2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Delaunay-based Derivative-free Optimization via Global Surrogates with Safe and Exact Function Evaluations

Muhan Zhao* Shahrouz Ryan Alimo† Pooriya Beyhaghi‡ Thomas R. Bewley*

Abstract— **Automatic parameter tuning is an important task in real-world (experimental) optimization, in order to safely (e.g. without crashing) explore an unknown environment. Example includes smooth open-loop control of trajectory planning where collisions must be avoided. Delaunay-based derivative-free optimization via Global Surrogate (Δ-DOGS) algorithms are a family of response surface method that efficiently and globally minimizes black-box, computationally expensive, nonconvex optimization problems; however, the challenge of restricting all function evaluations to be "safe" during the parameter tuning process has not yet been addressed in this family of algorithms. In this work, we develop a new, safety-constrained variant of this approach, dubbed S-DOGS, to automatically learn the safe region of parameter space, while simultaneously characterizing and optimizing the utility function under consideration, under the assumption that the underlying safety constraints are Lipschitz continuous and the safe region is connected and compact. Theoretical analysis and experimental results are provided to demonstrate that the resulting method is both efficient in terms of the rate of convergence with the number of function evaluations performed, and guaranteed to converge to the global minimum while respecting the safety constraints.**

I. Introduction

In autonomous systems, it is often necessary to tune parameters in order to optimize towards a given objective. However, users often do not know in advance the region(s) of the otherwise feasible parameter space which, if explored during the optimization process, could lead to severe damage of the experimental system. For such problems, it is useful to develop algorithms that automatically minimize a given performance measure with unknown mathematical form, while assuring at each function evaluation that the safety of the system is guaranteed [9]. Real-world applications of such problems include robot motion control, in which physical collisions must be avoided.

The idea of safe exploration was considered by [2] [3] [7] [13]and [14]. [9] proposed GP-UCB algorithm which maximizes the upper confidence bound of Gaussian process. Although GP-UCB automatically plays the tradeoff between exploitation and exploration, the safety issue has not been yet addressed. [11] proposed the SAFEOPT algorithm which implements Gaussian processes to safely optimize an objective function by sampling at the most uncertain point in the union of potential maximizer and the potential expansion sets. While SAFEOPT proves to converge to the global minimum within the reachable safe set, it might perform far too many function evaluations, since this paper only considers making function evaluation at the point with the maximum uncertainty in potential maximizers set and safe region expander set. The efficiency should be improved since the uncertainty information may not sufficiently capture the behavior of the underlying function. Moreover, in this approach, only the objective function itself is considered as the safety constraint. Subsequently, [4] proposed the SAFEOPT-MC algorithm to consider multiple safety constraints via Gaussian processes(GP). [12] proposes STAGEOPT algorithm that separates the safe learning task into exploration and exploitation phases. STAGEOPT is more efficient since the objective function considered during exploitation of STAGEOPT also takes a model of the utility function into account.

Response Surface Methods (RSMs) are an efficient family of derivative-free global optimization methods. Delaunay-based derivative-free optimization via global surrogate (Δ-DOGS) is a highly-efficient modern variant of RSM that, under the appropriate assumptions, guarantees convergence to the global solution for nonconvex optimization problems with computationally expensive objective functions [1][5].

In this paper, the framework of Δ-DOGS is extended such that user-supplied unknown safety constraints are ensured to be satisfied during each iteration of the optimization process. The algorithm developed, dubbed Safety-guaranteed Derivative-free Optimization via Global Surrogate (S-DOGS), can be implemented to automatically, efficiently, and safely learn the boundary of the underlying safe region, while simultaneously identifying the global minimizer of the objective function. Compared with these existing safe learning algorithms which rely on GP model, the novelty of S-DOGS algorithm is that it does not need the well specified kernel function which requires a fine-tuning process in order to guarantee safety and efficiency.

The structure of this paper is as follows: Section II briefly reviews the main framework of Δ-DOGS. Section III introduces the new S-DOGS algorithm. Section IV conducts the theoretical convergence analysis of the algorithm. Section V demonstrates the performance of the algorithm on a synthetic test problem as well as a simulation of quadrotor trajectory following problem. Section VI presents conclusions.

II. Review of Δ-DOGS with mesh refinement

In this section we briefly review the essential parts of Δ-DOGS[5]. Recently Δ-DOGS has been extended to address different problem settings, such as convex constraints (Δ-DOGS(C))[6], nonconvex or disconnected regions (Δ-DOGS(Ω))[1], implementing Cartesian grid to accelerate

*Flow Control Lab, Dept of MAE, UC San Diego, bewley@ucsd.edu and muz021@ucsd.edu
†Jet Propulsion Laboratory, California Institute of Technology sralimo@jpl.caltech.edu
‡ASML Inc, p.beyhaghi@gmail.com

978-1-7281-1398-2/19/$31.00 ©2019 IEEE

convergence (Δ-DOGS(Z)). [15] proposed an algorithm dealing with higher-dimensionality by applying dimension reduction of active subspace method to project Δ-DOGS onto a lower-dimensional embedding for parameters search.

