Peregrination through blackbox optimization : multimodality, stochasticity and risk aversion.

Romain Couderc

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES ET DE POLYTECHNIQUE MONTRÉAL

École doctorale : IMEP2
Spécialité : Génie industriel & Mathématiques
Unité de recherche : UMR INP/CNRS/UGA 527

Pérégrination à travers l'optimisation de boite noire: multimodalité, stochasticité et aversion au risque

Peregrination through blackbox optimization: multimodality, stochasticity and risk aversion

Présentée par :

Romain COUDERC

Direction de thèse :

Charles AUDET Ph.D., GERAD	Directeur de thèse
Jean BIGEON Ph.D., LS2N	Co-Directeur de thèse
Michael KOKKOLARAS Ph.D., McGill University	Co-Directeur de thèse

Thèse soutenue publiquement le **11/12/23**, devant le jury composé de :

Sébastien LE DIGABEL Ph.D., GERAD	Président du jury
Wei CHEN Ph.D., Northwestern University	Rapporteure
Sébastien DA VEIGA Ph.D., ENSAI	Rapporteur
Clémentine PRIEUR Ph.D., Université Grenoble Alpes	Examinatrice
Frédéric MESSINE Ph.D., Université de Toulouse	Examinateur

DEDICATION

À mes parents

"A mathematician who is not also something of a poet will never be a complete mathematician."
— Karl Weierstrass

ACKNOWLEDGEMENTS

Très cher lecteur (ou lectrice) des remerciements,

De manière presque sûre, il est probable que tu fasses partie d'un des deux ensembles disjoints suivants: celui des remerciés ou celui des oubliés. Dans une subtile disjonction des cas, je tiens par avance à m'excuser auprès des éléments du second ensemble et assure qu'il ne faut pas y voir un quelconque mépris de ma part mais plutôt une question de finitude de l'espace d'un manuscrit de thèse. En effet, celle-ci est en contradiction directe avec le caractère infini mais dénombrable du second ensemble (preuve laissée au lecteur). Afin d'éviter tout mécontentement, je propose d'utiliser une fonction de choix associant à chacun des éléments du second ensemble des remerciements dès à présent.

Concentrons nous désormais sur l'ensemble premier, ayant l'avantage ici d'être fermé et borné. Je commencerai par mes trois géants à moi, ceux m'ayant permis de voir bien plus loin qu'une ligne de niveau. Merci à toi, Michael, pour les nombreux dialogues au sujet des différents types d'incertitudes et pour tes relectures d'une qualité inversement proportionnelle à celle de mon anglais. Merci à toi, Jean, pour les nombreuses discussions, pour tes conseils sur la thèse et bien au-delà, et pour ton dévouement dans les tâches administratives les plus sombres. Merci à toi, Charles, pour ton enthousiasme dans mes recherches, pour cette capacité à me rassurer en toute circonstance et pour ton expertise. Nous formions un très beau quartet et c'est avec un réel pincement au coeur que je vois cette aventure, scientifique mais également humaine, se terminer.

J'aimerais souligner aussi l'importance de tous les chercheurs et professeurs ayant contribué dans un front de Pareto commun à mon suivi et ma réussite. Que ce soit mon professeur de mathématique en classe préparatoire, M. Gonnord, pour m'avoir donné goût aux mathématiques ou les membres de mon CSI et de mon jury pour avoir lu, commenté et questionné mon travail.

Une condition nécessaire concernant l'existence de cette thèse tient évidemment dans mes deux tutelles Polytechnique Montréal et l'université Grenoble-Alpes, respectivement incarnées par le GERAD et GSCOP. Je tiens à féliciter l'ensemble de leurs personnels administratifs, et en particulier Melisa, pour avoir résolu ce problème NP-complet qu'est la gestion d'une thèse en cotutelle. Je remercie également les différents organismes qui ont bien voulu me financer: NSERC, IVADO, HUAWEI, le GERAD et le gouvernement français.

Finalement, tout ceci n'aurait pas été possible sans mes collègues et désormais amis : Alexis, le marcheur aléatoire de la Gaspésie, Joseph, ce précédent itéré qui m'a tant inspiré et bien entendu Pierre-Yves, mon binôme dans la dernière ligne droite.

Ni sans mes amis anciennement collègues. Je parle notamment de mes requins de l'Ensimag avec lesquels j'ai pu découvrir les saveurs des lemmes et la valeur des théorèmes (pour être mathématiquement correct). Il y a également mes hx du Parc, avec qui l'intégration mène toujours vers des dérives. Parmi eux, je voudrais mettre en avant ceux qui ont osé le saut de dualité jusqu'à la belle province: Alice et Mathilde pour leurs descentes (de gradient), Val pour son étanchéité, ne prenant jamais d'eau tel une palplanche (à part sur un canot) et Anouk pour sa capacité à gérer les contraintes les plus pointues (surtout de nuit et sans matelas). Sans oublier mes collègues de toujours: ma constante Philou et mon invariant Alex.

Ou encore sans ma team du volley, le L, le Q et le double A avec qui les spikes sont parfois paraboliques mais les rigolades, elles, restent toujours aussi carrés. Et encore moins sans mes collègues, amis et colocataires qui m'ont supporté sans discontinuité pendant de longs mois. Je parle évidemment de Geoffroy, qui, sans perte de généralité, est le concepteur du célèbre algorithme RIP-QP et de Paul, mon partiellement inséparable coach de volley et de vie.

Mes derniers mots, les réels cette fois, vont à ma quadrature du cercle familial, et spécifiquement à mes parents. Papa, Maman, si nos relations furent parfois complexes, dans mon imaginaire vous avez toujours été à la racine de ma passion et, rien que pour cela, je vous dois un infini merci.

RÉSUMÉ

Pour aborder l'optimisation de boite noire, ce projet doctoral comporte trois contributions, dont la conception est agencée autour d'une unique notion: l'exploration Gaussienne de l'espace. Cette exploration consiste à échantillonner des points à partir d'une moyenne et d'un écart-type donnés. Dans une approche directe, l'algorithme se déplace directement vers un point minimisant une certaine quantité d'intérêt dépendant de la fonction objectif et/ou des contraintes. Dans une approche indirecte, les points échantillonnés sont utilisés pour estimer le gradient d'une approximation lisse de la boite noire. Ces deux approches ont pour avantage de ne pas dépendre de la dimension de la boite noire et de ne se fonder que sur les valeurs des fonctions retournées par celle-ci. Elles s'adaptent donc parfaitement au contexte de l'optimisation de boite noire. L'objectif de cette thèse est donc de développer des algorithmes autour de ces approches et d'étudier leurs propriétés de convergence ainsi que leurs efficacités en pratique.

Le premier projet de la thèse traite de la problématique de la multimodalité dans un cadre de boite noire déterministe. La méthode de l'entropie croisée (CE) est intégrée dans l'algorithme de recherche directe par treillis adaptatif en tant qu'étape de recherche. Cette étape a pour but d'explorer l'espace des variables de conceptions et d'éviter de converger prématurément vers un minimum local. L'algorithme résultant bénéficie des propriétés de convergence de l'algorithme MADS. Des comparaisons numériques ont été menées avec d'autres algorithmes sur un ensemble de problèmes multimodaux et sur des problèmes d'ingénierie. Les résultats permettent de démontrer la compétitivité de l'algorithme sur ces types de problèmes.

Le second projet de thèse aborde les problèmes d'optimisation stochastique de boite noire sans contrainte. Dans ce projet, un algorithme séquentiel (SSO) est développé afin de résoudre une suite d'approximations lisses de plus en plus fines du problème original. Chaque sous problème est résolu grâce à un algorithme de descente de gradient stochastique, appelé ZO-Signum, où les gradients sont estimés à partir d'évaluation de la boite noire seulement et dont la direction de descente est déterminée par le signe d'un vecteur moment. Les propriétés de convergence des deux algorithmes ont été étudiées. Si la boite noire est supposée lisse et est localement convexe autour de ses minima locaux, alors nous avons démontré le taux de convergence d'une sous suite d'itérés de l'algorithme SSO vers un point stationnaire du problème. Finalement, des tests numériques ont été réalisés sur une simulation de centrale solaire et pour la génération d'images adverses. Ils montrent l'efficacité de l'algorithme comparé à d'autres algorithmes de la littérature.

Le troisième projet de thèse traite des problèmes d'optimisation de boite noire sous contraintes et soumis à des incertitudes aléatoires et épistémiques. La valeur conditionelle au risque (CVaR) est

utilisée pour gérer les incertitudes dans la fonction objectif et les contraintes. Cette formulation a l'avantage de pouvoir choisir le degré de fiabilité et de traiter les incertitudes épistémiques avec une approche du pire cas lorsque ce degré est pris suffisamment proche de 1. Pour résoudre la relaxation Lagrangienne du problème CVaR-contraint, un algorithme d'approximation stochastique à multi-échelle de temps (RAMSA) est développé. Nous avons prouvé que l'algorithme RAMSA converge presque-sûrement vers un point réalisable du problème CVaR-contraint dont la valeur de la fonction objectif est arbitrairement proche de celle d'une solution locale. Enfin, des tests numériques ont été réalisés avec les buts suivants: établir des stratégies permettant de déterminer la valeur des hyperparamètres de l'algorithme, comparer différents estimateurs du gradient et montrer l'efficacité de l'algorithme sur des problèmes soumis à des incertitudes aléatoires et epistémiques.

ABSTRACT

To tackle blackbox optimization, this thesis consists of three contributions, designed around a single notion: Gaussian exploration of space. This exploration consists of sampling points from a given mean and standard deviation. In a direct approach, the algorithm moves directly to a point minimizing a certain quantity of interest, depending on the objective function and/or constraints. In an indirect approach, the sampled points are used to estimate the gradient of a smooth blackbox approximation. The advantage of both approaches is that they do not depend on the size of the blackbox, and are based only on the output values returned by the blackbox. Therefore, they are perfectly suited to the context of blackbox optimization. Thus, the aim of this thesis is to develop algorithms based on these approaches and to study their convergence properties as well as their effectiveness in practice.

The first contribution deals with the problem of multimodality in a deterministic blackbox framework. The CE method is integrated into the MADS algorithm as a search step. The aim of this step is to explore the space of design variables and avoid converging prematurely to a local minimum. The resulting algorithm benefits from the convergent properties of the MADS algorithm. Numerical comparisons with other algorithms have been performed on a set of multimodal and engineering problems. The results demonstrate the competitiveness of the algorithm on these types of problems.

The second contribution deals with unconstrained stochastic blackbox optimization problems. A SSO algorithm is developed to solve a sequence of increasingly finer smooth approximations of the original problem. Each subproblem is solved using a stochastic gradient descent algorithm called ZO-Signum, where gradients are estimated only from blackbox evaluations and the descent direction is determined by the sign of a momentum vector. The convergence properties of both algorithms have been studied. If the blackbox is assumed smooth and is locally convex around its local minima, then a subsequence of iterates of the SSO algorithm is proved to converge to a stationary point. The convergence rate is also analyzed. Finally, numerical tests have been conducted on a simulation of a solar power plant and for the generation of adversarial images. They show the efficiency of the algorithm compared to other state-of-the-art algorithms.

The third contribution deals with blackbox constrained optimization problems subject to aleatory and epistemic uncertainties. Conditional value at risk is used to manage uncertainties in the objective function and constraints. This formulation has the advantage of being able to choose the degree of reliability and to handle epistemic uncertainties with a worst case approach when this is taken sufficiently close to 1. To solve the Lagrangian relaxation of the CVaR-constrained problem,

a RAMSA algorithm is introduced. Under mild assumptions, the RAMSA algorithm almost surely converges to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution. Finally, numerical tests have been performed with the following goals: to establish strategies for determining the value of the hyperparameters; to compare different gradient estimators and to demonstrate the effectiveness of the algorithm on problems subject to aleatory and epistemic uncertainties.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

BBO	Blackbox Optimization
DFO	Derivative-free Optimization
ZOO	Zeroth-Order Oracle
DSM	Direct Search Methods
MBM	Model-Based Methods
MADS	Mesh Adaptive Direct Search
SSO	Sequential Stochastic Optimization
CVaR	Conditional Value-a-Risk
ODE	Ordinary Differential Equation
pdf	probability density function
IS	Important Sampling
CE	Cross Entropy
RAMSA	Risk-Averse Multi-timescale Stochastic Approximation

CHAPTER 1 INTRODUCTION

The concept of optimization is deeply rooted in human nature and remarkably intuitive. Essentially, it entails the creation of a strategy within a given context to solve a problem. Even young children can grasp and excel at this concept, which often emerges during their first games of tic-tac-toe. Later in life, as they navigate their way into a physics class, they will encounter it again. In that class, they will discover many physical phenomena that adhere to the principle of least action, a principle defined by Maupertuis in 1744 as follows [171]: "When a change occurs in nature, the amount of action required for that change is minimized." A striking example of this principle is how a marble thrown into a bowl always lands at the bottom. Ultimately, it was the search for a beautiful, inexpensive vacation destination not too far from home, with a high likelihood of a mild climate, that inspired them to begin a thesis in this area - a subject that pervades every crossroads in life.

The above examples illustrate vividly the myriad of situations in our daily lives in which optimization plays a central role, whether to maximize or minimize some objective. Often, these optimization challenges require the satisfaction of specific constraints and may occur in uncertain or unpredictable environments. Typical examples include the design of aircraft, automobiles, and electronics. Applied mathematics provide a structured framework for dealing effectively with such problems, offering efficient methods for identifying the extrema of functions depending on one or more variables while satisfying predefined constraints. The following sections of this chapter are devoted to setting the framework for this research. They present some of the challenges arising in this context, outline the objectives of this thesis, and summarize the approaches developed to address them.

1.1 Problem statement

The first family of optimization problems considered in this thesis can be formulated as in [143]

$$\min_{\mathbf{x}\in\Omega} \; f(\mathbf{x}), \tag{1.1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of design variables that belongs to $\mathbb{R}^n$ and $f : \mathcal{X} \to \mathbb{R}$ is the (single) objective function to be minimized. The set $\mathcal{X} \subset \mathbb{R}^n$ represents the domain of the function; in this thesis $\mathcal{X}$ will be generally a closed subset of $\mathbb{R}^n$. Finally, Ω is the feasible set in which the

solution points must belong to, usually defined as

$$\Omega = \Big\{ \mathbf{x} \in \mathcal{X} \;\Big|\; h_i(\mathbf{x}) = 0,\ i \in \{1, \ldots, k\},\ g_j(\mathbf{x}) \leq 0,\ j \in \{1, 2, \ldots, m\} \Big\}, \tag{1.2}$$

where $h_j : \mathcal{X} \to \mathbb{R}$ and $g_j : \mathcal{X} \to \mathbb{R}$ are the constraint functions. This general formulation may be used to model most deterministic optimization problems. It is worth noting that in the context of Blackbox Optimization (BBO), only general inequality constraints may be handled as no efficient method exists to handle general equality constraints [14]. Once the problem is introduced, a characterization of the solutions to Problem (1.1) can be formalized as follows.

Definition 1.1.1. *A point $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimum of Problem (1.1) if $\mathbf{x}^* \in \Omega$ and if there exists a scalar $\epsilon > 0$ such that*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \text{ for all } \mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}^*) \cap \Omega, \tag{1.3}$$

where $\mathcal{B}_\epsilon(\mathbf{x}^*) = \{\mathbf{x} \mid ||\mathbf{x}^* - \mathbf{x}|| < \epsilon\}$. Moreover, it is called a global minimum if the inequality in eq. (1.3) holds for all $\mathbf{x} \in \Omega$.

Finding methods for approaching a point that satisfies Definition 1.1.1 is the goal of optimization research. A common technique is to generate a sequence of points $(\mathbf{x}^k)_{k \in \mathbb{N}}$ that converge to $\mathbf{x}^* \in \Omega$. In an unconstrained problem (i.e., $\Omega = \mathbb{R}^n$), this means moving from $\mathbf{x}^k$ to $\mathbf{x}^{k+1}$, following a descent direction $\mathbf{d}$, which allows to decrease the value of the objective function f. Mathematically, a descent direction can be defined as follows:

Definition 1.1.2. *A direction $\mathbf{d} \in \mathbb{R}^n$ is said to be a descent direction of the function f at a point $\mathbf{x} \in \mathbb{R}^n$ if and only if there exists a scalar $\bar{t} > 0$ such that*

$$f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x}) \text{ for all } t \in (0, \bar{t}\,].$$

Many optimization methods aim to compute a sequence of descent directions that converge to $\mathbf{x}^*$ as efficiently as possible. An example of such methods consists in computing the gradient at each point $\mathbf{x}^k$. Indeed, if f is differentiable at $\mathbf{x}^k$ and its gradient $\nabla f(\mathbf{x}^k)$ is not null, then $-\nabla f(\mathbf{x}^k)$ is a descent direction. This process allows to converge to a point $\tilde{\mathbf{x}}$ where $\nabla f(\tilde{\mathbf{x}}) = 0$. This point is called a stationary point and can be a local minimum (or even a global minimum) depending on some properties of the objective function f. This example gives an idea of why the gradient plays such an important role in many optimization methods. In constrained optimization, the gradient of the constraint functions is equally important because a stationary point can be characterized by

the colinearity between the gradients of the objective and constraint functions. While numerical optimization was a relatively new field three or four decades ago, the development of information technology has made it a very active area of research. As a result, it is no longer possible to provide an exhaustive description of all the methods. After presenting the basic concepts used in optimization methods, the remainder of this section focuses on methods in which the gradient cannot be used directly to find a descent direction. Readers interested in more general information on gradient-based numerical optimization are invited to consult [143].

The set of algorithms that do not use directly the gradient in their process can be divided into two parts. On the first hand, there is a field called Derivative-Free Optimization (DFO), which is "the mathematical study of optimization algorithms that do not use derivatives" [14]. In this field, the derivatives of the objective and constraint functions may exist, but obtaining or estimating them may be computationally expensive. A typical application of DFO is the minimization of an objective function resulting from a computer simulation whose output does not contain derivatives. On the other hand, there is a field called BBO, which is "the study of the design and analysis of algorithms that assume that the objective and/or constraint functions are given by blackboxes". A blackbox, in an optimization framework, is any process that returns an output given an input, but whose inner workings are not analytically available. In the machine learning community, the expression Zeroth-Order Oracle (ZOO) can also be used to replace the term blackbox. In BBO, there is typically no assumption of any form of continuity, differentiability, or convexity on the output. The most common uses of BBO involve computer simulations, but it can also appear in other areas, such as when the optimization problem involves a laboratory experiment [2]. Finally, BBO problems are often time-consuming due to their application areas. Each evaluation of the blackbox may take from seconds to days, requiring the use of efficient algorithms in terms of function evaluations.

Recently, one of the major concerns in BBO is the case where the output of the blackbox is subject to uncertainties. This type of problem may arise when the evaluation of the blackbox involves a Monte Carlo simulation, or when some input parameters of the blackbox are stochastic. This is the second family of optimization problems addressed in this thesis. In this case, a constrained blackbox optimization problem is formulated as in [173],

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \Xi_0[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \Xi_j[G_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in \{1,2,\ldots,m\}, \end{array} \tag{1.4}$$

where the vector $\boldsymbol{\xi}$ represents the uncertainties whose distribution is usually unknown. This vector can model all uncertainties present either in the design variables, in the parameters, in the blackbox, or even a combination of them. In practice, the uncertainty vector $\boldsymbol{\xi}$ may also de-

pend on $\mathbf{x}$. $F(\cdot, \boldsymbol{\xi})$ denotes the uncertain output of the objective function, $f : \mathcal{X} \to \mathbb{R}$ while for all $j \in \{1, 2, \ldots, m\}$, $C_j(\cdot, \boldsymbol{\xi})$ denote the uncertain versions of the constraints $c_j : \mathcal{X} \to \mathbb{R}$. Since the objective and constraint functions depend on the uncertainty vector, the measures $\Xi_j, j \in \{0, 1, \ldots, m\}$ are used to map the uncertain objective and constraint functions to $\mathbb{R}$. It follows from this formulation that the choice of the uncertainty model —from which the choice of the measures Ξ_j is derived— is critical and will be further developed in Section 2.3.2.

1.2 Challenges addressed in this thesis

Because DFO and BBO assume limited information about the structure of the problem to be solved, they may be applied to a wide range of problems. It should be noted, however, that if any information about a problem is available, it should be used in a specific algorithm [14]. This fine-tuned algorithm will generally be much more efficient than any DFO or BBO algorithm. Furthermore, the more different optimization problems an algorithm is able to handle, the more practical challenges it will face. The main practical challenges encountered by DFO and BBO algorithms are listed below:

- **Absence of gradients.** An immediate problem for the DFO and BBO algorithms is that, by definition, the gradient cannot be used directly to find a descent direction. However, using the gradient is by far the simplest and most efficient way to proceed. Traditionally, there are three main approaches that do not use the gradient to obtain a descent direction. The first approach is referred to as Direct Search Methods (DSM) [14, Part 3]. These methods find a descent direction by evaluating a finite number of points (which form a positive spanning set of $\mathbb{R}^n$) and comparing their function values. The second approach is called Model-Based Methods (MBM) [14, Part 4]. These methods construct a sufficiently accurate smooth local model of the function from given sample points. Then, an improving point is searched in the model, hoping that it is also one for the true function. Finally, the last approach is grouped under the name of metaheuristics [1]. These methods generally use different strategies based on stochastic sampling of the function. More details about these different approaches are given in chapter 2.

- **Multimodality.** A function, and by extension an optimization problem, is said to be multimodal if it has multiple local minima. Without any additional assumption about the problem, multimodality is common for problem such as (1.1). Unfortunately, algorithms tend to converge only to a local minimum. There are methods to explore the space of design variables to avoid getting stuck in local optima. However, it is known that there are currently no methods that can guarantee convergence to a global optima in finite time [177] (this claim may be

false using quantum computing). Thus, multimodality is an intractable problem, especially in BBO, where each function evaluation is time-consuming and the number of iterations is consequently limited.

- **Problem dimension.** The dimension of the problem particularly affects DFO and BBO algorithms. Again, since the gradient is not directly available, increasing the dimension of the problem complicates the search for an improving point. Consequently, traditional approaches in DFO and BBO are limited to problems with a size of a few dozens of variables. However, with the development of the machine learning community, some problems emerge with thousands or hundreds of thousands of variables (for example, the problem of black-box adversarial attack [192]). Therefore, it is necessary to develop new methods capable of dealing with this type of problem.

- **Stochasticity.** Finally, randomness makes all the previous challenges even more difficult. Indeed, when the objective function is stochastic, the definition of descent direction given in Definition 1.1.2 no longer applies. In the presence of noise, it becomes necessary to compute sufficiently accurate estimates of the uncertain objective function to obtain a reliable descent direction approximation. It is even worse when the constraint functions are also uncertain. In fact, the feasibility set of the stochastic problem depends on the measure chosen to deal with the uncertainties in the constraints. Moreover, regardless of the measure chosen, obtaining a high-confidence feasible solution requires large sampling of the constraint functions. It is also recommended to minimize the risk of outliers that could lead to an underperforming design or a design with significant failures. This is referred to as risk aversion.

Traditional approaches have primarily focused on finding efficiently local solutions for deterministic DFO and BBO problems with a limited number of variables [14]. Currently, with the development of information technology and new applications, there is a need for derivative-free methods capable of handling larger scale problems and escaping from local minima. Most importantly, dealing with stochasticity and risk aversion in problems brings a new paradigm. Although this challenge has existed for some time, it has recently experienced a resurgence [105]. In addition, most existing approaches involve retrofitting algorithms originally designed for deterministic problems to stochastic problems. While this can yield satisfactory results, these algorithms were not originally designed for this specific purpose. Therefore, it would be interesting to develop novel approaches specifically tailored to this new paradigm.

1.3 Research objectives

This work takes place in the context described in the previous section and aims to make new practical and theoretical contributions to the field of DFO and BBO. In particular, it focuses on the four issues described above: absence of gradients, multimodality, stochasticity and risk aversion; knowing that the dimensionality issue underlies each of these problems. Interestingly, all four issues can be addressed using a single mathematical concept: Gaussian exploration of the space of design variables [141]. Gaussian exploration consists of evaluating some points generated by a Gaussian distribution centered on a given mean point and with a given standard deviation. Then, there are essentially two different approaches to exploit the output of the blackbox at the evaluated points. The direct approach involves sorting the various points according to some quantity of interest and declaring the new current point as the best point found. The indirect approach is to compute a possible descent direction from the blackbox returned values and move along that direction. These approaches are attractive in the context of DFO or BBO. On the one hand, the direct approach may help to efficiently explore the space of design variables and escape from local minima [99]. On the other hand, the indirect approach may efficiently approximate stochastic gradients [141] using only noisy function evaluations. Therefore, the various goals of this thesis are articulated around these two approaches as follows.

- The first goal is to fully exploit the properties of Gaussian random exploration to develop algorithms that address the issues identified above in the context of DFO/ BBO: absence of the gradient, multimodality, stochasticity and risk aversion.
- Theoretically, these algorithms must rely on strong convergence properties as a guarantee of reliability. In particular, in the case of stochastic blackbox optimization, almost sure convergence to an optimal point or a neighborhood of an optimal point must be shown. Furthermore, the rate of convergence must be derived whenever possible.
- In practice, the developed algorithms must be adapted to the context of DFO/BBO. In particular, they must be efficient in terms of function evaluations. They must also be applicable to a wide range of problem dimensions.
- Finally, the developed algorithms must be easily implementable and numerical comparisons with other state-of-the-art algorithms must demonstrate their competitiveness.

CHAPTER 2 LITTERATURE REVIEW

The first part of this chapter (Section 2.1) is devoted to the mathematical backgrounds of the various topics discussed in the thesis. Section 2.2 and Section 2.3 present the state-of-the-art on these subjects.

2.1 Theoretical background

This section introduces some general notions of optimization (pp. 9-13), probability (pp. 13-17), and ordinary differential equations (pp. 17-19). These notions will be used in various parts of the thesis for the definitions of the algorithms and their convergence results. They will not be developed here, but references with details are given at the beginning of each subsection. Readers can go to the pages (indicated above) of the topic they wish to investigate, or even directly to the literature review in Section 2.2.

2.1.1 Fundamental principles of optimization theory

Reminders about optimization are mainly taken from [14, Chapter 2 and 6] and [173, Chapter 9.1]. The most important optimization results rely on structural properties of the underlying function. In this section, the most common properties are defined: the degree of continuity and smoothness, as well as convexity.

Definition 2.1.1 (Continuity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be continuous at a point $\mathbf{x} \in \mathbb{R}^n$ if*

$$\forall \epsilon > 0,\ \exists \rho \text{ such that } \forall \mathbf{y} \in \mathbb{R}^n, ||\mathbf{x} - \mathbf{y}|| < \rho \implies |f(\mathbf{x}) - f(\mathbf{y})| < \epsilon.$$

The notation $f \in \mathcal{C}^0$ means that f is continuous at any point $\mathbf{x} \in \mathbb{R}^n$.

Definition 2.1.2 (Lipschitz continuity)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be Lipschitz continuous if and only if there exists a scalar $L > 0$ for which*

$$||f(\mathbf{x}) - f(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}|| \text{ for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n,$$

where L is called the Lipschitz constant of f.

The notation $f \in \mathcal{C}^{0+}$ means that f is Lipschitz continuous.

Definition 2.1.3 (Differentiability)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be differentiable at a point $\mathbf{x} \in \mathbb{R}^n$ if and only if there exists a vector $\mathbf{g} \in \mathbb{R}^n$ such that*

$$\lim_{\mathbf{y} \to \mathbf{x}} \frac{f(\mathbf{y}) - f(\mathbf{x}) - \mathbf{g}^T(\mathbf{y} - \mathbf{x})}{||\mathbf{y} - \mathbf{x}||} = 0.$$

If f is differentiable at $\mathbf{x}$, then there is a unique such vector $\mathbf{g}$, and it is called the gradient of f at $\mathbf{x}$, denoted $\nabla f(\mathbf{x})$.

When f is differentiable at some point $\mathbf{x} \in \mathbb{R}^n$, the gradient is the vector of the n partial derivatives of f

$$\forall \mathbf{x},\ \nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right]^T.$$

The notation $f \in \mathcal{C}^1$ means that f is differentiable and its partial derivatives are continuous. We extend this notation to $f \in \mathcal{C}^{1+}$ to mean that f is differentiable and its gradient is Lipschitz continuous. If f belongs to both $\mathcal{C}^{0+}$ and $\mathcal{C}^{1+}$, then $L^0(f)$ and $L^1(f)$ denote the Lipschitz constants of the function and its gradient, respectively. Finally, the following lemma gives a characterization of the function of class $\mathcal{C}^{1+}$. This characterization is often employed to calculate the convergence rate of an algorithm.

Lemma 2.1.4. *A function $f \in \mathcal{C}^{1+}$ if and only if the following inequality holds:*

$$|f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})| \leq \frac{1}{2} L^1(f) ||\mathbf{x} - \mathbf{y}||^2, \text{ for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

It is possible for a function to be differentiable only in a finite set of directions. In this case, only some directional derivatives exist. The directional derivatives of f at $\mathbf{x}$ in the direction $\mathbf{d} \in \mathbb{R}^n$ are defined as

$$f'(\mathbf{x}; \mathbf{d}) := \lim_{t \searrow 0} \frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{h}.$$

It is worth noting that directional derivatives "generalize" the notion of differentiability, since the directional derivatives of the function may still exist even if $f \notin \mathcal{C}^1$ (e.g., $f(x) = |x|$). However, the assumption of the existence of a directional derivative remains a strong assumption in a black-box optimization framework. Thus, a generalization of directional derivatives is needed to be used with any Lipschitz function.

Definition 2.1.5 (Clarke generalized directional derivative [52]). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz near $\mathbf{x} \in \mathbb{R}^n$. The Clarke generalized directional derivative of f at $\mathbf{x}$ in the direction $\mathbf{d} \in \mathbb{R}^n$ is*

$$f^\circ(\mathbf{x}; \mathbf{d}) = \limsup_{y \to x,\ t \searrow 0} \frac{f(\mathbf{y} + t\mathbf{d}) - f(\mathbf{y})}{h}.$$

The differences with the directional derivatives are due to the limit superior and to the implicit consideration of all sequences $\mathbf{y}$ that converge to $\mathbf{x}$.

Definition 2.1.6 (Limit superior). *The limit superior of the function $f : \mathbb{R}^n \to \mathbb{R}$ at $\mathbf{x} \in \mathbb{R}^n$ is defined to be*

$$\limsup_{y \to x} = \lim_{r \searrow 0} u_{\mathbf{x}}(r),$$

with

$$u_{\mathbf{x}}(r) = \sup_{\mathbf{y} \in \mathcal{B}_r(\mathbf{x})} f(\mathbf{y}), \text{ where } \mathcal{B}_r(\mathbf{x}) = \{\mathbf{y} \mid ||\mathbf{y} - \mathbf{x}|| < r\}.$$

Finally, an important class of functions in optimization are convex functions.

Definition 2.1.7 (Convex function). *The function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}) \text{ for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \text{ and } t \in [0, 1].$$

Once the structural properties of a function are defined, the necessary optimality conditions may be established. In the unconstrained case, when the function is differentiable, the first-order optimality condition may be stated as follows.

Theorem 2.1.8 (Fermat's Theorem). *If $f \in \mathcal{C}^1$ and $\mathbf{x}^*$ is a local minimum of f, then $\nabla f(\mathbf{x}) = 0$.*

Note that this condition is not sufficient. A point satisfying Fermat's Theorem is called a stationary point. In the blackbox framework, since the gradient may not exist, the first-order optimality condition is given in terms of the generalized directional derivatives.

Theorem 2.1.9 (First-order optimality condition via f°). *If f is Lipschitz continuous near $\mathbf{x}^*$, a local minimum of f, then $f^\circ(\mathbf{x}; \mathbf{d}) \geq 0$ for every $\mathbf{d} \in \mathbb{R}^n$.*

When the problem is constrained, the conditions become a bit more complex and need the notion of cone.

Definition 2.1.10 (Cone)**.** *A set $T \subset \mathbb{R}^n$ is said to be a cone if and only if $\lambda \mathbf{d} \in T$ for every scalar $\lambda > 0$ and for every $\mathbf{d} \in T$.*

Again, there is a distinction between the differentiable and the non-differentiable case. In the differentiable case, the optimality condition is based on the tangent cone.

Definition 2.1.11 (Tangent cone)**.** *The tangent cone to Ω at $\mathbf{x}$, denoted $T_\Omega(\mathbf{x})$, is the set of all tangent directions to Ω at $\mathbf{x}$, i.e., the directions $\mathbf{d} \in \mathbb{R}^n$ for which there exists sequences $(\mathbf{d}^k)_{k\in\mathbb{N}}$, with $\forall k,\ \mathbf{d}^k \in \mathbb{R}^n$, and $(t^k)_{k\in\mathbb{N}}$, with $\forall k,\ t^k \in \mathbb{R}^+$, such that*

$$\mathbf{x} + t^k \mathbf{d}^k \in \Omega, \forall k,\ \mathbf{d}^k \to \mathbf{d} \ \textit{and}\ \ t^k \searrow 0.$$

Once the tangent cone is defined, the first-order optimality condition is stated in the following theorem.

Theorem 2.1.12 (Constrained first-order optimality condition)**.** *If $f \in \mathcal{C}^1$ and $\mathbf{x}^*$ is a local minimum of Problem (1.1), then*

$$\nabla f(\mathbf{x}^*)^\top \mathbf{d} \geq 0, \textit{ for all } \mathbf{d} \in T_\Omega(\mathbf{x}^*).$$

In the non-differentiable case, the optimality condition is based on the hypertangent cone.

Definition 2.1.13 (Hypertangent cone)**.** *The hypertangent cone to Ω at $\mathbf{x}$, denoted $T_\Omega^H(\mathbf{x})$, is the set of all hypertangent directions to Ω at $\mathbf{x}$, i.e, the directions $\mathbf{d} \in \mathbb{R}^n$ for which there exists sequences $(\mathbf{x}^k)_{k\in\mathbb{N}}$, with $\forall k\, \mathbf{x}^k \in \Omega$, $(\mathbf{d}^k)_{k\in\mathbb{N}}$, with $\forall k,\ \mathbf{d}^k \in \mathbb{R}^n$, and $(t^k)_{k\in\mathbb{N}}$, with $\forall k,\ t^k \in \mathbb{R}^+$, such that*

$$\mathbf{x}^k + t^k \mathbf{d}^k \in \Omega\ \forall k,\ \mathbf{x}^k \to \mathbf{x},\ \mathbf{d}^k \to \mathbf{d} \ \textit{and}\ \ t^k \searrow 0.$$

Now, the constrained first order optimality condition may be stated in terms of generalized direction derivatives.

Theorem 2.1.14. *If f is Lipschitz continuous near $\mathbf{x}^*$, a local solution to Problem (1.1), then*

$$f^\circ(\mathbf{x}^*, \mathbf{d}) \geq 0, \text{ for all } \in T_\Omega^H(\mathbf{x}^*).$$

Finally, another way to deal with constrained problems is to use the Lagrangian relaxation. Let consider the Problem (1.1) where Ω only contains m inequality constraints. The Lagrangian of the problem is defined as the function $L : \mathcal{X} \times \mathbb{R}^m \to \mathbb{R}$

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x}).$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ is the vector of Lagrangian multipliers. The Lagrangian relaxation is often used in optimization to handle constraint functions. The saddle points of the Lagrangian function may be defined as follows.

Definition 2.1.15 (Saddle point)**.** *A saddle point of L is a couple $(\mathbf{x}^*, \boldsymbol{\lambda}^*) \in \mathcal{X} \times \mathbb{R}^m$ such that for some $r > 0$, $\forall \mathbf{x} \in \mathcal{X} \cap \mathcal{B}_r(\mathbf{x}^*)$ and for all $\boldsymbol{\lambda} \geq \mathbf{0}$, we have*

$$L(\mathbf{x}^*, \boldsymbol{\lambda}) \leq L(\mathbf{x}^*, \boldsymbol{\lambda}^*) \leq L(\mathbf{x}, \boldsymbol{\lambda}^*),$$

where $\mathcal{B}_{\mathbf{x}^*}(r)$ is a hyper-dimensional ball centered at $\mathbf{x}^*$ with radius $r > 0$.

The link between the saddle points of the Lagrangian function and Problem (1.1) is specified in the next theorem.

Theorem 2.1.16 (Saddle point theorem)**.** *If $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a saddle point of the Lagrangian function L, then $\mathbf{x}^*$ is a local solution to Problem (1.1).*

2.1.2 Fundamental principles of probability theory

Reminders about probability are mainly taken from [173, Chapter 9.2] for general results and from [38, Chapter 11.3] for random processes. First, the concepts of probability space, random variables, and expectation are defined. Although quite intuitive, the formal definitions of these concepts are difficult because of their links to measure theory. It starts with the concepts of σ-field, measurable space and measurable function.

Definition 2.1.17 (σ-field and measurable space)**.** *Let Ω be a set, a σ-field $\mathcal{F}$ on Ω is a nonempty collection of subsets of Ω such that:*

- $\Omega \in \mathcal{F}$*;*
- *if $B \in \mathcal{F}$, then its complement B^c is also in $\mathcal{F}$;*
- *if a countable set of elements $B_1, B_2 \ldots$ are in $\mathcal{F}$, then so is $B = B_1 \cup B_2 \cup \ldots$.*

The tuple $(\Omega, \mathcal{F})$ is said to be a measurable space.

Definition 2.1.18 (Measure)**.** *Let $(\Omega, \mathcal{F})$ be a measurable space. An application $\mu : \mathcal{F} \to \mathbb{R}^+$ is said to be a measure if the following holds.*

- $\mu(\emptyset) = 0$*;*
- *if a countable set of elements $B_1, B_2, \ldots$ are in $\mathcal{F}$ and are two-by-two disjoint, then*

$$\mu\left(\bigcup_{k=1}^{\infty} B_k\right) = \sum_{k=1}^{\infty} \mu(B_k). \tag{2.1}$$

Definition 2.1.19 (Measurable function)**.** *Let $(\Omega, \mathcal{F})$ and $(\Xi, \mathcal{E})$ be two measurable space. A function $f : \Omega \to \Xi$ is said to be $\mathcal{E}$-measurable if*

$$\forall B \in \mathcal{E},\ f^{-1}(B) = \{\omega \in \Omega | f(\omega) \in B\} \in \mathcal{F}.$$

Now these concepts are defined, it is possible to present the particularity of probability theory, based on the probability measure, the probability space and the random variables.

Definition 2.1.20 (Probability measure)**.** *Let $\mathbb{P}$ be a measure on the measurable space $(\Omega, \mathcal{F})$ such that $\mathbb{P}(\Omega) = 1$. Then $\mathbb{P}$ is said to be a probability measure. Moreover, the triplet $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space.*

Definition 2.1.21 (Random variables)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\Xi, \mathcal{E})$ a measurable space. Any measurable function ξ from Ω to Ξ is called a random variable.*

Definition 2.1.22 (Generated σ-field)**.** *Let $\{\xi_i\}_{i\in\mathbb{N}}$ a collection of random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ and having their images in the measurable spaces $(\Xi_i, \mathcal{E}_i)_{i\in\mathbb{N}}$. The smallest σ-field which contains the union of $\xi_i^{-1}(\mathcal{E}_i)$ is called the generated σ-field.*

When $(\Xi, \mathcal{E}) = (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mathcal{B}(\mathbb{R}^d))$ is the Borel sigma-field, i.e., the one generated by the open sets of $\mathbb{R}^d$, the random variables are called real random vectors. Finally, one of the main concepts of probability may be defined (when it exists): the expectation.

Definition 2.1.23 (Integrability)**.** *Let $p \in [1, \infty)$ and $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. The space $L^p(\Omega, \mathcal{F}, \mathbb{P})$ which contains the p-integrable random variables is the set of all real random variables X such that*

$$||X||_p = \left(\int_\Omega |X(\omega)|^p \mathbb{P}(d\omega)\right)^{\frac{1}{p}} < +\infty.$$

Definition 2.1.24 (Expectation)**.** *Let $X \in L^1(\Omega, \mathcal{F}, \mathbb{P})$, the expectation of ξ is defined as follows*

$$\mathbb{E}_\xi[X] := \int_\Omega X(\omega)\mathbb{P}(d\omega).$$

The deviation of a random variable from its expectation is often of statistical interest. In this thesis, two measures for a real random variable are employed: the quantile and the variance. The quantile of a real random variable X can be defined as follows

Definition 2.1.25 (Quantile)**.** *The (left-side) quantile of X at level α is*

$$q(\alpha) := \inf\{x \mid \mathbb{P}(X \leq x) \geq \alpha\}.$$

The variance is the expected value of the squared deviation from the mean of a random variable.

Definition 2.1.26 (Variance)**.** *Let $X \in L^2(\Omega, \mathcal{F}, \mathbb{P})$, the variance is defined as*

$$\mathbb{V}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2].$$

Another notion widely utilized in this thesis is the expectation given that a certain set of events is known to occur, this is called the conditional expectation. Let $\mathcal{G}$ a sub σ-field of $\mathcal{F}$.

Definition 2.1.27 (Conditional expectation)**.** *Let X be an integrable random variable defined on the space $(\Omega, \mathcal{F})$. The conditional expectation of X given $\mathcal{G}$ is the unique random variable $\mathcal{G}$-measurable $Z = \mathbb{E}[X|\mathcal{G}]$ such that for all integrable $\mathcal{G}$-measurable, the following equality holds*

$$\mathbb{E}[XY] = \mathbb{E}[ZY].$$

The conditional expectation is particularly useful for studying stochastic random processes. Its properties are given in the following proposition.

Proposition 2.1.28. *The conditional expectation satisfies the following properties.*

- *Linearity, let a, b two scalars it follows that*

$$\mathbb{E}[aX + bY|\mathcal{G}] = a\mathbb{E}[X \mid \mathcal{G}] + b\mathbb{E}[Y \mid \mathcal{G}].$$

- *Law of total expectation*

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|\mathcal{G}]].$$

- *If X is $\mathcal{G}$-measurable, then*

$$\mathbb{E}[X|\mathcal{G}] = X.$$

- *If X is independent of $\mathcal{G}$*

$$\mathbb{E}[X|\mathcal{G}] = \mathbb{E}[X].$$

- *If $\mathcal{H} \subset \mathcal{G}$*

$$\mathbb{E}[X|\mathcal{H}] = \mathbb{E}[\mathbb{E}[X|\mathcal{G}]|\mathcal{H}].$$

- *Jensen's inequality (which is true for the expectation as well). If f is a convex function, then*

$$f(\mathbb{E}[X|\mathcal{G}]) \leq \mathbb{E}[f(X)|\mathcal{G}].$$

Like the conditional expectation, the conditional variance can also be defined.

Definition 2.1.29 (Conditional variance)**.** *Let X be a squared integrable random variable, the conditional variance is defined as*

$$\mathbb{V}[X|\mathcal{G}] = \mathbb{E}[(X - \mathbb{E}[X|\mathcal{G}])^2|\mathcal{G}].$$

Discrete-time stochastic processes

In this work, the properties of a sequence of random variables will be studied. Such a sequence defines a stochastic (discrete-time) process.

Definition 2.1.30. *A discrete time stochastic process is a sequence of random vectors $(\mathbf{x}^k)_{k\in\mathbb{N}}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with their values in $\mathbb{R}^n$.*

The convergence of this type of sequence is not usual, here are presented two types of convergence result. The first type of convergence is the convergence in expectation.

Definition 2.1.31 (Convergence in expectation)**.** *A discrete time stochastic process $(\mathbf{x}^k)_{k\in\mathbb{N}}$ converges in expectation to $\mathbf{x}$, defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with its value in $\mathbb{R}^n$, if*

$$\mathbb{E}[||\mathbf{x}^k - \mathbf{x}||] \to 0, \text{ when } k \to +\infty.$$

A stronger type of convergence (in the sens that it implies the convergence in expectation) is the almost-sure convergence.

Definition 2.1.32 (Convergence almost-sure)**.** *A discrete time stochastic process $(\mathbf{x}^k)_{k\in\mathbb{N}}$ converges almost surely (a.s.) to $\mathbf{x}$ if and only if there exist $A \in \mathcal{F}$ such that $\mathbb{P}(A) = 0$ and*

$$\forall \omega \in A^c,\ \mathbf{x}^k(\omega) \underset{k\to\infty}{\longrightarrow} \mathbf{x}(\omega).$$

Another important topic of the discrete-time stochastic processes is the martingale. First, the definition of a filtration is needed.

Definition 2.1.33 (Filtration)**.** *A filtration is a sequence of increasing subalgebra* $(\mathcal{F}^k)$*, i.e,*

$$\mathcal{F}^1 \subset \mathcal{F}^2 \subset \ldots \subset \mathcal{F}^k.$$

The filtration contains all the information acquired in the stochastic process until time k.

Definition 2.1.34 (Martingale)**.** *Let* $(\mathbf{x}^k) \in \mathbb{R}^n$ *be an integrable discrete time stochastic process and* $(\mathcal{F}^k)$ *be its associated* σ*-field (i.e,* $\mathbb{E}[\mathbf{x}^k|\mathcal{F}^k] = \mathbf{x}^k$*). The process is a martingale if*

$$\forall k,\ \mathbb{E}[\mathbf{x}^{k+1}|\mathcal{F}^k] = \mathbf{x}^k \text{ a.s.}$$

Besides, the process is called a martingale difference sequence if

$$\forall k,\ \mathbb{E}[\mathbf{x}^{k+1}|\mathcal{F}^k] = 0 \text{ a.s.}$$

2.1.3 Fundamental principles of Ordinary Differential Equation (ODE)

This section is required since some convergence result are obtained thanks to an ordinary differential equation (ODE) approach. The main results of this section come from the work in [91, Chapter 4]. Consider the following ODE

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{2.2}$$

where $f : D \to \mathbb{R}^n$ is a locally Lipschitz function and $D \subset \mathbb{R}^n$. Let $\mathbf{x}^*$ be an equilibrium point of the ODE, i.e. $f(\mathbf{x}^*) = 0$. The goal is to characterize the stability of $\mathbf{x}^*$.

Definition 2.1.35 (Stability)**.** *The equilibrium point* $\mathbf{x}^*$ *of Equation* (2.2) *is*

- *stable, if for each* $\epsilon > 0$*, there is* $\delta > 0$ *such that*

$$||\mathbf{x}(0) - \mathbf{x}^*|| < \delta \implies ||\mathbf{x}(t) - \mathbf{x}^*|| < \epsilon,\ \forall t \geq 0,$$

- *asymptotically stable if it is stable and* δ *can be chosen such that*

$$||\mathbf{x}(0) - \mathbf{x}^*|| < \delta \implies \lim_{t\to\infty} ||\mathbf{x}(t) - \mathbf{x}^*|| = 0.$$

A first result on the stability of an equilibrium point is given in the next theorem.

Theorem 2.1.36 (Lyapunov stability theorem)**.** *Let $\mathbf{x}^*$ an equilibrium point for Equation (2.2) and $D \subset \mathbb{R}^n$ be a domain containing $\mathbf{x}^*$. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that*

- $V(\mathbf{x}) = 0$ *if and only if* $\mathbf{x} = \mathbf{x}^*$,
- $V(\mathbf{x}) > 0$ *if and only if* $\mathbf{x} \neq \mathbf{x}^*$,
- $\dot{V}(\mathbf{x}) = \frac{d}{dt}V(\mathbf{x}) = (\nabla V)^T f(\mathbf{x}) \leq 0$ *for all values of* $\mathbf{x} \neq \mathbf{x}^*$.

Then, V is called a Lyapunov function and the equilibrium $\mathbf{x}^*$ is stable. Moreover, if the last item is replaced by $\dot{V}(\mathbf{x}) < 0$ for all values of $\mathbf{x} \neq \mathbf{x}^*$, then $\mathbf{x}^*$ is asymptotically stable.

When the point $\mathbf{x}^*$ is asymptotically stable, it can be interesting to know how far from the equilibrium the started point may be and still converge to the equilibrium when t goes to infinity.

Definition 2.1.37 (Region of attraction)**.** *Let $\phi(t; \mathbf{x})$ be the solution of Equation (2.2) that starts at initial state $\mathbf{x}$ at time $t = 0$. Then, the region of attraction is defined as the set of all points $\mathbf{x}$ such that $\phi(t; \mathbf{x})$ is defined for all $t \geq 0$ and $\lim_{t\to\infty} \phi(t; \mathbf{x}) = \mathbf{x}^*$.*

Definition 2.1.38 (Globally asymptotically stable equilibrium)**.** *If for any initial state $\mathbf{x}$, the trajectory $\phi(t; \mathbf{x})$ approaches $\mathbf{x}^*$ as $t \to \infty$, no matter how large $||\mathbf{x}||$ is, then $\mathbf{x}^*$ is said to be globally asymptotically stable.*

It is possible to know if an equilibrium is globally asymptotically stable thanks to the Lyapunov function.

Theorem 2.1.39. *Let $\mathbf{x}^*$ be an equilibrium point for Equation (2.2) and let V be Lyapunov function such that*

- $\dot{V} < 0$ *for all values of* $\mathbf{x} \neq \mathbf{x}^*$,
- *V is radially unbounded, i.e.,* $||\mathbf{x}|| \to \infty \implies V(\mathbf{x}) \to \infty$,

then $\mathbf{x}^*$ is globally asymptotically stable.

Finally, in order to obtain the same stability result with milder assumption on the function V, the following definition is needed.

Definition 2.1.40. *A set M is said to be a (positively) invariant set if*

$$\mathbf{x}(0) \in M \implies \mathbf{x}(t) \in M,\ \forall t(\geq 0).$$

Now, the LaSalle invariance principle may be stated.

Theorem 2.1.41 (LaSalle theorem)**.** *Let $\Omega \subset D$ be a compact set that is positively invariant with respect to Equation (2.2). Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(\mathbf{x}) \leq 0$ in Ω. Let E be the set of all points in Ω where $\dot{V}(\mathbf{x}) = 0$. Let M the largest invariant set in E. Then, every solution starting in Ω approaches M as $t \to \infty$.*

2.2 Deterministic blackbox optimization

This section presents a historical overview of DFO and BBO algorithms in the context of deterministic blackbox optimization problems. Section 2.2.1 describes some metaheuristics used in this field, Section 2.2.2 details the model-based methods and Section 2.2.3 depicts the direct search methods. The main difference between the first category of methods and the last two is that the first does not rely on a proof of convergence, unlike the other two. A detailed review of derivative-free algorithms (without metaheuristics) can be found in [105] and one of metaheuristics in [1].

2.2.1 Main metaheuristics

The word "metaheuristic" has been introduced to refer to heuristics that are not problem specific. All metaheuristics are based on the same mechanism: a trade-off between exploitation and exploration [1]. Exploitation is the process of finding the best solutions in a neighboring region, while exploration is the process of finding the most promising region. While these methods may produce excellent results in practice, it is worth noting that, first, they are usually approximate. Second, on average, all metaheuristics achieve the same performance on all problems, i.e., they perform as well as a random search [189]. Therefore, it may often be necessary to combine them with other algorithms to ensure their convergence/performance.

Many metaheuristics are population-based algorithms. The common feature of these approaches is the population, which is a set of points in the design space that evolves towards a minimum of the function. The approaches differ in their mechanism for evolving the population. There are three main types: genetic algorithms, swarm intelligence, and evolution strategies. Since the work on population-based algorithms is tremendous, only well-established methods will be reviewed in this

thesis. Emphasis will also be placed on the methods used in the numerical comparisons of this thesis. References to more detailed work are provided for the interested reader.

Genetic algorithms [88] are evolutionary algorithms that use "survival of the fittest" in their inherent mechanism to search for a better solution. In keeping with the biological theme of the method, the process to obtain a new point uses mainly four steps called : encoding, selection, crossover and mutations. A pseudocode of a classical genetic algorithm is given in Algorithm 1. Research on

Algorithm 1 Genetic algorithm

K: Maximum number of iterations
S: Size of the popultation
$k = 0$
while $k \leq K$ **do**
    Selection: select a pair of points according to its fitness value, i.e., its quality
    Crossover: use the two parent points to create an offspring
    Mutation: mutate the offspring with a certain probability
    Update: Replace the old population with the newly generated population
    $k \leftarrow k + 1$
end while
Return the best point

genetic algorithms is concerned with the choice of fitness measure, the type of selection used, the encoding scheme, and the choice of crossover and mutation operators. A study of the pros and cons of different techniques is given in the work in [88].

Swarm intelligence algorithms are evolutionary algorithms that use collective behavior to update their population. The main examples of these methods are Ant Colony Optimization (ACO) [60] and Particle Swarm Optimization (PSO) [90]. Since Ant Colony Optimization is more suited to combinatorial optimization, the focus will be on the PSO algorithms. A basic variant of the PSO algorithm works with a population (called a swarm) of candidate solutions (called a particle) that move through the search space according to a few simple formulae (given below). These formulae were originally based on a model simulating the movement of a group of birds. The movement of each particle is guided by its own best-known position in the search space, as well as by the best-known position of the entire swarm. As better positions are found, they guide the swarm's movements. The process is then repeated in the hope of finding a satisfactory solution. Traditionally, at each iteration k, the motion of a particle $\mathbf{x}_i$ is determined by three components: its velocity $\mathbf{V}_i^k$, its own best solution $\mathbf{x}_i^{k,*}$, at iteration k, and the best solution shared by all particles $\mathbf{x}^{k,*}$. The equations governing the motion of a particle are as follows,

$$\begin{aligned}\mathbf{V}_i^{k+1} &= \omega V_i^k + b_1(\mathbf{x}_i^k - \mathbf{x}_i^{k,*}) + b_2(\mathbf{x}_i^k - \mathbf{x}^{k,*}),\\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{V}_i^{k+1},\end{aligned}$$

where ω is a deterministic parameter, while b_1 and b_2 are two random variables uniformly chosen in $[0, \phi_1]$ and $[0, \phi_2]$. Thus, PSO algorithms have the advantage of requiring only three hyperparameters. Of course, there are many variants of PSO algorithms, which the interested reader can find in [186]. Finally, as for genetic algorithms, the PSO algorithms were originally made for unconstrained problem and has been extended to handle general inequality constraints, see for example the work in [146].

Evolution strategies are a group of algorithms based on artificial evolution methods (as opposed to genetic algorithms, which are based on biological evolution methods). The best known of these methods is the covariance matrix adaptation evolution strategy (CMA-ES) [79]. CMA-ES is based on a selection and adaptation process of a candidate population. In CMA-ES, at each generation, λ offspring candidates are generated from μ parents. Unlike the genetic algorithm, the offspring candidates are generated from a multivariate normal distribution. In the next generation, to select the new parents, the best μ offspring candidates are selected in terms of their ranking according to their objective function values. This process is repeated iteratively to hopefully converge and shrink around the global minimum. A pseudocode of this algorithm is given in Algorithm 2. Like

Algorithm 2 CMA-ES algorithm

K: maximum number of iterations
λ the number of offspring candidates and μ the number of parents
$\mathbf{m}^0$: a random point of the search space and μ different weights w_i
σ^0 the stepsize and $\mathbf{C}^0 = \mathbf{I}$ the covariance matrix
$k = 0$
while $k \leq K$ **do**
    Sampling: generate λ offspring candidates according to

$$\mathbf{z}_t^{k+1} \sim \mathbf{m}^k + \sigma^k \mathcal{N}(\mathbf{0}, \mathbf{C}^k),\ t \in [1, \lambda] \tag{2.3}$$

    Ranking: evaluate the candidates and rank them based on their objective function value
    Averaging: compute the weighted average solution

$$\mathbf{m}^{k+1} = \sum_{t=1}^{\mu} w_i \mathbf{z}_t^{k+1} \tag{2.4}$$

    Update: the covariance matrix $\mathbf{C}^{k+1}$ and the stepsize σ^{k+1} as in [79]
    $k \leftarrow k + 1$
end while
Return the best point

the other metaheuristics, CMA-ES was developed to solve unconstrained optimization problems. Versions of CMA-ES adapted for constrained problems are based on the penalty approach [126] or modifications of the original algorithm [7].

Another type of metaheuristic is based on neighborhood search. Neighborhood search aims to escape from a local minimum when the algorithm seems to be stuck there. The three most popular

methods of this type are: variable neighborhood search, simulated annealing, and tabu search.

The Variable Neighborhood Search (VNS) algorithm [135] explores distant neighborhoods of the current incumbent solution, and moves from there to a new solution if and only if an improvement has been made. The VNS algorithm requires two elements: a neighborhood structure and a method to search locally for a better solution within the neighborhood structure. The advantage of the VNS algorithm is that it can be coupled with any other local search method. A pseudocode of this algorithm is given in Algorithm 3.

Algorithm 3 Variable neighborhood search algorithm

K: Maximum number of iterations
ρ^0: initial search length and $\rho^{\max}$ the maximal search length
$\delta \in \mathbb{N}$
$k = 0$
while $k \leq K$ **do**
    $\rho^k \leftarrow \rho^0$
    while $\rho^k \leq \rho^{\max}$ **do**
        $\mathbf{x}' \leftarrow shaking(\mathbf{x}^k, \rho^k)$
        $\mathbf{x}'' \leftarrow localsearch(\mathbf{x}')$
        if $f(\mathbf{x}'') < f(\mathbf{x}^k)$ **then**
            $x^{k+1} \leftarrow \mathbf{x}''$ and $\rho^{k+1} \leftarrow \rho^0$
        else
            $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$ and $\rho^{k+1} \leftarrow \rho^k + \delta$
        end if
        $k \leftarrow k + 1$
    end while
end while
Return the best point

Tabu search [74] is a local search whose basic rule has been relaxed so that worse moves can be accepted if no better move is available. Like the VNS algorithm, the tabu search algorithm requires a neighborhood structure within which it can move. In addition, to avoid returning to a previously visited solution, the algorithm has a short-term memory that stores these solutions. Since tabu search is mainly used in combinatorial optimization, it will not be discussed further.

Finally, Simulated Annealing [95] is based on the evolution of a thermodynamic system. In analogy to the physical process, the function to be minimized becomes the energy E. In addition, a fictitious parameter is introduced, the temperature T of the system. By modifying a given state of the system, we obtain another state. This state either improves the value of the objective function - the energy of the system is said to be reduced - or degrades it. If we accept a state that improves the objective function value, we are looking for a local solution. On the contrary, accepting a degrading state allows exploring a larger part of the state space and tends to avoid getting bogged down too quickly in the search for a local optimum. As with the previous algorithms in this paragraph, in practice the algorithm needs a neighborhood structure to identify neighboring points. At each iteration k,

a neighboring point is evaluated. This modification leads to a variation Δf in the value of the objective function. If this variation is negative (i.e., the iteration is an improvement), the algorithm moves to the neighboring point. Otherwise, it moves only with probability $e^{-\frac{\Delta f}{T}}$. This choice of exponential probability is known as the Metropolis rule. The process is then iterated, lowering the temperature or keeping it constant, depending on the approaches.

Probabilistic-based metaheuristics may also be used like Importance Sampling (IS) [102]. It may be used in two different perspectives:

- Estimation [102]: typically, estimate $\ell = \mathbb{E}[h(\mathbf{X})]$, where $\mathbf{X}$ is a random vector having its value in $\mathcal{X}$ a subset of $\mathbb{R}^n$ and h is a function on $\mathcal{X}$.
- Optimization [36, 99]: minimize a given objective function f over all $\mathbf{x} \in \mathcal{X}$.

These two perspectives are in fact two points of view of a same image. In the following, the concept of IS will be presented under the estimation perspective and explanations will be given to obtain the optimization perspective. Consider the estimation of the following probability

$$\ell = \mathbb{P}(f(\mathbf{X}) \leq \gamma) = \mathbb{E}[I_{\{f(\mathbf{X})\leq\gamma\}}] = \int I_{\{f(\mathbf{X})\leq\gamma\}}\phi(\mathbf{x})d\mathbf{x},$$

where f is a function, γ is a given threshold, and $\mathbf{X}$ has a probability density function (pdf) $\phi(\cdot)$. If ℓ is a very small probability (called a rare event), its estimation from the pdf ϕ requires a huge number of samples. Therefore, the IS method looks for another pdf ψ that is better able to estimate ℓ. Using the pdf ψ, ℓ is equal to

$$\ell = \int \frac{I_{\{f(\mathbf{X})\leq\gamma\}}\phi(\mathbf{x})}{\psi(\mathbf{x})}\psi(\mathbf{x})d\mathbf{x} = \mathbb{E}\left[\frac{I_{\{f(\mathbf{X})\leq\gamma\}}\phi(\mathbf{X})}{\psi(\mathbf{X})}\right],$$

where $\mathbf{X} \sim \psi$. Therefore, if $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are N independent vectors sampling from the pdf ψ, an estimator $\hat{\ell}$ of ℓ may be computed as follows

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N}\frac{I_{\{f(\mathbf{X}_i)\leq\gamma\}}\phi(\mathbf{X}_i)}{\psi(\mathbf{X}_i)}.$$

It is an unbiased estimator: a so-called importance sampling estimator and ψ is the importance sampling pdf. The optimal sampling pdf ψ^*, the one for which the variance of $\hat{\ell}$ is minimal, is then the density of $\mathbf{X}$ conditional on the event $f(\mathbf{X}) \leq \gamma$, that is

$$\psi^*(\mathbf{x}) = \frac{\phi(\mathbf{x})I_{\{f(\mathbf{X})\leq\gamma\}}}{\ell}.$$

Unfortunately, this optimal density is intractable because it involves the unknown quantity of inter-

est ℓ. Importance sampling aims to approximate this optimal auxiliary density, and there are two main methods: the parametric and the nonparametric approaches.

Parametric Important Sampling (IS) methods approach the optimal sampling density ψ^* using a parameterized auxiliary pdf family (usually the Gaussian). The goal is to determine the parameters $\mathbf{p}$ of the auxiliary pdf $\psi(\cdot, \mathbf{p})$ to minimize the "divergence" between the optimal and the auxiliary pdf. The best known method is the Cross Entropy (CE) method [168]. It is based on an iterative process that aims to minimize the Kullback-Leibler (KL) divergence. The KL divergence between the distributions defined by ψ^* and $\psi(\cdot, \mathbf{p})$ is given by

$$D\left(\psi^* || \psi(\cdot, \mathbf{p})\right) = \int \psi^*(\mathbf{x}) \ln\left(\frac{\psi^*(\mathbf{x})}{\psi(\mathbf{x})}\right) d\mathbf{x}.$$

The CE methods is reduced to find an optimal reference of a parameter vector $\mathbf{p}^*$ by minimizing the previous quantity

$$\mathbf{p}^* \in \operatorname*{argmin}_{\mathbf{p}} D\left(\psi^* || \psi(\cdot, \mathbf{p})\right).$$

Now, it follows that

$$\begin{aligned}
\min_{\mathbf{p}} D\left(\psi^* || \psi(\cdot, \mathbf{p})\right) &= \min_{\mathbf{p}} \int \psi^*(\mathbf{x}) \ln(\psi^*(\mathbf{x})) d\mathbf{x} - \int \psi^*(\mathbf{x}) \ln(\psi(\mathbf{x}, \mathbf{p})) d\mathbf{x} \\
&= \max_{\mathbf{p}} \int \psi^*(\mathbf{x}) \ln(\psi(\mathbf{x}, \mathbf{p})) d\mathbf{x} \\
&= \max_{\mathbf{p}} \int \frac{\phi(\mathbf{x}) I_{\{f(\mathbf{X}) \leq \gamma\}}}{\ell} \ln(\psi(\mathbf{x}, \mathbf{p})) d\mathbf{x} \\
&= \max_{\mathbf{p}} \mathbb{E}_{\phi}[I_{\{f(\mathbf{X}) \leq \gamma\}} \ln(\psi(\mathbf{x}, \mathbf{p}))],
\end{aligned}$$

using again importance sampling, with a change of distribution $\psi(\mathbf{x}, \mathbf{v})$, the following optimization problem is obtained

$$\max_{\mathbf{p}} \mathbb{E}_{\psi(\cdot, \mathbf{v})} \left[I_{\{f(\mathbf{X}) \leq \gamma\}} \ln(\psi(\mathbf{x}, \mathbf{p})) \frac{\phi(\mathbf{X})}{\psi(\mathbf{x}, \mathbf{v})} \right].$$

This problem may be then used to obtain an estimator of $\mathbf{p}^*$ which is

$$\hat{p} \in \operatorname*{argmax}_{\mathbf{p}} \frac{1}{N} \sum_{i=1}^{N} I_{\{f(\mathbf{X}) \leq \gamma\}} \ln(\psi(\mathbf{X}_i, \mathbf{p})) \frac{\phi(\mathbf{X})}{\psi(\mathbf{x}, \mathbf{v})}, \tag{2.5}$$

where $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim \psi(\cdot, \mathbf{v})$. This estimator can be obtained in an explicit form if the family of distributions is the exponential family. Finally, solving Equation (2.5) is a priori difficult, since $I_{\{f(\mathbf{X}) \leq \gamma\}}$ very often takes the value 0 since $f(\mathbf{X}) \leq \gamma$ is a rare event. In this case, a sequence of parameters $\hat{\mathbf{p}}^k$ and levels $\hat{\gamma}^k$ is constructed with the goal that the former converges to $\mathbf{p}^*$ and

the latter converges to γ. To do this, the following iterative process is applied: sample N independent random variables from the density $\psi(\cdot; \hat{\mathbf{v}}^{k-1})$ and let $\hat{\gamma}^k$ be the $(1-\alpha)$-quantile of the objective function values $f(\mathbf{X}_1), \ldots, f(\mathbf{X}_N)$. Then $\hat{\mathbf{v}}^{k-1}$ is updated to $\hat{\mathbf{v}}^k$ by solving the problem in Equation (2.5) with the same N samples.

Non-parametric importance sampling methods [198] have also been proposed using a kernel density estimator. The goal of non-parametric importance sampling is to approximate the optimal density without the need to choose a pdf family. The process in these methods is similar to that of the CE method: at each iteration, a threshold is calculated based on a quantile of the samples generated. Then, a kernel-based sample pdf is created from the samples whose the objective function value is below the threshold. The parameters of the kernel pdf are then updated by the asymptotic mean of the integrated squared error between the sample kernel density and the optimal density. The kernel density function has the advantage of being able to handle multiple optimal densities. In an optimization context, it may be useful to search for several minima at once. However, non-parametric importance sampling is not efficient in dimensions larger than 10.

Direct search includes all methods that, at each iteration, search -within a given set of points- for a point whose objective function value is lower than that of the current solution. Among these methods, there are two well-known heuristics, the Nelder-Mead algorithm [140] and the DIRECT algorithm [86]. The Nelder-Mead algorithm uses the concept of a simplex, which is a polytope of $n+1$ vertices in n dimensions, such that the convex hull has nonempty interior. At each iteration, the algorithm maintains a set of $n+1$ test points arranged as a simplex. Then, according to the objective function values of the simplex points, a strategy is chosen to evaluate a new candidate point. If one of them has a better objective function value than the worst point of the simplex, it is replaced. Otherwise, the simplex is shrinked. The Nelder-Mead algorithm is usually efficient for approximating a local solution of a smooth objective function. In contrast, the DIRECT algorithm is an algorithm that seeks to approximate the global minimum of a function defined on a hyperrectangle of $\mathbb{R}^n$. For this purpose, the DIRECT algorithm works by dividing the hyperrectangle into subrectangles, with the property that the objective function has been evaluated at the center of each rectangle. At each iteration, certain "potentially interesting" rectangles are selected for further search. Their centers are evaluated and the process continues. DIRECT is particularly well suited to finding the global minimum of small-dimensional problems ($n \leq 6$).

All previous methods do not benefit from convergence properties, which is their main drawback. The next subsections are devoted to methods that rely on convergence analysis.

2.2.2 Model-Based Methods

In the DFO framework, even if the derivatives are not available, there are many cases where it is reasonable to assume that the underlying objective function is smooth or locally smooth. In these cases, it may be particularly interesting to locally approximate the function by a model whose gradient is known in order to guide the optimization process. The set of such methods is called model-based methods. This section starts with a short description of the techniques to build a "good" model. Then, two groups of generic methods are described [14]: the model-based descent methods and the model-based trust-region methods.

When constructing a model of a function, the first question that comes to mind is about the quality of that model. The class of fully linear models formalizes this notion of quality.

Definition 2.2.1 (Class of fully linear models)**.** *Given $f \in \mathcal{C}^1$, $\mathbf{X} \in \mathbb{R}^n$ and $\bar{\Delta} > 0$, the class of function $\{\tilde{f}_\Delta\}_{\Delta\in(0,\bar{\Delta}]}$ is said to be a class of fully linear models of f at $\mathbf{x}$ parameterized by Δ if there exists a pair of strictly positive scalar $\kappa_f(\mathbf{x})$ and $\kappa_g(\mathbf{x})$ such that, given any $\Delta \in (0, \bar{\Delta}]$ the model $\tilde{f}_\Delta$ satisfies*

$$\begin{aligned} |f(\mathbf{y}) - \tilde{f}_\Delta(\mathbf{y})| &\leq \kappa_f(\mathbf{x})\Delta^2 && \text{for all } \mathbf{y} \in \mathcal{B}_\Delta(\mathbf{x}); \\ ||\nabla f(\mathbf{y}) - \tilde{g}_\Delta(\mathbf{y})|| &\leq \kappa_g(\mathbf{x})\Delta && \text{for all } \mathbf{y} \in \mathcal{B}_\Delta(\mathbf{x}); \end{aligned}$$

where $\tilde{g}_\Delta = \nabla\tilde{f}_\Delta$.

If a class of fully linear models can be constructed for any $\mathbf{x} \in \mathbb{R}^n$ with κ_f and κ_g independent of $\mathbf{x}$, the models are said to be fully linear. The question now is how to construct a fully linear model. In the DFO framework, the model must be built from function evaluations only. A common idea is to use linear interpolation. The linear interpolation function is defined as follows

$$\tilde{f}_\Delta(\mathbf{x}) = L(\mathbf{x}) = \alpha_0 + \boldsymbol{\alpha}^T\mathbf{x},$$

where $\alpha_0 \in \mathbb{R}$ and $\boldsymbol{\alpha} \in \mathbb{R}^n$ are taken such that $L(\mathbf{y}_i) = f(\mathbf{y}_i),\ \forall i[0, n]$ where the $\mathbf{y}_i \in \mathbb{R}^n$ are a set of $n + 1$ points. In order to have a unique α and $\boldsymbol{\alpha}$ satisfying this condition, the points $\mathbf{y}_i$ must satisfy the following condition:

Definition 2.2.2 (Poised for linear interpolation)**.** *The set $\mathbb{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n\}$ is poised for linear interpolation if the $(n+1)\times(n+1)$ matrix $[\mathbf{1}, \mathbf{Y}^T]$ is of full rank, with $\mathbf{Y} = [\mathbf{y}_0, \ldots, \mathbf{y}_n]$ and $\mathbf{1} \in \mathbb{R}^{n+1}$ is a vector of ones.*

Finally, if this condition is met, the linear interpolation model is a fully linear model [14, Theorem 9.5].

Theorem 2.2.3 (Linear approximation is fully linear). *Let* $\mathbf{x} \in \mathbb{R}^n$ *and* $f \in \mathcal{C}^{1+}$ *on* $\mathcal{B}_{\hat{\Delta}}(\mathbf{x})$ *with constant* K. *Let* $\mathbb{Y}$ *be poised for linear interpolation with* $\mathbf{y}_0 = \mathbf{x}$ *and* $\Delta \leq \hat{\Delta}$, *then*

$$|f(\mathbf{y}) - L(\mathbf{y})| \leq \left(\frac{1}{2}K(1+\sqrt{n}C)\right)\Delta^2 \qquad \text{for all } \mathbf{y} \in \mathcal{B}_{\Delta}(\mathbf{x}),$$
$$||\nabla f(\mathbf{y}) - \nabla L(\mathbf{y})|| \leq \left(\frac{1}{2}K(2+\sqrt{n}C)\right)\Delta \qquad \text{for all } \mathbf{y} \in \mathcal{B}_{\Delta}(\mathbf{x}),$$

with C *a positive constant depending on the set* $\mathbb{Y}$.

