Les Cahiers du GERAD ISSN: 0711–2440

Risk averse constrained blackbox optimization under mixed aleatory/epistemic uncertainties

C. Audet, J. Bigeon, R. Couderc, M. Kokkolaras

G–2023–46

October 2023

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2023
– Bibliothèque et Archives Canada, 2023

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2023
– Library and Archives Canada, 2023

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Risk averse constrained blackbox optimization under mixed aleatory/epistemic uncertainties

Charles Audet [a, b]

Jean Bigeon [a, c]

Romain Couderc [a, b, d]

Michael Kokkolaras [a, e]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[b] *Département de mathématiques et génie industriel, École Polytechnique de Montréal, Montréal (Qc), Canada, H3C 3A7*

[c] *Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F–44000 Nantes, France*

[d] *Univ. Grenoble Alpes, CNRS, Grenoble INP*, G-SCOP, 38000 Grenoble, France*

[e] *Department of Mechanical Engineering, McGill University, Montréal (Qc), Canada, H3A 0C3*

charles.audet@gerad.ca
jean.bigeon@ls2n.fr
romain.couderc@grenoble-inp.fr
michael.kokkolaras@mcgill.ca

October 2023
Les Cahiers du GERAD
G–2023–46

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

1 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 [8]. This type of problem is common in signal processing [17], machine learning [40], and engineering design [1, 25]. In the presence of uncertainties, a constrained blackbox optimization problem may be formulated as follows

$$\begin{array}{cl} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & \Xi_0[C_0(\mathbf{x},\boldsymbol{\xi})] \\ \text{s.t.} & \Xi_j[C_j(\mathbf{x},\boldsymbol{\xi})] \leq 0,\ \forall j \in [1,m], \end{array} \tag{1}$$

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.

1.1 Related work

In probabilistic reliability-based design optimization (RBDO), uncertainties are considered as random vectors with known probabilistic distributions. In this field [18], Problem (1) is transformed into the following

$$\begin{array}{cl} \min\limits_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^n} & C_0(\mathbf{x},\mathbf{p}) \\ \text{s.t.} & \mathbb{P}[C_j(\mathbf{x},\mathbf{p}) \leq 0] \geq \alpha_j,\ \forall j \in [1,m], \end{array} \tag{2}$$

where $\alpha_j, j \in \{1, \ldots, m\}$ 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 (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 reliability methods (FORM/SORM) [16] are utilized. These methods transform the uncertainty vectors into uncorrelated Gaussian random vectors using the Rosenblatt or Nataf transformation [27], then the constraints are

[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.

approximated linearly or quadratically. Therefore, the probabilistic constraints in Problem (2) are reformulated as a deterministic optimization problem, reducing the task of solving Problem (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) [4], a single loop, such as the Single Loop Approach (SLA) [32], or decoupled approaches like the Sequential Optimization and Reliability Assessment (SORA) approach [19] or the Sequential Approximate Programming (SAP) approach [14]. These methods prove to be efficient even when dealing with nonlinear problems, and when gradients are approximated using finite differences [4]. 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 [51].

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 [11, 60], line sampling [3], subset simulation algorithms [6], or surrogate modeling strategies [29, 41]. 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 [28]. 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 [27, 28] 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 [43]. 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 or point data. Using probabilistic models for epistemic uncertainties may lead to infeasible designs in practice [43]. 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 [43]. A poorly chosen model can result in underperforming designs or designs with significant failures [45]. When epistemic uncertainties are involved in reliability analysis, non-probabilistic approaches based on evidence theory [52], possibility theory [20], or fuzzy sets [33, 61] may be used.

Recently, some approaches have utilized ellipsoidal sets to model uncertainties [36, 56]. When both types of uncertainties are present, combining probabilistic and non-probabilistic models to address these uncertainties may be an interesting option [21, 35]. Alternatively, distributionally robust chance-constrained programming [57] or a Bayesian probabilistic approach using Gaussian processes [2, 38] also appear promising. Finally, scenario optimization, that tackles the problem (1) using available data without prescribing a specific model (or a set of models) for the uncertainty, has been explored [44]. 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 [2], which is limited to parameter uncertainties. Another significant drawback is the lack of a convergence proof to an optimal point of the problem. Table 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.

Table 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 [4, 14, 19, 32]	O	✗	✗	✗	✗	✗
RBRDO [51]	O	✗	✓	✓	✗	✗
Importance Sampling [11, 60]	O	✓	✓	✗	✗	✗
Line Sampling [3]	RA	✓	N/A	N/A	✗	✓
Subset simulation [6]	RA	✓	N/A	N/A	✓	✗
Surrogate modelling [29, 41]	RA	✓	N/A	N/A	✓	✗
Mixed approaches [21, 35]	O	✗	✗	✗	✗	✓
Ellipsoidal set [36, 56]	O	✗	✗	✗	✗	✓
Bayesian approach (I) [38]	RA	✓	N/A	N/A	✓	✓
Bayesian approach (II) [2]	O	✓	✓	✓[3]	✓	✗
Scenario Optimization [44]	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.

1.2 Contributions

To account for the uncertainties in both the objective and constraint functions, methods utilizing the conditional value-at-risk (CVaR) have been developed [30, 49]. 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 [37], in engineering design problems [23, 31, 47, 58], and in constrained reinforcement learning [15, 55].