The problem considered in Δ-DOGS is defined as

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

Δ-DOGS builds up a response surface which is iteratively minimized to provide information of where the minimizer of the objective function locates with the highest probability. This response surface is constructed based on an interpolant $p(x)$, which curves the fidelity of the truth function, and an artificially designed uncertainty function $e(x)$, which quantifies how much uncertain the unexplored regions have.

Definition 1: Suppose $S_k = \{x_i\}_{i=1}^k$ is the set of evaluated data points until k-th iteration. The Delaunay triangulation Δ is determined based on S_k and vertices V of the box domain B. Suppose r_i and o_i are the circumradius and the circumcenter of the circumsphere of each Delaunay simplex Δ_i respectively. In each Delaunay simplex Δ_i, the local uncertainty function $e_i(x)$ is defined as,

$$e_i(x) = r_i^2 - ||x - o_i||_2^2, \quad x \in \Delta_i. \tag{2}$$

The constant K continuous search function is defined as the surrogate model which is to be minimized at each iteration, where the scalar parameter K is introduced as the tradeoff between the exploitation of the underlying truth function and the global exploration of the uncovered regions in the domain.

Definition 2: Consider S_k as the evaluated data points and the uncertainty function is constructed in Definition 1. Assume that $p(x)$ is a robust interpolant based on S_k, then $p(x) = f(x)$ for $x \in S_k$. The constant K continuous search function is defined as

$$s(x) = p(x) - K \cdot e(x), \quad x \in B. \tag{3}$$

As Δ-DOGS proceeds with minimizing equation (3), the minimizer x_k is quantized onto the Cartesian grid defined as follows.

Definition 3: The Cartesian grid over the entire feasible box domain B with level ℓ is denoted as

$$M_\ell = \left\{ x \in B |\ x = a + \frac{j}{2^\ell}(b-a)e_i, 1 \leq i \leq n, 0 \leq j \leq 2^\ell \right\} \tag{4}$$

The finest mesh grid is defined as $M_{\ell_{\max}}$ with highest mesh grid level $\ell_{\max}$. For each mesh grid level ℓ, the mesh size δ_ℓ is computed as $\delta_\ell = \frac{1}{2^\ell}$. At each step, the mesh grid M_ℓ quantizes x_k as $z_k = M_\ell(x_k)$.

III. S-DOGS: Safe learning with Δ-DOGS

A. Problem Statement

In this paper, we consider the following optimization problem,

Definition 4: Consider minimizing a black-box and non-convex function $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$ with m unknown safety functions $\psi(x) : \mathbb{R}^n \mapsto \mathbb{R}^m$,

$$\text{minimize } f(x) \text{ with } \ x \in \Sigma : B \cap C \tag{5}$$

$$\text{where } \ B = \{x | a \leq x \leq b\}, \ C = \{x | \psi(x) \geq 0\} \tag{6}$$

where $\psi(x) = [\psi_1(x), ..., \psi_m(x)]^T$ and the notation $\psi(x) \geq 0$ denotes that every function $\psi_i(x)$ holds the inequality $\psi_i(x) \geq 0$. $\psi_i(x)$ can be a function or even a subroutine that performs a complex specific task and could return the safety function values given a set of optimization parameters. The safety function $\psi_i(x) : \mathbb{R}^n \mapsto \mathbb{R}$ is Lipschitz continuous with a finite constant L_i, $i \in \{1, ..., m\}$,

$$|\psi_i(x_1) - \psi_i(x_2)| \leq L_i ||x_1 - x_2||, \ \forall x_1, x_2 \in \Sigma \tag{7}$$

We assume that the closed-form expression of $f(x)$ and $\psi(x)$ are unknown but both of them could be evaluated given parameters $x \in B$. Both of the objective function $f(x)$ and safety functions $\psi(x)$ are exact measurements which contain no noise. Due to the fact that all Lipschitz constants L_i of $\psi_i(x)$ are finite real positive numbers, there exists an upper bound $\bar{L}$ such that $\bar{L} \geq L_i, \forall 1 \leq i \leq m$. Although the safety functions are unknown, we assume that the safety function could be estimated using Lipschitz continuity property. In this paper we consider that the underlying safe region where the safety constraints $\psi(x) \geq 0$ are satisfied should be compact and connected while convexity is not necessarily required.

To set up the problem, we assume that users have some priori knowledge about the safety conditions of the system which could offer a safe initial parameters $x_0 \in \Sigma$.