There are other techniques to build a model of the function, such as incorporating order-2 information of the function or using more than $n+1$ points to build the model. Since this is not the focus of this research, readers may consult [14, Chapter 9] and references therein.

Model-based descent is a first group of methods to use model in a DFO algorithm. These methods rely on the concept of line-search to improve the current solution. Line-search is a popular technique in optimization. It consists of finding a descent direction and then searching for a solution along that descent direction. The interest is to reduce the minimization problem to a one-dimensional minimization problem at each iteration. To be efficient, the line search must satisfy a certain condition called the Armijo condition [6]. A pseudocode of a model-based descent algorithm is given in Algorithm 4. The convergence result of the model-based descent algorithm is given in the following theorem [14, Theorem 10.6].

Theorem 2.2.4 (Convergence of the model-based descent algorithm). *Suppose* $f \in \mathcal{C}^1$ *is bounded below and* $\tilde{g}_{\Delta}$ *is the gradient of a fully linear model. Suppose that there exists* $\bar{t} > 0$ *such that* $t^k||\mathbf{d}^k|| \geq \bar{t}$ *for all* k. *If Algorithm 4 is run with* $\epsilon = 0$, *then*

$$\lim_{k\to\infty} ||\nabla f(\mathbf{x})|| = 0.$$

While the model-based descent algorithm uses only the gradient of the model to guide the optimization process, the model-based trust region aims to fully exploit the information given by the model. To do so, it uses not only the gradient, but also the approximations to the function values given by the model. Given a fully linear model, at each iteration the model-based trust region aims to minimize the model $\tilde{f}_{\Delta}^k$ within a ball of radius Δ^k. This ball is called the trust region and corresponds to a region within the model that should be an accurate approximation of the true function.

Algorithm 4 Model-based Descent algorithm

Inputs:
$\Delta^0 \in (0, \infty)$ the initial model accuracy parameter and μ^0 the initial target accuracy parameter
$\eta \in (0, 1)$ an Armijo parameter
$\epsilon \in [0, \infty)$
$\tilde{g}^0$ initial gradient approximation of ∇f
while $\Delta^k < \epsilon$ and $||\tilde{g}^k|| < \epsilon$ **do**
    Model:
    Use Δ^k and finite number of point to build a fully linear model and set $\tilde{g}^k$ as the gradient of the model
    Check model accuracy:
    if $\Delta^k > \mu^k ||\tilde{g}^k||$ **then**
        Declare the model inaccurate : $\Delta^{k+1} = \frac{1}{2}\Delta^k$, $\mu^{k+1} = \mu^k$, $k \leftarrow k + 1$ and go to **Model** step
    else
        Declare the model accurate and go to **Line-search** step
    end if
    Line search:
    Select $\mathbf{d}^k \in \mathbb{R}^n$ such that $\left(\frac{\mathbf{d}^k}{||\mathbf{d}^k||}\right)^T \left(\frac{\tilde{\mathbf{g}}^k}{||\tilde{g}^k||}\right) < \epsilon_d$ with $\epsilon_d \in (0, 1)$ a parameter controlling the descent angle. Then, perform a line-search in the direction $\mathbf{d}^k$ to seek t^k such that

$$f(\mathbf{x}^k + t^k \mathbf{d}^k) < f(\mathbf{x}^k) + \eta t^k (\mathbf{d}^k)^T \tilde{\mathbf{g}}^k \text{ (Armijo)}$$

    Update:
    if Line-search is a success **then**
        Set $\mathbf{x}^{k+1}$ be any point such that $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k + t^k \mathbf{d}^k)$, $\Delta^{k+1} = \Delta^k$ and $\mu^{k+1} = \mu^k$
    else
        set $\mathbf{x}^{k+1} = \mathbf{x}^k$, $\Delta^{k+1} = \Delta^k$ and $\mu^{k+1} = \frac{1}{2}\mu^k$
    end if
    $k \leftarrow k + 1$
end while

The solution obtained by solving the approximation subproblem is referred to as the candidate solution. The quality of the candidate solution is checked and either the new point is accepted and the radius of the trust region is increased, or the point is rejected and the radius of the trust region is decreased. A pseudocode of the model-based trust region algorithm is given in Algorithm 5. The convergence properties of this algorithm are detailed in the following theorem.

Theorem 2.2.5 (Convergence of the model based trust region algorithm)**.** *Suppose $f \in \mathcal{C}^1$ is bounded below, and $\tilde{f}_\Delta$ are quadratic fully linear model. With additional technical assumptions on the model and on the iterates produced by Algorithm 5, if the algorithm runs with $\epsilon = 0$, then*

$$\lim_{k \to \infty} \inf ||\nabla f(\mathbf{x}^k)|| = 0.$$

The methods described so far are not adapted to handle derivative-free constrained optimization problems. However, these methods have been adapted for constrained problems. The best known

Algorithm 5 Model-based trust region algorithm

Inputs:
$\Delta^0 \in (0, \infty)$ the initial model trust region radius and μ the model accuracy parameter
$\eta \in (0, 1)$ sufficient decrease test parameter and $\gamma \in (0, 1)$ trust region update parameter.
$\epsilon \in [0, \infty)$
$\tilde{f}^0$ initial model
while $\Delta^k < \epsilon$ and $||\nabla \tilde{f}^k(\mathbf{x}^k)|| < \epsilon$ **do**
    Model:
    if $\Delta^k > \mu^k ||\tilde{g}^k||$ or $\tilde{f}^k$ is not a fully linear model in $\mathcal{B}_{\Delta^k}(\mathbf{x}^k)$ **then**
    end if
    Decrease $\Delta^{k+1} = \gamma \Delta^k$, $\mu^{k+1} = \mu^k$ and create a fully linear model $\tilde{f}^k$, increment $k \leftarrow k + 1$
    Trust region subproblem
    Solve or approximate $\mathbf{x}' \in \operatorname{argmin}_{\mathbf{x}} \{\tilde{f}^k(\mathbf{x}) \mid \mathbf{x} \in \mathcal{B}_{\Delta^k}(\mathbf{x}^k)\}$
    Check quality of $\mathbf{x}'$:
    Compute the ratio

$$\rho^k = \frac{f(\mathbf{x}^k) - f(\mathbf{x}')}{\tilde{f}^k(\mathbf{x}^k) - \tilde{f}^k(\mathbf{x}')}$$

    Update:
    if $\rho^k > \eta$ (sufficient decrease) **then**
        Set $\mathbf{x}^{k+1}$ be any point such that $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}')$, increase $\Delta^{k+1} = \gamma^{-1} \Delta^k$ and build $\tilde{f}^{k+1}$
        using $\tilde{f}^k$ and $\mathbf{x}^{k+1}$
    else
        set $\mathbf{x}^{k+1} = \mathbf{x}^k$, $\Delta^{k+1} = \gamma \Delta^k$ and keep $\tilde{f}^{k+1} = \tilde{f}^k$
    end if
    $k \leftarrow k + 1$
end while

confidence region algorithms developed for this purpose are Powell's algorithms BOBYQA [151] for bounded constrained problems, LINCOA [150] for linear constrained problems, and COBYLA [149] for nonlinear constrained problems. In the COBYLA algorithm, the constraint and objective functions are approximated by linear interpolation models, and then a trust-region algorithm is used to solve the problem. More recently, a trust-region method using fully linear models of both the constraint and objective functions has been developed in [24]. It introduces a method to locally convexify the blackbox constraints. The trust region subproblem minimizes the model of the objective function as the intersection of the trust region and a restricted feasible set. The resulting algorithm is called NOWPAC. Other methods based on penalization have also been developed. In [120], the problem is transformed into an unconstrained problem thanks to an exact penalty function. Sometimes, the Lagrangian relaxation is used in addition with the construction of models [77]. A line search algorithm is then employed to solve a smoothed approximation of the unconstrained problem.

2.2.3 Direct Search Methods

This section focuses on local optimization utilizing direct search methods. Hooke and Jeeves [85] describe the direct search methods as:

> *The sequential examination of candidates by comparison with the best solution obtained until now and the introduction of a strategy in order to pick the next best promising candidates.*

Coordinate search (CS) algorithm. The first direct search algorithm is the CS algorithm [67]. Its main idea is to evaluate the objective function in a collection of points, looking for an improvement over the current solution. In this method, the points used are those along the coordinate directions and within a δ^k step-length of the current solution. If an improved solution is found, the current solution is updated, otherwise the step length is reduced. The main drawback of the coordinate search algorithm is that it cannot handle simple problems such as minimizing $f : \mathbb{R}^2 \to \mathbb{R}$ [14, Example 3.3] where $f(\mathbf{x}) = ||\mathbf{x}||_\infty$. Indeed, if the algorithm starts at $x_0 = (1, 1)$, it will always stay at x^0 until the step-length equals 0, whereas the optimal solution lies at $x^* = (0, 0)$. To solve this problem, the CS algorithm has been generalized and made more flexible.

Generalized Pattern Search (GPS) algorithm. In 1997, Torczon [184] proposed a generalization of various pattern search methods, grouping them into a common structure and giving a proof of convergence for this class of algorithms. This new structure is called the GPS algorithm. The GPS algorithm increases flexibility in the way points are selected. Coordinate directions are replaced by a more general set of directions: the polling directions. In the GPS algorithm, the set of polling directions can change from one iteration to the next. In addition, under certain conditions, it is possible to evaluate points outside the set of polling directions. All this new flexibility is made possible by the use of the mesh, which is an enumerable discretization of the space of variables.

Definition 2.2.6 (Mesh)**.** *Let $G \in \mathbb{R}^{n\times n}$ be a full-rank matrix and $Z \in \mathbb{Z}^{n\times p}$ be such that its columns form a positive spanning set, i.e. a set of nonnegative linear combination of vectors and spanning $\mathbb{R}^n$. Then denoting $D = GZ$, the mesh at a point $\mathbf{x}^k \in \mathbb{R}^n$ of coarseness $\delta^k > 0$ is defined by*

$$M^k := \{\mathbf{x}^k + \delta^k D\mathbf{y} \mid \mathbf{y} \in \mathbb{N}^p\} \subset \mathbb{R}^n.$$

Each iteration of the GPS algorithm is divided into two main steps. First, the search step, where any strategy can be used to obtain an improvement over the current solution, provided that only a finite number of points are considered and that these points are projected onto the mesh before

being evaluated. If the search step does not yield an improvement, the poll step is performed. This step simply consists of evaluating the neighboring mesh points of the current solution $\mathbf{x}^k$. If either the search or the poll step leads to a successful iteration, the mesh size parameter δ^k is doubled, otherwise it is halved. The poll step allows to ensure the convergence of the algorithm. The main convergence result is that if the objective function f is locally Lipschitz, then the ieterates of the GPS algorithm converge to a limit point where the generalized directional derivatives are non-negative for a finite set of directions. Nevertheless, the GPS can only handle unconstrained optimization problems.

Mesh Adaptive Direct Search (MADS). The MADS [12] algorithm was developed to overcome the two drawbacks of the GPS algorithm : its proof of convergence (limited to a finite set of directions) and its inability to handle constrained optimization problems. To solve the first problem, another structure of space is defined in addition to the mesh: the frame.

Definition 2.2.7. *Let $G \in \mathbb{R}^{n\times n}$ be a full-rank matrix and $Z \in \mathbb{Z}^{n\times p}$ be such that its columns form a positive spanning set for $\mathbb{R}^n$. Set $D = GZ$ and select a mesh size parameter $\delta^k > 0$ and let the frame size parameter Δ^k be such that $\delta^k \leq \Delta^k$. The frame of extend Δ^k and generated by D at a point $\mathbf{x}^k \in \mathbb{R}^n$ is defined by*

$$F^k := \{\mathbf{x} \in \mathbf{M}^k \mid ||\mathbf{x} - \mathbf{x}^k||_\infty \leq \Delta^k b\},$$

with $b = \max\{||\mathbf{d}'||_\infty \mid \mathbf{d}' \in D\}$.

In the MADS algorithm, the poll set is constructed by selecting the poll directions from within the frame F^k. If the mesh size parameter decreases faster than the frame size parameter, this allows to obtain asymptotically dense selections of poll directions. To address the second issue, the MADS algorithm can use two different strategies. The simplest is the extreme barrier strategy [12]. This strategy transforms the constrained problem into an unconstrained one by using the following objective function:

$$f_\Omega(\mathbf{x}) := \begin{cases} f(\mathbf{x}) & \text{if } x \in \Omega, \\ +\infty & \text{otherwise.} \end{cases}$$

A finer way to handle the constraints is the progressive barrier [11]. This strategy allows searching for a feasible point while taking into account the value of the objective function outside of Ω. It allows MADS to be run from an infeasible point. The progressive barrier is based on the constraint

violation measure [69] defined by

$$h(\mathbf{x}) = \sum_{j=1}^{m} (\max\{c_j(\mathbf{x}), 0\})^2.$$

The value of the constraint violation function $h(\mathbf{x})$ is 0 if and only if the point $\mathbf{x}$ belongs to Ω, otherwise it is strictly positive. This function allows to rank any pair of test points using the following dominance relation [14].

Definition 2.2.8. *The feasible point $\mathbf{x} \in \Omega$ is said to dominate $\mathbf{y} \in \Omega$ when $f(\mathbf{x}) < f(\mathbf{y})$. The infeasible point $\mathbf{x} \in \mathcal{X} \setminus \Omega$ is said to dominate $\mathbf{y} \in \mathcal{X} \setminus \Omega$ when $f(\mathbf{x}) \leq f(\mathbf{y})$ and $h(\mathbf{x}) \leq h(\mathbf{y})$ with at least one strict inequality.*

The progressive barrier method approaches an optimal solution by exploring around two current solutions. The feasible solution $\mathbf{x}^{feas} \in \Omega$ and the infeasible solution $\mathbf{x}^{inf}$, which is an undominated infeasible point with a value of h lower than a threshold called $h_{\max}$. Then the poll step is applied around these two incumbent solutions. An iteration of the MADS algorithm with the progressive barrier is

- dominating, when a dominating trial point with respect to $\mathbf{x}^{inf}$ or $\mathbf{x}^{feas}$ is found. In this case, the threshold is updated to $h_{\max}^{k+1} = h_I^k$.
- improving, when it is not a dominating iteration, but a trial point improves the threshold $h_{\max}^k$. In this case, threshold is updated to $h_{\max}^{k+1} = \max\{h(\mathbf{v}) : h(\mathbf{v}) < h_I^k, \mathbf{v} \in V^{k+1}\}$.
- or unsuccessful, when it is neither a dominating nor an improving iteration. In this case, the threshold is updated to $h_{\max}^{k+1} = h_I^k$.

where h_I^k is defined by

$$h_I^k = \begin{cases} h(\mathbf{x}) & \text{for any } \mathbf{x} \in I^k = \{\underset{\mathbf{x} \in U^k}{\operatorname{argmin}}\{f(\mathbf{x}) : 0 < h(\mathbf{x}) \leq h_{\max}^k\}\}, \\ \infty & \text{if } I^k = \emptyset, \end{cases}$$

with U^k the set of infeasible undominated points. Algorithm 6 provides a description of the MADS with progressive barrier algorithm The convergence of the MADS algorithm with the progressive barrier rely on the concept of refining subsequences, points and directions.

Algorithm 6 The Mesh Adaptive Direct Search algorithm

0. Initialization:
   - A set of starting point: $V^0 \subset \mathbb{R}^n$
   - An initial poll size vector: $\mathbf{\Delta}^0$
   - The iteration counter: $k \leftarrow 0$
   - The mesh size adjustment parameter: $\tau \in \mathbb{Q} \cap (0,1)$
   - The initial threshold: $h^0_{\max} = \infty$
   - Define $\boldsymbol{\delta_j^0} = \min\{{\mathbf{\Delta}_j}^0, ({\mathbf{\Delta}_j}^0)^2\},\ \forall j \in [1,n]$
1. Search step (optional):
   - Launch the simulation on a finite set S^k of mesh points.
   - If successful, go to 3.
2. Poll step:
   - Launch the simulation on the set P^k of poll points.
3. Updates:
   - Update the cache V^{k+1}.
   - If the iteration is dominating :
     - update $\mathbf{x}^{k+1}$, $h^{k+1}_{\max} = h^k_I$ and $\mathbf{\Delta}^{k+1} = \tau^{-1}\mathbf{\Delta}^k$.
   - Else, if the iteration is improving:
     - update $\mathbf{x}^{k+1}$,$h^{k+1}_{\max} = \max\{h(\mathbf{v}) : h(\mathbf{v}) < h^k_I, \mathbf{v} \in V^{k+1}\}$ and $\mathbf{\Delta}^{k+1} = \mathbf{\Delta}^k$.
   - Otherwise:
     - Set $\mathbf{\Delta}^{k+1} = \tau\mathbf{\Delta}^k$ and $h^{k+1}_{\max} = h^k_I$.
   - Set $\boldsymbol{\delta_j^{k+1}} = \min\{{\mathbf{\Delta}_j}^{k+1}, ({\mathbf{\Delta}_j}^{k+1})^2\},\ \forall j \in [1,n]$.
   - Increase the iteration counter $k \leftarrow k+1$ and go to 1.

Definition 2.2.9. *A convergent subsequence of mesh local minimum $\{\mathbf{x}^k\}_{k\in K}$ (for some subset of indices K) is said to be a refining subsequence if and if $\lim_{k\in K} \delta^k = 0$. The limit $\hat{\mathbf{x}}$ is called a refined point. A direction $\mathbf{d}$ is a refining direction if and only if there exists an infinite subset $L \subset K$ with poll directions $\mathbf{d}^k$ such that $\mathbf{x}^k + \delta^k\mathbf{d}^k \in \Omega$ and $\lim_{k\in L} \frac{\mathbf{d}^k}{||\mathbf{d}^k||} = \frac{\mathbf{d}}{||\mathbf{d}||}$.*

Now the main result may be stated.

Theorem 2.2.10. *Let f be locally Lipschitz continuous near a refined point $\hat{\mathbf{x}} \in \Omega$ and $\mathbf{d} \in T_\Omega(\hat{\mathbf{x}})$ be a refining direction. Then,*

$$f^\circ(\hat{\mathbf{x}}; \mathbf{d}) \geq 0.$$

Like the GPS algorithm, the great flexibility of the MADS algorithm comes from the search step. This step allows embedding any heuristics or others algorithms aiming to accelerate the convergence to a minimum or better explore the space of variables. For instance, a variable neighborhood search step has been implemented [10], in addition with a Nelder-Mead search step [20] and the use of models [53].

2.2.4 Summary

In this first section of the literature review, different methods for deterministic blackbox optimization problems have been described. The methods are grouped into three different classes. Table 2.1 summarizes the main advantages and drawbacks of the different methods according to five perspectives: the assumptions made about the regularity of the function, the existence of a proof of convergence, the range of problem sizes handled, the initial design for handling constraints and the type of minima sought.

Table 2.1 Drawbacks and advantages of the main classes of algorithms for deterministic blackbox optimization.

Class	Assumption	Convergence proof	Range of n	Constraints	Type of search
Metaheuristics	✗	✗	$[2, 1000]$	✗	global
Model-based methods	$\mathcal{C}^1$	✓	$[2, 50]$	✗	local
Direct search methods	✗	✓	$[2, 50]$	✓	local/global

2.3 Uncertain blackbox optimization

In this section, the work specific to uncertain blackbox optimization is detailed. First, the definition, classification and mathematical model of uncertainties are introduced in Section 2.3.1. Then, different problem formulations depending on the chosen model are detailed. Finally, methods from two formulations are depicted more precisely. First, algorithms for unconstrained blackbox stochastic optimization problems are presented in Section 2.3.2 . Second, algorithms for stochastic constrained blackbox optimization problems (1.4) are presented in Section 2.3.3.

2.3.1 Uncertainty modeling

Many optimization solvers are able to handle uncertainties. In practical applications, the sources of uncertainty are numerous, e.g., due to variations in parameters, operating conditions, and modeling and simulation [27]. Following this assessment, the next step is to classify the uncertainties into categories. Depending on the field of application and the granularity of the studied phenomena, there are different ways to classify the uncertainty. However, a consensus has been reached on two main categories [183]: aleatory and epistemic uncertainty.

Aleatory uncertainty is due to the inherent variability of a physical system and/or its environment. It

cannot be reduced by collecting more information or data. It is also known as stochastic uncertainty or irreducible uncertainty. Classical examples of aleatory uncertainty are: pressure, temperature, or wind direction. Epistemic uncertainty arises from any lack of knowledge, simplifications about the phenomena being modeled, or measurement inaccuracies. It can be reduced by collecting more information or paying a higher cost. It is also referred to as reducible uncertainty. A typical example is the cutting of a sheet of metal. The accuracy of a hydraulic press cut will be on the order of a tenth of a millimeter, while the accuracy of a more expensive laser cut may be on the order of a micrometer.

Finally, both categories of uncertainty must be modeled with a mathematical formalism in order to be handled by an optimization solver. Due to their different natures, the two types of uncertainty cannot be modeled in the same way. In fact, improper modeling of uncertainty can lead to underperforming designs or even significant design failures [157]. While random uncertainties are commonly modeled using probability theory, the choice of which model to use for epistemic uncertainty is particularly complex. When there are not enough data to construct a precise statistical distribution, it is necessary to resort to others techniques. More appropriate formalisms have been developed, such as evidence theory [172], interval analysis [136], or fuzzy sets [196]. Although interesting, these methods require the definition of a new arithmetic based on interval or fuzzy sets. This is beyond the scope of this work. Two methods based on probability theory have also emerged to deal with epistemic uncertainty. First, the Bayesian theory [30] allows to update the data distribution thanks to new samples obtained from new experiments. Second, the distributionally robust optimization [115, 154] where the uncertainties are modeled thanks to a family of distributions, thus it does not require precise information about the distribution of the uncertainties.

The formulation of Problem (1.4) depends on the model of uncertainty and the measure Ξ used to deal with it. In the following, the main formulations are described according to the degree of knowledge we have about the uncertainties. First, the *robust optimization* formulation [28] is a "worst-case" approach, requiring only the support of the uncertainties.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \sup_{\boldsymbol{\xi}\in\mathcal{U}}[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \sup_{\boldsymbol{\xi}\in\mathcal{U}}[G_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in \{1,2,\ldots,m\}. \end{array} \tag{2.6}$$

This formulation is well suited for cases where little information about uncertainties is available. However, this approach is often overly conservative, and the feasible set may be tiny or even non-existent. Moreover, if the structure of the problem is known, as in the case of a linear or conic problem, it is possible to transform the problem into a tractable one, i.e., in a deterministic problem with a known structure [28]. In a DFO or BBO context, however, it is generally impossible to achieve the same transformation. In the absence of constraints, work has nevertheless been

done [32, 131].

Second, the *distributionally robust optimization* formulation [58, 154] needs only partial distributional information.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \sup_{\phi\in\Phi} \rho_\phi[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \sup_{\phi\in\Phi} \rho_\phi[G_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in \{1,2,\ldots,m\}, \end{array} \tag{2.7}$$

where Φ denotes the ambiguity set of probability measures, i.e., a family of measures that are consistent with prior knowledge about the uncertainty. The ρ_ϕ function is used to quantify the uncertainty in the outcome of the objective and constraint functions, for a given fixed probability measure ϕ, e.g. $\rho_\phi = \mathbb{E}_\phi[\cdot]$ (others examples are given below). This approach is very attractive because it allows to use all available knowledge about the uncertainties. Moreover, the ambiguity set Φ remains very general and may, in practice, describe a large class of family distributions. However, these problems are generally infinite dimensional, since the supremum is taken in a set of probability measures. Thus, without assumptions about the structure of the functions F and G_j, the problem is often intractable in practice. Therefore, to the best of our knowledge, there is no work that uses distributionally robust optimization in a derivative-free setting.

Finally, the *stochastic optimization* formulation needs complete distributional information [173]

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \rho_{\boldsymbol{\xi}}[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \rho_{\boldsymbol{\xi}}[G_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in \{1,2,\ldots,m\}, \end{array} \tag{2.8}$$

where $\rho_{\boldsymbol{\xi}}$ is used to quantify the uncertainty in the outcome of the objective and constraint functions for the distribution of $\boldsymbol{\xi}$. Note that in practice this distribution does not need to be known to solve Problem (2.8), except to compute the analytical formulation of the problem. In DFO, the analytical expression of the objective and constraint functions are unknown, so this would have been impossible anyway. However, it is clear that this formulation does not seem to be suitable for dealing with epistemic uncertainty. Nevertheless, given the difficulty of using the previous formulations, it is the one most often used in practice.

Depending on the choice of $\rho_{\boldsymbol{\xi}}$, different formulations are derived from the stochastic formulation given in Equation (2.8). Using $\rho = \mathbb{E}_{\boldsymbol{\xi}}[\cdot]$, the following *risk neutral* [173] formulation is obtained

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \mathbb{E}_{\boldsymbol{\xi}}[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \mathbb{E}_{\boldsymbol{\xi}}[G_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in \{1,2,\ldots,m\}. \end{array} \tag{2.9}$$

This formulation is limited because the feasible set is given in mean, i.e. it does not take into account the deviation. When the deviation is large, satisfying the constraints in mean gives little

indication of the quality of the solution obtained. Nevertheless, this formulation was used in [65]. However, in the absence of constraints, this formulation is widely spread and is the subject of Section 2.3.2. A more interesting formulation is

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \mathbb{E}_{\boldsymbol{\xi}}[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \mathbb{P}_{\boldsymbol{\xi}}(G_j(\mathbf{x},\boldsymbol{\xi})\leq 0)\geq \alpha_j,\ \forall j\in\{1,2,\ldots,m\}, \end{array} \tag{2.10}$$

where α_j are the reliability indices. The advantage of this formulation is that it tends to satisfy the constraint with some probability. However, the computational cost of this probability estimation can be particularly high, especially in the context of DFO. A special survey of methods that attempt to solve this type of formulation is given in Section 2.3.3. Finally, a promising formulation is based on the Conditional Value-at-Risk (CVaR) risk measure [162]. The CVaR risk measure is a probability measure based on the quantile of the level $\alpha\in(0,1)$ of a random variable and can be defined for a random function $F(\mathbf{x},\boldsymbol{\xi})$ as follows

$$\mathrm{CVaR}_\alpha(F(\mathbf{x},\boldsymbol{\xi}))=\min_{t\in\mathbb{R}}\mathbb{E}\left[t+\frac{1}{1-\alpha}\max(0,F(\mathbf{x},\boldsymbol{\xi})-t)\right].$$

The formulation of the stochastic optimization Problem (2.8) with CVaR is

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \mathrm{CVaR}_{\alpha_0}[F(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \mathrm{CVaR}_{\alpha_j}[G_j(\mathbf{x},\boldsymbol{\xi})]\leq 0,\ \forall j\in\{1,2,\ldots,m\}, \end{array} \tag{2.11}$$

where $(\alpha_0,\ldots,\alpha_m)$ are the various desired reliability indices. This formulation is a conservative approximation of the formulation given in Equation (2.10) [173, Chapter 6]. Thanks to α_j, the optimizer benefits from the ability to choose the desired degree of reliability. Indeed, choosing α close to 1 tends to adopt a "worst-case" approach, as in Problem (2.6), while choosing α close to 0 tends to adopt a "risk-neutral" approach, as in Problem (2.8). Thus, this probabilistic approach can deal with epistemic uncertainty. Moreover, it has been shown to be closely related to distributionally robust optimization ([154]). This formulation has been used in the reinforcement learning community [50, 181], with model-based methods [130], and in engineering [160]. Chapter 6 is dedicated to efficiently solve problem Equation (2.11) in a BBO context.

2.3.2 The unconstrained case

This section is dedicated to methods aimed at solving the unconstrained version of Problem (2.8), i.e. solving the following problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[F(\mathbf{x},\boldsymbol{\xi})], \tag{2.12}$$

where the distribution of $\boldsymbol{\xi}$ may be unknown. This means that the methods discussed are based only on samples of F. In recent years, several algorithms have been developed to solve this problem (2.12). Most of the algorithms are adaptations of deterministic blackbox algorithms. However, there is also a large class of algorithms based on zero-order gradient approximation.

The analysis of the model-based methods in Section 2.2.2 depends on the construction of fully linear of fully quadratic models of a deterministic function f, see Definition 2.2.1. Thus, a natural extension of model-based algorithms is to build a model thanks to the realization value of the stochastic function. If the model is fully linear, then the adapted algorithm should be able to solve the problem (2.12). However, the model is stochastic since it depends on $\boldsymbol{\xi}$. This change of paradigm implies to modify Definition 2.2.1 to adapt it to the stochastic model. In [26], the definition of probabilistically fully linear models is introduced, which essentially says that the condition in Definition 2.2.1 needs to be satisfied at a given iteration only with some probability. Based on this definition, [104] and [45] developed a model-based confidence region algorithm. It was proved that under the smoothness assumption of the objective function, the algorithm converges to a stationary point with probability one [45]. Later, [34] derives a worst-case complexity (WCC) rate of convergence of this algorithm in terms of the expected number of iterations. Using a similar analysis, [145] derives a WCC rate of convergence for the stochastic model-based descent algorithm.

As with model-based methods, handling stochasticity with a direct search algorithm requires fundamental changes to the core of the algorithm. These changes are necessary because, due to the stochasticity of the function, it is not sufficient to compare objective function values at the sampling stage. The idea developed in [23] is to run the MADS algorithm on estimates of the stochastic function value. These estimates must be sufficiently accurate with a sufficiently high probability to ensure convergence. With an additional assumption on the smoothness of the function, a WCC convergence rate to a stationary point for a large class of direct search algorithms was then given in [64].

This paragraph depicts the algorithms based on zeroth-order gradient approximation. These methods rely on the same type of algorithms as deterministic or stochastic gradient descent, but use function evaluations to obtain an estimate of the gradient. The methods then differ in the way the gradient estimate is computed and in the way the estimate is used. A pseudocode of a gradient descent algorithm based on ZO gradient approximation is given in Algorithm 7.

Algorithm 7 Gradient descent based on ZO approximation of the gradient

1: Choose starting point $\mathbf{x}^0$, sequence of step-sizes $(\alpha^k)_{k\in\mathbb{N}}$ and sequence of difference parameters $(\beta^k)_{k\in\mathbb{N}}$
2: **for** $k = 1, 2, \ldots$ **do**
3: Compute a ZO-gradient estimator $\tilde{\mathbf{g}}^k(\mathbf{x}^k, \beta^k, \boldsymbol{\xi}^k)$
4: Update:

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \alpha^k \tilde{\mathbf{g}}^k(\mathbf{x}^k, \beta^k, \boldsymbol{\xi}^k)$$

5: **end for**

The first work on this type of algorithm is the one in [92], which is based on the work on stochastic approximation made in [156]. In this work, the gradient is estimated by the central finite difference of stochastic realizations of the function F, that is

$$\tilde{g}_j^k(\mathbf{x}^k, \beta^k, \boldsymbol{\xi}^k) = \frac{F(\mathbf{x}^k + \beta^k \mathbf{e}_j, \boldsymbol{\xi}_j^{k,+}) - F(\mathbf{x}^k - \beta^k \mathbf{e}_j, \boldsymbol{\xi}_j^{k,-})}{2\beta^k}, \ \forall j \in [1, n].$$

At each iteration, estimating the gradient requires $2n$ function evaluations. Even with forward finite difference the algorithms still need to evaluate the function $n + 1$ times. Solutions to evaluate the function only 2 times per iteration, regardless of the dimension of the problem, were discovered years later. First, in [166], where the original stochastic function is perturbed in all directions simultaneously. The original idea is to approximate the gradient of the expected performance by its convolution with a multivariate Gaussian distribution. The resulting function is thus smoothed by the convolution. The gradient approximation of the approximate smoothed functions is computed as follows,

$$\tilde{\mathbf{g}}^k(\mathbf{x}^k, \beta^k, \boldsymbol{\xi}^k) = \frac{F(\mathbf{x}^k + \beta \mathbf{u}^k, \boldsymbol{\xi}_+^k) - F(\mathbf{x}^k - \beta \mathbf{u}^k, \boldsymbol{\xi}_-^k)}{2\beta} \mathbf{u}^k, \tag{2.13}$$

where $\mathbf{u}^k$ is sampled from a multivariate Gaussian distribution. Another approximation was developed in [175] with the same idea of perturbing the function simultaneously along its directions. The gradient estimator is computed as follows

$$\tilde{\mathbf{g}}_i^k(\mathbf{x}^k, \beta^k, \boldsymbol{\xi}^k) = \frac{F(\mathbf{x}^k + \beta^k \Delta^k, \boldsymbol{\xi}_+^k) - F(\mathbf{x}^k - \beta^k \Delta^k, \boldsymbol{\xi}_-^k)}{2\beta^k \Delta_i^k}, \ \forall i \in [1, n],$$

where $\Delta^k = (\Delta_1^k, \ldots, \Delta_n^k)$ and the Δ_i^k for $i \in [1, n]$ are commonly independent symmetric Bernoulli-distributed random variables. Under the assumption of smoothness and technical requirements on the step sizes, these methods have all been shown to converge to a neighborhood of a stationary point of the problem (2.12) with probability one (see [33] for the various convergence proofs). These methods have the ability to handle blackboxes where the uncertainties are embed-

ded in the blackbox process. The work of [166] is the subject of renewed interest in the context where the methods have control over the selection of the random variables $\boldsymbol{\xi}$. In this case, with the assumption of smoothness of the function F for any $\boldsymbol{\xi}$, a WCC in mean was found in [71], i.e.

$$\mathbb{E}[||\nabla f(\mathbf{x}^R)||_2] \leq O\left(\frac{\sqrt{\sigma} n^{\frac{1}{4}}}{K^{\frac{1}{4}}}\right),$$

where R is randomly uniformly picked in $[1, K]$ and with the additional assumption that $\mathbb{E}[||\nabla f(\mathbf{x}^k) - \tilde{g}(\mathbf{x}^k, \boldsymbol{\xi}^k)||] \leq \sigma^2$. Other work has followed - with different updates of Algorithm 7 - using projection [141], ZO sign-gradient descent [116], momentum update [47], or conditional gradient [25]. Independently of these developments, similar techniques have also emerged in the convex case, the so-called bandit feedback methods. One-point bandit methods refer to those methods that do not have control over the uncertainty vectors. These methods are based on an approximation of the same type as in Equation (2.13), but with $\mathbf{u}^k$ uniformly chosen on the random sphere [68] or on mirror descent [70]. Multipoint bandit methods are methods that have control over the uncertainty vector $\boldsymbol{\xi}$. They are often based on mirror descent algorithms as well [63].

2.3.3 The constrained case

This section focuses on algorithms for solving the problem (2.10). The main difficulty in this problem is to efficiently estimate the probabilistic constraints. Conventional algorithms use two nested loops. In the inner loop, the probabilistic constraints are estimated by a method called the *reliability assessment method*. In the outer loop, the previously computed estimator is used to replace the probabilistic constraints and the problem (2.10) is solved. Here are some reliability assessment methods for aleatory, epistemic, or mixed aleatory/epistemic uncertainty.

First-order reliability methods (FORM) [51] aim to find the point on the boundary that is closest to the solution. This point is called the "most probable point" of failure. To do this, they make three assumptions. First, they assume that the underlying distribution of uncertainties is fully known. Then, an isoprobabilistic transformation (Rosenblatt [165] or Nataf [108]) is applied to transform the uncertainties $\boldsymbol{\xi}$ into random Gaussian vectors. Second, the objective and constraint are assumed to be sufficiently smooth to perform a first-order approximation of the functions. Finally, either the objective function is assumed to be deterministic or the uncertainties are assumed to be only in the design variable, in which case it follows that

$$\mathbb{E}[F(\mathbf{x}, \boldsymbol{\xi})] = \mathbb{E}[f(\mathbf{x} + \boldsymbol{\xi})] \approx f(\mathbf{x}) + \mathbb{E}[\nabla f(\mathbf{x})^T \boldsymbol{\xi}] = f(\mathbf{x}), \tag{2.14}$$

with $\mathbb{E}[\mathbf{x} + \boldsymbol{\xi}] = \mathbf{x}$. Once the transformation and first-order approximation are done, the probabilis-

tic constraint in Problem (2.10) can be estimated by solving a deterministic optimization problem. Since the objective function is assumed to be deterministic, solving Problem (2.10) involves solving two nested deterministic optimization problems. Based on this concept, several approaches have been developed: the double loop (e.g., performance measure approach or reliability index approach [5]), the single loop (e.g., single loop approach [114]), or the decoupled approach (e.g., sequential optimization and reliability evaluation [61] or sequential approximate programming [49]). These methods are particularly efficient in terms of function evaluation because the problem is deterministic and classical optimization algorithms can be applied. However, the assumptions required to apply these methods are often not met in practice.

Following this observation, several risk assessment methods have been developed that require milder assumptions than the FORM-based ones. In [44, 195], the reliability assessment is performed thanks to importance sampling. Therefore, instead of estimating the probability boundary directly, they use an auxiliary distribution to sample points close to the probabilistic boundary. Moreover, in [44], they cleverly reuse previously sampled points to reduce the computational cost of the methods. Other methods [111, 147] use surrogate modeling to approximate the constraint functions and then perform a reliability analysis on the surrogate. In particular, [147] used the CE methods and the multifidelity model to reduce the computational cost. These previous methods have the advantage of performing the reliability evaluation without using the smoothness assumption on the constraint functions. However, they rely on the perfect knowledge of the distribution of $\boldsymbol{\xi}$. Moreover, these methods are focused on reliability estimation and cannot be directly applied to solve Problem (2.10).

Finally, methods for dealing with both types of uncertainty are described. In [4], a generalized model of uncertainties is defined that takes the form of an interval and allows to deal with both epistemic and aleatory uncertainties. Then, an advanced line search method computes the lower and upper bounds of the probabilistic constraints. In [66], the authors separate aleatory and epistemic uncertainty. They then use nested loops to propagate the uncertainties. The epistemic uncertainties are handled in the outer loop using various techniques such as Dempster-Schafer [172] or interval analysis [136]. The aleatory uncertainties are handled in the inner loop via polynomial chaos extension [191] or stochastic collocation [142]. A final approach considers Bayesian optimization [139] to incorporate epistemic uncertainties into FORM-based methods. Contrary to the previous work, it has the main advantage of not requiring interval formalism. All these methods seem particularly interesting for reliability analysis, but it needs work to be adapted for solving Problem (2.10), because this work considers the optimization problem. In [128], Problem (2.10) is solved in the presence of both aleatory and epistemic uncertainty, however the problem is based on analytical expression of the objective and constraint functions.

CHAPTER 3 THESIS OUTLINE AND CONTRIBUTIONS

This thesis is organized as follows. Chapter 2 defines the mathematical concepts employed in this thesis, followed by a critical review of the literature. This review is divided into three distinct parts. The first part details the methods developed to solve the DFO/BBO deterministic problem given in Equation (1.1). The second part concerns the methods dealing with the unconstrained uncertain problem given in Equation (1.4), while the last part describes the methods dealing with the uncertain constrained problems. Then, the three main contributions of this thesis are developed, based on three articles (two published and one submitted). Each contribution is preceded by a short introduction that outlines its concept and objectives.

The first contribution, in Chapter 4, aims to address the problem of multimodality. A heuristic, the cross-entropy method [167], is adapted to handle the constrained case by using the progressive barrier [11]. It is then incorporated as a search step in the Mesh Adaptive Direct Search (MADS) algorithm [12]. This step has the peculiarity of being executed only at certain iterations, depending on the problem, in order to avoid excessive function evaluations. This combination allows an efficient exploration of the design variable space and gives the algorithm the ability to escape certain local minima. Finally, it benefits from the convergence analysis of the MADS algorithm. The algorithm outperforms other state-of-the-art heuristics in terms of function evaluations on global optimization benchmarks and real-world engineering problems.

The second contribution, in Chapter 5, tackles the unconstrained stochastic optimization problems. Inspired by the work in [178], the unconstrained stochastic blackbox optimization problem is transformed into a sequence of smooth approximation subproblems. The smooth approximations are obtained using a smoothed functional method [166, Chapter 7.6]. Among all the properties of the smooth approximation, the most interesting is the ability to obtain an estimate of its gradient using only two function evaluations, regardless of the size of the problem. The sequence of subproblems is then solved by the sequential stochastic optimization Sequential Stochastic Optimization (SSO) algorithm. In its inner process, each subproblem is solved thanks to a zeroth-order version of the Sign Stochastic Gradient Descent with Momentum (ZO-Signum) algorithm. This algorithm benefits from a stopping criterion that depends on the norm of the momentum vector. In the outer loop, the accuracy of the stopping criterion required to consider a subproblem solved increases progressively. Under mild conditions, the convergence rate in mean of the ZO-Signum algorithm to a stationary point of a smoothed subproblem is derived. Under additional assumptions, it is proved that a subsequence of iterates of the SSO algorithm converges to an optimal point of the original problem, and its associated rate of convergence is derived. Finally, numerical experiments are

conducted to compare this algorithm with other algorithms and to demonstrate its competitiveness.

In Chapter 6, the last contribution deals with constrained problems subject to mixed aleatory/ epistemic uncertainty. The CVaR measure [162] is used to handle both types of uncertainty in the problem given in Equation (1.4). The CVaR measure depends on a parameter α that allows to choose the desired level of reliability in each constraint function. Then, the CVaR-constrained blackbox optimization problem is approximated by the smoothed functional method [33] based on the truncated Gaussian distribution. The smooth problem is shown to be a conservative approximation of the CVaR-constrained blackbox optimization problem. Finally, it is solved using a multi-timescale stochastic approximation algorithm [38]. The convergence analysis of this algorithm is based on a ODE approach and the Lyapunov theory. We prove that the algorithm converges to a locally optimal point of the smooth CVaR-constrained problem with probability one. Moreover, for values of α sufficiently close to 1, this algorithm almost surely converges to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution. Numerical experiments are performed on analytical engineering problems, first to experimentally set the various hyperparameters of the algorithm; second, to evaluate the effectiveness of the truncated Gaussian gradient estimator; third to demonstrate the ability of the algorithm to handle problems subject to mixed aleatory/epistemic uncertainty.

Finally, Chapter 8 summarizes the contributions of the thesis, underlines their limitations and proposes future research directions.

CHAPTER 4 ARTICLE 1: COMBINING CROSS ENTROPY AND MADS METHODS FOR INEQUALITY CONSTRAINED GLOBAL OPTIMIZATION

Title Combining Cross-Entropy and MADS methods for Inequality Constrained Global Optimization.

Authors Charles Audet, Jean Bigeon and Romain Couderc.

Journal Published in *Operation Research Forum* [21] on 12/07/2021.

4.1 Context

The first contribution of this thesis is an algorithm for solving deterministic blackbox optimization problems given in Equation (1.1). What emerges from Section 2.2 and is summarized in Table 2.1 is the ability of direct search algorithms to perform both local and global searches. This ability comes from the search step, which allows the use of arbitrary metaheuristics to explore the space of design variables or to speed up the convergence of the algorithm. To fully exploit the search step, a software package called NOMAD [17, 106] has been developed based on the MADS algorithm. In this software many different search steps have been implemented based on Latin hypercube sampling [176], variable neighborhood search [10], Nelder-Mead [20] or surrogate models [16]. However, only two search steps are specifically designed for global optimization: Latin Hypercube Sampling and Variable Neighborhood Search.

Given the flexibility offered by the search step and the numerous types of metaheuristics, there are many ways to create an exploration search step. Moreover, according to the "no free lunch theorems" [189], it is known that there is no theoretical best option. Therefore, we choose the Cross Entropy (CE) method because it is compatible with the MADS algorithm for two main reasons. First, it was originally developed for combinatorial optimization. Since the MADS algorithm runs on a discretization of the continuous space of design variables, this suggests that the method will also be efficient in this context. Second, the standard deviation computed at each iteration of the CE method can be used as an indicator of whether or not to perform an exploration search step. Thus, building on these ideas, the following work was developed.

4.2 Article

Abstract This paper proposes a way to combine the Mesh Adaptive Direct Search (Mads) algorithm with the Cross-Entropy (CE) method for nonsmooth constrained optimization. The CE method is used as an exploration step by the Mads algorithm. The result of this combination retains the convergence properties of Mads and allows an efficient exploration in order to move away from local minima. The CE method samples trial points according to a multivariate normal distribution whose mean and standard deviation are calculated from the best points found so far. Numerical experiments show the efficiency of this method compared to other global optimization heuristics. Moreover, applied on complex engineering test problems, this method allows an important improvement to reach the feasible region and to escape local minima.

Keywords: Cross Entropy, MADS, Global optimization, Derivative-free optimization, Blackbox optimization and Constrained optimization

Declarations

- **Funding:** Audet is supported by Ivado's fundamental research grant PRF-2019-8079623546. Couderc is supported by a French ministerial grant 8542z.
- **Conflict of interests:** The authors declare that they have no conflict of interest.
- **Availability of data and material:** The authors declare that they do not use any data.
- **Code availability:** The authors declare that they use the open source NOMAD software (available at https://www.gerad.ca/nomad/) and custom code.

4.3 Introduction

This work studies inequality constrained blackbox optimization problems of the form:

$$\min_{\mathbf{x}\in\Omega\subseteq\mathbb{R}^n} f(\mathbf{x}), \tag{4.1}$$

with

$$\Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j = 1, 2, ..., m\},$$

where $f : \mathbb{R}^n \to \bar{\mathbb{R}} = \mathbb{R} \cup \pm\infty$, $c : \mathbb{R}^n \to \bar{\mathbb{R}}^m$ and the set $\mathcal{X}$ represents bound constraints of type $\boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}$ with $\boldsymbol{\ell}, \mathbf{u} \in \bar{\mathbb{R}}^n$. The specificity of this work is due to the form of the objective

function f and of the constraints c_j. They can be the result of a simulation of complex physical phenomena. These simulations can take an important amount of time or present some discontinuities and therefore classical optimization methods are difficult to apply. Especially, when the gradient of the objective function and/or of the constraints is not explicitly known, hard to compute or its estimation is time consuming. This field is called derivative-free optimization (DFO). In the worst case, the gradient does not even exist, which is called blackbox optimization (BBO).

Specialized BBO and DFO algorithms have been developed in order to solve this kind of problem. There are two main categories: model based algorithms [54] and direct search algorithms [14]. This work deals with direct search algorithms which benefit from theoretical convergence results and adding some modifications may improve their performance. In particular, the Mesh Adaptive Direct Search (Mads) algorithm [12] ensures convergence to a point satisfying necessary conditions based on the Clarke calculus [52]. This theoretical guarantee is a solid basis for blackbox optimization. However, blackbox optimization algorithms must take into account two other types of difficulties. First, algorithms must be efficient in terms of simulation evaluations (constraints and objective function). Indeed, the simulation in an engineering context is often time consuming. Second, blackbox simulations may involve multi-extrema functions, so algorithms must be able to escape from local minima. Mads may be trapped in a local minimum.

To address the second difficulty, the Mads algorithm may be combined with Variable Neighborhood Search (VNS) [10] and Latin Hypercube Sampling (LHS) [176] techniques to escape local minima. Other heuristics of global optimization with no convergence guarantees exist including Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [78], Genetic Algorithm (GA) [75], Differential Evolution (DE) [153] or Particule Swarm Optimization (PSO) [89]. However, these heuristics often require a large number of function evaluations which is incompatible with the first difficulty. Moreover, no specific mechanism has been developed to enable these methods to deal with inequality constraints. In contrast, some methods have been developed to address the first problem, by reducing the overall number of simulation evaluation such as: the use of ensembles of surrogate [16] or of quadratic models [53] and the integration of the Nelder-Mead (NM) algorithm [20]. These different methods improve the efficiency of the Mads algorithm but do not address the difficulty of local optima.

The objective of the present research is to propose an alternative strategy: the Cross Entropy (CE) [167] method in hopes of escaping local minima. This method is a trade-off between a more global search and a limited number of blackbox evaluations. It was introduced in 1997, first in a context of rare

events in discrete optimization [168] and then adapted to continuous optimization [100]. The main benefit of using CE is that it often converges rapidly to a promising region in the space of variables. However, this method does not benefit from theoretical guarantees, and once it has found a promising region, it requires a large number of simulation evaluations to improve the local accuracy. The two aims of this work are: global exploration with limited number of iterations while preserving the theoretical convergence guarantees. In this purpose, CE is used as a step in the Mads algorithm. The present work proposes a way to include a CE global exploration strategy within the Mads algorithm.

This paper is divided as follows, Section 2 proposes an overview of the Mads and Cross Entropy methods. Section 3 presents an algorithm combining CE and Mads. Finally, Section 4 shows the main numerical results comparing the proposed method with other Mads type algorithm and state-of-the-art heuristics. Section 5 concludes on future work and on the contributions of this paper.

4.4 Description of Mads and Cross Entropy algorithms

This Section describes the Mads and CE algorithms.

4.4.1 The Mads constrained optimization algorithm

The present work considers the Mads algorithm with the progressive barrier (PB) [11] to handle inequality constraints and with dynamic scaling [18] to handle the varying magnitudes of the variables. Mads is a direct search algorithm (see algorithm 8 below). It proceeds iteratively where the blackbox functions are evaluated at some trial points. These points are either accepted as new iterates or rejected, depending on the value of the objective function and of the constraints violations. A key principle of the Mads algorithm is that the candidate points may only be chosen on a discretization of the space of variables called the mesh. This discretization is adaptative and its fineness is controlled by the mesh size vector $\boldsymbol{\delta}^k \in \mathbb{R}^n_+$. In its simplest form, the mesh [18] is defined as follows:

$$M^k = V^k + \{diag(\boldsymbol{\delta}^k)\mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$$

where V^k is the cache containing all points visited by the start of iteration k. A point $\mathbf{x}$ belongs to V^k if and only if both the objective function and the constraints were evaluated by the start of iteration k. The first set V^0 may be initialized by the user or by a collection of points generated by LHS for instance.

Each iteration includes three steps. The first is an optional step called the SEARCH where various strategies may be used to explore the space of variables. In practice, the SEARCH accelerates the

convergence to an optimum and it may attempt to escape from local minima. The only rules to follow are that the trial points must remain on the current mesh M^k and that the search terminates in finite time. It is in this step of the algorithm that CE was integrated.

The second step is mandatory and called the POLL. It is the where the space of variable is locally explored by following strict rules guaranteeing convergence. The POLL is confined to a region delimited by the so called poll size vector $\boldsymbol{\Delta}^k \in \mathbb{R}^n_+$ which is taken such that $\boldsymbol{\Delta}^k \leq \boldsymbol{\delta}^k$. This region is defined as follows:

$$F^k = \{\mathbf{x} \in M^k : |\mathbf{x}_j - \mathbf{x}^k_j| \leq \boldsymbol{\Delta}^k_j,\ \forall j \in [1, n]\}$$

with $\mathbf{x}^k$ the current incumbent solution . Then, this step only consist to select a positive spanning set $\mathbb{D}^k_{\Delta^k}$ such that

$$P^k = \{\mathbf{x}^k + \boldsymbol{\delta}^k \mathbf{d} : \mathbf{d} \in \mathbb{D}^k_{\boldsymbol{\Delta}^k}\}$$

is a subset of F^k of extent $\mathbb{D}^k_{\boldsymbol{\Delta}^k}$ and to evaluate the objective functions and the different constraints at these points.

Finally, the last step updates the mesh and the poll size vectors at the end of each iteration. The values of both vectors are reduced when an iteration fails to improve the current solution and they are increased or remain at the same value otherwise. To handle the constraints, the PB is used. This method is based on the constraints violation function [69]

$$h(\mathbf{x}) := \begin{cases} \sum_{j=1}^{m} \left(\max\{c_j(\mathbf{x}), 0\}\right)^2 & \text{if } \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n \\ \infty & \text{otherwise.} \end{cases}$$

The constraints violation function value $h(\mathbf{x})$ is equal to 0 if and only if the point $\mathbf{x}$ belongs to Ω and is strictly positive otherwise. This function allows to rank any pair of trial points by using the following dominance relation [14].

Definition 4.4.1. *The feasible point $\mathbf{x} \in \Omega$ is said to dominate $\mathbf{y} \in \Omega$ when $f(\mathbf{x}) < f(\mathbf{y})$. The infeasible point $\mathbf{x} \in \mathcal{X} \setminus \Omega$ is said to dominate $\mathbf{y} \in \mathcal{X} \setminus \Omega$ when $f(\mathbf{x}) \leq f(\mathbf{y})$ and $h(\mathbf{x}) \leq h(\mathbf{y})$ with at least one strict inequality.*

The PB method approaches an optimal solution by locally exploring around two incumbent solutions. The feasible incumbent solution $\mathbf{x}^{feas} \in \Omega$ and the infeasible incumbent solution $\mathbf{x}^{inf}$ which is the undominated infeasible point with a value of h lower than a threshold called $h_{\max}$. In practice, the threshold $h^k_{\max}$ decreases progressively toward zero without ever reaching it. Exploring around $\mathbf{x}^{inf}$ may be interesting because it is possible that while the threshold is pushing towards zero a

feasible candidate point with a low objective function is generated. Thus, the poll step is applied around these two incumbent solutions. An iteration of the Mads algorithm with the progressive barrier may be of three types:

- A dominating iteration occurs when a dominating trial point with respect to $\mathbf{x}^{inf}$ or $\mathbf{x}^{feas}$ is found. In this case, the threshold is updated to $h_{\max}^{k+1} = h_I^k$.
- An improving iteration occurs when it is not a dominating iteration but a trial point improves the threshold $h_{\max}^k$. In this case, threshold is updated to $h_{\max}^{k+1} = \max\{h(\mathbf{v}) : h(\mathbf{v}) < h_I^k, \mathbf{v} \in V^{k+1}\}$.
- An unsuccessful iteration occurs when it is neither a dominating nor an improving iteration. In this case, the threshold is updated to $h_{\max}^{k+1} = h_I^k$.

where h_I^k is defined by

$$h_I^k = \begin{cases} h(\mathbf{x}) & \text{for any } \mathbf{x} \in I^k = \{\underset{\mathbf{x} \in U^k}{\operatorname{argmin}}\{f(\mathbf{x}) : 0 < h(\mathbf{x}) \leq h_{\max}^k\}\}, \\ \infty & \text{if } I^k = \emptyset, \end{cases}$$

with U^k the set of infeasible undominated points. Algorithm 1 provides a description of Mads with progressive barrier algorithm, the reader may consult [14] for more details, and to [11] for a complete presentation.

The fundamental convergence result [11] of the Mads algorithm with progressive barrier states that if the entire sequence of trial points belongs to a bounded set, then there exists an accumulation point $\mathbf{x}^*$ such that the generalized directional derivative $f^\circ(\mathbf{x}^*; \mathbf{d})$ of Clarke [52] is nonnegative in every hypertangent [83] direction $\mathbf{d}$ to the domain Ω at $\mathbf{x}^*$ provided that $\mathbf{x}^*$ is feasible. A similar result holds for the constraint violation function h over the set $\mathcal{X}$ in situations where the iterates never approach the feasible region.

4.4.2 The Cross Entropy method for continuous optimization

The Cross Entropy method was introduced by Rubinstein in 1997 in the context of a minimization algorithm for estimating probabilities of rare events [168]. Later, it was modified to solve combinatorial optimization problems [167] and then in 2006 to solve continuous problems [100]. The main idea of this method is as follows. First, each optimization problem is transformed into a rare event estimation problem called associated stochastic problem (ASP). For instance, the deterministic Problem (1) is transformed as the minimization of the expectation:

Algorithm 8 The Mesh Adaptive Direct Search algorithm (Mads)

0. Initialization:
   A set of starting point: $V^0 \subset \mathbb{R}^n$
   An initial poll size vector: $\boldsymbol{\Delta}^0$
   The iteration counter: $k \leftarrow 0$
   The mesh size adjustment parameter: $\tau \in \mathbb{Q} \cap\ (0,1)$
   The initial threshold: $h^0_{\max} = \infty$
   Define $\boldsymbol{\delta}^{\mathbf{0}}_{\boldsymbol{j}} = \min\{{\boldsymbol{\Delta}_{\boldsymbol{j}}}^0,\ ({\boldsymbol{\Delta}_{\boldsymbol{j}}}^0)^2\},\ \forall j \in [1,n]$
1. Search step (optional):
   Launch the simulation on a finite set S^k of mesh points.
   If successful, go to 3.
2. Poll step:
   Launch the simulation on the set P^k of poll points.
3. Updates:
   Update the cache V^{k+1}.
   If the iteration is dominating :
      update $\mathbf{x}^{k+1}$, $h^{k+1}_{\max} = h^k_I$ and $\boldsymbol{\Delta}^{k+1} = \tau^{-1}\boldsymbol{\Delta}^k$.
   Else if the iteration is improving:
      update $\mathbf{x}^{k+1}$,$h^{k+1}_{\max} = \max\{h(\mathbf{v}) : h(\mathbf{v}) < h^k_I, \mathbf{v} \in V^{k+1}\}$ and $\boldsymbol{\Delta}^{k+1} = \boldsymbol{\Delta}^k$.
   Otherwise:
      Set $\boldsymbol{\Delta}^{k+1} = \tau\boldsymbol{\Delta}^k$ and $h^{k+1}_{\max} = h^k_I$.
   Set $\boldsymbol{\delta}^{\boldsymbol{k+1}}_{\boldsymbol{j}} = \min\{{\boldsymbol{\Delta}_{\boldsymbol{j}}}^{k+1},\ ({\boldsymbol{\Delta}_{\boldsymbol{j}}}^{k+1})^2\},\ \forall j \in [1,n]$.
   Increase the iteration counter $k \leftarrow k+1$ and go to 1.

$$\min_{\mathbf{X}\in\Omega,\gamma\in\mathbb{R}} P(f(\mathbf{X}) \leq \gamma) = \min_{\mathbf{X}\in\Omega,\gamma\in\mathbb{R}} E(I_{\{f(\mathbf{X})\leq\gamma\}}) \tag{4.2}$$

where $\mathbf{X}$ is a random vector and $I_{\{f(\mathbf{X}\leq\gamma\}}$ is the indicator function. Then, this ASP is tackled efficiently by an adaptive algorithm. This algorithm constructs a sequence of solutions which converging to the optimal solution of the ASP. The CE method is composed of two iterative steps:

- generation a sample of random data according to a density of probability;
- density parameters update thanks to the data sampled to create a new sample in the next iteration.

It results that this method often escapes from local minima.

4.4.2.1 An introductory example

For clarity, consider the example from [100] of minimizing the function:

$$f(x) = -e^{-(x-2)^2} - 0.8e^{-(x+2)^2}, x \in \mathbb{R}. \tag{4.3}$$

The function f has two local minima and a single global minimum at $x = 2$.

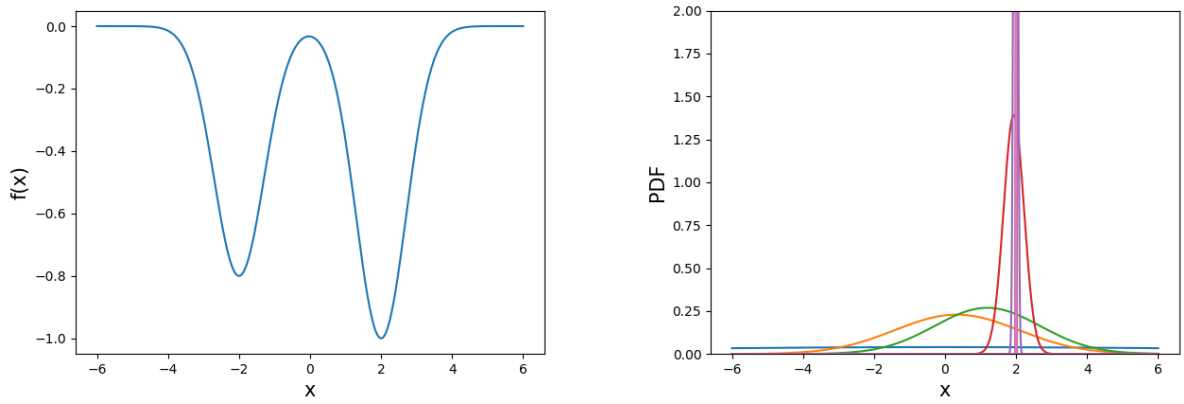

Figure 4.1 (Figure inspired by [100]) Graph of objective function f (left) and evolution of the normal distribution during the seven first iterations with $\mu_0 = 0$, $\sigma_0 = 10$, $N_e = 10$ and $N_s = 50$ (right).

Using a normal distribution the CE procedure is the following:

- Initialization : at the first iteration $k = 0$, a mean $\boldsymbol{\mu}^0 \in \mathbb{R}^n$ and a standard deviation $\boldsymbol{\sigma}^0 \in \mathbb{R}^n$ (with n the dimension of the problem) are arbitrarily chosen. A large value of $\boldsymbol{\sigma}^0$ is taken in order to escape from local solutions.
- Iterative part: at each iteration $k \geq 1$:
  - First, a sample $\mathbf{X}_1, ..., \mathbf{X}_{N_s}$ of points in $\mathbb{R}^n$ is generated from a normal law $\mathcal{V}(\boldsymbol{\mu}^{k-1}, \boldsymbol{\sigma}^{k-1})$ where N_s is the number of samples.
  - Then, f is evaluated at each sampled points and a number of *elite* points N_e, with the lowest value of f. $\boldsymbol{\mu}^k$ and $\boldsymbol{\sigma}^k$ are the mean and standard deviation of these N_e points.
  - Termination: once the standard deviation becomes sufficiently small, the procedure is stopped.

The sequence of normal distribution is illustrated in the right part of Figure 4.1. This example

shows how the CE procedure escapes from the local minimum at $x = -2$ and converges in seven iterations to the neighborhood of $x = 2$.

4.4.2.2 The general CE method

Before presenting the CE method introduced in [100], the ASP is considered and the two iterative steps of the algorithm are precised. Problem 4.1 is transformed into an ASP. Using a family of probability distribution functions (pdf) $\{g(\cdot;\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}$ where g is the law chosen to sample the different points at each iteration. $\mathcal{V}$ is the set of vector parameters of the pdf g which are calculated at each iteration. In the previous example g is taken as the normal law and the $\mathbf{v}^k \in \mathcal{V}$ is composed of the mean and standard deviation $\mathbf{v}^k = (\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k)$. Having explained the law and its parameters, the ASP related to Problem 4.1 can be defined as follows:

$$\min_{\mathbf{X}\in\Omega,\gamma\in\mathbb{R}} P_{\mathbf{v}}(f(\mathbf{X}) \leq \gamma) = \min_{\mathbf{X}\in\Omega,\gamma\in\mathbb{R}} E_{\mathbf{v}}(I_{\{f(\mathbf{X})\leq\gamma\}}) \tag{4.4}$$

where $\mathbf{v} \in \mathcal{V}$ is a vector of parameter, $\mathbf{X}$ is a random vector with a pdf $g(\cdot;\mathbf{v})$ and γ is a variable. At this stage, for a given value of γ, the parameter $\mathbf{v}$ may be estimated. Conversely, given a vector of parameters $\mathbf{v}$, the value γ may be also estimated. The CE method is based on these two estimations, at each iterations, the algorithm estimates one then the other. In the iterative part of Example 4.3, the first item corresponds to the estimation of γ and the second one to the estimation of $\mathbf{v}$. More precisely, we denote $\gamma^* \in \mathbb{R}$ as the infemum of the objective function, $\mathbf{v}^*$ the parameters and $g(\cdot;\mathbf{v}^*)$ the pdf associated to this infemum. The goal is to generate a sequence $(\gamma^k, \mathbf{v}^k)$ converging to $(\gamma^*, \mathbf{v}^*)$. To achieve this goal, a sequence of pdf $g(\cdot;\mathbf{v}^0), g(\cdot;\mathbf{v}^1), ...$ converging to $g(\cdot;\mathbf{v}^*)$ is created. To assure the convergence, one must have a “measure" of the difference between the iterate pdf $g(\cdot;\mathbf{v}^k)$ and the objective one $g(\cdot;\mathbf{v}^*)$. The Kullback-Leibler (KL) divergence [101] is used:

$$D(g(\cdot;\mathbf{v}^*)||g(\cdot;\mathbf{v}^k)) = \int_{-\infty}^{\infty} g(\mathbf{x};\mathbf{v}^*) \ln\left(\frac{g(\mathbf{x};\mathbf{v}^*)}{g(\mathbf{x};\mathbf{v}^k)}\right) d\mathbf{x}. \tag{4.5}$$

The iterative steps may now be described. ρ is defined as a very small quantity, corresponding to the proportion of elite points which are kept from an iteration to another. The procedure is:

- **Adaptive update of** γ^k. With a fixed parameter of pdf $\mathbf{v}^{k-1}$, γ^k is defined such that it is the $(1-\rho)$-quantile of $f(\mathbf{X})$ under v_{k-1}. Then, γ^k satisfies:

$$P_{\mathbf{v}^{k-1}}(f(\mathbf{X}) \leq \gamma^k) \geq \rho, \tag{4.6}$$

$$P_{\mathbf{v}^{k-1}}(f(\mathbf{X}) \geq \gamma^k) \geq 1-\rho \tag{4.7}$$

where $X \sim g(\cdot; \mathbf{v}^{k-1})$. The γ^k is denoted $\hat{\gamma}^k$. To obtain this estimator, a sample $\mathbf{X}_1, ..., \mathbf{X}_{N_s}$ is drawn from $g(\cdot; \mathbf{v}^{k-1})$ and evaluated. Then, the $(1-\rho)$ quantile is:

$$\hat{\gamma}^k = f_{\lceil\ (1-\rho)N_s \rceil}. \tag{4.8}$$

- **Adaptive update of $\mathbf{v}^k$.** With a fixed γ^k and knowing $\mathbf{v}^{k-1}$, $\mathbf{v}^k$ is a solution of:

$$\begin{aligned} \max_{\mathbf{v}} D(\mathbf{v}) &= \max_{\mathbf{v}} E_{\mathbf{v}^{k-1}} \left(I_{\{f(\mathbf{X}) \leq \gamma^k\}} \ln(g(\mathbf{X}; \mathbf{v})) \right) \\ &= \min_{\mathbf{v}} E_{\mathbf{v}^{k-1}} \left(I_{\{f(\mathbf{X}) \leq \gamma^k\}} \ln \left(\frac{I_{\{f(\mathbf{X}) \leq \gamma^k\}}}{g(\mathbf{X}; \mathbf{v})} \right) \right) \end{aligned} \tag{4.9}$$

  which is the minimization of the KL divergence at iteration k (with the convention $0 \ln(0) = 0$). Nevertheless, in practice, the real expectation and the real γ^k are not known, estimators must be used and the following equation is solved:

$$\tilde{\mathbf{v}}^k \in \operatorname*{argmax}_{\mathbf{v}} \widehat{D}(\mathbf{v}) = \frac{1}{N_s} \sum_{i=1}^{N_s} I_{\{f(\mathbf{X}_i) \leq \widehat{\gamma}^k\}} \ln(g(\mathbf{X}_i; \mathbf{v}). \tag{4.10}$$

Last but not least, $\hat{\mathbf{v}}^k$ is not set to $\tilde{\mathbf{v}}^k$. There are two reasons for that: first the value of $\tilde{\mathbf{v}}^k$ are smoothed. Second, some components of $\tilde{\mathbf{v}}^k$ could be set to 0 or 1 at the first few iterations and the algorithm could converge to non optimal solution. To avoid these problems, the authors of [100] propose to use the following convex combination:

$$\hat{\mathbf{v}}^k = \alpha \tilde{\mathbf{v}}^k + \ (1-\alpha) \hat{\mathbf{v}}^{k-1} \tag{4.11}$$

with $0 < \alpha \leq 1$. Theoretically, any distribution converging in the neighborhood where the global maximum is attained can be used including normal, double exponential or beta distribution. Nevertheless, the beta distribution has for support $[0, 1]$ which is not suitable for global exploration, the double exponential may introduce some discontinuities and the updating step is quite simple with the normal distribution. Thus, in practice the normal distribution is often chosen [121, 133, 179]. Therefore, the detailed algorithm is presented next:

4.5 The CE-MADS constrained optimization algorithm

This section presents the CE-inspired SEARCH step of Mads. Section 4.1 describes how to handle constraints, the update of the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ and the condition to enter the CE search step. Section 4.2 presents the algorithm of the CE SEARCH step.

Algorithm 9 The Cross Entropy (CE) algorithm with normal law

Choose $\hat{\boldsymbol{\mu}}^0 \in \mathbb{R}^n$ and $\hat{\boldsymbol{\sigma}}^0 \in \mathbb{R}^n$
Set the iteration counter: $k \leftarrow 0$
N_s number of sampled data at each iteration
N_e number of elite population
α the parameter of convex combination

1. Estimation of γ^k:
   Generate a random sample $\mathbf{X}_1, .., \mathbf{X}_{N_s}$ from $N(\hat{\boldsymbol{\mu}}^{k-1}, \hat{\boldsymbol{\sigma}}^{k-1})$ distribution.
   Evaluation of the N_s points by the simulation and then go to 2.
2. Estimation of mean and standard deviation
   Let E^k be the indices of the N_e best perfoming samples.
   Set $\tilde{\boldsymbol{\mu}}^k = \frac{1}{N_e} \sum_{j \in E^k} \mathbf{X}_j$
   and $[\tilde{\boldsymbol{\sigma}}^k]_i = \sqrt{\frac{1}{N_e - 1} \sum_{j \in E^k} [\mathbf{X}_j - \boldsymbol{\mu}^k]_i^2}, \ \forall i \in [1, n]$.
3. Updates:
   Apply the convex combinations:
   $\hat{\boldsymbol{\mu}}^k = \alpha \tilde{\boldsymbol{\mu}}^k + \ (1 - \alpha) \hat{\boldsymbol{\mu}}^{k-1}$
   $\hat{\boldsymbol{\sigma}}^k = \alpha \tilde{\boldsymbol{\sigma}}^k + \ (1 - \alpha) \hat{\boldsymbol{\sigma}}^{k-1}$
   Increase the iteration counter $k \leftarrow k + 1$ and go to 1.

4.5.1 The CE-SEARCH step

4.5.1.1 The choice of the elite points

Section 2 presented the CE method for unconstrained optimization. In [100], the bound constrained case is treated using a truncated normal law and a penalty approach is used for inequality constraints. In our work, the truncated normal law is also used to treat the bound constraints. For general inequality constraints, the algorithm does not use the penalty approach. The proposed approach is derived from the progressive barrier method described in Section 4.4.1. In fact, when the algorithm chooses the elite sample, it uses the following function Best (defined in [20] and recalled here). Thus, any points in the cache may be selected even if its value of constraint violation is over the threshold of progressive barrier [11]. The definition relies on both the objective and the constraint violation functions f and h.

Definition 4.5.1. *The function* Best : $\mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$

$$\mathsf{Best}(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \boldsymbol{x} & \textit{if } \boldsymbol{x} \textit{ dominates } \boldsymbol{y} \textit{ or if } h(\boldsymbol{x}) < h(\boldsymbol{y}), \\ \boldsymbol{y} & \textit{if } \boldsymbol{y} \textit{ dominates } \boldsymbol{x} \textit{ or if } h(\boldsymbol{y}) < h(\boldsymbol{x}), \\ \mathsf{Older}(\boldsymbol{x},\boldsymbol{y}) & \textit{Otherwise} \end{cases}$$

returns the best *of two points.*

The function Older gives the point which was generated before the former one. Thanks to this definition, CE may treat the general inequality constraints with the terminology used in Mads.