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 corresponds to the supremum of the function over the support of the uncertainties [48]. 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 [53, 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 [15, 42]. Unfortunately, the methods utilized in the referenced papers cannot be directly applied to solve a CVaR formulation of the problem (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 3, 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 [9, 39] 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 [22, 39], 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 (1). The properties of this new approximation and its associated unbiased gradient estimator are provided.

Second, Problem (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 [15], a Lagrangian relaxation is applied to the problem. The method used to solve the relaxed problem is developed in Section 4. 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 5 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, 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 7.

2 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 1. 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 1.1 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 3 to bound the approximation of the constrained CVaR blackbox problem and in Section 5 to study the convergence of the proposed method. Finally, the assumptions are increasingly strong, i.e., Assumption 1.3 implies Assumption 1.2, which implies Assumption 1.1.

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 [5] 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 [49]
$$\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \ := \ \min_{t \in \mathbb{R}} V_{\alpha_j}(\mathbf{x}, t), \tag{3}$$
where
$$V_{\alpha_j}(\mathbf{x}, t) \ = \ t + \frac{1}{1 - \alpha_j} \mathbb{E}_{\boldsymbol{\xi}}[(C_j(\mathbf{x}, \boldsymbol{\xi}) - t)^+], \tag{4}$$
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 (1) can be reformulated as a CVaR-constrained blackbox optimization problem:
$$\begin{array}{ll} \min\limits_{\mathbf{x} \in \mathcal{X}} & \mathrm{CVaR}_{\alpha_0}(\mathbf{x}) \\ \text{s.t.} & \mathrm{CVaR}_{\alpha_j}(\mathbf{x}) \leq 0, \ \forall j \in [1, m]. \end{array} \tag{5}$$
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 (5) a conservative approximation of Problem (2) [Chapter 6, [53]]. Thus, this formulation guarantees a conservative result in terms of failure probability, see e.g. [45]. To solve Problem (5), it is usually reformulated with the function V_α as follows
$$\begin{array}{ll} \min\limits_{(\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \mathbb{R}^{m+1}} & V_{\alpha_0}(\mathbf{x}, t_0) \\ \text{s.t.} & V_{\alpha_j}(\mathbf{x}, t_j) \leq 0, \ \forall j \in [1, m]. \end{array} \tag{6}$$
The equivalence between Problem (5) and Problem (6) is shown in the following lemma.

Lemma 2.1. Suppose the solution sets of Problem (5) and Problem (6) are not empty. Then these problems are equivalent in the sense that, $\mathbf{x}^*$ is a solution of Problem (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), and the optimal values are the same.

Proof. By the definition of the Conditional Value-at-Risk given in Equation (3), Problem (5) may be reformulated as follows

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \left(\min_{t_0\in\mathbb{R}} \quad V_{\alpha_0}(\mathbf{x}, t_0) \right) \\ &\text{s.t.} \left(\min_{t_j\in\mathbb{R}} \quad V_{\alpha_j}(\mathbf{x}, t_j) \right) \leq 0, \ \forall j \in [1, m]. \end{aligned} \tag{7}$$

Now, the following relations hold

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \left(\min_{t_0\in\mathbb{R}} V_{\alpha_0}(\mathbf{x}, t_0) \right) = \min_{(\mathbf{x},t_0)\in\mathcal{X}\times\mathbb{R}} V_{\alpha_0}(\mathbf{x}, t_0) \\ &\left(\min_{t_j\in\mathbb{R}} V_{\alpha_j}(\mathbf{x}, t_j) \right) \leq 0, \ \forall j \in [1, m] \iff \forall j, \ \exists t_j \text{ s.t. } V_{\alpha_j}(\mathbf{x}, t_j) \leq 0. \end{aligned}$$

Therefore, the Problems (7) and (6) are equivalent. Now, let $\mathbf{x}^*$ be a solution of Problem (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 (7) and as a consequence of Problem (6) which ends the proof. □

Despite this property, Problem (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.

3 Smooth approximation and Lagrangian relaxation of the problem

This section introduces a method for solving the Problem (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.

3.1 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 [50, 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

$$c^\beta(\mathbf{x}) \ := \int_{-\infty}^{+\infty} c(\mathbf{x} - \beta\mathbf{u})\phi(\mathbf{u})d\mathbf{u} \ = \ \mathbb{E}_{\mathbf{u}}[c(\mathbf{x} + \beta\mathbf{u})]. \tag{8}$$

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 [22, 39]. 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 [26], 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 3.1. 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

$$\tilde{\nabla} c^\beta(\mathbf{x}) \;=\; \frac{(\mathbf{u} - \boldsymbol{\mu})c(\mathbf{x} + \beta\mathbf{u}) - (\mathbf{u} - \boldsymbol{\mu})c(\mathbf{x})}{\beta}, \tag{9}$$

   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

$$\tilde{\nabla} c^\beta(\mathbf{x}) \;=\; \frac{(\mathbf{u}_1 - \boldsymbol{\mu}_1)(c(\mathbf{x} + \beta\mathbf{u}_1) - c(\mathbf{x})) - (\mathbf{u}_2 - \boldsymbol{\mu}_2)(c(\mathbf{x} - \beta\mathbf{u}_2) - c(\mathbf{x}))}{2\beta}, \tag{10}$$

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 $\mathbf{1}_{[\cdot]}(\cdot)$ 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

$$\frac{\partial c^\beta(\mathbf{x})}{\partial x_j} = \frac{\partial h_1(\mathbf{x})}{\partial x_j}\left(h_2 * h_3\right)(\mathbf{x}) + h_1(\mathbf{x})\left(h_2 * \frac{\partial h_3}{\partial x_j}\right)(\mathbf{x}).$$