B. Overview

The general idea of S-DOGS algorithm is to implement derivative-free optimization scheme with safety functions $\psi(x)$ satisfied for the parameter sampling at each iteration. The goal is to iteratively learn the boundary of the underlying safe region as well as to minimize the unknown objective $f(x)$ without violating the unknown safety functions $\psi(x)$.

S-DOGS separates the optimization process into two phases: At the beginning, the algorithm tends to sample the parameters far apart with each other to efficiently explore the safety of the domain. It is intuitive to primarily have a general knowledge about which part of the parameter space is classified as safe on a coarse mesh grid. Once the known safe region is established on the current mesh, the algorithm focuses on minimizing the surrogate model to exploit inside the known safe region. Eventually the algorithm would identify a region of interest to locally refine the utility function on the finer mesh grid.

C. Safe Region Exploration

The exploration of the safe region is carried on using the Lipschitz continuity of $\psi(x)$. Given the safe initial point x_0, there exists an open hypersphere of x_0, denoted as Ψ_0, such that the points inside Ψ_0 are labeled as safe points. As the algorithm evaluates new points, the known safe region will be expand to include the point that has not yet been labeled as safe on the current mesh grid.

Definition 5: Suppose a safe initial point x_0 is pre-defined and S_k denotes the evaluated data points. The convex hull of S_k is denoted as $CH(S_k)$. The **known safe region** Ψ_k^ℓ based on the current mesh grid M_ℓ at iteration k is defined

as

$$\Psi_k^\ell = S_k \cup \{x \in B \cap M_\ell |\ \exists x' \in S_k \\ \text{s.t. } \psi(x') - \bar{L}||x - x'||_2 \geq 0\} \tag{8}$$

The **known unsafe region** $(\Psi_k^\ell)^c$ is denoted as the complement of Ψ_k^ℓ within B, $(\Psi_k^\ell)^c = B \backslash \Psi_k^\ell$.

As the algorithm samples the points close to the boundary of Ψ_k^ℓ, its boundary will be further spreading out. Eventually, as the iteration goes to infinity, each expanding step of the safe exploration will become smaller and smaller. There exists a limit of the known safe region called as maximum reachable safe set.

Definition 6: Given a safe initial point x_0 and the finest mesh grid level $\ell_{\max}$ the maximum reachable safe set $\bar{\Psi}(x_0)$ is defined as

$$\bar{\Psi}(x_0) = \lim_{k \to \infty} \Psi_k^{\ell_{\max}} \tag{9}$$

For simplicity, we use $\bar{\Psi}$ to refer the maximum reachable safe set $\bar{\Psi}(x_0)$.

Note that given different safe initial point x_0, the maximum reachable safe set $\bar{\Psi}(x_0)$ might be different. $\bar{\Psi}$ is connected but might be nonconvex. It is intuitive that based on safety-guarantee it is not possible to leap from one safe region to another safe region which are separated by an unsafe region between them. As a result, the goal of the algorithm is seeking the global minima within $\bar{\Psi}$.

$$x^* = \arg\min_{x \in \bar{\Psi}} f(x)$$

To expand Ψ_k^ℓ more efficiently, it is necessary to estimate the safety conditions of the unevaluated safe points x inside Ψ_k^ℓ. This is accomplished by building a smooth interpolant $q_i(x)$ for each safety function $\psi_i(x)$ over the entire domain B.

Definition 7: Given the evaluated data points S_k. Compute the interpolant $q_i(x)$ for each safety functions $\psi_i(x)$. $r(x_1)$ and $\hat{r}(x_2)$ denote the safe radius of $x_1 \in S_k$ and the estimated safe radius of $x_2 \in \Psi_k^\ell \setminus S_k$ respectively.

$$r(x_1) = \frac{\min\{\psi_i(x_1)\}_{i=1}^m}{\bar{L}}, \hat{r}(x_2) = \frac{\min\{q_i(x_2)\}_{i=1}^m}{\bar{L}} \tag{10}$$

The expansion set is a bunch of points inside the known safe region which have not been evaluated yet. Performing function evaluation on such points could possibly expand the boundary of Ψ_k^ℓ based on current mesh grid.