4.5.1.2 Update of the mean and standard deviation

Three elements differ compared to classical CE method concerning the mean and the standard deviation. First, the elite points taken to compute the mean and the standard deviation are not only the N_s points generated by the normal law. The elite points are chosen from the cache at the iteration k, denotes $V^k \subset \mathbb{R}^n$, so any points generated by the Mads algorithm may be selected. This set is ordered with the function Best, in order to select the N_e elite points, it is sufficient to take the N_e first points of V^k.

Second, the mean and the standard deviation initialization procedure differs from the CE method proceeds. Indeed, Mads always begins with a starting point, thus there is at least one point in the cache (the set of evaluated points). Moreover, to avoid generating trial points far from the current solution, bounds are added to the problem as follows (using $\bar{\mathbf{x}}^k$ the poll center at iteration k):

$$(\ell_i^k, u_i^k) = \begin{cases} (\ell_i, u_i) & \text{if } \ell_i \neq -\infty \text{ and } u_i \neq \infty, \\ (\bar{x}_i^k - \tau^{-1}\Delta_i^{\max}, u_i) & \text{if } \ell_i = -\infty \text{ and } u_i \neq \infty, \\ (\ell_i, \bar{x}_i^k + \tau^{-1}\Delta_i^{\max}) & \text{if } \ell_i \neq -\infty \text{ and } u_i = \infty, \end{cases} \quad \forall i \in [1,n]. \tag{4.12}$$

where $\Delta_i^{\max} = \max\{\Delta_i^0, ..., \Delta_i^k\}$ for each $i \in [1,n]$. The sequance $(\Delta_i^{\max})_k$ is non decreasing with respect to k and the product $\tau^{-1}\Delta_i^{\max}$ is always larger than Δ_i since $\tau^{-1} > 1$. Once the problem has finite bound constraints, there are two cases to calculate the mean $\boldsymbol{\mu}^k \in \mathbb{R}^n$ and the standard deviation $\boldsymbol{\sigma}^k \in \mathbb{R}^n$:

- In case where the number of points in the cache is too small to be relevant, i.e. fewer points that the number N_e required, then the mean and the standard deviation are determined such

that:

$$\boldsymbol{\mu}^k = \bar{\mathbf{x}}^k \tag{4.13}$$

$$\boldsymbol{\sigma}^k = 2\,(\mathbf{u}^k - \boldsymbol{\ell}^k) \tag{4.14}$$

- In the others cases, the same calculations are made that in the original CE process:

$$\boldsymbol{\mu}^k = \frac{1}{N_e} \sum_{j \in E^k} \mathbf{X}_j$$

$$[\boldsymbol{\sigma}^k]_i = \sqrt{\frac{1}{N_e - 1} \sum_{j \in E^k} [\mathbf{X}_j - \boldsymbol{\mu}^k]_i^2}, \ \forall i \in [1, n].$$

Third, to generate the point during the CE search, the truncated normal law was always used with the bounds created in (12). Moreover, the elite points come not only from the previous normal sampling but also of the other kind of search step. That gives a vector of standard deviation which tends to zero very quickly, the other methods doing generally a local search. That is why, the standard deviation is calculated as in 4.11 with a coefficient $\alpha = 0.7$.

4.5.1.3 The condition to pass in the CE-SEARCH step

The goal of the CE method is to explore in few evaluations the space to determine the promising region. The number of evaluations used by the CE SEARCH step must be quite small. For this purpose, Mads does not perform the CE SEARCH step at every iteration. The standard deviation can be seen as a measure of the incertitude on the data and is used to determine whether to launch the SEARCH step or not. First, a new variable called $\boldsymbol{\sigma}^p$ is introduced, it represents the incertitude measured the last time the algorithm passed through the CE SEARCH step and generated trial points. This variable is initialized to ∞. Then, the condition to launch the CE SEARCH is the following:

$$||\boldsymbol{\sigma}^k|| < ||\boldsymbol{\sigma}^p|| \tag{4.15}$$

This conditions means that the current incertitude is smaller than the previous one. Each time this conditions is respected, $\boldsymbol{\sigma}^p$ is updated with the standard deviation obtained after the CE step. Last but not least, there is a special case. The CE method being associate with Mads which is a local search, it is possible that the points become rapidly close to each others, reducing the standard deviation. In some cases, that avoids to escape from unfeasible region. Thus, in case where several iterations of Mads algorithm are passed and the feasible region is still not reached, then the CE-SEARCH is launched with a mean equal to the current best point and a standard deviation equal to

2 times the initial standard deviation until a feasible point is found.

4.5.2 The complete algorithm

The CE SEARCH step of Mads algorithm is presented here:

Algorithm 10 The CE SEARCH step

1. Calculation of $\boldsymbol{\mu}^k$ and $\boldsymbol{\sigma}^k$:
   - if $card(V^k) < N_e$:
     - $\boldsymbol{\mu}^k = \bar{\mathbf{x}}_0$
     - $\boldsymbol{\sigma}^k = 2\,(\mathbf{u}^k - \boldsymbol{\ell}^k)$
   - else:
     - $\boldsymbol{\mu}^k = \frac{1}{N_e} \sum_{j \in E^k} \mathbf{X}_j$
     - $[\boldsymbol{\sigma}^k]_i = \sqrt{\frac{1}{N_e - 1} \sum_{j \in E^k} [\mathbf{X}_j - \boldsymbol{\mu}^k]_i^2},\ \forall i \in [1, n]$
2. CE SEARCH
   - If $||\sigma^k|| < ||\sigma^p||$:
     - Generate a random sample $\mathbf{X}_1, .., \mathbf{X}_{N_s}$ from $\mathcal{N}(\boldsymbol{\mu}^k, 2\boldsymbol{\sigma}^k)$ distribution and project them on the mesh.
     - Evaluation of the N_s points by the simulation.
     - Update:
     - $\boldsymbol{\mu}^{k+1} = \frac{1}{N_e} \sum_{j \in E^k} \mathbf{X}_j$
     - $[\boldsymbol{\sigma}^{k+1}]_i = \sqrt{\frac{1}{N_e - 1} \sum_{j \in E^k} [\mathbf{X}_j - \boldsymbol{\mu}^{k+1}]_i}$
     - $(\boldsymbol{\sigma}^p)^2 = (\boldsymbol{\sigma}^{k+1})^2$

4.6 Computational experiments

The present work uses data profiles to compare the different algorithm. Data profiles [138] allow to assess if algorithms are successful in generating solution values close to the best objective function values. To identify a successful run, a convergence test is required. Let denote $\mathbf{x}_e$ the best iterates obtained by one algorithm on one problem after e evaluations, f_{fea} a common reference for a given problem obtained by taking the max feasible objective function values on all run instances of that problem for all algorithms and f^* the best solution obtained by all tested algorithms on all run instances of that problem. Then, the problem is said to be solved within the convergence tolerance τ when:

$$f_{fea} - f(\mathbf{x}_e) \geq (1 - \tau)(f_{fea} - f^*).$$

Different initial points constitute different problems. Moreover, an instance of a problem corresponds to a particular pseudo-random generator seeds. The horizontal axis of a data profile represents the number of evaluations for problems of fixed dimension, and represents group of $n + 1$ evaluations when problems of different dimension are involved. The vertical axis corresponds to the proportion of problems solved within a given tolerance τ. Each algorithm has its curve to allow comparison of algorithms capability to converge to the best objective function value.

This section presents the numerical experiments. It is divided in two subsections. The numerical experiments of Section 4.1 are performed on analytical test problems to calibrate the CE-SEARCH parameters. Section 4.2 compares CE-Mads with others state-of-the-art global optimization method. Finally, section 4.3 compares Mads, LH-Mads, VNS-Mads and CE-Mads without the use of models on three real engineering problems.

4.6.1 Preliminary experiments to calibrate parameters

Computational experiments are conducted using the version 3.9.1 of NOMAD [106] software package. All tests use the Mads strategy with the use of the NM search [20] and without the use of models [16, 53]. When the CE-SEARCH is used, it is the first SEARCH step to be applied.

Numerical experiments on analytical test problems are conducted to set default values for the three algorithmic parameters: the parameter of the convex combination α, the number of sampled data at each iteration N_s and the number of elite population N_e. CE-Mads is tested on 100 analytical problems from the optimization literature. The characteristics and sources of these problems are summarized in Table 1 in appendix 4.8. The number of variables ranges from 2 to 60; 28 problems have constraints other than bound constraints. In order to have a more precise idea of the effect between the hyper-parameters (N_e and N_s), three series of tests are conducted:

- A series of tests on the 69 unconstrained test problems having a dimension from 2 to 20.
- A series of tests on the 25 constrained test problems having a dimension from 2 to 20.
- A series of tests on the 6 larger problems in term of dimension (from 50 to 60), three are constrained and three are not.

For each test, the maximal number of function evaluations is set to $1000(n + 1)$, where n is the number of variables and each problem is run with 3 different random seeds.First, for each series of tests, the five following CE-Mads setup of hyper-parameters are compared: $(N_e, N_s) \in \{ (2, n),\ (4, 2n),\ (6, 3n),\ (8, 4n),\ (10, 5n)\}$ with n the dimension of the test problem and the α

value is fixed to 0.7 as in the example 3.1 of [100].A run called NOMAD default is added in each series of test to compare our results with the current NOMAD software. Data profiles are presented on Figure 4.2, 4.4 and 4.6 with different values of the tolerance τ.

These results are analysed by series of problems:

- On the unconstrained problems (see Figure 4.2), no algorithm really stands out regardless of the value of τ, it is difficult to choose one hyper-parameter rather than another one even if the couple $N_e = 4$ and $N_s = 2n$ appears to be more efficient.
- On the constrained problems (see Figure 4.4), there are different behaviors according to the value of τ. For $\tau = 10^{-3}$, no algorithm appears to be dominant. However, for $\tau = 10^{-5}$, it happens that greater are the values of N_s and N_e, higher is the percentage of problems solved finally. That can be explained because great N_s and N_e allow a better exploration of the space, and so a more precise result at the end.
- Finally (see Figure 4.6), on the large test problems, and for small values of the tolerance τ the CE-Mads is outperformed by the Mads algorithm with default values. It seems that the CE method is not useful for problems with a large number of variables.

Second, N_e and N_s are fixed to 4 and $2n$ repectively Then, the five following CE-Mads setup of hyper-parameters are compared: $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Data profiles are presented on Figure 4.3, 4.5 with different values of the tolerance τ. The tests are not run on the problem with large dimension given the poor performance of CE-Mads on this kind of problems. The results show that none α value outperformed the other ones in any runs of tests. Inspection of the logs of the hyper-parameter calibration reveals the two following observations:

- The CE-Mads performance is not very sensitive to the hyper-parameter values. This allows to avoid some calibration experiments before applying the algorithm on a new test problem. This is particularly interesting in an engineering context.
- For problems with a large number of variables, our tests suggest to avoid using of the CE-SEARCH. Nevertheless, this point has not been confirmed on real engineering problems given that we do not have access to engineering test problems with large dimension.

In the remainder of the paper, the CE-SEARCH values are set to $\alpha = 0.7$, $N_e = 4$ and $N_s = 2n$ as they often perform well.

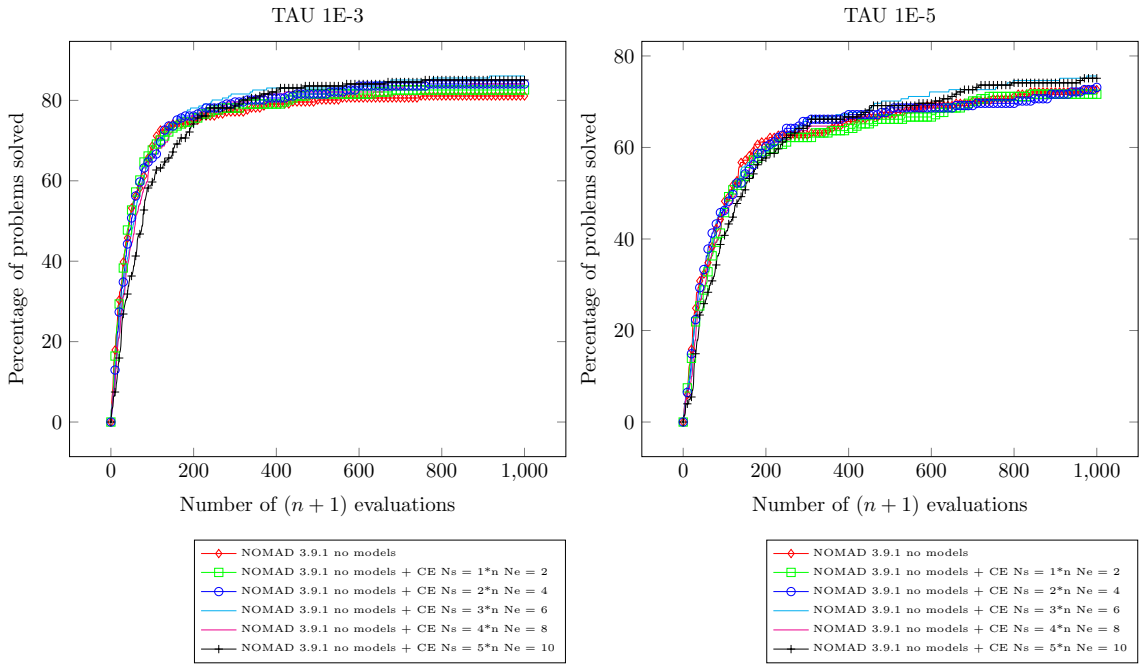

Figure 4.2 Result of calibration of the hyper-parameters N_e and N_s of CE-MADS on the 69 unconstrained test problems

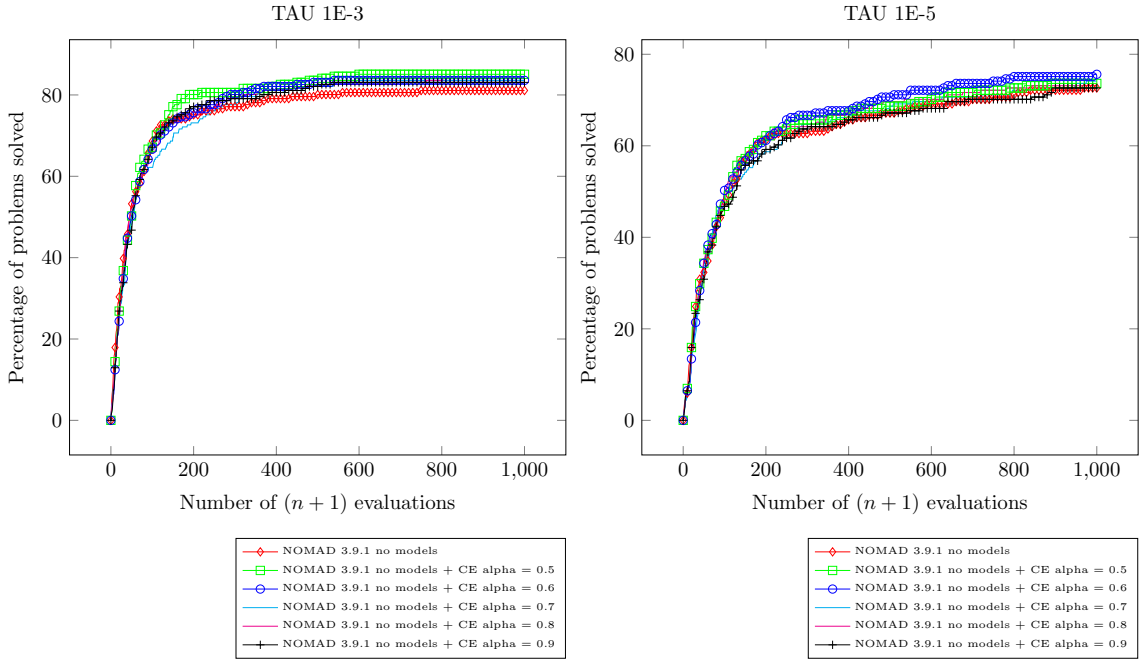

Figure 4.3 Result of calibration of the hyper-parameter α of CE-MADS on the 69 unconstrained test problems

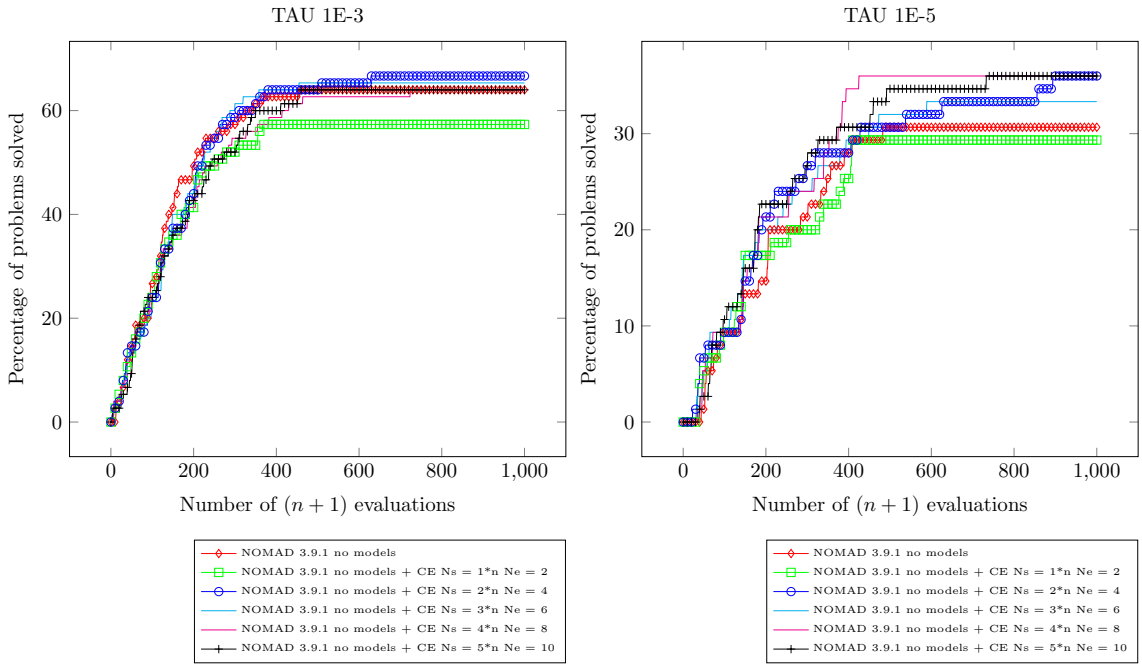

Figure 4.4 Result of calibration of the hyper-parameters N_e and N_s of CE-MADS on the 25 constrained test problems

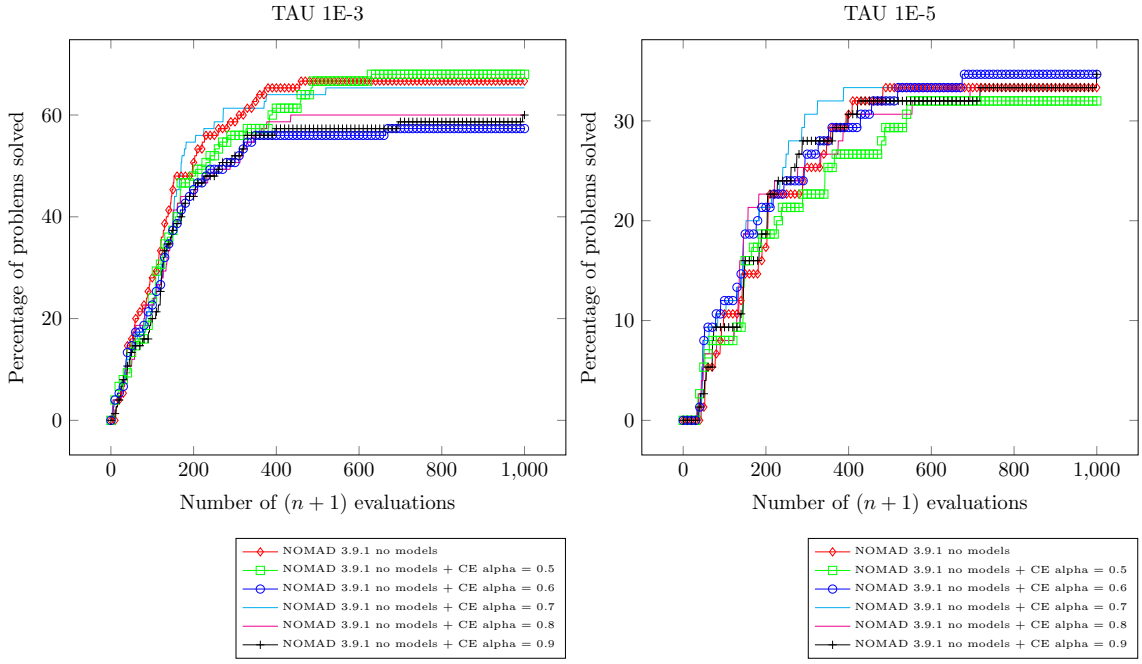

Figure 4.5 Result of calibration of the hyper-parameter α of CE-MADS on the 25 constrained test problems

4.6.2 Comparisons between CE-Mads and some of state-of-the art global optimization heuristics

In this section, we compared CE-Mads with other well-known global optimization heuristics designed to escape local minima on a collection of unconstrained global optimization benchmark

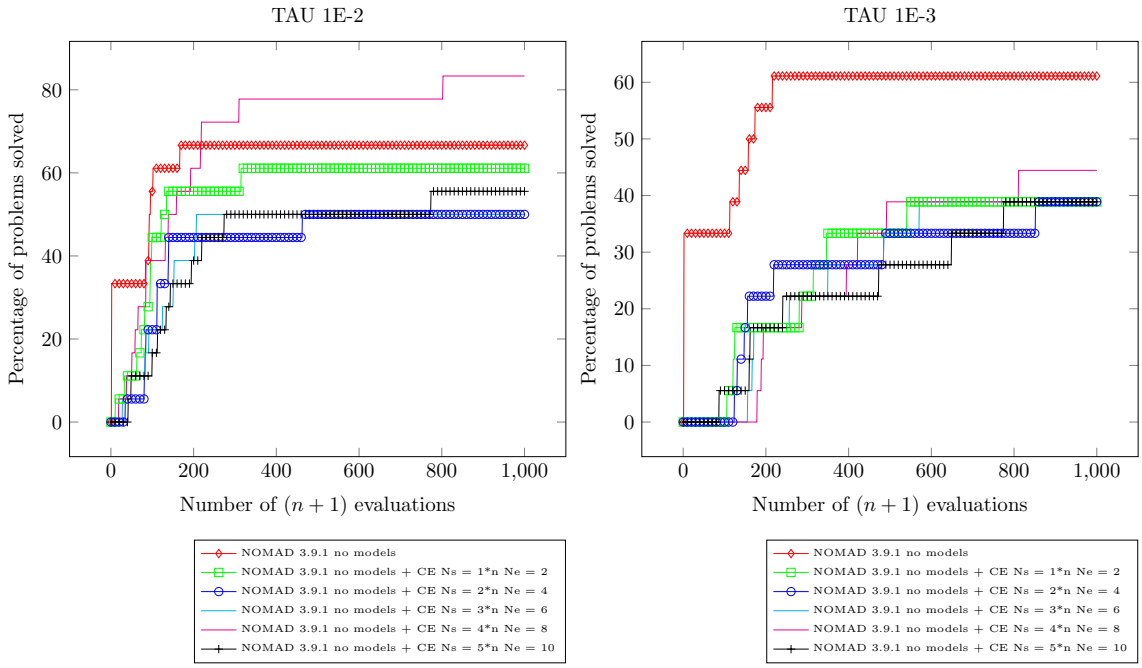

Figure 4.6 Result of calibration of the hyper-parameters N_e and N_s of CE-MADS on the 6 large test problems

problems. The framework used to compare the CE-Mads algorithm with others global optimization method is pymoo [35]. This framework proposes a variety of global optimization algorithm. The CE-Mads method is compared to four of them :

- Genetic Algorithm [75], a method based on biological inspired operators such as mutation, cross-over and selection. No special advice on the hyper-parameter are given in the pymoo framework, however after few tests, it seems that a population size of 40 performs well. We run the different tests with this value.

- Differential Evolution [103], a method which combines evolutionary strategies with geometrical search techniques. In the pymoo framework, the authors advise to test with the following settings: the crossover constant CR = 0.9, the select weighting factor F = 0.8 and the method is “DE/rand/1/bin". The size of the population is set to 20 which seems to perform well.

- Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) [78], a method based also on biological inspired operators. Its name comes from the adaptation of the covariance matrix of the multivariate normal distribution used during the mutation. The setting used for the tests are the default setting.

- Particle Swarm Optimization [89], a method inspired by the birds movement and more generally on the collaboration between the individuals. No indication are given in the pymoo

framework but it seems that a population size of 15 performs well.

To compare the different algorithms, we use the test problem common between the global optimization benchmark problems of pymoo and the problems provided in the appendix 4.8. That gives 19 unconstrained test problems (marked in with an asterisk in the appendix) and we run the algorithms with five different seeds in order to reduce the effect of randomness. The maximum number of function evaluations is fixed to 3000. Results are provided on Figure 4.7. The MADS-

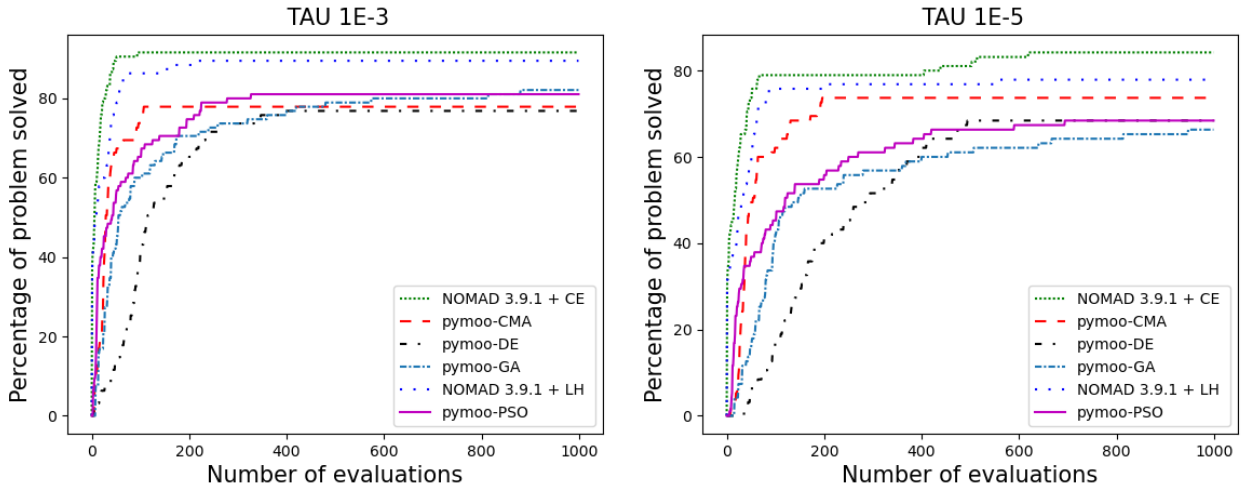

Figure 4.7 Result on the 19 global optimization test problems between CE-Mads, LH-Mads, GA, DE, CMA-ES and PSO for $\tau = 10^{-3}$ (left) and $\tau = 10^{-5}$ (right).

type algorithms are more efficient than the heuristic of global optimization on this test set. The heuristic's performances are comparable when the required accuracy is 10^{-3}. If a higher accuracy is desired, it would seem that CMA-ES is the more appropriate method. The gap between heuristics and MADS-type algorithms seems to widen when greater precision is required. This is normal considering that MADS is a local search algorithm originally. The interest of CE-MADS stands out since it allows to combine both a global and a local search, which explains its better performance.

4.6.3 Test on engineering problems

In this section, the CE-Mads algorithm is tested on three different engineering problems and compared to three algorithms: the Mads-default (without models), the VNS-Mads where a VNS-SEARCH is used and the LH-Mads which is a default Mads with in addition a LHS search. The comparison with the two last algorithms is crucial because they are methods aiming to explore the space of design variables. The Latin Hypercube SEARCH strategy is used with two parameters $n_{init} = 100$ and $n_{iter} = 10$: n_{init} is the number of LH trial points generated at the first iteration

of Mads and n_{iter} the number of LH trial points generated at each subsequent iteration. The Variable Neighbour Search is used with the default parameters [10]. It is an metaheuristic allowing to explore distant neighborhoods of the current incumbent solution.

4.6.3.1 The MDO problem

The Mads-default (no models), CE-Mads, VNS-Mads and LH-Mads are tested to solve a simple multidisciplinary wing design optimization problem [73]. Each initial point defines a MDO problem. Solving the problem consists in maximizing the range of an aircraft subject to 10 general constraints. The problem has 10 scaled design variables bounded in $[0; 100]$. Figure 4.8 shows the result on a data profile when solving 20 MDO problems on different initial points using 3000 function evaluations or less. The initial points are real randomly selected within the bounds. Each run is done with three different seeds in order to minimize the impact of the seed.

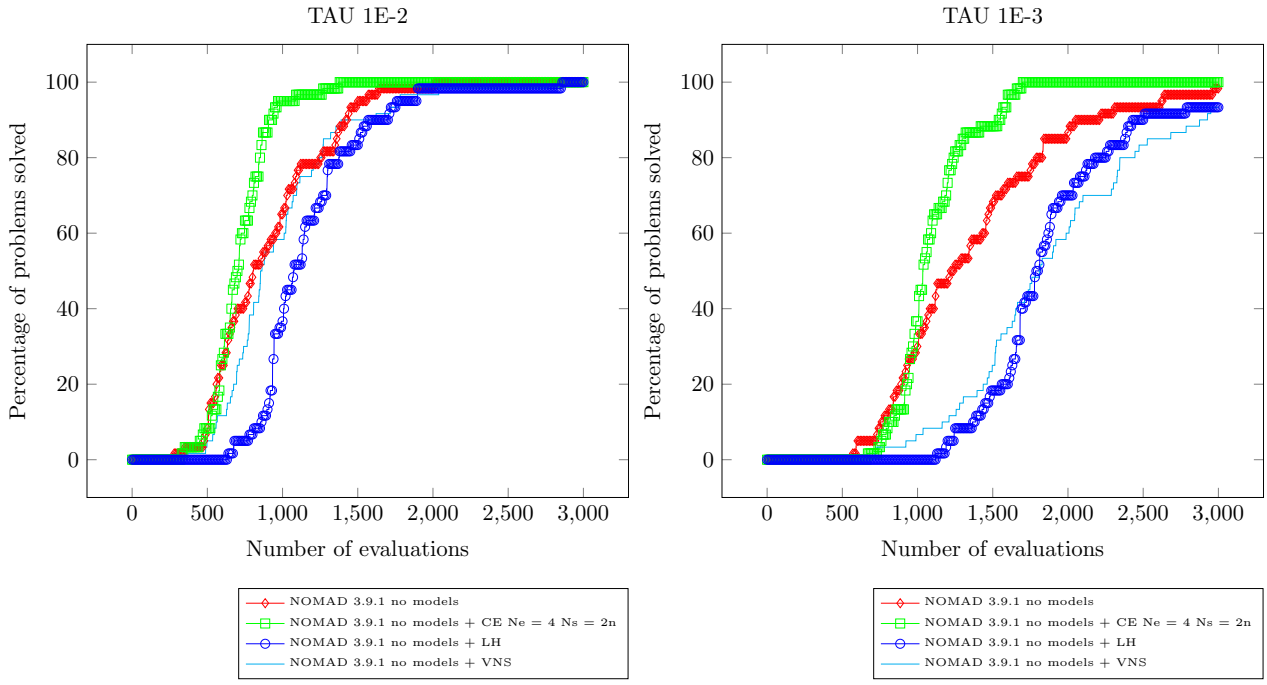

Figure 4.8 Result on the 60 MDO instances between Mads (no models), CE-Mads, VNS-Mads and LH-Mads for $\tau = 10^{-2}$ (left) and $\tau = 10^{-3}$ (right).

Figure 4.8 shows that the CE-Mads outperforms the other algorithms for all values of τ. Given that the computational time of engineering test problem is relatively low, the comparison between the heuristics and the MADS-type algorithm can be done. The inequality constraints are handled by the default setting in pymoo which is a penalization method. Given that Mads and VNS-Mads require a starting point to be run which can impact their performance, they are not used in this comparison.

Therefore, we do not use any starting point but the different algorithms are run with 20 different seeds. The result are given on the figure 4.9. The heuristics perform poorly compared to the Mads-type algorithm. That is not surprising given that Mads benefits of a specialized method to handle the constraints and is specialized in blackbox optimization. The tests on the other engineering test problems are not presented given the great computational time required and the even harder optimization of the blackbox.

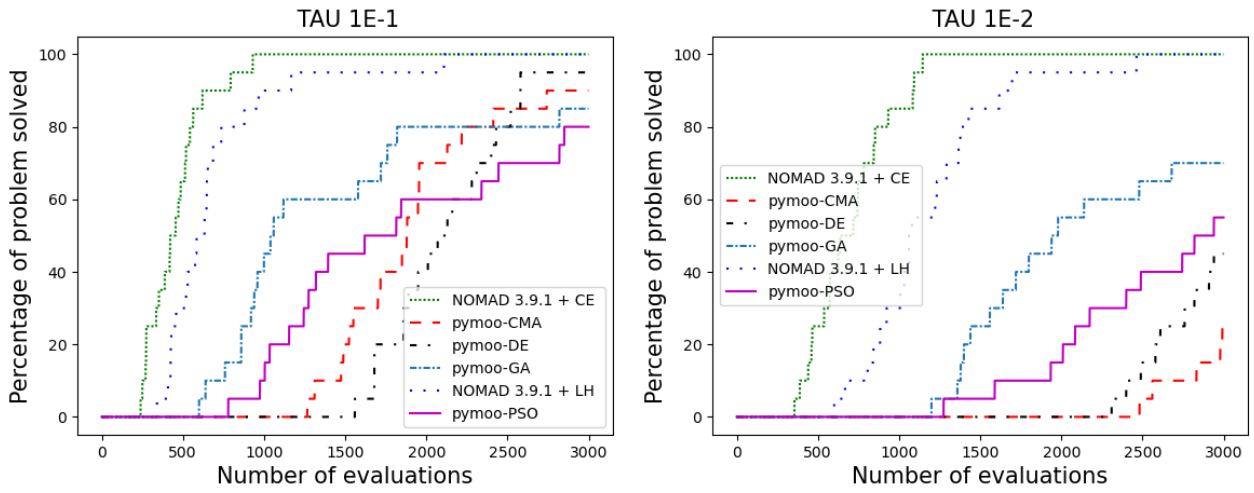

Figure 4.9 Result on the 20 MDO instances between CE-Mads, LH-Mads, GA, DE, CMA-ES and PSO for $\tau = 10^{-1}$ (left) and $\tau = 10^{-2}$ (right).

4.6.3.2 The STYRENE problem

The Mads-default (no models), CE-Mads, VNS-Mads and LH-Mads algorithms are tested to optimize a styrene production process [10], called STYRENE. This problem is a simulation of a chemical process. This process relies on a series of interdependent calculation of blocks using common numerical tools as Runge-Kutta, Newton, fixed point and also chemical related solver. The particularity of this problem is the presence of “hidden" constraints, i.e. sometimes the process does not finish and just return an error. In the case where the chemical process ends, the constraints (not hidden) and the objective functions may be evaluated during a post-processing. The objective is to maximize the net value of the styrene production process with 9 industrial and environmental regulations constraints. In this work, a STYRENE problem possesses eight independent variables influencing the styrene production process. The variables considered during the optimization process are all scaled and bounded in $X = [0, 100]^8$. As it was done for the MDO test problems, the four algorithms are tested with 20 different starting points taken in $\mathcal{X}$. A maximal number of evaluations of 3000 is used and each problem is run with three different seeds. The

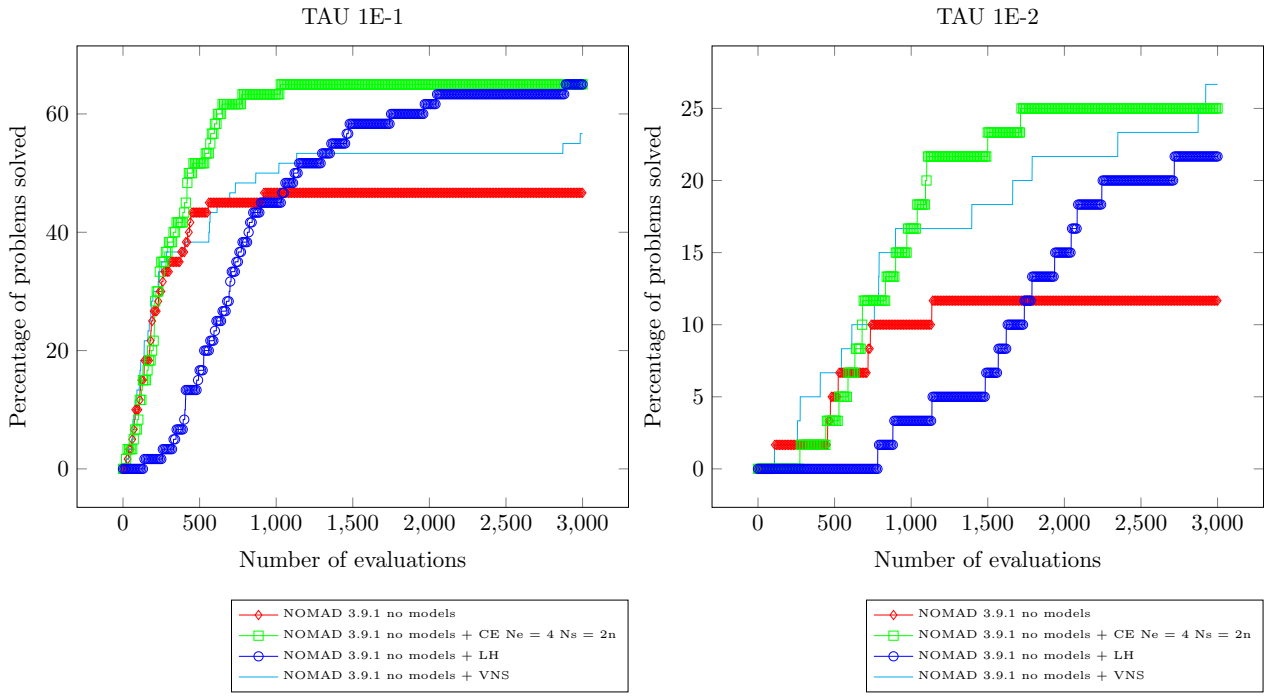

Figure 4.10 Result on the 60 STYRENE instances between Mads (no models), CE-Mads and LH-Mads for $\tau = 10^{-1}$ (left) and $\tau = 10^{-2}$ (right).

STYRENE problems is particularly interesting in this study, because there are two minima as it is shown in [20]. The results with $\tau = 10^{-1}$ allow to know the percentage of problems having found the global minimum. The results are provided on Figure 4.10. On the left plot, it is interesting to notice that the CE-Mads algorithm find the global minimum the same number of times that the LH-Mads algorithm but is more efficient. On the right plot, the CE-Mads algorithm seems to have the same accuracy that the VNS-Mads algorithm and is slightly more efficient.

4.6.3.3 The LOCKWOOD problem

Finally, the Mads default (no models), LH-Mads and CE-Mads algorithms without quadratic models are tested to solve the basic version of a pump-and-treat groundwater remediation problem from Montana Lockwood Solvent Groundwater Plume Site [124], called LOCKWOOD. The problem has 6 design variables bounded in $X = [0, 20000]^6$ and 4 constraints. A particularity of this problem is that each simulation run take several seconds, so the maximum number of blackbox evaluations is set to 1500. The algorithms are started from 20 different randomly selected initial points in X and three different seeds are used as previously. The results are provided on Figure 4.11. In this problem, reach the feasible region is not easy. Here again, the results at $\tau = 10^{-1}$ allows to give an idea of the number of times the algorithm reach the feasible region. For instance,

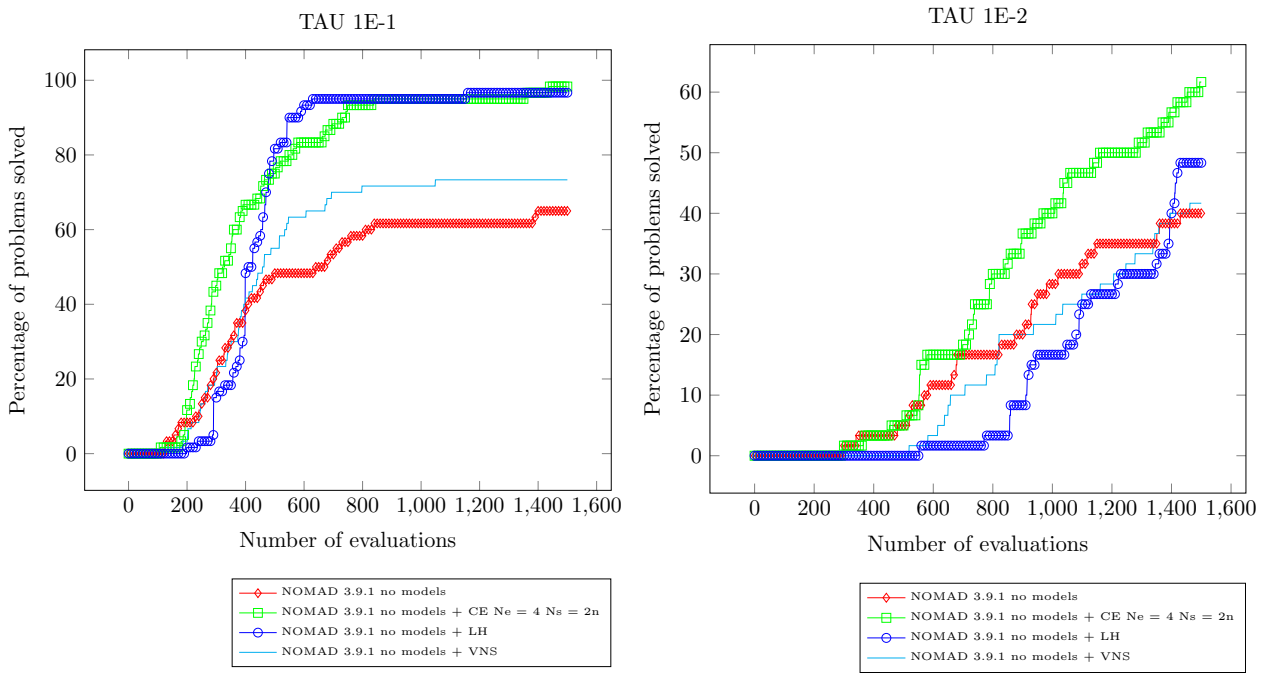

Figure 4.11 Result on the 20 LOCKWOOD instances between Mads-default (no models), CE-Mads, VNS-Mads and LH-Mads for $\tau = 10^{-1}$ (left) and $\tau = 10^{-2}$ (right).

CE-Mads and LH-Mads always reach the feasible region while Mads default reaches the feasible only 41 times on 60 instances and VNS-Mads only 46 times. The efficiency of CE-Mads and LH-Mads is comparable. However, on the right plot, a better accuracy is reached with a greater efficiency by the CE-Mads algorithm.

4.7 Discussion

This paper introduces a way to combine the CE algorithm and the Mads algorithm in order to allow a better space exploration. This is achieved by defining a CE-SEARCH step within the Mads algorithm. The CE-SEARCH generates some points according to a normal distribution whose mean and standard deviation is calculated from the best points stored in the cache. This approach allows to handle the constraints in a different way. Moreover, the particularity of this SEARCH is that it is not performed at each iteration of the Mads algorithm, but according to a criterion based on the value of the norm of the standard deviation of the best points.

Numerical experiments show that in the cases where the problem has different minima or a feasible region hard to reach, the CE-Mads algorithm performs well. Indeed, it attains as often as the LH-Mads the feasible region or the global minimum but it is far more efficient, especially when a tight accuracy is considered. Moreover, even on problem, as MDO, where the classical exploration

SEARCH, LH and VNS, do not work well, the CE-Mads algorithm gives interesting results.Finally, comparison with other global algorithms has been made, two conclusions are drawn. First, CE-Mads works better than the heuristics on the unconstrained global optimization test problems. That shows its real ability to escape local optima. Second, CE-Mads outperforms the heuristic on the engineering problems, which is not particularly relevant because it benefits to the Mads ability of performing well on this kind of problems.

Further works will be devoted to improve the link between the Mads algorithm and the CE algorithm by adjusting the size of the mesh with the standard deviation calculated in CE.

4.8 Appendix

Table 4.1 Description of the set of 100 analytical problems.

#	Name	Source	n	m	Bnds	#	Name	Source	n	m	Bnds
1	ARWHEAD10	[76]	10	0	no	51	MXHILB	[122]	50	0	no
2	ARWHEAD20	[76]	20	0	no	52	OPTENG_RBF	[96]	3	4	yes
3	BARD	[137]	3	0	no	53	OSBORNE1	[137]	5	0	no
4	BDQRTIC10	[76]	10	0	no	54	OSBORNE2	[122]	11	0	no
5	BDQRTIC20	[76]	20	0	no	55	PBC1	[122]	5	0	no
6	BEALE*	[137]	2	0	no	56	PENALTY1_4*	[76]	4	0	no
7	BIGGS	[76]	6	0	no	57	PENALTY1_10*	[76]	10	0	no
8	BOX	[137]	3	0	no	58	PENALTY1_20*	[76]	20	0	no
9	BRANIN*	[80]	2	0	yes	59	PENALTY2_4*	[76]	4	0	no
10	BROWNAL5	[76]	5	0	no	60	PENALTY2_10*	[76]	10	0	no
11	BROWNAL7	[76]	7	0	no	61	PENALTY2_20*	[76]	20	0	no
12	BROWNAL10	[76]	10	0	no	62	PENTAGON	[122]	6	15	no
13	BROWNAL20	[76]	20	0	no	63	PIGACHE_X00	[148]	4	11	yes
14	BROWNDENNIS	[137]	4	0	no	64	PIGACHE_X01	[148]	4	11	yes
15	BROWN_BS	[137]	2	0	no	65	POLAK2	[122]	10	0	no
16	B250	[37]	60	1	yes	66	POWELL_BS	[137]	2	0	no
17	B500	[37]	60	1	yes	67	POWELLSG4*	[76]	4	0	no
18	CHENWANG_F2_X0	[46]	8	6	yes	68	POWELLSG8	[76]	8	0	no
19	CHENWANG_F2_X1	[46]	8	6	yes	69	POWELLSG12	[76]	12	0	no
20	CHENWANG_F3_X0	[46]	10	8	yes	70	POWELLSG20	[76]	20	0	no
21	CHENWANG_F3_X1	[46]	10	8	yes	71	RADAR	[134]	7	0	yes
22	CRESCENT	[11]	10	2	no	72	RANA*	[84]	2	0	yes
23	DISK	[11]	10	1	no	73	RASTRIGIN*	[80]	2	0	yes
24	DIFFICULT2	[11]	10	0	no	74	RHEOLOGY	[14]	3	0	no
25	ELATTAR	[122]	6	0	no	75	ROSENBROCK*	[137]	2	0	yes
26	EVD61	[122]	6	0	no	76	SHOR	[122]	5	0	no
27	FILTER	[122]	9	0	no	77	SNAKE	[11]	2	2	no
28	FREUDENSTEINROTH*	[137]	2	0	no	78	SPRING_X00	[164]	3	4	yes
29	GAUSSIAN	[137]	3	0	no	79	SPRING_X01	[164]	3	4	yes
30	G2_10	[13]	10	2	yes	80	SROSENBR6	[76]	6	0	no
31	G2_20	[13]	20	2	yes	81	SROSENBR8	[76]	8	0	no
32	G2_50	[13]	50	2	yes	82	SROSENBR10	[76]	10	0	no
33	GOFFIN	[122]	50	0	no	83	SROSENBR20	[76]	20	0	no
34	GRIEWANK*	[80]	10	0	yes	84	TAOWANG_F2_X00	[182]	7	4	yes
35	GULFRD*	[19]	3	0	no	85	TAOWANG_F2_X01	[182]	7	4	yes
36	HELICALVALLEY*	[137]	3	0	no	86	TREFETHEN*	[84]	2	0	yes
37	HS19	[82]	2	2	yes	87	TRIDIA10	[76]	10	0	no
38	HS78	[122]	5	0	no	88	TRIDIA20	[76]	20	0	no
39	HS83_X0	[82]	5	6	yes	89	TRIGONOMETRIC	[137]	10	0	no
40	HS83_X1	[82]	5	6	yes	90	VARDIM8	[76]	8	0	no
41	HS114_X0	[122]	9	6	yes	91	VARDIM10	[76]	10	0	no
42	HS114_X1	[122]	9	6	yes	92	VARDIM20	[76]	20	0	no
43	JENNRICHSAMPSON	[137]	2	0	no	93	WANGWANG_F3	[187]	2	0	yes
44	KOWALIKOSBORNE*	[137]	4	0	no	94	WATSON9	[137]	9	0	no
45	L1HILB	[122]	50	0	no	95	WATSON12	[137]	12	0	yes
46	MAD6_X0	[122]	5	7	no	96	WONG1	[122]	7	0	no
47	MAD6_X1	[122]	5	7	no	97	WONG2	[122]	10	0	no
48	MCKINNON	[125]	2	0	no	98	WOODS4	[76]	4	0	no
49	MEYER*	[137]	3	0	no	99	WOODS12	[76]	12	0	no
50	MEZMONTES	[132]	2	2	yes	100	WOODS20	[76]	20	0	no

Nomenclature

The following list describes symbols used within the body of the document. In what follows, if the symbol is bold then it is a vector otherwise it is a scalar.

$\boldsymbol{\ell}$	The lower bound of a decision variable
Δ^k	The frame size parameter at iteration k
δ^k	The mesh size parameter at iteration k
ϵ	The stopping criterion
γ	A parameter to estimate in an associated stochastic problem
$\mathbf{u}$	The upper bound of a decision variable
$\mathcal{E}$	The expectation
$\mathcal{N}$	The normal distribution
$\mathcal{V}$	Set of parameters of a probability density function
$\mathcal{X}$	The bounded constraints set of type $\ell \leq x \leq u$
μ	The mean
Ω	The feasible set
ρ	A percentage of quantile
σ	The standard deviation
τ	The mesh size adjustment parameter
$\mathbf{X}$	A random vector
c_j	The j^{th} constraint
D	A positive spanning set
E^k	The set of indices of elite points
F^k	The frame at iteration k

$g(\cdot;\cdot)$	A probability density function
h	The measure of constraints violation
I_x	Indicator function of x
k	The iteration counter
M^k	The mesh at iteration k
n	The dimension of a problem
N_s	Number of sampled data at each iteration
N_e	Number of elite population
V	The cache
v	A parameter of a probability density function

CHAPTER 5 ARTICLE 2: SEQUENTIAL STOCHASTIC BLACKBOX OPTIMIZATION WITH ZEROTH-ORDER GRADIENT ESTIMATORS

Title Sequential stochastic blackbox optimization with zeroth-order gradient estimators.

Authors Charles Audet, Jean Bigeon, Romain Couderc and Michael Kokkolaras.

Journal Published in *AIMS mathematics* [22] on 08/09/2023.

5.1 Context

The second contribution of this thesis targets unconstrained stochastic optimization problems of the form given in Equation (2.12). Specifically, the uncertainties have an unknown distribution and are considered uncontrollable, i.e., two evaluations of the blackbox at the same point $\mathbf{x}$ can yield two different outputs (this is also called one-point bandit feedback in stochastic convex optimization [131]). As shown in Section 2.3.2, this problem is the subject of many work. Mathematically, it is particularly interesting because in the context of DFO, if the underlying function is continuously differentiable, convergence rates in mean to a stationary point of the problem can be derived. In practice, however, these results are of limited interest because they formally require many runs of the algorithm to be guaranteed [71]. A more promising result might be to obtain the convergence rate to a stationary point according to some probability. This type of result was developed in [71]. Although the convergence rate was obtained with probability one in [34, 145], the drawback of these results is that they are obtained in terms of iteration and not in terms of blackbox queries. Thus , that gives limited information about the practical interest of these methods.

This second contribution presents a new method capable of overcoming the aforementioned difficulties. This method is based on a Gaussian exploration of the space of design variables. However, in contrast to the first contribution, the Gaussian exploration allows smoothing the problem and obtaining estimates of the gradient of the smooth approximation. The gradient estimates require only two blackbox evaluations to be computed, regardless of the size of the problem [33, 71]. However, they require many more samples to be accurate [29]. Therefore, to avoid costly sampling at each iteration, the authors of [47] used instead an exponential moving average (also called momentum in the machine learning community [31]). This moving average of gradient estimates is used to approximate a descent direction, rather than using only the estimates at the current iteration. This allows the variance of the estimates to be smoothed over all iterations [31] and provides a more

efficient algorithm in terms of blackbox evaluations. In our work, inspired by [71] and more generally by the multi-timescale stochastic approximation [38, Chapter 6], the decay parameter of the moving average is gradually reduced to 0. This small change allows us to prove that the momentum vector eventually converges in mean to the gradient of the objective function. We also derive the rate of this convergence. This result is important because the norm of the momentum vector can be used as a stopping criterion, since its norm converges to 0 during the optimization process. Based on this result, a sequential optimization algorithm is developed in which each subproblem is a smooth approximation of the original problem and is solved using a momentum-based method. The stopping criterion is used to consider a subproblem to be approximately solved and to proceed to the next one. For a carefully chosen sequence of stopping criteria, we prove that the sequence of points satisfying these stopping criteria converges to a stationary point of the problem. The convergence rate is also analyzed.

5.2 Article

Abstract This work considers stochastic optimization problems in which the objective function values can only be computed by a blackbox corrupted by some random noise following an unknown distribution. The proposed method is based on sequential stochastic optimization (SSO), the original problem is decomposed into a sequence of subproblems. Each subproblem is solved using a zeroth-order version of a sign stochastic gradient descent with momentum algorithm (i.e, ZO-signum) and with increasingly fine precision. This decomposition allows a good exploration of the space while maintaining the efficiency of the algorithm once it gets close to the solution. Under the Lipschitz continuity assumption on the blackbox, a convergence rate in mean is derived for the ZO-signum algorithm. Moreover, if the blackbox is smooth and convex or locally convex around its minima, the rate of convergence to an ϵ-optimal point of the problem may be obtained for the SSO algorithm. Numerical experiments are conducted to compare the SSO algorithm with other state-of-the-art algorithms and to demonstrate its competitiveness.

Keywords: stochastic blackbox optimization; gradient approximation; sequential optimization; momentum-based method; convergence rate analysis

5.3 Introduction

The present work targets stochastic blackbox optimization problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) \qquad \text{where} \qquad f(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}\left[F(\mathbf{x},\boldsymbol{\xi})\right], \tag{5.1}$$

and $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a blackbox [14] that takes two inputs: a vector of design variables $\mathbf{x} \in \mathbb{R}^n$ and a vector $\boldsymbol{\xi} \in \mathbb{R}^m$ that represents uncertainties with unknown distributions. The function F is called a stochastic zeroth-order oracle [71]. The objective function f is obtained by taking the expectation of F over all possible values of the uncertainties $\boldsymbol{\xi}$. This optimization problem can be found in two different fields. The first is in a machine learning framework wherein the loss function's gradient is unavailable or difficult to compute, such as in the optimization of neural network architecture [155], design of adversarial attacks [47] or game content generation [185]. The second field is when the function F is evaluated by means of a computational procedure [97]. In many cases, it depends on an uncertainty vector $\boldsymbol{\xi}$ due to environmental conditions, costs or effects of repair actions that are unknown [160]. Another source of uncertainty appears when the optimization is conducted at the early stages of the design process, where knowledge, information and data are very limited.

5.3.1 Related work

Stochastic derivative-free optimization has been the subject of research for many years. Traditional derivative-free methods may be divided into two categories [54]: direct search and model-based methods. Algorithms corresponding to both methods have been adapted to a stochastic ZO oracle. Examples include the stochastic Nelder-Mead algorithm [43] and the stochastic versions of the mesh adaptive direct search algorithm [15, 23] for the direct search methods. For model-based methods, most studies consider extensions of the trust region method [45, 56, 123]. A major shortcoming of these methods is their difficulty to scale to large problems.

Recently, another class of methods, named ZO methods, has been attracting increasing amounts of attention. These methods use stochastic gradient estimators, which are based on the seminal work in [92, 156], and they have been extended in [71, 141, 166, 175]. These estimators have the appealing property of being able to estimate the gradient with only one or two function evaluations, regardless of the problem size. ZO methods take advantage of this property to extend first-order methods. For instance, the well known first-order methods conditional gradient, sign stochastic gradient descent (signSGD) [31] and adaptive momentum (ADAM) [94] have been extended to ZSCG [25], ZO-signSGD [116] and ZO-adaMM [47], respectively. More methods, not only based on first-order algorithms, have also emerged to solve regularized optimization problems [41], for very high dimensional blackbox optimization problems [39] and stochastic composition optimization problems [72]. Methods using second-order information based limited function queries have been developed [93]. Some methods handle situations in which the optimizer has only access to a comparison oracle that indicates which of two points has the highest value [40]. For an overview on ZO methods, readers may consult [117].

5.3.2 Motivation

Formally, stochastic gradient estimators involve a smooth approximation f^β (see Chapter 7.6 in [166]) which is a convolution product between f and a kernel $h^\beta(\mathbf{u})$

$$f^\beta(\mathbf{x}) := \int_{-\infty}^{\infty} h^\beta(\mathbf{u}) f(\mathbf{x}-\mathbf{u}) d\mathbf{u} = \int_{-\infty}^{\infty} h^\beta(\mathbf{x}-\mathbf{u}) f(\mathbf{u}) d\mathbf{u}. \tag{5.2}$$

The kernel must fulfill a set of conditions[166, pp. 263]:

(1) $h^\beta(\mathbf{u}) = \frac{1}{\beta^n} h(\frac{\mathbf{u}}{\beta})$ is a piecewise differentiable function;

(2) $\lim_{\beta\to 0} h^\beta(\mathbf{u}) = \delta(\mathbf{u})$, where $\delta(v)$ is Dirac's delta function;

(3) $\lim_{\beta\to 0} f^\beta(\mathbf{x}) = f(\mathbf{x})$ if $\mathbf{x}$ is a point of continuity of f;

(4) The kernel $h^\beta(\mathbf{u})$ is a probability density function, that is $f^\beta(\mathbf{x}) = \mathbb{E}_{\mathbf{U}\sim h^\beta(\mathbf{u})}[f(\mathbf{x}-\mathbf{U})] = \mathbb{E}_{\mathbf{U}\sim h(\mathbf{u})}[f(\mathbf{x}-\beta U)]$.

Frequently used kernels include the Gaussian distribution and the uniform distribution on a unit ball. Three properties of the smooth approximation are worth noting. First, the smooth approximation may be interpreted as a local weighted average of the function values in the neighborhood of $\mathbf{x}$. Condition 5.3.2 implies that it is possible to obtain a solution that is arbitrarily close to a local minimum f^*. Second, the smooth approximation is infinitely differentiable as a consequence of the convolution product, regardless of the degree of smoothness of f. Moreover, according to the chosen kernel, stochastic gradient estimators may be calculated. These estimators are unbiased estimators of ∇f^β and may be constructed on the basis of observations of $F(\mathbf{x}, \boldsymbol{\xi})$ alone. Finally, the smooth approximation allows convexification of the original function f. Previous studies [166, 178] show that greater values of β result in better convexification, as illustrated in Figure 5.1. Additionally, a larger β leads to greater exploration of the space during the calculation of the gradient estimator. It has also been demonstrated in [118] that if the smoothing parameter is too small, the difference in function values cannot be used to accurately represent the function differential, particularly when the noise level is significant.

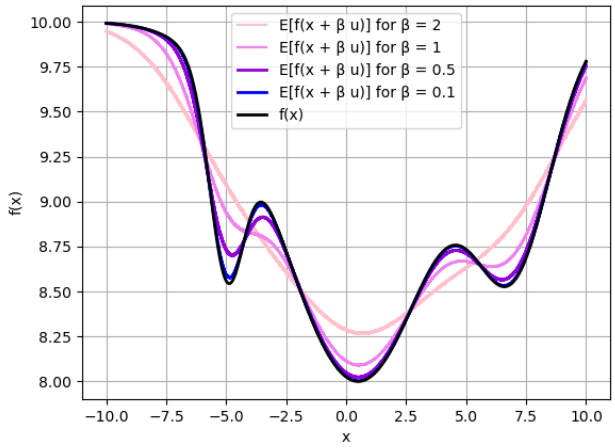

Figure 5.1 Curves of f^β for $u \sim \mathcal{N}(0,1)$ and different values of β.

Although the two first properties of the smooth approximation are exploited by ZO methods, the last property has not been utilized since the work in [178]. This may be because the convexification phenomenon becomes insignificant when dealing with high-dimensional problems [1]. However, for problems of relatively small size ($n \simeq 10$), this property can be useful. The authors of [178] use

[1]Note that a blackbox optimization problem with dimensions ranging from 100 to 1000 may be considered large, while problems with $n \geq 10000$ may be considered very large.

an iterative algorithm to minimize the sequence of subproblems

$$\min_{\mathbf{x}\in\mathbb{R}^n} f^{\beta^i}(\mathbf{x}), \tag{5.3}$$

where β^i belongs to a finite prescaled sequence of scalars. This approach is limited because the sequence β^i does not necessarily converge to 0 and the number of iterations to go from subproblem i to $i+1$ is arbitrarily fixed a priori. Furthermore, neither a convergence proof nor a convergence rate are provided for the algorithm. Finally, although promising, numerical results are only presented for analytical test problems. These shortcomings motivate the research presented here.

5.3.3 Contributions

The main contributions of this paper can be summarized as follows:

- A sequential stochastic optimization (SSO) algorithm is developed to solve the sequence of subproblems in Eq (5.3). In the inner loop, a subproblem is solved according to the ZO version of the signum algorithm [31]. The stopping criterion is based on the norm of the momentum, which must be below a certain threshold. In the outer loop, the sequence of β^i is proportional to the threshold needed to consider a subproblem solved, and it is driven to 0. Therefore, the smaller the value of β^i (and thus better the approximation given by f^{β^i}), the larger the computational budget allotted for the resolution of the subproblem.

- A theoretical analysis of this algorithm is conducted. First, the expectation of the norm of the momentum is proved to converge to 0, with a convergence rate that depends on the step sizes. Then, the convergence rate in mean of the ZO-signum algorithm toward a stationary point of f^β is derived under Lipschitz continuity of the function F. Finally, if the function F is smooth and f^β is convex or becomes convex around its local minima, the rate of convergence to an ϵ-optimal point is derived for the SSO algorithm.

- Numerical experiments were conducted to evaluate the performance of the proposed algorithm for two applications. First, a comparison is made with traditional derivative-free algorithms in terms of the optimization of the storage cost of a solar thermal power plant model, which is a low-dimensional problem. Second, a comparison is made with other ZO algorithms in order to generate blackbox adversarial attacks, which are large-sized problems.

The remainder of this paper is organized as follows. In Section 5.4, the main assumptions and the Gaussian gradient estimator are described. In Section 5.5, the sequential optimization algorithm is presented, and its convergence properties are studied in Section 5.6. Section 5.7 presents numerical results, and Section 5.8 draws conclusions and discusses future work.

5.4 Gaussian gradient estimator

The assumptions concerning the stochastic blackbox function F are as follows:

Assumption 1. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space.*

(a) The function satisfies $F(\cdot, \boldsymbol{\xi}) \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ and $f(\mathbf{x}) := \mathbb{E}_{\boldsymbol{\xi}}[F(\mathbf{x}, \boldsymbol{\xi})]$ for all $\mathbf{x} \in \mathbb{R}^n$.

(b) $F(\cdot, \boldsymbol{\xi})$ is Lipschitz-continuous for any fixed value of $\boldsymbol{\xi} = (\boldsymbol{\xi}^1, \boldsymbol{\xi}^2)$, with the constant $L_0(F) > 0$, that is

$$|F(\mathbf{x}, \boldsymbol{\xi}^1) - F(\mathbf{y}, \boldsymbol{\xi}^2)| \leq L_0(F)||\mathbf{x} - \mathbf{y}||.$$

Assumption 1(a) implies that the expectation of $F(\mathbf{x}, \boldsymbol{\xi})$ with respect to $\boldsymbol{\xi}$ is well defined on $\mathbb{R}^n$ and that the estimator $F(\mathbf{x}, \boldsymbol{\xi})$ is unbiased. Assumption 1(b) is commonly used to ensure convergence and bound the variance of the stochastic ZO oracle. It is worth noticing that no assumption is made regarding the differentiability of the objective function f or of its estimate F with respect to $\mathbf{x}$, contrary to most work on ZO methods.

Under Assumption 1, a smooth approximation of the function f may be constructed via its convolution with a Gaussian random vector. Let $\mathbf{u}$ be an n-dimensional standard Gaussian random vector and $\beta > 0$ be the smoothing parameter. Then, a smooth approximation of f is defined as

$$f^\beta(\mathbf{x}) := \frac{1}{(2\pi)^{\frac{n}{2}}} \int f(\mathbf{x} + \beta\mathbf{u}) e^{-\frac{||u||^2}{2}} d\mathbf{u} = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x} + \beta\mathbf{u})]. \tag{5.4}$$

This estimator has been studied in the literature (especially in [141]); it has the benefits of several appealing properties. The properties used in this work are summarized in the following lemma:

Lemma 5.4.1. *Under Assumption 1, the following statements hold for any integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and its approximation f^β parameterized by $\beta > 0$.*

(1) f^β is infinitely differentiable: $f^\beta \in \mathcal{C}^\infty$.

(2) A one-sided unbiased estimator of ∇f^β is

$$\tilde{\nabla} f^\beta(\mathbf{x}) := \frac{\mathbf{u}\,(f(\mathbf{x} + \beta\mathbf{u}) - f(\mathbf{x}))}{\beta}. \tag{5.5}$$

(3) Let $\beta^2 \geq \beta^1 \geq 0$; then, $\forall \mathbf{x} \in \mathbb{R}^n$

$$||\nabla f^{\beta^1}(\mathbf{x}) - \nabla f^{\beta^2}(\mathbf{x})|| \leq L_1(f^{\beta^1})\,(\beta^2 - \beta^1)\,(n+3)^{\frac{3}{2}}.$$

Moreover, for $\beta > 0$, f^β is $L_1(f^\beta)$-smooth, i.e., $f^\beta \in \mathcal{C}^{1+}$ with $L_1(f^\beta) = \frac{2\sqrt{n}}{\beta} L_0(F)$.

(4) If f is convex, then f^β is also convex.

Proof. (1) It is a consequence of the convolution product between an integrable function and an infinitely differentiable kernel.

(2) See [141, Eq (22)].

(3) If $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, define the following for all $\mathbf{x} \in \mathbb{R}^n$

$$g(\mathbf{x}) = f^{\beta^1}(\mathbf{x}) = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x} + \beta^1 \mathbf{u})].$$

Let $\mu = \beta^2 - \beta^1 \geq 0$; it follows that for all $\mathbf{x} \in \mathbb{R}^n$

$$g^\mu(\mathbf{x}) = \mathbb{E}_{\mathbf{u}}[g(\mathbf{x}+\mu\mathbf{u})] = \mathbb{E}_{\mathbf{u}}[f^{\beta^1}(\mathbf{x}+\mu\mathbf{u})] = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x}+\mu\mathbf{u}+\beta^1\mathbf{u})] = \mathbb{E}_{\mathbf{u}}[f(\mathbf{x}+\beta^2\mathbf{u})] = f^{\beta^2}(\mathbf{x}).$$

Then, since by [141, Lemma 2] under Assumption 1, f^{β^1} is Lipschitz continuously differentiable, [141, Lemma 3] may be applied to the function g and it follows that

$$||\nabla f^{\beta^1}(\mathbf{x}) - \nabla f^{\beta^2}(\mathbf{x})|| = ||\nabla g(\mathbf{x}) - \nabla g^\mu(\mathbf{x})|| \leq L_1(f^{\beta^1})\mu\,(n+3)^{\frac{3}{2}} = L_1(f^{\beta^1})\,(\beta^2 - \beta^1)\,(n+3)^{\frac{3}{2}}.$$

(4) See [141, pp. 5]. □

The estimator obtained in Eq (5.5) may be adapted to the stochastic ZO oracle F. For instance, a one-sided (mini-batch) estimator of the noised function F is

$$\tilde{\nabla} f^\beta(\mathbf{x}, \boldsymbol{\xi}) = \frac{1}{q} \sum_{j=1}^{q} \frac{\mathbf{u}^j \,(F(\mathbf{x} + \beta \mathbf{u}^j, \boldsymbol{\xi}^j) - F(\mathbf{x}, \boldsymbol{\xi}^0))}{\beta}, \tag{5.6}$$

where $\{u^j\}_{j=1}^q$ and $\{\boldsymbol{\xi}^j\}_{j=0}^q$ are q Gaussian random directional vectors and their associated q estimate values of the function F, respectively. This is still an unbiased estimator of ∇f^β because

$$\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}}[\tilde{\nabla} f^\beta(\mathbf{x}, \boldsymbol{\xi})] = \mathbb{E}_{\mathbf{u}}[\mathbb{E}_{\boldsymbol{\xi}}[\tilde{\nabla} f^\beta(\mathbf{x}, \boldsymbol{\xi})|\mathbf{u}]] = \nabla f^\beta(\mathbf{x}). \tag{5.7}$$

The result of Lemma 5.4.1(3) is essential to understand why solving a sequence of optimization problems defined by Eq (5.3) may be efficient, although it might seem counterproductive at first sight. Below are examples of the advantages of treating the problem with sequential smoothed function optimization.

- The subproblems are approximations of the original problem and it is not necessary to solve them exactly. Thus, an appropriate procedure for solving these problems with increasingly fine precision can be used. Moreover, as seen in Lemma 5.4.1(3), the norm of the gradient obtained in a subproblem is close to the one of the following subproblem. The computational effort to find a solution to the second subproblem from the solution of the first should therefore not be important.

- The information collected during the optimization process for a subproblem may be reused in the subsequent subproblems since they are similar.

- A specific interest in the case of smooth approximation is the ability of using a larger value of β to solve the first subproblems. It allows for a better exploration of the space and convexification phenomenon of the function (see Figure 5.1). Moreover, the new step size may be used for each subproblem; it allows for an increase in the step size momentarily, in the hope of having a greater chance of escaping a local minimum.

5.5 SSO algorithm

Section 3.1 presents a ZO version of the signum algorithm [31] to solve Subproblem (5.3) for a given β^i and Section 3.2 presents the complete algorithm used to solve the sequential optimization problem.

5.5.1 The ZO signum algorithm

A ZO version of the signum algorithm (Algorithm 2 of [31]) is used to solve the subproblems. The signum algorithm is a momentum version of the sign-SGD algorithm. In [116], the authors extended the original sign-SGD algorithm to a ZO version of this algorithm. However, a ZO version of signum is not studied in the work of [116]. As the signum algorithm has been shown to be competitive with the ADAM algorithm [31], a ZO version of this algorithm seems interesting to consider. For completeness, the versions of the sign-SGD and the signum algorithms as they originally appeared in [31] are given in Section 5.9.3. There is an important difference between the original signum algorithm and its ZO version presented in Algorithm 11. Indeed, while the step size of the momentum $1 - s_2$ is kept constant in the work of [31], it is driven to 0 in our work.

Algorithm 11 ZO-signum algorithm to solve subproblem $i \in \mathbb{N}$.