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 [39] 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

$$\ln\left(\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} e^{-\frac{||\mathbf{v}-\mathbf{x}||^2}{2\beta^2}} d\mathbf{v}\right) \;=\; n\ln(\beta) + \frac{n}{2}\ln(2\pi) + \sum_{i=1}^{n} \ln\left(\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)\right). \tag{11}$$

Now, the derivative of the left-hand-side of Equation (11) with respect to β is given by

$$\begin{aligned}
\frac{\partial}{\partial\beta}\ln\left(\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} e^{-\frac{||\mathbf{v}-\mathbf{x}||^2}{2\beta^2}} d\mathbf{v}\right) &= \frac{1}{\kappa\beta^n}\int_{\mathbf{b}_\ell}^{\mathbf{b}_u} \frac{||\mathbf{v}-\mathbf{x}||^2}{\beta^3} e^{-\frac{||\mathbf{U}-\mathbf{x}||^2}{2\beta^2}} d\mathbf{v} \\
&= \frac{1}{\kappa\beta}\int_{\frac{\mathbf{b}_\ell-\mathbf{x}}{\beta}}^{\frac{\mathbf{b}_u-\mathbf{x}}{\beta}} ||\mathbf{u}||^2 e^{-\frac{||\mathbf{u}||^2}{2}} d\mathbf{u} \qquad\qquad \text{since } \frac{\mathbf{v}-\mathbf{x}}{\beta} = \mathbf{u} \\
&= \frac{1}{\beta}\mathbb{E}_\mathbf{u}[||\mathbf{u}||^2]
\end{aligned}$$

and the derivative of the right-hand-side of Equation (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,

$$\mathbb{E}_\mathbf{u}[||\mathbf{u}||^2] \;=\; n + \sum_{i=1}^{n} \frac{\frac{b_{\ell_i}-x_i}{\beta}\phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right) - \frac{b_{u_i}-x_i}{\beta}\phi\left(\frac{b_{u_i}-x_i}{\beta}\right)}{\Phi\left(\frac{b_{u_i}-x_i}{\beta}\right) - \Phi\left(\frac{b_{\ell_i}-x_i}{\beta}\right)} \;\leq\; n, \tag{12}$$

where the inequality holds because the sum is negative for $\mathbf{x} \in \mathcal{X}$. Finally, with the result in Equation (12) and the results of Lemma 1 of [39], 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

$$\tilde{\nabla}C^\beta(\mathbf{x}, \boldsymbol{\xi}) \;=\; \frac{(\mathbf{u}-\boldsymbol{\mu})\left(C(\mathbf{x}+\beta\mathbf{u}, \boldsymbol{\xi}_1) - C(\mathbf{x}, \boldsymbol{\xi}_2)\right)}{\beta}, \tag{13}$$

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

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

3.2 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 optimization problem is not novel; it has been explored in prior works [34, 54]. 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

$$\begin{aligned} V^{\beta}_{\alpha_j}(\mathbf{x}, t_j) &= \mathbb{E}_{\mathbf{u},\mathbf{v}}[V_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u}, t_j + \beta_2 v_j)], \quad \text{and} \\ \text{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) &= \min_{t_j \in \mathbb{R}} \mathbb{E}_{\mathbf{u},\mathbf{v}}[V_{\alpha_j}(\mathbf{x} + \beta_1 \mathbf{u}, t_j + \beta_2 v_j)]. \end{aligned}$$

Then, the smooth approximation of the Problem (6) may be formulated as follows

$$\begin{array}{cl} \displaystyle\min_{(\mathbf{x},\mathbf{t}) \in \mathcal{X} \times \mathbb{R}^{m+1}} & V^{\beta}_{\alpha_0}(\mathbf{x}, t_0) \\ \text{s.t.} & V^{\beta}_{\alpha_j}(\mathbf{x}, t_j) \leq 0, \ \forall j \in [1, m]. \end{array} \tag{14}$$

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 3.2. Under Assumption 1.2, the following holds.

1. $|\text{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) - \text{CVaR}_{\alpha_j}(\mathbf{x})| \leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}$ for all $j \in [0, m]$ and $\mathbf{x} \in \mathcal{X}$;
2. $|\text{CVaR}^{\beta}_{\alpha_0}(\tilde{\mathbf{x}}^*) - \text{CVaR}_{\alpha_0}(\mathbf{x}^*)| \leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}$, where $\tilde{\mathbf{x}}^*$ and $\mathbf{x}^*$ are solutions of Problem (14) and (5) respectively.
3. If Assumption 1.3 holds, then there exists a threshold $\bar{\alpha}_j \in (0, 1]$ such that for all $\alpha_j \geq \bar{\alpha}_j$

$$\text{CVaR}_{\alpha_j}(\mathbf{x}) \ \leq \ \text{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) \ \leq \ \text{CVaR}_{\alpha_j}(\mathbf{x}) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}.$$

Thus, for $\alpha_j \geq \bar{\alpha}_j$, if $\tilde{x}^*$ is a solution of Problem (14), then it is a feasible point for Problem (5).

