Safe Zeroth-Order Optimization Using Linear Programs

Guo, Baiwei ; Wang, Y.; Kamgarpour, Maryam; Ferrari-Trecate, Giancarlo

DOI
10.1109/CDC49753.2023.10384023

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 62nd IEEE Conference on Decision and Control (CDC 2023)

2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Safe Zeroth-Order Optimization Using Linear Programs

Baiwei Guo, Yang Wang, Yuning Jiang, Maryam Kamgarpour, Giancarlo Ferrari-Trecate

***Abstract*— To solve unmodeled optimization problems with hard constraints, this paper proposes a novel zeroth-order approach called Safe Zeroth-order Optimization using Linear Programs (SZO-LP). The SZO-LP method solves a linear program in each iteration to find a descent direction, followed by a step length determination. We prove that, under mild conditions, the iterates of SZO-LP have an accumulation point that is also the primal of a KKT pair. We then apply SZO-LP to solve an Optimal Power Flow (OPF) problem on the IEEE 30-bus system. The results demonstrate that SZO-LP requires less computation time and samples compared to state-of-the-art approaches.**

I. Introduction

A variety of applications, including power network operations, machine learning, trajectory optimization and optimal control, require solving complex optimization problems with hard safety constraints. However, it is not always possible to obtain the expressions of the objective and constraint functions, or sufficient data on feasible system trajectories for modeling. In this context, safe zeroth-order optimization methods can be used to address unmodeled optimization problems with constraints. These methods rely solely on sampling by evaluating unknown objective and constraint functions at selected points [1] and the term "safe" refers to the feasibility of the samples (i.e., the satisfaction of the constraints).

Prominent safe zeroth-order methods include SafeOPT and its variations [2], [3]. These approaches assume knowledge of a Lipschitz constant of the objective and constraint functions, while [4] utilizes a Lipschitz constant of function gradients (the smoothness constants). By using these quantities, one can build local proxies for the constraint functions. Starting from a feasible point, [2]–[4] utilize these proxies to search for potential minimizers. However, for each search, one has to use a global optimization method to solve a non-convex subproblem, which makes the algorithm computationally intractable for problems with many decision variables.

To reduce the computational complexity, another research direction involves incorporating barrier functions in the objective to penalize proximity to the boundary of the feasible set [5], [6]. The Extremum-Seeking methods [7] and the LB-SGD algorithm [8] minimize a cost equipped with log-barrier penalty terms based on the estimated gradient. Although they do not require solving optimization subproblems, the performance of these two methods might not be satisfactory due to the log penalties. In Extremum Seeking, it can be challenging to tune the weight of the penalty term because a large weight can lead to suboptimality while a small weight might result in infeasibility. In LB-SGD, large values of the log barrier term and its derivative, when the iterates approach the boundary of the feasible set, can result in small step lengths and slow down convergence.

Another approach to safe zeroth-order optimization is SZO-QQ proposed in [9]. It avoids log barrier penalties while still ensuring sample feasibility and is more sample-efficient than LB-SGD [9]. This is accomplished by utilizing convex quadratic proxies for the constraint functions to construct local feasible sets, over which the proxy for the objective function is then minimized. Unlike SafeOPT, the subproblems of SZO-QQ are convex Quadratically Constrained Quadratic Programs (QCQPs), which can be solved much faster than the non-convex subproblems in SafeOPT. However, SZO-QQ falls behind LB-SGD and Extremum Seeking in terms of computational efficiency (see Section V) when dealing with large problems (with hundreds of constraints) because the size of each QCQP subproblem is almost the same as the original problem. In this paper, we propose a novel, safe zeroth-order method whose subproblems have much fewer constraints and can be computationally efficient.

Optimal Power Flow (OPF) is an example of large-scale optimization problems that can benefit from zeroth-order optimization. Its objective is to allocate the active and reactive power generation, transmission line flows and voltage levels to minimize costs while satisfying operational and security constraints such as transmission line capacity and voltage level limits. In recent years, OPF has gained considerable attention due to the rising demand for efficient and reliable operation of power systems, as well as the integration of renewable energy sources and energy storage systems [10]. However, the application of OPF to power system operation is a significant challenge due to the difficulties in accurately deriving a system model. Therefore, we consider applying our model-free method to solve OPF problems.

The contributions of this paper are summarized as follows:

- We present a novel approach called Safe Zeroth-Order optimization using Linear Programs (SZO-LP). This method iteratively solves linear programming subproblems to derive descent directions and then decides the step length by sampling;

This work was supported by the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40_180545).

Baiwei Guo and Giancarlo Ferrari-Trecate are with the DECODE group at the Institute of Mechanical Engineering, EPFL. {baiwei.guo, giancarlo.ferraritrecate}@epfl.ch.

Yang Wang is with Delft Center for Systems and Control, Delft University of Technology. {y.wang-40}@tudelft.nl.