Input: $\mathbf{x}^{i,0}, \mathbf{m}^{i,0}, \beta^i, s_1^{i,0}, s_2^{i,0}, L, q, M$
Set $k = 0$
Define step-size sequences $s_1^{i,k} = \frac{s_1^{i,0}}{(k+1)^{\alpha_1}}$ and $s_2^{i,k} = \frac{s_2^{i,0}}{(k+1)^{\alpha_2}}$
while $||\mathbf{m}^{i,k}|| > \frac{L\beta^i}{4\beta^0}$ or $k \leq M$ **do**
    Draw q samples $\mathbf{u}^k$ from the Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$
    Calculate the average of the q Gaussian estimate $\tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,k}, \boldsymbol{\xi}^{i,k})$ from Eq (5.6)
    Update:

$$\mathbf{m}^{i,k+1} = s_2^{i,k}\tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,k}, \boldsymbol{\xi}^k) + (1 - s_2^{i,k})\mathbf{m}^{i,k} \tag{5.8}$$

$$x_j^{i,k+1} = x_j^{i,k} - s_1^k \text{sign}(m_j^{i,k+1}) \; \forall j \in [1, n] \tag{5.9}$$

    $k \leftarrow k + 1$
end while
Return $\mathbf{m}^{i,k}$ and $\mathbf{x}^{i,k}$

This leads to two consequences. First, the variance is reduced since the gradient is averaged on a longer time horizon, without using mini-batch sampling. Second, as it has been demonstrated in other stochastic approximation works ([33, Section 3.3] and [169]), with carefully chosen step sizes the norm of the momentum goes to 0 with probability one. In the ZO-signum algorithm, the norm of the momentum is thus used as a stopping criterion.

5.5.2 The SSO algorithm

The optimization of the subproblem sequence described in Eq (5.3) is driven by the SSO algorithm presented in Algorithm 12. The value of β plays a critical role, as it serves as both the smoothing parameter and the stopping criterion for Algorithms 11 and 12. Algorithm 12 is inspired by the MADS algorithm [12] as it is based on two steps: a search step and a local step. The search step is optional, may use any heuristics and is required only for problems with relatively small dimensions. In Algorithm 12, an example of a search is given; it consists of updating $\mathbf{x}$ after M iterations of the ZO-signum algorithm with the best known $\mathbf{x}$ found so far. The local step is then used: Algorithm 11 is launched for each subproblem i with specific values of β^i and step-size sequences. Once Algorithm 11 meets the stopping criterion (which depends on the value of β^i), the value of β^i and the initial step-sizes $s_1^{i,0}$ and $s_2^{i,0}$ are reduced, and the algorithm proceeds to the next subproblem. The convergence is guaranteed by the local step, since the search step is run only a finite number of times.

Algorithm 12 SSO algorithm.

1: **Initialization:**
2: Set $\mathbf{x}^{0,0} \in \mathbb{R}^n$, $\beta^0 > 0$ and N as the maximum number of function calls for the search step
3: Set q as the number of gradient estimates at each iteration of ZO-signum (ZOS) algorithm
4: Set M the minimum number of iterations made by the ZOS algorithm on a subproblem
5: $\mathcal{C}$ is the cache containing all of the evaluated points
6: Set $\mathbf{m}^{0,0} = \tilde{\nabla} f^{\beta^0}(\mathbf{x}^{0,0}, \boldsymbol{\xi}^0)$ and $L = +\infty$
7: Set $s_1^{0,0} > 0$ and $s_2^{0,0} > 0$
8: Set $i = 0$
9: **Search step (optional):**
10: **while** $M(i+1)q \leq N$: **do**
11: Solve subproblem i with Algorithm 11:

$$\mathbf{m}^{i+1,0} = ZOS(\mathbf{x}^{i,0}, \mathbf{m}^{i,0}, \beta^i, s_1^{i,0}, s_2^{i,0}, L, q, M)$$
$$\mathbf{x}^{i+1,0} \in \operatorname*{argmin}_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x}, \boldsymbol{\xi})$$

12: Update β^i, $s_1^{i,0}$ and $s_2^{i,0}$ as in Step 18
13: **end while**
14: $L = ||\mathbf{m}^{0,0}||$
15: **Local step:**
16: **while** $\beta^i > \epsilon$ **do**
17: Solve subproblem i with Algorithm 11:

$$\mathbf{m}^{i+1,0}, \mathbf{x}^{i+1,0} = ZOS(\mathbf{x}^{i,0}, \mathbf{m}^{i,0}, \beta^i, s_1^{i,0}, s_2^{i,0}, L, q, M)$$

18: Update:

$$\beta^i = \frac{\beta^0}{(i+1)^2}, s_1^{i,0} = \frac{s_1^{0,0}}{(i+1)^{\frac{3}{2}}}, \quad s_2^{i,0} = \frac{s_2^{0,0}}{i+1}$$
$$i \leftarrow i + 1$$

19: **end while**
20: Return $\mathbf{x}^i$

It is worth noting that the decreasing rate of β^i is chosen so that the difference between subproblems i and $i+1$ is not significant. Therefore, the information collected in subproblem i, through the momentum vector $\mathbf{m}$, can be used in subproblem $i+1$. Furthermore, the initial step sizes $s_1^{i,0}$ and

$s_2^{i,0}$ decrease with each iteration, allowing us to focus our efforts quickly toward a local optimum when $s_1^{0,0}$ and β^0 are chosen to be relatively large.

5.6 Convergence analysis

The convergence analysis is conducted in two steps: first the convergence rate in mean is derived for Algorithm 11 and then the rate of convergence to an ϵ-optimal point is derived for Algorithm 12.

5.6.1 Convergence rate of the ZO-signum algorithm

The analysis of Algorithm 11 follows the general methodology given in [31, Appendix E]. In the following subsection, the main result in [31] is recalled for completeness. The next subsections are devoted to bound the variance and bias terms when $\lim_{k\to\infty} s_2^{i,k} = 0$. Finally, these results are used to obtain the convergence rate in mean of Algorithm 11 in the non-convex and convex case. The last subsection is devoted to a theoretical comparison with other ZO methods of the literature. The subproblem index i is kept constant throughout this section. In order to better convey the convergence analysis of the ZO-signum algorithm, a hierarchical workflow of the different theoretical results is presented in Table 5.1. The main results are presented in Theorem 5.6.9 and its corollary for the non-convex case, and Theorem 5.6.11 for the convex case.

Table 5.1 Workflow of lemmas/propositions/theorems for the ZO-signum convergence analysis.

Assumptions on F	Preliminary results	Intermediate results	Main results	When f^β is convex
Assumption 1 which implies that $L_1(f^{\beta^i}) = \frac{2\sqrt{n}L_0(F)}{\beta^i}$		Proposition 5.6.1	Theorem 5.6.9 Corollary 5.6.10	Theorem 5.6.11
	Lemma 5.6.2 Lemma 5.6.3 Lemma 5.6.4	Proposition 5.6.5		
	Lemma 5.6.6 Lemma 5.6.7	Proposition 5.6.8		

5.6.1.1 Preliminary result [31]

The following proposition uses the Lipschitz continuity of the function f^{β^i} (proved in Lemma 5.4.1) to bound the gradient at the kth iteration.

Proposition 5.6.1. *[31] For the subproblem $i \in \mathbb{N}$, under Assumption 1 and in the setting of*

Algorithm 11, we have

$$s_1^{i,k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,k+1})] + \frac{nL_1(f^{\beta^i})}{2}(s_1^{i,k})^2 + 2s_1^{i,k}\underbrace{\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1]}_{bias} + 2s_1^{i,k}\sqrt{n}\sqrt{\underbrace{\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||_2^2]}_{variance}}, \tag{5.10}$$

where $\bar{m}_j^{i,k+1}$ is defined recursively as $\bar{m}_j^{i,k+1} = s_2^{i,k}\nabla f^{\beta^i}(\mathbf{x}^{i,k}) + (1 - s_2^{i,k})\bar{m}_j^{i,k}$.

Proof. See Appendix B. □

Now, it remains to bound the three terms on the right side of Inequality (5.10).

5.6.1.2 Bound on the variance term

The three following lemmas are consecrated to bound the variance term. Unlike the work reported in [31], the variance reduction is conducted by driving the step size of the momentum to 0. It avoids the need to sample an increasing number of stochastic gradients at each iteration, which may be problematic, as noted in [116]. To achieve this, the variance term is first decomposed in terms of expectation of the squared norm of the stochastic gradient estimators $\tilde{g}$.

Lemma 5.6.2. *For the subproblem $i \in \mathbb{N}$, let $k \in \mathbb{N}$ and $j \in [1, n]$; we have*

$$\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2] \leq (s_2^{i,k})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,k}||^2] + \sum_{r=0}^{k-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1 - s_2^{i,t+1})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2] + \prod_{t=0}^{k}(1 - s_2^{i,t})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,0}||^2],$$

where $\tilde{g}_j^{i,r} = \tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,r}, \boldsymbol{\xi}^r), \forall r \in [0, k]$ is defined in Eq (5.6) and the norm is $||\cdot||_2$.

Proof. Let $k \in \mathbb{N}$; by definition of $\mathbf{m}^{i,k}$ and $\bar{\mathbf{m}}^{i,k}$, it follows that

$$||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2 = (s_2^{i,k})^2||\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||^2 + (1 - s_2^{i,k})^2||\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k}||^2 + 2s_2^{i,k}(1 - s_2^{i,k})(\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k}))^T(\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k}).$$

The expectation of this expression is

$$\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2] = (s_2^{i,k})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||^2] + (1 - s_2^{i,k})^2\mathbb{E}[||\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k}||^2] + 2s_2^{i,k}(1 - s_2^{i,k})\mathbb{E}[(\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(x^{i,k}))^T (\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k})]. \tag{5.11}$$

Now, introducing the associated sigma field of the process $\mathcal{F}^{i,k} = \sigma(\mathbf{x}^{j,t}, \mathbf{m}^{j,t}, \bar{\mathbf{m}}^{j,t}; j \leq i, t \leq k)$ by the law of total expectation, it follows that

$$\begin{aligned}\mathbb{E}[(\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(x^{i,k}))^T (\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k})] &= \mathbb{E}[\mathbb{E}[(\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k}))^T (\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k})|\mathcal{F}^{i,k}]] \\ &= \mathbb{E}[(\mathbb{E}[\tilde{\mathbf{g}}^{i,k}|\mathcal{F}^{i,k}] - \nabla f^{\beta^i}(x^{i,k}))^T (\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k})] \\ &= 0,\end{aligned}$$

where the second equality holds because $\mathbf{m}^{i,k}, \bar{\mathbf{m}}^{i,k}$ and $\nabla f^{\beta^i}(\mathbf{x}^{i,k})$ are fixed conditioned on $\mathcal{F}^{i,k}$ and because $\mathbb{E}[\tilde{\mathbf{g}}^{i,k}|\mathbf{x}^{i,k}] = \nabla f^{\beta^i}(\mathbf{x}^{i,k})$ where $\tilde{\mathbf{g}}^{i,k}$ is an unbiased estimator of the gradient by Eq (5.7). By substituting this result in (5.11), it follows that

$$\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2] = (s_2^{i,k})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||^2] + (1 - s_2^{i,k})^2\mathbb{E}[||\mathbf{m}^{i,k} - \bar{\mathbf{m}}^{i,k}||^2].$$

By repeating this process iteratively, we obtain

$$\begin{aligned}\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2] = &(s_2^{i,k})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||^2] \\ &+ \sum_{r=0}^{k-1} (s_2^{i,r})^2 \prod_{t=r}^{k-1} (1 - s_2^{i,t+1})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,r} - \nabla f^{\beta^i}(\mathbf{x}^{i,r})||^2] \\ &+ \prod_{t=0}^{k} (1 - s_2^{i,t})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,0} - \nabla f^{\beta^i}(x^{i,0})||^2].\end{aligned} \tag{5.12}$$

Finally, by observing that $\forall r \in [0, k], \mathbb{E}[\tilde{\mathbf{g}}^{i,r}|\mathbf{x}^{i,r}] = \nabla f^{\beta^i}(x^{i,r})$ and by the law of total expectation, we obtain

$$\begin{aligned}\mathbb{E}[||\tilde{\mathbf{g}}^{i,r} - \nabla f^{\beta^i}(x^{i,r})||^2] &= \mathbb{E}[||\tilde{\mathbf{g}}^{i,r} - \mathbb{E}[\tilde{\mathbf{g}}^{i,r}|\mathbf{x}^{i,r}]||^2] \\ &= \mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2] - \mathbb{E}[||\nabla f^{\beta^i}(x^{i,r})||^2] \\ &\leq \mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2].\end{aligned}$$

Introducing this inequality to Eq (5.12) completes the proof. □

Second, the expectation of the squared norm of the stochastic gradient estimators are bounded by a constant depending quadratically on the dimension.

Lemma 5.6.3. *Let $i \in \mathbb{N}$, $r \in [0, k]$ and $j \in [1, n]$; then under Assumption 1, we have*

$$\mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2] \leq L_0(F)^2 \, (n+4)^2,$$

where $L_0(F)$ is the Lipschitz constant of F.

Proof. By Eq (5.6) with $q = 1$, it follows that

$$\begin{aligned}
\mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2] &= \mathbb{E}\left[\frac{||u||^2}{(\beta^i)^2}\left(F(\mathbf{x}^{i,r} + \beta^i \mathbf{u}, \boldsymbol{\xi}^1) - F(\mathbf{x}^{i,r}, \boldsymbol{\xi}^0)\right)^2\right] \\
&\leq L_0(F)^2 \mathbb{E}[||u||^4] \\
&\leq L_0(F)^2 \, (n+4)^2
\end{aligned}$$

where the first inequality follows from Assumption 1(b) and the second by [141, Lemma 1]. □

Finally, a technical lemma bounds the second term of the decomposition of Lemma 5.6.2 by a decreasing sequence. It achieves the same rate of convergence as in [31] without sampling any stochastic gradient.

Lemma 5.6.4. *For the subproblem $i \in \mathbb{N}$, let $s_2^{i,k}$ be defined such that $s_2^{i,k} = \frac{s_2^{i,0}}{(k+1)^{\alpha_2}}$ with $\alpha_2 \in (0, 1)$ and $s_2^{i,0} \in (0, 1)$; then, for k such that*

$$\frac{k}{(k+1)^{\alpha_2}} \geq \frac{\ln(s_2^{i,0}) + (1+\alpha_2)\ln(k)}{s_2^{i,0}}, \tag{5.13}$$

the following inequality holds

$$\sum_{r=0}^{k-1} (s_2^{i,r})^2 \prod_{t=r}^{k-1} (1 - s_2^{i,t+1})^2 \leq \frac{9 s_2^{i,0}}{k^{\alpha_2}}. \tag{5.14}$$

Proof. Let $k \in \mathbb{N}$; as in [31], the strategy consists of breaking up the sum in order to bound both

terms separately.

$$\begin{aligned}
\sum_{r=0}^{k-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1-s_2^{i,t+1})^2 &= \sum_{r=0}^{\lfloor k/2\rfloor-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1-s_2^{i,t+1})^2+\sum_{r=\lfloor k/2\rfloor}^{k-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1-s_2^{i,t+1})^2\\
&\leq (1-s_2^{i,k})^{2\lfloor k/2\rfloor}\sum_{r=0}^{\lfloor k/2\rfloor-1}(s_2^{i,r})^2+(s_2^{i,\lfloor k/2\rfloor-1})^2\sum_{r=\lfloor k/2\rfloor}^{k-1}(1-s_2^{i,k})^{2(k-r-1)}\\
&\leq (s_2^{i,0})^2\lfloor k/2\rfloor\,(1-s_2^{i,k})^{2\lfloor k/2\rfloor}+\frac{8(s_2^{i,0})^2}{k^{2\alpha_2}}\sum_{r=0}^{\lfloor k/2\rfloor}(1-s_2^{i,k})^{2r}\\
&\leq (s_2^{i,0})^2k\,(1-s_2^{i,k})^{2\lfloor k/2\rfloor}+\frac{8\,(s_2^{i,0})^2}{k^{2\alpha_2}(1-\,(1-s_2^{i,k})^2)}\\
&\leq (s_2^{i,0})^2k\,(1-s_2^{i,k})^{2\lfloor k/2\rfloor}+\frac{8s_2^{i,0}}{k^{\alpha_2}(2-s_2^{i,k})}.
\end{aligned}$$

Now, we are looking for k such that

$$s_2^{i,0}k\,(1-s_2^{i,k})^{2\lfloor k/2\rfloor}\leq\frac{1}{k^{\alpha_2}}\Leftrightarrow e^{2\lfloor k/2\rfloor\ln(1-s_2^{i,k})}\leq\frac{1}{(s_2^{i,0})k^{1+\alpha_2}}.$$

As, $\ln(1-x)\leq -x$, it is sufficient to find k such that

$$\begin{aligned}
e^{-s_2^{i,0}\frac{k}{(k+1)^{\alpha_2}}} &\leq \frac{1}{(s_2^{i,0})k^{1+\alpha_2}}\\
\Leftrightarrow \frac{k}{(k+1)^{\alpha_2}} &\geq \frac{\ln(s_2^{i,0})+\,(1+\alpha_2)\ln(k)}{s_2^{i,0}}.
\end{aligned}$$

Taking such a k allows us to complete the proof. □

Combining the three previous lemmas allows the bounding of the variance term in Proposition 5.6.1.

Proposition 5.6.5. *In the setting of Lemmas 5.6.3 and 5.6.4 and under Assumption 1(b), the variance term of Proposition 5.6.1 is bounded by*

$$\mathbb{E}[||\mathbf{m}^{i,k+1}-\bar{\mathbf{m}}^{i,k+1}||_2^2]\leq\frac{9s_2^{i,0}L_0(F)^2\,(n+4)^2}{k^{\alpha_2}}+o\left(\frac{1}{k^{\alpha_2}}\right).$$

Proof. By Lemmas 5.6.2 and 5.6.3, it follows that

$$\begin{aligned}\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2] \leq\ & (s_2^{i,k})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,k}||^2] + \sum_{r=0}^{k-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1-s_2^{i,t+1})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,r}||^2] \\ &+ \prod_{t=0}^{k}(1-s_2^{i,t})^2\mathbb{E}[||\tilde{\mathbf{g}}^{i,0}||^2] \\ &\leq \left((s_2^{i,k})^2 + \sum_{r=0}^{k-1}(s_2^{i,r})^2\prod_{t=r}^{k-1}(1-s_2^{i,t+1})^2 + \prod_{t=0}^{k}(1-s_2^{i,t})^2\right) L_0(F)^2\,(n+4)^2.\end{aligned}$$

Now as $(s_2^{i,k})^2 = o\left(\frac{1}{k^{\alpha_2}}\right)$ and $\prod_{t=0}^{k}(1-s_2^{i,t})^2 = o\left(\frac{1}{k^{\alpha_2}}\right)$, the result follows from Lemma 5.6.4. □

5.6.1.3 Bound on the bias term

First, the bias term is bounded by a sum depending on s_1^k and s_2^k.

Lemma 5.6.6. *For the subproblem $i \in \mathbb{N}$ and at iteration $k \in \mathbb{N}$ of the Algorithm 11, we have*

$$\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq 2nL_1(f^{\beta^i})\left(\sum_{l=0}^{k-1} s_1^{i,l}\prod_{t=l}^{k-1}(1-s_2^{i,t+1})\right).$$

Proof. Foremost, observe that the quantity

$$S^{i,k} := \begin{cases} 1, & \text{if } k=0, \\ s_2^{i,k} + \sum_{r=0}^{k-1} s_2^{i,r}\prod_{t=r}^{k-1}(1-s_2^{i,t+1}) + \prod_{t=0}^{k}(1-s_2^{i,t}), & \text{otherwise,}\end{cases} \tag{5.15}$$

may be written recursively as

$$S^{i,k} = \begin{cases} 1, & \text{if } k=0, \\ s_2^{i,k} + (1-s_2^{i,k})S^{i,k-1}, & \text{otherwise.}\end{cases}$$

Note that in its second expression $S^{i,k} = 1$ for all k. Therefore, by definition of $\bar{m}_j^{i,k}$ and the previous result on $S^{i,k}$, it follows that

$$\begin{aligned}\bar{\mathbf{m}}^{i,k} &= s_2^{i,k}\nabla f^{\beta^i}(\mathbf{x}^{i,k}) + \sum_{r=0}^{k-1} s_2^{i,r}\prod_{t=r}^{k-1}(1-s_2^{i,t+1})\nabla f^{\beta^i}(\mathbf{x}^{i,r}) + \prod_{t=0}^{k}(1-s_2^{i,t})\nabla f^{\beta^i}(\mathbf{x}^{i,0}), \\ \nabla f^{\beta^i}(\mathbf{x}^{i,k}) &= \left(s_2^{i,k} + \sum_{r=0}^{k-1} s_2^{i,r}\prod_{t=r}^{k-1}(1-s_2^{i,t+1}) + \prod_{t=0}^{k}(1-s_2^{i,t})\right)\nabla f^{\beta^i}(\mathbf{x}^{i,k}).\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \sum_{r=0}^{k-1} s_2^{i,r} \prod_{t=r}^{k-1} (1 - s_2^{i,t+1}) \mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,r}) - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \\ + \prod_{t=0}^{k} (1 - s_2^{i,t}) \mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,0}) - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1].\end{aligned} \tag{5.16}$$

By the smoothness of the function f^{β^i}, Lemma F(3) of [31] ensures that $\forall r \in [0, k-1]$

$$||\nabla f^{\beta^i}(\mathbf{x}^{i,r}) - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1 \leq \sum_{l=r}^{k-1} ||\nabla f^{\beta^i}(\mathbf{x}^{i,l+1}) - \nabla f^{\beta^i}(\mathbf{x}^{i,l})||_1 \ \leq \ 2nL_1(f^{\beta^i}) \sum_{l=r}^{k-1} s_1^{i,l}.$$

Substituting this inequality into Eq (5.16) gives

$$\mathbb{E}[||\bar{m}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq 2nL_1(f^{\beta^i}) S_1^{i,k}, \tag{5.17}$$

where

$$S_1^{i,k} = \sum_{r=0}^{k-1} s_2^{i,r} \sum_{l=r}^{k-1} s_1^{i,l} \prod_{t=r}^{k-1} (1 - s_2^{i,t+1}) + \sum_{l=0}^{k-1} s_1^{i,l} \prod_{t=0}^{k} (1 - s_2^{i,t}).$$

Reordering the terms in S_1^k, we obtain

$$\begin{aligned}S_1^{i,k} &= \sum_{l=0}^{k-1} s_1^{i,l} \left(\sum_{r=0}^{l} s_2^{i,r} \prod_{t=r}^{k-1} (1 - s_2^{i,t+1}) + \prod_{t=0}^{k} (1 - s_2^{i,t}) \right) \\ &= \sum_{l=0}^{k-1} s_1^{i,l} \left(s_2^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) + \sum_{r=0}^{l-1} s_2^{i,r} \prod_{t=r}^{k-1} (1 - s_2^{i,t+1}) + \prod_{t=0}^{k} (1 - s_2^{i,t}) \right) \\ &= \sum_{l=0}^{k-1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) \underbrace{\left(s_2^{i,l} + \sum_{r=0}^{l-1} s_2^{i,r} \prod_{t=r}^{l-1} (1 - s_2^{i,t+1}) + \prod_{t=0}^{l} (1 - s_2^{i,t}) \right)}_{S^{i,l}=1} \\ &= \sum_{l=0}^{k-1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}),\end{aligned}$$

which completes the proof. □

Second, the sum may be bounded by a term decreasing with k.

Lemma 5.6.7. *For the subproblem $i \in \mathbb{N}$, let $s_2^{i,k} = \frac{s_2^{i,0}}{(k+1)^{\alpha_2}}$ and $s_1^{i,k} = \frac{s_1^{i,0}}{(k+1)^{\alpha_1}}$ with $s_1^{i,0} \in (0,1)$, $s_2^{i,0} \in (0,1)$ and $0 < \alpha_2 < \alpha_1 < 1$; then, for k such that*

$$\frac{k}{(k+1)^{\alpha_2}} \geq \frac{2\left(\ln(s_2^{i,0}) + (1+\alpha_1-\alpha_2)\ln(k)\right)}{s_2^{i,0}}, \tag{5.18}$$

the following inequality holds

$$\sum_{l=0}^{k-1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) \leq \frac{5 s_1^{i,0}}{s_2^{i,0} k^{\alpha_1-\alpha_2}}. \tag{5.19}$$

Proof. The proof follows the proof of Lemma 5.6.4. The sum is partitioned as follows:

$$\begin{aligned}
\sum_{l=0}^{k-1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) &= \sum_{l=0}^{\lfloor k/2 \rfloor - 1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) + \sum_{l=\lfloor k/2 \rfloor - 1}^{k-1} s_1^{i,l} \prod_{t=l}^{k-1} (1 - s_2^{i,t+1}) \\
&\leq (1 - s_2^{i,k})^{\lfloor k/2 \rfloor} \sum_{l=0}^{\lfloor k/2 \rfloor - 1} s_1^{i,l} + s_1^{i,\lfloor k/2 \rfloor - 1} \sum_{l=\lfloor k/2 \rfloor - 1}^{k-1} (1 - s_2^{i,k})^{k-r-1} \\
&\leq s_1^{i,0} k \, (1 - s_2^{i,k})^{\lfloor k/2 \rfloor} + \frac{4 s_1^{i,0}}{k^{\alpha_1}(1 - (1 - s_2^{i,k}))} \\
&= \frac{s_1^{i,0} s_2^{i,0} k \, (1 - s_2^{i,k})^{\lfloor k/2 \rfloor}}{s_2^{i,0}} + \frac{4 s_1^{i,0}}{s_2^{i,0} k^{\alpha_1 - \alpha_2}}.
\end{aligned}$$

Now, as in Lemma 5.6.4 taking k such that

$$\frac{k}{(k+1)^{\alpha_2}} \geq \frac{2\left(\ln(s_2^{i,0}) + (1+\alpha_1-\alpha_2)\ln(k)\right)}{s_2^{i,0}}$$

ensures that $s_2^{i,0} k \, (1 - s_2^{i,k})^{\lfloor k/2 \rfloor} \leq \frac{1}{k^{\alpha_1 - \alpha_2}}$, which completes the proof. □

Finally, using the two previous lemmas allows for bounding of the bias term.

Proposition 5.6.8. *In the setting of Lemma 5.6.7, the bias term of Proposition 5.6.1 is bounded by*

$$\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq 10 n L_1(f^{\beta^i}) \frac{s_1^{i,0}}{s_2^{i,0} k^{\alpha_1 - \alpha_2}}.$$

Proof. The proof is a straightforward consequence of Lemmas 5.6.6 and 5.6.7. □

5.6.1.4 Convergence rate in mean of the ZO-signum algorithm

As the different terms in the inequality of Proposition 5.6.1 have been bounded, the main result of this section may be derived as in the following theorem.

Theorem 5.6.9. *For a subproblem $i \in \mathbb{N}$ and under Assumption 1, let $\alpha_1 \in (0,1)$, $\alpha_2 \in (0, \alpha_1)$, $0 < s_1^{i,0}$, $s_2^{i,0} < 1$ and $K > C$ where $C \in \mathbb{N}$ satisfies Eqs (5.13) and (5.18); we have*

$$\begin{aligned}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||_1] \leq \frac{1}{K^{1-\alpha_1} - \frac{C}{K^{\alpha_1}}} \Bigg(& \frac{D_f^i}{s_1^{i,0}} + \frac{n\sqrt{n} L_0(F) s_1^{i,0}}{\beta^i} \sum_{k=C}^{K} \frac{1}{k^{2\alpha_1}} \\ & + 6\sqrt{s_2^{i,0}} L_0(F) \sqrt{n}(n+4) \sum_{k=C}^{K} \frac{1}{k^{\alpha_1 + \frac{\alpha_2}{2}}} \\ & + \frac{40 L_0(F) s_1^{i,0} n\sqrt{n}}{s_2^{i,0} \beta^i} \sum_{k=C}^{K} \frac{1}{k^{2\alpha_1 - \alpha_2}} \Bigg),\end{aligned} \tag{5.20}$$

where $f^{\beta^i}(\mathbf{x}^{i,C}) - \min_{\mathbf{x}} f^{\beta^i}(\mathbf{x}) \leq D_f^i$, $L_0(F)$ is the Lipschitz constant of F and R is randomly picked from a uniform distribution in $[C, K]$.

Proof. Let $C \in \mathbb{N}$ satisfy Eqs (5.13) and (5.18) and, summing over the inequality in Proposition 5.6.1, it follows that

$$\begin{aligned}\sum_{k=C}^{K} s_1^{i,k} \mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \mathbb{E}[f^{\beta^i}(x^{i,C}) - f^{\beta^i}(\mathbf{x}^{i,K+1})] + \frac{n L_1(f^{\beta^i})}{2} \sum_{k=C}^{K} (s_1^{i,k})^2 \\ + 2\sqrt{n} \sum_{k=C}^{K} s_1^{i,k} \sqrt{\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||_2^2]} + 2 \sum_{k=C}^{K} s_1^{i,k} \mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1].\end{aligned}$$

By substituting the results of Propositions 5.6.5 and 5.6.8 in the previous inequality, we obtain

$$\begin{aligned}\sum_{k=C}^{K} s_1^{i,k} \mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \mathbb{E}[f^{\beta^i}(x^{i,C}) - f^{\beta^i}(\mathbf{x}^{i,K+1})] + \frac{n L_1(f^{\beta^i})}{2} \sum_{k=C}^{K} (s_1^{i,k})^2 \\ + 6\sqrt{s_2^{i,0}} L_0(F)\ (n+4)\sqrt{n} \sum_{k=C}^{K} \frac{s_1^{i,0}}{k^{\alpha_1 + \frac{\alpha_2}{2}}} + \frac{20 L_1(f^{\beta^i}) s_1^{i,0} n}{s_2^{i,0}} \sum_{k=C}^{K} \frac{s_1^{i,0}}{k^{2\alpha_1 - \alpha_2}}.\end{aligned}$$

Dividing both sides by $s_1^{i,0} K^{-\alpha_1}\ (K - C)$, picking R randomly uniformly in $[C, K]$ and using the

definition of D_f^i given that $\min_{\mathbf{x}} f(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x}$, we get

$$\begin{aligned}\mathbb{E}[||\nabla f^{\beta^i}(x^{i,R})||_1] &= \frac{1}{K-C}\sum_{k=C}^{K}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \frac{1}{K-C}\sum_{k=C}^{K}\frac{K^{\alpha_1}}{k^{\alpha_1}}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \\ &\leq \frac{1}{K^{1-\alpha_1}-\frac{C}{K^{\alpha_1}}}\left(\frac{D_f^i}{s_1^{i,0}} + \frac{nL_1(f^{\beta^i})s_1^{i,0}}{2}\sum_{k=C}^{K}\frac{1}{k^{2\alpha_1}}\right. \\ &+ 6\sqrt{s_2^{i,0}}L_0(F)\,(n+4)\sqrt{n}\sum_{k=C}^{K}\frac{1}{k^{\alpha_1+\frac{\alpha_2}{2}}} \\ &\left.+ \frac{20L_1(f^{\beta^i})s_1^{i,0}n}{s_2^{i,0}}\sum_{k=C}^{K}\frac{1}{k^{2\alpha_1-\alpha_2}}\right).\end{aligned}$$

Recalling that $L_1(f^{\beta^i}) = \frac{2\sqrt{n}L_0(F)}{\beta^i}$ (see [141, Lemma 2]) completes the proof. □

This theorem allows one to prove the convergence rate in mean of the norm of the gradient when α_1 and α_2 are chosen adequately. In particular, the following corollary provides the convergence rate when $\alpha_1 = \frac{3}{4}$ and $\alpha_2 = \frac{1}{2}$.

Corollary 5.6.10. *Under the same setting of Theorem 5.6.9 with $\beta^i \approx 1$ $\alpha_1 = \frac{3}{4}$, $\alpha_2 = \frac{1}{2}$, $s_1^{i,0} = \frac{1}{n^{\frac{3}{4}}}$ and $s_2^{i,0} \approx 1$, we have*

$$\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||_2] = O\left(\frac{n^{\frac{3}{2}}}{K^{1/4}}\ln(K)\right). \tag{5.21}$$

Proof. The result is a direct consequence of Theorem 5.6.9 with the specified constant, and it can be obtained by noting that $||\cdot||_2 \leq ||\cdot||_1$ in $\mathbb{R}^n$. □

In [47, 71, 116], the function F is assumed to be smooth with a Lipschitz continuous gradient. In the present work, F is only assumed to be Lipschitz continuous. This has two main consequences on the result of convergence: the dependence of the dimension on the convergence rate is larger. Furthermore, while β must be chosen relatively small in the smooth case, it is interesting to note that it does not have to be this way in the nonsmooth case.

5.6.1.5 The convex case

The convergence rate results for the ZO-signum algorithm has been derived in the non-convex case. In the next theorem, they are derived for the case when the function f^{β^i} is convex.

Theorem 5.6.11. *Under Assumption 1, suppose moreover that f^{β^i} is convex and there exists ρ such that $\rho = \max_{k\in\mathbb{N}} ||\mathbf{x}^{i,k} - \mathbf{x}^{i,*}||$; then, by setting*

$$s_1^{i,k} = \frac{2\rho}{(k+1)},\ s_2^{i,k} = \frac{1}{(k+1)^{\frac{2}{3}}} \text{ and } \Gamma^k := \prod_{l=2}^{k}\left(1 - \frac{2}{k+1}\right) = \frac{2}{k(k+1)} \text{ with } \Gamma^1 = 1, \tag{5.22}$$

it follows that

$$\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,K}) - f^{\beta^i}(\mathbf{x}^*)] \leq \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i K^{\frac{1}{3}}} \tag{5.23}$$

and

$$\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||] \leq \frac{2L_0(F)}{K^2} + \frac{4\rho n\sqrt{n}L_0(F)}{\beta^i K^{\frac{1}{3}}}, \tag{5.24}$$

where R is a random variable in $[0, K-1]$ whose the probability distribution is given by

$$\mathbb{P}(R = k) = \frac{s_1^{i,k}/\Gamma^{k+1}}{\sum_{k=0}^{K-1} s_1^{i,k}/\Gamma^{k+1}}.$$

Proof. Under the assumptions in the statement of Theorem 5.6.11, it follows by Proposition 5.6.1 that

$$\begin{aligned}
\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k+1}) - f^{\beta^i}(\mathbf{x}^{i,*})] &\leq \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*})] - s_1^{i,k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||] \\
&\quad + \frac{nL_1(f^{\beta^i})}{2}(s_1^{i,k})^2 + 2s_1^{i,k}\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \\
&\quad + 2s_1^{i,k}\sqrt{n}\sqrt{\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||^2]} \\
&\leq \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*})] - s_1^{k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||] \\
&\quad + \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i\ (k+1)^{\frac{4}{3}}},
\end{aligned} \tag{5.25}$$

where the last inequality follows thanks to Propositions 5.6.5 and 5.6.8 with $L_1(f^{\beta^i}) = \frac{2L_0(F)\sqrt{n}}{\beta^i}$ and the values of $s_1^{i,k}$ and $s_2^{i,k}$. Now, by convexity assumption of f^{β^i} and the bound ρ, the following holds

$$\begin{aligned}
f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*}) &\leq \nabla f^{\beta^i}(\mathbf{x}^{i,k})^T\ (\mathbf{x}^{i,k} - \mathbf{x}^{i,*}) \\
&\leq ||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||\ ||\mathbf{x}^{i,k} - \mathbf{x}^{i,*}|| \\
&\leq \rho||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||.
\end{aligned}$$

Thus, by substituting this result into Eq (5.25), it follows that

$$\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k+1}) - f^{\beta^i}(\mathbf{x}^{i,*})] \leq \left(1 - \frac{2}{(k+1)}\right)\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*})] + \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i\ (k+1)^{\frac{4}{3}}}.$$

Now by dividing both sides of the equation by Γ^{k+1} and summing up the inequalities, it follows that

$$\begin{aligned}\frac{\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,K}) - f^{\beta^i}(\mathbf{x}^{i,*})]}{\Gamma^K} &\leq \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i}\sum_{k=0}^{K-1}\frac{1}{\Gamma^{k+1}\ (k+1)^{\frac{4}{3}}} \\ &\leq \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i}\sum_{k=0}^{K-1}(k+1)^{\frac{2}{3}}.\end{aligned}$$

Thus

$$\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,K}) - f^{\beta^i}(\mathbf{x}^{i,*})] \leq \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i}\Gamma^K\sum_{k=0}^{K-1}(k+1)^{\frac{2}{3}} \leq \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i K^{\frac{1}{3}}}.$$

Now, the second part of the proof may be demonstrated. By Eq (5.25), it also follows that

$$s_1^{i,k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||] \leq \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*})] - \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k+1}) - f^{\beta^i}(\mathbf{x}^{i,*})] + \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i\ (k+1)^{\frac{4}{3}}}.$$

As in the previous part, by dividing both sides by Γ^{k+1}, summing up the inequalities and noting that $\bar{f^k} = \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,*})]$, we obtain

$$\sum_{k=0}^{K-1}\frac{s_1^{i,k}}{\Gamma^{k+1}}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||] \leq \sum_{k=0}^{K-1}\frac{\bar{f^k} - \bar{f^{k+1}}}{\Gamma^{k+1}} + \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i}\sum_{k=0}^{K-1}\frac{1}{\Gamma^{k+1}\ (k+1)^{\frac{4}{3}}}.$$

Then, again by dividing both sides by $\sum_{k=0}^{K-1}\frac{s_1^{i,k}}{\Gamma^{k+1}}$ it follows that

$$\begin{aligned}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||] &= \frac{\sum_{k=0}^{K-1}\frac{s_1^{i,k}}{\Gamma^{k+1}}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||]}{\sum_{k=0}^{K-1}\frac{s_1^{i,k}}{\Gamma^{k+1}}} \\ &\leq \frac{1}{\sum_{k=0}^{K-1}\frac{s_1^{i,k}}{\Gamma^{k+1}}}\left(\sum_{k=0}^{K-1}\frac{\mathbb{E}[\bar{f^k} - \bar{f^{k+1}}]}{\Gamma^{k+1}} + \frac{4\rho^2 n\sqrt{n}L_0(F)}{\beta^i}\sum_{k=0}^{K-1}\frac{1}{\Gamma^{k+1}\ (k+1)^{\frac{4}{3}}}\right),\end{aligned}$$

where R is a random variable whose distribution is given in the statement of the theorem. Now, as

in Eq (2.21) of [25], the following inequalities hold

$$\sum_{k=0}^{K-1} \frac{\bar{f}^k - \bar{f}^{k+1}}{\Gamma^{k+1}} \leq \bar{f}^0 + \sum_{k=1}^{K-1} \frac{2}{\Gamma^{k+1}(k+1)} \bar{f}^k \qquad \text{and} \qquad \sum_{k=0}^{K-1} \frac{s_1^{i,k}}{\Gamma^{k+1}} = \frac{\rho}{\Gamma^K}.$$

Thus, by substituting these in the inequality involving the expectation, we obtain

$$\begin{aligned}
\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||] &\leq \frac{\Gamma^K}{\rho} \left(\mathbb{E}[\bar{f}^0] + \sum_{k=1}^{K-1} \frac{2}{\Gamma^{k+1}(k+1)} \mathbb{E}[\bar{f}^k] + \frac{4\rho^2 n\sqrt{n} L_0(F)}{\beta^i} \sum_{k=0}^{K-1} \frac{1}{\Gamma^{k+1}\,(k+1)^{\frac{4}{3}}} \right) \\
&\leq \frac{\Gamma^K}{\rho} \left(\mathbb{E}[\bar{f}^0] + \frac{8\rho n\sqrt{n} L_0(F)}{\beta^i} \sum_{k=0}^{K-1} \frac{1}{\Gamma^{k+1}\,(k+1)^{\frac{4}{3}}} \right) \\
&\leq \frac{2L_0(F)}{K^2} + \frac{8\rho n\sqrt{n} L_0(F)}{\beta^i K^{\frac{1}{3}}},
\end{aligned}$$

where the second inequality follows from Eq (5.23). □

5.6.1.6 Summary of convergence rates and complexity guarantees

The result obtained in Eq (5.21) is consistent with the convergence results of other ZO methods. To gain a better understanding of its performance, this result is compared with those of four other algorithms from different perspectives: the assumptions, the measure used, the convergence rate and the function query complexity. All methods seek a solution to a stochastic optimization problem; the comparison is presented in Table 5.2. Since the convergence rates of the ZO-signum and ZO-signSGD algorithms are measured by using $||\nabla f(\mathbf{x})||$, although $||\nabla f(x)||^2$ is used for ZO-adaMM and ZO-SGD, Jensen's inequality is used to rewrite the convergence rates in terms of the gradient norm.

- for ZO-SGD [71]

$$\mathbb{E}[||\nabla f(\mathbf{x})||] \leq \sqrt{\mathbb{E}[||\nabla f(\mathbf{x})||^2]} \leq \sqrt{O\left(\frac{\sigma\sqrt{n}}{\sqrt{K}} + \frac{n}{K}\right)} \leq O\left(\frac{\sqrt{\sigma} n^{\frac{1}{4}}}{K^{\frac{1}{4}}} + \frac{\sqrt{n}}{\sqrt{K}}\right);$$

- for ZO-adaMM [47]

$$\begin{aligned}
\mathbb{E}[||\nabla f(\mathbf{x})||] \leq \sqrt{\mathbb{E}[||\nabla f(\mathbf{x})||^2]} &\leq \sqrt{O\left(\left(\frac{n}{\sqrt{K}} + \frac{n^2}{K}\right)\sqrt{\ln(K) + \ln(n)}\right)} \\
&\leq O\left(\left(\frac{\sqrt{n}}{K^{\frac{1}{4}}} + \frac{n}{\sqrt{K}}\right)(\ln(K) + \ln(n))^{\frac{1}{4}}\right),
\end{aligned}$$

where the third inequalities are due to $\sqrt{a^2+b^2} \leq a+b$, for $a, b \geq 0$. For ZO-signSGD, unless the value of b depends on K, the algorithm's convergence is only guaranteed within some ball around the solution, making it difficult to compare with other methods. Thus, in the non-convex case, after this transformation, it becomes apparent that ZO-signum has a convergence rate that is $O\left(\frac{n^{\frac{3}{4}}}{\sqrt{\sigma}}\right)$ and $O(\sqrt{n})$ worse than those of ZO-SGD and ZO-adaMM, respectively. This may be attributed to the milder assumption made on the function F in the present work, which also explains why the convergence is relative to f^{β}. In the convex case, ZO-signum has a convergence rate that is $O\left(\frac{nK^{\frac{1}{6}}}{\sigma}\right)$ worse that of ZSCG and $O\left(\sqrt{n}K^{\frac{1}{6}}\right)$ worse that of ZO-SGD. This may be explained by the $sign(\cdot)$ operator loosing the magnitude information of the gradient when it is applied. This problem may be fixed as in [87] but it outside the scope of this work. Finally, all methods except ZO-signSGD are momentum-based versions of the original ZO-SGD method. Although the momentum-based versions are mostly used in practice, it is interesting to notice that none of these methods possess a better convergence rate than the original ZO-SGD method. The next section provides some clues about the interests of the momentum-based method.

Table 5.2 Summary of convergence rates and query complexity for various ZO algorithms given K iterations.

Method	Assumptions	Measure	Convergence rate	Queries
ZO-SGD [71]	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{1+}$ $\mathbb{E}[\|\|\nabla F(\mathbf{x},\boldsymbol{\xi}) - \nabla f(\mathbf{x})\|\|^2] \leq \sigma^2$	$\mathbb{E}[\|\|\nabla f(\mathbf{x}^R)\|\|_2]$	$O\left(\frac{\sqrt{\sigma}n^{\frac{1}{4}}}{K^{\frac{1}{4}}} + \frac{\sqrt{n}}{\sqrt{K}}\right)$	$O(K)$
ZO-signSGD [116]	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{0+}$ $F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{1+}$ $\|\|\nabla F(\mathbf{x},\boldsymbol{\xi})\|\|_2 \leq \eta$	$\mathbb{E}[\|\|\nabla f(\mathbf{x}^R)\|\|_2]$	$O\left(\frac{\sqrt{n}}{\sqrt{K}} + \frac{\sqrt{n}}{\sqrt{b}} + \frac{n}{\sqrt{bq}}\right)$	$O(bqK)$
ZO-adaMM [47]	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{0+}$ $F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{1+}$ $\|\|\nabla F(\mathbf{x},\boldsymbol{\xi})\|\|_\infty \leq \eta$	$\mathbb{E}[\|\|\nabla f(\mathbf{x}^R)\|\|_2]$	$O\Big(\left(\frac{\sqrt{n}}{K^{\frac{1}{4}}} + \frac{n}{\sqrt{K}}\right) \times (\ln(K) + \ln(n))^{\frac{1}{4}}\Big)$	$O(K)$
ZO-Signum	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{0+}$	$\mathbb{E}[\|\|\nabla f^{\beta}(\mathbf{x}^R)\|\|_2]$	$O\left(\frac{n\sqrt{n}}{K^{\frac{1}{4}}}\ln(K)\right)$	$O(K)$
ZO-Signum	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{0+}$, f convex	$\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,K}) - f^{\beta^i}(\mathbf{x}^{i,*})]$	$O\left(\frac{n\sqrt{n}}{K^{\frac{1}{3}}}\right)$	$O(K)$
ZO-SGD [141]	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{0+}$, f convex	$\mathbb{E}[f(\mathbf{x}^{i,K}) - f(\mathbf{x}^{i,*})]$	$O\left(\frac{n}{\sqrt{K}}\right)$	$O(K)$
Modified ZSCG [25]	$F(\cdot,\boldsymbol{\xi}) \in \mathcal{C}^{1+}$, F convex $\mathbb{E}[\|\|\nabla F(\mathbf{x},\boldsymbol{\xi}) - \nabla f(\mathbf{x})\|\|^2] \leq \sigma^2$	$\mathbb{E}[f(\mathbf{x}^{i,K}) - f(\mathbf{x}^{i,*})]$	$O\left(\frac{\sigma\sqrt{n}}{\sqrt{K}}\right)$	$O(K)$

5.6.2 Convergence rate of the SSO algorithm

The convergence analysis from the previous subsection is in mean, i.e., it establishes the expected convergence performance over many executions of the ZO-signum algorithm. As in [71], we now focus on the performance of a single run. A second hierarchical workflow of the different theoreti-

cal results is presented in Table 5.3.

Table 5.3 Workflow of Lemmas/Propositions/Theorems for the SSO convergence analysis.

Assumptions on F	Preliminary results			Intermediate results	Main result	When f^{β} is convex
Assumptions 1, 2 and 3 which imply $L_1(f^{\beta^i}) \leq L_1(f)$	Lemma 5.6.12 Proposition 5.6.8 Lemma 5.4.1(3)	Lemma 5.6.13 Theorem 5.6.9 Proposition 5.6.5 Proposition 5.6.8	Lemma 5.6.14	Lemma 5.6.15 Lemma 5.6.16	Theorem 5.6.17 (i)	Theorem 5.6.17 (ii)

Unlike [71], our analysis is based on a sequential optimization framework rather than a post-optimization process. Our SSO algorithm uses the norm of the momentum as an indicator of the quality of the current solution. In order to analyze the rate of convergence of this algorithm, the following additional assumptions are made regarding the function F. The first assumption concerns the smoothness of the function F. The assumption of smoothness is used only to guarantee that $L_1(f^{\beta^i})$ is a constant with respect to β^i, contrary to the non-smooth case (see [141, Eq (12)]).

Assumption 2. *The function $F(\cdot, \boldsymbol{\xi})$ has a $L_1(F)$-Lipschitz continuous gradient.*

The second assumption concerns the local convexity of the function f^{β}.

Assumption 3. *Let $(\mathbf{x}^{i,0})$ be a sequence of points produced by Algorithm 12 and $\mathbf{x}^{i,*}$ a sequence of local minima of f^{β^i}. We assume that there exists a threshold $I \in \mathbb{N}$ and a radius $\rho > 0$ such that $\forall i \geq I$:*

(1) f^{β^i} is convex on the ball $\mathcal{B}_{\rho}(\mathbf{x}^{i,*}) := \{\mathbf{x} \in \mathbb{R}^n : ||x - x^{i,*}|| < \rho\}$;

(2) $\mathbf{x}^{i,0} \in \mathcal{B}_{\rho}(\mathbf{x}^{i,*})$.

Under these assumptions, we will prove that if the norm of the momentum vector $\mathbf{m}$ is below some threshold, then this threshold can be used to bound the norm of the gradient. Second, an estimate for the number of iterations required to reduce the norm of $\mathbf{m}$ below the threshold is provided. The next lemma is simply technical and demonstrates the link between $\bar{\mathbf{m}}$ and $\mathbf{m}$.

Lemma 5.6.12. *For any subproblem $i \in \mathbb{N}$ and iteration $k \geq 1$, the following equality holds*

$$\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k-1}] = \mathbb{E}[\bar{\mathbf{m}}^{i,k}|\mathbf{x}^{i,k-1}],$$

where $\bar{\mathbf{m}}^{i,k}$ is defined recursively in Proposition 5.6.1.

Proof. The proof is conducted by induction on k. For $k = 1$, setting $\mathbf{m}^{i,0} = \tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,0}, \boldsymbol{\xi}^0)$ implies that

$$\mathbf{m}^{i,1} = s_2^{i,0}\tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,0}, \boldsymbol{\xi}^0) + \ (1 - s_2^{i,0})\mathbf{m}^{i,0} = \tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,0}, \boldsymbol{\xi}^0).$$

In the same way, $\bar{\mathbf{m}}^{i,1} = \nabla f^{\beta^i}(\mathbf{x}^{i,0})$. Therefore, we have

$$\mathbb{E}[\mathbf{m}^{i,1}|\mathbf{x}^{i,0}] = \mathbb{E}[\tilde{\nabla} f^{\beta^i}(\mathbf{x}^{i,0}, \boldsymbol{\xi}^0)|\mathbf{x}^{i,0}] = \nabla f^{\beta^i}(\mathbf{x}^{i,0}) = \mathbb{E}[\nabla f^{\beta^i}(\mathbf{x}^{i,0})|\mathbf{x}^{i,0}] = \mathbb{E}[\bar{\mathbf{m}}^{i,1}|\mathbf{x}^{i,0}].$$

Now, suppose that the induction assumption is true for a given $k \in \mathbb{N}$; then,

$$\mathbb{E}[\mathbf{m}^{i,k+1}|\mathbf{x}^{i,k}] = s_2^{i,k}\nabla f^{\beta^i}(\mathbf{x}^{i,k}) + \ (1 - s_2^{i,k})\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k}].$$

Now, by the law of total expectation

$$\begin{aligned}
\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k}] &= \mathbb{E}[\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k}, \mathbf{x}^{i,k-1}]|\mathbf{x}^{i,k}] \\
&= \mathbb{E}[\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k-1}]|\mathbf{x}^{i,k}] \\
&= \mathbb{E}[\mathbb{E}[\bar{\mathbf{m}}^{i,k}|\mathbf{x}^{i,k-1}]|\mathbf{x}^{i,k}] \quad \text{(by the induction assumption)} \\
&= \mathbb{E}[\bar{\mathbf{m}}^{i,k}|\mathbf{x}^{i,k}].
\end{aligned}$$

Thus as $\mathbb{E}[\nabla f^{\beta^i}(\mathbf{x}^{i,k})|\mathbf{x}^{i,k}] = \nabla f^{\beta^i}(\mathbf{x}^{i,k})$, it follows that

$$\begin{aligned}
\mathbb{E}[\mathbf{m}^{i,k+1}|\mathbf{x}^{i,k}] &= s_2^{i,k}\nabla f^{\beta^i}(\mathbf{x}^{i,k}) + \ (1 - s_2^{i,k})\mathbb{E}[\mathbf{m}^{i,k}|\mathbf{x}^{i,k}] \\
&= s_2^{i,k}\mathbb{E}[\nabla f^{\beta^i}(\mathbf{x}^{i,k})|\mathbf{x}^{i,k}] + \ (1 - s_2^{i,k})\mathbb{E}[\bar{\mathbf{m}}^{i,k}|\mathbf{x}^{i,k}] \\
&= \mathbb{E}[\bar{\mathbf{m}}^{i,k+1}|\mathbf{x}^{i,k}],
\end{aligned}$$

which completes the proof. □

The following lemma shows that if $||\mathbf{m}||$ is below a certain threshold, then this threshold can be used to bound the norm of the gradient.

Lemma 5.6.13. *For a subproblem $i \in \mathbb{N}$, let $K_i \in \mathbb{N}$ denote the first iteration in Algorithm 11 for which $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$; then, under Assumption 3 the norm of the gradient of the function f^{β^i} at $\mathbf{x}^{i,K}$ may be bounded as follows*

$$||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i})|| \leq \frac{L\beta^i}{4\beta^0} + 10nL_1(F)\frac{s_1^{i,0}}{s_2^{i,0}K_i^{\alpha_1-\alpha_2}}.$$

Moreover, if the problem $i+1$ is considered, the gradient of the function $f^{\beta^{i+1}}$ may be bounded at the point $\mathbf{x}^{i,K_i} = \mathbf{x}^{i+1,0}$ as follows:

$$||\nabla f^{\beta^{i+1}}(\mathbf{x}^{i+1,0})|| \leq ||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i})|| + L_1(F)\ (n+3)^{\frac{3}{2}}\ (\beta^i - \beta^{i+1}).$$

Proof. Let K_i be taken as in the statement of the lemma. The norm of the gradient may be bounded as follows:

$$\begin{aligned} ||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i})|| &\leq ||\mathbb{E}[\mathbf{m}^{i,K_i}|\mathbf{x}^{i,K_i}]|| + ||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i}) - \mathbb{E}[\mathbf{m}^{i,K_i}|\mathbf{x}^{i,K_i}]|| \\ &\leq \mathbb{E}[||\mathbf{m}^{i,K_i}||\ |\mathbf{x}^{i,K_i}] + ||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i}) - \mathbb{E}[\bar{\mathbf{m}}^{i,K_i}|\mathbf{x}^{i,K_i}]||, \end{aligned}$$

where the second inequality follows from Jensen's inequality and Lemma 5.6.12. Now, using $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$, $\mathbb{E}[\nabla f^{\beta^i}(\mathbf{x}^{i,K})|\mathbf{x}^{i,K_i}] = \nabla f^{\beta^i}(\mathbf{x}^{i,K_i})$, $L_1(f^{\beta^i}) \leq L_1(F)$ and the result of Proposition 5.6.8 completes the first part of the proof

$$\begin{aligned} ||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i})|| &\leq \frac{L\beta^i}{4\beta^0} + \mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,K_i}) - \bar{\mathbf{m}}^{i,K_i}||\ |\mathbf{x}^{i,K_i}] \\ &\leq \frac{L\beta^i}{4\beta^0} + 10nL_1(F)\frac{s_1^{i,0}}{s_2^{i,0}K_i^{\alpha_1-\alpha_2}}. \end{aligned}$$

The second part of the proof follows directly by applying the triangular inequality and the result in Lemma 5.4.1(3) because $\mathbf{x}^{i,K_i} = \mathbf{x}^{i+1,0}$. □

Under Assumption 2, the expected difference between the values of f^{β^i} at $\mathbf{x}^{i,0}$ and its optimal value is bounded in the next lemma.

Lemma 5.6.14. *Let I be the threshold from Assumption 2. If $i \geq I$, then*

$$\mathbb{E}[f^{\beta^{i+1}}(\mathbf{x}^{i+1,0}) - f^{\beta^{i+1}}(\mathbf{x}^{i+1,*})] \leq \rho\Bigg(\frac{L\beta^i}{4\beta^0} + 10nL_1(F)\frac{s_1^{i,0}}{s_2^{i,0}K_i^{\alpha_1-\alpha_2}} + L_1(F)\,(n+3)^{\frac{3}{2}}\,(\beta^i - \beta^{i+1})\Bigg). \tag{5.26}$$

Proof. Convexity of the function f^{β^i} on the ball $\mathcal{B}_\rho(\mathbf{x}^{i,*})$ implies that

$$\begin{aligned}\mathbb{E}[f^{\beta^{i+1}}(\mathbf{x}^{i+1,0}) - f^{\beta^{i+1}}(\mathbf{x}^{i+1,*})] &\leq \mathbb{E}[\langle \nabla f^{\beta^{i+1}}(\mathbf{x}^{i+1,0}), \mathbf{x}^{i+1,0} - \mathbf{x}^{i+1,*}\rangle] \\ &\leq \mathbb{E}[||\nabla f^{\beta^{i+1}}(\mathbf{x}^{i+1,0})||\,||\mathbf{x}^{i+1,0} - \mathbf{x}^{i+1,*}||].\end{aligned}$$

The result follows by using Lemma 5.6.13 and because $\mathbf{x}^{i+1,0}$ belongs to the ball $\mathcal{B}^\epsilon(\mathbf{x}^{i,*})$. □

Moreover, an estimate of the number of iterations required to reduce the norm of the gradient below some threshold may be given.

Lemma 5.6.15. *Under Assumptions 1–3, for a subproblem $i > I$ and in the setting of Algorithm 12, let $s_2^{i,0} \in \mathbb{R}^+$ be such that $k = 1$ in Eqs (5.13) and (5.18); assume that $L = \max(L_0(F), L_1(F))$, $\alpha_1 = \frac{3}{4}$ and $\alpha_2 = \frac{1}{2}$. Then, for a uniformly randomly chosen $R \in [0, K_i]$, it follows that*

$$\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{\beta^i K_i^{\frac{1}{4}}}\,(A^i + B^i),$$

where A^i and B^i are defined in Eq (5.27).

Proof. Markov's inequality implies that

$$\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||]}{L\beta^i}.$$

Now, given the result of Theorem 5.6.9 with the specified values of α_1 and α_2 and the fact that $L_1(f^{\beta^i}) \leq L_1(F)$ together with Lemma 5.6.14, it follows that

$$\frac{4\beta^0\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||]}{L\beta^i} \leq \frac{4\beta^0}{\beta^i K_i^{\frac{1}{4}}}\,(A^i + B^i),$$

where

$$A^i = \frac{\rho}{s_1^{i,0}}\left(\frac{\beta^{i-1}}{4\beta^0} + 10n\frac{s_1^{i-1,0}}{s_2^{i-1,0}K_{i-1}^{\frac{1}{4}}} + (n+3)^{\frac{3}{2}}(\beta^i - \beta^{i+1})\right),$$
$$B^i = \frac{ns_1^{i,0}}{2}H_k^{(-\frac{3}{2})} + \ln(K_i)\left(6\sqrt{s_2^{i,0}}(n+4)\sqrt{n} + \frac{20ns_1^{i,0}}{s_2^{i,0}}\right), \tag{5.27}$$

K_i is the iteration number for subproblem i and $H_k^{(-\frac{3}{2})}$ is the generalized harmonic number. □

The following lemma provides an estimate of the number of iterations required to bound the norm of the difference between $\mathbf{m}$ and the gradient below a certain threshold.

Lemma 5.6.16. *For a subproblem $i \in \mathbb{N}$ and in the setting of Algorithm 12, let $s_2^{i,0} \in \mathbb{R}^+$ be such that $k = 1$ in Eqs (5.13) and (5.18); assume that $L = \max(L_0(F), L_1(F))$, $\alpha_1 = \frac{3}{4}$ and $\alpha_2 = \frac{1}{2}$. Then, for a uniformly randomly chosen $R \in [0, K_i]$, it follows that*

$$\mathbb{P}\left(||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{\beta^i K_i^{\frac{1}{4}}}\left(3\sqrt{s_2^{i,0}}(n+4)\sqrt{n} + \frac{10ns_1^{i,0}}{s_2^{i,0}}\right).$$

Proof. By Markov's inequality, it follows that

$$\mathbb{P}\left(||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0\mathbb{E}[||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})||]}{L\beta^i}$$
$$= \frac{4\beta^0}{L\beta^i K_i}\sum_{k=0}^{K_i}\mathbb{E}[||\mathbf{m}^{i,k} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||] \leq \frac{4\beta^0}{\beta^i K_i^{\frac{1}{4}}}\left(3\sqrt{s_2^{i,0}}(n+4)\sqrt{n} + \frac{10ns_1^{i,0}}{s_2^{i,0}}\right),$$

where the last inequality holds by Propositions 5.6.5 and 5.6.8 with $\alpha_1 = \frac{3}{4}$ and $\alpha_2 = \frac{1}{2}$. □

Finally, the main theorem of this section may be stated.

Theorem 5.6.17. *Let Assumptions 1–3 hold and let I be the threshold from Assumption 3.*

(i) For $i \in \mathbb{N}$, set

$$\beta^i = \frac{1}{\sqrt{n}(i+1)^2}, s_1^{i,0} = \frac{1}{6n(i+1)^{3/2}} \text{ and } s_2^{i,0} = \frac{s_2}{(i+1)}$$

with s_2 so that Eqs (5.13) and (5.18) are satisfied for $k = 1$. Moreover, let us denote K_i as the first iteration for which $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$ and that without loss of generality $L =$

$\max(L_0(F), L_1(F))$. *Let* $\epsilon > 0$ *be the desired accuracy and let* $i^* \geq \sqrt{\frac{L}{\epsilon}} \geq I$. *If for any* $i \geq I, K_i \geq (i+1)^6$*; then after at most*

$$O\left(\frac{n^6 L^{7/2}}{\epsilon^{7/2}}\right)$$

function evaluations, the following inequality holds

$$||\nabla f^{\beta^{i^*}}(x^{i^*,0})|| \leq \epsilon. \tag{5.28}$$

(ii) Furthermore, when for every $i \in \mathbb{N}$, f^{β^i} *is convex; then, under the same setting as Theorem 5.6.11 given in Eq (5.22), it follows that after at most*

$$O\left(\frac{n^{\frac{9}{2}} L^{7/2}}{\epsilon^{7/2}}\right)$$

function evaluations, the inequality given by Eq (5.28) holds.

Proof. For a subproblem $i \in \mathbb{N}$, a probabilistic upper bound on the iteration $K_i \in \mathbb{N}$ such that $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$ may be provided. We have

$$||\mathbf{m}^{i,K_i}|| = \min_{k\in[0,K_i]} ||\mathbf{m}^{i,k}|| \leq ||\mathbf{m}^{i,R}|| \leq ||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})|| + ||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||, \tag{5.29}$$

where $R \sim \mathcal{U}[0, K_i]$. Now, probabilistic upper bounds on the number K_i are required to obtain that both terms on the right-hand side of the previous inequality are below $\frac{L\beta^i}{4\beta^0}$. For the first term of the right-hand side in Eq (5.29), using the specified values of $s_1^{i,0}$, $s_2^{i,0}$ and β^i, Lemma 5.6.16 ensures that

$$\begin{aligned}\mathbb{P}\left(||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) &\leq \frac{4\beta^0}{\beta^i K_i^{\frac{1}{4}}}\left(3\sqrt{s_2^{i,0}}\,(n+4)\sqrt{n} + \frac{10ns_1^{i,0}}{s_2^{i,0}}\right) \\ &\leq O\left(\frac{n\sqrt{n}\,(i+1)^{\frac{3}{2}}}{K_i^{\frac{1}{4}}}\right).\end{aligned}$$

The second term of the right-hand side in Eq (5.29) depends on the value of I. For subproblems

$i \leq I$, it follows by Markov's inequality and Theorem 5.6.9 that

$$\begin{aligned}
&\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{L\beta^i}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||] \\
&\leq \frac{4\beta^0}{\beta^i}\left(\frac{D_f^i}{s_1^{i,0}} + \frac{ns_1^{i,0}}{2}H_k^{(-\frac{3}{2})} + \ln(K_i)\left(6\sqrt{s_2^{i,0}}\,(n+4)\sqrt{n} + \frac{40s_1^{i,0}n}{s_2^{i,0}}\right)\right) \\
&\leq O\left(\frac{\max\left(\frac{n(i+1)^{\frac{7}{2}}}{L}, n\sqrt{n}\ln(K_i)(i+1)^{\frac{3}{2}}\right)}{K_i^{\frac{1}{4}}}\right).
\end{aligned}$$

For subproblems $i > I$, Lemma 5.6.15 ensures that

$$\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{\beta^i K^{\frac{1}{4}}}\,(A^i + B^i),$$

where A^i and B^i are given by Eq (5.27). Now, given the condition on K_i, it follows that

$$A^i = \rho n\,(i+1)^{3/2}\left(\frac{1}{i^2} + \frac{10}{s_2 i^2} + \frac{2\,(n+3)}{i^2(i+1)}\right)$$

and

$$B^i = \frac{H_k^{(-\frac{3}{2})}}{2(i+1)^{3/2}} + \ln(K_i)\left(\frac{6n\sqrt{n+3}\sqrt{s_2}}{\sqrt{i+1}} + \frac{12}{s_2\sqrt{i+1}}\right).$$

Thus, we obtain

$$\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq O\left(\frac{n\sqrt{n}\,(i+1)^{\frac{3}{2}}\ln(K_i)}{K_i^{\frac{1}{4}}}\right). \tag{5.30}$$

Therefore, to obtain that $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$, it takes at most the following number of iterations:

$$K_i = \begin{cases} O\left(\max\left(n^4\,(i+1)^{14}, n^6\,(i+1)^6\right)\right), & \text{if } i \leq I, \\ O\left(\left(n^6\,(i+1)^6\right)\right), & \text{otherwise.} \end{cases}$$

Thus, by taking $i^* \geq \sqrt{\frac{L}{\epsilon}}$, it follows that the number of iterations needed to reach the subproblem

i^* is

$$\begin{aligned}\sum_{i=1}^{i^*} K_i = \sum_{i=1}^{I} K_i + \sum_{i=I+1}^{i^*} K_i &= O\left(\max\left(n^4\,(I+1)^{15}, n^6\,(I+1)^7\right)\right) + O(n^6\,(i^*)^7) \\ &= O\left(\frac{n^6 L^{7/2}}{\epsilon^{7/2}}\right),\end{aligned} \tag{5.31}$$

where I is a constant with respect to ϵ. Once this number of iterations is reached, it follows that $||\mathbf{m}^{i^*,0}|| \leq \frac{L}{(i^*+1)^2} \leq \epsilon$ and by Lemma 5.6.13

$$||\nabla f^{\beta^{i^*}}(\mathbf{x}^{i^*,K_{i^*}})|| \leq \frac{L}{(i^*+1)^2} + \frac{L}{\sqrt{i^*+1}\,(i^*)^{\frac{3}{2}}} \leq 2\epsilon.$$

For the second part of the proof, the bounds on Eq (5.29) do not depend on the value of I since f^{β^i} is assumed convex for every $i \in \mathbb{N}$. With the setting in Eq (5.22), it follows that

$$\mathbb{P}\left(||\nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{\beta^i}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,R})||] \leq 16\frac{\rho n\sqrt{n}\,(i+1)^2}{K_i^{\frac{1}{3}}}$$

and

$$\mathbb{P}\left(||\mathbf{m}^{i,R} - \nabla f^{\beta^i}(\mathbf{x}^{i,R})|| \geq \frac{L\beta^i}{4\beta^0}\right) \leq \frac{4\beta^0}{L\beta^i} n\sqrt{n} L \frac{\sum_{k=0}^{K_i-1} \frac{2\rho}{\Gamma^{k+1}(k+1)^{\frac{4}{3}}}}{\sum_{k=0}^{K_i-1} \frac{2\rho}{\Gamma^{k+1}(k+1)}} \leq 8\frac{n\sqrt{n}\,(i+1)^2}{K_i^{\frac{1}{3}}},$$

where the first inequality follows by Theorem 5.6.11 and the second one by the definition of the probability density of R together with Propositions 5.6.5 and 5.6.8. Therefore, it takes at most $K_i = O(n^{\frac{9}{2}}(i+1)^6)$ iterations to obtain $||\mathbf{m}^{i,K_i}|| \leq \frac{L\beta^i}{4\beta^0}$. Thus, by taking $i^* \geq \sqrt{\frac{L}{\epsilon}}$, it follows that the number of iterations needed to reach the subproblem i^* is

$$\sum_{i=1}^{i^*} K_i = O(n^{\frac{9}{2}}(i^*)^7) = O\left(\frac{n^{\frac{9}{2}} L^{\frac{7}{2}}}{\epsilon^{\frac{7}{2}}}\right).$$

It remains to apply Lemma 5.6.13 as previously done to complete the proof. □

We would like to make a few remarks about this theorem. First, one approach to satisfy the condition $K_i \geq (i+1)^6$ for any $i \in \mathbb{N}$ is to incorporate it into the stopping criterion of Algorithm 11. However, due to the limited number of iterations in practice, this condition is typically replaced by a weaker one, $K_i \geq M$, where $M > 0$ is a constant. Second, the main result of Theorem 5.6.13 establishes the rate of convergence to an ϵ-optimal point for a single run of the SSO algorithm, which is the first of its kind to the best of our knowledge. This was made possible by decomposing

the problem given in Eq (5.1) into a sequence of subproblems, each of which is solved by using carefully chosen stopping criteria and step sizes. It is worth noting that, in [71], the (ϵ, Λ)-solution of the norm of the gradient is obtained after at most $O\left(\frac{nL^2\sigma^2}{\epsilon^4}\right)$ function evaluations. Although this bound has a weaker dependence on n and L, it is worse in terms of ϵ. Third, the first term in Eq (5.31) may be significant even if it is fixed, particularly if the region where the function is convex is difficult to reach; indeed, this constant disappears when f^{β^i} is convex for every index i. Nevertheless, the bounds given represent the worst set and may be considerably smaller in practice. A way to decrease this term is to decrease the power on i in the denominator of β^i, $s_1^{i,0}$ and $s_2^{i,0}$ but this would also decrease the asymptotic rate of convergence. Finally, the process used in the SSO algorithm may be extended to other momentum-based methods and give an appealing property for these methods compared to the classical SGD.

5.7 Numerical experiments

The numerical experiments are conducted for two bounded constrained blackbox optimization problems. In order to handle the bound constraints $\mathbf{x} \in [\ell, \mathbf{u}] \subset \mathbb{R}^n$, the update given by Eq (5.9) is simply projected such that $\mathbf{x} \leftarrow \max(\ell, \min(\mathbf{x}, \mathbf{u}))$.

5.7.1 Application to a solar thermal power plant

The first stochastic test problem is SOLAR [110], which simulates a thermal solar power plant and contains several instances allowing for selection of the number of variables, the types of constraints and the objective function to optimize. All of the instances of SOLAR are stochastic and have non-convex constraints and integer variables. In this work, the algorithms developed do not deal with integer variables. Therefore, the problem is altered: all integer variables are fixed to their initial values and the problem is to obtain a feasible solution by optimizing the expectation of constraint violations over the remaining variables. Numerical experiments were conducted for the second instance of the SOLAR framework, which considers 12 variables (2 integers) and 12 constraints

$$\min_{\mathbf{x}\in[0,1]^{12}} \mathbb{E}\left[\sum_{j=1}^{m} \max(0, c_j(\mathbf{x}, \boldsymbol{\xi}))^2\right],$$

where c_j denotes the original stochastic constraints and the bound constraints have been normalized. The second instance of SOLAR is computationally expensive; a run may take between several seconds and several minutes. Therefore, the maximum number of function evaluations was set to 1000. Four algorithms were used:

- SSO, whose hyperparameters values are given in Table 5.4. The search step given in Algo-

rithm 12 was used for this experiment. A truncated version of the Gaussian gradient based estimate was used for this experiment.

Table 5.4 List of hyperparameters for the SSO algorithm.

Problem	β^i	$s_1^{i,k}$	$s_2^{i,k}$	M	q
Cifar10	$\frac{0.005}{(i+1)^2}$	$\frac{0.005}{(i+1)^{\frac{3}{2}}\sqrt{k+1}}$	$\frac{0.9}{(i+1)\ (k+1)^{\frac{1}{4}}}$	60	10
ImageNet	$\frac{0.001}{(i+1)^2}$	$\frac{0.003}{(i+1)^{\frac{3}{2}}\sqrt{k+1}}$	$\frac{0.7}{(i+1)\ (k+1)^{\frac{1}{4}}}$	100	10
Solar	$\frac{0.3}{(i+1)^2}$	$\frac{0.1}{(i+1)^{\frac{3}{2}}\sqrt{k+1}}$	$\frac{0.5}{(i+1)\ (k+1)^{\frac{1}{4}}}$	5	10

- ZO-adaMM [47] which is a ZO version of the original Adam algorithm. This algorithm appears as one of the most effective according to [47, 117] in terms of distortion value, number of function evaluations and success rate. The default parameters defined experimentally in [47] were used for this problem, except that $\beta = 0.05$ and the learning rate was equal to 0.3. Moreover, the same gradient estimator as that for ZO-signum was used to eliminate its impact on the performance.