Proof. 1. Under Assumption 1.2, it follows that for all $(\mathbf{x}, t_j) \in \mathcal{X} \times \mathbb{R}$

$$\begin{aligned} |V^{\beta}_{\alpha_j}(\mathbf{x}, t_j) - V_{\alpha_j}(\mathbf{x}, t_j)| &= \frac{1}{1 - \alpha_j} \left| \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - (t_j + \beta_2 v_j))^+ - (C_j(\mathbf{x}, \boldsymbol{\xi}_2) - t_j)^+] \right| \\ &\leq \frac{1}{1 - \alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[|(C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - (t_j + \beta_2 v_j))^+ - (C_j(\mathbf{x}, \boldsymbol{\xi}_2) - t_j)^+|] \\ &\leq \frac{1}{1 - \alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[|C_j(\mathbf{x} + \beta_1 \mathbf{u}, \boldsymbol{\xi}_1) - \beta_2 v_j - C_j(\mathbf{x}, \boldsymbol{\xi}_2)|] \\ &\leq \frac{1}{1 - \alpha_j} \mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[\kappa_2(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \beta_1 ||\mathbf{u}|| + \beta_2 |v_j|] \\ &\leq \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}, \end{aligned}$$

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 1.2 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 $\mathbf{x} \in \mathcal{X}$

$$V^{\beta}_{\alpha_j}(\mathbf{x}, \tilde{t}_j^*) \ \leq \ V^{\beta}_{\alpha_j}(\mathbf{x}, t_j^*) \ \leq \ V_{\alpha_j}(\mathbf{x}, t_j^*) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}.$$

Conversely, it also follows that

$$V_{\alpha_j}(\mathbf{x}, t_j^*) \ \leq \ V_{\alpha_j}(\mathbf{x}, \tilde{t}_j^*) \ \leq \ V^{\beta}_{\alpha_j}(\mathbf{x}, \tilde{t}_j^*) + \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j}.$$

Recalling that $\mathrm{CVaR}_{\alpha_j}(\mathbf{x}) = V_{\alpha_j}(\mathbf{x}, t_j^*)$ and $\mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) = V^{\beta}_{\alpha_j}(\mathbf{x}, \tilde{t}_j^*)$, we obtain that

$$|\mathrm{CVaR}^{\beta}_{\alpha_j}(\mathbf{x}) - \mathrm{CVaR}_{\alpha_j}(\mathbf{x})| \ \leq \ \frac{L_2 \beta_1 \sqrt{n} + \beta_2}{1 - \alpha_j} \ \forall j \in [1, m].$$

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 1.3 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 [46] 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 3.2.2 shows that the difference in the values of objective function of Problem (14) and Problem (5) is bounded by a constant that depends on the values of α_j, β_1, and β_2. Theorem 3.2.3 demonstrates that, with additional mild conditions, if α_j is chosen sufficiently close to 1, the solution obtained in Problem (14) is feasible for Problem (5). Therefore, the solution of Problem (14) may be feasible for Problem (5) and its value can be arbitrarily close to that of Problem (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 [12].

To solve Problem (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.

$$\max_{0 \le \boldsymbol{\lambda} \in \mathbb{R}^m} \min_{(\mathbf{x},\mathbf{t}) \in \mathcal{X} \times \mathbb{R}^{m+1}} L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) := V^{\beta}_{\alpha_0}(\mathbf{x},t_0) + \sum_{j=1}^{m} \lambda_j V^{\beta}_{\alpha_j}(\mathbf{x},t_j), \tag{15}$$

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 (15) whose the definition is recalled here.

Definition 3.3 (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 $\boldsymbol{\lambda} \ge 0$, we have

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

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

4 A risk averse multi-timescale stochastic approximation algorithm

Section 4.1 presents the multi-timescale stochastic approximation methods, and Section 4.2 describes the complete algorithm used to solve the Problem (15).

4.1 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 [9, 10] 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 [15, 42] 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 [10, Chapter 6] or [9, 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 2 holds.

Assumption 2. For $k \ge 0$, the step sizes sequences s_1^k, s_2^k, s_3^k and s_4^k are strictly positive and satisfy the requirements:

$$\begin{aligned}
&\sum s_1^k = \sum s_2^k = \sum s_3^k = \sum s_4^k = +\infty,\\
&\sum \left((s_1^k)^2 + (s_2^k)^2 + (s_3^k)^2 + (s_4^k)^2\right) < \infty,\\
&\lim_{k\to\infty} \frac{s_1^k}{s_2^k} = \lim_{k\to\infty} \frac{s_2^k}{s_3^k} = \lim_{k\to\infty} \frac{s_3^k}{s_4^k} = 0.
\end{aligned}$$

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 $\boldsymbol{\lambda}$.