Definition 8: Given the evaluated data points S_k and the known safe region Ψ_k^ℓ on the mesh grid M_ℓ, the expansion set G_k^ℓ is defined as

$$G_k^\ell = \left\{x \in \Psi_k^\ell \mid P_k^\ell(x) > 0,\ \hat{r}(x) \geq \delta_\ell\right\} \\ \text{where } P_k^\ell(x) = \left|\{y \in (\Psi_k^\ell)^c \mid \right. \\ \left. q(x) - \bar{L}||x - y|| \geq 0,\ \hat{r}(y) \geq \delta_\ell\}\right| \tag{11}$$

The function $P_k^\ell(x)$ quantifies the cardinality of how many unsafe points could possibly be classified as safe if x is safely sampled. The restriction $\hat{r}(x)$ and $\hat{r}(y)$ are proposed to efficiently sample the points on the current mesh grid. If $\hat{r}(x)$ is too small, the algorithm tends not to evaluate x because they can not expand Ψ_k^ℓ on M_ℓ. To efficiently expand Ψ_k^ℓ, each iteration S-DOGS selects the point x_k that has the largest value of $P_k^\ell(x)$.

$$x_k = \arg\max_{x \in G_k^\ell} P_k^\ell(x) \tag{12}$$

The construction of the Delaunay triangulation over the entire parameter space B requires the vertices V of the box domain B. However, user usually may not have much knowledge about the safety conditions at boundary points V and thus those vertices will often be classified in the unsafe region $(\Psi_k^\ell)^c$. Therefore, the Delaunay triangulation has two different kinds of Delaunay simplex.

Definition 9: Consider S_k as the evaluated data points and V as the vertices of box domain B. At iteration k, the Delauanay triangulation Δ_k over points $S_k \cup V$ has two types of Delaunay simplex defined as

- The **interior Delaunay simplex** Δ_k^Ψ: Lies inside $CH(S_k)$, all vertices of Δ_k^Ψ have already been evaluated;
- The **exterior Delaunay simplex** $\Delta_k^{\Psi^c}$: Lies outside of $CH(S_k)$, there exists at least one vertex of $\Delta_k^{\Psi^c}$ which has not yet been evaluated.

D. Uncertainty Function

Within Δ_k^Ψ, we expect that the uncertainty of the unexplored region approaches maximum at the circumcenter of interior Delaunay simplex. However, in $\Delta_k^{\Psi^c}$ there exists at least one vertex about which we do not have any safety information. Therefore, we expect the uncertainty increases as a bumpy shape when driving away from the evaluated data points towards unsafe vertices of $\Delta_k^{\Psi^c}$. A new uncertainty is proposed for exterior Delaunay simplex.

Definition 10: Consider S_k as the evaluated data points. Inside the interior Delaunay simplex Δ_k^Ψ, the uncertainty function is defined in Definition 1. In contrast, the uncertainty function in the exterior Delaunay simplex with parameters $0 < b < 1$ and $c > 0$ is defined as follows,

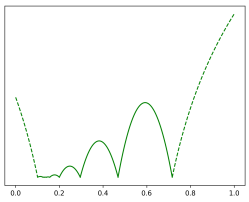
$$e(x) = (||x - \hat{x}|| + c)^b - c^b \\ \text{where } \hat{x} = \arg\min_{x' \in S_k} ||x' - x||_2 \tag{13}$$

To balance the search in safe region and unsafe region, it is critical to set the magnitude of two uncertainty functions within approximately the same range. If the uncertainty function in the exterior Delaunay simplex is much larger than the uncertainty in the interior Delaunay simplex, it is motivated to explore the region close to the boundary of the safe region, and vice versa. More details about the procedures to determine the parameters b and c could be found in Lemma 1. Illustrations of $e(x)$ are shown in Fig. 1.

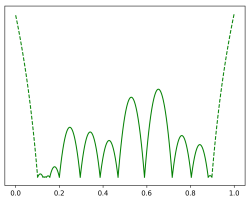
E. Exploitation

After obtaining a general knowledge of the safe region on the current mesh grid, the algorithm iteratively identifies the promising parameters via minimizing the surrogate $s(x)$ constrained with safety functions.

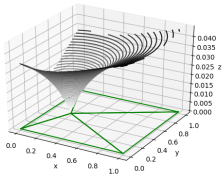
Definition 11: Consider S_k as the evaluated data points and V as the vertices of B. The Delaunay triangulation is constructed based on $S_k \cup V$. Construct the uncertainty function defined in equations (2) and (13) respectively based



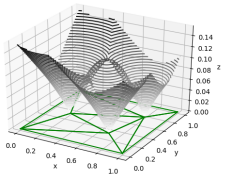
(a) 1D illustration at $|S| = 9$



(b) 1D illustration at $|S| = 14$



(c) 2D illustration at $|S| = 1$



(d) 2D illustration at $|S| = 3$

Fig. 1. The illustration of $e(x)$. S denotes the set of evalauted data points. In top figures, $e(x)$ inside Δ_k^{Ψ} is denoted in solid green line, $e(x)$ in $\Delta_k^{\Psi^c}$ denoted in dotted green line. Bottom figures: 2D illustration of $e(x)$ within interior and exterior Delaunay simplices. For $x \in S$, $e(x) = 0$.