Yuning Jiang is with the PREDICT group at the Institute of Mechanical Engineering, EPFL. {yuning.jiang}@ieee.org.

Maryam Kamgarpour is with SYCAMORE group at the Institute of Mechanical Engineering, EPFL. {maryam.kamgarpour}@epfl.ch.

2023 62nd IEEE Conference on Decision and Control (CDC) | 979-8-3503-0124-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CDC49753.2023.10384023

- We show that, under mild assumptions, a subsequence of SZO-LP's iterates converges to the primal of a KKT pair (see Definition 1);
- By application to an IEEE 30-bus benchmark problem, we show that SZO-LP can efficiently solve an OPF problem with 11 decision variables and 158 constraints. We compare SZO-LP with state-of-the-art approaches and demonstrate its advantages in terms of computation time and the number of samples required.

Notations: We use $e_i \in \mathbb{R}^d$ to define the i-th standard basis of vector space $\mathbb{R}^d$ and $\|\cdot\|$ to denote the two norms throughout the paper. Given a vector $x \in \mathbb{R}^d$ and a scalar $\epsilon > 0$, we write $x = [x^{(1)}, \dots, x^{(d)}]^\top$ and $\mathcal{B}_\epsilon(x) = \{y : \|y - x\| \le \epsilon\}$. We use $\mathbb{Z}_i^j = \{i, i+1, \dots, j\}$ to define the set of integers ranging from i to j with $i < j$. For two vectors $x, y \in \mathbb{R}^d$, we use $\langle x, y\rangle := x^\top y$ to define the inner product.

II. Problem Formulation

We consider the constrained optimization problem

$$\min_{x\in\mathbb{R}^d} \quad f_0(x) \quad \text{subject to } \; x \in \Omega, \tag{1}$$

where $\Omega := \{x : f_i(x) \le 0, i \in \mathbb{Z}_1^m\}$ is the feasible set. The functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $i \in \mathbb{Z}_0^m$, are unknown but can be sampled at query points. Throughout this paper, we make the following assumptions on the smoothness of the objective and constraint functions, availability of a strictly feasible point x_0 and boundedness of a sublevel set that includes x_0.

Assumption 1 *The functions $f_i(x)$, $i \in \mathbb{Z}_0^m$ are continuously differentiable and there are known constants $L_i, M_i > 0$ such that for any x_1, $x_2 \in \Omega$,*

$$|f_i(x_1) - f_i(x_2)| \le L_i\|x_1 - x_2\|, \tag{2a}$$

$$\|\nabla f_i(x_1) - \nabla f_i(x_2)\| \le M_i\|x_1 - x_2\|. \tag{2b}$$

We assume $L_i > \inf\{L_i :$ (2a) $\text{holds}, \forall x_1, x_2 \in \Omega\}$ and $M_i > \inf\{M_i :$ (2b) $\text{holds}, \forall x_1, x_2 \in \Omega\}$ so that (2a) *and* (2b) *always hold.*

In the remainder of this paper, we also define $L_{\max} = \max_{i\ge1} L_i$ and $M_{\max} = \max_{i\ge1} M_i$.

Assumption 2 *There exists a known strictly feasible point x_0, i.e., $f_i(x_0) < 0$ for all $i \in \mathbb{Z}_1^m$.*

Assumption 3 *There exists $\beta \in \mathbb{R}$ such that the sublevel set $\mathcal{P}_\beta = \{x \in \Omega : f_0(x) \le \beta\}$ is bounded and includes the initial feasible point x_0.*

Assumption 2 is common in safe zeroth-order methods [2], [8], [9], [11]. Without the initial feasible point, it would be impossible to ensure the feasibility of all the samples. Assumption 3 is not strong since it holds as long as the feasible region Ω is bounded.

Our aim is to derive an optimization algorithm where a subsequence of the iterates converges to the primal of a KKT pair.

Definition 1 *If a pair (x, λ) with $x \in \Omega$ and $\lambda \in \mathbb{R}^m_{\ge 0}$ satisfies*

$$\|\nabla f_0(x) + \sum_{i=1}^m \lambda^{(i)} \nabla f_i(x)\| = 0, \tag{3a}$$

$$|\lambda^{(i)} f_i(x)| = 0, \quad i \in \mathbb{Z}_1^m, \tag{3b}$$

we say that (x, λ) is a KKT pair of the problem (1) *and $x \in \Omega$ is the primal of the KKT pair.*

For a non-convex optimization problem, the primal of the KKT pair can be a global minimum, local minimum, or a saddle point of the objective function.

In the following section, we design a safe zeroth-order algorithm whose iterates, under mild assumptions, have an accumulation point that is the primal of a KKT pair of (1).

III. Algorithm: SZO-LP

In this section, we first describe how to estimate gradients of the functions in (1) and construct local feasible sets. These are the essential tools used by our zeroth-order optimization method.