Algorithm 1 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 2 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:
        $$\tilde{L}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}, \boldsymbol{\xi}) = \tilde{V}_{\alpha_0}(\mathbf{x}, t_0, \boldsymbol{\xi}) + \sum_{j=1}^{m} \lambda_j \tilde{V}_{\alpha_j}(\mathbf{x}, t_j, \boldsymbol{\xi}) \tag{16}$$
    where $\tilde{V}_{\alpha_j}(\mathbf{x}, t_j, \boldsymbol{\xi}) = t_j + \frac{1}{1-\alpha_j}\left(C_j(\mathbf{x}, \boldsymbol{\xi}) - t_j\right)^+$.
    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:
        $$\begin{aligned} \tilde{\mathbf{g}}_{\mathbf{x}}^k &= \frac{\left(\tilde{L}(\mathbf{x}^k + \beta_1 \mathbf{u}^k, \mathbf{t}^k + \beta_2 \mathbf{v}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_1^k) - \tilde{L}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_2^k)\right)(\mathbf{u}^k - \mu_1^k)}{\beta_1}, \\ \tilde{\mathbf{g}}_{\mathbf{t}}^k &= \frac{\left(\tilde{L}(\mathbf{x}^k + \beta_1 \mathbf{u}^k, \mathbf{t}^k + \beta_2 \mathbf{v}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_1^k) - \tilde{L}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k, \boldsymbol{\xi}_2^k)\right)(\mathbf{v}^k - \mu_2^k)}{\beta_2}, \\ \tilde{\mathbf{g}}_{\lambda_j}^k &= \tilde{V}_{\alpha_j}(\mathbf{x}^k, t_j^k, \boldsymbol{\xi}_1^k) \; \forall j \in [1, m]. \end{aligned} \tag{17}$$
    Update the long term gradient estimators:
        $$\begin{aligned} \mathbf{M}^{k+1} &= s_4^k \tilde{\mathbf{g}}^k + (1 - s_4^k)\mathbf{M}^k \\ \mathbf{V}^{k+1} &= s_4^k (\tilde{\mathbf{g}}^k)^2 + (1 - s_4^k)\mathbf{V}^k \end{aligned} \tag{18}$$
    Update the current iterates $\mathbf{x}^k$, $\mathbf{t}^k$ and $\boldsymbol{\lambda}^k$;
        $$\mathbf{t}^{k+1} = \Pi_{\mathcal{T}}\left[\mathbf{t}^k - s_3^k \frac{\mathbf{M}_{\mathbf{t}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{t}}^{k+1}} + \epsilon}\right] \tag{19}$$
        $$\mathbf{x}^{k+1} = \Pi_{\mathcal{X}}\left[x^k - s_2^k \frac{\mathbf{M}_{\mathbf{x}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{x}}^{k+1}} + \epsilon}\right] \tag{20}$$
        $$\boldsymbol{\lambda}^{k+1} = \Pi_{\mathcal{L}}\left[\boldsymbol{\lambda}^k + s_1^k \frac{\mathbf{M}_{\boldsymbol{\lambda}}^{k+1}}{\sqrt{\mathbf{V}_{\boldsymbol{\lambda}}^{k+1}} + \epsilon}\right] \tag{21}$$
    $k \leftarrow k + 1$
**end while**
Return $\mathbf{x}^k$
```

4.2 The RAMSA algorithm

Algorithm 1 summarizes the different updates. Note that when the square $(\cdot)^2$, the square root $\sqrt{\cdot}$ or division $\dot{-}$ operators are applied to a vector, it is elementwise. Further remarks about algorithm 1 are outlined:

- The updates (18) are the updates used to aggregate information about the gradient and are computed from the unbiased estimator defined in Equation (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 $||\mathbf{V}^k - (\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}))^2|| \to 0$ 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 (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 hyperrectangles, 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.

5 Convergence analysis

The convergence of the RAMSA algorithm is stated in the following theorem.

Theorem 5.1. Under Assumption 1.3 and Assumption 2, let further assume that the problem given in Equation (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 (14).

While the technical details of the proof of this theorem are given in Appendix A, 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 (14).

This convergence proof procedure is standard for multi-timescale stochastic approximation algorithms, see [9, chapter 10], [10, chapter 6] or [9, 15], 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

$$\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + s_1^k \left(f_1(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_1^{k+1}\right), \tag{22}$$

$$\mathbf{x}_2^{k+1} = \mathbf{x}_2^k + s_2^k \left(f_2(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_2^{k+1}\right), \tag{23}$$

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

$$\mathbf{x}_1^{k+1} = \mathbf{x}_1^k + s_2^k \left(\frac{s_1^k}{s_2^k}\left(f_1(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_1^{k+1}\right)\right), \tag{24}$$

$$\mathbf{x}_2^{k+1} = \mathbf{x}_2^k + s_2^k \left(f_2(\mathbf{x}_1^k, \mathbf{x}_2^k) + \delta_2^{k+1}\right). \tag{25}$$

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 [10, 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 5.1, it is proved that the iterations converge to a locally optimal solution of the problem given in Equation (14). It is possible to obtain a result for the original CVaR-constrained problem given in Equation (5) by utilizing Theorem 3.2. This is the subject of the following corollary.

Corollary 5.2. Under the same assumptions as Theorem 5.1, 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 (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 (5).

Proof. The proof is straightforward, considering the result of Theorems 5.1 and 3.2. □

This corollary is particularly interesting because it ensures the almost sure convergence of Algorithm 1 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 Computational implementations and numerical experiments

This section is divided into five parts: details of the numerical implementation are given in Section 6.1. Section 6.2 describes the setup of the experiments. Section 6.3 presents the experiments aimed at finding relations between the hyperparameters and the problems to be solved. Finally, Section 6.4 exhibits the results obtained using the truncated Gaussian gradient estimator instead of its classical counterpart, while Section 6.5 shows the results when the problem is subject to mixed aleatory/epistemic uncertainties.

6.1 Computational implementation

In this section, practical details of the implementation of Algorithm 1 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 [62]. 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 1. Here, α_j^* are the desired reliability levels and $\gamma \in [0, 1)$ is a fixed threshold.