on different types of Delaunay simplices. The safe constrained surrogate $s(x)$ is defined as

$$\begin{aligned} &\min \;\; s(x) = p(x) - K \cdot e(x), \\ &\text{s.t. } \hat{\psi}(x) = \max_{x' \in S} \; \{\psi(x') - \bar{L}||x - x'||\} \geq 0, \end{aligned} \tag{14}$$

The framework of the safe learning Delaunay-based algorithm S-DOGS is shown in Algorithm 1. There are 3 different type of iterations in Algorithm 1:

- *Safe exploration sampling*: The sampling point x_k is determined by G_k^{ℓ} in Line 8 of Algorithm 1. The known safe region Ψ_k^{ℓ} will be expanded by sampling at site x_k.
- *Exploitation sampling*: Either G_k^{ℓ} is empty or the maximum uncertainty in G_k^{ℓ} is less than ε. S-DOGS exploits the interior part of Ψ_{k-1}^{ℓ} via minimizing the surrogate model in Line 10 of Algorithm 1.
- *Mesh refinement sampling*: The sampling point x_k already exists in S_{k-1} which leads to the mesh refinement and the increase of K.

IV. Convergence Analysis

In this section, we only provide part of the essential lemmas and theorems for S-DOGS convergence analysis due to the restriction of pages. More detailed proof could be found at `http://fccr.ucsd.edu/pubs/zapb19.pdf` and be refined in a future version of this work.

Theorem 1: Given S_k as the set of evaluated data. Assume that the objective function $f(x)$ and the interpolation $p_k(x)$ are twice-differentiable and Lipschitz continuous with constant L_f and L_p respectively. Assume that the uncertainty function outside of $CH(S_k)$ is defined in equation (13) with parameters b and c found in Lemma 1. Assume that K is chosen such that Lemma 6 and 7 are satisfied. Suppose Lemma 5 is satisfied and define x_k and x^* as the minimizer

Algorithm 1 S-DOGS algorithm

1: **Input:** $S_0 = x_0$; Measurements $f(x_0)$ and $\psi(x_0)$; Box domain B; Safe Lipschitz bound $\bar{L}$; K and mesh grid M_0; Tolerance ε.
2: **repeat**
3: Calculate (or, update) the interpolant $p_k(x)$ of $f(x)$ and $q_k(x)$ for $\psi(x)$ over S_k.
4: Calculate (or, update) the Delaunay triangulation Δ_k over $S_k \cup V$.
5: Calculate b and c for $e(x)$ in $\Delta_k^{\Psi^c}$ using Lemma 1.
6: Calculate (or, update) the known safe region Ψ_k^{ℓ} and expansion set G_k^{ℓ}.
7: **if** $G_k^{\ell} \neq \emptyset$ and $\max_{x \in G_k^{\ell}} e(x) > \varepsilon$ **then**
8: $x_k = \arg\max_{x \in G_k^{\ell}} P_k^{\ell}(x)$
9: **else**
10: $x_k = \arg\min_{x \in \Psi_k^{\ell}} s(x)$
11: **end if**
12: **if** $z_k = M_{\ell}(x_k) \in S_{k-1}$ **then**
13: Set $\ell = \ell + 1$, $K \leftarrow 2K$.
14: **else**
15: Evaluate $f(z_k)$ and $\psi(z_k)$.
16: **end if**
17: **until** ℓ achieves $\ell_{\max}$ or target value achieved.

to constrained nonlinear programming (14) at iteration k and the global minimizer inside $\bar{\Psi}$ respectively. Then

$$\begin{aligned} &0 \leq f(\hat{x}) - f(x^*) \leq \varepsilon_k, \quad \text{with} \;\; \hat{x} \in S_k, \\ &\text{where} \;\; \hat{x} = \arg\min_{x \in S_k} ||x - x_k||_2, \\ &\varepsilon_k = (L_p + 2KR_{\max})\delta_k, \quad \delta_k = ||\hat{x} - x_k||_2 \\ &\text{and} \;\; \psi(x_k) \geq 0, \;\; \psi(\hat{x}) \geq 0 \end{aligned} \tag{15}$$

Proof: Lemma 2 shows that the uncertainty function is Lipschitz continuous with $2R_{\max}^k$. Since $p_k(x)$ is also Lipschitz continuous with L_p, for $\hat{x}$ and x_k,

$$\begin{aligned} &|p_k(\hat{x}) - p_k(x_k)| \leq L_p\delta_k, \\ &|e_k(\hat{x}) - e_k(x_k)| \leq 2R_{\max}^k\delta_k, \\ &|s_k(\hat{x}) - s_k(x_k)| \leq (L_p + 2KR_{\max}^k)\delta_k. \end{aligned} \tag{16}$$