A. Gradient estimation and local feasible set construction

We estimate the gradient through finite difference, i.e.,

$$\nabla^\nu f_i\left(x\right) := \sum_{j=1}^d \frac{f_i\left(x + \nu e_j\right) - f_i\left(x\right)}{\nu} e_j. \tag{4}$$

The following lemma gives a method to control the estimation error

$$\Delta_i^\nu(x) := \nabla^\nu f_i\left(x\right) - \nabla f_i(x). \tag{5}$$

Lemma 1 ([12], Theorem 3.2) *Under Assumption 1, we have*

$$\|\Delta_i^\nu(x)\|_2 \le \frac{\sqrt{d}M_i}{2}\nu. \tag{6}$$

By letting $\nu = \nu(\epsilon) := \frac{2\epsilon}{\sqrt{d}M_{\max}}$ we have $\|\Delta_i^{\nu(\epsilon)}(x)\| \le \epsilon$.

By using the estimated gradient, we can build a local feasible set around x_0. We let

$$l_0^* = \min_{i\in\{1,\dots,m\}} -f_i(x_0)/L_{\max}, \tag{7}$$

and $\nu_0^*(\epsilon) := \min\{l_0^*/\sqrt{d}, \nu(\epsilon)\}$. Since $\nu_0^*(\epsilon) \le \nu(\epsilon)$, we have $\|\Delta_i^{\nu_0^*(\epsilon)}(x_0)\| \le \epsilon$ for any $\epsilon > 0$ and $i \ge 1$.

Then the set

$$\begin{aligned} &\mathcal{S}^{(0)}(x_0) := \cap_{i=1}^m \mathcal{S}_i^{(0)}(x_0), \text{ where} \\ &\mathcal{S}_i^{(0)}(x_0) := \{x : f_i(x_0) + \nabla^{\nu_0^*(\epsilon_0)} f_i\left(x_0\right)^\top \left(x - x_0\right) + \\ &\qquad\qquad 2M_i\|x - x_0\|^2 \le 0\} \end{aligned} \tag{8}$$

is feasible as shown in the following theorem.

Theorem 1 ([9], Theorem 1) *All the samples used to construct $\mathcal{S}^0(x_0)$ are feasible. Moreover, the set $\mathcal{S}^{(0)}(x_0)$ is convex and any $x \in \mathcal{S}^{(0)}(x_0)$ is strictly feasible.*

In the lack of explicit constraint functions, a local feasible set is a common tool of several zeroth-order methods [2], [8], [9] to ensure the feasibility of the iterates, though the specific formulations are different. In the following, we propose our method where the local feasible sets are used to select the step length for the derived descent direction.

B. Algorithm: Safe Zeroth-Order Optimization Using Linear Programs (SZO-LP)

The main idea of the SZO-LP method, shown in Algorithm 1, is to iteratively select a descent direction by executing in Line 7 $\texttt{LP}(x_k, \epsilon_k)$ defined in (11). Thanks to the tightening contant ϵ_k in the linear program involved in $\texttt{LP}(x_k, \epsilon_k)$, the descent direction we obtain points into the iterior of the feasible set. Along this direction, we select the step length (Line 9-14) based on local feasible sets and the pre-defined length

$$\gamma(\epsilon_k) := \frac{\epsilon_k}{4(M_{\max} + L_{\max})}.$$

This step length is guaranteed to give a non-trivial descent (see Lemma 2) and is useful for the proof of the iterates' convergence properties (see Theorem 3).

Algorithm 1 Safe Zeroth-Order optimization using Linear Programs (SZO-LP)

```
Input: $\epsilon_0$, $\epsilon_{\min}$, $K_{\text{switch}}$, initial feasible point $x_0 \in \Omega$
Output: $\tilde{x}$
1: $k \leftarrow 0$
2: while $\epsilon_k > \epsilon_{\min}$ do
3:     $s_{\text{tmp}} \leftarrow \texttt{LP}(x_k, 2\epsilon_k)$
4:     if $\nabla^{\nu_k^*(2\epsilon_k)} f_0(x_k)^\top s_{\text{tmp}} \leq -4\epsilon_k$ then
5:         $\epsilon_{k+1} \leftarrow 2\epsilon_k$, $x_{k+1} \leftarrow x_k$
6:     else
7:         $s_k^* = \texttt{LP}(x_k, \epsilon_k)$
8:         if $\nabla^{\nu_k^*(\epsilon_k)} f_0(x_k)^\top s_k^* \leq -2\epsilon_k$ then
9:             if $k < K_{\text{switch}}$ then
                   $\beta_k = \arg\max_{\beta \geq 0} \beta$ s.t. $x_k + \beta s_k^* \in \mathcal{S}^{(k)}(x_k)$,   (9)
                   $\alpha_k = \arg\min_{\alpha \in \{\beta_k, \gamma(\epsilon_k)\}} f_0(x_k + \alpha s_k^*)$   (10)
10:                $x_{k+1} \leftarrow x_k + \alpha_k s_k^*$, $\epsilon_{k+1} \leftarrow \epsilon_k$
11:            else
12:                $x_{k+1} \leftarrow x_k + \gamma(\epsilon_k) s_k^*$, $\epsilon_{k+1} \leftarrow \epsilon_k$
13:            end if
14:        else
15:            $\epsilon_{k+1} \leftarrow \epsilon_k/2$, $x_{k+1} \leftarrow x_k$
16:        end if
17:    end if
18:    $k \leftarrow k + 1$
19: end while
```

