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4), A1500-A1524.

[16] Lukaczyk, T. W., Constantine, P., Palacios, F., & Alonso, J. J. (2014). Active subspaces for shape optimization. In 10th AIAA Multidisciplinary Design Optimization Conference (p. 1171).

[17] Mattheiss, T. H., & Schmidt, B. K. (1980). Computational results on an algorithm for finding all vertices of a polytope. Mathematical Programming, 18(1), 308-329.

[18] Alimo, S. R., Beyhaghi, P., & Bewley, T. R. (2017, December). Optimization combining derivative-free global exploration with derivative-based local refinement. In Decision and Control (CDC), 2017 IEEE 56th Annual Conference on (pp. 2531-2538). IEEE.