Since $\hat{x} \in S_k$, it is obvious that $s_k(\hat{x}) = p_k(\hat{x}) = f(\hat{x})$, combining these inequalities in equation (16) gives

$$f(\hat{x}) \leq s_k(x_k) + (L_p + 2KR_{\max}^k)\delta_k \tag{17}$$

Since x^* either stays in Δ_k^{Ψ} or $\Delta_k^{\Psi^c}$, by Lemma 6 and 7, we have $s_k(x_k) \leq f(x^*)$. Finally, we have the inequality,

$$f(\hat{x}) \leq f(x^*) + (L_p + 2KR_{\max}^k)\delta_k \tag{18}$$

As the algorithm proceeds, Algorithm 1 determines a sequence of global minimizers of $s_k(x)$, denoted as $\{x_k\}$. By the Bolzano-Weierstrass theorem and the search domain $B \in \mathbb{R}^n$ is compact, there exists a subsequence of $\{x_k\}$ that converges to a limit point, thus we have $\delta_k = ||\hat{x} - x_k||_2$ converges to zero. As a result, Algorithm 1 converges to the global minimizer x^*. ■

V. Experimental results

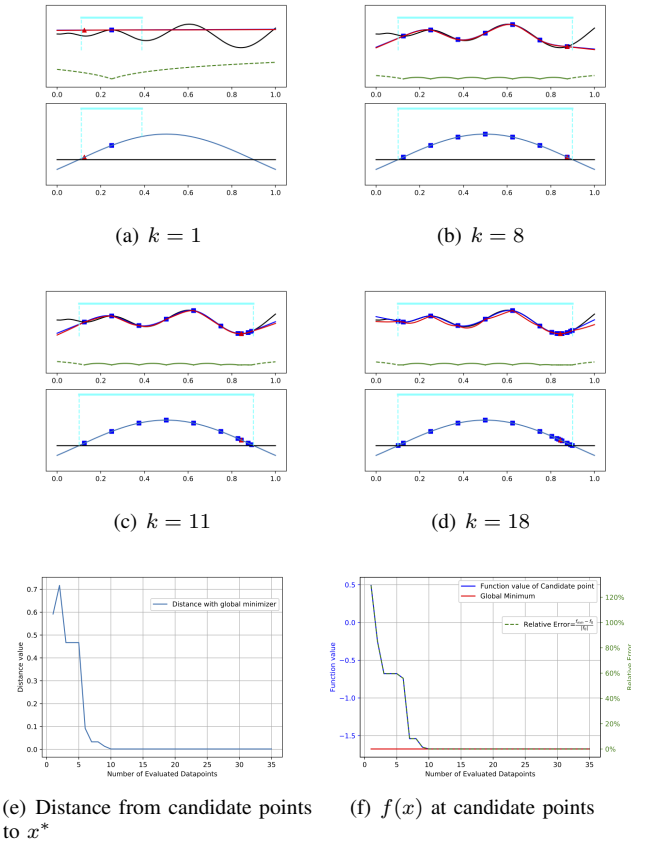
The performance of S-DOGS is shown in the following two test problems. In this work, all the results are generated by solving the nonlinear constrained optimization stated in equation (14) using SNOPT [8]. Due to the pages restriction only partial of the experimental results are shown below and more test results are under preparation.

A. Synthetic test: 1D $f(x)$-Schwefel with 1D $\psi(x)$-Sinusoid

The objective function $f(x)$ is an 1D Schwefel defined in the search domain $[0, 1]$ and is shown in equation (19). The safety function is a sinusoidal function with the underlying safe region $[0.1, 0.9]$. The objective function $f(x)$ has 2 local minimas and 1 global minimum inside the safe region. The global minima locates at $x^* = 0.8419$ with $f(x^*) = -1.6759$. The initial parameters are set as $K = 3$ and $\bar{L} = 4$. The initial mesh size is set to be 0.125 and $\ell_{\max} = 7$. Within 18 safe function evaluations, the relative error has been reduced to 0.0247%. The sampling results of S-DOGS are shown in Figure 2. Figures 2(c) and 2(d) shows that the sampling point is concentrated around x^* eventually.

$$f(x) = -2x\sin(\sqrt{500|x|}), \quad \psi(x) = \sin(\frac{5}{4}(x - 0.1)\pi). \tag{19}$$

We also implemented SAFEOPT-MC algorithm on this test problem by setting the variance of the noise terms to be 0 in order to cancel the effect of noise. The test results were generated using default choice of kernel. More tuning on the kernel function would give us fairly better results. The results of SAFEOPT-MC on this test problem is illustrated in Fig. 3 which shows the SAFEOPT-MC converges to a local minima close to the initial parameter x_0. And one unsafe parameter close to the boundary 1 is evaluated.