The essential steps are as follows:

1) Providing the input data: The input includes an initial strictly feasible point x_0 (see Assumption 2) and a tightening constant ϵ_0. Each iteration of the algorithm generates a new tightening constant ϵ_k, which can be equal to ϵ_{k-1} (Line 10 or 12), $2\epsilon_{k-1}$ (Line 5) or $\epsilon_{k-1}/2$ (Line 15), and thus the selection of the initial value for the tightening constant is not critical. Since ϵ_k converges to 0 (see Theorem 2), the user can control the termination by providing a lower bound $\epsilon_{\min}$ for ϵ_k. The parameter K_{switch} marks the boundary of two methods for selecting step length, see the last bullet point.

2) Building local feasible sets: For a strictly feasible x_k, we use (7) to define l_k^* and

$$\nu_k^*(\epsilon_k) := \min\{l_k^*/\sqrt{d}, \nu(\epsilon_k)\}.$$

We then use $\nu_k^*(\epsilon_k)$ and (8) to define $\mathcal{S}^{(k)}(x_k)$, a local feasible set around x_k. From Theorem 1 we know that if $x_{k+1} \in \mathcal{S}^{(k)}(x_k)$ then x_{k+1} is also strictly feasible.

3) Solving subproblems for the descent diretion: In each iteration, we execute in Line 7 $\texttt{LP}(x_k, \epsilon_k)$ to derive a search direction, which returns

$$\begin{aligned} \underset{\|s\|_1 \leq 1}{\arg\min} \quad & (\nabla^{\nu_k^*(\epsilon_k)} f_0(x_k))^\top s \\ \text{s.t.} \quad & (\nabla^{\nu_k^*(\epsilon_k)} f_i(x_k))^\top s + 2\epsilon_k \leq 0, \\ & \forall i \in \mathcal{A}(x_k, \epsilon_k), \end{aligned} \tag{11}$$

or NaN if (11) is not feasible. Here, $\mathcal{A}(x, \epsilon) := \{i : f_i(x) \geq -2\epsilon\}$ is the near-active constraint index set. The solution to (11) is a direction that not only gives a fast descent but also points into the interior of the feasible region Ω (away from the boundary). In (11), due to the tightening constant ϵ_k, along the direction s_k^* in Line 7, the constraint function values decrease. Therefore, moving along the direction s_k^* we indeed stay away from the boundary of Ω. This direction helps to avoid small values of $-f_i(x_k)$, which lead to conservative local feasible sets $\mathcal{S}^{(k)}(x_k)$. Moreover, the inclusion of only near-active constraints makes (11) small-size and easy to solve. We will later see in Theorem 2 that ϵ_k converges to 0. Therefore, it is still possible that a subsequence of the iterates converges to a point on the feasible set boundary.

We also let $s_{\text{tmp}} = \texttt{LP}(x_k, 2\epsilon_k)$ and check in Line 4 whether $\nabla^{\nu_k^*(2\epsilon_k)} f_0(x_k)^\top s_{\text{tmp}} \leq -4\epsilon_k$, which allows us to have Proposition 1, the proof of which is in Appendix A of the techical report [13]. This proposition will be later used to show in Theorem 3 the properties of the $\{x_k\}_{k \geq 1}$ as k goes to infinity.

Proposition 1 *Any ϵ_k entering Line 7 satisfies*

$$\begin{aligned} \epsilon_k \geq \frac{1}{8}\sup\{\epsilon : s = \texttt{LP}(x_k, \epsilon) \textit{ verifies} \\ \nabla^{\nu_k^*(\epsilon)} f_0(x_k)^\top s \leq -2\epsilon\}. \end{aligned} \tag{12}$$

4) Deciding the step length: When a direction s_k^* derived in Line 7 gives sufficient descent (i.e., $\nabla^{\nu_k^*(\epsilon_k)} f_0(x_k)^\top s_k^* \leq -2\epsilon_k$), we move along the tentative direction s_k^*. To decide the step length, we consider the local feasible set and the pre-defined step length $\gamma(\epsilon_k)$ that is guaranteed to achieve a non-trivial descent (see Lemma 2). In (9), we calculate by bisection the largest step length within the local feasible set to derive α_k in (10). The use of local feasible sets in Line 10 allows us to obtain a larger step length than $\gamma(\epsilon)$, when x_k is not close to the boundary of the feasible set.