6.2 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 [59], a Welded Beam Design (WBD) problem [59], a Vehicle Side Impact (VSI) problem [59], and a Speed Reducer Design (SRD) problem [13]. These problems are decribed in Appendix B, 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

$$\begin{array}{cl} \min\limits_{\mathbf{x} \in [0,1]^n} & \mathbb{E}_{\boldsymbol{\xi}}[\tilde{C}_0(\mathbf{x}, \boldsymbol{\xi})] \\ \text{s.t.} & \mathbb{P}(\tilde{C}_j(\mathbf{x}, \boldsymbol{\xi}) \leq 0) \geq 0.99, \ \forall j \in [1, m]. \end{array} \tag{26}$$

It is important to note that Problem (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 2 and 3 it solves formally a smooth Lagrangian relaxation of Problem (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 [13, 59]. 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 6.3 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 6.4, a comparison is made between the classical Gaussian gradient estimator and the truncated Gaussian gradient estimator introduced in Section 3.1. Third, the VSI problem is described slightly differently in [59], 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 [38]. Detailed descriptions of the conducted experiments are presented in Section 6.5.

6.3 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 2 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) [9, 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 (14) is particularly conservative for Problem (5), as shown in Theorem 3.2, and even more so for Problem (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 1 are conducted with different sets of hyperparameter values and the classical Gaussian gradient estimator [39, 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 2, while in Table 3 the associated average results of the trials are presented. Detailed results from the 100 runs of the trials are provided in Appendix C in the form of boxplots.

Table 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 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 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 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 (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 4. It is observed that, selecting β_1 to minimize the average variance and halving it, yields to similar results to those of Table 2.

Table 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 5 for each problem. The second line displays the result obtained in Table 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}.

Table 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

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 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 2 and $s_4^0 = 0.2$. The results obtained with this set of values are shown in Table 6.

Table 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 [12] 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 [7]. Nevertheless, this topic falls beyond the scope of the present paper and is not explored further here.

6.4 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 [26], 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 Appendix B 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 0.01for 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 7, and the detailed results from the 100 runs are depicted in boxplots in Appendix C.

In Table 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

Table 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

a side effect of using the truncated Gaussian distribution. However, a comprehensive investigation of this issue requires dedicated research, left for future work.

6.5 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 [38]. In the VSI problem presented in [59], 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 [59] 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, $j \in [1, m]$

$$\max_{(\mu_{\xi_8}, \mu_{\xi_9}) \in [0.192, 0.345]^2} C_j(\mathbf{x}^*, \mathbb{E}[\boldsymbol{\xi}]). \tag{27}$$

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 (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 $\mathbb{P}(C_7(\mathbf{x}^*_{SORA}, \boldsymbol{\xi}) \leq 0) \approx 0.88$.

Table 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 8, and for mixed aleatory/interval epistemic uncertainties in Table 9.

Table 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 (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 (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.

7 Concluding remarks

This work targets the constrained blackbox optimization problem given in Equation (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.

A Proof of Theorem 5.1

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 A.1. Under Assumption 1.3, 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 $\alpha_j \in (0,1)$

$$\begin{aligned}
&|\tilde{V}_{\alpha_j}(\mathbf{x} + \beta_1\mathbf{u}, (t_j + \beta_2 v_j, \boldsymbol{\xi}_1) - \tilde{V}_{\alpha_j}(\mathbf{y} + \beta_1\mathbf{u}, s_j + \beta_2 v_j, \boldsymbol{\xi}_2)| \\
&\leq |t_j - s_j| + \left|\left(C_j(\mathbf{x} + \beta_1\mathbf{u}, \boldsymbol{\xi}_1) - (t_j + \beta_1 v_j)\right)^+ - \left(C_j(\mathbf{y} + \beta_1\mathbf{u}, \boldsymbol{\xi}_2) - (s_j + \beta_1 v_j)\right)^+\right| \\
&\leq 2|t_j - s_j| + |C_j(\mathbf{x} + \beta_1\mathbf{u}, \boldsymbol{\xi}_1) - C_j(\mathbf{x} + \beta_1\mathbf{u}, \boldsymbol{\xi}_2)| \leq 2|t_j - s_j| + L_3||\mathbf{x} - \mathbf{y}|| \text{ a.s. },
\end{aligned}$$

where the second inequality follows from $|\max(a,0) - \max(b,0)| \leq |a - b|$ and the third is due to Assumption 1.3. 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 1.3, 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 A.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 A.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 [10, chapter 6], the convergence properties of the updates in Equation (18) 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