(a) $k = 1$ (b) $k = 8$ (c) $k = 11$ (d) $k = 18$

(e) Distance from candidate points to x^* (f) $f(x)$ at candidate points

Fig. 2. S-DOGS on 1D Schwefel test problem (19). Top plot of the first four figures: Truth function $f(x)$, uncertainty $e(x)$, interpolation $p(x)$, surrogate $s(x)$, estimated safe region $\hat{\psi}(x) \leq 0$. Bottom plot of the first four figures: Safety function $\psi(x)$, 0 horizontal line in black solid line. Evaluated data in blue squares and surrogate minima in red squares.

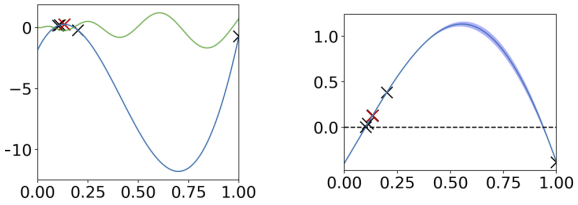


Fig. 3. SAFEOPT-MC on 1D test problem (19). Left figure: Truth function, initial parameter x_0 in red cross, evaluated data points in black cross. Right figure: Safety function.

B. Quadrotor trajectory following problem

In this section S-DOGS is applied to optimize the parameters of the quadrotor trajectory following problem. Suppose the desired trajectory is given, our goal is to determine the parameters of the quadrotor dynamic system such that the actual trajectory is as close to the reference trajectory as possible while avoiding the collision with the obstacle.

The dynamic of the quadrotor is setup by the states of positions $\mathbf{x} = [x, y, z]^T$. In the inertial frame it is described as

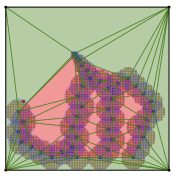
$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \mathbf{R}(t) \begin{bmatrix} 0 \\ 0 \\ c(t) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{20}$$

where $\mathbf{R}(t)$ is the rotation matrix from body frame to the inertial frame, $c(t)$ is the mass-normalized thrust and g is the acceleration of the gravity.

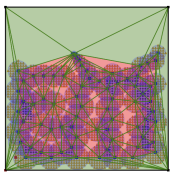
The control inputs are the roll and pitch angles θ and ϕ. In this test problem we only focus on the position of x and y directions and z direction is fixed by control laws. The PD-controller for this problem is defined as

$$\begin{cases} \phi_k = k_1(x_k - x_k^{\text{des}}) + k_2(\dot{x} - \dot{x}_k^{\text{des}}) \\ \theta_k = k_1(y_k - y_k^{\text{des}}) + k_2(\dot{y} - \dot{y}_k^{\text{des}}) \end{cases} \tag{21}$$

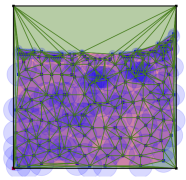
where $\mathbf{k} = (k_1, k_2)$ are two tuning parameters to control the position of the quadrotor. The thrust c is solved via the estimates of ZYX Euler angles (ϕ, θ, ψ). We assume that at $x = 3.5$ there exists an obstacle with which we would like our quadrotor to avoid collision. The box domain is assumed to be $[-1, 0]^2$. The parameters are set as $K = 3$, $\bar{L} = 3.5$, the initial safe parameter $\mathbf{k}_0 = (-0.5798, -0.2850)$, $\ell_0 = 3$ and $\ell_{\max} = 7$. S-DOGS automatically performs 3 times of mesh refinement at the beginning since the safe radius of x_0 is relatively small. The objective function is the L_2 norm distance from the desired trajectory to the experimental trajectory, $f(\mathbf{k}) = \sqrt{\sum_{i=1}^{N} \left(x_i(\mathbf{k}) - x_i^{\text{des}}\right)^2}$.



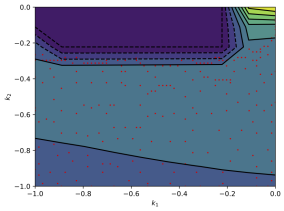
(a) Exploration: $k = 32$



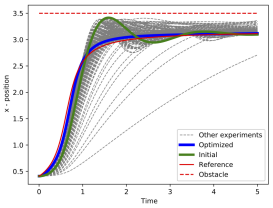
(b) Exploration: $k = 74$



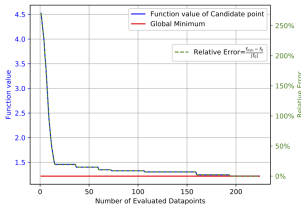
(c) Exploitation: x^* identified



(d) Sampling Contour



(e) Quadrotor trajectories



(f) $f(x)$ convergence

Fig. 4. First 3 figures: The quadrotor parameters sampling with S-DOGS: Green line-Boundary of Delaunay simplices, red background-Δ_k^{Ψ}, background-$\Delta_k^{\Psi^c}$, blue squares-S_k, triangles-G_k^{ℓ}, red squares-x_k, circles-Ψ_k^{ℓ}, red star at bottom left-x^*, blue spheres-$r(x), x \in S_k$.