This is because, from the formulation (8), smaller values of $f_i(x_k)$ lead to larger sizes of $\mathcal{S}_i^{(k)}(x_k)$ while $\gamma(\epsilon_k)$ is independent of how far the iterates are from the feasible set boundary. When $k > K_{\mathrm{switch}}$, we let the step length be $\gamma(\epsilon_k)$ as in Line 12, which is useful for the proof of the iterates' properties as k goes to infinity (see Theorem 3). The selection of K_{switch} is not critical since we use the step length in Line 10 for $k < K_{\mathrm{switch}}$ instead of that defined in Line 12 only to accelerate the descent in the early iterations of the algorithm.

On the other hand, if the direction s_k^* cannot give sufficient descent, we let $\epsilon_{k+1} = \epsilon_k/2$ in Line 15 to relax the tightened constraints in (11). This relaxation makes it easier for s_{k+1}^* to give sufficient descent, i.e., to satisfy $\nabla^{\nu_k^*(\epsilon_{k+1})} f_0(x_{k+1})^\top s_{k+1}^* \leq -2\epsilon_{k+1}$. Only when s_{k+1}^* gives sufficient descent will we move along s_{k+1}^* to a new point.

We refer the readers to Remark 1 for how SZO-LP is compared with some state-of-the-art methods.

IV. Convergence properties of the approach

In this section, we aim to show that, under mild conditions and by letting $\epsilon_{\min} = 0$, the sequence $\{x_k\}_{k\geq 1}$ produced in Algorithm 1 has an accumulation point x_c that is also the primal of a KKT pair of (1). To start with, we show in Lemma 2 that, whenever $x_{k+1} \neq x_k$, the new iterate x_{k+1} is strictly feasible and the objective function value gets a non-trivial decrease.

Lemma 2 *Suppose s_k^* derived in Line 7 of Algorithm 1, satisfies*

$$\nabla^{\nu_k^*(\epsilon_k)} f_0(x_k)^\top s_k^* \leq -2\epsilon_k.$$

We have that $x_k + \gamma(\epsilon_k)s_k^*$ is strictly feasible. Furthermore $x_k + \gamma(\epsilon_k)s_k^*$ satisfies

$$f_0(x_k + \gamma(\epsilon_k)s_k^*) - f_0(x_k) < -\epsilon_k^2/(8(M_{\max} + L_{\max})). \quad (13)$$

The proof of Lemma 2 is in Appendix B of the technical report [13]. The main idea is to utilize the smoothness constants in Assumption 1 to upper-bound $f_i(x_k + \gamma(\epsilon_k))$ for $i \in \mathbb{Z}_0^m$. Based on this lemma, we have the following theorem on the sequences $\{x_k\}_{k\geq 1}$ and $\{\epsilon_k\}_{k\geq 1}$ as k goes to infinity.

Theorem 2 *The following arguments hold:*

1. *The sequence $\{f_0(x_k)\}_{k\geq 1}$ is non-increasing;*
2. *There exists at least one accumulation point of the sequence $\{x_k\}_{k\geq 1}$. For any accumulation point x_c,*
$$\lim_{k\to\infty} f_0(x_k) = f_0(x_c) > -\infty.$$
3. *The sequence $\{\epsilon_k\}_{k\geq 1}$ converges to 0.*

The first point is a direct consequence of Lemma 2. The second is due to the first point, Assumption 3 and Bolzano–Weierstrass theorem. The third can be shown through contradiction. If $\{\epsilon_k\}_{k\geq 1}$ did not converge to 0, the decrease of the cost function would not diminish either (see Lemma 2) and therefore $\{f_0(x_k)\}_{k\geq 1}$ goes to $-\infty$, which contradicts with Assumption 3. The detailed proof of Theorem 2 can be found in Section IV of the technical report [13].

Theorem 2 offers us the essential tools to show in Theorem 3 the properties of an accumulation point of $\{x_k\}_{k\geq 1}$ under Assumption 4.

Assumption 4 *At least one accumulation point x_c of $\{x_k\}_{k\geq 1}$ satisfies Linear Independent Constraint Qualification (LICQ), which is to say the gradients $\nabla f_i(x_c)$ with $i \in \mathcal{A}(x_c, 0)$ are linearly independent.*

Assumption 4 is widely used in optimization [14]. For example, it is used to prove the properties of the limit point of the Interior Point Method [15].

Theorem 3 *Regarding the accumulation point x_c in Assumption 4, there exists $\lambda_c \in \mathbb{R}_{\geq 0}^m$ such that (x_c, λ_c) is a KKT pair of* (1).

The proof, in Appendix C of the technical report [13], is based on contradiction. If x_c is not the primal of a KKT pair, we can find $r > 0$, $\epsilon > 0$ and $s_\epsilon \in \mathbb{R}^d$ such that for any $x_k \in \mathcal{B}_r(x_c)$ the solution $s = \mathtt{LP}(x_k, \epsilon)$ verifies $\nabla^{\nu_k^*(\epsilon)} f_0(x_k)^\top s \leq -2\epsilon$. There are infinitely many k such that $x_k \in \mathcal{B}_r(x_c)$ and s_k^* is derived through Line 7 in Algorithm 1. For any of these ks, according to (12), $\epsilon_k \geq \epsilon/8$, which contradicts Point 3 of Theorem 2.