- CMA-ES [78] an algorithm based on biologically inspired operators. Its name comes from the adaptation of the covariance matrix of the multivariate normal distribution used during the mutation. The version of CMA-ES used was the one of the pymoo [35] library with the default setting.

- The NOMAD 3.9.1 software [106], based on the MADS [12] algorithm, a popular blackbox optimization solver.

The results are presented in Figure 5.2, which plots the average best result obtained by each algorithm with five different seeds. In this experiment, SSO achieved similar performance to NOMAD and CMA-ES which are state-of-the-art algorithms for this type of problem. ZO-adaMM had difficulty converging even though it is a ZO algorithm.

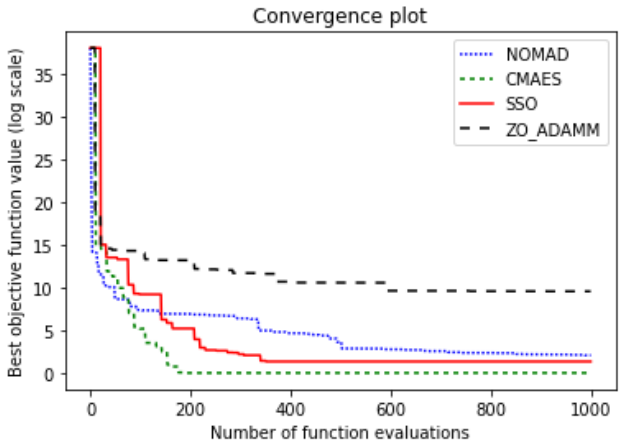

Figure 5.2 Average of five different seed runs for the NOMAD, CMAES, SSO and ZO-adaMM algorithms.

5.7.2 Application to blackbox adversarial attack

This section demonstrates the competitiveness of the SSO algorithm through experiments involving the generation of blackbox adversarial examples for deep neural networks (DNNs) [192]. Generating an adversarial example for a DNN involves adding a well-designed perturbation to the original legal input to cause the DNN to misclassify it. In this work, the attacker considers the DNN model to be unknown, hence the term blackbox. Adversarial attacks against DNNs are not just theoretical, they pose a real safety issue [144]. Having an algorithm that generates effective adversarial examples enables modification of DNN architecture to enhance its robustness against such attacks. An ideal adversarial example is one that can mislead a DNN to recognize it as any target image label, while appearing visually similar to the original input, making the perturbations indiscernible to human eyes. The similarity between the two inputs is typically measured by an ℓ_p norm. Mathematically, a blackbox adversarial attack can be formalized as follows. Let $(\mathbf{y}, \ell)$ denote a legitimate image $\mathbf{y}$ with the true label $\ell \in [1, M]$, where M is the total number of image classes. Let $\mathbf{x}$ denote the adversarial perturbation; the adversarial example is then given by $\mathbf{y}' = \mathbf{y} + \mathbf{x}$, and the goal is to solve the problem [47]

$$\begin{aligned} &\min_{\mathbf{x}} \lambda f(\mathbf{y}+\mathbf{x}) + ||\mathbf{x}||_2, \\ &\text{subject to } \ (\mathbf{y}+\mathbf{x}) \in [-0.5, 0.5]^n, \end{aligned}$$

where $\lambda > 0$ is a regularization parameter and f is the blackbox attack loss function. In our experiments, $\lambda = 10$ and the loss function is defined for an untargeted attack [42], i.e.,

$$f(\mathbf{y}') = \max\{Z(\mathbf{y}')_\ell - \max_{j \neq \ell} Z(\mathbf{y})_j, 0\},$$

where $Z(\mathbf{y}')_k$ denotes the prediction score for class k given the input $\mathbf{y}'$. Thus, the minimum value of 0 is reached as the perturbation succeeds to fool the neural network.

The experiments of generating blackbox adversarial examples were first performed by using an adapted AlexNet [98] on the dataset Cifar10 and then by using InceptionV3 [180] on the dataset ImageNet [59]. Since the NOMAD algorithm is not recommended for large problems, three algorithms are compared: SSO (without search), ZO-adaMM and CMA-ES. In the experiments, the hyperparameters of the ZO-adaMM algorithm were taken as in [47], and those of SSO are given in Table 5.4; the uniform gradient based estimate is used for both algorithms. Moreover, for the Cifar10 dataset, different initial learning rates for ZO-adaMM were used to observe its influence on the success rate. Experiments were conducted for 100 randomly selected images with a starting point corresponding to a null distortion; the maximum number of function queries was set to 5000. Thus, as the iteration increases, the attack loss decreases until it converges to 0 (indicating a successful attack) while the norm of the distortion could increase.

The best attack performance involves a trade-off between a fast convergence to a 0 attack loss in terms of function evaluations, a high rate of success, and a low distortion (evaluated by the ℓ_2-norm). The results for the Cifar10 dataset are given in Table 5.5.

Table 5.5 Results of blackbox adversarial attack for the Cifar10 dataset ($n = 3 \times 32 \times 32$).

Method	Attack success rate	$\|\|\ell_2\|\|$ first success	Average # of function evaluations
ZO-adaMM $lr = 0.01$	79 %	0.14	582
ZO-adaMM $lr = 0.03$	96%	0.97	310
ZO-adaMM $lr = 0.05$	98%	2.10	215
CMAES $\sigma = 0.005$	99%	0.33	862
SSO	100%	0.55	442

Except for ZO-adaMM with an initial learning rate equal to 0.01, all algorithms achieved a success rate above 95%. Among these algorithms, ZO-adaMM with a learning rate equal to 0.05, had the best convergence rate in terms of function evaluations but it had the worst value of distortion. On the contrary, CMA-ES obtained the best value of distortion but had the worst convergence rate. The SSO algorithm achieved balanced results, and it was the only one to reach full success rate.

Table 5.6 displays the results for the ImageNet dataset. Only two algorithms are compared since dimensions were too large to invert the covariance matrix in CMA-ES. For this dataset, ZO-adaMM and SSO had the same convergence rate. However, SSO outperformed ZO-adaMM in terms of success rate while having a slightly higher level of distortion.

Table 5.6 Results of blackbox adversarial attack for the ImageNet dataset ($n = 3 \times 299 \times 299$).

Method	Attack success rate	$\|\|\ell_2\|\|$ first success	Average # of function evaluations
ZO-adaMM $lr = 0.01$	59 %	19	1339
SSO	73 %	33	1335

5.8 Concluding remarks

This paper presents a method for computationally expensive stochastic blackbox optimization. The approach uses ZO gradient estimates, which provides three advantages. First, they require few function evaluations to estimate the gradient, regardless of the problem's dimensions. Second, under mild conditions on the noised objective function, the problem is formulated as optimization of a smooth approximation. Third, the smooth approximation may appear to be locally convexified near a local minima.

Based on these three features, the SSO algorithm was proposed. This algorithm is a sequential one and comprises two steps. The first is an optional search step that improves the exploration of the decision variable space and the algorithm's efficiency. The second is a local search, which ensures the convergence of the algorithm. In this step, the original problem is decomposed into subproblems solved by a ZO-version of a sign stochastic descent with momentum algorithm. More specifically, when the momentum's norm falls below a specified threshold that depends on the smoothing parameter, the subproblem is considered solved. The smoothing parameter's value is then decreased, and the SSO algorithm moves on to the next subproblem.

A theoretical analysis of the algorithm has been conducted. Under Lipschitz continuity of the stochastic ZO oracle, a convergence rate in mean of the ZO-signum algorithm is derived. Under additional assumptions of smoothness and convexity or local convexity of the objective function near its minima, the rate of convergence of the SSO algorithm to an ϵ-optimal point of the problem has been derived, which is, to the best of our knowledge, the first of its kind.

Finally, numerical experiments were conducted based on a solar power plant simulation and adversarial blackbox attacks. Both examples were computationally expensive, the former was a small-sized problem ($n \approx 10$) and the latter was a large-sized problem (up to $n \approx 10^5$). The results

demonstrate the SSO algorithm's competitiveness in terms of both performance and convergence rate compared to state-of-the-art algorithms. Further work will extend this approach to constrained stochastic optimization.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgements

The authors are grateful to the anonymous reviewers for their helpful comments and the English editor for improving the English quality of this text.

This work was financed by the IVADO Fundamental Research Projects Grant PRF-2019-8079623546 and by the NSERC Alliance grant 544900-19 in collaboration with Huawei-Canada.

Conflict of interest

All authors declare no conflicts of interest that may influence the publication of this paper.

5.9 Appendix

5.9.1 Appendix A. Notations

The following list describes symbols used within the body of the document. Throughout the paper, when a symbol is shown in bold then it is a vector; otherwise, it is a scalar.

n	The dimension of the space of the design variables
Ω	The sample space of $\boldsymbol{\xi}$, i.e., the set of all possible outcomes of $\boldsymbol{\xi}$
$\boldsymbol{\xi} : \Omega \to \mathbb{R}^m$	The vector of uncertainties
$\mathbb{E}_{\boldsymbol{\xi}}[\cdot]$	The expectation with respect to the random vector $\boldsymbol{\xi}$
$F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$	The stochastic zeroth-order oracle that takes into account the uncertainty $\boldsymbol{\xi}$
$f : \mathbb{R}^n \to \mathbb{R}$	The expectation of F with respect to $\boldsymbol{\xi}$
$\beta \in \mathbb{R}^{+*}$	A strictly positive scalar for use as a smoothing parameter
$\mathbf{u} \in \mathbb{R}^n$	A Gaussian random vector
$f^\beta = \mathbb{E}[f(\mathbf{x} + \beta\mathbf{u})]$	A smooth approximation of a function f
$L_0(f)$	The Lipschitz constant associated with a function f
$L_1(f)$	The Lipschitz constant associated with the gradient of a function f
∇f	The gradient of a function f
$\tilde{\nabla} f$	An estimator of the gradient of a function f
$\tilde{\mathbf{g}}$	An estimator of the gradient of a function f based on outputs of the stochastic zeroth-order oracle $F(\mathbf{x}, \boldsymbol{\xi})$
$j \in [1, n]$	The counter associated with the dimension
$i \in \mathbb{N}$	The outer iteration counter associated with a subproblem
$k \in \mathbb{N}$	The inner iteration counter
$\mathbf{m} \in \mathbb{R}^n$	The momentum vector
$s_2^{i,k} \in (0, 1)$	The step size associated with the momentum
$s_1^{i,k} \in (0, 1)$	The step size associated with $\mathbf{x}$
$L \in \mathbb{R}^{+*}$	An approximation of the Lispchitz constant
$q \in \mathbb{N}$	The size of the mini batch used to estimate $\tilde{\nabla}$
$M \in \mathbb{N}$	The minimum number of iterations used in the ZO-signum algorithm
$H_k^{(\alpha)}$	The generalized harmonic number of order α
$\mathcal{C}^{0+}$	Class of Lipschitz continuous functions
$\mathcal{C}^{1+}$	Class of differentiable functions whose gradient is Lipschitz
$\mathcal{C}^{\infty}$	Class of infinitely differentiable functions

5.9.2 Appendix B. Proof of Proposition 5.6.1

Proposition 5.9.1 ([31]). *For the subproblem $i \in \mathbb{N}$, under Assumption 1 and in the setting of Algorithm 11, we have*

$$\begin{aligned} s_1^{i,k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \leq \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k}) - f^{\beta^i}(\mathbf{x}^{i,k+1})] + \frac{nL_1(f^{\beta^i})}{2}(s_1^{i,k})^2 \\ + 2s_1^{i,k}\underbrace{\mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1]}_{\textit{bias}} + 2s_1^{i,k}\sqrt{n}\sqrt{\underbrace{\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||_2^2]}_{\textit{variance}}}, \end{aligned} \tag{B.32}$$

where $\bar{m}_j^{i,k+1}$ is defined recursively as $\bar{m}_j^{i,k+1} = s_2^{i,k}\nabla f^{\beta^i}(\mathbf{x}^{i,k}) + (1 - s_2^{i,k})\bar{m}_j^{i,k}$.

Proof. By $L_1(f^{\beta^i})$-Lipschitz smoothness of f^{β^i} (see Lemma 5.4.1(3)), it follows that

$$\begin{aligned} f^{\beta^i}(\mathbf{x}^{i,k+1}) &\leq f^{\beta^i}(\mathbf{x}^{i,k}) + \langle \nabla f^{\beta^i}(\mathbf{x}^{i,k}), \mathbf{x}^{i,k+1} - \mathbf{x}^{i,k}\rangle + \frac{L_1(f^{\beta^i})}{2}||\mathbf{x}^{i,k+1} - \mathbf{x}^{i,k}||_2^2 \\ &= f^{\beta^i}(\mathbf{x}^{i,k}) - s_1^{i,k}\langle \nabla f^{\beta^i}(\mathbf{x}^{i,k}), \text{sign}(\mathbf{m}^{i,k+1})\rangle + \frac{L_1(f^{\beta^i})\,(s_1^{i,k})^2}{2}||\text{sign}(\mathbf{m}^{i,k+1})||_2^2 \\ &= f^{\beta^i}(\mathbf{x}^{i,k}) - s_1^{i,k}||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1 + \frac{nL_1(f^{\beta^i})}{2}(s_1^{i,k})^2 \\ &+ 2s_1^{i,k}\sum_{j=1}^{n}|\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|\mathbf{1}\{\text{sign}(m_j^{i,k+1}) \neq \text{sign}(\nabla_j f^{\beta^i}(\mathbf{x}^{i,k}))\}, \end{aligned}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Now, as in [31, 116], the expected improvement conditioned on $\mathbf{x}^{i,k}$ is given by

$$\begin{aligned} \mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k+1}) - f^{\beta^i}(\mathbf{x}^{i,k})|\mathbf{x}^{i,k}] \leq -s_1^{i,k}||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1 + \frac{nL_1(f^{\beta^i})}{2}(s_1^{i,k})^2 \\ + 2s_1^{i,k}\sum_{j=1}^{n}|\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|\mathbb{E}[\mathbf{1}\{\text{sign}(m_j^{i,k+1}) \neq \text{sign}(\nabla_j f^{\beta^i}(\mathbf{x}^{i,k}))\}|\mathbf{x}^{i,k}]. \end{aligned} \tag{B.33}$$

Again, as in [31, 116], the expectation that the sign of $m_j^{i,k+1}$ is different from the sign of $\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})$ is relaxed by considering that the set

$$\{m_j^{i,k+1} : \text{sign}(m_j^{i,k+1}) \neq \text{sign}(\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})\} \subset \{m_j^{i,k+1} : \ |m_j^{i,k+1} - \nabla_j f^{\beta^i}(\mathbf{x}^{i,k})| \geq |\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|\}.$$

Therefore, it follows that

$$\mathbb{E}[\mathbf{1}\{\text{sign}(m_j^{i,k+1}) \neq \text{sign}(\nabla_j f^{\beta^i}(\mathbf{x}^{i,k}))\}|\mathbf{x}^{i,k}] \leq \mathbb{E}[\mathbf{1}\{|m_j^{i,k+1} - \nabla_j f^{\beta^i}(\mathbf{x}^{i,k})| \geq |\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|\}|\mathbf{x}^{i,k}] \leq \frac{\mathbb{E}[|m_j^{i,k+1} - \nabla_j f^{\beta^i}(\mathbf{x}^{i,k})| \ |\mathbf{x}^{i,k}]}{|\nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|}, \tag{B.34}$$

where the second inequality comes from the conditional Markov's inequality. Substituting Eq (B.34) into Eq (B.33) and taking the expectation over all of the randomness we obtain

$$\begin{aligned}\mathbb{E}[f^{\beta^i}(\mathbf{x}^{i,k+1}) - f^{\beta^i}(\mathbf{x}^{i,k})] \leq & - s_1^{i,k}\mathbb{E}[||\nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] + \frac{nL}{2}(s_1^{i,k})^2 \\ & + 2s_1^{i,k}\sum_{j=1}^{n}\mathbb{E}[|m_j^{i,k+1} - \nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|].\end{aligned} \tag{B.35}$$

Moreover, by adding and subtracting $\bar{\mathbf{m}}^{i,k+1}$ in terms of the sum of Eq (B.35), one gets

$$\begin{aligned}\sum_{j=1}^{n}\mathbb{E}[|m_j^{i,k+1} - \nabla_j f^{\beta^i}(\mathbf{x}^{i,k})|] &= \mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1} + \bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \\ &\leq \sqrt{n}\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||_2] + \mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1] \\ &\leq \sqrt{n}\sqrt{\mathbb{E}[||\mathbf{m}^{i,k+1} - \bar{\mathbf{m}}^{i,k+1}||_2^2]} + \mathbb{E}[||\bar{\mathbf{m}}^{i,k+1} - \nabla f^{\beta^i}(\mathbf{x}^{i,k})||_1],\end{aligned}$$

where the first inequality comes from $||\cdot||_1 \leq \sqrt{n}||\cdot||_2$ and the second one from Jensen's inequality. Finally, incorporating the last inequality in Eq (B.35) completes the proof. □

5.9.3 Appendix C. Original signSGD and signum algorithms

Below are the original versions of the signSGD and signum algorithms.

Algorithm 13 signSGD algorithm.

1: **Input:** $\mathbf{x}^0$, $s_1 \in (0, 1)$
2: **for** $k = 0, 1, \ldots$ **do**
3: Calculate an estimate of the stochastic gradient $\tilde{\nabla} f(\mathbf{x}^k)$ and update:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - s_1 \text{sign}(\tilde{\nabla} f(\mathbf{x}^k))$$

4: **end for**
5: Return $\mathbf{x}^k$

Algorithm 14 Signum algorithm.

1: **Input:** $\mathbf{x}^0, \mathbf{m}^0, s_1 \in\ (0,1), s_2 \in\ (0,1)$

2: **for** $k = 0, 1, \ldots$ **do**

3: Calculate an estimate of the stochastic gradient $\tilde{\nabla} f(\mathbf{x}^k)$ and update:

$$\mathbf{m}^{k+1} = s_2 \mathbf{m}^k + \ (1 - s_2)\tilde{\nabla} f(\mathbf{x}^k)$$
$$\mathbf{x}^{k+1} = \mathbf{x}^k - s_1 \text{sign}(\mathbf{m}^{k+1})$$

4: **end for**

5: Return $\mathbf{x}^k$

5.9 Addenda

The previous paper was presented at the MOPTA 2023 conference and several remarks were made to add a theoretical result and numerical experiments. First, a corollary that completes Theorem 5.6.17 is stated and proved. Second, additional experiments are conducted to compare the SSO algorithm and the ZO-Signum algorithms in order to study the practical implications of sequential optimization. Finally, experiments are performed on blackbox adversarial attacks in a stochastic context. The blackbox adversarial attack presented in [47] was indeed deterministic, so the same problem was used in our paper to conform to the state of the art. Here we have more freedom to perform the numerical experiments.

5.9.1 An additional theoretical result

First, note that a typo was made in Theorem 5.6.17, the norm of the gradient must be taken at $\mathbf{x}^{i^*+1,0}$ and not at $\mathbf{x}^{i^*,0}$. Since the blackbox f is assumed to be $\mathcal{C}^1$ in the theorem, the result obtained for the function $f^{\beta^{i^*}}$ can be extended to the original blackbox f. This is the subject of the next corollary.

Corollary 5.9.1. *Under the same statement as Theorem 5.6.17 except that* $\beta^i = \frac{1}{(n+3)^{\frac{3}{2}}(i+1)^2}$, *then after at most*

$$O\left(\frac{n^6 L^{7/2}}{\epsilon^{7/2}}\right)$$

function evaluations, the following inequality holds

$$||\nabla f(\mathbf{x}^{i^*+1,0})|| \leq \epsilon.$$

Proof. By the triangular inequality, it follows that

$$\begin{aligned} ||\nabla f(\mathbf{x}^{i^*+1,0})|| &\leq ||\nabla f^{\beta^{i^*}}(\mathbf{x}^{i^*+1,0})|| + ||\nabla f(\mathbf{x}^{i^*+1,0}) - \nabla f^{\beta^{i^*}}(\mathbf{x}^{i^*+1,0})|| \\ &\leq \epsilon + L\beta^{i^*}\,(n+3)^{\frac{3}{2}}, \end{aligned}$$

where the second inequality is due to Theorem 5.6.17 and Lemma 5.4.1.3. Now, as $i^*+1 \geq \sqrt{\frac{L}{\epsilon}}+1$, it follows that

$$L\beta^{i^*}\,(n+3)^{\frac{3}{2}} \leq \epsilon.$$

The result of the corollary follows directly. □

5.9.2 Comparison between the SSO and the ZO-Signum algorithms

The ZO-Signum algorithm is used to solve each subproblem of the SSO algorithm. It can be seen as an SSO algorithm where the hyperparameter M is fixed to infinity. Therefore, that remains to compare two SSO algorithms with a different hyperparameter value. In this case, the simplest way to compare the two versions of an algorithm is to use a benchmark of analytical test problems and present data profiles [138].

First, data profiles must be adapted for unconstrained stochastic optimization. Data profiles can be used to assess whether the algorithms are successful in generating solution values close to the best objective function values. To identify a successful run, a convergence test is required. In a deterministic case, this test is based on the best function value found by the algorithm. However, if the function is stochastic, the best function value is inappropriate. Here, an estimated mean of the function value is used for the convergence test. Let μ^e be the best estimated mean obtained by an algorithm on a problem after e evaluations, μ^0 be the mean of the function at the starting point, and μ^* be the best mean obtained by all tested algorithms on all instances of the problem. Then the problem is said to be solved with respect to the mean within the convergence tolerance τ if

$$\mu^0 - \mu^e \geq (1-\tau)\,(\mu^0 - \mu^*).$$

An instance of a problem corresponds to a particular seed of a pseudorandom generator. The number of samples used for the mean estimation must be adapted to the desired convergence tolerance. In this work, 1000 independent identically distributed samples of $\boldsymbol{\xi}$ are used to estimate the mean. Therefore, the convergence tolerances used are all greater than 0.01. Finally, the horizontal axis of a data profile represents groups of $n+1$ evaluations. The vertical axis corresponds to the proportion of problems solved within a given tolerance τ. Each algorithm has its curve to compare the algorithms' ability to converge to the best mean.

To compare the two algorithms, 16 test problems from the literature are used. These test problems can be grouped into 4 classes of problems. Each problem consists of an objective function and a distribution of uncertainties. The vector $\boldsymbol{\xi}$ is composed of independent, identically distributed variables. Here are the characteristics and the origin of the different test problems

- Stochastic Rosenbrock problem [23]

$$f_1(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{n-1} \left(10(x_{i+1} + \xi_{i+1} - (x_i + \xi_i)^2) + \xi_{n+i}\right)^2 + \left((1 - (x_i + \xi_i)) + \xi_{2n-1+i}\right)^2.$$

- Stochastic Powell problem [199]

$$f_2(\mathbf{x},\boldsymbol{\xi}) = \sum_{i=1}^{n/4}\Big((x_{4i-3}+\xi_{4i-3}+10(x_{4i-2}+\xi_{4i-2}))^2 + 5(x_{4i-1}+\xi_{4i-1} - (x_{4i}+\xi_{4i})^2 + (x_{4i-2}+\xi_{4i-2}-2(x_{4i-1}+\xi_{4i-1}))^4 + 10(x_{4i-3}+\xi_{4i-3} - (x_{4i}+\xi_{4i})^4\Big) + \xi_{n+1}\sqrt{1+100\sum_{i=1}^{n}(x_i-1)^2}.$$

- Stochastic Levy problem [199]

$$f_3(\mathbf{x},\boldsymbol{\xi}) = \sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i-1)^2[1+10\sin^2(\pi w_i+1)] + (w_n-1)^2(1+\sin^2(2\pi w_n)) + \xi_{n+1}\sqrt{1+10\sum_{i=1}^{n}(x_i-2)^2}$$

with:

$$w_i = 1+\frac{x_i+\xi_i-1}{4}.$$

- Stochastic Schwefel problem

$$f_4(\mathbf{x},\boldsymbol{\xi}) = 418.9829n + \sum_{i=1}^{n}(x_i+\xi_i)\sin(\sqrt{|x_i+\xi_i|})] + \frac{\xi_{n+1}}{100}\sqrt{1+100\sum_{i=1}^{n}(x_i-1)^2}.$$

Table 5.7 Problem parameters and uncertainties distributions

	n	$\mathbf{x}^0$	Bounds	$\boldsymbol{\xi}_i$, $i \in [1,n]$	$\boldsymbol{\xi}_j$, $j \geq n$
f_1	$\{2,10,50,100\}$	$[-1.2,1]^{n/2}$	$[-1.5,1.5]^n$	$\mathcal{U}([-0.25,0.25])$	$\mathcal{U}([-3,3]^{2n-2})$[1]
f_2	$\{4,12,20,40\}$	$[3.25,4.6]^{n/2}$	$[-4,5]^n$	$\mathcal{B}(2,2)$ [2]	$\mathcal{U}([-4,4])$
f_3	$\{2,10,20,50\}$	$[-7.2,9.6]^{n/2}$	$[-10,10]^n$	$\mathcal{VM}(0,4)$[3]	$\mathcal{U}([-3,3])$
f_4	$\{2,10,20,50\}$	$[305,305]^{n/2}$	$[200,500]^n$	$\mathcal{T}(-0.5,0.1,0.5)$[4]	$\mathcal{U}([-3,3])$

[1] $\mathcal{U}$ the uniform distribution,

[2] $\mathcal{B}(2,2)$ the beta distribution,

[3] $\mathcal{VM}(0,4)$ the Von Mises distribution,

[4] $\mathcal{T}(0.5,1,0)$ the triangular distribution.

The data profiles are computed using the previous benchmark with 5 different seeds for each problem. For each value of $\tau \in \{0.1, 0.01\}$, the data profiles are plotted in terms of the mean. In

order to better illustrate the results according to the dimension of the problem, the benchmark of test problems is divided into two groups: one with problem size inferior to 10 and another with problem size strictly superior to 10. Each problem is run with ten different random seeds to reduce the randomness. In addition, the results of the NOMAD software, the CMA-ES algorithm, and the ZO-adaMM algorithm are presented. The hyperparameter values for the different algorithms are as follows:

- SSO:

$$\beta^i = \frac{\min\{20,\ (u_0 - \ell_0)/3\}}{(i+1)^2}, s_1^{i,k} = \frac{\min\{60,\ (u_0 - \ell_0)/3\}}{\sqrt{n}(i+1)^{\frac{3}{2}}\sqrt{k+1}},\ s_2^{i,k} = \frac{0.7}{(i+1)(k+1)^{\frac{1}{4}}},$$
$$M = \frac{K^{\max}}{5}, q = 2;$$

- ZO-Signum:

$$\beta = \min\{20,\ (u_0 - \ell_0)/3\},\ s_1^k = \frac{\min\{60,\ (u_0 - \ell_0)/3\}}{\sqrt{n}\sqrt{k+1}},\ s_2^k = \frac{0.7}{(k+1)^{\frac{1}{4}}},\ M = \infty,\ q = 2;$$

- ZO-adaMM:

$$\beta = \frac{\min\{20,\ (u_0 - \ell_0)/3\}}{\sqrt{n}},\ lr = \min\{200, 20(u_0 - \ell_0)\},\ \beta^1 = 0.9,\ \beta^2 = 0.7,\ q = 2;$$

- CMA-ES: $\beta = \frac{(u_0 - \ell_0)}{30}$;

- NOMAD version 3.9 [106] with the Robust-MADS option [15];

where $K^{\max}$ is the maximum number of iterations assigned to a problem, and u_0 and ℓ_0 are the first variables of the lower and upper bounds. In the experiments, $K^{\max}$ is 500 for problems of size $n = 2$, 1000 for problems of size $n = 4$, 2500 for problems of size $n = 10$, 3000 for problems of size $n = 12$, and 10000 for $n \geq 20$.

Results on test problems with dimension $n \leq 10$ are shown in figure 5.3. For $\tau = 0.1$ it appears that the SSO and ZO-Signum algorithms have the same curves. This is because the convergence tolerance is too small to detect a difference between the two algorithms. However, both algorithms outperform the other algorithms, either in terms of efficiency (i.e., solving as many problems as possible in as few function evaluations as possible) or consistency (i.e., having a large percentage of problems solved within the allotted budget of evaluations). However, with a finer convergence tolerance, the results change drastically. While in terms of efficiency, the R-MADS algorithm performs better than a group composed of the SSO, ZO-Signum, and ZO-adaMM algorithms, in

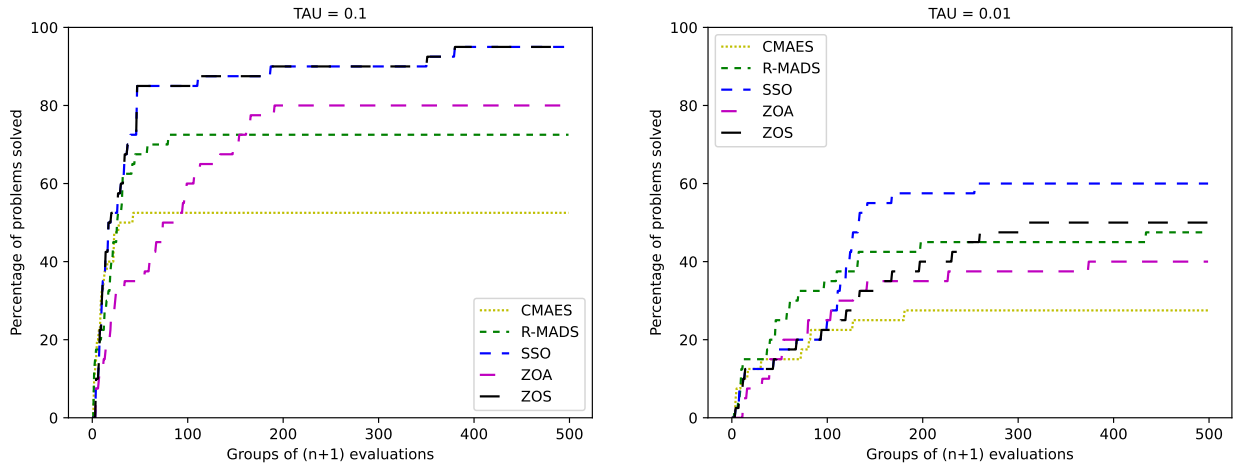

Figure 5.3 Results on test problems with $n \leq 10$ for $\tau = 0.1$ (left) and $\tau = 0.01$ (right) with 5 different seeds for each problem.

terms of consistency, the SSO algorithm outperforms the R-MADS, ZO-Signum, and ZO-adaMM algorithms, which achieve similar results.

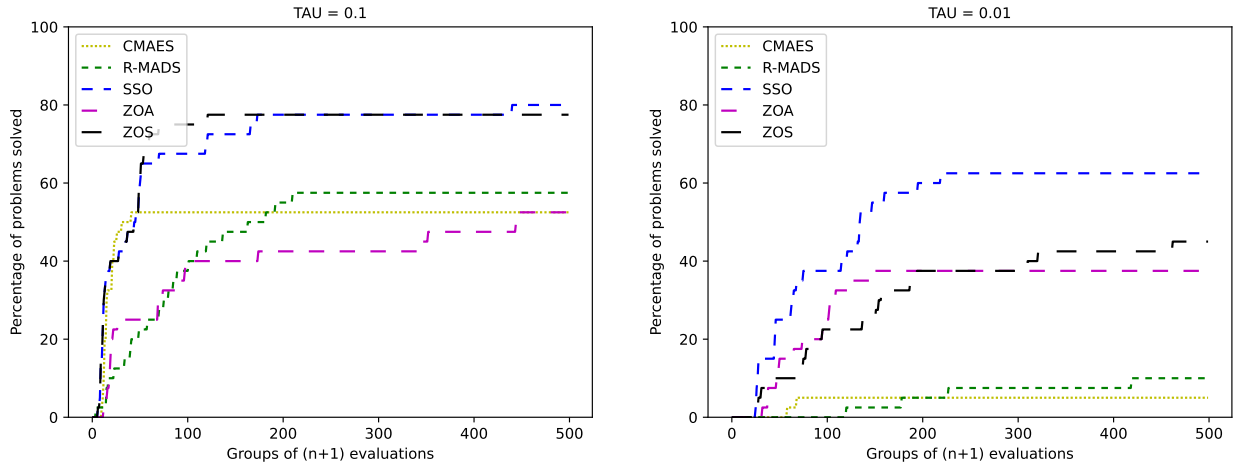

Figure 5.4 Results on test problems with $n \geq 10$ for $\tau = 0.1$ (left) and $\tau = 0.01$ (right) with 5 different seeds for each problem.

Results on test problems with dimension $n \geq 10$ are shown in figure 5.4. These results confirm and reinforce the previous results. For a convergence tolerance of $\tau = 0.1$, the SSO and ZO-Signum algorithms perform equally well and outperform the other algorithms. For a finer convergence tol-

erance, the SSO algorithm outperforms all other algorithms, while the ZO-type algorithms perform equally well and outperform the R-MADS and CMA-ES algorithms. To summarize the results, it appears that in small dimension, the R-MADS algorithm performs better, although the ZO-type algorithms achieve satisfactory results. In large dimension, the ZO-type algorithm achieves a better result than the R-MADS and CMA-ES algorithms. In both cases, the SSO algorithm allows to obtain better optimal values thanks to its sequential optimization process.

5.9.3 Universal targeted blackbox adversarial attack

This subsection completes the experiments on generating adversarial examples for DNNs. Two main changes are made to the experiments performed in section Section 5.7.2. First, the problem is targeted. This means that the perturbation must be designed so that the DNN misclassifies the adversarial image as a pre-specified target label. Second, the problem is made stochastic. This is done by selecting a set of 15 images that have been correctly classified by the DNN. Then, instead of looking for a perturbation for a single image, the perturbation must be designed for the 15 images. For this purpose, it is assumed that the gradient can only be estimated for a randomly chosen image among the 15. Therefore, compared to Section 5.7.2, the goal of the algorithm is to solve the following problem

$$\begin{aligned} &\min_{\mathbf{x}} \lambda \mathbb{E}[f(\mathbf{y}_i + \mathbf{x})] + ||\mathbf{x}||_2, \\ &\text{subject to } \ (\mathbf{y}_i + \mathbf{x}) \in [-0.5, 0.5]^n, \end{aligned}$$

where i is taken randomly uniformly in $\{1, \ldots, 15\}$. Moreover, as the problem is for targeted attack [42], the function f is also slightly modified

$$f(\mathbf{y}_i') = \max\{\max_{j \neq \ell'} Z(\mathbf{y}_i')_j - Z(\mathbf{y}_i')_{\ell'} -, 0\},$$

where $\mathbf{y}_i' = \mathbf{y}_i + \mathbf{x}$ and $\mathbf{y}_i$ is an image correctly classified as label ℓ, $\ell' \neq \ell$ is the target label, and $Z(\mathbf{y}_i')_j$ is the prediction score of class j given input $\mathbf{y}_i'$.

The results are given in Table 5.8 for a set of 15 images of the Cifar10 dataset and a budget of function evaluations fixed to 50000. In this experiment, the origin label is 5, corresponding to images of dogs, while the target label is 3, corresponding to images of cats. The hyperparameter values are set to the same values as in the deterministic adversarial attack experiment, except that $s_1^0 = 0.01$, $s_2^0 = 0.3$, and $M = 500$. The results show that the SSO algorithm achieves better results than the ZO-adaMM algorithm on the 15 selected images, but has a similar success rate when all correctly classified images are considered. Furthermore, the norm of the distortion is smaller for

the SSO algorithm than for the ZO-adaMM algorithm. Although the results are promising, further experiments are needed to confirm them.

Table 5.8 Results of stochastic targeted blackbox attack for the Cifar10 dataset $n = 3 \times 32 \times 32$

Method	% of 15 images classed as targeted label	% of the correctly classified images classed as targeted label	$\|\|\ell_2\|\|$ final
ZO-adaMM $lr = 0.05$	46.66	10.8	3.39
SS0	86.66	11.3	2.90

CHAPTER 6 ARTICLE 3: RISK-AVERSE CONSTRAINED BLACKBOX OPTIMIZATION UNDER MIXED ALEATORY/EPISTEMIC UNCERTAINTIES

Title Risk averse constrained blackbox optimization under mixed aleatory/epistemic uncertainties

Authors Charles Audet, Jean Bigeon, Romain Couderc and Michael Kokkolaras

Journal Submitted for publication to Computational Optimization and Applications on 29/10/2023.

6.1 Context

The third contribution of this thesis addresses the problem given in Equation (1.4). This problem is very general, as it allows to consider any type of uncertainty. As presented in Section 2.3.1, there are two main types of uncertainty. Dealing with both types is particularly challenging in a BBO context. For example, several papers on this topic are of limited interest because they focus only on the reliability analysis problem [4, 139]. For a given point $\mathbf{x}$, these methods try to efficiently approximate the uncertain constraints at that point, and no hint is given on how to handle an uncertain objective function. Furthermore, another drawback of these methods is that they are heuristic based and do not benefit from convergence analysis. Although this does not question the efficiency of these methods in practice, it may limit their legitimacy to be used.

In this contribution, a CVaR-based approach allows to deal with both types of uncertainty [112]. In fact, if a risk level sufficiently close to 1 is desired, this approach mimics the worst-case approach for epistemic uncertainty. However, in a BBO context, estimating the CVaR of the objective and/or constraint functions is challenging. Monte Carlo simulation is usually prohibitively expensive, requiring at least $\frac{1}{1-\alpha}$ samples to get a satisfactory estimate of CVaR. As in the previous paper, multi-timescale stochastic approximation algorithms have been developed to avoid computationally expensive sampling [50, 152]. This algorithm cannot be directly applied in our case because a gradient estimate is computed based on Markov decision process properties. Therefore, in this work, Gaussian exploration is used to smooth the problem and obtain approximate derivatives of the problem, as in the previous contribution. However, a new problem arises since the classical Gaussian distribution cannot be used in the case where bound constraints are not relaxable in the sense of [107]. Therefore, an approximation based on the truncated Gaussian distribution is developed. Finally, thanks to all the previous elements, we prove the almost sure convergence of our algorithm to a feasible point of the CVaR-constrained problem whose objective function value

is arbitrarily close to that of a local solution. This proof is based on classical results from multi-timescale stochastic approximation and Lyapunov theory.

6.2 Article

Abstract This paper addresses risk averse constrained optimization problems where the objective and constraint functions can only be computed by a blackbox subject to unknown uncertainties. To handle mixed aleatory/epistemic uncertainties, the problem is transformed into a conditional value-at-risk (CVaR) constrained optimization problem. General inequality constraints are managed through Lagrangian relaxation. A convolution between a truncated Gaussian density and the Lagrangian function is used to smooth the problem. A gradient estimator of the smooth Lagrangian function is derived, possessing attractive properties: it estimates the gradient with only two outputs of the blackbox, regardless of dimension, and evaluates the blackbox only within the bound constraints. This gradient estimator is then utilized in a multi-timescale stochastic approximation algorithm to solve the smooth problem. Under mild assumptions, this algorithm almost surely converges to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution. Finally, numerical experiments are conducted to serve three purposes. Firstly, they provide insights on how to set the hyperparameter values of the algorithm. Secondly, they demonstrate the effectiveness of the algorithm when a truncated Gaussian gradient estimator is used. Lastly, they show its ability to handle mixed aleatory/epistemic uncertainties in practical applications.

Keywords: Risk averse optimization; constrained blackbox optimization; multi-timescale stochastic approximation; conditional value-at-risk; mixed aleatory/epistemic uncertainties; truncated Gaussian gradient estimator

6.3 Introduction

Blackbox optimization (BBO) is concerned with optimization problems where the functions used to compute the objective and the constraints are blackboxes. In optimization, a blackbox is any process that returns an output when an input is provided, but the inner workings of that process are not analytically available [14]. This type of problem is common in signal processing [55], machine learning [144], and engineering design [2, 97]. In the presence of uncertainties, a constrained blackbox optimization problem may be formulated as follows

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \Xi_0[C_0(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \Xi_j[C_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in [1,m], \end{array} \tag{6.1}$$

where $\mathbf{x}$ is the vector of the design variables, $\mathcal{X} := [\mathbf{b}_\ell, \mathbf{b}_u]$ is a hyperectangle, and $\boldsymbol{\xi}$ is the vector modelling the uncertainties. The source of uncertainties may arise from the design variables, the

parameters, the inner processes of the blackbox (for example, when Monte Carlo simulation is used in the blackbox), or even combinations of these factors. Uncertainties may or may not depend on $\mathbf{x}$. $C_0(\cdot, \boldsymbol{\xi})$ denotes the version of the objective function $c_0 : \mathcal{X} \to \mathbb{R}$ subject to uncertainties, while for all $j \in \{1, 2, \ldots, m\}, C_j(\cdot, \boldsymbol{\xi})$ denotes the version of the constraint $c_j : \mathcal{X} \to \mathbb{R}$ subject to the uncertainties (also called the limit state function in the reliability community). Since the objective function and the constraints depend on the uncertainty vector, the measures $\Xi_j, j \in \{0, 1, \ldots, m\}$ are used to map them into $\mathbb{R}$. It follows from this formulation that the key factor is the selection of the uncertainties model, which in turn determines the choice of the measures Ξ_j. In the following, various methods commonly found in the literature are presented, depending on the assumptions made, the chosen uncertainty model, and the level of information available about these uncertainties.

6.3.1 Related work

In probabilistic reliability-based design optimization (RBDO), uncertainties are considered as random vectors with known probabilistic distributions. In this field [57], Problem (6.1) is transformed into the following

$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n} & C_0(\mathbf{x}, \mathbf{p}) \\ \text{s.t.} & \mathbb{P}[C_j(\mathbf{x}, \mathbf{p}) \leq 0] \geq \alpha_j, \ \forall j \in [1, m], \end{array} \tag{6.2}$$

where $\alpha_j, j \in \{1, 2, \ldots, m\}$ are the desired reliability levels and $\mathbf{x}$ and $\mathbf{p}$ are the means of the noised design variables and parameters respectively. In this reformulation, the expectation is utilized to handle the uncertainties in the objective function, and a linear approximation is employed to derive the deterministic objective function [1]. To address the uncertainties in the constraints, a probability measure is employed. The conventional approach to solving Problem (6.2) involves two nested loops: the outer loop searches for an optimal design, while the inner loop evaluates the feasible probability of the optimal candidate.

The inner loop is often computationally demanding due to the time-consuming estimation of feasible probabilities. To address this challenge, numerically efficient methods for RBDO problems have been developed. In a first set of methods, the inner loop involves solving a deterministic optimization problem. The fundamental idea behind this class of methods is to identify a point on the constraint boundary that is closest to the solution, known as the "most probable point" (MPP) of failure. Then, the task consists in finding this point efficiently. Typically, first or second-order relia-

[1] For a differentiable function C_0 perturbed only by uncertainties in its design variables. These uncertainties can be written as $\mathbf{x} + \boldsymbol{\xi}_\mathbf{x}$ where $\mathbf{x} = \mathbb{E}_{\boldsymbol{\xi}_\mathbf{x}}[\mathbf{x} + \boldsymbol{\xi}_\mathbf{x}]$. Then a first-order Taylor approximation of the function gives that $\mathbb{E}_{\boldsymbol{\xi}_\mathbf{x}}[C_0(\mathbf{x} + \boldsymbol{\xi}_\mathbf{x})] \approx \mathbb{E}_{\boldsymbol{\xi}_\mathbf{x}}[C_0(\mathbf{x}) + \nabla C_0(\mathbf{x})^T \boldsymbol{\xi}_\mathbf{x}] = C_0(\mathbf{x})$. A similar observation holds for the parameters.

bility methods (FORM/SORM) [51] are utilized. These methods transform the uncertainty vectors into uncorrelated Gaussian random vectors using the Rosenblatt or Nataf transformation [108], then the constraints are approximated linearly or quadratically. Therefore, the probabilistic constraints in Problem (6.2) are reformulated as a deterministic optimization problem, reducing the task of solving Problem (6.2) to two nested deterministic optimizations. Various approaches have been employed to solve it with a double loop, such as the Performance Measure Approach (PMA) or the Reliability Index Approach (RIA) [5], a single loop, such as the Single Loop Approach (SLA) [114], or decoupled approaches like the Sequential Optimization and Reliability Assessment (SORA) approach [61] or the Sequential Approximate Programming (SAP) approach [49]. These methods prove to be efficient even when dealing with nonlinear problems, and when gradients are approximated using finite differences [5]. Additionally, methods known as reliability-based robust design optimization (RBRDO) have been developed to handle uncertainties in the objective function by employing a bi-objective formulation of the problem [170].

However, a major drawback of FORM-based methods is their reliance on linear approximations of the objective and constraint functions. These approximations can be inaccurate in practice if the underlying problem is not smooth. Therefore, other methods have been developed that do not rely on linear approximations. Similar to FORM-based methods, these approaches generally use a double-loop strategy. In the inner loop, a reliability analysis estimates the feasible probability. Examples of such methods include important sampling [44, 195], line sampling [4], subset simulation algorithms [9], or surrogate modeling strategies [111, 147]. Subsequently, the estimation of the feasible probability is incorporated into the RBDO problem, resulting in a deterministic problem if the objective function is unnoised or a linear approximation of the objective function can be made.

In addition to the linear approximation, FORM-based methods suffer from another major drawback: they depend on the precise characterization of the uncertainty model of the variables and parameters (required for applying the Nataf transformation). However, the Nataf transformation cannot always be applied, especially when the blackbox inherently contains noise. Even when applicable, the Nataf transformation assumes a specific dependence structure of the uncertainties [109]. Nevertheless, in the absence of sufficient data, justifying and enforcing a specific dependency assumption becomes challenging and unwarranted due to its biasing effect on the final solution. The papers of R. Lebrun and A. Dutfoy [108, 109] provide a detailed discussion of these issues related to using Nataf's transformation in FORM-based methods.

Uncertainties are commonly classified into two categories: aleatory uncertainties and epistemic uncertainties [157]. Aleatory uncertainties represent the stochastic behavior and randomness of events and variables. Epistemic uncertainty is generally associated with a lack of knowledge about phenomena, imprecision in measurements, and poorly designed models. Aleatory uncertainties

can be modeled by random variables, while epistemic uncertainties can be represented by interval, point data or modeled using Gaussian processes [197]. Using probabilistic models for epistemic uncertainties may lead to infeasible designs in practice [157]. Even for aleatory uncertainties, selecting an appropriate probabilistic model can be challenging, especially when the dimension of the uncertainties is large or when dependencies are unknown due to data scarcity [157]. A poorly chosen model can result in underperforming designs or designs with significant failures [163]. When epistemic uncertainties are involved in reliability analysis, non-probabilistic approaches based on evidence theory [172], possibility theory [62], or fuzzy sets [119, 196] may be used.

Recently, some approaches have utilized ellipsoidal sets to model uncertainties [129, 188]. When both types of uncertainties are present, combining probabilistic and non-probabilistic models to address these uncertainties may be an interesting option [66, 128]. Alternatively, distributionally robust chance-constrained programming [190] or a Bayesian probabilistic approach using Gaussian processes [3, 139] also appear promising. Finally, scenario optimization, that tackles Problem (6.1) using available data without prescribing a specific model (or a set of models) for the uncertainty, has been explored [158]. Unfortunately, the described approaches are primarily used for reliability analysis, and they do not handle uncertainties in the objective function, except in the work in [3], which is limited to parameter uncertainties. Another significant drawback is the lack of a convergence proof to an optimal point of the problem. Table 6.1 summarizes the different methods based on several criteria. The first two criteria assess whether the methods may deal with nonsmooth problems, while the third evaluates the ability of the method to handle noise in the objective function as well as in the constraints. The fourth criterion examines whether the method requires a precise characterization of the distribution that models the aleatory uncertainties, (e.g. for applying the Nataf transformation). Finally, the last criterion assesses the capability of the method to handle uncertainties in the absence of perfect knowledge of the data.

6.3.2 Contributions

To account for the uncertainties in both the objective and constraint functions, methods utilizing the conditional value-at-risk (CVaR) have been developed [112, 162]. CVaR is a coherent risk measure that evaluates the risk associated with a design solution by combining the probability of undesired events with a measure of the magnitude or severity of those events. CVaR methods have found extensive applications in risk averse optimization like in trust-region algorithms [130], in engineering design problems [81, 113, 161, 193], and in constrained reinforcement learning [50, 181].

One of the main interest of the CVaR measure lies in the flexibility provided by the parameter α. When $\alpha = 0$, the CVaR measure corresponds to the expectation, whereas as α approaches 1, it

Table 6.1 Summary of the different methods and their limits

Methods	Type [1]	Handles nonsmooth constraints	Handles nonsmooth objective	Handles noisy objective	Allows unknown aleatory uncertainty	Allows lack of data [2]
FORM-based [5, 49, 61, 114]	O	✗	✗	✗	✗	✗
RBRDO [170]	O	✗	✓	✓	✗	✗
Importance Sampling [44, 195]	O	✓	✓	✗	✗	✗
Line Sampling [4]	RA	✓	N/A	N/A	✗	✓
Subset simulation [9]	RA	✓	N/A	N/A	✓	✗
Surrogate modelling [111, 147]	RA	✓	N/A	N/A	✓	✗
Mixed approaches [66, 128]	O	✗	✗	✗	✗	✓
Ellipsoidal set [129, 188]	O	✗	✗	✗	✗	✓
Bayesian approach (I) [139]	RA	✓	N/A	N/A	✓	✓
Bayesian approach (II) [3]	O	✓	✓	✓[3]	✓	✗
Scenario Optimization [158]	O	✗	✗	✗	✓	Only point data
This work	O	✓	✓	✓	✓	✓[4]

[1] The type indicates if the method handle the whole stochastic constrained optimization problem (O) or is limited to reliability analysis (RA).
[2] Only points or interval data are available.
[3] Only parameters uncertainties.
[4] For interval data, the method allows only to obtain worst-case solution.

corresponds to the supremum of the function over the support of the uncertainties [160]. This versatility allows to handle both aleatory and epistemic uncertainties, albeit in a worst-case scenario only. However, substituting failure probability constraints with CVaR constraints is a conservative approach [173, chapter 6] that might render the problem infeasible in the worst case. Moreover, the closer the value of α is to 1, the more sensitive the measure becomes to the uncertainty model, particularly in the tails. Managing this heightened sensitivity necessitates an untractable number of samples. While the former issue is challenging to avoid a priori, the latter can be partially addressed by employing a multi-timescale stochastic approximation algorithm to estimate the CVaR value [50, 152]. Unfortunately, the methods utilized in the referenced papers cannot be directly applied to solve a CVaR formulation of Problem (6.1). In fact, these methods cleverly leverage the properties of the Markov Decision Process to compute estimates of the gradients, a strategy that is

impossible to use in the context of the present study. The contributions of this work are outlined as follows.

First, in Section 6.5, the process of smoothing the problem and obtaining analytical gradient estimates from noisy measurements of the blackbox is described. A smooth approximation of the gradient [33, 141] is employed. The concept involves approximating the original function by its convolution with a multivariate density function. The resulting approximation possesses several desirable properties: it is infinitely differentiable even if the original function is only piecewise continuous, it preserves the structural properties (such as convexity and Lipschitz constant) of the original function, and an unbiased estimator of the gradient of the smooth approximation can be calculated from only two measurements of the blackbox. In most studies [71, 141], Gaussian or uniform density functions are utilized for the approximation. However, in this paper, a truncated Gaussian density function is developed to satisfy the bound constraints of the problem (6.1). The properties of this new approximation and its associated unbiased gradient estimator are provided.

Second, Problem (6.1) is reformulated as a CVaR-constrained problem, wherein the objective function and the constraints are approximated by their smooth truncated Gaussian counterparts. The quality of this approximation is theoretically examined and depends on several parameters such that the value of α, the dimension and the value of the smoothing parameter. Subsequently, following the approach in [50], a Lagrangian relaxation is applied to the problem. The method used to solve the relaxed problem is developed in Section 6.6. It involves a four-timescale stochastic approximation algorithm. The first timescale aggregates information about the gradient, the second estimates the quantile of the objective and constraint functions, the third updates the design variables in a descent direction, and the last one updates the Lagrange multiplier in the ascent direction. The convergence analysis of this algorithm is studied in Section 6.7 and is conducted using an Ordinary Differential Equation (ODE) approach. Under mild assumptions, this algorithm almost surely converges to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution.

Finally, in Section 6.8, practical implementation details are provided to minimize the number of hyperparameters in the developed algorithm. Numerical experiments are conducted to estimate the values of the remaining hyperparameters. Then, comparisons are made between the algorithm using the Gaussian gradient estimator and its truncated counterpart. In the last subsection, the efficiency of the algorithm is demonstrated on problems involving mixed aleatory/epistemic uncertainties. Conclusions are drawn in Section 6.9.

6.4 Problem formulation

In order to formally settle the problem and to develop the convergence analysis, the following assumptions are made on the functions C_j and used throughout the paper.

Assumption 4. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and consider $C_j(\mathbf{x}, \boldsymbol{\xi}) : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}, j \in [0, m]$ where $\boldsymbol{\xi} : \Omega \to \Xi \subset \mathbb{R}^d$ is the vector modelling the uncertainties. Then, the following hold for all $j \in [0, m]$.*

1. *There exists a measurable function $\kappa_1(\boldsymbol{\xi}) : \Xi \to \mathbb{R}$ such that $\mathbb{E}_{\boldsymbol{\xi}}[\kappa_1(\boldsymbol{\xi})] \leq L_1 < \infty$ and for which*
$$|C_j(\mathbf{x}, \boldsymbol{\xi})| \leq \kappa_1(\boldsymbol{\xi}),\ \forall \mathbf{x} \in \mathcal{X} \text{ and } \boldsymbol{\xi} \in \Xi.$$

2. *There exists a measurable function $\kappa_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) : \Xi \times \Xi \to \mathbb{R}$ where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are i.i.d. random vectors such that $\mathbb{E}_{\boldsymbol{\xi}}[\kappa_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)] \leq L_2 < \infty$ and for which*
$$|C_j(\mathbf{x}, \boldsymbol{\xi}_1) - C_j(\mathbf{y}, \boldsymbol{\xi}_2)| \leq \kappa_2(\boldsymbol{\xi})||\mathbf{x} - \mathbf{y}||,\ \forall\ (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X} \text{ and } (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \in \Xi \times \Xi.$$

3. *The function $C_j(\cdot, \boldsymbol{\xi})$ has a continuous cumulative distribution function and there exists a measurable function $\kappa_3(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) : \Xi \times \Xi \to \mathbb{R}$, where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are i.i.d. random vectors such that $\mathbb{P}_{\boldsymbol{\xi}}(\kappa_3(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \leq L_3) = 1$ with $L_3 < \infty$ and for which*
$$|C_j(\mathbf{x}, \boldsymbol{\xi}_1) - C_j(\mathbf{x}, \boldsymbol{\xi}_2)| \leq \kappa_3\ (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)||\mathbf{x} - \mathbf{y}||,\ \forall\ (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X} \text{ and } (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \in \Xi \times \Xi.$$

Three comments on these assumptions. First, note that no assumptions are made about the differentiability of the functions C_j. Second, Assumption 4.1 will be made throughout this paper because it allows the value-at-risk (VaR) and the CVaR of the functions C_j to be well defined. The other assumptions are used in Section 6.5 to bound the approximation of the constrained CVaR blackbox problem and in Section 6.7 to study the convergence of the proposed method. Finally, the assumptions are increasingly strong, i.e., Assumption 4.3 implies Assumption 4.2, which implies Assumption 4.1.

Now, the VaR at level $\alpha \in (0, 1)$ of the objective and constraint functions may be defined. It is originally derived from the left-side quantile of level α of a given random variable. Given $j \in \{0, 1, \ldots, m\}$ and a reliability level $\alpha_j \in (0, 1)$, the VaR of a function $C_j(\mathbf{x}, \boldsymbol{\xi})$ is defined as

$$\mathrm{VaR}_{\alpha_j}(\mathbf{x}) := \inf\{t \,|\, \mathbb{P}(C_j(\mathbf{x}, \boldsymbol{\xi}) \leq t) \geq \alpha_j\}.$$

The VaR of a function has several interesting properties. When the cumulative distribution function

$\mathbb{P}(C_j(\mathbf{x}, \boldsymbol{\xi}) \leq u)$ is right continuous with respect to t, the infemum is a minimum and if it is, in addition, continuous and strictly increasing, then VaR_{α_j} is the unique t such that $\mathbb{P}(C_j(\mathbf{x}, \boldsymbol{\xi}) \leq t) = \alpha$. However, the VaR of a function is computationally intractable, is not a coherent risk measure [8] and does not take into account the magnitude/severity of the undesired events. Therefore, in practice another measure is used: the Conditional Value-at-Risk. The CVaR of a function $C_j(\cdot, \boldsymbol{\xi})$, for a level $\alpha_j \in (0, 1)$ at a point, $\mathbf{x}$ may be defined as [162]

$$\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) := \min_{t \in \mathbb{R}} V_{\alpha_j}(\mathbf{x}, t), \tag{6.3}$$

where

$$V_{\alpha_j}(\mathbf{x}, t) = t + \frac{1}{1 - \alpha_j} \mathbb{E}_{\boldsymbol{\xi}}[(C_j(\mathbf{x}, \boldsymbol{\xi}) - t)^+], \tag{6.4}$$

where the superscript plus denotes the function $(t)^+ := \max\{0, t\}$. The level α_j gives the possibility to choose the desired degree of reliability. Choosing a level close to 0 is tantamount to taking the expectation measure into account, i.e. adopting a "risk neutral" approach. On the other hand, choosing a level close to 1 is tantamount to taking a "worst-case" approach. In this way, different values of α_j can be used for the different objective and constraint functions, depending on the degree of reliability desired for each of them. Now, problem (6.1) can be reformulated as a CVaR-constrained blackbox optimization problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}} \quad & \mathrm{CVaR}_{\alpha_0}(\mathbf{x}) \\ \text{s.t.} \quad & \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \leq 0, \ \forall j \in [1, m]. \end{aligned} \tag{6.5}$$

This formulation is a convex program if the objective and constraint functions are convex in the design space. This convexification of the design space makes Problem (6.5) a conservative approximation of Problem (6.2) [Chapter 6, [173]]. Thus, this formulation guarantees a conservative result in terms of failure probability, see e.g. [163]. To solve Problem (6.5), it is usually reformulated with the function V_α as follows

$$\begin{aligned} \min_{(\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \mathbb{R}^{m+1}} \quad & V_{\alpha_0}(\mathbf{x}, t_0) \\ \text{s.t.} \quad & V_{\alpha_j}(\mathbf{x}, t_j) \leq 0, \ \forall j \in [1, m]. \end{aligned} \tag{6.6}$$

The equivalence between Problem (6.5) and Problem (6.6) is shown in the following lemma.

Lemma 6.4.1. *Suppose the solution sets of Problem (6.5) and Problem (6.6) are not empty. Then these problems are equivalent in the sense that, $\mathbf{x}^*$ is a solution of Problem (6.5) if and only if there exist $\mathbf{t}^* \in \mathbb{R}^{m+1}$ such that $(\mathbf{x}^*, \mathbf{t}^*)$ is a solution of Problem (6.6), and the optimal values are the same.*

Proof. By the definition of the Conditional Value-at-Risk given in Equation (6.3), Problem (6.5) may be reformulated as follows

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \left(\min_{t_0\in\mathbb{R}} \quad V_{\alpha_0}\left(\mathbf{x}, t_0\right)\right) \\ &\text{s.t.} \left(\min_{t_j\in\mathbb{R}} \quad V_{\alpha_j}(\mathbf{x}, t_j)\right) \leq 0,\ \forall j \in [1, m]. \end{aligned} \tag{6.7}$$

Now, the following relations hold

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \left(\min_{t_0\in\mathbb{R}} V_{\alpha_0}\left(\mathbf{x}, t_0\right)\right) = \min_{(\mathbf{x},t_0)\in\mathcal{X}\times\mathbb{R}} V_{\alpha_0}\left(\mathbf{x}, t_0\right) \\ &\left(\min_{t_j\in\mathbb{R}} V_{\alpha_j}(\mathbf{x}, t_j)\right) \leq 0,\ \forall j \in [1, m] \iff \forall j,\ \exists t_j \text{ s.t. } V_{\alpha_j}(\mathbf{x}, t_j) \leq 0. \end{aligned}$$

Therefore, the Problems (6.7) and (6.6) are equivalent. Now, let $\mathbf{x}^*$ be a solution of Problem (6.5), it is possible to construct the associated vector $\mathbf{t}^*(\mathbf{x}^*)$ where $t_j^*(\mathbf{x}^*) = \mathrm{VaR}_{\alpha_j}(\mathbf{x}^*), \forall j \in [0, m]$. The tuple $(\mathbf{x}^*, \mathbf{t}^*(\mathbf{x}^*))$ is then solution of Problem (6.7) and as a consequence of Problem (6.6) which ends the proof. □

Despite this property, Problem (6.5) is difficult to solve for two main reasons. First, since the functions C_j are the outputs of a blackbox, the gradients of these functions may not exist, and even if they do, their analytic formulations are not available. Second, the problem is highly sensitive to the values of α_j, and the closer the values are to 1, the harder the problem is to solve. The next section describes the strategy used in this paper to overcome these difficulties.

6.5 Smooth approximation and Lagrangian relaxation of the problem

This section introduces a method for solving the Problem (6.5). To obtain a more tractable problem, the original problem is approximated by a smooth problem using truncated Gaussian smoothing. The quality of the approximation is then studied and a Lagrangian relaxation of the smooth problem is given.

6.5.1 Truncated Gaussian smooth approximation

In a blackbox optimization framework, all we know is that for any given input, the blackbox will return an output, which may be subject to uncertainties. To obtain a more tractable problem, a smooth approximation may be used [166, pp. 263]. The principle of this method is to approximate the function by its convolution with a kernel density function. Formally, if c is an integrable function, $\beta > 0$ is a scalar, and $\mathbf{u}$ is a random vector with distribution ϕ, the smooth approximation of c can be defined as

$$c^\beta(\mathbf{x}) \ := \ \int_{-\infty}^{+\infty} c(\mathbf{x} - \beta\mathbf{u})\phi(\mathbf{u})d\mathbf{u} \ = \ \mathbb{E}_{\mathbf{u}}[c(\mathbf{x} + \beta\mathbf{u})]. \tag{6.8}$$

The smooth approximation benefits from several attractive properties. First, it can be interpreted as a local weighted average of the function values in the neighborhood of $\mathbf{x}$. If c is continuous at $\mathbf{x}$, it is possible to obtain a value of $c^\beta(\mathbf{x})$ that is arbitrarily close to the value of $c(\mathbf{x})$ by using an appropriate value of β. Second, it inherits the degree of smoothness of the density function as a consequence of the convolution product. Finally, depending on the chosen kernel, stochastic gradient estimators can be computed. They are unbiased estimators of the gradient of c^β and can be constructed only from values of $c(\mathbf{x})$ and $c(\mathbf{x} + \beta\mathbf{u})$.

The most commonly used kernels are the Gaussian distribution and the uniform distribution on a sphere [71, 141]. However, if the problem has bound constraints, a significant drawback of these distributions is that the random vector $\mathbf{x}+\sigma\mathbf{u}$ may fall outside the bound constraints. For instance, if $\mathbf{u} \sim \mathcal{N}(0, 1)$, $\mathbf{x}+\sigma\mathbf{u}$ might be sampled outside the bounds. This issue persists even with a uniform distribution if $\mathbf{x}$ is near the bounds. However, the bound constraints are usually non-relaxable in the sense of [107], meaning that the output of the blackbox lacks significance for optimization outside the bound constraints. This can occur due to physical phenomena or when the blackbox is undefined beyond the bounds. In such cases, the gradient estimate of c^β, computed from the values of the function c at the points $\mathbf{x}$ and $\mathbf{x} + \beta\mathbf{x}$, becomes unreliable. To address this issue, a truncated Gaussian estimator is developed in this paper, and its main properties are summarized in the following lemma.

Lemma 6.5.1. *Let c be an integrable function on $\mathcal{X}$, the smooth approximation c^β is defined as*

$$c^\beta(\mathbf{x}) \ = \ \mathbb{E}_{\mathbf{u}}[c(\mathbf{x} + \beta\mathbf{u})],$$

where $\mathbf{u} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell - \mathbf{x}}{\beta}, \frac{\mathbf{b}_u - \mathbf{x}}{\beta})$, $\mathbf{b}_\ell$ and $\mathbf{b}_u$ are respectively the lower and the upper bounds of the problem. In what follows, ϕ and Φ denote respectively the probability density function (p.d.f.) and the cumulative density function (c.d.f.) of the standard Gaussian distribution.

Now, the following holds.

1. *c^β is infinitely differentiable: $c^\beta \in \mathcal{C}^\infty$.*

2. *A one-sided unbiased estimator of ∇c^β is*

$$\tilde{\nabla} c^\beta(\mathbf{x}) \;=\; \frac{(\mathbf{u}-\boldsymbol{\mu})c(\mathbf{x}+\beta\mathbf{u}) - \;(\mathbf{u}-\boldsymbol{\mu})c(\mathbf{x})}{\beta}, \tag{6.9}$$

   where $\boldsymbol{\mu}$ is the mean of the truncated Gaussian vector, i.e,

$$\mu_i \;=\; \frac{\phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right) - \phi\left(\frac{b_{u_i}-x_i}{\beta}\right)}{\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)}, \quad \forall i \in [1,n].$$

3. *Let $\mathbf{u}_1 \sim \mathcal{TN}(\mathbf{0},\mathbf{I},\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta},\frac{\mathbf{b}_u-\mathbf{x}}{\beta})$ and $\mathbf{u}_2 \sim \mathcal{TN}(\mathbf{0},\mathbf{I},\frac{\mathbf{x}-\mathbf{b}_u}{\beta},\frac{\mathbf{x}-\mathbf{b}_\ell}{\beta})$, a two-sided unbiased estimator of ∇c^β is*

$$\tilde{\nabla} c^\beta(\mathbf{x}) \;=\; \frac{(\mathbf{u}_1-\boldsymbol{\mu}_1)(c(\mathbf{x}+\beta\mathbf{u}_1)-c(\mathbf{x})) - \;(\mathbf{u}_2-\boldsymbol{\mu}_2)(c(\mathbf{x}-\beta\mathbf{u}_2)-c(\mathbf{x}))}{2\beta}, \tag{6.10}$$

4. *In addition, if c is a L-Lipschitz continuous function, let $\beta \geq 0$, then $\forall \mathbf{x} \in \mathbb{R}^n$*

$$|c^\beta(\mathbf{x}) - c(\mathbf{x})| \;\leq\; L\beta\sqrt{n}.$$

Proof. 1.) This can be shown by noting that the truncated Gaussian kernel is infinitely differentiable within the bounds. However, to obtain the above estimators, the calculation must be done. Therefore, using the above notation, and given that the components u_i of $\mathbf{u}$ are mutually independent, it follows that

$$\begin{aligned}
\mathbb{E}_{\mathbf{u}}[c(\mathbf{x}+\beta\mathbf{u})] &= \int_{\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}}^{\frac{\mathbf{b}_u-\mathbf{x}}{\beta}} c(\mathbf{x}+\beta\mathbf{u}) \prod_{i=1}^{n} \frac{\phi(\mathbf{u}_i)}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right) - \Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)} d\mathbf{u} \\
&= \int_{\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}}^{\frac{\mathbf{b}_u-\mathbf{x}}{\beta}} \frac{1}{(2\pi)^{\frac{n}{2}}} c(\mathbf{x}+\beta\mathbf{u}) \prod_{i=1}^{n} \frac{e^{-\frac{u_i^2}{2}}}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right) - \Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)} d\mathbf{u} \\
&= \frac{1}{(2\pi)^{\frac{n}{2}}} \left(\prod_{i=1}^{n} \frac{1}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right) - \Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)} \right) \int_{-\infty}^{\infty} \mathbf{1}_{\left[\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}, \frac{\mathbf{b}_u-\mathbf{x}}{\beta}\right]}(\mathbf{u}) c(\mathbf{x}+\beta\mathbf{u}) \prod_{i=1}^{n} e^{-\frac{u_i^2}{2}} d\mathbf{u},
\end{aligned}$$

where $\mathbf{1}_{[.]}(\cdot)$ denotes the indicator function. Substituting $\mathbf{v} = \mathbf{x} + \beta\mathbf{u}$ leads to:

$$\mathbb{E}_{\mathbf{u}}[c(\mathbf{x}+\beta\mathbf{u})] = \frac{1}{(2\pi)^{\frac{n}{2}}\beta^n}\left(\prod_{i=1}^{n}\frac{1}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right)-\Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)}\right)\int_{-\infty}^{\infty}\mathbf{1}_{[\mathbf{b}_\ell,\mathbf{b}_u]}(\mathbf{v})c(\mathbf{v})\prod_{i=1}^{n}e^{-\frac{(x_i-v_i)^2}{2\beta^2}}d\mathbf{v}.$$

By setting

$$h_1(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}\beta^n}\prod_{i=1}^{n}\frac{1}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right)-\Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)},$$
$$h_2(\mathbf{x}) = \mathbf{1}_{[\mathbf{b}_\ell,\mathbf{b}_u]}(\mathbf{x})c(\mathbf{x}) \qquad \text{and} \qquad h_3(\mathbf{x}) = \prod_{i=1}^{n}e^{-\frac{(x_i)^2}{2\beta^2}},$$

$c^\beta(\mathbf{x})$ may be compactly written as

$$c^\beta(\mathbf{x}) = h_1(\mathbf{x})(h_2 * h_3)(\mathbf{x}),$$

where $*$ is the convolution product between two functions. As $h_3 \in \mathcal{C}^\infty(\mathbb{R}^n)$ and $h_2 \in \mathcal{L}^1(\Omega,\mathcal{F},\mathbb{P})$ then $(h_2 * h_3) \in \mathcal{C}^\infty(\mathbb{R}^n)$ (property of convolution product). Moreover, $h_1 \in \mathcal{C}^\infty(\mathbb{R}^n)$ as well, therefore $c^\beta(\mathbf{x}) \in \mathcal{C}^\infty(\mathbb{R}^n)$ as it is the product of infinitely continuously differentiable functions.

2.) By using the same notation as above, the partial derivative of c^β may be computed, for $j \in [1,n]$ as

$$\frac{\partial c^\beta(\mathbf{x})}{\partial x_j} = \frac{\partial h_1(\mathbf{x})}{\partial x_j}(h_2 * h_3)(\mathbf{x}) + h_1(\mathbf{x})\left(h_2 * \frac{\partial h_3}{\partial x_j}\right)(\mathbf{x}).$$

Yet, we have

$$\frac{\partial h_1(\mathbf{x})}{\partial x_j} = \frac{1}{(2\pi)^{\frac{n}{2}}\beta^n}\left(\prod_{i=1}^{n}\frac{1}{\Phi\left(\frac{\mathbf{b}_{u_i}-\mathbf{x}_i}{\beta}\right)-\Phi\left(\frac{\mathbf{b}_{\ell_i}-\mathbf{x}_i}{\beta}\right)}\right)\frac{\phi\left(\frac{b_{u_j}-x_j}{\beta}\right)-\phi\left(\frac{\mathbf{b}_{\ell_j}-x_j}{\beta}\right)}{\beta\left(\Phi\left(\frac{\mathbf{b}_{u_j}-\mathbf{x}_j}{\beta}\right)-\Phi\left(\frac{\mathbf{b}_{\ell_j}-\mathbf{x}_j}{\beta}\right)\right)} = -\frac{\mu_j h_1(\mathbf{x})}{\beta}$$
$$\frac{\partial h_3(x)}{\partial x_j} = -\frac{x_j}{\beta^2}h_3(\mathbf{x}).$$

Thus, we obtain

$$\frac{\partial c^\beta(\mathbf{x})}{\partial x_j} = \mathbb{E}_{\mathbf{u}}\left[\frac{u_j-\mu_j}{\beta}c(\mathbf{x}+\beta\mathbf{u})\right].$$

From this result, an unbiased estimator of the gradient of c^β is

$$\tilde{\nabla}c^\beta(\mathbf{x}) = \frac{\mathbf{u}-\boldsymbol{\mu}}{\beta}c(\mathbf{x}+\beta\mathbf{u}).$$

As the variance of this estimator gets unbounded as β goes to 0, in practice the following estimator is used

$$\tilde{\nabla} c^{\beta}(\mathbf{x}) = \frac{(\mathbf{u}-\boldsymbol{\mu})c(\mathbf{x}+\beta\mathbf{u}) - (\mathbf{u}-\boldsymbol{\mu})c(\mathbf{x})}{\beta}.$$

This estimator is still unbiased since $\mathbb{E}_{\mathbf{u}}[(\mathbf{u}-\boldsymbol{\mu})c(\mathbf{x})] = 0$.