$$\mathbf{M}^{k+1} = \mathbf{M}^k + s_4^k \left(\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) - \mathbf{M}^k + \delta_{\mathbf{M}}^{k+1} \right), \tag{28}$$

$$\mathbf{V}^{k+1} = \mathbf{V}^k + s_4^k \left((\nabla L(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 + \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k) - \mathbf{V}^k + \delta_{\mathbf{V}}^{k+1} \right), \tag{29}$$

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 $\mathcal{F}^k = \sigma(\mathbf{x}^r, \mathbf{t}^r, \boldsymbol{\lambda}^r, \mathbf{M}^r, \mathbf{V}^r; r \le k)$. Now, the following Lemma may be stated to prove the convergence properties of the updates $\mathbf{M}$ and $\mathbf{V}$.

Lemma A.3. Consider the following continuous time system dynamics of the updates,

$$\begin{aligned} \dot{\mathbf{M}} &= h_1(\mathbf{M}, \mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) := \nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) - \mathbf{M}, \\ \dot{\mathbf{V}} &= h_2(\mathbf{M}, \mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) := (\nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) - \mathbf{V}, \\ (\dot{\mathbf{x}}, \dot{\mathbf{t}}, \dot{\boldsymbol{\lambda}}) &= (\mathbf{0}, \mathbf{0}, \mathbf{0}). \end{aligned} \tag{30}$$

This o.d.e. has a globally asymptotically stable equilibrium

$$\left\{ \left(\nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}), \nabla L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}))^2 + \mathbb{V}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}), \mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}\right) \,\middle|\, (\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) \in \mathcal{X} \times \mathcal{T} \times \mathcal{L} \right\},$$

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 (30) and the second part consists of verifying that all the assumptions needed to apply Lemma 1 in [10, 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 (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 (30). By a corollary of the LaSalle invariance theorem (see for instance [24, 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 A.1.

Now, we use the framework of the Lemma 1 in [10, Chapter 6].

(i) By Lemma A.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 A.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 2.

Under these conditions, Lemma 1 in [10, Chapter 6] may be applied, and the claim follows directly. □

Step 2: Convergence of the 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 [10, 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 A.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

$$\mathbf{t}^{k+1} = \Pi_{\mathcal{T}}\left[\mathbf{t}^k + s_3^k\left(-\Psi_{\mathbf{x}^k, \boldsymbol{\lambda}^k}(\mathbf{t}^k) + \delta_{\mathbf{t}}^{k+1}\right)\right], \text{ where } \begin{cases} \Psi_{\mathbf{x}^k, \boldsymbol{\lambda}^k}(\mathbf{t}^k) & = \frac{\nabla_{\mathbf{t}}^k L^\beta}{\sqrt{(\nabla_{\mathbf{t}}^k L^\beta)^2 + \mathbb{V}_{\mathbf{t}}^k + \epsilon}}, \\ \delta_{\mathbf{t}}^{k+1} & = \Psi_{\mathbf{x}^k, \boldsymbol{\lambda}^k}(\mathbf{t}^k) - \frac{\mathbf{M}_{\mathbf{t}}^{k+1}}{\sqrt{\mathbf{V}_{\mathbf{t}}^{k+1} + \epsilon}}. \end{cases} \tag{31}$$

Now, the following Lemma may be stated to prove the convergence properties of the update $\mathbf{t}$.

Lemma A.4. Consider the following continuous time system dynamics of the updates,

$$\begin{aligned} &\dot{\mathbf{t}} = \Gamma_{\mathbf{t}}\left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})\right] = \Gamma_{\mathbf{t}}\left[\frac{-\nabla_{\mathbf{t}} L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})}{\sqrt{\nabla_{\mathbf{t}} L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) + \mathbb{V}_{\mathbf{t}}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) + \epsilon}}\right], \\ &(\dot{\mathbf{x}}, \dot{\boldsymbol{\lambda}}) = (\mathbf{0}, \mathbf{0}), \end{aligned} \tag{32}$$

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 (32) is described. Let $(\mathbf{x}, \boldsymbol{\lambda}) \in \mathcal{X} \times \mathcal{L}$ be fixed and consider the following function

$$\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}) = L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) - L^\beta(\mathbf{x}, \mathbf{t}^*, \boldsymbol{\lambda}),$$

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

  $$\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})}{d\tau} = \nabla_{\mathbf{t}} L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})^T \; \Gamma_{\mathbf{t}} \left[-\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}) \right]$$

  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):

    $$\frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})}{d\tau} = -\sum_{j=0}^{m} \frac{\left(\frac{\partial L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})}{\partial t_j} \right)^2}{\sqrt{\left(\frac{\partial L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})}{\partial t_j} \right)^2 + \mathbb{V}_{t_j}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) + \epsilon}}.$$

  - 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 : $S^{\min} = \{j \in [0, m] \mid t_j = -(\mathbf{t}_{\max})_j\}, S^{\max} = \{j \in [0, m] \mid t_j = (\mathbf{t}_{\max})_j\}$ or $S^\circ = \{j \in [0, m] \mid t_j = (-(\mathbf{t}_{\max})_j, (\mathbf{t}_{\max})_j)\}$. Then, for the variables t_j whose the indices are in S°, there exists $\eta > 0$, sufficiently small such that $(t - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j \in (-(\mathbf{t}_{\max})_j, (\mathbf{t}_{\max})_j)$. For the variables t_j whose the variables are in $S^{\min}$, then either $(\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j \leq 0$, so $\Pi_{t_j}[(-\mathbf{t}_{\max} - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j] = (-\mathbf{t}_{\max} - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j$; or $(\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j > 0$ so $\Pi_{t_j}[-\mathbf{t}_{\max} - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j] = -(\mathbf{t}_{\max})_j$. For the variables t_j whose the variables are in $S^{\max}$, the symmetric result may be obtained. Therefore, it follows that