The safety function is defined as the distance between the maximum position along x-direction of the trajectory and the wall, $\psi(\mathbf{k}) = 3.5 - \max(x(\mathbf{k}))$.

The result of quadrotor trajectory following optimization is shown in Fig. 4. Fig 4(a), 4(b) and 4(c) demonstrates the parameter sampling as S-DOGS iterates. Notice that the safe region are illustrated in pink in those figures. Fig. 4(d) shows the positions where we sampled parameters in the entire domain using red dot while unsafe region is depicted in purple region. From the comparison of Fig. 4(c) and Fig. 4(d), the underlying safe region has already been well explored by S-DOGS sampling schemes around 160 iterations. Fig. 4(e) shows that the trajectory of the initial safe parameter x_0 is far away from the reference trajectory, but eventually S-DOGS identified the optimized trajectory which is fairly close to the reference trajectory. Notice that the reference trajectory is generated using arctan wave which cannot be perfectly followed by the second-order system. All experiments are carried on without violating safe constraints. Fig. 4(f) shows the value of the objective function of candidate points at each iteration.

VI. Conclusions and Future Work

We extended Delaunay-based derivative-free optimization method to incorporate safety exploration tasks. The proposed safety learning algorithm S-DOGS provides theoretical approaches outside of Gaussian process which heavily depends on a well-specified kernel function. S-DOGS also enables efficient and safety-guaranteed data sampling scheme for optimal parameters tuning approach. The theoretical proof and experimental result are provided to demonstrate the convergence to the global minimum within the maximum reachable safe region.

Acknowledgments

The authors gratefully acknowledge funding from Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA) in support of this work. The author would like to thank Meng Jin and Tianwen Chong for their helpful comments of this work.

References

[1] 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. 261-271.

[2] Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8), 3861-3876.

[3] Akametalu, A. K., Fisac, J. F., Gillula, J. H., Kaynama, S., Zeilinger, M. N., & Tomlin, C. J. (2014, December). Reachability-based safe learning with Gaussian processes. *In 53rd IEEE Conference on Decision and Control* (pp. 1424-1431). IEEE.

[4] Berkenkamp, Felix, Andreas Krause, and Angela P. Schoellig. "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics." *arXiv preprint arXiv:1602.04450 (2016).*

[5] Beyhaghi, P., Cavaglieri, D., & Bewley, T. (2016). Delaunay-based derivative-free optimization via global surrogates, part I: linear constraints. *Journal of Global Optimization*, 66(3), 331-382.

[6] Beyhaghi, P., & Bewley, T. R. (2016). Delaunay-based derivative-free optimization via global surrogates, part II: convex constraints. *Journal of Global Optimization*, 66(3), 383-415.

[7] Biyik, E., Margoliash, J., Alimo, S. R., & Sadigh, D., (2019, June) Efficient and Safe Exploration in Deterministic Markov Decision Processes with Unknown Transition Models. *In 2019 American Control Conference (ACC).*

[8] Gill, P. E., & Wong, E. (2012). Sequential quadratic programming methods. *In Mixed integer nonlinear programming* (pp. 147-224). Springer, New York, NY.

[9] Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995.*

[10] Strens, M. (2000, June). A Bayesian framework for reinforcement learning. *In ICML* (Vol. 2000, pp. 943-950).

[11] Sui, Y., Gotovos, A., Burdick, J., & Krause, A. (2015, June). Safe exploration for optimization with Gaussian processes. *In International Conference on Machine Learning* (pp. 997-1005).

[12] Sui, Y., Zhuang, V., Burdick, J. W., & Yue, Y. (2018). Stagewise safe bayesian optimization with gaussian processes. *arXiv preprint arXiv:1806.07555.*

[13] Turchetta, M., Berkenkamp, F., & Krause, A. (2016). Safe exploration in finite markov decision processes with gaussian processes. *In Advances in Neural Information Processing Systems* (pp. 4312-4320).

[14] Wachi, A., Sui, Y., Yue, Y., & Ono, M. (2018, April). Safe exploration and optimization of constrained mdps using gaussian processes. *In Thirty-Second AAAI Conference on Artificial Intelligence.*

[15] Zhao, M., Alimo, S. R., & Bewley, T. R. (2018, December). An active subspace method for accelerating convergence in Delaunay-based optimization via dimension reduction. *In 2018 IEEE Conference on Decision and Control (CDC)* (pp. 2765-2770). IEEE.