3.) Symmetrically, if $\mathbf{u} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{x}-\mathbf{b}_u}{\beta}, \frac{\mathbf{x}-\mathbf{b}_\ell}{\beta})$, an unbiased estimator is

$$\tilde{\nabla} c^{\beta}(\mathbf{x}) = \frac{(\boldsymbol{\mu}-\mathbf{u})c(\mathbf{x}-\beta\mathbf{u}) - (\boldsymbol{\mu}-\mathbf{u})c(\mathbf{x})}{\beta}.$$

thus, by summation of the two one-sided estimator, the two-sided estimator is obtained.

4.) Finally, we have, with $\mathbf{u}_1 \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell-\mathbf{x}}{\beta^1}, \frac{\mathbf{b}_u-\mathbf{x}}{\beta^1})$ and $\mathbf{u}_2 \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell-\mathbf{x}}{\beta^2}, \frac{\mathbf{b}_u-\mathbf{x}}{\beta^2})$

$$|c^{\beta}(\mathbf{x}) - c(\mathbf{x})| = |\mathbb{E}_{\mathbf{u}}[c(\mathbf{x}+\beta\mathbf{u})] - c(\mathbf{x})| \leq \mathbb{E}_{\mathbf{u}}[|c(\mathbf{x}+\beta\mathbf{u}) - c(\mathbf{x})|] \leq L\beta\mathbb{E}_{\mathbf{u}}[||\mathbf{u}||].$$

where the first inequality comes from the Jensen's inequality and the second one comes from the L-Lipschitz continuity of c. It remains to bound $\mathbb{E}_{\mathbf{u}}[||\mathbf{u}||]$ when $\mathbf{u}$ is a truncated Gaussian vector, for this purpose, the proof of Lemma 1 of [141] is adapted for truncated Gaussian distribution. The following identity is used:

$$\int_{\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}}^{\frac{\mathbf{b}_u-\mathbf{x}}{\beta}} e^{-\frac{||u||^2}{2}} d\mathbf{u} \;=\; (2\pi)^{n/2} \prod_{i=1}^{n} \left(\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)\right) := \kappa.$$

By setting $\mathbf{v} = \mathbf{x} + \beta\mathbf{u}$ and multiplying by β^n, the last equalities become

$$\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} e^{-\frac{||\mathbf{v}-\mathbf{x}||^2}{2\beta^2}} d\mathbf{v} \;=\; (2\pi)^{n/2} \prod_{i=1}^{n} \left(\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)\right) \beta^n \;=\; \kappa\beta^n.$$

Taking the logarithm yields

$$\ln\left(\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} e^{-\frac{||\mathbf{v}-\mathbf{x}||^2}{2\beta^2}} d\mathbf{v}\right) \;=\; n\ln(\beta) + \frac{n}{2}\ln(2\pi) + \sum_{i=1}^{n} \ln\left(\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)\right). \quad (6.11)$$

Now, the derivative of the left-hand-side of Equation (6.11) with respect to β is given by

$$\begin{aligned}\frac{\partial}{\partial\beta}\ln\left(\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} e^{-\frac{||\mathbf{v}-\mathbf{x}||^2}{2\beta^2}}d\mathbf{v}\right) &= \frac{1}{\kappa\beta^n}\int_{\mathbf{b}_\ell}^{\mathbf{b}_u}\frac{||\mathbf{v}-\mathbf{x}||^2}{\beta^3}e^{-\frac{||\mathbf{U}-\mathbf{x}||^2}{2\beta^2}}d\mathbf{v}\\ &= \frac{1}{\kappa\beta}\int_{\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}}^{\frac{\mathbf{b}_u-\mathbf{x}}{\beta}}||\mathbf{u}||^2 e^{-\frac{||\mathbf{u}||^2}{2}}d\mathbf{u} \qquad \text{since } \frac{\mathbf{v}-\mathbf{x}}{\beta}=\mathbf{u}\\ &= \frac{1}{\beta}\mathbb{E}_{\mathbf{u}}[||\mathbf{u}||^2]\end{aligned}$$

and the derivative of the right-hand-side of Equation (6.11) is given by

$$\frac{n}{\beta}+\sum_{i=1}^{n}\frac{\frac{b_{\ell_i}-x_i}{\beta^2}\phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)-\frac{b_{u_i}-x_i}{\beta^2}\phi\left(\frac{b_{u_i}-x_i}{\beta}\right)}{\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right)-\Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)}.$$

Thus,

$$\mathbb{E}_{\mathbf{u}}[||\mathbf{u}||^2] \;=\; n+\sum_{i=1}^{n}\frac{\frac{b_{\ell_i}-x_i}{\beta}\phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)-\frac{b_{u_i}-x_i}{\beta}\phi\left(\frac{b_{u_i}-x_i}{\beta}\right)}{\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right)-\Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)} \;\leq\; n, \tag{6.12}$$

where the inequality holds because the sum is negative for $\mathbf{x}\in\mathcal{X}$. Finally, with the result in Equation (6.12) and the results of Lemma 1 of [141], the following bound appears

$$\mathbb{E}_{\mathbf{u}}[||\mathbf{u}||]\leq\sqrt{n}.$$

□

When only noisy outputs of the blackbox are available, the following estimator is used

$$\tilde{\nabla}C^{\beta}(\mathbf{x},\boldsymbol{\xi}) \;=\; \frac{(\mathbf{u}-\boldsymbol{\mu})\left(C(\mathbf{x}+\beta\mathbf{u},\boldsymbol{\xi}_1)-C(\mathbf{x},\boldsymbol{\xi}_2)\right)}{\beta}, \tag{6.13}$$

where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are two independent identically distributed realizations of a random vector $\boldsymbol{\xi}$. This estimator is still unbiased because

$$\mathbb{E}_{\mathbf{u},\boldsymbol{\xi}}[\tilde{\nabla}c^{\beta}(\mathbf{x},\boldsymbol{\xi})] \;=\; \mathbb{E}_{\mathbf{u}}[\mathbb{E}_{\boldsymbol{\xi}}[\tilde{\nabla}c^{\beta}(\mathbf{x},\boldsymbol{\xi})|\mathbf{u}]]=\nabla c^{\beta}(\mathbf{x}).$$

6.5.2 Smooth approximation of CVaR-constrained blackbox optimization problem

The non-smoothness of a CVaR-constrained blackbox optimization problem arises from two elements: the potential non-smoothness of the functions C_j and the non-smoothness introduced by the function $\max$ in the CVaR formulation. The concept of smoothing a CVaR-constrained opti-

mization problem is not novel; it has been explored in prior works [127, 174]. In this study, this concept is applied to both sources of non-smoothness using the aforementioned truncated Gaussian smoothing. As $\mathbf{t}$ is an unconstrained vector, arbitrarily large bounds are introduced for this vector. Let $\beta_1, \beta_2 > 0$ be two scalars, $\mathbf{u} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell - \mathbf{x}}{\beta_1}, \frac{\mathbf{b}_u - \mathbf{x}}{\beta_1})$ a random vector of size n and $\mathbf{v} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{-\mathbf{t}_{\max}}{\beta_2}, \frac{\mathbf{t}_{\max}}{\beta_2})$, a random vector of size $m+1$, where $\mathbf{t}_{\max}$ is chosen to be sufficiently large, the smooth approximation of V_{α_j} and CVaR_{α_j} for all $j \in [0, m]$ are defined respectively as

$$
\begin{aligned}
V^{\beta}_{\alpha_j}(\mathbf{x}, t_j) &= \mathbb{E}_{\mathbf{u},\mathbf{v}}[V_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u}, t_j + \beta_2 v_j)], \quad \text{and} \\
\mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) &= \min_{t_j \in \mathbb{R}} \mathbb{E}_{\mathbf{u},\mathbf{v}}[V_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u}, t_j + \beta_2 v_j)].
\end{aligned}
$$

Then, the smooth approximation of the Problem (6.6) may be formulated as follows

$$
\begin{array}{ll}
\min\limits_{(\mathbf{x},\mathbf{t}) \in \mathcal{X} \times \mathbb{R}^{m+1}} & V^{\beta}_{\alpha_0}(\mathbf{x}, t_0) \\
\text{s.t.} & V^{\beta}_{\alpha_j}(\mathbf{x}, t_j) \leq 0, \ \forall j \in [1, m].
\end{array}
\tag{6.14}
$$

Now, the quality of this smooth approximation is studied. The following Lemma states properties of the truncated Gaussian smoothing approximation applied with the CVaR measure.

Theorem 6.5.2. *Under Assumption 4.2, the following holds.*

1. $|\mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) - \mathrm{CVaR}_{\alpha_j}(\mathbf{x})| \leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}$ *for all* $j \in [0, m]$ *and* $\mathbf{x} \in \mathcal{X}$*;*
2. $|\mathrm{CVaR}^{\beta}_{\alpha_0}(\tilde{\mathbf{x}}^*) - \mathrm{CVaR}_{\alpha_0}(\mathbf{x}^*)| \leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}$*, where* $\tilde{\mathbf{x}}^*$ *and* $\mathbf{x}^*$ *are solutions of Problem (6.14) and (6.5) respectively.*
3. *If Assumption 4.3 holds, then there exists a threshold* $\bar{\alpha}_j \in (0, 1]$ *such that for all* $\alpha_j \geq \bar{\alpha}_j$

$$
\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \ \leq \ \mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) \ \leq \ \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}.
$$

Thus, for $\alpha_j \geq \bar{\alpha}_j$*, if* $\tilde{x}^*$ *is a solution of Problem (6.14), then it is a feasible point for Problem (6.5).*

Proof. 1. Under Assumption 4.2, it follows that for all $(\mathbf{x}, t_j) \in \mathcal{X} \times \mathbb{R}$

$$\begin{aligned}
|V_{\alpha_j}^{\beta}(\mathbf{x}, t_j) - V_{\alpha_j}(\mathbf{x}, t_j)| &= \frac{1}{1-\alpha_j} \left| \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - \ (t_j + \beta_2 v_j))^+ - \ (C_j(\mathbf{x}, \boldsymbol{\xi}_2) - t_j)^+] \right| \\
&\leq \frac{1}{1-\alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[|(C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - \ (t_j + \beta_2 v_j))^+ - \ (C_j(\mathbf{x}, \boldsymbol{\xi}_2) - t_j)^+|] \\
&\leq \frac{1}{1-\alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[|C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - \beta_2 v_j - C_j(\mathbf{x}, \boldsymbol{\xi}_2)|] \\
&\leq \frac{1}{1-\alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[\kappa_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)\beta_1 ||\mathbf{u}|| + \beta_2 |v_j|] \\
&\leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1-\alpha_j},
\end{aligned}$$

where the first inequality follows from Jensen's inequality, the second from the following inequality $|\max(0, a) - \max(0, b)| \ \leq \ |a - b|$, the third from Assumption 4.2 and the last one from the independence of $\mathbf{u}$ and $\kappa_2(\boldsymbol{\xi})$ and the bound on the expectation of the norm of (truncated) Gaussian random vectors. This is true for all tuples $(\mathbf{x}, t_j) \in \mathcal{X} \times \mathbb{R}$, in particular for $t_j^* \in \operatorname{argmin} V(\mathbf{x}, t_j)$ and $\tilde{t}_j^* \in \operatorname{argmin} \mathbb{E}_{\mathbf{u},\mathbf{v}}[V_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u}, t_j + \beta_2 v_j)]$. Therefore, it follows that for any $j \in [0, m]$ and any $\mathbf{x} \in \mathcal{X}$

$$V_{\alpha_j}^{\beta}(\mathbf{x}, \tilde{t}_j^*) \ \leq \ V_{\alpha_j}^{\beta}(\mathbf{x}, t_j^*) \ \leq \ V_{\alpha_j}(\mathbf{x}, t_j^*) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1-\alpha_j}.$$

Conversely, it also follows that

$$V_{\alpha_j}(\mathbf{x}, t_j^*) \ \leq \ V_{\alpha_j}(\mathbf{x}, \tilde{t}_j^*) \ \leq \ V_{\alpha_j}^{\beta}(\mathbf{x}, \tilde{t}_j^*) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1-\alpha_j}.$$

Recalling that $\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) = V_{\alpha_j}(\mathbf{x}, t_j^*)$ and $\mathrm{CVaR}_{\alpha_j}^{\beta}(\mathbf{x}) = V_{\alpha_j}^{\beta}(\mathbf{x}, \tilde{t}_j^*)$, we obtain that

$$|\mathrm{CVaR}_{\alpha_j}^{\beta}(\mathbf{x}) - \mathrm{CVaR}_{\alpha_j}(\mathbf{x})| \ \leq \ \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1-\alpha_j} \ \forall j \in [1, m].$$

2. Using the same previous argument but with respect to $\mathbf{x}$ instead of t allows to obtain the second inequality.

3. Consider $\mathbf{x} \in \mathcal{X}$ and suppose that Assumption 4.3 holds. It follows that for all $j \in [0, m]$ and $\boldsymbol{\xi} \in \Xi$

$$|C_j(\mathbf{x}, \boldsymbol{\xi})| - |C_j(\mathbf{0}, \mathbf{0})| \ \leq \ |C_j(\mathbf{x}, \boldsymbol{\xi}) - C_j(\mathbf{0}, \mathbf{0})| \ \leq \ \kappa_3(\boldsymbol{\xi}, \mathbf{0})||\mathbf{x}||,$$

which implies that $|C_j(\mathbf{x}, \boldsymbol{\xi})|$ is almost surely bounded by a function depending on $\mathbf{x}$. Now, for all $\mathbf{x} \in \mathcal{X}$, $M_j(\mathbf{x})$ is defined as the essential supremum of $C_j(\mathbf{x}, \boldsymbol{\xi})$, i.e,

$$M_j(\mathbf{x}) := \inf\{t \in \mathbb{R} \mid C_j(\mathbf{x}, \boldsymbol{\xi}) \leq t \text{ for almost every } \boldsymbol{\xi} \in \Xi\}.$$

Now, we have by definition

$$\mathrm{VaR}_{\alpha_j=1}(\mathbf{x}) \;=\; \inf\{t \mid \mathbb{P}(C_j(\mathbf{x}, \boldsymbol{\xi}) \leq t) = 1\} \;=\; M_j(\mathbf{x}).$$

As the c.d.f. of $C_j(\cdot, \boldsymbol{\xi})$ is assumed continuous, then it follows by [159] that

$$\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \;=\; \frac{1}{1-\alpha_j}\int_{\alpha_j}^{1} \mathrm{VaR}_{\tau}(\mathbf{x}) d\tau.$$

As for $\tau \in [\alpha_j, 1]$, the VaR_τ function is continuous with respect to τ with $\mathrm{VaR}_{\alpha_j}(\mathbf{x}) \leq \mathrm{VaR}_\tau(\mathbf{x}) \leq \mathrm{VaR}_{\alpha_j=1}(\mathbf{x}) = M_j(\mathbf{x})$, the mean value theorem ensures

$$\mathrm{VaR}_{\alpha_j}(\mathbf{x}) \;\leq\; \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \;\leq\; M_j(\mathbf{x}).$$

Thus, for all $\mathbf{x} \in \mathcal{X}$, $\lim_{\alpha_j \to 1} \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) = M_j(\mathbf{x})$ and we can set $\mathrm{CVaR}_{\alpha_j=1}(\mathbf{x}) = M_j(\mathbf{x})$ which ensures continuity of the CVaR_{α_j} function with respect to α_j for $\alpha_j \in (0, 1]$. Now,

$$\mathrm{CVaR}^{\beta}_{\alpha_j=1}(\mathbf{x}) \;=\; \mathrm{VaR}^{\beta}_{\alpha_j=1}(\mathbf{x}) \;=\; \inf\{t \mid \mathbb{P}(C_j(\mathbf{x} + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \beta_2 v_j \leq t) = 1\},$$

where the probability measure is taken with respect to $\boldsymbol{\xi}$, $\mathbf{u} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell - \mathbf{x}}{\beta_1}, \frac{\mathbf{b}_u - \mathbf{x}}{\beta_1})$ and $\mathbf{v} \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{-\mathbf{t}_{\max}}{\beta_2}, \frac{\mathbf{t}_{\max}}{\beta_2})$. It follows that

$$\mathrm{VaR}^{\beta}_{\alpha_j=1}(\mathbf{x}) \;=\; \sup_{\substack{\boldsymbol{\xi}\in\Xi,\, \mathbf{u}\in[\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta_1}, \frac{\mathbf{b}_u-\mathbf{x}}{\beta_1}],\\ v_j \in [(\frac{-\mathbf{t}_{\max}}{\beta_2})_j, (\frac{\mathbf{t}_{\max}}{\beta_2})_j]}} C_j(\mathbf{x} + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \beta_2 v_j \;=\; \sup_{\mathbf{x}\in\mathcal{X}} M_j(\mathbf{x}) + (\mathbf{t}_{\max})_j,$$

where the sup is understood as the essential supremum of the function. Thus, for any $\mathbf{x} \in \mathcal{X}$, $\mathrm{CVaR}_{\alpha_j=1}(\mathbf{x}) < \mathrm{CVaR}^{\beta}_{\alpha_j=1}(\mathbf{x})$. Therefore, by continuity of CVaR_{α_j} with respect to α_j, there exists $\bar{\alpha}_j \in (0, 1]$ such that for all $\alpha_j \geq \bar{\alpha}_j$

$$\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \;\leq\; \mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) \;\leq\; \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) + \frac{L_2\beta_1\sqrt{n} + \beta_2}{1-\alpha_j},$$

where the second inequality comes from the first part of the theorem. □

Theorem 6.5.2.2 shows that the difference in the values of objective function of Problem (6.14)

and Problem (6.5) is bounded by a constant that depends on the values of α_j, β_1, and β_2. Theorem 6.5.2.3 demonstrates that, with additional mild conditions, if α_j is chosen sufficiently close to 1, the solution obtained in Problem (6.14) is feasible for Problem (6.5). Therefore, the solution of Problem (6.14) may be feasible for Problem (6.5) and its value can be arbitrarily close to that of Problem (6.5) with sufficiently small values of β_1 and β_2. However, it is important to note that in practice, if β_1 and β_2 are chosen too small, the difference between the empirical values of the function will also be too small to represent the function differential [47].

To solve Problem (6.14) and to avoid the use of inner loops, which are computationally intractable, a Lagrangian relaxation is employed. This approach leads to the following unconstrained problem.

$$\max_{0\leq\boldsymbol{\lambda}\in\mathbb{R}^m} \min_{(\mathbf{x},\mathbf{t})\in\mathcal{X}\times\mathbb{R}^{m+1}} L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) := V^{\beta}_{\alpha_0}(\mathbf{x},t_0) + \sum_{j=1}^{m} \lambda_j V^{\beta}_{\alpha_j}(\mathbf{x},t_j), \tag{6.15}$$

where $\mathbf{t} = (t_0,\ldots,t_m) \in \mathbb{R}^{m+1}$. The next section describes a method allowing convergence to a saddle point of the Problem (6.15) whose the definition is recalled here.

Definition 6.5.3 (Saddle point)**.** *A saddle point of* $L(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})$ *is a point* $(\mathbf{x}^*,\mathbf{t}^*,\boldsymbol{\lambda}^*)$ *such that for some* $r>0$*,* $\forall\,(\mathbf{x},\mathbf{t}) \in \mathcal{X}\times\mathbb{R}^{m+1}\bigcap\mathcal{B}_{(\mathbf{x}^*,\mathbf{t}^*)}(r)$ *and for all* $\boldsymbol{\lambda}\geq 0$*, we have*

$$L(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}^*) \geq L(\mathbf{x}^*,\mathbf{t}^*,\boldsymbol{\lambda}^*) \geq L(\mathbf{x}^*,\mathbf{t}^*,\boldsymbol{\lambda}),$$

where $\mathcal{B}_{(\mathbf{x}^*,\mathbf{t}^*)}(r)$ *is a hyper-dimensional ball centred at* $(\mathbf{x}^*,\mathbf{t}^*)$ *with radius* $r>0$*.*

6.6 A Risk Averse Multi-timescale Stochastic Approximation Algorithm

Section 6.6.1 presents the multi-timescale stochastic approximation methods, and Section 6.6.2 describes the complete algorithm used to solve the Problem (6.15).

6.6.1 Multi-timescale Stochastic approximation methods

Multi-timescale is used to address the second difficulty raised at the end of Section 2, i.e., to avoid using nested loops to estimate a quantile of the level α and to compute the probabilistic constraints. Multi-timescale stochastic approximation [33, 38] is a method that utilizes updates with different step-size schedules. Multi-timescale algorithms are useful when, between two successive updates of the algorithm, an inner-loop procedure must be performed recursively until it converges. Employing a multi-timescale algorithm allows both updates (for the inner and outer loops) to run together and converge to the desired point. In conditional value-at-risk (CVaR) optimization, this

is typically the case for updating the additional variable $\mathbf{t}$ that could have been updated in an inner loop procedure. For example, the work [50, 152] use a multi-timescale algorithm to update the additional variable. Other cases where multi-timescale can be applied include aggregating information about the gradient through an exponential moving average and updating the Lagrangian multipliers in the case of a Lagrangian relaxation. For more details on multi-timescale stochastic approximation, readers may refer to [38, Chapter 6] or [33, Section 3.3].

In this work, four different timescales are used. The four different step sizes s_1^k, s_2^k, s_3^k and s_4^k are chosen so that Assumption 5 holds.

Assumption 5. *For $k \geq 0$, the step sizes sequences s_1^k, s_2^k, s_3^k and s_4^k are strictly positive and satisfy the requirements:*

$$\begin{aligned}
&\sum s_1^k = \sum s_2^k = \sum s_3^k = \sum s_4^k = +\infty, \\
&\sum \Big((s_1^k)^2 + (s_2^k)^2 + (s_3^k)^2 + (s_4^k)^2 \Big) < \infty, \\
&\lim_{k\to\infty} \frac{s_1^k}{s_2^k} = \lim_{k\to\infty} \frac{s_2^k}{s_3^k} = \lim_{k\to\infty} \frac{s_3^k}{s_4^k} = 0.
\end{aligned}$$

These four step sizes differ by their speed to reach the infinity. In fact, under the previous assumption, there exists an integer k_0 such that, for every $K \geq k_0$, the partial sums satisfy

$$\sum_{k=0}^{K} s_1^k < \sum_{k=0}^{K} s_2^k$$

and the gap between the above two summations increases with K. Thus, the time scale associated with s_2 is said to be faster than the time scale associated with s_1. In this work, the fastest timescale is used to aggregate information about the gradient, the first intermediate timescale is used to update the additional variable t and the second intermediate timescale is used to update the design vector $\mathbf{x}$, and the slowest timescale is used to update the Lagrangian multipliers $\boldsymbol{\lambda}$.

6.6.2 The RAMSA algorithm

Algorithm 15 summarizes the different updates. Note that when the square $(\cdot)^2$, the square root $\sqrt{\cdot}$ or division $\frac{\cdot}{\cdot}$ operators are applied to a vector, it is elementwise. Further remarks about Algorithm 15 are outlined:

- The updates (6.18) are the updates used to aggregate information about the gradient and are computed from the unbiased estimator defined in Equation (6.13). It will be shown later in the convergence proof that in fact $||\mathbf{M}^k - \nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda})|| \to 0$ and $||\mathbf{V}^k - (\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}))^2|| \to$

Algorithm 15 Risk Averse Multi-timescale Stochastic Approximation (RAMSA) algorithm

Input: $\mathbf{x}^0, \mathcal{X}, \mathcal{T}, \mathcal{L}, K^{\max}$.
Set $k = 0$ be an iteration counter
Define stepsize sequences (s_1^k), (s_2^k), (s_3^k) and (s_4^k) having the following form :

$$s_i^k = \frac{s_i^0}{(k+1)^{\tau_i}}, \forall i \in \{1, 2, 3, 4\}$$

where the exponential decays $\tau_i, i = 1, \ldots, 4$ are chosen such that the Assumption 5 are satisfied.
Set $\mathbf{M}^0 = \tilde{\mathbf{g}}^0, \mathbf{V}^0 = (\mathbf{M}^0)^2$ and $\mathbf{t}^0 = 0$
while $k \leq K^{\max}$ **do**
    Draw samples $\mathbf{u}^k \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{\mathbf{b}_\ell - \mathbf{x}^k}{\beta_1}, \frac{\mathbf{b}_u - \mathbf{x}^k}{\beta_1})$ and $\mathbf{v}^k \sim \mathcal{TN}(\mathbf{0}, \mathbf{I}, \frac{-\mathbf{t}_{\max} - \mathbf{t}^k}{\beta_2}, \frac{\mathbf{t}_{\max} - \mathbf{t}^k}{\beta_2})$.
    Recall that an unbiased output of the Lagrangian is given by:

$$\tilde{L}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}, \boldsymbol{\xi}) = \tilde{V}_{\alpha_0}(\mathbf{x}, t_0, \boldsymbol{\xi}) + \sum_{j=1}^{m} \lambda_j \tilde{V}_{\alpha_j}(\mathbf{x}, t_j, \boldsymbol{\xi}) \tag{6.16}$$

    where $\tilde{V}_{\alpha_j}(\mathbf{x}, t_j, \boldsymbol{\xi}) = t_j + \frac{1}{1-\alpha_j}(C_j(\mathbf{x}, \boldsymbol{\xi}) - t_j)^+$.
    Calculate the gradient estimate $\tilde{\mathbf{g}} := (\tilde{\mathbf{g}}_{\mathbf{x}}, \tilde{\mathbf{g}}_{\mathbf{t}}, \tilde{\mathbf{g}}_{\boldsymbol{\lambda}}) \in \mathbb{R}^n \times \mathbb{R}^{m+1} \times \mathbb{R}^m$ with respect to $\mathbf{x}$, $\mathbf{t}$ and $\boldsymbol{\lambda}$ with:

$$\begin{aligned}
\tilde{\mathbf{g}}_{\mathbf{x}}^k &= \frac{\left(\tilde{L}(\mathbf{x}^k + \beta_1 \mathbf{u}^k, \mathbf{t}^k + \beta_2 \mathbf{v}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_1^k) - \tilde{L}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_2^k)\right)(\mathbf{u}^k - \mu_1^k)}{\beta_1}, \\
\tilde{\mathbf{g}}_{\mathbf{t}}^k &= \frac{\left(\tilde{L}(\mathbf{x}^k + \beta_1 \mathbf{u}^k, \mathbf{t}^k + \beta_2 \mathbf{v}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_1^k) - \tilde{L}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_2^k)\right)(\mathbf{v}^k - \mu_2^k)}{\beta_2}, \\
\tilde{\mathbf{g}}_{\lambda_j}^k &= \tilde{V}_{\alpha_j}(\mathbf{x}^k, t_j^k, \boldsymbol{\xi}_1^k) \; \forall j \in [1, m].
\end{aligned} \tag{6.17}$$

    Update the long term gradient estimators:

$$\begin{aligned}
\mathbf{M}^{k+1} &= s_4^k \tilde{\mathbf{g}}^k + (1 - s_4^k)\mathbf{M}^k \\
\mathbf{V}^{k+1} &= s_4^k (\tilde{\mathbf{g}}^k)^2 + (1 - s_4^k)\mathbf{V}^k
\end{aligned} \tag{6.18}$$

    Update the current iterates $\mathbf{x}^k$, $\mathbf{t}^k$ and $\boldsymbol{\lambda}^k$;

$$\mathbf{t}^{k+1} = \Pi_{\mathcal{T}} \left[\mathbf{t}^k - s_3^k \frac{\mathbf{M}_{\mathbf{t}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{t}}^{k+1}} + \epsilon} \right] \tag{6.19}$$

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{X}} \left[x^k - s_2^k \frac{\mathbf{M}_{\mathbf{x}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{x}}^{k+1}} + \epsilon} \right] \tag{6.20}$$

$$\boldsymbol{\lambda}^{k+1} = \Pi_{\mathcal{L}} \left[\boldsymbol{\lambda}^k + s_1^k \frac{\mathbf{M}_{\boldsymbol{\lambda}}^{k+1}}{\sqrt{\mathbf{V}_{\boldsymbol{\lambda}}^{k+1}} + \epsilon} \right] \tag{6.21}$$

    $k \leftarrow k + 1$
end while
Return $\mathbf{x}^k$

0 almost surely when $k \to \infty$. The $\mathbf{M}^k$ iterates can be thought of as an exponential moving average of the gradient estimators and aim to aggregate information about the direction of the

gradient. The $\mathbf{V}^k$ iterates aim to avoid exploding gradient updates and aggregate information about the magnitude of the gradient.

- The update of the variable $\mathbf{t}$ is done in the update (6.19). The interest of updating $\mathbf{t}$ with a faster timescale than those of $\mathbf{x}$ is that $\mathbf{x}$ will be quasi static compared to $\mathbf{t}$. Thus, for a given $\mathbf{x}$, the updates of $\mathbf{t}$ will appear to have converged to a point $\mathbf{t}^*(\mathbf{x})$, where $\mathbf{t}^*$ is an estimate of the VaR at the point $\mathbf{x}$ of the objective and constraint functions.

- A projection is employed in the updates of the variables $\mathbf{x}$, $\mathbf{t}$ and $\boldsymbol{\lambda}$. This projection is required in the case of $\mathbf{x}$ because the space of the design variables is bounded. For $\mathbf{t}$ and $\boldsymbol{\lambda}$, the projection is required for convergence analysis. Since the bounds on $\mathbf{t}$ and $\boldsymbol{\lambda}$ can be arbitrarily large, this is not a problem in practice. In the algorithm, the sets $\mathcal{X}$, $\mathcal{T}$, and $\mathcal{L}$ are all hyper-rectangles, i.e., sets of type $[\mathbf{b}_\ell, \mathbf{b}_u] \subset \mathbb{R}^d$ where d is a given dimension. Furthermore, the projection operator $\Pi_{\mathcal{X}}(\mathbf{x})$ is defined as $\Pi_{\mathcal{X}}(\mathbf{x}) = (\Pi_1(x_1), \ldots, \Pi_d(x_d))$, where the individual projection operators $\Pi_j : \mathbb{R} \to \mathbb{R}$ are defined by $\Pi_j(x_j) = \min(\,(\mathbf{b}_u)_j, \max(\,(\mathbf{b}_\ell)_j, x_j))$ for all $j \in [1, d]$. The projection operators for the variables $\mathbf{t}$ and $\boldsymbol{\lambda}$ are defined in the same way.

6.7 Convergence analysis

The convergence of the RAMSA algorithm is stated in the following theorem.

Theorem 6.7.1. *Under Assumption 4.3 and Assumption 5, let further assume that the problem given in Equation (6.14) is strictly feasible and there exists $K \in \mathbb{N}$ such that $\mathbf{x}^K$ and $\boldsymbol{\lambda}^K$ are in the domain of attraction of $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ with $\boldsymbol{\lambda}^* \in \mathcal{L}^\circ$ respectively. Then, the iterates $(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$, produced by the RAMSA algorithm, converge almost surely to a saddle point of the Lagrangian function L^β and $(\mathbf{x}^*, \mathbf{t}^*)$ is a locally optimal solution for the smooth CVaR-constrained problem given in Equation (6.14).*

While the technical details of the proof of this theorem are given in Section 6.10.1, a high-level overview of the proof steps is given below.

- First, for each timescale, a discrete stochastic approximation analysis is used to prove the almost sure convergence of the iterates $(\mathbf{M}^k, \mathbf{V}^k, \mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$ to a stationary point $(\mathbf{M}^*, \mathbf{V}^*, \mathbf{x}^*, \mathbf{t}^*, \boldsymbol{\lambda}^*)$ of the corresponding continuous-time system.

- Then, to show that the continuous-time system is locally asymptotically stable at the stationary point, a Lyapunov analysis is performed.

- Finally, considering the iterates $(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$, the Lyapunov function used in the above analysis is the Lagrangian function $L(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})$. Therefore, the stationary point $(\mathbf{x}^*, \mathbf{t}^*, \boldsymbol{\lambda}^*)$ is a saddle point. Thus, by the saddle point theorem, we deduce that $\mathbf{x}^*$ is a locally optimal solution to the smooth CVaR-constrained blackbox optimization problem given in Equation (6.14).

This convergence proof procedure is standard for multi-timescale stochastic approximation algorithms, see [33, chapter 10], [38, chapter 6] or [33, 50], for further references. Note that this procedure must be done for each timescale, requiring four similar proof steps. This is due to the different speeds of the timescales. Here, the updates $(\mathbf{M}^k, \mathbf{V}^k)$ converge on a faster timescale than $\mathbf{t}^k$, which converges on a faster timescale than $\mathbf{x}^k$, while $\boldsymbol{\lambda}^k$ converges on the slowest timescale. The idea of multi-timescale convergence analysis is then to assume that, given a timescale, the updates made on faster timescales are quasi-equilibrated, i.e. have already converged to an equilibrium point. The updates made on slower timescales are quasi-static, i.e. fixed with respect to the given timescale. Therefore, the convergence analysis of the updates of the given timescale is done by considering all other updates as fixed. To illustrate the mathematical meaning of this assumption, consider two updates $\mathbf{x}^k, \mathbf{x}_2^k \in \mathcal{X}_1 \times \mathcal{X}_2$ such that

$$\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + s_1^k \left(f_1(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_1^{k+1} \right), \tag{6.22}$$

$$\mathbf{x}_2^{k+1} = \mathbf{x}_2^k + s_2^k \left(f_2(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_2^{k+1} \right), \tag{6.23}$$

where f_1 and f_2 are Lipschitz continuous function and δ_1, δ_2 are square integrable martingale difference sequence with respect to the σ-field $\sigma(\mathbf{x}_1^i, \mathbf{x}_2^i, \delta_1^i; i \leq k)$ and $\sigma(\mathbf{x}_1^i, \mathbf{x}_2^i, \delta_2^i; i \leq k)$. If s_1^k and s_2^k are non-summable and square summable step sizes with s_2^k which is a faster timescale than s_1^k, i.e., $s_1^k = o(s_2^k)$. Then, the previous recursion may be rewritten as follows

$$\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + s_2^k \left(\frac{s_1^k}{s_2^k} \left(f_1(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_1^{k+1} \right) \right), \tag{6.24}$$

$$\mathbf{x}_2^{k+1} = \mathbf{x}_2^k + s_2^k \left(f_2(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_2^{k+1} \right). \tag{6.25}$$

As $s_1^k = o(s_2^k)$, this recursion may be seen as a noisy discretization of the ODEs $\dot{\mathbf{x}_1} = 0$ and $\dot{\mathbf{x}_2} = f_2(\mathbf{x}_1, \mathbf{x}_2)$. Since $\dot{\mathbf{x}_1} = 0$, $\mathbf{x}_1$ is a constant and the second ODE may be replace with $\dot{\mathbf{x}_2} = f_2(\mathbf{x}_1^0, \mathbf{x}_2)$, where $\mathbf{x}_1^0$ is a constant. Finally it can be proved [38, Chapter 6, Theorem 2] that $(\mathbf{x}_1^k, \mathbf{x}_2^k)$ converge $(\mathbf{x}_1^*, \mu(\mathbf{x}_1^*))$, where μ is a Lipschitz continuous function, $\mu(\mathbf{x}_1^*)$ is a locally stable equilibrium of the ODE $\dot{\mathbf{x}_2} = f_2(\mathbf{x}_1^*, \mathbf{x}_2)$ and $\mathbf{x}_1^*$ is a locally stable equilibrium of the ODE $\dot{\mathbf{x}_1} = f_1(\mathbf{x}_1, \mu(\mathbf{x}_1))$.

In Theorem 6.7.1, it is proved that the iterations converge to a locally optimal solution of the problem given in Equation (6.14). It is possible to obtain a result for the original CVaR-constrained

problem given in Equation (6.5) by utilizing Theorem 6.5.2. This is the subject of the following corollary.

Corollary 6.7.2. *Under the same assumptions as Theorem 6.7.1, it follows that there exists a threshold* $\bar{\alpha} \in (0, 1]$ *such that if* $\alpha_j \geq \alpha$ *for all* $j \in [0, m]$*, then the iterates* $\mathbf{x}^k$ *converge almost surely to a feasible solution* $\mathbf{x}^*$ *of Problem (6.5) whose the objective function value is within* $\frac{L_2\beta_1\sqrt{n}+\beta_2}{1-\max_{j\in[1,m]}\alpha_j}$ *of that of a local solution of Problem (6.5).*

Proof. The proof is straightforward, considering the result of Theorems 6.7.1 and 6.5.2. □

This corollary is particularly interesting because it ensures the almost sure convergence of Algorithm 15 to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution. To the best of our knowledge, this result is the first of its kind in the area of derivative-free RBDO with unknown uncertainty distribution.

6.8 Computational implementations and numerical experiments

This section is divided into five parts: details of the numerical implementation are given in Section 6.8.1. Section 6.8.2 describes the setup of the experiments. Section 6.8.3 presents the experiments aimed at finding relations between the hyperparameters and the problems to be solved. Finally, Section 6.8.4 exhibits the results obtained using the truncated Gaussian gradient estimator instead of its classical counterpart, while Section 6.8.5 shows the results when the problem is subject to mixed aleatory/epistemic uncertainties.

6.8.1 Computational implementation

In this section, practical details of the implementation of Algorithm 15 are given. They aim to reduce the number of hyperparameters required by the algorithm and improved its practical efficiency.

The first difficulty the algorithm faces is when the bounds of the decision variables differ in magnitude. A first approach is then to adjust the initial step sizes according to each coordinate. However, this increases the number of hyperparameter values to be set. Another approach, which requires only one step size for all coordinates $j \in [0, m]$, is to map the initial hyperrectangle to the hypercube $[0, 1]^n$. The output of the blackbox $C_j : \mathcal{X} \to \mathbb{R}$ is simply replaced by $C_j^1 : [0, 1]^n \to \mathbb{R}$, where

$$C_j^1(\mathbf{x}, \boldsymbol{\xi}) = C_j(\mathbf{b}_\ell + (\mathbf{b}_u - \mathbf{b}_\ell)\mathbf{x}, \boldsymbol{\xi}).$$

The algorithm encounters a second difficulty related to the Lagrangian relaxation, where the values of the objective function and constraints are added together. When constraint magnitudes differ, the algorithm is biased towards the larger ones. To mitigate this bias, a solution consists of choosing different step sizes for updating $\boldsymbol{\lambda}$ but that increases the number of hyperparameters. Alternatively, a transformation may be applied to normalize the values, allowing the use of a single step size. In this method, the $\arctan(\cdot)$ function is employed to map the blackbox output values to the range of $[-\frac{\pi}{2}, \frac{\pi}{2}]$. However, there is an issue when the bounds of the $\arctan$ function are approached because the gradient estimator is computed from the difference between the values returned by the $\arctan$ function. If this difference is too small, especially in the presence of noisy blackbox outputs, the quality of the gradient estimator decreases. To address this issue, the cubic root function is applied beforehand to increase the difference between these values. That leads to the following transformation

$$C_j^2(\mathbf{x}, \boldsymbol{\xi}) = \arctan\left(\sqrt[3]{C_j(\mathbf{x}, \boldsymbol{\xi})}\right), \forall j \in [0, m].$$

In the rest of the paper, we refer to $\tilde{C}_j : [0,1]^n \to [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\forall j \in [0, m]$, the map corresponding to the two previous transformations applied to the outputs of the blackbox.

Finally, in practical applications, it appears that initiating the process directly at the intended reliability level can be counterproductive [199]. To overcome this difficulty, the values of $\alpha_j, \forall j \in [0, m]$ are initially set to 0. Then, these values are gradually increased until the desired reliability levels are reached. This is done by inserting reliability level setting

$$\alpha_j^{k+1} = \alpha_j^* + \gamma\left(\alpha_j^k - \alpha_j^*\right)$$

for every index $j \in [0, m]$ in between lines 12 and 13 of Algorithm 15. Here, α_j^* are the desired reliability levels and $\gamma \in [0, 1)$ is a fixed threshold.

6.8.2 Numerical experiments

Before proceeding to the numerical experiments, this section describes the test problems chosen, the way the experiments are performed, and the objectives of the different experiments.

First, four analytical test problems, each with a known practical optimum, are chosen from existing literature. These problems include a Steel Column Design (SCD) problem [194], a Welded Beam Design (WBD) problem [194], a Vehicle Side Impact (VSI) problem [194], and a Speed Reducer Design (SRD) problem [48]. These problems are decribed in Section 6.10.2, and further information regarding their physical interpretations can be found in the associated references. Except in

the last subsection, the goal is to solve the following standard RBDO problem

$$\begin{aligned} \min_{\mathbf{x}\in[0,1]^n} \quad & \mathbb{E}_{\boldsymbol{\xi}}[\tilde{C}_0(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} \quad & \mathbb{P}(\tilde{C}_j(\mathbf{x},\boldsymbol{\xi}) \leq 0) \geq 0.99,\ \forall j \in [1,m]. \end{aligned} \tag{6.26}$$

It is important to note that Problem (6.26), unlike the classical FORM-based problem, incorporates uncertainties not only in the constraints but also in the objective function. Moreover, despite the analytical expressions of the problems are available and the uncertainty distributions are known, the RAMSA algorithm operates without utilizing these information. As outlined in Sections 6.4 and 6.5 it solves formally a smooth Lagrangian relaxation of Problem (6.26).

In order to make comparisons, it is essential to devise a strategy for evaluating the quality of solutions generated by the RAMSA algorithm. As both the problem and the algorithm are subject to uncertainties, multiple runs of the RAMSA algorithm are necessary, and the values of the proposed solutions need to be estimated using Monte Carlo simulations. In this work, a trial consists of running the algorithm 100 times with the same set of hyperparameters values. For each run, a maximum budget of 5000 function evaluations is allocated. At the end of these 100 runs, the final solution points are recorded. For each solution point, the mean of the objective function and the probabilisty to satisfy the constraints are estimated through 10000 Monte Carlo simulations. A run is deemed successful if all constraints are satisfied with a probability greater than 0.99. Moreover, the mean solution point over the 100 runs, denoted as $\bar{\mathbf{x}}^*$, is calculated as well as its standard deviation. That allows to check that the RAMSA algorithm consistently converges to the same neighborhood of an optimal point. To further validate the results, this point is also compared with the solution obtained by the SORA algorithm in [48, 194]. Note that the aim is not to directly compare the RAMSA and SORA algorithms since the SORA algorithm takes advantage of the analytical expressions of the problems and knowledge of uncertainty distributions. When a trial is consistent for a set of hyperparameter, the set and the trial are said to be satisfactory.

Now, the objectives of the upcoming experimental sections are threefold. First, despite the transformations introduced in the previous section, there are still some hyperparameters that need to be configured. Section 6.8.3 provides guidelines on how to set these hyperparameters. Second, a critical aspect is the selection of the kernel density used to estimate gradients during the optimization process. In Section 6.8.4, a comparison is made between the classical Gaussian gradient estimator and the truncated Gaussian gradient estimator introduced in Section 6.5.1. Third, the VSI problem is described slightly differently in [194], allowing the means of the uncertainty variables $\boldsymbol{\xi}_8$ and $\boldsymbol{\xi}_9$ to take two values: 0.192 and 0.345. This is an opportunity to employ the RAMSA algorithm for solving the VSI problem under mixed aleatory/epistemic uncertainties. In fact, the uncertainty in distribution parameters can be regarded as a source of epistemic uncertainty [139]. Detailed

descriptions of the conducted experiments are presented in Section 6.8.5.

6.8.3 Hyperparameters setting rules

The RAMSA algorithmn involves four types of hyperparameters: the exponential decays of the step sizes $\tau \in (\frac{1}{2}, 1)^4$, the threshold for the adaptive reliability level γ, the initial step sizes $s^0 \in \mathbb{R}^4_+$, and the smoothing parameters $\beta \in \mathbb{R}^2_{+*}$. Two strategies can be employed to determine the values of these hyperparameters.

On the one hand, theoretical considerations are employed to set some hyperparameter values. This approach is employed to set the values of the exponential decays. These values must satisfy Assumption 5 to ensure the convergence of the algorithm. Moreover, they must be distinct enough to achieve the desired multi-timescale effect, but also not too different, otherwise, either the fastest timescale is too fast (leading to increased noise) or the slowest timescale is overly slow (impeding the convergence process) [33, Chapter 6]. Thus, the decays are arbitrarily set to $\tau = (0.8, 0.7, 0.6, 0.501)$. The threshold for the adaptive reliability level γ can be determined similarly. This hyperparameter depends only on the value of $K^{\max}$ because for $j \in [0, m]$, it follows that $\forall k \in \mathbb{N}$, $\alpha_j^{k+1} = \alpha_j^*(1 - \gamma^k)$. Thus γ can be chosen such that $\alpha_j^{K^{\max}} \approx \alpha_j^*$. However, if α_j^* is chosen close to 1, the problem given in Equation (6.14) is particularly conservative for Problem (6.5), as shown in Theorem 6.5.2, and even more so for Problem (6.26). Therefore, to avoid overly conservative results, γ is chosen to be equal to $1 - \frac{5}{2K^{\max}}$ so that $\alpha_j^{K_{\max}} \approx 0.9$ provided that $K_{\max} = 2500$ and $\alpha^* = 0.99$.

On the other hand, there are some hyperparameters values that cannot be determined theoretically. In this case, they have to be computed experimentally. This is achieved through a two-step strategy. The set of test problems is divided into two groups: the experimental test problems and the validation test problems. In the first step, for each experimental problem, a set of hyperparameters, that gives satisfactory results on this test problem, is identified. By analyzing the results obtained on the different problems and the associated hyperparameter values, a distinction may be deduced between the hyperparameters which are problem-dependent and which are not. For problem-dependent hyperparameters, we try to establish correlations between the hyperparameter values and relevant problem-related quantities. Examples of such quantities include the objective function value, the gradient norm, or its variance at the starting point. Then, the validation step is undertaken to check the rules derived from the experimental step. During this phase, the rules are applied to the validation test problems to determine the hyperparameter values of the RAMSA algorithm. If the results obtained with this set of hyperparameters are satisfactory, the rules are deemed effective.

In this study, the two-step strategy is applied as follows. The experimental test problems selected

are the VCD, WBD, and VSI problems. Trials of Algorithm 15 are conducted with different sets of hyperparameter values and the classical Gaussian gradient estimator [141, Equation (26)]. For the sake of brevity, only one set of satisfactory hyperparameters and its associated results are presented for each problem. The values of this set are listed in Table 6.2, while in Table 6.3 the associated average results of the trials are presented. Detailed results from the 100 runs of the trials are provided in Section 6.10.3 in the form of boxplots.

Table 6.2 Satisfactory set of hyperparameter values found for each problem

Problem	β_1	β_2	s_1^0	s_2^0	s_3^0	s_4^0
SCD	0.05	0.0001	0.01	0.05	0.001	0.2
WBD	0.002	0.0001	0.01	0.001	0.001	0.4
VSI	0.1	0.0001	0.01	0.5	0.001	0.5

Table 6.3 Average result over 100 runs obtained for each problem

Problem/ Algo	Average of $\mathbb{E}[C(\mathbf{x}^*, \boldsymbol{\xi})]$	Average of $\mathbb{P}(C_j(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0)$	Average result point $\bar{\mathbf{x}}^*$ (and standard deviation)	Number of successful runs	Function queries
SCD	3967	$[0.9938]$	$[229.7, 15.03, 103.1]$ $[\pm 4.4, \pm 0.25, \pm 3.8]$	100	5000
SORA	3989	$[0.9947]$	$[258, 13.5, 100]$	N/A	216
WBD	2.53	$[1.0, 1.0, 0.9995, 1.0, 1.0]$	$[6.36, 158, 211, 6.59]$ $[\pm 0.01, \pm 0.29, \pm 0.29, \pm 0.02]$	100	5000
SORA	2.49	$[1.0, 1.0, 1.0, 1.0, 1.0]$	$[5.92, 181, 211, 6.22]$	N/A	505
VSI	28.38	$[1.0, 1.0, 1.0, 1.0, 0.9993$ $1.0, 1.0, 0.9925, 1.0,$ $0.9996]$	$[0.88, 1.34, 0.51, 1.49,$ $[\pm 0.03, \pm 0.004, \pm 0.02, \pm 0.009]$ $1.29, 1.19, 0.45]$ $\pm 0.07, \pm 0.01, \pm 0.08]$	95	5000
SORA	29.55	$[1.0, 1.0, 1.0, 1.0, 0.9987,$ $1.0, 0.9987, 0.9983, 1.0,$ $0.9993]$	$[0.78, 1.35, 0.69, 1.5,$ $1.07, 1.2, 0.78]$	N/A	8054

Table 6.3 shows that the RAMSA algorithm achieves satisfactory results in all three problems. Interestingly, it appears to perform better on problems with higher dimensions and more constraints. This phenomenon can be attributed to the approximation of the gradient used in the RAMSA algorithm. This approximation estimates the gradient of the Lagrangian function with only two blackbox evaluations, regardless of the dimension or number of constraints. Upon analyzing Table 6.2, it seems that β_2, s_1^0, and s_3^0 are problem-independent. Moreover, the value of s_4^0 falls within a relatively narrow interval of $[0.1, 0.6]$. In contrast, the smoothing parameter β_1 and the initial step size s_2^0, both associated with the design vector $\mathbf{x}$, exhibit variations from one problem to another. This variability suggests the problem dependency of these hyperparameters.

The first claim to be proven experimentally is the following: an appropriate order of magnitude of β_1 is so that the variance of the gradient estimator at the starting point is minimal. A such value should reduce the variability during the initial stages of the optimization process and thus improve the convergence rate. To validate this assertion, the gradient is approximated by computing N Lagrangian gradient estimators given in Equation (6.17), at the point $(\mathbf{x}^0, \mathbf{0}, \mathbf{0})$. The gradient is approximated for only 6 different values of β_1 to prevent excessive computations. The values chosen are $[0.001, 0.005, 0.01, 0.05, 0.1, 0.2]$. Then, the variance of the first n components of the gradient (i.e., the components of $\tilde{\mathbf{g}}_{\mathbf{x}}$) is computed, and the average of these variances is calculated for each value of β_1. The value of β_1 is finally chosen as the one leading to the smallest average variance. If the minimum is reached for two different values of β_1, the larger value is selected. The results for the three different problems are presented in Table 6.4. It is observed that, selecting β_1 to minimize the average variance and halving it, yields to similar results to those of Table 6.2.

Table 6.4 Average variance of N gradient approximations for different values of the smoothing parameter β_1

Value of β_1	0.001	0.005	0.01	0.05	0.1	0.2
Average variance for SCD problem	4.2	0.16	0.04	0.004	0.003	0.94
Average variance for WBD problem	2.1	1.75	1.78	5.2	15	8.9
Average variance for VSI problem	0.67	0.03	0.008	0.0018	0.0016	0.0016

The second claim to be experimentally shown is that: there is a correlation between the norm of the stochastic gradient and the value of the initial step size s_2^0. Intuitively, that means that the smaller the gradient norm, the larger the initial step size should be, and vice versa. To validate this hypothesis, N stochastic gradients with $\beta_1 = 0.1$ are computed, and the norm of their mean is calculated. The result, normalized by the square root of the dimension, is presented in the third line of Table 6.5 for each problem. The second line displays the result obtained in Table 6.2, and the last line shows the corresponding correlation coefficients. Based on these results, it can be deduced that the correlation coefficient should be around 10^{-3}.

In the conducted experiments, the value of N is set to 10000. It is worth noting that while this large sample size is suitable for these experiments, in a BBO context, such a number might be intractable due to its computational cost. However, the methodology employed here can be adapted to work with smaller sample sizes. The goal of this approach is to provide only an order of magnitude for the hyperparameter values. Thus, a reduced number of samples can be used in a BBO context. Additionally, it is worth mentioning that the calculated gradients used to estimate the value of β_1

Table 6.5 Correlation between the norm of the gradient and the initial step size s_2^0

	SCD	WBD	VSI
Value of s_2^0	0.05	0.001	0.5
Estimated value of $\frac{\|\|\nabla_{\mathbf{x}} L^{\beta}(\mathbf{x}^0,\boldsymbol{\xi})\|\|_2}{\sqrt{n}}$	≈ 0.02	≈ 1.4	≈ 0.01
$s_2^0 \times \frac{\|\|\nabla_{\mathbf{x}} L^{\beta}(\mathbf{x}^0,\boldsymbol{\xi})\|\|_2}{\sqrt{n}}$	≈ 0.001	≈ 0.001	≈ 0.005

can also be used to estimate the value of s_2^0, reducing the computational cost of the method.

To validate the experimental step, the claims previously stated are applied to compute the hyperparameter values for solving the SRD problem. For this problem, the minimum value of the average variance occurs for $\beta_1 = 0.1$, and the norm of the Lagrangian gradient (normalized by the dimension) is estimated to be 0.006. These values are then utilized to set the values of $\beta_1 = 0.05$ and $s_2^0 = 0.15$. The values of the others hyperparameters are set as in Table 6.2 and $s_4^0 = 0.2$. The results obtained with this set of values are shown in Table 6.6.

Table 6.6 Average result over 100 runs for Speed Reducer design problem

Problem/ Algo	Average of $\mathbb{E}[C(\mathbf{x}^*,\boldsymbol{\xi})]$	Average of $\mathbb{P}(C_j(\mathbf{x}^*,\boldsymbol{\xi}) \leq 0)$	Average result point $\bar{\mathbf{x}}^*$ (and standard deviation)	Number of successful runs	Function queries
SRD	3148	$[1.0, 1.0, 1.0, 1.0, 1.0$ $1.0, 1.0, 0.9996, 1.0,$ $1.0, 1.0]$	$[3.6, 0.7, 17.0, 7.41,$ $[\pm 0.0, \pm 0.0, \pm 0.06, \pm 0.04]$ $7.99, 3.51, 5.37]$ $\pm 0.04, \pm 0.01, \pm 0.01]$	100	5000
SORA	3038	$[1.0, 1.0, 1.0, 1.0, 0.9975,$ $0.9986, 1.0, 0.9986, 1.0,$ $1.0, 0.9986]$	$[3.57, 0.7, 17, 7.3,$ $7.75, 3.36, 5.3]$	N/A	2486

Based on these results, it appears that the rules established for setting the hyperparameter values lead to satisfactory solutions. The consistency observed in the solution points, as indicated by the small standard deviations obtained, suggests that the algorithm consistently converges to the same vicinity. Moreover, this solution is relatively close to the optimal point found by the SORA algorithm. Note, however, that these rules do not guarantee to find the best possible set of hyperparameters. For example, by retaining all hyperparameter values but adjusting β_1 to 0.01, similar values of probabilistic constraints can be achieved, with an average objective function value of 3066.

In summary, the rules established in this section provide valuable insights into obtaining a satisfactory set of hyperparameter values for the RAMSA algorithm. However, they must be used with

caution due to the limited number of problems used to derive them, especially for the value of β_1. It is known [47] that setting the appropriate β_1 value is a challenging task in practice. One potential approach to address this challenge is to dynamically decrease the value of β_1 during the optimization process, as done in [22]. Nevertheless, this topic falls beyond the scope of the present paper and is not explored further here.

6.8.4 Truncated Gaussian vs Gaussian gradient estimator

In this section, the focus is on investigating the behavior of the algorithm when the bound constraints are unrelaxable [107], meaning that the outputs of the blackbox are not meaningful for the optimization process. This situation can arise when the blackbox is not defined outside its bounds or due to physical phenomenon. In this section, the uncertainties specified in Section 6.10.2 are truncated, ensuring that $\mathbf{x} + \boldsymbol{\xi} \in \mathcal{X}$ for every realization of $\boldsymbol{\xi}$. Moreover, to solve the constrained problem, the algorithm is executed using the truncated Gaussian gradient estimator instead of the classical Gaussian gradient estimator utilized in the previous section. This modification guarantees that all the candidate points are evaluated inside the bound constraints $\mathcal{X}$.

To determine the hyperparameter values for the algorithm using the truncated Gaussian gradient estimator, the methodology introduced in the previous section is applied. The values of β_1 that minimize the variance of the truncated Gaussian estimator are found to be $0.2, 0.005, 0.2$ and $0.01$for the SCD, WBD, VSI, and SRD problems, respectively. Consequently, the values of β_1 are set to $0.1, 0.0025, 0.1$ and 0.025. Furthermore, the correlation coefficient between the norm of the approximate gradient and the initial step size s_2^0 is approximately 5×10^{-4}. Thus, the values of s_2^0 are set to $0.1, 0.0008, 0.6$ and 0.01 for the SCD, WBD, VSI, and SRD problems, respectively. Finally, the values of s_4^0 are set to $0.25, 0.4, 0.6$, and 0.2. The results of these experiments are presented in Table 6.7, and the detailed results from the 100 runs are depicted in boxplots in Section 6.10.3.

In Table 6.7, it is shown that utilizing the truncated Gaussian gradient approximation leads to satisfactory results. However, the algorithm convergence is significantly slower than with classical Gaussian gradient approximation, requiring three times more function queries. This phenomenon cannot be attributed to the chosen hyperparameter values, as experiments with different sets of values do not significantly improve the results. Our main hypothesis is that this phenomenon may come from a side effect of using the truncated Gaussian distribution. However, a comprehensive investigation of this issue requires dedicated research, left for future work.

Table 6.7 Best average result over 100 runs obtained for each problem with truncated Gaussian gradient estimator with 15000 function evaluations by run

Problem	Average of $\mathbb{E}[C(\mathbf{x}^*, \boldsymbol{\xi})]$	Average of $\mathbb{P}(C_j(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0)$	Average result point $\bar{\mathbf{x}}^*$ (and standard deviation)	Number of successful runs
SCD	3957	$[0.9958]$	$[226, 15, 106]$ $[\pm 10, \pm 0.6, \pm 6]$	97
WBD	2.53	$[0.999, 1.0, 1.0, 1.0, 1.0]$	$[6.37, 158, 211, 6.59]$ $[\pm 0.01, \pm 0.24, \pm 0.24, \pm 0.01]$	99
VSI	28.97	$[1.0, 1.0, 1.0, 1.0, 0.9998$ $1.0, 1.0, 0.9923, 1.0,$ $0.9977]$	$[1.00, 1.35, 0.54, 1.49,$ $[\pm 0.03, \pm 0.004, \pm 0.03, \pm 0.008]$ $1.21, 1.19, 0.48]$ $\pm 0.1, \pm 0.008, \pm 0.1]$	91
SRD	3093	$[1.0, 1.0, 1.0, 1.0, 1.0$ $1.0, 1.0, 0.994, 1.0,$ $1.0, 0.999]$	$[3.58, 0.7, 17.0, 7.30,$ $[\pm 0.002, \pm 0.0, \pm 0.03, \pm 0.004]$ $7.78, 3.42, 5.31]$ $\pm 0.003, \pm 0.002, \pm 0.0008]$	100

6.8.5 Solving problems under mixed aleatory/epistemic uncertainties

In this section, the behavior of the algorithm in the presence of mixed aleatory and epistemic uncertainties is examined. Epistemic uncertainties may arise from uncertainties about distribution parameters [139]. In the VSI problem presented in [194], it is noted that the mean of the uncertainty variables ξ_8 and ξ_9 can take two different values: 0.192 and 0.345. While both values were fixed to 0.345 in [194] and in the previous experiments, in this section, these means are treated as epistemic uncertainties. Two types of epistemic uncertainty are studied: points epistemic uncertainty where the means μ_{ξ_8} and μ_{ξ_9} of ξ_8 and ξ_9 belong to $\{(0.192, 0.192),\ (0.192, 0.345),\ (0.345, 0.192),$ $(0.345, 0.345)\}$ and interval epistemic uncertainty where μ_{ξ_8} and μ_{ξ_9} belong to the same interval $[0.192, 0.345]$. The others uncertain variables remain the same (no truncated) and are considered as aleatory uncertainties.

In this type of problems, a solution is deemed feasible if, for any values μ_{ξ_8} and μ_{ξ_9} , the probabilistic constraints are satisfied with a probability greater than 0.99. Checking solution feasibility is more complex than in the previous section. In the case of points epistemic uncertainty, checking feasibility remains relatively straightforward since it involves evaluating the solution for the four possible pairs of means. However, when dealing with interval epistemic uncertainty, there is no ideal method for this verification. The approach adopted in this paper involves seeking the worst possible values of the epistemic uncertainties, μ_{ξ_8} and μ_{ξ_9}, at a candidate solution $\mathbf{x}^*$. To achieve

this, the following problem is solved for each constraint $C_j,\ j \in [1, m]$

$$\max_{(\mu_{\xi_8},\mu_{\xi_9})\in[0.192,0.345]^2} C_j(\mathbf{x}^*, \mathbb{E}[\boldsymbol{\xi}]). \tag{6.27}$$

This problem aims to find the most challenging combination of μ_{ξ_8} and μ_{ξ_9}. In this problem, all the uncertainties are fixed to their means and therefore the problem is deterministic. For each constraint, the couples solution of Problem (6.27) are recorded. Next, the aleatory uncertainties are introduced. For each pair of μ_{ξ_8} and μ_{ξ_9} obtained , the probabilities of satisfying the constraints at $\mathbf{x}^*$ are computed using the original distribution of the aleatory uncertainties. If these probabilities are all larger than 0.99, then the candidate solution is considered feasible. This approach provides a robust assessment of feasibility under interval epistemic uncertainty. It is noteworthy that applying this methodology to the solution point obtained by the SORA algorithm reveals that this point is infeasible in the presence of epistemic uncertainty. For instance, if the means μ_{ξ_8} and μ_{ξ_9} are taken to be equal to $(0.192, 0.345)$, the probability of satisfying the 7th constraint is $\mathbb{P}\left(C_7(\mathbf{x}^*_{SORA}, \boldsymbol{\xi}) \leq 0\right) \approx 0.88$.

Table 6.8 Average result over 100 runs obtained with mixed aleatory/points epistemic uncertainty in the VSI problem with 15000 function evaluations by run

Value of $(\mu_{\xi_8}, \mu_{\xi_9})$	Average of $\mathbb{E}[C(\mathbf{x}^*, \boldsymbol{\xi})]$	Average of $\mathbb{P}(C_j(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0)$	Average result point $\bar{\mathbf{x}}^*$ (and standard deviation)	Number of successful runs
$(0.192, 0.192)$	30.38	$[1.0, 1.0, 1.0, 0.9986, 1.0$ $1.0, 0.9993, 0.9930, 1.0, 1.0]$	$[1.27, 1.35, 0.51, 1.49,$ $[\pm 0.05, \pm 0.0, \pm 0.02, \pm 0.007$ $1.26, 1.19, 0.47]$ $\pm 0.08, \pm 0.01, \pm 0.1]$	98
$(0.192, 0.345)$		$[1.0, 1.0, 1.0, 1.0, 1.0,$ $1.0, 0.9993, 0.9930, 1, 0.9995]$		98
$(0.345, 0.192)$		$[1.0, 1.0, 1.0, 1.0, 1.0,$ $1.0, 1.0, 0.9930, 1.0, 1.0]$		98
$(0.345, 0.345)$		$[1.0, 1.0, 1.0, 1.0, 1.0,$ $1.0, 1.0, 0.9929, 1.0, 0.9995]$		99

To address this type of problems with the RAMSA algorithm, it is necessary to associate a probability distribution with the mean of ξ_8 and ξ_9. It is important to underline that this does not imply making an assumption about the distribution of the epistemic uncertainty itself. The distribution is just utilized to generate blackbox outputs. That allows to approach the problem from a worst-case perspective, leveraging the CVaR properties when the values of α_j are taken sufficiently close to 1. In the algorithm, the Bernoulli distribution is employed to generate the means for points epistemic uncertainty, while the uniform distribution is used to generate the means for interval epistemic uncertainty. The results for mixed aleatory/points epistemic uncertainties are presented in Table 6.8,

and for mixed aleatory/interval epistemic uncertainties in Table 6.9.

Table 6.9 Average result over 100 runs obtained with mixed aleatory/interval epistemic uncertainty in the VSI problem with 10000 function evaluations by run.

Solution of Problem (6.27)[1]	Average of $\mathbb{E}[C(\mathbf{x}^*, \boldsymbol{\xi})]$	Average of $\mathbb{P}(C_j(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0)$	Average result point $\bar{\mathbf{x}}^*$ (and standard deviation)	Number of successful runs
$(0.192, 0.345)$	29.71	$[1.0, 1.0, 1.0, 1.0, 1.0$ $1.0, 0.9974, 0.9929, 1.0, 0.9994]$	$[1.15, 1.35, 0.51, 1.49,$ $[\pm 0.04, \pm 0.00005, \pm 0.01, \pm 0.01$ $1.23, 1.19, 0.48]$ $\pm 0.07, \pm 0.02, \pm 0.1]$	99

[1] There is only one point because for each run, the solutions $(\mu_{\xi_8}, \mu_{\xi_9})$ of (6.27) are always the same.

In both cases, the RAMSA algorithm achieves satisfactory results. An interesting observation is that the results obtained with mixed aleatory/interval epistemic uncertainties are better to those with mixed aleatory/points epistemic uncertainties. This observation might appear counterintuitive since, in this experiment, points epistemic uncertainty is a subset of interval epistemic uncertainty. However, this phenomenon could be explained because the algorithm is better at handling continuous distributions than discrete distributions. The continuous nature of interval epistemic uncertainty could potentially make it more amenable for the gradient estimator, leading to enhanced performance in these cases.

6.9 Concluding remarks

This work targets the constrained blackbox optimization problem given in Equation (6.1), where the output of the blackbox is subject to uncertainties. To deal with the uncertainties, a CVaR-constrained problem formulation is adopted. This formulation allows the selection of the desired level of reliability. A smooth approximation of the CVaR-constrained problem is then derived by convolving the objective and constraint functions with a truncated multivariate Gaussian density. The use of the truncated Gaussian density, as opposed to the classical Gaussian density, ensures that sampling points are drawn within the bound constraints. Consequently, this approach avoids numerical failures that may occur when functions are undefined outside their bounds. Then, a Lagrangian relaxation is applied to handle the constraints. The resulting Lagrangian function possesses several appealing properties for optimization. First, it is infinitely differentiable since it is a sum of smooth approximations of the objective and constraint functions. Second, gradient estimators of the Lagrangian function can be computed with only two noisy blackbox outputs, making it computationally efficient. Theoretical bounds on the quality of the approximation have been derived. These bounds depend on the size of the problem, the value of the smoothing parameters, and

the desired level of reliability. It is worth noting that it has been proved that for a reliability level sufficiently close to 1, a feasible solution of the approximated problem remains a feasible solution of the original CVaR-constrained problem.

A new algorithm has been proposed to find a saddle point of the Lagrangian function. This algorithm is based on multi-timescale stochastic approximation updates. In this work, four different timescales are used. On the fastest timescale, the updates aggregate information about the gradient of the smooth Lagrangian function. On a first intermediate timescale, they estimate the value-at-risk of the objective and constraint functions. On a second intermediate timescale, the updates compute the optimal solution with respect to $\mathbf{x}$, while on the slowest timescale, the updates compute the optimal values of the Lagrangian multipliers. A convergence analysis based on Lyapunov theory shows that the different updates almost surely converge to a saddle point of the Lagrangian function. This point is locally optimal for the smooth approximation of the CVaR-constrained problem. Furthermore, using the previous result on the quality of the approximation, we prove that for reliability level values sufficiently close to one, this point is feasible and its value may be arbitrarily close to an optimal value of the CVaR-constrained problem.

Once theoretical results have been stated, details of the numerical implementations are given. These details mainly concern two transformations: one mapping the design variables into $[0,1]^n$ and another mapping the blackbox outputs into $[-\frac{\pi}{2},\frac{\pi}{2}]^{m+1}$. These transformations are designed to scale the design variables and the blackbox outputs, effectively reducing the number of hyperparameters. Then, numerical experiments are performed. In these experiments, the primary objective is to establish rules for selecting the values of the remaining hyperparameters. The results reveal that all hyperparameter values, except two, are independent of the problem and can be pre-specified using the values determined in this work. The first problem-dependent hyperparameter identified is the initial value of the step size for updating $\mathbf{x}$. It is determined that this value can be estimated from the norm of the gradient estimator at the starting point. The second problem-dependent hyperparameter is the value of the smoothing parameter. It is found that this parameter can be chosen in such a way that its value minimize the variance of the gradient estimator at the starting point.

The secondary objective is to compare the effectiveness of the methods when truncated Gaussian gradient estimators are used instead of the classical Gaussian gradient estimator. The proposed strategy for setting the hyperparameters is applied to experiments conducted with the truncated Gaussian gradient estimator. However, its use come at a cost. In the conducted experiments, it is observed that the truncated estimator is approximately three times less efficient than the classical Gaussian gradient estimator in terms of blackbox evaluations.

The tertiary objective of the experiments is to apply the algorithm to problems involving mixed aleatory/epistemic uncertainties. In these experiments, the epistemic uncertainties are related to

the parameter distribution of the uncertainty variables. Two types of epistemic uncertainty are explored: points epistemic uncertainty and interval epistemic uncertainty. The algorithm demonstrated significant efficacy in handling both types of uncertainties. Notably, it performed particularly well in cases involving interval uncertainties, yielding promising results.

Future work will focus on validating these results using real-world industrial test cases. Additionally, there are plans to compare the RAMSA algorithm with other state-of-the-art algorithms to further assess its performance and competitiveness on problems subject to mixed aleatory/epistemic uncertainties.

6.10 Appendix

6.10.1 Appendix A. Proof of Theorem 6.7.1

First, two technical lemmas are stated to show that the iterates $\mathbf{M}^k$ and $\mathbf{V}^k$ are uniformly bounded almost surely. For this purpose, properties about the random gradient estimator must be shown.

Lemma 6.10.1. *Under Assumption 4.3, the random gradient estimator* $\tilde{\mathbf{g}} := (\tilde{\mathbf{g}}_{\mathbf{x}}, \tilde{\mathbf{g}}_{\mathbf{t}}, \tilde{\mathbf{g}}_{\boldsymbol{\xi}})$ *is almost surely Lipschitz continuous with respect to* $\mathbf{x}, \mathbf{t}$ *and* $\boldsymbol{\lambda}$*. Moreover,* $||\tilde{\mathbf{g}}||$ *is almost surely bounded.*

Proof. Let $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^2$, $(\mathbf{t}, \mathbf{s}) \in \mathbb{R}^{m+1} \times \mathbb{R}^{m+1}$ and consider any fixed realization of $\mathbf{u}, \mathbf{v}, \boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$, it follows that for $\alpha_j \in (0, 1)$

$$
\begin{aligned}
&|\tilde{V}_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u},\ (t_j + \beta_2 v_j, \boldsymbol{\xi}_1) - \tilde{V}_{\alpha_j}(\mathbf{y} + \beta_1 \mathbf{u}, s_j + \beta_2 v_j, \boldsymbol{\xi}_2)| \\
&\leq |t_j - s_j| + \left| \Big(C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - \ (t_j + \beta_1 v_j) \Big)^+ - \Big(C_j(\mathbf{y} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_2) - \ (s_j + \beta_1 v_j) \Big)^+ \right| \\
&\leq 2|t_j - s_j| + |C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_2)| \leq 2|t_j - s_j| + L_3 ||\mathbf{x} - \mathbf{y}|| \text{ a.s. },
\end{aligned}
$$

where the second inequality follows from $|\max(a, 0) - \max(b, 0)| \leq |a - b|$ and the third is due to Assumption 4.3. Therefore, $\tilde{V}_{\alpha_j}$ is almost surely Lipschitz continuous with respect to $\mathbf{x} \in \mathcal{X}$ and $t_j \in \mathbb{R}$. As, $\tilde{L}$ is a sum of almost surely Lipschitz continuous functions with respect to $\mathbf{x}$ and $\mathbf{t}$, it is also an almost surely Lipschitz continuous function. Moreover, $\tilde{L}$ is a linear function with respect to λ and thus Lipschitz continuous with respect to λ.