Remark 1 *Like SZO-QQ [9] and LB-SGD [8], the samples in SZO-LP are all feasible and the iterates, under mild assumptions, have an accumulation point that is also the primal of a KKT pair. In contrast, the tightening constant ϵ_k of SZO-LP keeps the iterates away from the boundary of the feasible set and leads to less conservative local feasible sets than those used in SZO-QQ and LB-SGD. Moreover, due to the use of the near-active set $\mathcal{A}(x_k, \epsilon_k)$ the subproblems* (11) *are smaller-size and easier to solve than the QCQPs in SZO-QQ and nonconvex subproblems in Safe Bayesian Optimization methods [2], [16]. However, to rigorously show these advantages, we need to upper bound the number of iterations needed by SZO-LP given certain accuracy requirements, which is left as future work.*

V. Experiment on an OPF problem

To illustrate the performance of SZO-LP, we consider applying it to an OPF problem on the IEEE 30-bus system.

A. Formulation of the OPF problem

To formulate an OPF problem, we introduce the following notations and assumptions:

- Let $B = \{b_1, b_2, \ldots, b_n\}$ be the bus set and let $T = \{(b_i, b_j)$: there is a transmission line between b_i and $b_j\}$ be a set of undirected edges representing the transmission lines;

- We denote P_{G_i}, P_{L_i}, Q_{L_i}, U_i and θ_i as the active power generation, active power consumption, reactive power consumption, voltage and voltage angle at b_i;
- From the b_i to b_j, the active power and the reactive power transferred are written respectively as $P_{ij}(U_i, U_j, \theta_i, \theta_j)$ and $Q_{ij}(U_i, U_j, \theta_i, \theta_j)$, while the current is denoted as $I_{ij}(U_i, U_j, \theta_i, \theta_j)$. We refer the readers to [17] for the explicit expressions of these functions;
- We also assume that there are n_G generators at the buses b_i, $i \in \mathbb{Z}_1^{n_G}$ and b_1 is a slack bus providing active power to maintain the power balance within the network and has a voltage angle of 0.

Then the OPF problem is formulated [17] as

$$\min_{P_{G_i}, U_i, \theta_i} \sum_{i=1}^{n_G} C_i(P_{G_i}) \tag{14a}$$

subject to

$$P_{G_i} = P_{L_i} + \sum_{(i,j)\in T} P_{ij}(U_i, U_j, \theta_i, \theta_j), \ \ \forall i \tag{14b}$$

$$-Q_{L_i} = \sum_{(i,j)\in T} Q_{ij}(U_i, U_j, \theta_i, \theta_j), \ \ i > n_G \tag{14c}$$

$$P_{G_i} = 0, \ \ \text{for } i > n_G, \ \ \theta_1 = 0, \tag{14d}$$

$$P_{G,\min} \le P_{G_i} \le P_{G,\max}, \text{for } \ i \le n_G, \tag{14e}$$

$$I_{ij,\min} \le I_{ij}(U_i, U_j, \theta_i, \theta_j) \le I_{ij,\max}, \ \ \forall (i,j) \in T, \tag{14f}$$

$$U_{\min} \le U_i \le U_{\max}, \forall i. \tag{14g}$$

where $C_i(\cdot)$ is a quadratic function accounting for the generation cost and the equations (14e)-(14g) give the safe intervals for the corresponding variables.

The main challenges of OPF applications lie in modelling the system and deriving the accurate expressions of (14). The difficulties include the nonlinearity of device dynamics, slowly changing physical parameters and disturbances [18]. Inaccurate models can result in suboptimal OPF solutions (leading to more generation cost) or violate the true hard constraints (causing damages to devices) [19]. Therefore, we consider the black-box setting and use SZO-LP.

To this aim, we reformulate (14) as optimization with only inequality constraint to fit (1) used by SZO-LP. Let $\{P_{G_i}\}_{i=2}^{n_G}$ and $\{U_i\}_{i=1}^{n_G}$ be the main decision variables. Then by assigning values to $\{P_{G_i}\}_{i=2}^{n_G}$ and $\{U_i\}_{i=1}^{n_G}$, one can solve the power flow equations (14b)-(14d) to derive the values for all the other decision variables in (14). Therefore, (14b)-(14d) give us the functions

$$\begin{aligned} U_i &= U_i(\{P_{G_j}\}_{j=2}^{n_G}, \{U_j\}_{j=1}^{n_G}), \ \ i = n_G+1, \ldots, n, \\ \theta_i &= \theta_i(\{P_{G_j}\}_{j=2}^{n_G}, \{U_j\}_{j=1}^{n_G}), \ \ i = 1, \ldots, n. \end{aligned} \tag{15}$$

By substituting (15) to (14), we obtain a reformulation where $\{\{P_{G_j}\}_{j=2}^{n_G}, \{U_j\}_{j=1}^{n_G}\}$ are the only decision variables and there are not equality constraints.