Finally, by Assumption 4.3, we have for all $\mathbf{x} \in \mathbf{X}$ and $\boldsymbol{\xi} \in \Xi$

$$
|C_j(\mathbf{x}, \boldsymbol{\xi})| - |C_j(\mathbf{0}, \mathbf{0})| \leq |C_j(\mathbf{x}, \boldsymbol{\xi}) - C_j(\mathbf{0}, \mathbf{0})| \leq \kappa_3(\boldsymbol{\xi}, \mathbf{0}) ||\mathbf{x}||.
$$

Thus, the function C_j is almost surely bounded. Since $\tilde{L}$ is a sum of almost surely bounded functions, $\mathbf{x}, \mathbf{t}$ and $\boldsymbol{\lambda}$ are taken in compact sets and $\mathbf{v}$ and $\mathbf{u}$ are truncated Gaussian random vectors, it follows directly that $||\tilde{\mathbf{g}}||$ is almost surely bounded. □

Once this was shown, $\mathbf{M}^k$ and $\mathbf{V}^k$ may be bounded.

Lemma 6.10.2. *The sequence of updates* $\mathbf{M}^k$ *and* $\mathbf{V}^k$ *are uniformly bounded with probability one.*

Proof. Let $k \in \mathbb{N}$, we have

$$\mathbf{M}^{k+1} = s_4^k \tilde{\mathbf{g}}^k + \sum_{r=0}^{k-1} s_4^l \prod_{q=r}^{k-1} (1 - s_4^{q+1}) \tilde{\mathbf{g}}^r + \prod_{q=0}^{k} (1 - s_4^q) \tilde{\mathbf{g}}^0.$$

It follows directly by triangular inequality that

$$||\mathbf{M}^{k+1}|| \leq s_4^k ||\tilde{\mathbf{g}}^k|| + \sum_{r=0}^{k-1} s_4^l \prod_{q=r}^{k-1} (1 - s_4^{q+1}) ||\tilde{\mathbf{g}}^r|| + \prod_{q=0}^{k} (1 - s_4^q) ||\tilde{\mathbf{g}}^0||.$$

Now according to Lemma 6.10.1, for all $r \in \mathbb{N}$, the random gradient estimator is almost surely bounded. Therefore, we have

$$||\mathbf{M}^{k+1}|| \leq \left(s_4^k + \sum_{r=0}^{k-1} s_4^l \prod_{q=r}^{k-1} (1 - s_4^{q+1}) + \prod_{q=0}^{k} (1 - s_4^q) \right) \sup_{r \in [0,k]} ||\tilde{\mathbf{g}}^r|| < +\infty.$$

The same arguments may be applied for $\mathbf{V}^k$, thus the claim follows directly. □

The remainder of the section is composed of four steps.

Step 1: Convergence of $\mathbf{M}$ and $\mathbf{V}$ updates. Since $\mathbf{M}$ and $\mathbf{V}$ converge on the fastest timescale, according to Lemma 1 in [38, chapter 6], the convergence properties of the updates in Equation (5.8) may be analyzed for arbitrary quantities of $\mathbf{x}$, $\mathbf{t}$ and $\boldsymbol{\lambda}$ (here $\mathbf{x} = \mathbf{x}^k$, $\mathbf{t} = \mathbf{t}^k$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^k$ are used). These updates may be rewritten as follows

$$\mathbf{M}^{k+1} = \mathbf{M}^k + s_4^k \left(\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) - \mathbf{M}^k + \delta_{\mathbf{M}}^{k+1} \right), \tag{6.28}$$

$$\mathbf{V}^{k+1} = \mathbf{V}^k + s_4^k \left((\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 + \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) - \mathbf{V}^k + \delta_{\mathbf{V}}^{k+1} \right), \tag{6.29}$$

where $\delta_{\mathbf{M}}^{k+1} = \tilde{\mathbf{g}}^k - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$ and $\delta_{\mathbf{V}}^{k+1} = (\tilde{\mathbf{g}}^k)^2 - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)^2 - \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$, with $\mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) = \mathbb{E}[(\tilde{\mathbf{g}}^k - \mathbb{E}[\tilde{\mathbf{g}}^k | \mathcal{F}^k])^2 | \mathcal{F}^k]$ the variance conditioned by the associated sigma field $\mathcal{F}^k = \sigma(\mathbf{x}^r, \mathbf{t}^r, \boldsymbol{\lambda}^r, \mathbf{M}^r, \mathbf{V}^r; r \leq k)$. Now, the following Lemma may be stated to prove the convergence properties of the updates $\mathbf{M}$ and $\mathbf{V}$.

Lemma 6.10.3. *Consider the following continuous time system dynamics of the updates,*

$$\begin{aligned} \dot{\mathbf{M}} &= h_1(\mathbf{M}, \mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) := \nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) - \mathbf{M}, \\ \dot{\mathbf{V}} &= h_2(\mathbf{M}, \mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) := (\nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) - \mathbf{V}, \\ (\dot{\mathbf{x}}, \dot{\mathbf{t}}, \dot{\boldsymbol{\lambda}}) &= (\mathbf{0}, \mathbf{0}, \mathbf{0}). \end{aligned} \tag{6.30}$$

This o.d.e. has a globally asymptotically stable equilibrium

$$\Big\{\Big(\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}), \nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}), \mathbf{x},\mathbf{t},\boldsymbol{\lambda}\Big) \;\Big|\; (\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) \in \mathcal{X}\times\mathcal{T}\times\mathcal{L}\Big\},$$

and the sequences $(\mathbf{M}^k, \mathbf{v}^k, \mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$ *converge almost surely to this equilibrium.*

Proof. The proof may be decomposed in two parts: the first part consists of analyzing the solutions of the two first o.d.e. given in Equation (6.30) and the second part consists of verifying that all the assumptions needed to apply Lemma 1 in [38, Chapter 6] are satisfied.

First, let $(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) \in \mathcal{X}\times\mathcal{T}\times\mathcal{L}$ be fixed and consider the following functions,

$$\begin{aligned}\mathcal{L}^1_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{M}) &= ||\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) - \mathbf{M}||^2,\\ \mathcal{L}^2_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{V}) &= ||(\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) - \mathbf{V}||^2.\end{aligned}$$

Let denote $\mathbf{M}^* = \nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})$ and $\mathbf{V}^* = (\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})$ the equilibrium points of the two first equations in Equation (6.30). The both functions satisfy the following conditions:

- They are globally positive definite, i.e, $\mathcal{L}^1_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{M}) > 0$, for all $\mathbf{M} \neq \mathbf{M}^*$ and $\mathcal{L}^2_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{V}) > 0$, for all $\mathbf{V} \neq \mathbf{V}^*$.
- They are radially unbounded since $||\mathbf{M}|| \to \infty \implies \mathcal{L}^1_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{M}) \to \infty$ and $||\mathbf{V}|| \to \infty \implies \mathcal{L}^2_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{V}) \to \infty$.
- The time derivatives of the both functions are globally negative definite since $\frac{d}{d\tau}\mathcal{L}^1_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{M}(\tau)) = -2||\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) - \mathbf{M}(\tau)||^2$ and $\frac{d}{d\tau}\mathcal{L}^2_{\mathbf{x},\mathbf{t},\boldsymbol{\lambda}}(\mathbf{V}(\tau)) = -2||(\nabla L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) - \mathbf{V}(\tau)||^2$.

Thus, both functions are Lyapunov functions associated to the two first o.d.e. given in Equation (6.30). By a corollary of the LaSalle invariance theorem (see for instance [91, Corollary 4.2]), the equilibrium points M^* and V^* are globally asymptotically stable. Moreover, ∇L^{β} is Lipschitz with respect to $\mathbf{x},\mathbf{t}$ and $\boldsymbol{\lambda}$ since it is a continuously differentiable function defined on a bounded space. The same may be applied for the function $(\nabla L^{\beta})^2$. Finally, the function $\mathbb{V}$ is also Lipschitz, since $\tilde{\mathbf{g}}$ is Lipschitz by Lemma 6.10.1.

Now, we use the framework of the Lemma 1 in [38, Chapter 6].

(i) By Lemma 6.10.2, the updates $\mathbf{M}^k$ and $\mathbf{V}^k$ are uniformly bounded almost surely. The same goes for the updates $\mathbf{x}^k, \mathbf{t}^k$ and $\boldsymbol{\lambda}^k$ because of the projection operator.

(ii) The functions h_1 and h_2 are Lipschitz continuous with respect to $\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}, \mathbf{M}$ and $\mathbf{V}$ by properties of ∇L^β and $\mathbf{V}$.

(iii) The sequence $(\delta_{\mathbf{M}}^{k+1})$ is a martingale difference sequence with respect to the increasing sigma fields $\mathcal{F}^k = \sigma(\mathbf{x}^r, \mathbf{t}^r, \boldsymbol{\lambda}^r, \mathbf{M}^r, \mathbf{V}^r; r \leq k)$ since, by properties of truncated Gaussian smoothing, it follows that

$$\mathbb{E}[\delta_{\mathbf{M}}^{k+1}|\mathcal{F}^k] = \mathbb{E}[\tilde{\mathbf{g}}^k|\mathcal{F}^k] - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) = 0.$$

This sequence is also square integrable since

$$\mathbb{E}[||\delta_{\mathbf{M}}^{k+1}||^2|\mathcal{F}^k] \leq 2(\mathbb{E}[||\tilde{g}||^2|\mathcal{F}^k] + \mathbb{E}[||\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)||^2]) < \infty,$$

because $||a-b||^2 \leq 2(||a||^2 + ||b||^2)$, $\tilde{g}$ is almost surely bounded by Lemma 6.10.1 and ∇L^β is a continuous function taking inputs in a compact set.

(iv) The sequence $(\delta_{\mathbf{V}}^{k+1})$ is a martingale difference sequence with respect to $\mathcal{F}^k$ since

$$\mathbb{E}[\delta_{\mathbf{V}}^{k+1}|\mathcal{F}^k] = \mathbb{E}[(\tilde{\mathbf{g}}^k)^2|\mathcal{F}^k] - \ (\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 - \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) = 0,$$

by definition of conditional variance $\mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) = \mathbb{E}[(\tilde{\mathbf{g}}^k)^2|\mathcal{F}^k] - \ (\mathbb{E}[\tilde{\mathbf{g}}^k|\mathcal{F}^k])^2$ and is square integrable

$$\mathbb{E}[||\delta_{\mathbf{V}}^{k+1}||^2|\mathcal{F}^k] \leq 2(\mathbb{E}[||(\tilde{\mathbf{g}})^2||^2] + ||(\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 + \mathbb{V}[\tilde{\mathbf{g}}|\mathcal{F}^k]||^2 < +\infty,$$

thanks to the same arguments as for $\delta_{\mathbf{M}}^{k+1}$.

(v) Finally, the step sizes s_1^k, s_2^k, s_3^k and s_4^k satisfy Assumption 5.

Under these conditions, Lemma 1 in [38, Chapter 6] may be applied, and the claim follows directly. □

Step 2: Convergence of the $\mathbf{t}$-update. The $\mathbf{t}$-update converges on a faster timescale than the ones on $\mathbf{x}$ and $\boldsymbol{\lambda}$, while $\mathbf{M}$ and $\mathbf{V}$ converge faster than $\mathbf{t}$, thus, according to Lemma 1 in [38, Chapter 6] the convergence of the $\mathbf{t}$ update may be proved for any arbitrary $\boldsymbol{\lambda}$ and $\mathbf{x}$ (here $\mathbf{x} = \mathbf{x}^k$ and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^k$ are taken). Furthermore, in the $\mathbf{M}$-updates and $\mathbf{V}$-updates, as a result of Lemma 6.10.3 the following limits hold $||\mathbf{M}^k - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \to 0$ and $||\mathbf{V}^k - \ (\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 - \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \to 0$ almost surely. Consequently, by defining

$$\nabla_{\mathbf{t}}^k L^\beta = \nabla_{\mathbf{t}} L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) \text{ and } \mathbb{V}_{\mathbf{t}}^k = \mathbb{V}_{\mathbf{t}}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k),$$

the update on $\mathbf{t}$ may be rewritten as follows

$$\mathbf{t}^{k+1} = \Pi_{\mathcal{T}}\left[\mathbf{t}^k + s_3^k\left(-\Psi_{\mathbf{x}^k,\boldsymbol{\lambda}^k}(\mathbf{t}^k) + \delta_{\mathbf{t}}^{k+1}\right)\right],$$
$$\text{where } \begin{cases} \Psi_{\mathbf{x}^k,\boldsymbol{\lambda}^k}(\mathbf{t}^k) & = \frac{\nabla_{\mathbf{t}}^k L^\beta}{\sqrt{(\nabla_{\mathbf{t}}^k L^\beta)^2 + \mathbb{V}_{\mathbf{t}}^k} + \epsilon}, \\ \delta_{\mathbf{t}}^{k+1} & = \Psi_{\mathbf{x}^k,\boldsymbol{\lambda}^k}\left(\mathbf{t}^k\right) - \frac{\mathbf{M}_{\mathbf{t}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{t}}^{k+1}} + \epsilon}. \end{cases} \tag{6.31}$$

Now, the following Lemma may be stated to prove the convergence properties of the update $\mathbf{t}$.

Lemma 6.10.4. *Consider the following continuous time system dynamics of the updates,*

$$\dot{\mathbf{t}} = \Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right] = \Gamma_{\mathbf{t}}\left[\frac{-\nabla_{\mathbf{t}} L^\beta(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\sqrt{\nabla_{\mathbf{t}} L^\beta(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) + \mathbb{V}_{\mathbf{t}}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})} + \epsilon}\right], \tag{6.32}$$
$$(\dot{\mathbf{x}},\dot{\boldsymbol{\lambda}}) = (\mathbf{0},\mathbf{0}),$$

where

$$\Gamma_{\mathbf{t}}[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})] := \lim_{0<\eta\to 0} \frac{\Pi_{\mathcal{T}}[\mathbf{t} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})] - \Pi_{\mathcal{T}}[\mathbf{t}]}{\eta}.$$

This o.d.e. has an asymptotically globally stable equilibrium

$$\left\{\left(\mathbf{x}, \mathbf{t}^*(\mathbf{x},\boldsymbol{\lambda}), \boldsymbol{\lambda}\right) \;\middle|\; (\mathbf{x},\boldsymbol{\lambda}) \in \mathcal{X}\times\mathcal{L}\right\},$$

where $t^*(\mathbf{x},\boldsymbol{\lambda}) = \{\mathbf{t} \mid \Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right] = 0\}$ *and the sequences* $(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k)$ *converge almost surely to this equilibrium.*

It is worth noting that $\Gamma_{\mathbf{t}}[K(\mathbf{t})]$ is the left directional derivative of the function $\Pi_{\mathbf{t}}[\mathbf{t}]$ in the direction of $K(\mathbf{t})$. By using the left directional derivative $\Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right]$ in the gradient descent algorithm for $\mathbf{t}$, the gradient will point in the descent direction along the boundary of $\mathcal{T}$ whenever the $\mathbf{t}$-update hits its boundary.

Proof. Similar to the analysis made for the $\mathbf{M}$-update and $\mathbf{V}$-update, the proof is decomposed in two parts. First, the solution of the first o.d.e. given in Equation (6.32) is described. Let $(\mathbf{x},\boldsymbol{\lambda}) \in \mathcal{X}\times\mathcal{L}$ be fixed and consider the following function

$$\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right) = L^\beta(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) - L^\beta(\mathbf{x},\mathbf{t}^*,\boldsymbol{\lambda}),$$

where $\mathbf{t}^*$ is a minimum point (for any $(\mathbf{x},\boldsymbol{\lambda})$, the function L^β is convex in $\mathbf{t}$). This function satisfies the following conditions:

- The function is positive definite since $\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}) > 0$, for all $\mathbf{t} \neq \mathbf{t}^*$ and radially unbounded since $||\mathbf{t}|| \to \infty, \implies \mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}) \to \infty$.

- The time derivative of the function is

$$\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)}{d\tau} = \nabla_{\mathbf{t}} L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})^T \, \Gamma_{\mathbf{t}} \left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right]$$

  and the goal is to show that this quantity is negative definite. There are two sets of cases to study:

  - The cases where $\mathbf{t} \in \mathcal{T}^{\circ} = \mathcal{T} \backslash \partial\mathcal{T}$. In all this cases, there exist $\eta > 0$ sufficiently small such that $t - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}) \in \mathcal{T}$, therefore by definition of $\Gamma_{\mathbf{t}}$ and $\Psi_{\mathbf{t}}$, it follows that (recall that the operators on the vectors are elementwise):

$$\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)}{d\tau} = -\sum_{j=0}^{m} \frac{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2}{\sqrt{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2 + \mathbb{V}_{t_j}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) + \epsilon}}.$$

  - The cases where $\mathbf{t} \in \partial\mathcal{T}$. When $\mathbf{t} \in \partial\mathcal{T}$, the indices $j \in [0,m]$ of the variables of $\mathbf{t}$ may be grouped in three complementary sets : $S^{\min} = \{j \in [0,m] \mid t_j = -\left(\mathbf{t}_{\max}\right)_j\}$, $S^{\max} = \{j \in [0,m] \mid t_j = \left(\mathbf{t}_{\max}\right)_j\}$ or $S^{\circ} = \{j \in [0,m] \mid t_j = \left(-\left(\mathbf{t}_{\max}\right)_j, \left(\mathbf{t}_{\max}\right)_j\right)\}$. Then, for the variables t_j whose the indices are in S°, there exists $\eta > 0$, sufficiently small such that $\left(t - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)\right)_j \in \left(-\left(\mathbf{t}_{\max}\right)_j, \left(\mathbf{t}_{\max}\right)_j\right)$. For the variables t_j whose the variables are in $S^{\min}$, then either $\left(\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)\right)_j \leq 0$, so $\Pi_{t_j}[(-\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right))_j] = \left(-\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)\right)_j$; or $\left(\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)\right)_j > 0$ so $\Pi_{t_j}[-\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right))_j] = -\left(\mathbf{t}_{\max}\right)_j$. For the variables t_j whose the variables are in $S^{\max}$, the symmetric result may be obtained. Therefore, it follows that

$$\begin{aligned}
\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right)}{d\tau} &= \lim_{0<\eta\to 0} \nabla_{\mathbf{t}} L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})^T \left(\frac{\Pi_{\mathbf{t}}[\mathbf{t} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})] - \mathbf{t}}{\eta}\right) \\
&= \lim_{0<\eta\to 0} \Bigg(-\sum_{j\in S^{\circ}} \frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j} \left(\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right)_j \\
&\qquad - \sum_{j\in S^{\min}} \frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j} \frac{\Pi_{t_j}[(-\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right))_j] + \left(\mathbf{t}_{\max}\right)_j}{\eta} \\
&\qquad - \sum_{j\in S^{\max}} \frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j} \frac{\Pi_{t_i}[(\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}\left(\mathbf{t}\right))_j] - \left(\mathbf{t}_{\max}\right)_j}{\eta} \Bigg)
\end{aligned}$$

$$\leq -\sum_{j\in S^{\circ}} \frac{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2}{\sqrt{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2 + \mathbb{V}_{t_j}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) + \epsilon}},$$

Therefore, $\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}\,(\mathbf{t})}{d\tau} < 0$ whenever $\Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right] \neq 0$, i.e, is negative definite.

Thus, the function $\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}$ is a Lyapunov function and by [91, Corrolary 4.2], the equilibrium point $t^*(\mathbf{x},\boldsymbol{\lambda}) = \{\mathbf{t} \mid \Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right] = 0\}$ is globally asymptotically stable. Moreover, since ∇L^{β} is Lipschitz continuous with respect to $\mathbf{x}$ and $\boldsymbol{\lambda}$, it follows that $t^*(\mathbf{x},\boldsymbol{\lambda})$ is Lipschitz continuous with respect to these vectors as well. Now, the framework of the Lemma 1 and Theorem 2 in [38, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma 6.10.3 are still satisfied.
- The function $\Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right]$ is Lipschitz continuous by properties of ∇L^{β}.
- The random sequence (δ_t^{k+1}) converges asymptotically to 0 by Lemma 6.10.3.

Therefore, the $\mathbf{t}$-update is a stochastic approximation with a null martingale difference sequence term and an additional error term δ_t^{k+1}. Then, by applying Theorem 2 in [38, Chapter 6] and the enveloppe theorem [50, Theorem 16], the claim follows directly. □

Step 3: Convergence of the $\mathbf{x}$-update. The convergence of the $\mathbf{x}$-update is very similar to the convergence of the $\mathbf{t}$-update. The $\mathbf{x}$-update converges on a faster timescale than the one of $\boldsymbol{\lambda}$, while $\mathbf{t}$, $\mathbf{M}$ and $\mathbf{V}$ converge on faster timescales than $\mathbf{x}$, thus, according to [38, Chapter 6] the convergence of the $\mathbf{x}$ update may be proved for any arbitrary $\boldsymbol{\lambda}$ (here $\boldsymbol{\lambda} = \boldsymbol{\lambda}^k$ is taken). Furthermore, in the $\mathbf{t}$, $\mathbf{M}$ and $\mathbf{V}$ updates, as a result of Lemma 6.10.3 and Lemma 6.10.4 the following limits hold $||\mathbf{M}^k - \nabla L^{\beta}(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k)|| \to 0$, $||\mathbf{V}^k - (\nabla L^{\beta}(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k))^2 - \mathbb{V}(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k)|| \to 0$ and $||\mathbf{t}^k - \mathbf{t}^*(\mathbf{x}^k,\boldsymbol{\lambda}^k)|| \to 0$ almost surely. Consequently by defining

$$\nabla_{\mathbf{x}}^k L^{\beta} = \nabla_{\mathbf{x}} L^{\beta}(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k) \text{ and } \mathbb{V}_{\mathbf{x}}^k = \mathbb{V}_{\mathbf{x}}(\mathbf{x}^k,\mathbf{t}^k,\boldsymbol{\lambda}^k),$$

the update on $\mathbf{x}$ may be rewritten as follows

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}^k + s_2^k\left(-\Psi_{\boldsymbol{\lambda}^k}(\mathbf{x}^k) + \delta_{1,\mathbf{x}}^{k+1} + \delta_{2,\mathbf{x}}^{k+1}\right)\right], \tag{6.33}$$

where

$$\Psi_{\boldsymbol{\lambda}^k}(\mathbf{x}^k) = \frac{\nabla_{\mathbf{x}} L^\beta(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)}{\sqrt{(\nabla_{\mathbf{x}} L^\beta(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k))^2 + \mathbb{V}_{\mathbf{x}}(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)} + \epsilon},$$
$$\delta_{1,\mathbf{x}}^{k+1} = \frac{\nabla_{\mathbf{x}}^k L^\beta}{\sqrt{(\nabla_{\mathbf{x}}^k L^\beta)^2 + \mathbb{V}_{\mathbf{x}}^k} + \epsilon} - \frac{\mathbf{M}_{\mathbf{x}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{x}}^{k+1}} + \epsilon},$$
$$\delta_{2,\mathbf{x}}^{k+1} = \Psi_{\boldsymbol{\lambda}^k}(\mathbf{x}^k) - \frac{\nabla_{\mathbf{x}}^k L^\beta}{\sqrt{(\nabla_{\mathbf{x}}^k L^\beta)^2 + \mathbb{V}_{\mathbf{x}}^k} + \epsilon}.$$

Now, the following Lemma may be stated to prove the convergence properties of the update $\mathbf{x}$.

Lemma 6.10.5. *Consider the following continuous time system dynamics of the updates,*

$$\dot{\mathbf{x}} = \Gamma_{\mathbf{x}}\left[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})\right] = \Gamma_{\mathbf{x}}\left[\frac{-\nabla_{\mathbf{x}} L^\beta(\mathbf{x}, \mathbf{t}^*(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda})}{\sqrt{\nabla_{\mathbf{x}} L^\beta(\mathbf{x}, \mathbf{t}^*(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}) + \mathbb{V}_{\mathbf{x}}(\mathbf{x}, \mathbf{t}^*(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda})} + \epsilon}\right], \tag{6.34}$$
$$\dot{\boldsymbol{\lambda}} = \mathbf{0},$$

where

$$\Gamma_{\mathbf{x}}[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})] := \lim_{0<\eta\to 0} \frac{\Pi_{\mathcal{X}}[\mathbf{x} - \eta\Psi_{\boldsymbol{\lambda}}(\mathbf{x})] - \Pi_{\mathcal{X}}[\mathbf{x}]}{\eta}.$$

Assume there exists $K_1 \in \mathbb{N}$ such that $\mathbf{x}^{K_1}$ is in the domain of attraction of $\mathbf{x}^*$ where $\mathbf{x}^*$ is some local minimum of L^β with respect to $\mathbf{x}$. Then, this o.d.e. has a locally asymptotically stable equilibrium

$$\left\{\left(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}\right) \,\middle|\, \boldsymbol{\lambda} \in \mathcal{L}\right\}, \tag{6.35}$$

where $\mathbf{x}^*(\boldsymbol{\lambda}) = \{\mathbf{x} \in \mathcal{X} \mid \Gamma_{\mathbf{x}}\left[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})\right] = 0\}$ is the local minima of the assumption and the sequences $(\mathbf{x}^k, \boldsymbol{\lambda}^k)$ converge almost surely to the set given in Equation (6.35).

Proof. First, the solutions of the first o.d.e. in Equation (6.34) is described. Let $\boldsymbol{\lambda} \in \mathcal{L}$ be fixed and consider the following function

$$\mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{x}) = L^\beta(\mathbf{x}, \mathbf{t}^*(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda}) - L^\beta(\mathbf{x}^*, \mathbf{t}^*(\mathbf{x}^*, \boldsymbol{\lambda}), \boldsymbol{\lambda}),$$

where $\mathbf{x}^*$ is the local minimum in $\mathcal{X}$ defined in the statement of the Lemma. This function is locally positive definite and its time derivatives is

$$\frac{d\mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{x})}{d\tau} = \nabla_{\mathbf{x}} L^\beta(\mathbf{x}, \mathbf{t}^*(\mathbf{x}, \boldsymbol{\lambda}), \boldsymbol{\lambda})^T \, \Gamma_{\mathbf{x}}\left[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})\right],$$

which is negative definite (the proof may be done in the exact same way as the one given in Lemma 6.10.4

and is omitted here). Therefore, the function is a Lyapunov function and, by Lyapunov stability theorem [91, Theorem 4.1], $\mathbf{x}^*(\boldsymbol{\lambda}) = \{\mathbf{x} \mid \Gamma_{\mathbf{x}}[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})] = 0\}$ is a locally asymptotically stable equilibrium. Since ∇L^β is Lipschitz continuous with respect to $\boldsymbol{\lambda}$, it follows that $\mathbf{x}^*(\boldsymbol{\lambda})$ is Lipschitz as well. Now, the framework in [38, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma 6.10.3 are still satisfied.
- The function $\Gamma_{\mathbf{x}}\left[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})\right]$ is Lipschitz continuous by properties of ∇L^β.
- The random sequence $(\delta_{1,\mathbf{x}}^{k+1})$ and $(\delta_{2,\mathbf{x}}^{k+1})$ converges asymptotically to 0 by Lemma 6.10.3 and Lemma 6.10.4.

By assumption, the iterates $\mathbf{x}^{K_1}$ belongs to the domain of attraction of $\mathbf{x}^*$ for some $K_1 \in \mathbb{N}$. By definition of the domain of attraction, $\mathbf{x}^k$ is in the domain of attraction for all $k \geq K_1$. Thus, by applying Theorem 2 in [38, Chapter 6] from the iteration K, the claim follows directly. □

At this stage, the results obtained in Lemma 6.10.4 and Lemma 6.10.5 allows concluding that for any fixed $\boldsymbol{\lambda} \in \mathcal{L}$, the following holds:

$$(\mathbf{x}^k, \mathbf{t}^k) \rightarrow \ (\mathbf{x}^*(\boldsymbol{\lambda}), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda})) \in \mathcal{X} \times \mathcal{T}.$$

Moreover, $\mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda})$ is a minimum of L^β with respect to $\mathbf{t}$ while $\mathbf{x}^*(\boldsymbol{\lambda})$ is a local minimum of L^β with respect to $\mathbf{x}$. Since we have

$$\min_{\mathbf{x}\in\mathcal{X}} \left(\min_{\mathbf{t}\in\mathcal{T}} L(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) \right) = \min_{(\mathbf{x},\mathbf{t})\in\mathcal{X}\times\mathcal{T}} L(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}),$$

it follows that this point is a local minimum for the function L^β.

Step 4: Convergence of the $\boldsymbol{\lambda}$-update. Since the $\boldsymbol{\lambda}$-update converges in the slowest time scale, according to previous analysis, the following limits hold $||\mathbf{M}^k - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \rightarrow 0$, $||\mathbf{V}^k - (\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 - \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \rightarrow 0$, $||\mathbf{t}^k - \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k)|| \rightarrow 0$ and $||\mathbf{x}^k \rightarrow \mathbf{x}^*(\boldsymbol{\lambda})|| \rightarrow 0$, almost surely. Therefore, by defining

$$\nabla_{\boldsymbol{\lambda}}^k L^\beta = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k), \mathbb{V}_{\boldsymbol{\lambda}}^k = \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k), \nabla_{\boldsymbol{\lambda}}^* L^\beta = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)$$
$$\text{and } \mathbb{V}_{\boldsymbol{\lambda}}^* = \mathbb{V}(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k),$$

the $\boldsymbol{\lambda}$-update rule can be re-written as follows

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{L}} \left[\boldsymbol{\lambda}^k + s_1^k \left(\Psi(\boldsymbol{\lambda}^k) + \delta_{1,\boldsymbol{\lambda}}^{k+1} + \delta_{2,\boldsymbol{\lambda}}^{k+1} + \delta_{3,\boldsymbol{\lambda}}^{k+1} \right) \right], \tag{6.36}$$

where

$$\Psi(\boldsymbol{\lambda}^k) = \frac{\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)}{\sqrt{(\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k))^2 + \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)} + \epsilon},$$

$$\delta_{1,\boldsymbol{\lambda}}^{k+1} = \frac{\mathbf{M}_{\boldsymbol{\lambda}}^{k+1}}{\sqrt{\mathbf{V}_{\boldsymbol{\lambda}}^{k+1}} + \epsilon} - \frac{\nabla_{\boldsymbol{\lambda}}^k L^\beta}{\sqrt{(\nabla_{\boldsymbol{\lambda}}^k L^\beta)^2 + \mathbb{V}_{\boldsymbol{\lambda}}^k} + \epsilon},$$

$$\delta_{2,\boldsymbol{\lambda}}^{k+1} = \frac{\nabla_{\boldsymbol{\lambda}}^k L^\beta}{\sqrt{(\nabla_{\boldsymbol{\lambda}}^k L^\beta)^2 + \mathbb{V}_{\boldsymbol{\lambda}}^k} + \epsilon} - \frac{\nabla_{\boldsymbol{\lambda}}^* L^\beta}{\sqrt{(\nabla_{\boldsymbol{\lambda}}^* L^\beta)^2 + \mathbb{V}_{\boldsymbol{\lambda}}^*} + \epsilon},$$

$$\delta_{3,\boldsymbol{\lambda}}^{k+1} = \frac{\nabla_{\boldsymbol{\lambda}}^* L^\beta}{\sqrt{(\nabla_{\boldsymbol{\lambda}}^* L^\beta)^2 + \mathbb{V}_{\boldsymbol{\lambda}}^*} + \epsilon} - \Psi(\boldsymbol{\lambda}^k).$$

Now, the following Lemma may be stated to prove the convergence properties of the update $\boldsymbol{\lambda}$.

Lemma 6.10.6. *Let consider the following continuous time system dynamics of the updates,*

$$\begin{aligned}\dot{\boldsymbol{\lambda}} &= \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right] \\ &= \Gamma_{\boldsymbol{\lambda}}\left[\frac{\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)}{\sqrt{(\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k))^2 + \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)} + \epsilon}\right],\end{aligned} \tag{6.37}$$

where

$$\Gamma_{\boldsymbol{\lambda}}[\Psi(\boldsymbol{\lambda})] := \lim_{0<\eta\to 0} \frac{\Pi_{\mathcal{L}}[\boldsymbol{\lambda} - \eta\Psi(\boldsymbol{\lambda})] - \Pi_{\mathcal{L}}[\boldsymbol{\lambda}]}{\eta}.$$

Assume there exists $K_2 \in \mathbb{N}$ such that $\boldsymbol{\lambda}^{K_2}$ is in the domain of attraction of $\boldsymbol{\lambda}^*$ where $\boldsymbol{\lambda}^*$ is some local maximum of L^β with respect to $\boldsymbol{\lambda}$. Then, this o.d.e. has a locally asymptotically stable equilibrium

$$\boldsymbol{\lambda}^* = \{\boldsymbol{\lambda} \in \mathcal{L} \mid \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right] = 0\}, \tag{6.38}$$

and the sequences $(\boldsymbol{\lambda}^k)$ converges almost surely to this local maximum given in Equation (6.38).

Proof. The proof is analog to the proof of convergence for the $\mathbf{x}$-update. First, the solutions of the first o.d.e. in Equation (6.37) is described. Let consider the following function

$$\mathcal{L}(\boldsymbol{\lambda}) = -L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}), \boldsymbol{\lambda}) + L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^*), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*),$$

where $\boldsymbol{\lambda}^*$ is the local maximum in $\mathcal{L}$ defined in the statement of the Lemma. This function is locally positive definite and its time derivatives is

$$\frac{d\mathcal{L}\left(\boldsymbol{\lambda}\right)}{d\tau} = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}), \boldsymbol{\lambda})^T \, \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right],$$

which is negative definite (the proof may be done in the exact same way as the one given in Lemma 6.10.4 and is omitted here). Therefore, the function is a Lyapunov function and, by Lyapunov stability theorem [91, Theorem 4.1], $\boldsymbol{\lambda}^* = \{\boldsymbol{\lambda} \mid \Gamma_{\boldsymbol{\lambda}}[\Psi(\boldsymbol{\lambda})] = 0\}$ is a locally asymptotically stable equilibrium. Now, the framework in [38, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma 6.10.3 are still satisfied.
- The function $\Gamma_{\boldsymbol{\lambda}}[\Psi(\boldsymbol{\lambda})]$ is Lipschitz continuous by properties of ∇L^{β}.
- The random sequence $(\delta_{1,\boldsymbol{\lambda}}^{k+1})$, $(\delta_{2,\boldsymbol{\lambda}}^{k+1})$ and $(\delta_{3,\boldsymbol{\lambda}}^{k+1})$ converges asymptotically to 0 by Lemma 6.10.3, Lemma 6.10.4 and Lemma 6.10.5.

By assumption, the iterates $\boldsymbol{\lambda}^k$ belongs to the domain of attraction of $\boldsymbol{\lambda}^*$ for some $K_2 \in \mathbb{N}$. By definition of the domain of attraction, $\boldsymbol{\lambda}^k$ is in the domain of attraction for all $k \geq K_2$. Thus, by applying Theorem 2, in [38, Chapter 6] from the iteration $K = \max(K_1, K_2)$, the claim follows directly. □

Main result: convergence to a saddle point. By letting $\mathbf{x}^* = \mathbf{x}^*(\boldsymbol{\lambda}^*)$ and $\mathbf{t}^* = \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^*), \boldsymbol{\lambda}^*)$, it will be shown that $(\mathbf{x}^*, \mathbf{t}^*, \boldsymbol{\lambda}^*)$ is a saddle point of the Lagrangian function L^{β} if $\boldsymbol{\lambda}^* \in \mathcal{L}^{\circ}$ and thus by the saddle point theorem, $\mathbf{x}^*$ is a locally optimal solution for the smooth CVaR-constrained problem given in Equation (6.14). This result is formally settled in Theorem 6.7.1 which is recalled here;

Theorem 6.10.7. *Under Assumption 4.3 and Assumption 5, let further assume that the problem given in Equation (6.14) is strictly feasible and there exists $K \in \mathbb{N}$ such that $\mathbf{x}^K$ and $\boldsymbol{\lambda}^K$ are in the domain of attraction of $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ with $\boldsymbol{\lambda}^* \in \mathcal{L}^{\circ}$ respectively. Then, the iterates $(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)$ converge almost surely to a saddle point of the Lagrangian function L^{β} and $\mathbf{x}^*$ is a locally optimal solution for the smooth CVaR-constrained problem given by Equation (6.14).*

Proof. Under the assumptions of the theorem, since $(\mathbf{x}^*, \mathbf{t}^*)$ is a local minimum of $L^{\beta}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})$ over the bounded set $(\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \mathcal{T}$, there exists a $r > 0$ such that

$$L^{\beta}(\mathbf{x}^*, \mathbf{t}^*, \boldsymbol{\lambda}^*) \leq L^{\beta}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}^*),\ \forall\, (\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \mathcal{T} \cap \mathcal{B}_r(\mathbf{x}^*, \mathbf{t}^*).$$

In order to complete the proof, we must show that for all $j \in [1, m]$

$$c_j(\mathbf{x}^*, \mathbf{t}^*) := t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \ (t_j^* + \beta_2 v_j))^+] \leq 0 \ \text{ and} \tag{6.39}$$

$$\lambda_j^* c_j(\mathbf{x}^*, \mathbf{t}^*) = \lambda_j^* \left(t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \ (t_j^* + \beta_2 v_j))^+]\right) = 0. \tag{6.40}$$

The proof of the inequality given in Equation (6.39) is made by contradiction. Suppose that

$$c_j(\mathbf{x}^*, \mathbf{t}^*) = t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \ (t_j^* + \beta_2 v_j))^+] > 0.$$

This implies for $\lambda_j \in \mathcal{L}_j^\circ$ that for any $\eta \in \ (0, \bar{\eta}]$

$$\begin{aligned}\Pi_{\mathcal{L}}\left[\lambda_j^* - \eta\left(t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - \ (t_j^* + \beta_2 v_j))^+]\right)\right] &= \Pi_{\mathcal{L}}\left[\lambda_j^* - \eta c_j(\mathbf{x}^*, \mathbf{t}^*)\right] \\ &= \lambda_j^* - \eta c_j(\mathbf{x}^*, \mathbf{t}^*),\end{aligned}$$

with $\bar{\eta}$ sufficiently small. Therefore, it follows that $\Gamma_{\lambda_j}[\ (\Psi(\boldsymbol{\lambda}^*))_j] = c_j(\mathbf{x}^*, \mathbf{t}^*) > 0$, which contradicts the definition of $\boldsymbol{\lambda}^*$ given in Equation (6.38). Thus, the inequality given in Equation (6.39) holds. To show the result given in Equation (6.40), it is sufficient to show that $\lambda_j^* = 0$ when $c_j(\mathbf{x}^*, \mathbf{t}^*) < 0$. For $\lambda_j^* \in \mathcal{L}^\circ$, there exists a sufficiently small $\eta > 0$ such that

$$\frac{\Pi_{\mathcal{L}}\left[\lambda_j^* + \eta c_j(\mathbf{x}^*, \mathbf{t}^*)\right] - \lambda_j^*}{\eta} = c_j(\mathbf{x}^*, \mathbf{t}^*) < 0.$$

This is again in contradiction with the definition of $\boldsymbol{\lambda}^*$ given in Equation (6.38) and thus the equality in Equation (6.40) holds. Finally, by the local saddle point theorem, it follows that $\mathbf{x}^*$ is a locally optimal solution for the smooth CVaR-constrained problem given by Equation (6.14). □

6.10.2 Appendix B. Analytical problems description

Here are the list of analytical problems considered in Section 6.8.1.

Steel column problem [194]

- Dimension: $n = 3$ and $m = 1$.
- Original lower bounds: $\mathbf{b}_\ell = \ (200, 10, 100)$
- Original upper bounds: $\mathbf{b}_u = \ (400, 30, 500)$
- Original $\mathbf{x}_0$: $(200, 10.5, 100)$

- Equations:

$$C_0(\mathbf{x}, \boldsymbol{\xi}) = (x_1 + \xi_1)(x_2 + \xi_2) + 5(x_3 + \xi_3),$$
$$C_1(\mathbf{x}, \boldsymbol{\xi}) = F\left(\frac{1}{A_s} + \frac{\xi_8 e_b}{U_s(e_b - F)}\right) - \xi_4,$$
$$\text{with } A_s = 2(x_1 + \xi_1)(x_2 + \xi_2),\ U_s = (x_1 + \xi_1)(x_2 + \xi_2)(x_3 + \xi_3),\ e_b = \frac{\pi^2 \xi_9 U_i}{L^2},$$
$$U_i = \frac{1}{2}(x_1 + \xi_1)(x_2 + \xi_2)(x_3 + \xi_3)^2 \text{ and } F = \xi_5 + \xi_6 + \xi_7.$$

- Uncertainties: $\xi_1 \sim \mathcal{N}(0, 0.1x_1)$, $\xi_2 \sim \mathcal{N}(0, 0.1x_2)$, $\xi_3 \sim \mathcal{N}(0, 0.1x_3)$, $\xi_4 \sim \mathcal{N}(400, 40)$, $\xi_5 \sim \mathcal{N}(5 \times 10^5, 5 \times 10^4)$, $\xi_6 \sim \mathcal{N}(6 \times 10^5, 6 \times 10^4)$, $\xi_7 \sim \mathcal{N}(6 \times 10^5, 6 \times 10^4)$, $\xi_8 \sim \mathcal{N}(30, 3)$, $\xi_9 \sim \mathcal{N}(21000, 2100)$ and $L = 7500$.

- Solution in [194]: $\mathbf{x}^* = (257.7806, 13.5335, 100)$ with $\mathbb{E}[C_0(\mathbf{x}^*, \boldsymbol{\xi})] = 3988.95$ and $\mathbb{P}(C_1(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0) = 0.9947$ (estimated in this work from 10^6 samples).

Welded Beam problem [194]

- Dimension: $n = 4$ and $m = 5$.

- Original lower bounds: $\mathbf{b}_\ell = (3.175, 0.0, 0.0, 0.0)$

- Original upper bounds: $\mathbf{b}_u = (50.8, 254, 254, 50.8)$

- Original $\mathbf{x}_0$: $(6.208, 157.82, 210.62, 6.208)$

- Equations:

$$C_0(\mathbf{x}, \boldsymbol{\xi}) = \kappa_1(x_1 + \xi_1)^2(x_2 + \xi_2) + \kappa_2\,(x_3 + \xi_3)(x_4 + \xi_4)(\kappa_3 + x_2 + \xi_2)$$
$$C_1(\mathbf{x}, \boldsymbol{\xi}) = \frac{\tau}{93.77} - 1 \text{ with}$$
$$\tau = \sqrt{\tau_1^2 + 2\frac{\tau_1 \tau_2 (x_2 + \xi_2)}{2R} + \tau_2^2},\ \tau_1 = \frac{\kappa_4}{\sqrt{2}(x_1 + \xi_1)(x_2 + \xi_2)},$$
$$R = \frac{\sqrt{(x_2 + \xi_2)^2 + (x_1 + \xi_1 + x_3 + \xi_3)^2}}{2},\ M = \kappa_4\left(\kappa_3 + \frac{x_2 + \xi_2}{2}\right),$$
$$J = \sqrt{2}(x_1 + \xi_1)(x_2 + \xi_2)\left(\frac{(x_2 + \xi_2)^2}{12} + \frac{(x_1 + \xi_1 + x_3 + \xi_3)^2}{4}\right),\ \tau_2 = \frac{MR}{J},$$

$$C_2(\mathbf{x},\boldsymbol{\xi}) = \frac{\sigma}{206.85} - 1 \text{ with } \sigma = \frac{6\kappa_4\kappa_3}{(x_3+\xi_3)^2(x_4+\xi_4)},$$
$$C_3(\mathbf{x},\boldsymbol{\xi}) = \frac{x_1+\xi_1}{x_4+\xi_4} - 1,$$
$$C_4(\mathbf{x},\boldsymbol{\xi}) = \frac{\delta}{6.35} - 1 \text{ with } \delta = \frac{4\kappa_4\,(\kappa_3)^3}{2.0685\times 10^5 (x_3+\xi_3)^3(x_4+\xi_4)},$$
$$C_5(\mathbf{x},\boldsymbol{\xi}) = 1 - \frac{P}{\kappa_4} \text{ with } P = \frac{4.013(x_3+\xi_3)(x_4+\xi_4)^3\sqrt{\kappa_5\kappa_6}}{6\,(\kappa_3)^2}\left(1 - \frac{x_3+\xi_3}{4\kappa_3}\sqrt{\frac{\kappa_5}{\kappa_6}}\right),$$

where $\kappa_1 = 6.74135\times 10^{-5}$, $\kappa_2 = 2.93585\times 10^{-6}$, $\kappa_3 = 3.556\times 10^2$, $\kappa_4 = 2.6688\times 10^4$, $\kappa_5 = 2.0685\times 10^5$ and $\kappa_6 = 8.274\times 10^4$.

- Uncertainties: $\xi_1 \sim \mathcal{U}(-0.1693, 0.1693)$, $\xi_2 \sim \mathcal{U}(-0.1693, 0.1693)$, $\xi_3 \sim \mathcal{U}(-0.0107, 0.0107)$, $\xi_4 \sim \mathcal{U}(-0.0107, 0.0107)$.
- Solution in [194]: $x^* = [5.9188, 181.2849, 210.6114, 6.2253]$ with $\mathbb{E}[C_0(\mathbf{x}^*,\boldsymbol{\xi})] = 2.4948$ and $\forall j \in [1,5]$, $\mathbb{P}(C_j(\mathbf{x}^*,\boldsymbol{\xi}) \leq 0) = 1.0$ (estimated from 10^6 samples).

Vehicle Side Impact problem [194]

- Dimension: $n = 7$ and $m = 10$.
- Original lower bounds: $\mathbf{b}_\ell = (0.5, 0.45, 0.5, 0.5, 0.875, 0.4, 0.4)$
- Original upper bounds: $\mathbf{b}_u = (1.5, 1.35, 1.5, 1.5, 2.625, 1.2, 1.2)$
- Original $\mathbf{x}_0$: $(1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0)$
- Equations:

$$\begin{aligned}
C_0(\mathbf{x},\boldsymbol{\xi}) &= 1.98 + 4.9(x_1+\xi_1) + 6.67(x_2+\xi_2) + 6.98(x_3+\xi_3) + 4.01(x_4+\xi_4) \\
&\quad + 1.78(x_5+\xi_5) + 2.73(x_7+\xi_7), \\
C_1(\mathbf{x},\boldsymbol{\xi}) &= 1.16 - 0.3717(x_2+\xi_2)(x_4+\xi_4) - 0.00931(x_2+\xi_2)\xi_{10} - 0.484(x_3+\xi_3)\xi_9 \\
&\quad + 0.01343(x_6+\xi_6)\xi_{10} - 1, \\
C_2(\mathbf{x},\boldsymbol{\xi}) &= 0.261 - 0.0159(x_1+\xi_1)(x_2+\xi_2) - 0.188(x_1+\xi_1)\xi_8 - 0.019(x_2+\xi_2)(x_7+\xi_7) \\
&\quad + 0.0144(x_3+\xi_3)(x_5+\xi_5) + 0.00087570(x_5+\xi_5)\xi_{10} + 0.08045(x_6+\xi_6)\xi_9 \\
&\quad + 0.00139\xi_8\xi_{11} + 1.575\times 10^{-6}\xi_{10}\xi_{11} - 0.32,
\end{aligned}$$

$$
\begin{aligned}
C_3(\mathbf{x},\boldsymbol{\xi}) &= 0.2147 + 0.00817(x_5+\xi_5) - 0.131(x_1+\xi_1)\xi_8 - 0.0704(x_1+\xi_1)\xi_9 \\
&\quad + 0.03099(x_2+\xi_2)(x_6+\xi_6) - 0.018(x_2+\xi_2)(x_7+\xi_7) + 0.0208(x_3+\xi_3)\xi_8 \\
&\quad + 0.121(x_3+\xi_3)\xi_9 - 0.00364(x_5+\xi_5)(x_6+\xi_6) + 0.0007715(x_5+\xi_5)\xi_{10} \\
&\quad - 0.0005354(x_6+\xi_6)\xi_{10} + 0.00121\xi_8\xi_{11} + 0.00184\xi_9\xi_{10} \\
&\quad - 0.02(x_2+\xi_2)^2 - 0.32, \\
C_4(\mathbf{x},\boldsymbol{\xi}) &= 0.74 - 0.61(x_2+\xi_2) - 0.163(x_3+\xi_3)\xi_8 + 0.001232(x_3+\xi_3)\xi_{10} \\
&\quad - 0.166(x_7+\xi_7)\xi_9 + 0.227(x_2+\xi_2)^2 - 0.32, \\
C_5(\mathbf{x},\boldsymbol{\xi}) &= 28.98 + 3.818(x_3+\xi_3) - 4.2(x_1+\xi_1)(x_2+\xi_2) + 0.0207(x_5+\xi_5)\xi_{10} \\
&\quad + 6.63(x_6+\xi_6)\xi_9 - 7.77(x_7+\xi_7)\xi_8 + 0.32\xi_9\xi_{10} - 32, \\
C_6(\mathbf{x},\boldsymbol{\xi}) &= 33.86 + 2.95(x_3+\xi_3) + 0.1792\xi_{10} - 5.057(x_1+\xi_1)(x_2+\xi_2) - 11(x_2+\xi_2)\xi_8 \\
&\quad - 0.0215(x_5+\xi_5)\xi_{10} - 9.98(x_7+\xi_7)\xi_8 + 22\xi_8\xi_9 - 32, \\
C_7(\mathbf{x},\boldsymbol{\xi}) &= 46.36 - 9.9(x_2+\xi_2) - 12.9(x_1+\xi_1)\xi_8 \\
&\quad + 0.1107(x_3+\xi_3)\xi_{10} - 32, \\
C_8(\mathbf{x},\boldsymbol{\xi}) &= 4.72 - 0.54(x_4+\xi_4) - 0.19(x_2+\xi_2)(x_3+\xi_3) - 0.0122(x_4+\xi_4)\xi_{10} \\
&\quad + 0.009325(x_6+\xi_6)\xi_{10} + 0.000191\xi_{11}^2 - 4, \\
C_9(\mathbf{x},\boldsymbol{\xi}) &= 10.58 - 0.674(x_1+\xi_1)(x_2+\xi_2) - 1.95(x_2+\xi_2)\xi_8 + 0.028(x_6+\xi_6)\xi_{10} \\
&\quad + 0.02054(x_3+\xi_3)\xi_{10} - 0.0198(x_4+\xi_4)\xi_{10} - 9.9, \\
C_{10}(\mathbf{x},\boldsymbol{\xi}) &= 16.45 - 0.489(x_3+\xi_3)(x_7+\xi_7) - 0.843(x_5+\xi_5)(x_6+\xi_6) + 0.0432\xi_9\xi_{10} \\
&\quad - 0.0556\xi_9\xi_{11} - 0.000786\xi_{11}^2 - 15.69.
\end{aligned}
$$

- Uncertainties: $\forall i \in \{1,2,3,4,6,7\}$, $\xi_i \sim \mathcal{N}(0, 0.03)$, $\xi_5 \sim \mathcal{N}(0, 0.05)$, $\xi_8 \sim \mathcal{N}(0.345, 0.006)$, $\xi_9 \sim \mathcal{N}(0.345, 0.006)$, $\xi_{10} \sim \mathcal{N}(0, 10)$ and $\xi_{11} \sim \mathcal{N}(0, 10)$.
- Solution in [194]: $x^* = (0.7872, 1.35, 0.6887, 1.5, 1.0706, 1.2, 0.7284)$ with $\mathbb{E}[C_0(\mathbf{x}^*, \boldsymbol{\xi})] = 29.5585$ and $\forall j \in [1, 10]$, $\mathbb{P}(C_j(\mathbf{x}^*, \boldsymbol{\xi}) \leq 0) \geq 0.9982$ (estimated from 10^6 samples).

Speed Reducer problem [48]

- Dimension: $n = 7$ and $m = 11$.
- Original lower bounds: $\mathbf{b}_\ell = (2.6, 0.7, 17, 7.3, 7.3, 2.9, 5.0)$
- Original upper bounds: $\mathbf{b}_u = (3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5)$
- Original $\mathbf{x}_0$: $(3.5, 0.7, 17, 7.3, 7.72, 3.35, 5.29)$

- Equations:

$$
\begin{aligned}
C_0(\mathbf{x},\boldsymbol{\xi}) &= 0.7854\ (x_1+\xi_1)\ (x_2+\xi_2)^2(3.3333\ (x_3+\xi_3)^2+14.9334\ (x_3+\xi_3)-43.0934)\\
&\quad -1.508\ (x_1+\xi_1)(\ (x_6+\xi_6)^2+\ (x_7+\xi_7)^2)+7.477(\ (x_6+\xi_6)^3+\ (x_7+\xi_7)^3)\\
&\quad +0.7854(\ (x_4+\xi_4)\ (x_6+\xi_6)^2+\ (x_5+\xi_5)\ (x_7+\xi_7)^2),\\
C_1(\mathbf{x},\boldsymbol{\xi}) &= \frac{27}{(x_1+\xi_1)\ (x_2+\xi_2)^2\ (x_3+\xi_3)}-1,\\
C_2(\mathbf{x},\boldsymbol{\xi}) &= \frac{397.5}{(x_1+\xi_1)\ (x_2+\xi_2)^2\ (x_3+\xi_3)^2}-1,\\
C_3(\mathbf{x},\boldsymbol{\xi}) &= \frac{1.93\ (x_4+\xi_4)^3}{(x_2+\xi_2)\ (x_3+\xi_3)\ (x_6+\xi_6)^4}-1,\\
C_4(\mathbf{x},\boldsymbol{\xi}) &= \frac{1.93\ (x_5+\xi_5)^3}{(x_2+\xi_2)\ (x_3+\xi_3)\ (x_7+\xi_7)^4}-1,\\
C_5(\mathbf{x},\boldsymbol{\xi}) &= \frac{\sqrt{\left(\frac{745\ (x_5+\xi_5)}{(x_2+\xi_2)\ (x_3+\xi_3)}\right)^2+16.9\times 10^6}}{0.1\ (x_6+\xi_6)^3}-1100,\\
C_6(\mathbf{x},\boldsymbol{\xi}) &= \frac{\sqrt{\left(\frac{745\ (x_5+\xi_5)}{(x_2+\xi_2)\ (x_3+\xi_3)}\right)^2+157.5\times 10^6}}{0.1\ (x_7+\xi_7)^3}-850,\\
C_7(\mathbf{x},\boldsymbol{\xi}) &= \ (x_2+\xi_2)\ (x_3+\xi_3)-40,\\
C_8(\mathbf{x},\boldsymbol{\xi}) &= 5-\frac{(x_1+\xi_1)}{(x_2+\xi_2)}\\
C_9(\mathbf{x},\boldsymbol{\xi}) &= \frac{(x_1+\xi_1)}{(x_2+\xi_2)}-12,\\
C_{10}(\mathbf{x},\boldsymbol{\xi}) &= \frac{1.5\ (x_6+\xi_6)+1.9}{(x_4+\xi_4)}-1,\\
C_{11}(\mathbf{x},\boldsymbol{\xi}) &= \frac{1.1\ (x_7+\xi_7)+1.9}{(x_5+\xi_5)}-1.
\end{aligned}
$$

- Uncertainties: $\forall i \in [1,7],\ \xi_i \sim \mathcal{N}(0, 0.005)$.
- Solution in [194]: $x^* = \ (3.5765, 0.7, 17.0, 7.3, 7.7541, 3.3652, 5.3017)$ with $\mathbb{E}[C_0(\mathbf{x}^*,\boldsymbol{\xi})] = 3038.72$ and $\forall j \in [1,11],\ \mathbb{P}(C_j(\mathbf{x}^*,\boldsymbol{\xi}) \leq 0) \geq 0.9976$ (estimated from 10^6 samples).

6.10.3 Appendix C. Detailed numerical results

This section details the numerical results of Section 6.8.3 and Section 6.8.4. In these sections, only the average result over the 100 runs are presented. In this section, boxplots are used to describe the result of all the 100 runs. Each run is represented by a cross, the orange line is the mediane and the

bounds of the box are the first and third quartiles. Finally, the circled crosses are the outliers. Here are the results for Section 6.8.3.

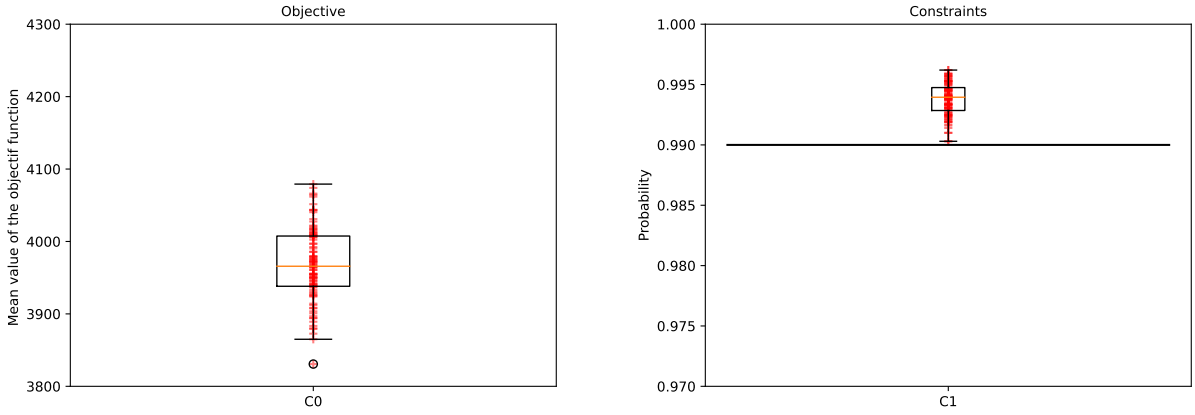

Figure 6.1 Detail result for Steel Column Design problem with classical Gaussian gradient approximation

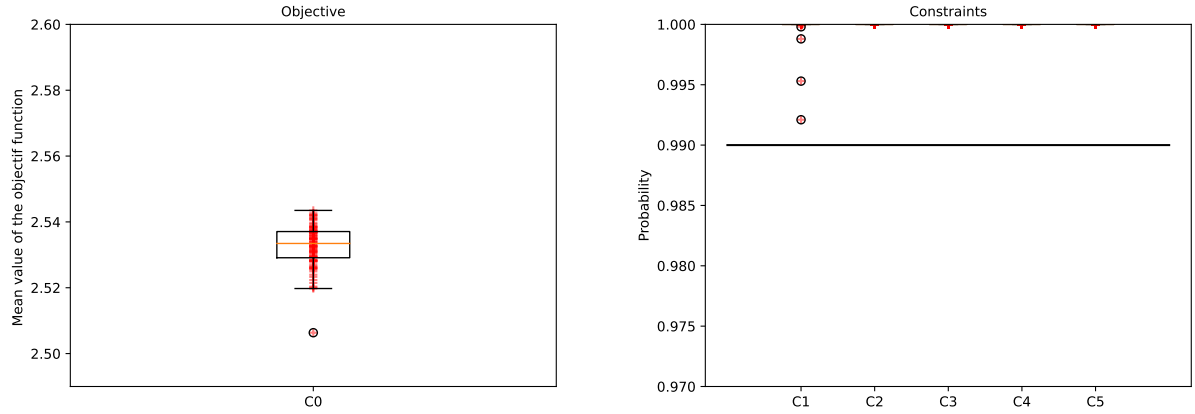

Figure 6.2 Detail result for Welded Beam Design problem with classical Gaussian gradient approximation

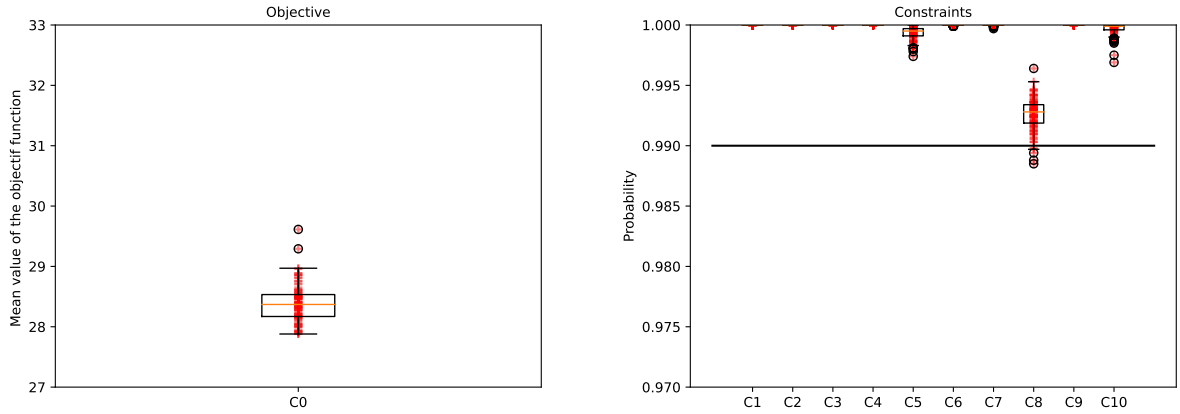

Figure 6.3 Detail result for Vehicle Side Impact problem with classical Gaussian gradient approximation

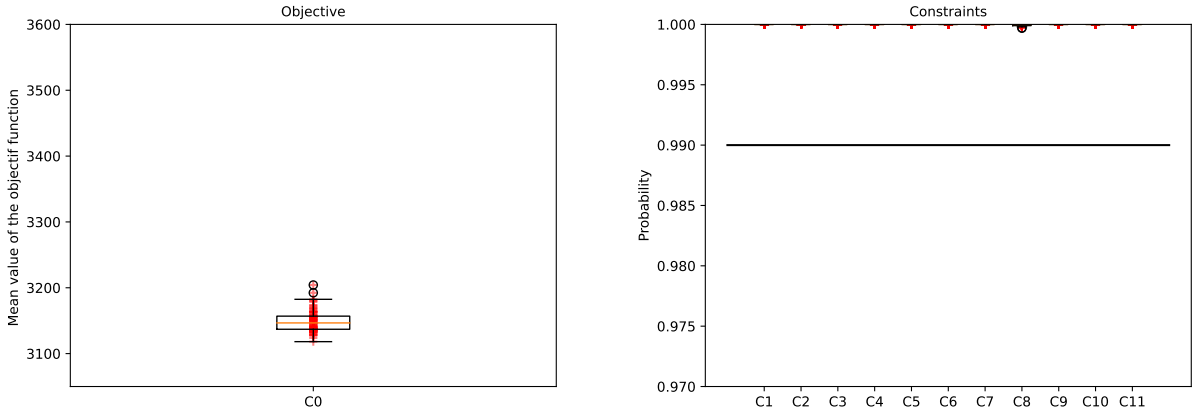

Figure 6.4 Detail result for Speed Reducer Design problem with classical Gaussian gradient approximation

Here are the results for Section 6.8.4.

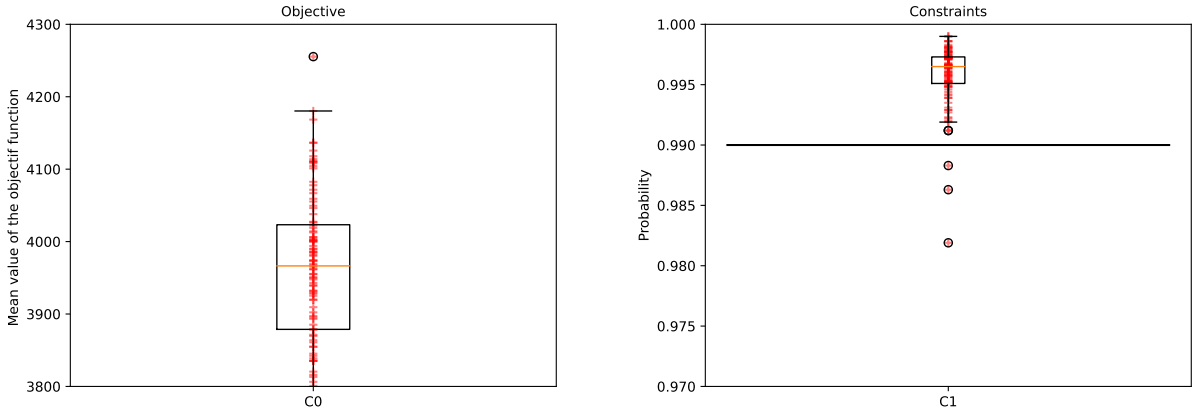

Figure 6.5 Detail result for Steel Column Design problem with truncated Gaussian gradient approximation

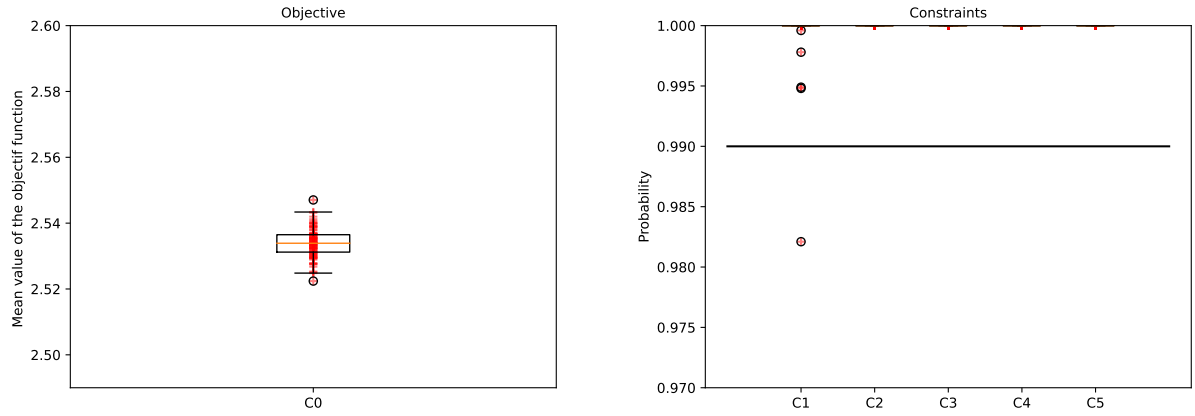

Figure 6.6 Detail result for Welded Beam Design problem with truncated Gaussian gradient approximation

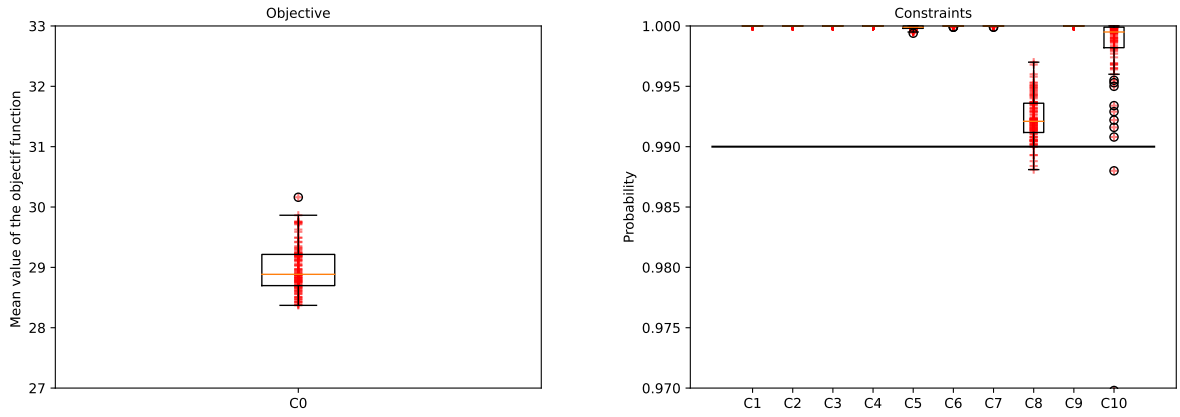

Figure 6.7 Detail result for Vehicle Side Impact problem with truncated Gaussian gradient approximation

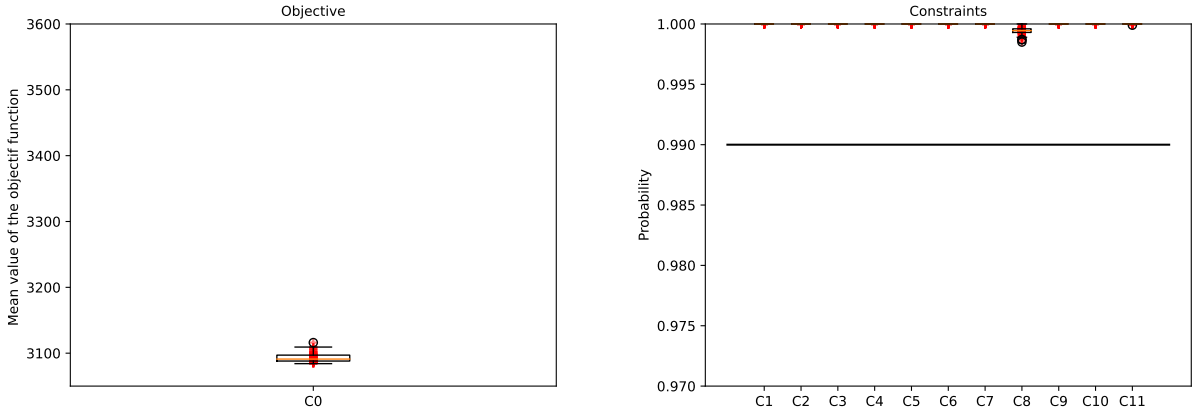

Figure 6.8 Detail result for Speed Reducer Design problem with truncated Gaussian gradient approximation

CHAPTER 7 GENERAL DISCUSSION

In this chapter, Section 7.1 summarizes the three different contributions of this thesis. Section 7.2 underlines their limitations, especially in the view of the research objectives stated in the introduction.

7.1 Summary of contributions

The contributions of this thesis focus on issues arising from the specific context of blackbox optimization: absence of gradients, multimodality, stochasticity, and risk aversion. To deal with these different issues, several approaches have been developed based on the singular concept of Gaussian exploration.

In the first contribution, a direct approach to Gaussian exploration is formalized by the cross-entropy method. Originally, this method is a metaheuristic and consists in updating the mean and standard deviation of the underlying normal distribution by taking the best points in terms of function values. In our work, the cross-entropy is adapted to be a search step of the mesh adaptive direct search algorithm, which aims at escaping local minima. This allows to guarantee the convergence properties of the resulting algorithm. The cross-entropy search step has two advantages compared to the other exploration-type search steps previously embedded in the NOMAD software. First, it is primarily designed to be applied to constrained blackbox problems. Therefore, the updating of the mean and standard deviation into the distribution depends not only on the objective function values, but also on the constraint values via the progressive barrier approach. Second, it has an internal mechanism choosing the iteration at which the cross-entropy search step must be performed or not. This mechanism is based on the standard deviation of the few best points. Thus, it depends on the problem and not on an a priori chosen hyperparameter. The resulting algorithm is compared to several state-of-the-art metaheuristics on a specific global optimization benchmark and its competitiveness is shown. Finally, it is compared with the other search step implemented in the NOMAD software for the purpose of escaping local minima on three industrial test cases. Again, promising results are obtained.