B. Experiment results

We run SZO-LP to solve a specific OPF problem on the IEEE 30-bus system where $n_G = 6$. In total, there are 11 decision variables and 158 constraints. We do not assume knowledge of the system model for the optimization task. However, given a set of values for all 11 decision variables, we can use a black-box simulation model in `Matpower` [20] to sample the voltages of all the 30 buses and the current through all the transmission lines in the network. Additionally, we assume the availability of initial values for all the decision variables to start the SZO-LP algorithm from a feasible point.

We employ SZO-LP to reduce the quadratic cost induced by the initial decision values. The numerical experiments are executed on a PC with an Intel Core i9 processor. The solver we adopt for subproblems (11) is LINPROG in Matlab. We let $M_i = M_{\max} = 0.13$ and $L_i = L_{\max} = 0.5$. The tuning of these two parameters is described in [9]. Moreover, we set $\epsilon_0 = 0.05$, $\epsilon_{\min} = 10^{-6}$ and $K_{\text{switch}} = 200$.

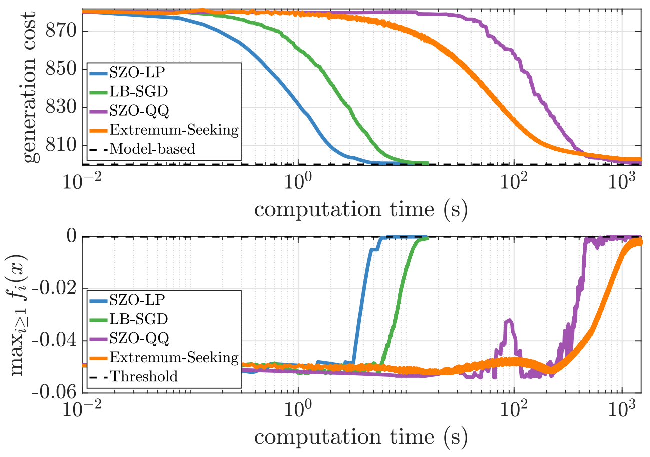

Fig. 1: Decrease of cost and growth of the largest constraint function values with respect to computation time

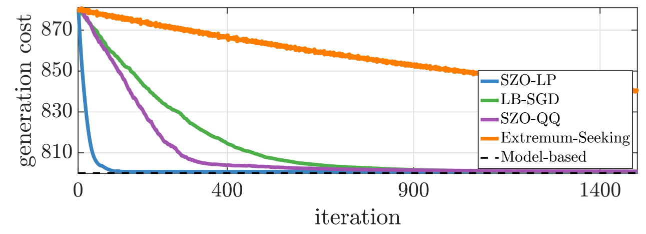

Fig. 2: Decrease of cost with respect to the number of iterations

In Figures 1 and 2 , we present the results of our numerical experiments, where we compare the performance of SZO-LP with SZO-QQ [9], LB-SGD [8] and Extremum Seeking [21]. The QCQP subproblems in SZO-QQ are solved using MOSEK. The reference solution of the OPF problem is returned by the optimization based on the true model and utilizing Gurobi [22] as the solver. The computation time in Figure 1 includes that consumed by power grid simulation (through Matpower) when we query the objective and constraint functions. We observe that all four methods keep the iterates feasible and eventually achieve a generation cost very close to that (800.14) derived based on the true model. However, SZO-LP achieves a faster decrease in the generation cost than the other methods.

One main reason for the superior performance of SZO-LP over SZO-QQ with respect to computation time shown in Figure 1, is that the linear programming subproblems can be solved faster. We notice that to finish the first 60 subproblems, SZO-LP takes 5.63 seconds while SZO-QQ takes 72.06 seconds. Firstly, the subproblem in SZO-LP only takes into account the near-active constraints while the subproblem in SZO-QQ involves all constraints. Among the iterations of SZO-LP, the largest number of constraints is 2. Secondly, although the big gap in efficiency shown in Figure 1 may be due to the specific solvers we select, linear programs, in general, are open to a wider selection of solvers and thus allow for more efficient implementations.

Unlike SZO-LP and SZO-QQ, LB-SGD and Extremum Seeking do not require solving any subproblems, thus allowing for more iterations within a certain time length. This is why LB-SGD can also achieve a low generation cost in a short time. However, considering the four methods take the same number of samples every iteration, LB-SGD and Extremum Seeking are less sample-efficient than SZO-LP and SZO-QQ since they require more iterations as shown in Figure 2. Moreover, since LB-SGD and Extremum Seeking are based on log barriers, these two methods require tuning of the barrier function coefficients. Improper tuning might lead to suboptimality in LB-SGD or even infeasibility in Extremum Seeking.

SZO-LP has another advantage over SZO-QQ, which is the feature of SZO-LP keeping the iterates away from the feasible set boundary before getting close to the primal of a KKT pair. Iterates getting too close to the feasible set boundary might impede the decrease of the cost. To see this point, we notice from Figure 1 that in SZO-QQ the decrease of the generation cost slows down when the largest constraint function value is larger than -0.005. The reason is that, when the largest constraint function value is close to 0, the local feasible set constructed in SZO-QQ gets conservative, and thus the step length becomes small. When the largest constraint function value gets larger than -0.005 for the first time, the generation cost in SZO-QQ is 805.27 while the corresponding cost in SZO-LP is 801.77, which is much closer to 800.14 (derived by optimization based on the true model). Therefore, we see that in SZO-QQ the decrease of the objective function value can slow down at a much earlier stage.

In conclusion, from the experiment results, we see that SZO-LP is the most computation-efficient and sample-efficient method, among the four approaches.

VI. Conclusion

In this paper, we proposed a safe zeroth-order method SZO-LP, which iteratively solves linear programs to obtain descent directions and determines the step lengths. We showed that, under mild conditions, the iterates of SZO-LP have an accumulation point that is also the primal of a KKT pair. Through an experiment on an OPF problem, we see that SZO-LP is both computation-efficient and sample-efficient. Our future directions include the derivation of the computation complexity of SZO-LP to check whether it is efficient in general and the extension of SZO-LP to account for measurement noises.

References

[1] I. Bajaj, A. Arora, and M. Hasan, *Black-Box Optimization: Methods and Applications*, pp. 35–65. Springer, 2021.

[2] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *International Conference on Machine Learning*, pp. 997–1005, PMLR, 2015.

[3] L. Sabug Jr, F. Ruiz, and L. Fagiano, "Smgo-δ: Balancing caution and reward in global optimization with black-box constraints," *Information Sciences*, vol. 605, pp. 15–42, 2022.

[4] A. P. Vinod, A. Israel, and U. Topcu, "Constrained, global optimization of unknown functions with lipschitz continuous gradients," *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 1239–1264, 2022.

[5] R. M. Lewis and V. Torczon, "A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 1075–1089, 2002.

[6] C. Audet and J. E. Dennis Jr, "A progressive barrier for derivative-free nonlinear programming," *SIAM Journal on optimization*, vol. 20, no. 1, pp. 445–472, 2009.

[7] L. Hazeleger, D. Nešić, and N. van de Wouw, "Sampled-data extremum-seeking framework for constrained optimization of nonlinear dynamical systems," *Automatica*, vol. 142, p. 110415, 2022.

[8] I. Usmanova, Y. As, M. Kamgarpour, and A. Krause, "Log barriers for safe black-box optimization with application to safe reinforcement learning," *arXiv preprint arXiv:2207.10415*, 2022.

[9] B. Guo, Y. Jiang, G. Ferrari-Trecate, and M. Kamgarpour, "Safe zeroth-order optimization using quadratic local approximations," *arXiv preprint arXiv:2303.16659*, 2023.

[10] H. Abdi, S. D. Beigvand, and M. La Scala, "A review of optimal power flow studies applied to smart grids and microgrids," *Renewable and Sustainable Energy Reviews*, vol. 71, pp. 742–766, 2017.

[11] I. Usmanova, A. Krause, and M. Kamgarpour, "Safe convex learning under uncertain constraints," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2106–2114, PMLR, 2019.

[12] A. S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, "A theoretical and empirical comparison of gradient approximations in derivative-free optimization," *Foundations of Computational Mathematics*, vol. 22, no. 2, pp. 507–560, 2022.

[13] B. Guo, Y. Wang, Y. Jiang, M. Kamgarpour, and G. Ferrari-Trecate, "Safe zeroth-order optimization using linear programs," *arXiv preprint arXiv:2304.01797*, 2023.

[14] G. Wachsmuth, "On LICQ and the uniqueness of lagrange multipliers," *Operations Research Letters*, vol. 41, no. 1, pp. 78–80, 2013.

[15] J. Nocedal and S. J. Wright, *Numerical optimization*. Spinger, 2006.

[16] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *Machine Learning*, pp. 1–35, 2021.

[17] J. Das, *Load flow optimization and optimal power flow*. Crc Press, 2017.

[18] Z. Chu, S. Lakshminarayana, B. Chaudhuri, and F. Teng, "Mitigating load-altering attacks against power grids using cyber-resilient economic dispatch," *IEEE Transactions on Smart Grid*, 2022. Early access.

[19] D. Lee, K. Turitsyn, D. K. Molzahn, and L. A. Roald, "Robust AC optimal power flow with robust convex restriction," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 4953–4966, 2021.

[20] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.

[21] D. B. Arnold, M. Negrete-Pincetic, M. D. Sankur, D. M. Auslander, and D. S. Callaway, "Model-free optimal control of var resources in distribution systems: An extremum seeking approach," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3583–3593, 2015.

[22] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022.