In the second contribution, the Gaussian exploration is applied to solve unconstrained blackbox optimization problems. Here, the Gaussian exploration is used to estimate the gradient of a smooth approximation of the original blackbox. The smooth Gaussian approximation benefits from several appealing properties: it is infinitely differentiable, it convexifies the blackbox around its local minima and estimates of its gradient may be computed using only two, potentially noisy, outputs

of the blackbox. All of these properties depend on a critical hyperparameter called the smoothing parameter. Choosing a particular value of the smoothing parameter is difficult in practice. A sequential optimization algorithm is developed in which the value of the smoothing parameter is progressively reduced. For solving the smooth subproblem, a new algorithm, called ZO-Signum, is implemented. To avoid overly expensive sampling at each iteration to compute an accurate gradient estimate, the sign of the direction is taken from a momentum vector. The momentum vector is a moving average of all gradient estimates over the iterations. This algorithm also features a stopping criterion based on the momentum norm. The interest of this stopping criterion is twofold in the sequential process. First, it allows the transition from one subproblem to another. Second, it is proportional to the smoothing parameter and decreases progressively. So, more importance is given to the smooth approximations that are the closest to the original problem, i.e. those with the smallest smoothing parameter values. The remainder of this paper is largely devoted to the theoretical analysis of both algorithms. On the one hand, under Lipschitz continuity assumptions, the convergence rate in mean of the ZO-Signum algorithm to a stationary point of a smooth approximation of the original problem is derived. Under further assumptions on the differentiability and local convexity of the original blackbox, we show that a subsequence of iterates of the SSO algorithm converges to a stationary point of the blackbox. In addition, the worst-case complexity of this convergence is given in terms of blackbox evaluations. Finally, the SSO algorithm is compared with other state-of-the-art algorithms on a simulation of a thermal solar power plant and on the generation of blackbox adversarial images for a deep neural network. In both experiments, the SSO algorithm shows its competitiveness.

In the third contribution, a Gaussian exploration similar to the previous one is used to solve constrained blackbox optimization problems subject to mixed aleatory/epistemic uncertainty. In this work, a CVaR formulation of the problem is chosen. This formulation has three main advantages. First, it gives the possibility to choose a desired reliability level to handle the probability constraints. Second, it can deal with both types of uncertainty. When the blackbox is subject to epistemic uncertainty, choosing a reliability level close to 1 allows the uncertainty to be treated similarly to a worst-case approach. Third, solving a CVaR-constrained optimization problem can be reformulated into an equivalent expectation-constrained blackbox optimization problem by adding $m + 1$ design variables. Then, to obtain a more tractable problem, the original problem is replaced by a smooth approximation. However, instead of using a Gaussian smooth approximation as in the previous work, we use its truncated counterpart to satisfy the bound constraints of the problem. The truncated Gaussian smooth approximation is shown to inherit the same properties as the Gaussian, and its own gradient estimator is derived. Theoretically, we show that under mild assumptions on the blackbox, for a value of the reliability level sufficiently close to 1, a feasible solution of the smooth approximation is also feasible for the original CVaR-constrained problem. Moreover,

the difference between the optimal function values of the approximation and the original problem can be bounded by an arbitrarily small value depending on: the dimension, the reliability level, and the smoothing parameter. A four timescale algorithm is developed to solve the Lagrangian relaxation of the smooth CVaR-constrained problem approximation. On the fastest timescale, information about the gradient is aggregated as a moving average of the truncated gradient estimates. On the first intermediate timescale, the values of the $m + 1$ additional variables are updated. On the second intermediate and slowest timescales the design variables and the Lagrangian multipliers are updated, respectively. The convergence of the algorithm is based on Lyapunov theory, and we prove that the algorithm converges to a locally optimal solution of the smooth problem with probability 1. It is then deduced that the algorithm converges, with probability one, to a feasible point of the CVaR-constrained problem whose objective function value is arbitrarily close to that of a local solution. Finally, several experiments are conducted with three different objectives. The first objective is to provide guidelines on how to experimentally determine the value of the different hyperparameters. The main findings are that all hyperparameters are problem independent except two: the value of the initial stepsize along $\mathbf{x}$ and the value of the smoothing parameter on the design variables. Experimentally, the value of the smoothing parameter can be chosen to minimize the average variance of the gradient at $\mathbf{x}^0$ over all directions. On the other hand, the value of the step size is correlated with the norm of the gradient at $\mathbf{x}^0$. The second objective is to study the effect of using the truncated Gaussian gradient estimator instead of its Gaussian counterpart. It appears that the algorithm can achieve similar results in both cases, but with a higher number of blackbox evaluations when the truncated estimator is used. Finally, experiments show the ability of the algorithm to deal with both types of uncertainty: aleatory and epistemic.

7.2 Limitations

To assess the limitations of the current work, we refer to the objectives stated in Section 1.3. While the first and second research objectives seem to be generally met, several limitations appear from the perspective of the third and fourth objectives:

- The second and third contributions are based on stochastic gradient descents (or ascents) with zeroth-order gradient approximations. A major drawback of these methods is that their efficiency in terms of function evaluation depends on hyperparameter values that are difficult to set a priori, especially the initial stepsizes. Despite the work done to reduce the number of hyperparameters and to provide experimental rules for setting them, the number of test problems used for this purpose is too small to ensure the generalization of these rules. Moreover, due to their lack of specific mathematical structure, blackbox problems pose a variety of challenges to optimization algorithms. It is highly likely that no rule will ultimately be

generalized.

- The numerical comparisons of the different contributions are also limited. In the first paper, despite the efforts made in this direction, comparisons with model-based methods or other methods combining a heuristic and a local search are missing. In the second paper, there are only two test problems, including one that is not stochastic. This is too few to ensure the practical interest of the methods, although this gap has been partially filled in the addenda. Finally, in the last paper, no comparison with other state-of-the-art algorithms is made and all numerical experiments are performed on analytical test problems. Therefore, these experiments need to be strengthened if their results are to be truly meaningful.

- Finally, although all the algorithms developed in this thesis are relatively easy to implement, they are not publicly available. This is an obstacle for other researchers who want to reproduce the results, perform additional experiments or numerical comparisons, or simply use the algorithms in their research.

CHAPTER 8 CONCLUSION

Finally, this section gives perspectives on future research work.

8.1 Future Research

In light of the limitations listed in the previous section, the following improvements may represent interesting lines of future research. First, it is necessary to propose quickly a publicly available version of the code used in this thesis. This will allow other researchers to use it.

The second research perspective concerns test problems. Currently, there are few benchmarks devoted to stochastic optimization or risk-averse optimization. SOLAR is a good example of what a benchmark for blackbox stochastic optimization should be. Other benchmarks of this kind, with different application should be implemented. Furthermore, it would be interesting to create more basic benchmarks to quickly test the relevance of a new algorithm. In these benchmarks, the user could choose the degree of uncertainty and the type of uncertainties: aleatory, epistemic or mixed.

The final research perspective looks for methods that avoid the use of step size, as this is one of the most difficult hyperparameters to set. A first approach could be based on a combination of direct and indirect Gaussian exploration, similar to what is done in the second paper for the SOLAR application. The idea would be to better integrate the two phases of exploration and exploitation. Another more promising approach could be based on stochastic linesearch or stochastic trust region. In fact, in deterministic gradient descent methods, the algorithms perform a linear search at each iteration to avoid the use of stepsize [6]. In the literature review it was mentioned that these methods have been adapted to the stochastic derivative-free context [34, 145]. Unfortunately, both of these algorithms require that the function be sampled a large number of times at each iteration to obtain a sufficiently accurate estimate of the gradient or model. To limit the number of samples to be estimated per iteration, one idea might be to use these methods based on the gradient estimated by a momentum vector (i.e., a moving average over all iterations).

CHAPTER 9 RÉSUMÉ ÉTENDU

L'optimisation de boîte noire porte sur le développement et l'analyse d'algorithmes conçus pour des problèmes dont la fonction objectif et/ou les contraintes sont les valeurs de sortie d'une boîte noire. Une boîte noire en optimisation peut être n'importe quel processus qui, à une entrée donnée, renvoie une sortie. Par définition, la structure d'une boîte noire est généralement inconnue, inexploitable, voire même inexistante. Ces problèmes surviennent usuellement lorsque les valeurs de la fonction objectif et des contraintes proviennent de simulations informatiques ou d'expériences menées en laboratoire. Ces situations sont courantes en ingénierie et dans certaines branches de l'apprentissage automatique.

En raison de l'absence de structure des fonctions sous-jacentes aux problèmes, les approches traditionnelles les plus efficaces, fondées sur le gradient, ne peuvent être directement utilisées en optimisation de boîte noire. Des stratégies doivent également être instaurées afin d'éviter aux algorithmes de converger prématurément vers une solution locale. En outre, il est fréquent que les valeurs retournées par la boîte noire soient soumises à des incertitudes. Les causes à l'origine de celles-ci sont variées et peuvent être dues par exemple à la prise en compte de paramètres aléatoires, à des imprécisions sur des mesures ou encore à l'estimation de certaines quantités par la méthode de Monte-Carlo. Par conséquent, les algorithmes développés doivent avoir la capacité de gérer les fluctuations induites par les incertitudes sur la fonction objectif et les contraintes, notamment en diminuant le risque que ces dernières prennent des valeurs aberrantes. Ces différents enjeux constituent les problématiques majeures de cette thèse : l'absence de gradient, la multimodalité, la stochasticité et l'aversion au risque.

Pour aborder ces problématiques, ce projet doctoral comporte trois contributions, dont la conception est agencée autour d'une unique notion : l'exploration Gaussienne de l'espace. Cette exploration consiste à échantillonner des points à partir d'une moyenne et d'un écart-type donnés. Dans une approche directe, l'algorithme se déplace directement vers un point minimisant une certaine quantité d'intérêt dépendant de la fonction objectif et/ou des contraintes. Dans une approche indirecte, les points échantillonnés sont utilisés pour estimer le gradient d'une approximation lisse de la boîte noire. Ces deux approches ont pour avantage de ne pas dépendre de la dimension de la boîte noire et de ne se fonder que sur les valeurs des fonctions retournées par celle-ci. Elles s'adaptent donc parfaitement au contexte de l'optimisation de boîte noire. L'objectif de cette thèse est donc de développer des algorithmes autour de ces approches et d'étudier leurs propriétés de convergence ainsi que leurs efficacités en pratique.

Le premier projet de la thèse traite de la problématique de la multimodalité dans un cadre de boîte

noire déterministe. Pour résoudre ce type de problème, il existe de nombreuses méthodes. Parmi elles, nous nous sommes concentrés sur la méthode de l'entropie croisée (CE). Cette méthode consiste à générer des points dans l'espace de recherche à partir d'une loi normale multivariée. La boîte noire est ensuite évaluée à ces différents points et les meilleurs points sont retenus afin de mettre à jour la moyenne et l'écart type de la loi normale. Cette méthode est intéressante car elle permet, dans un premier temps, d'explorer l'espace en identifiant des zones prometteuses de l'espace de recherche et, dans un second temps, d'échapper à des minima locaux. Le principal désavantage de cette méthode est qu'elle ne bénéficie pas de propriétés de convergence vers un minimum local de la fonction.

Dans cette première contribution, notre idée a été d'intégrer la méthode d'entropie croisée en tant qu'étape de recherche pour l'algorithme de recherche directe par treillis adaptatif (MADS). Dans l'algorithme MADS, l'espace de recherche est discrétisé suivant un certain treillis. Chacune des itérations de l'algorithme peut ensuite être décomposée en deux étapes: une étape de recherche (optionnelle) et une étape de sonde qui permet de garantir la convergence de l'algorithme. Dans l'étape de recherche, n'importe quelle méthode permettant d'accélérer la convergence ou d'éviter la convergence prématurée de l'algorithme dans un minimum local peut être utilisée. Pour s'assurer de la convergence, il est néanmoins nécessaire que le nombre de points évalués pendant cette étape soit fini. Dans l'étape de sonde, la boîte noire est évaluée dans un ensemble fini de directions formant un ensemble générateur positif de l'espace. Afin d'intégrer la méthode de l'entropie croisée en tant qu'étape de recherche de l'algorithme MADS, nous avons effectué trois changements par rapport à l'algorithme original.

Premièrement, dans le but de gérer les contraintes de bornes, la loi normale multivariée est tronquée. Deuxièmement, les points ne sont pas seulement ordonnés à partir de la valeur de leur fonction objectif mais également en prenant compte de leur valeur de la fonction de violation des contraintes. Cela permet d'adapter la méthode d'entropie croisée à la gestion des contraintes générales d'inégalité. Enfin, la particularité de cette étape de recherche est qu'elle n'est pas effectuée à chaque itération de l'algorithme mais seulement lorsque la norme de l'écart-type entre les meilleurs points trouvés jusqu'alors est inférieure à un certain seuil, qui est mis à jour au fur et à mesure du processus d'optimisation. Cela permet de sélectionner les itérations où il semble pertinent de procéder à une étape de recherche et ainsi de limiter le nombre d'évaluations de la boîte noire qui peut être coûteuse en terme de temps de calcul. Finalement, l'algorithme résultant bénéficie à la fois des propriétés de convergence de l'algorithme MADS et de celles d'exploration de la méthode de l'entropie croisée.

Cette étape de recherche a été comparée avec d'autres méthodes de recherche pour l'exploration fondées notamment sur la recherche par voisinage ou par l'échantillonnage dans un hypercube

latin. En outre, elle a été comparée à d'autres métaheuristiques. Ces comparaisons ont été menées des problèmes analytiques multimodaux mais également sur des problèmes d'ingénierie. Dans les deux cas, l'algorithme CE-MADS présente des résultats intéressants en terme de performance et d'efficacité.

Le second projet de thèse aborde les problèmes d'optimisation stochastique de boîte noire sans contrainte. Dans ce projet, un algorithme séquentiel (SSO) est développé afin de résoudre une suite d'approximations lisses Gaussienne. Ces approximations Gaussiennes lisses sont obtenues en prennant l'espérance par rapport à un vecteur $\mathbf{u}$ Gaussian de la fonction objectif f pris en $\mathbf{x}+\beta\mathbf{u}$ où β est le paramètre de lissage. Ces approximations bénéficient de plusieurs propriétés intéressantes pour l'optimisation. Premièrement, elles sont indéfiniment différentiables car cette espérance peut être vu comme un produit de convolution entre la fonction f et la fonction de densité de probabilité Gaussienne. Deuxièmement, si β est pris suffisamment large, un phénomène de convexification de la fonction apparait autour de ses minima locaux. A l'inverse, lorsque β tend vers 0, l'approximation tend vers la fonction originale. Enfin, la propriété la plus intéressante est que le gradient de l'approximation peut être estimé à partir de deux évaluations de la fonction objectif, quelque soit la dimension. Cet estimateur est de pauvre qualité en pratique mais il existe des techniques fondées sur les moyennes mobiles pour parvenir à l'exploiter. Pour ce faire, le problème original est décomposé en suite de sous problème. Chaque sous problème consiste à minimiser la valeur d'une approximation Gaussienne paramétrée par β. Dans cette suite d'approximation lisse, la valeur du paramètre β tend vers 0.

Dans l'algorithme SSO, chaque sous problème est résolu grâce à une version sans dérivées de l'algorithme de descente du signe du gradient stochatique avec momentum, appelé ZO-Signum. Cette version sans dérivées diffère de l'algorithme originel Signum pour trois raisons. Premièrement, le gradient de l'approximation Gaussienne est estimé à partir d'évaluations de la fonction objectif seulement. Deuxièmement, la taille de pas utilisée pour agréger les estimations du gradient dans le vecteur de moment tend vers 0, alors que la taille de pas est constante dans l'algorithme originel. Cette décroissance de la taille de pas vers 0 est particulièrement importante théoriquement puisqu'elle assure la décroissance de la variance du vecteur moment. Enfin, l'algorithme ZO-Signum bénéficie d'un critère d'arrêt fondé sur la norme du vecteur moment. Cela permet de passer d'un sous problème à l'autre dans l'algorithme SSO. En effet, une fois que l'algorithme ZO-Signum atteint le critère d'arrêt, l'algorithme SSO procède à la mise à jour du paramètre de lissage β pour le nouveau sous problème et des valeurs initiales des deux tailles de pas utilisées dans l'algorithme ZO-Signum.

Les propriétés de convergence des deux algorithmes ont été étudiées. Lorsque la boîte noire est Lipschitz continue, le taux de convergence en moyenne de l'algorithme ZO-Signum vers un point

stationnaire d'une approximation lisse du problème original a été calculé. Ce résultat a été comparé avec d'autres taux de convergence obtenus dans la littérature lorsque la fonction f est supposée $\mathcal{C}^1$. Il s'avère que, du à la plus faible hypothèse sur la structure de la fonction, le taux de convergence de l'algorithme ZO-Signum a une plus grande dépendance dans la dimension du problème mais obtient la même dépendance en terme d'itérations. Si la boîte noire est supposée $\mathcal{C}^1$ et est localement convexe autour de ses minima locaux, alors nous avons démontré le taux de convergence d'une sous suite d'itérés de l'algorithme SSO vers un point stationnaire du problème. Ce résultat est plus fort que les autres résultats de la littérature car il n'est pas obtenu en moyenne mais de manière déterministe.

Finalement, des tests numériques ont été réalisés sur des problèmes analytiques de différentes tailles, sur une simulation de centrale solaire et pour la génération d'images adverses. L'algorithme SSO a été comparé à une version sans dérivées de l'algorithme ADAM, à une version spécifique pour l'optimisation stochastique de l'algorithme MADS et à l'algorithme CMA-ES. Les résultats démontrent l'efficacité de l'algorithme sur les différents problèmes tests utilisés en comparaison des autres algorithmes.

Le troisième projet de thèse traite des problèmes d'optimisation de boîte noire sous contraintes et soumis à des incertitudes aléatoires et épistémiques. La valeur conditionnelle au risque (CVaR) est utilisée pour gérer les incertitudes dans la fonction objectif et les contraintes. Cette formulation a l'avantage de pouvoir choisir le degré de fiabilité α et de traiter les deux types d'incertitudes avec une approche du pire cas lorsque α est pris suffisamment proche de 1.

Étant donné que la fonction objectif et les contraintes proviennent d'une boîte noire et que la formulation avec CVaR est non différentiable, une approximation Gaussienne du problème est réalisée. Néanmoins, contrairement au travail précédent, le problème original peut contenir des contraintes de borne. Or si ces contraintes ne sont pas relaxables, c'est à dire que les valeurs retournées à l'extérieur des bornes n'ont pas de sens pour l'optimisation, l'approximation Gaussienne classique est inexploitable. C'est pourquoi nous utilisons une approximation lisse par Gaussienne tronquée. Nous avons prouvé que cette approximation hérite de tous les résultats théoriques de l'approximation Gaussienne. En outre, son gradient peut être estimé à partir de la sortie de la boîte noire et de points localisés à l'intérieur des bornes du problème. Finalement, nous avons prouvé également que pour une valeur α suffisamment proche de 1, la solution du problème approché est une solution réalisable pour le problème original dont la valeur de la fonction objectif peut être choisie arbitrairement proche de la valeur d'une solution du problème original.

Finalement, le problème approché est reformulé comme un problème sous contraintes de borne uniquement en utilisant la relaxation Lagrangienne. Un algorithme d'approximation stochastique à quatre échelles de temps (RAMSA) est développé pour résoudre ce problème. La première échelle

de temps sert à agréger les gradient stochastiques sous forme d'un vecteur de moment. La seconde échelle de temps estime la valeur conditionelle au risque de la fonction objectif et des contraintes. La troisième met à jour les variables de conception tandis que la dernière met à jour les multiplicateurs de Lagrange. Nous avons prouvé que l'algorithme RAMSA converge presque-sûrement vers un point réalisable du problème CVaR-contraint dont la valeur de la fonction objectif est arbitrairement proche de celle d'une solution locale. Pour chaque échelle de temps, la preuve de convergence est fondée sur une suite de différence de martingale. Ensuite, le résultat est obtenu en se ramenant à un système d'équations différentielles ordinaires ainsi que de résultats tirés du principe d'invariance de La Salle.

Enfin, des tests numériques ont été réalisés sur quatre problèmes analytiques de la littérature dont un étant soumis aux deux types d'incertitudes (les autres n'étant soumis qu'aux incertitudes aléatoires). Ces tests avaient trois buts différents. Premièrement, ils devaient servir à établir une stratégie permettant de déterminer la valeur des hyperparamètres en pratique. Nous avons pu établir que l'algorithme dépendait principalement du choix de la valeur du paramètre de lissage β et du choix de la taille de pas initial servant à la mise à jour des variables de décision. Empiriquement, il semble que la valeur de β doit être choisie de manière à minimiser la variance de l'estimateur du gradient. En outre, la taille du pas initial semble être corrélée avec la norme du gradient au point de départ. Deuxièmement, nous avons pu comparer l'estimateur Gaussien avec celui tronqué, il est apparu qu'utiliser le second rendait l'algorithme moins efficace en terme d'évaluations de la boîte noire. Finalement, ces tests nous ont permis de démontrer l'efficacité de l'algorithme pour des problèmes sujet à des incertitudes aléatoires et epistémiques. Dans nos tests numériques, les incertitudes épistémiques étaient modélisées par des points ou par des intervalles. Paradoxalement, l'algorithme performe mieux pour les incertitudes épistémiques modélisées sous forme d'intervalles.

En conclusion, cette thèse est fondée sur trois contributions principales abordant plusieurs problématiques de l'optimisation de boîte noire. Ces trois contributions ont fait l'objet d'articles de revue, les deux premièr ont été publiés dans les journaux Operations Research Forum et AIMS mathematics respectivement. Le dernier est en cours de revue dans la revue Computational Optimization and Applications. Cela prouve la qualité des contributions présentées.

Cependant, ces contributions ont également leurs limites. Le code permettant de reproduire les résultats, bien que très détaillé dans chacune des contributions, n'est pas encore disponible publiquement. Par ailleurs, les deux derniers algorithmes sont particulièrement dépendants vis à vis de certains hyperparamètres, notamment de la taille de pas initiale pour la mise à jour des variables de décision. Il pourrait être intéressant pour surmonter cette difficulté d'utiliser une recherche linéaire permettant de fixer la taille de pas à chaque itération. L'idéal serait d'utiliser le vecteur de moment pour obtenir la direction de la recherche. Néanmoins, cela soulève le problème de la mise

à jour de ce vecteur lorsqu'une recherche linéaire est utilisée et devra faire l'objet de travaux plus approfondis.

REFERENCES

[1] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review", in Elsevier, 2018, pp. 185–231. DOI: 10.1016/b978-0-12-813314-9.00010-4. [Online]. Available: http://dx.doi.org/10.1016/b978-0-12-813314-9.00010-4.

[2] S. Alarie, C. Audet, A. E. Gheribi, M. Kokkolaras, and S. Le Digabel, "Two decades of blackbox optimization applications", en, *EURO Journal on Computational Optimization*, vol. 9, p. 100 011, 2021. DOI: 10.1016/j.ejco.2021.100011. [Online]. Available: http://dx.doi.org/10.1016/j.ejco.2021.100011.

[3] R. E. Amri, R. Le Riche, C. Helbert, C. Blanchet-Scalliet, and S. Da Veiga, "A sampling criterion for constrained bayesian optimization with uncertainties", *arXiv preprint arXiv:2103.05706*, 2021.

[4] M. de Angelis, E. Patelli, and M. Beer, "Advanced line sampling for efficient robust reliability analysis", en, *Structural Safety*, vol. 52, pp. 170–182, Jan. 2015. DOI: 10.1016/j.strusafe.2014.10.002. [Online]. Available: http://dx.doi.org/10.1016/j.strusafe.2014.10.002.

[5] Y. Aoues and A. Chateauneuf, "Benchmark study of numerical methods for reliability-based design optimization", *Structural and Multidisciplinary Optimization*, vol. 41, pp. 277–294, 2010. DOI: 10.1007/s00158-009-0412-2. [Online]. Available: https://doi.org/10.1007/s00158-009-0412-2.

[6] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives", *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[7] D. V. Arnold and N. Hansen, "A (1+1)-CMA-ES for constrained optimisation", in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ACM, 2012. DOI: 10.1145/2330163.2330207. [Online]. Available: https://doi.org/10.1145%2F2330163.2330207.

[8] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk", in *Risk Management*, Cambridge University Press, 2002, pp. 145–175. DOI: 10.1017/cbo9780511615337.007. [Online]. Available: https://doi.org/10.1017%2Fcbo9780511615337.007.

[9] S. Au, J. Ching, and J. Beck, "Application of subset simulation methods to reliability benchmark problems", *Structural Safety*, vol. 29, no. 3, pp. 183–193, 2007. DOI: 10.1016/j.strusafe.2006.07.008. [Online]. Available: https://doi.org/10.1016%2Fj.strusafe.2006.07.008.

[10] C. Audet, V. Béchard, and S. Le Digabel, "Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search", *Journal of Global Optimization*, vol. 41, no. 2, pp. 299–318, 2008. DOI: 10.1007/s10898-007-9234-1. [Online]. Available: http://dx.doi.org/doi:10.1007/s10898-007-9234-1.

[11] 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. DOI: 10.1137/070692662. [Online]. Available: http://dx.doi.org/10.1137/070692662.

[12] C. Audet and J. E. Dennis, Jr., "Mesh Adaptive Direct Search Algorithms for Constrained Optimization", *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006. DOI: 10.1137/040603371. [Online]. Available: http://dx.doi.org/doi:10.1137/040603371.

[13] C. Audet, J. E. Dennis, Jr., and S. Le Digabel, "Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm", *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1150–1170, 2008. DOI: 10.1137/070707518. [Online]. Available: http://dx.doi.org/10.1137/070707518.

[14] C. Audet and W. Hare, *Derivative-Free and Blackbox Optimization* (Springer Series in Operations Research and Financial Engineering). Cham, Switzerland: Springer International Publishing, 2017. DOI: 10.1007/978-3-319-68913-5. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-68913-5.

[15] C. Audet, A. Ihaddadene, S. Le Digabel, and C. Tribes, "Robust optimization of noisy blackbox problems using the Mesh Adaptive Direct Search algorithm", *Optimization Letters*, vol. 12, no. 4, pp. 675–689, 2018. DOI: 10.1007/s11590-017-1226-6. [Online]. Available: https://doi.org/10.1007/s11590-017-1226-6.

[16] C. Audet, M. Kokkolaras, S. Le Digabel, and B. Talgorn, "Order-based error for managing ensembles of surrogates in mesh adaptive direct search", *Journal of Global Optimization*, vol. 70, no. 3, pp. 645–675, 2018. DOI: 10.1007/s10898-017-0574-1. [Online]. Available: http://rdcu.be/wGt6.

[17] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes, "Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm", *ACM Transactions on Mathematical Software*, vol. 48, no. 3, 35:1–35:22, 2022. DOI: 10.1145/3544489. [Online]. Available: https://dx.doi.org/10.1145/3544489.

[18] C. Audet, S. Le Digabel, and C. Tribes, "Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization", *Optimization and Engineering*, vol. 17, no. 2, pp. 333–358, 2016. DOI: 10.1007/s11081-015-9283-0. [Online]. Available: http://dx.doi.org/10.1007/s11081-015-9283-0.

[19] C. Audet, S. Le Digabel, and C. Tribes, "The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables", *SIAM Journal on Optimization*, vol. 29, no. 2, pp. 1164–1189, 2019. DOI: 10.1137/18M1175872. [Online]. Available: https://doi.org/10.1137/18M1175872.

[20] C. Audet and C. Tribes, "Mesh-based Nelder-Mead algorithm for inequality constrained optimization", *Computational Optimization and Applications*, vol. 71, no. 2, pp. 331–352, 2018. DOI: 10.1007/s10589-018-0016-0. [Online]. Available: https://link.springer.com/article/10.1007/s10589-018-0016-0.

[21] C. Audet, J. Bigeon, and R. Couderc, "Combining cross-entropy and MADS methods for inequality constrained global optimization", *Operations Research Forum*, vol. 2, no. 3, 2021. DOI: 10.1007/s43069-021-00075-y. [Online]. Available: https://doi.org/10.1007%2Fs43069-021-00075-y.

[22] C. Audet, J. Bigeon, R. Couderc, and M. Kokkolaras, "Sequential stochastic blackbox optimization with zeroth-order gradient estimators", *AIMS Mathematics*, vol. 8, no. 11, pp. 25 922–25 956, 2023, ISSN: 2473-6988. DOI: 10.3934/math.20231321. [Online]. Available: https://www.aimspress.com/article/doi/10.3934/math.20231321.

[23] C. Audet, K. J. Dzahini, M. Kokkolaras, and S. Le Digabel, "Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates", *Computational Optimization and Applications*, vol. 79, no. 1, pp. 1–34, Mar. 2021. DOI: 10.1007/s10589-020-00249-0. [Online]. Available: https://doi.org/10.1007/s10589-020-00249-0.

[24] F. Augustin and Y. M. Marzouk, "Nowpac: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints", *arXiv preprint arXiv:1403.1931*, 2014.

[25] K. Balasubramanian and S. Ghadimi, "Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points", *Foundations of Computational Mathematics*, vol. 22, no. 1, pp. 35–76, Mar. 2021. DOI: 10.1007/s10208-021-09499-8. [Online]. Available: https://doi.org/10.1007/s10208-021-09499-8.

[26] A. S. Bandeira, K. Scheinberg, and L. N. Vicente, "Convergence of trust-region methods based on probabilistic models", *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1238–1264, 2014. DOI: 10.1137/130915984. [Online]. Available: https://doi.org/10.1137%2F130915984.

[27] M. Beer and E. Patelli, "Editorial: Engineering analysis with vague and imprecise information", *Structural Safety*, vol. 52, p. 143, 2015. DOI: 10.1016/j.strusafe.2014.11.001. [Online]. Available: https://doi.org/10.1016%2Fj.strusafe.2014.11.001.

[28] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton University Press, 2009. DOI: 10.1515/9781400831050. [Online]. Available: https://doi.org/10.1515%2F9781400831050.

[29] 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, 2021. DOI: 10.1007/s10208-021-09513-z. [Online]. Available: https://doi.org/10.1007%2Fs10208-021-09513-z.

[30] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. Wiley, 1994. DOI: 10.1002/9780470316870. [Online]. Available: https://doi.org/10.1002%2F9780470316870.

[31] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "Signsgd: Compressed optimisation for non-convex problems", in *International Conference on Machine Learning*, PMLR, 2018, pp. 560–569.

[32] D. Bertsimas, O. Nohadani, and K. M. Teo, "Robust optimization for unconstrained simulation-based problems", *Operations Research*, vol. 58, no. 1, pp. 161–178, 2010. DOI: 10.1287/opre.1090.0715. [Online]. Available: https://doi.org/10.1287%2Fopre.1090.0715.

[33] S. Bhatnagar, H. Prasad, and L. Prashanth, *Stochastic Recursive Algorithms for Optimization*. Springer London, 2013. DOI: 10.1007/978-1-4471-4285-0. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-4285-0.

[34] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg, "Convergence rate analysis of a stochastic trust-region method via supermartingales", *INFORMS Journal on Optimization*, vol. 1, no. 2, pp. 92–119, 2019. DOI: 10.1287/ijoo.2019.0016. [Online]. Available: https://doi.org/10.1287%2Fijoo.2019.0016.

[35] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python", *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020. DOI: 10.1109/access.2020.2990567. [Online]. Available: https://doi.org/10.1109%2Faccess.2020.2990567.

[36] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method", *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005. DOI: 10.1007/s10479-005-5724-z. [Online]. Available: https://doi.org/10.1007%2Fs10479-005-5724-z.

[37] A. Booker, E. Cramer, P. Frank, J. Gablonsky, and J. E. Dennis, "MoVars: Multidisciplinary optimization via adaptive response surfaces", in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, 2007. DOI: 10.2514/6.2007-1927. [Online]. Available: https://doi.org/10.2514%2F6.2007-1927.

[38] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*. Cambridge, UK : New York: Cambridge University Press ; Hindustan Book Agency, 2008.

[39] H. Cai, Y. Lou, D. McKenzie, and W. Yin, "A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization", in *International Conference on Machine Learning*, PMLR, 2021, pp. 1193–1203.

[40] H. Cai, D. McKenzie, W. Yin, and Z. Zhang, "A one-bit, comparison-based gradient estimator", *Applied and Computational Harmonic Analysis*, vol. 60, pp. 242–266, Sep. 2022. DOI: 10.1016/j.acha.2022.03.003. [Online]. Available: https://doi.org/10.1016/j.acha.2022.03.003.

[41] H. Cai, D. McKenzie, W. Yin, and Z. Zhang, "Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling", *SIAM Journal on Optimization*, vol. 32, no. 2, pp. 687–714, Apr. 2022. DOI: 10.1137/21m1392966. [Online]. Available: https://doi.org/10.1137/21m1392966.

[42] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks", in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2017. DOI: 10.1109/sp.2017.49. [Online]. Available: https://doi.org/10.1109/sp.2017.49.

[43] K.-H. Chang, "Stochastic nelder–mead simplex method – a new globally convergent direct search method for simulation optimization", *European Journal of Operational Research*, vol. 220, no. 3, pp. 684–694, Aug. 2012. DOI: 10.1016/j.ejor.2012.02.028. [Online]. Available: https://doi.org/10.1016/j.ejor.2012.02.028.

[44] A. Chaudhuri, B. Kramer, and K. E. Willcox, "Information reuse for importance sampling in reliability-based design optimization", *Reliability Engineering & System Safety*, vol. 201, p. 106 853, 2020. DOI: 10.1016/j.ress.2020.106853. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2020.106853.

[45] R. Chen, M. Menickelly, and K. Scheinberg, "Stochastic optimization using a trust-region method and random models", *Mathematical Programming*, vol. 169, no. 2, pp. 447–487, Apr. 2017. DOI: 10.1007/s10107-017-1141-8. [Online]. Available: https://doi.org/10.1007/s10107-017-1141-8.

[46] X. Chen and N. Wang, "Optimization of short-time gasoline blending scheduling problem with a DNA based hybrid genetic algorithm", *Chemical Engineering and Processing: Process Intensification*, vol. 49, no. 10, pp. 1076–1083, 2010. DOI: 10.1016/j.cep.2010.07.014. [Online]. Available: http://dx.doi.org/10.1016/j.cep.2010.07.014.

[47] X. Chen *et al.*, "Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[48] Z. Chen, H. Qiu, L. Gao, and P. Li, "An optimal shifting vector approach for efficient probabilistic design", *Structural and Multidisciplinary Optimization*, vol. 47, no. 6, pp. 905–920, 2013. DOI: 10.1007/s00158-012-0873-6. [Online]. Available: https://doi.org/10.1007%2Fs00158-012-0873-6.

[49] G. Cheng, L. Xu, and L. Jiang, "A sequential approximate programming strategy for reliability-based structural optimization", *Computers & Structures*, vol. 84, no. 21, pp. 1353–1367, 2006. DOI: 10.1016/j.compstruc.2006.03.006. [Online]. Available: https://doi.org/10.1016%2Fj.compstruc.2006.03.006.

[50] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria", *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6070–6120, 2017.

[51] L. Cizelj, B. Mavko, and H. Riesch-Oppermann, "Application of first and second order reliability methods in the safety assessment of cracked steam generator tubing", *Nuclear Engineering and Design*, vol. 147, no. 3, pp. 359–368, 1994. DOI: 10.1016/0029-

5493(94)90218-6. [Online]. Available: https://doi.org/10.1016%2F0029-5493%2894%2990218-6.

[52] F. Clarke, *Optimization and Nonsmooth Analysis*. New York: John Wiley & Sons, 1983, Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics. [Online]. Available: http://www.ec-securehost.com/SIAM/CL05.html.

[53] A. R. Conn and S. Le Digabel, "Use of quadratic models with mesh-adaptive direct search for constrained black box optimization", *Optimization Methods and Software*, vol. 28, no. 1, pp. 139–158, 2013. DOI: 10.1080/10556788.2011.623162. [Online]. Available: http://dx.doi.org/10.1080/10556788.2011.623162.

[54] A. R. Conn, K. Scheinberg, and L. Vicente, *Introduction to Derivative-Free Optimization* (MOS-SIAM Series on Optimization). Philadelphia: SIAM, 2009, ISBN: 978-0-898716-68-9. DOI: 10.1137/1.9780898718768. [Online]. Available: http://dx.doi.org/10.1137/1.9780898718768.

[55] F. E. Curtis and K. Scheinberg, "Adaptive stochastic optimization: A framework for analyzing stochastic optimization algorithms", *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 32–42, 2020. DOI: 10.1109/msp.2020.3003539. [Online]. Available: https://doi.org/10.1109%2Fmsp.2020.3003539.

[56] F. E. Curtis, K. Scheinberg, and R. Shi, "A stochastic trust region algorithm based on careful step normalization", *INFORMS Journal on Optimization*, vol. 1, no. 3, pp. 200–220, Jul. 2019. DOI: 10.1287/ijoo.2018.0010. [Online]. Available: https://doi.org/10.1287/ijoo.2018.0010.

[57] K. Deb, S. Gupta, D. Daum, J. Branke, A. Mall, and D. Padmanabhan, "Reliability-based optimization using evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1054–1074, 2009. DOI: 10.1109/tevc.2009.2014361. [Online]. Available: https://doi.org/10.1109%2Ftevc.2009.2014361.

[58] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems", *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010. DOI: 10.1287/opre.1090.0741. [Online]. Available: https://doi.org/10.1287%2Fopre.1090.0741.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009. DOI: 10.1109/cvpr.2009.5206848. [Online]. Available: https://doi.org/10.1109/cvpr.2009.5206848.

[60] M. Dorigo and G. D. Caro, "Ant colony optimization: A new meta-heuristic", in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, IEEE. DOI: 10.1109/cec.1999.782657. [Online]. Available: https://doi.org/10.1109%2Fcec.1999.782657.

[61] X. Du and W. Chen, "Sequential optimization and reliability assessment method for efficient probabilistic design", *Journal of Mechanical Design*, vol. 126, no. 2, pp. 225–233, 2004. DOI: 10.1115/1.1649968. [Online]. Available: https://doi.org/10.1115%2F1.1649968.

[62] D. Dubois, "Possibility theory, probability theory and multiple-valued logics: A clarification", in *Computational Intelligence. Theory and Applications*, Springer Berlin Heidelberg, 2001, pp. 228–228. DOI: 10.1007/3-540-45493-4_26. [Online]. Available: https://doi.org/10.1007%2F3-540-45493-4_26.

[63] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations", *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, DOI: 10.1109/tit.2015.2409256. [Online]. Available: https://doi.org/10.1109%2Ftit.2015.2409256.

[64] K. J. Dzahini, "Expected complexity analysis of stochastic direct-search", *Computational Optimization and Applications*, vol. 81, no. 1, pp. 179–200, 2021. DOI: 10.1007/s10589-021-00329-9. [Online]. Available: https://doi.org/10.1007%2Fs10589-021-00329-9.

[65] K. J. Dzahini, M. Kokkolaras, and S. Le Digabel, "Constrained stochastic blackbox optimization using a progressive barrier and probabilistic estimates", *Mathematical Programming*, vol. 198, no. 1, pp. 675–732, 2022. DOI: 10.1007/s10107-022-01787-7. [Online]. Available: https://doi.org/10.1007%2Fs10107-022-01787-7.

[66] M. Eldred, L. Swiler, and G. Tang, "Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation", *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1092–1113, 2011. DOI: 10.1016/j.ress.2010.11.010. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2010.11.010.

[67] E. Fermi and N. Metropolis, "Numerical solution of a minimum problem", Los Alamos National Laboratory, Los Alamos, USA, Los Alamos Unclassified Report LA–1492, 1952.

[68] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient", in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, USA: Society for Industrial and Applied Mathematics, 2005, pp. 385–394, ISBN: 0898715857. DOI: 10.5555/1070432.1070486.

[69] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function", *Mathematical Programming*, vol. Series A, 91, pp. 239–269, 2002. DOI: 10.1007/s101070100244. [Online]. Available: http://dx.doi.org/10.1007/s101070100244.

[70] A. V. Gasnikov, E. A. Krymova, A. A. Lagunovskaya, I. N. Usmanova, and F. A. Fedorenko, "Stochastic online optimization. single-point and multi-point non-linear multi-armed bandits. convex and strongly-convex case", *Automation and Remote Control*, vol. 78, no. 2, pp. 224–234, 2017. DOI: 10.1134/s0005117917020035. [Online]. Available: https://doi.org/10.1134%2Fs0005117917020035.

[71] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming", *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, Jan. 2013. DOI: 10.1137/120880811. [Online]. Available: https://doi.org/10.1137/120880811.

[72] S. Ghadimi, A. Ruszczyński, and M. Wang, "A single timescale stochastic approximation method for nested stochastic optimization", *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 960–979, Jan. 2020. DOI: 10.1137/18m1230542. [Online]. Available: https://doi.org/10.1137/18m1230542.

[73] A. Giunta, *Aircraft multidisciplinary optimization using design of experiments theory and response surface modeling methods*, 1997.

[74] F. Glover, "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986. DOI: 10.1016/0305-0548(86)90048-1. [Online]. Available: https://doi.org/10.1016%2F0305-0548%2886%2990048-1.

[75] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[76] N. Gould, D. Orban, and P. Toint, "CUTEr (and SifDec): A constrained and unconstrained testing environment, revisited", *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 373–394, 2003. DOI: 10.1145/962437.962439. [Online]. Available: http://dx.doi.org/10.1145/962437.962439.

[77] R. B. Gramacy *et al.*, "Modeling an augmented lagrangian for blackbox constrained optimization", *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016. DOI: 10.1080/00401706.2015.1014065. [Online]. Available: https://doi.org/10.1080%2F00401706.2015.1014065.

[78] N. Hansen, "The CMA Evolution Strategy: A Comparing Review", in *Towards a New Evolutionary Computation*, ser. Studies in Fuzziness and Soft Computing, J. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., vol. 192, Springer Berlin Heidelberg, 2006, pp. 75–102. DOI: 10.1007/3-540-32494-1_4. [Online]. Available: http://dx.doi.org/10.1007/3-540-32494-1_4.

[79] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)", *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003. DOI: 10.1162/106365603321828970. [Online]. Available: https://doi.org/10.1162%2F106365603321828970.

[80] A.-R. Hedar, *Global Optimization Test Problems*, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm, (last accessed on 2017-10-20). [Online]. Available: http://goo.gl/0vxil.

[81] M. Heinkenschloss, B. Kramer, T. Takhtaganov, and K. Willcox, "Conditional-value-at-risk estimation via reduced-order models", *SIAM/ASA Journal on Uncertainty Quantification*, vol. 6, no. 4, pp. 1395–1423, 2018. DOI: 10.1137/17m1160069. [Online]. Available: https://doi.org/10.1137%2F17m1160069.

[82] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*. Springer Berlin Heidelberg, 1981. DOI: 10.1007/978-3-642-48320-2. [Online]. Available: https://doi.org/10.1007%2F978-3-642-48320-2.

[83] J. Jahn, *Introduction to the Theory of Nonlinear Optimization*. Springer Berlin Heidelberg, 1994. DOI: 10.1007/978-3-662-02985-5. [Online]. Available: https://doi.org/10.1007%2F978-3-662-02985-5.

[84] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems", *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013. DOI: 10.1504/IJMMNO.2013.055204. [Online]. Available: http://dx.doi.org/10.1504/IJMMNO.2013.055204.

[85] M. Johnson, *Nonlinear optimization using the algorithm of Hooke and Jeeves*, Software available in the Netlib Repository at http://www.netlib.org/opt/hooke.c, 1994. [Online]. Available: http://www.netlib.org/opt/hooke.c.

[86] D. Jones, C. Perttunen, and B. Stuckman, "Lipschitzian optimization without the Lipschitz constant", *Journal of Optimization Theory and Application*, vol. 79, no. 1, pp. 157–181, 1993. DOI: 10.1007/BF00941892. [Online]. Available: http://dx.doi.org/10.1007/BF00941892.

[87] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes", in *International Conference on Machine Learning*, PMLR, 2019, pp. 3252–3261.

[88] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future", *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2020. DOI: 10.1007/s11042-020-10139-6. [Online]. Available: https://doi.org/10.1007%2Fs11042-020-10139-6.

[89] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia: IEEE Service Center, Piscataway, 1995, pp. 1942–1948.

[90] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of ICNN 1995 - International Conference on Neural Networks*, IEEE. DOI: 10.1109/icnn.1995.488968. [Online]. Available: https://doi.org/10.1109%2Ficnn.1995.488968.

[91] H. Khalil, *Nonlinear Systems*. Prentice hall Upper Saddle River: Pearson, 3rd edition, 2001.

[92] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function", in *Collected Papers*, Springer US, 1985, pp. 60–64. DOI: 10.1007/978-1-4613-8505-9_4. [Online]. Available: https://doi.org/10.1007/978-1-4613-8505-9_4.

[93] B. Kim, H. Cai, D. McKenzie, and W. Yin, "Curvature-aware derivative-free optimization", *arXiv preprint arXiv:2109.13391*, 2021.

[94] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[95] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing", *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[96] S. Kitayama, M. Arakawa, and K. Yamazaki, "Sequential approximate optimization using radial basis function network for engineering optimization", *Optimization and Engineering*, vol. 12, no. 4, pp. 535–557, 2011. DOI: 10.1007/s11081-010-9118-y. [Online]. Available: http://dx.doi.org/10.1007/s11081-010-9118-y.

[97] M. Kokkolaras, Z. P. Mourelatos, and P. Y. Papalambros, "Impact of uncertainty quantification on design: An engine optimisation case study", *International Journal of Reliability and Safety*, vol. 1, 2006. DOI: https://doi.org/10.1504/IJRS.2006.010786.

[98] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. DOI: 10.1145/3065386. [Online]. Available: https://doi.org/10.1145/3065386.

[99] D. P. Kroese, S. Porotsky, and R. Y. Rubinstein, "The cross-entropy method for continuous multi-extremal optimization", *Methodology and Computing in Applied Probability*, vol. 8, pp. 383–407, 2006.

[100] D. P. Kroese, S. Porotsky, and R. Y. Rubinstein, "The cross-entropy method for continuous multi-extremal optimization", *Methodology and Computing in Applied Probability*, vol. 8, no. 3, pp. 383–407, 2006. DOI: 10.1007/s11009-006-9753-0. [Online]. Available: https://doi.org/10.1007%2Fs11009-006-9753-0.

[101] S. Kullback and R. A. Leibler, "On information and sufficiency", *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. DOI: 10.1214/aoms/1177729694. [Online]. Available: https://doi.org/10.1214%2Faoms%2F1177729694.

[102] P. L'Ecuyer, M. Mandjes, and B. Tuffin, *Importance sampling in rare event simulation*, 2009. DOI: 10.1002/9780470745403.ch2. [Online]. Available: https://doi.org/10.1002%2F9780470745403.ch2.

[103] J. Lampinen and R. Storn, "Differential evolution", in *New Optimization Techniques in Engineering*, Springer Berlin Heidelberg, 2004, pp. 123–166. DOI: 10.1007/978-3-540-39930-8_6. [Online]. Available: https://doi.org/10.1007%2F978-3-540-39930-8_6.

[104] J. Larson and S. C. Billups, "Stochastic derivative-free optimization using a trust region framework", *Computational Optimization and applications*, vol. 64, pp. 619–645, 2016.

[105] J. Larson, M. Menickelly, and S. M. Wild, "Derivative-free optimization methods", *Acta Numerica*, vol. 28, pp. 287–404, 2019. DOI: 10.1017/s0962492919000060. [Online]. Available: https://doi.org/10.1017%2Fs0962492919000060.

[106] S. Le Digabel, "Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm", *ACM Transactions on Mathematical Software*, vol. 37, no. 4, 44:1–44:15, 2011. DOI: 10.1145/1916461.1916468. [Online]. Available: http://dx.doi.org/10.1145/1916461.1916468.

[107] S. Le Digabel and S. M. Wild, *A taxonomy of constraints in black-box simulation-based optimization*, en, Sep. 2023. DOI: 10.1007/s11081-023-09839-3. [Online]. Available: http://dx.doi.org/10.1007/s11081-023-09839-3.

[108] R. Lebrun and A. Dutfoy, "A generalization of the nataf transformation to distributions with elliptical copula", *Probabilistic Engineering Mechanics*, vol. 24, no. 2, pp. 172–178, 2009. DOI: 10.1016/j.probengmech.2008.05.001. [Online]. Available: https://doi.org/10.1016%2Fj.probengmech.2008.05.001.

[109] R. Lebrun and A. Dutfoy, "An innovating analysis of the nataf transformation from the copula viewpoint", *Probabilistic Engineering Mechanics*, vol. 24, no. 3, pp. 312–320, 2009. DOI: 10.1016/j.probengmech.2008.08.001. [Online]. Available: https://doi.org/10.1016%2Fj.probengmech.2008.08.001.

[110] M. Lemyre Garneau, "Modelling of a solar thermal power plant for benchmarking black-box optimization solvers", Available at https://publications.polymtl.ca/1996/, M.S. thesis, École Polytechnique de Montréal, 2015. [Online]. Available: https://publications.polymtl.ca/1996/.

[111] J. Li and D. Xiu, "Evaluation of failure probability via surrogate models", *Journal of Computational Physics*, vol. 229, no. 23, pp. 8966–8980, 2010. DOI: 10.1016/j.jcp.2010.08.022. [Online]. Available: https://doi.org/10.1016%2Fj.jcp.2010.08.022.

[112] W. Li, C. Li, L. Gao, and M. Xiao, "Risk-based design optimization under hybrid uncertainties", *Engineering with Computers*, vol. 38, no. 3, pp. 2037–2049, 2020. DOI: 10.1007/s00366-020-01196-4. [Online]. Available: https://doi.org/10.1007%2Fs00366-020-01196-4.

[113] W. Li, M. Xiao, A. Garg, and L. Gao, "A new approach to solve uncertain multidisciplinary design optimization based on conditional value at risk", *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 356–368, 2021. DOI: 10.1109/tase.2020.2999380. [Online]. Available: https://doi.org/10.1109%2Ftase.2020.2999380.

[114] J. Liang, Z. P. Mourelatos, and E. Nikolaidis, "A single-loop approach for system reliability-based design optimization", in *Volume 1: 32nd Design Automation Conference, Parts A and B*, ASMEDC, 2006. DOI: 10.1115/detc2006-99240. [Online]. Available: https://doi.org/10.1115%2Fdetc2006-99240.

[115] F. Lin, X. Fang, and Z. Gao, "Distributionally robust optimization: A review on theory and applications", *Numerical Algebra, Control & Optimization*, vol. 12, no. 1, p. 159, 2022. DOI: 10.3934/naco.2021057. [Online]. Available: https://doi.org/10.3934%2Fnaco.2021057.

[116] S. Liu, P. Y. Chen, X. Chen, and M. Hong, "Sign-sgd via zeroth-order oracle", in *International Conference on Learning Representations*, 2018.

[117] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications", *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, Sep. 2020. DOI: 10.1109/msp.2020.3003837. [Online]. Available: https://doi.org/10.1109/msp.2020.3003837.

[118] S. Liu, B. Kailkhura, P.-Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization", *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[119] Z.-G. Liu, Y. Liu, J. Dezert, and F. Cuzzolin, "Evidence combination based on credal belief redistribution for pattern classification", *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 4, pp. 618–631, 2020. DOI: 10.1109/tfuzz.2019.2911915. [Online]. Available: https://doi.org/10.1109%2Ftfuzz.2019.2911915.

[120] G. Liuzzi and S. Lucidi, "A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an l_∞ penalty function", *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 1–29, 2009. DOI: 10.1137/070711451. [Online]. Available: https://doi.org/10.1137%2F070711451.

[121] P. Lopez-Garcia, E. Onieva, E. Osaba, A. Masegosa, and A. Perallos, "GACE: A metaheuristic based in the hybridization of genetic algorithms and cross entropy methods for continuous optimization", *Expert Systems with Applications*, vol. 55, pp. 508–519, 2016. DOI: 10.1016/j.eswa.2016.02.034. [Online]. Available: https://doi.org/10.1016%2Fj.eswa.2016.02.034.

[122] L. Luksan and J. Vlcek, "Test problems for nonsmooth unconstrained and linearly constrained optimization", ICS AS CR, Tech. Rep. V-798, 2000. [Online]. Available: http://www.cs.cas.cz/ics/reports/v798-00.ps.

[123] A. Maggiar, A. Wächter, I. S. Dolinskaya, and J. Staum, "A derivative-free trust-region algorithm for the optimization of functions smoothed via gaussian convolution using adaptive multiple importance sampling", *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1478–

1507, Jan. 2018. DOI: 10.1137/15m1031679. [Online]. Available: https://doi.org/10.1137/15m1031679.

[124] L. Matott, A. Rabideau, and J. Craig, "Pump-and-treat optimization using analytic element method flow models", *Advances in Water Resources*, vol. 29, no. 5, pp. 760–775, 2006. DOI: 10.1016/j.advwatres.2005.07.009. [Online]. Available: http://dx.doi.org/10.1016/j.advwatres.2005.07.009.

[125] K. McKinnon, "Convergence of the Nelder-Mead simplex method to a nonstationary point", *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 148–158, 1998. DOI: 10.1137/S1052623496303482 [Online]. Available: https://dx.doi.org/10.1137/S1052623496303482.

[126] V. V. de Melo and G. Iacca, "A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization", *Expert Systems with Applications*, vol. 41, no. 16, pp. 7077–7094, 2014. DOI: 10.1016/j.eswa.2014.06.032. [Online]. Available: https://doi.org/10.1016%2Fj.eswa.2014.06.032.

[127] F. Meng, J. Sun, and M. Goh, "A smoothing sample average approximation method for stochastic optimization problems with CVaR risk measure", *Computational Optimization and Applications*, vol. 50, no. 2, pp. 379–401, 2010. DOI: 10.1007/s10589-010-9328-4. [Online]. Available: https://doi.org/10.1007%2Fs10589-010-9328-4.

[128] Z. Meng, Y. Pang, Y. Pu, and X. Wang, "New hybrid reliability-based topology optimization method combining fuzzy and probabilistic models for handling epistemic and aleatory uncertainties", *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112 886, 2020. DOI: 10.1016/j.cma.2020.112886. [Online]. Available: https://doi.org/10.1016%2Fj.cma.2020.112886.

[129] Z. Meng and H. Zhou, "New target performance approach for a super parametric convex model of non-probabilistic reliability-based design optimization", *Computer methods in applied mechanics and engineering*, vol. 339, pp. 644–662, 2018.

[130] F. Menhorn, F. Augustin, H.-J. Bungartz, and Y. M. Marzouk, "A trust-region method for derivative-free nonlinear constrained stochastic optimization", *arXiv preprint arXiv:1703.04156*, 2017.

[131] M. Menickelly and S. M. Wild, "Derivative-free robust optimization by outer approximations", *Mathematical Programming*, vol. 179, no. 1-2, pp. 157–193, 2018. DOI: 10.1007/s10107-018-1326-9. [Online]. Available: https://doi.org/10.1007%2Fs10107-018-1326-9.

[132] E. Mezura-Montes and C. Coello, "Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms", in *Proceedings of the 4th Mexican International Conference on Advances in Artificial Intelligence*, ser. MICAI'05, Berlin, Heidelberg: Springer-Verlag, 2005, pp. 652–662. DOI: 10.1007/11579427_66. [Online]. Available: http://dx.doi.org/10.1007/11579427_66.

[133] F. MiarNaeimi, G. Azizyan, and M. Rashki, "Multi-level cross entropy optimizer (MCEO): An evolutionary optimization algorithm for engineering problems", *Engineering with Computers*, vol. 34, no. 4, pp. 719–739, 2017. DOI: 10.1007/s00366-017-0569-z. [Online]. Available: https://doi.org/10.1007%2Fs00366-017-0569-z.

[134] N. Mladenović, J. Petrovic, V. Kovacevic-Vujcic, and M. Cangalovic, "Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search", *European Journal of Operational Research*, vol. 151, no. 2, pp. 389–399, 2003. DOI: 10.1016/S0377-2217(02)00833-0. [Online]. Available: http://dx.doi.org/10.1016/S0377-2217(02)00833-0.

[135] N. Mladenović and P. Hansen, "Variable neighborhood search", *Computers & operations research*, vol. 24, no. 11, pp. 1097–1100, 1997.

[136] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009. DOI: 10.1137/1.9780898717716. [Online]. Available: https://doi.org/10.1137%2F1.9780898717716.

[137] J. J. Moré, B. Garbow, and K. E. Hillstrom, "Testing unconstrained optimization software", *ACM Transactions on Mathematical Software*, vol. 7, no. 1, pp. 17–41, 1981. DOI: 10.1145/355934.355936. [Online]. Available: http://dx.doi.org/10.1145/355934.355936.

[138] J. J. Moré and S. M. Wild, "Benchmarking derivative-free optimization algorithms", *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 172–191, 2009. DOI: 10.1137/080724083. [Online]. Available: http://dx.doi.org/10.1137/080724083.

[139] S. Nannapaneni and S. Mahadevan, "Reliability analysis under epistemic uncertainty", *Reliability Engineering & System Safety*, vol. 155, pp. 9–20, 2016. DOI: 10.1016/j.ress.2016.06.005. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2016.06.005.

[140] J. Nelder and R. Mead, "A simplex method for function minimization", *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965. DOI: 10.1093/comjnl/7.4.308. [Online]. Available: http://comjnl.oxfordjournals.org/content/7/4/308.abstract.

[141] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions", *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, Nov. 2015. DOI: 10.1007/s10208-015-9296-2. [Online]. Available: https://doi.org/10.1007/s10208-015-9296-2.

[142] F. Nobile, R. Tempone, and C. G. Webster, "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data", *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2411–2442, 2008. DOI: 10.1137/070680540. [Online]. Available: https://doi.org/10.1137%2F070680540.

[143] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Series in Operations Research). New York: Springer, 1999. [Online]. Available: http://www.springer.com/mathematics/book/978-0-387-30303-1.

[144] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning", in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ACM, Apr. 2017. DOI: 10.1145/3052973.3053009. [Online]. Available: https://doi.org/10.1145/3052973.3053009.

[145] C. Paquette and K. Scheinberg, "A stochastic line search method with expected complexity analysis", *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 349–376, 2020. DOI: 10.1137/18m1216250. [Online]. Available: https://doi.org/10.1137%2F18m1216250.

[146] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems", in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2005, pp. 582–591. DOI: 10.1007/11539902_71. [Online]. Available: https://doi.org/10.1007%2F11539902_71.

[147] B. Peherstorfer, B. Kramer, and K. Willcox, "Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation", *SIAM/ASA Journal on Uncertainty Quantification*, vol. 6, no. 2, pp. 737–761, 2018. DOI: 10.1137/17m1122992. [Online]. Available: https://doi.org/10.1137%2F17m1122992.

[148] F. Pigache, F. Messine, and B. Nogarede, "Optimal Design of Piezoelectric Transformers: A Rational Approach Based on an Analytical Model and a Deterministic Global Optimization", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 54, no. 7, pp. 1293–1302, 2007. DOI: 10.1109/TUFFC.2007.390. [Online]. Available: http://dx.doi.org/10.1109/TUFFC.2007.390.

[149] M. J. D. Powell, "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation", in *Advances in Optimization and Numerical Analysis*, ser. Mathematics and Its Applications, S. Gomez and J.-P. Hennart, Eds., vol. 275, Springer, 1994, pp. 51–67. DOI: 10.1007/978-94-015-8330-5_4. [Online]. Available: http://dx.doi.org/10.1007/978-94-015-8330-5_4.

[150] M. J. D. Powell, "On fast trust region methods for quadratic models with linear constraints", *Mathematical Programming Computation*, vol. 7, no. 3, pp. 237–267, 2015. DOI: 10.1007/s12532-015-0084-4. [Online]. Available: https://doi.org/10.1007%2Fs12532-015-0084-4.

[151] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives", Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England, Tech. Rep. DAMTP 2009/NA06, 2009. [Online]. Available: http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

[152] L. A. Prashanth, "Policy gradients for CVaR-constrained MDPs", in *Lecture Notes in Computer Science*, Springer International Publishing, 2014, pp. 155–169. DOI: 10.1007/978-3-319-11662-4_12. [Online]. Available: https://doi.org/10.1007%2F978-3-319-11662-4_12.

[153] K. P. R. Storne, "Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global optimization*, vol. 11, no. 4, pp. 341–359, 1997. DOI: 10.1023/a:1008202821328. [Online]. Available: http://dx.doi.org/10.1023/A:1008202821328.

[154] H. Rahimian and S. Mehrotra, "Distributionally robust optimization: A review", *arXiv preprint arXiv:1908.05659*, 2019.

[155] E. Real *et al.*, "Large-scale evolution of image classifiers", in *International Conference on Machine Learning*, PMLR, 2017, pp. 2902–2911.

[156] H. Robbins and S. Monro, "A stochastic approximation method", *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951. DOI: 10.1214/aoms/1177729586. [Online]. Available: https://doi.org/10.1214%2Faoms%2F1177729586.

[157] R. Rocchetta, M. Broggi, and E. Patelli, "Do we have enough data? robust reliability via uncertainty quantification", *Applied Mathematical Modelling*, vol. 54, pp. 710–721, 2018. DOI: 10.1016/j.apm.2017.10.020. [Online]. Available: https://doi.org/10.1016%2Fj.apm.2017.10.020.

[158] R. Rocchetta and L. G. Crespo, "A scenario optimization approach to reliability-based and risk-based design: Soft-constrained modulation of failure probability bounds", *Reliability Engineering & System Safety*, vol. 216, p. 107 900, 2021. DOI: 10.1016/j.ress.2021.107900. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2021.107900.

[159] R. T. Rockafellar and J. O. Royset, "Random variables, monotone relations, and convex analysis", *Mathematical Programming*, vol. 148, no. 1-2, pp. 297–331, 2014. DOI: 10.1007/s10107-014-0801-1. [Online]. Available: https://doi.org/10.1007%2Fs10107-014-0801-1.

[160] R. T. Rockafellar and J. O. Royset, "Risk measures in engineering design under uncertainty", in *Proc. International Conf. on Applications of Statistics and Probability in Civil Engineering*, 2015.

[161] R. T. Rockafellar and J. O. Royset, "Engineering decisions under risk averseness", *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 1, no. 2, 2015. DOI: 10.1061/ajrua6.0000816. [Online]. Available: https://doi.org/10.1061%2Fajrua6.0000816.

[162] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk", *The Journal of Risk*, vol. 2, no. 3, pp. 21–41, 2000. DOI: 10.21314/jor.2000.038. [Online]. Available: https://doi.org/10.21314%2Fjor.2000.038.

[163] R. Rockafellar and J. Royset, "On buffered failure probability in design and optimization of structures", *Reliability Engineering & System Safety*, vol. 95, no. 5, pp. 499–510, 2010. DOI: 10.1016/j.ress.2010.01.001. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2010.01.001.

[164] J. Rodríguez, J. E. Renaud, and L. Watson, "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization", *Journal of Mechanical Design*, vol. 120, no. 1, pp. 58–66, 1998. DOI: 10.1115/1.2826677. [Online]. Available: http://dx.doi.org/10.1115/1.2826677.

[165] M. Rosenblatt, "Remarks on a multivariate transformation", *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 470–472, 1952. DOI: 10.1214/aoms/1177729394. [Online]. Available: https://doi.org/10.1214%2Faoms%2F1177729394.

[166] R. Y. Rubinstein, Ed., *Simulation and the Monte Carlo Method* (Wiley Series in Probability and Statistics), en. Hoboken, NJ, USA: John Wiley & Sons, Inc., Apr. 1981. DOI: 10.1002/9780470316511. [Online]. Available: http://doi.wiley.com/10.1002/9780470316511 (visited on 01/25/2022).

[167] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. New York, USA: Springer: Berlin Heidelberg, 2004.

[168] R. Y. Rubinstein, "Optimization of computer simulation models with rare events", *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997. DOI: 10.1016/s0377-2217(96)00385-2. [Online]. Available: https://doi.org/10.1016%2Fs0377-2217%2896%2900385-2.

[169] A. Ruszczynski and W. Syski, "Stochastic approximation method with gradient averaging for unconstrained problems", *IEEE Transactions on Automatic Control*, vol. 28, no. 12, pp. 1097–1105, Dec. 1983. DOI: 10.1109/tac.1983.1103184. [Online]. Available: https://doi.org/10.1109/tac.1983.1103184.

[170] R. de S. Motta and S. M. B. Afonso, "An efficient procedure for structural reliability-based robust design optimization", *Structural and Multidisciplinary Optimization*, vol. 54, no. 3, pp. 511–530, 2016. DOI: 10.1007/s00158-016-1418-1. [Online]. Available: https://doi.org/10.1007%2Fs00158-016-1418-1.

[171] J.-J. Samueli and A. Moatti, "Euler en défense de maupertuis à propos du principe de moindre action", *Bibnum. Textes fondateurs de la science*, 2012.

[172] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976. DOI: 10.1515/9780691214696. [Online]. Available: https://doi.org/10.1515%2F9780691214696.

[173] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, 3rd. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2021. DOI: 10.1137/1.9781611976595. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611976595. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611976595.

[174] T. Soma and Y. Yoshida, "Statistical learning with conditional value at risk", *arXiv preprint arXiv:2002.05826*, 2020.

[175] J. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation", *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992, ISSN: 00189286. DOI: 10.1109/9.119632. [Online]. Available: http://ieeexplore.ieee.org/document/119632/ (visited on 01/25/2022).

[176] M. Stein, "Large sample properties of simulations using latin hypercube sampling", *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987. [Online]. Available: http://www.jstor.org/stable/1269769.

[177] C. P. Stephens and W. Baritompa, "Global optimization requires global information", *Journal of Optimization Theory and Applications*, vol. 96, no. 3, pp. 575–588, 1998. DOI: 10.1023/a:1022612511618. [Online]. Available: https://doi.org/10.1023%2Fa%3A1022612511618.

[178] M. Styblinski and T.-S. Tang, "Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing", *Neural Networks*, vol. 3, no. 4, pp. 467–483, Jan. 1990. DOI: 10.1016/0893-6080(90)90029-k. [Online]. Available: https://doi.org/10.1016/0893-6080(90)90029-k.

[179] M. S. P. Subathra, S. E. Selvan, T. A. A. Victoire, A. H. Christinal, and U. Amato, "A hybrid with cross-entropy method and sequential quadratic programming to solve economic load dispatch problem", *IEEE Systems Journal*, vol. 9, no. 3, pp. 1031–1044, 2015. DOI: 10.1109/JSYST.2013.2297471.

[180] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016. DOI: 10.1109/cvpr.2016.308. [Online]. Available: https://doi.org/10.1109/cvpr.2016.308.

[181] A. Tamar, Y. Glassner, and S. Mannor, "Optimizing the CVaR via sampling", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015. DOI: 10.1609/aaai.v29i1.9561. [Online]. Available: https://doi.org/10.1609%2Faaai.v29i1.9561.

[182] J. Tao and N. Wang, "DNA Double Helix Based Hybrid GA for the Gasoline Blending Recipe Optimization Problem", *Chemical Engineering and Technology*, vol. 31, no. 3, pp. 440–451, 2008. DOI: 10.1002/ceat.200700322. [Online]. Available: http://dx.doi.org/10.1002/ceat.200700322.

[183] D. P. Thunnissen, "Uncertainty classification for the design and development of complex systems", in *3rd annual predictive methods conference*, Newport Beach CA, vol. 16, 2003.

[184] V. Torczon, "On the convergence of pattern search algorithms", *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 1–25, 1997. DOI: 10.1137/S1052623493250780. [Online]. Available: http://dx.doi.org/10.1137/S1052623493250780.

[185] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi, "Evolving mario levels in the latent space of a deep convolutional generative adversarial network", in *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, Jul. 2018. DOI: 10.1145/3205455.3205517. [Online]. Available: https://doi.org/10.1145/3205455.3205517.

[186] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview", *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2017. DOI: 10.1007/s00500-016-2474-6. [Online]. Available: https://doi.org/10.1007%2Fs00500-016-2474-6.

[187] K. Wang and N. Wang, "A novel RNA genetic algorithm for parameter estimation of dynamic systems", *Chemical Engineering Research and Design*, vol. 88, no. 11, pp. 1485–1493, 2010. DOI: 10.1016/j.cherd.2010.03.005. [Online]. Available: https://doi.org/10.1016/j.cherd.2010.03.005.

[188] L. Wang, Y. Ma, Y. Yang, and X. Wang, "Structural design optimization based on hybrid time-variant reliability measure under non-probabilistic convex uncertainties", *Applied Mathematical Modelling*, vol. 69, pp. 330–354, 2019. DOI: 10.1016/j.apm.2018.12.019. [Online]. Available: https://doi.org/10.1016%2Fj.apm.2018.12.019.

[189] D. Wolpert and W. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. DOI: 10.1109/4235.585893. [Online]. Available: https://doi.org/10.1109%2F4235.585893.

[190] W. Xie, "On distributionally robust chance constrained programs with wasserstein distance", *Mathematical Programming*, vol. 186, no. 1-2, pp. 115–155, 2019. DOI: 10.1007/s10107-019-01445-5. [Online]. Available: https://doi.org/10.1007%2Fs10107-019-01445-5.

[191] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations", *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002. DOI: 10.1137/s1064827501387826. [Online]. Available: https://doi.org/10.1137%2Fs1064827501387826.

[192] K. Xu *et al.*, "Structured adversarial attack: Towards general implementation and better interpretability", in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=BkgzniCqY7.

[193] Y. Xu and P. Wang, "CVaR formulation of reliability-based design problems considering the risk of extreme failure events", in *2021 Annual Reliability and Maintainability Symposium (RAMS)*, IEEE, 2021. DOI: 10.1109/rams48097.2021.9605753. [Online]. Available: https://doi.org/10.1109%2Frams48097.2021.9605753.

[194] M. Yang, D. Zhang, and X. Han, "Enriched single-loop approach for reliability-based design optimization of complex nonlinear problems", *Engineering with Computers*, vol. 38, no. 3, pp. 2431–2449, 2020. DOI: 10.1007/s00366-020-01198-2. [Online]. Available: https://doi.org/10.1007%2Fs00366-020-01198-2.

[195] X. Yuan and Z. Lu, "Efficient approach for reliability-based optimization based on weighted importance sampling approach", *Reliability Engineering & System Safety*, vol. 132, pp. 107–114, 2014. DOI: 10.1016/j.ress.2014.06.015. [Online]. Available: https://doi.org/10.1016%2Fj.ress.2014.06.015.

[196] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978. DOI: 10.1016/0165-0114(78)90029-5. [Online]. Available: https://doi.org/10.1016%2F0165-0114%2878%2990029-5.

[197] H. Zhang, W. Chen, A. Iyer, D. W. Apley, and W. Chen, *Uncertainty-aware mixed-variable machine learning for materials design*, en, Nov. 2022. DOI: 10.1038/s41598-022-23431-2. [Online]. Available: http://dx.doi.org/10.1038/s41598-022-23431-2.

[198] P. Zhang, "Nonparametric importance sampling", *Journal of the American Statistical Association*, vol. 91, no. 435, pp. 1245–1253, 1996. DOI: 10.1080/01621459.1996.10476994. [Online]. Available: https://doi.org/10.1080%2F01621459.1996.10476994.

[199] H. Zhu, J. Hale, and E. Zhou, "Simulation optimization of risk measures with adaptive risk levels", en, *Journal of Global Optimization*, vol. 70, no. 4, pp. 783–809, Apr. 2018, ISSN: 0925-5001, 1573-2916. DOI: 10.1007/s10898-017-0588-8. [Online]. Available: http://link.springer.com/10.1007/s10898-017-0588-8 (visited on 01/25/2022).