    $$\begin{aligned} \frac{d\mathcal{L}_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})}{d\tau} &= \lim_{0<\eta\to 0} \; \nabla_{\mathbf{t}} L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})^T \left(\frac{\Pi_{\mathbf{t}}[\mathbf{t} - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t})] - \mathbf{t}}{\eta} \right) \\ &= \lim_{0<\eta\to 0} \left(-\sum_{j \in S^\circ} \frac{\partial L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})}{\partial t_j} (\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j \right. \\ &\qquad - \sum_{j \in S^{\min}} \frac{\partial L^\beta(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda})}{\partial t_j} \frac{\Pi_{t_j}[(-\mathbf{t}_{\max} - \eta \Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j] + (\mathbf{t}_{\max})_j}{\eta} \end{aligned}$$

$$- \sum_{j\in S^{\max}} \frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j} \frac{\Pi_{t_i}[(\mathbf{t}_{\max} - \eta\Psi_{\mathbf{x},\boldsymbol{\lambda}}(\mathbf{t}))_j] - (\mathbf{t}_{\max})_j}{\eta}\Bigg)$$

$$\leq - \sum_{j\in S^{\circ}} \frac{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2}{\sqrt{\left(\frac{\partial L^{\beta}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda})}{\partial t_j}\right)^2 + \mathbb{V}_{t_j}(\mathbf{x},\mathbf{t},\boldsymbol{\lambda}) + \epsilon}},$$

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 [24, 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 [10, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma A.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 A.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 [10, Chapter 6] and the enveloppe theorem [15, Theorem 16], the claim follows directly. □

Step 3: Convergence of the 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 [10, 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 A.3 and Lemma A.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

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{X}}\left[\mathbf{x}^k + s_2^k\left(-\Psi_{\boldsymbol{\lambda}^k}(\mathbf{x}^k) + \delta_{1,\mathbf{x}}^{k+1} + \delta_{2,\mathbf{x}}^{k+1}\right)\right], \tag{33}$$

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 A.5. Consider the following continuous time system dynamics of the updates,

$$\dot{\mathbf{x}} = \Gamma_{\mathbf{x}}\left[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})\right] = \Gamma_{\mathbf{x}}\left[\frac{-\nabla_{\mathbf{x}} L^{\beta}(\mathbf{x},\mathbf{t}^{*}(\mathbf{x},\boldsymbol{\lambda}),\boldsymbol{\lambda})}{\sqrt{\nabla_{\mathbf{x}} L^{\beta}(\mathbf{x},\mathbf{t}^{*}(\mathbf{x},\boldsymbol{\lambda}),\boldsymbol{\lambda}) + \mathbb{V}_{\mathbf{x}}(\mathbf{x},\mathbf{t}^{*}(\mathbf{x},\boldsymbol{\lambda}),\boldsymbol{\lambda}) + \epsilon}}\right], \tag{34}$$

$$\dot{\boldsymbol{\lambda}} = \mathbf{0},$$

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

$$\big\{\big(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}\big) \mid \boldsymbol{\lambda} \in \mathcal{L}\big\}, \tag{35}$$

where $\mathbf{x}^*(\boldsymbol{\lambda}) = \{\mathbf{x} \in \mathcal{X} \mid \Gamma_{\mathbf{x}}[-\Psi_{\boldsymbol{\lambda}}(\mathbf{x})] = 0\}$ 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 (35).

Proof. First, the solutions of the first o.d.e. in Equation (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 A.4 and is omitted here). Therefore, the function is a Lyapunov function and, by Lyapunov stability theorem [24, 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 [10, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma A.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 $(\delta^{k+1}_{1,\mathbf{x}})$ and $(\delta^{k+1}_{2,\mathbf{x}})$ converges asymptotically to 0 by Lemma A.3 and Lemma A.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 [10, Chapter 6] from the iteration K, the claim follows directly. □

At this stage, the results obtained in Lemma A.4 and Lemma A.5 allows concluding that for any fixed $\boldsymbol{\lambda} \in \mathcal{L}$, the following holds:

$$(\mathbf{x}^k, \mathbf{t}^k) \to (\mathbf{x}^*(\boldsymbol{\lambda}), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda})) \in \mathcal{X} \times \mathcal{T}.$$

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 $||\mathbf{M}^k - \nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \to 0$, $||\mathbf{V}^k - (\nabla L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k))^2 - \mathbb{V}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k)|| \to 0$, $||\mathbf{t}^k - \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k)|| \to 0$ and $||\mathbf{x}^k \to \mathbf{x}^*(\boldsymbol{\lambda})|| \to 0$ almost surely. Therefore, by defining

$$\nabla^k_{\boldsymbol{\lambda}} L^\beta = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k), \mathbb{V}^k_{\boldsymbol{\lambda}} = \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^k, \mathbf{t}^k, \boldsymbol{\lambda}^k), \nabla^*_{\boldsymbol{\lambda}} L^\beta = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^k, \mathbf{t}^*(\mathbf{x}^k, \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)$$

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

$$\mathbf{x}^{k+1} = \Pi_{\mathcal{L}}\left[\boldsymbol{\lambda}^k + s_1^k\left(\Psi(\boldsymbol{\lambda}^k) + \delta^{k+1}_{1,\boldsymbol{\lambda}} + \delta^{k+1}_{2,\boldsymbol{\lambda}} + \delta^{k+1}_{3,\boldsymbol{\lambda}}\right)\right], \tag{36}$$

where

$$\begin{aligned}
\Psi(\boldsymbol{\lambda}^k) &= \frac{\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)}{\sqrt{(\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k))^2 + \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)} + \epsilon},\\
\delta^{k+1}_{1,\boldsymbol{\lambda}} &= \frac{\mathbf{M}^{k+1}_{\boldsymbol{\lambda}}}{\sqrt{\mathbf{V}^{k+1}_{\boldsymbol{\lambda}}} + \epsilon} - \frac{\nabla^k_{\boldsymbol{\lambda}} L^\beta}{\sqrt{(\nabla^k_{\boldsymbol{\lambda}} L^\beta)^2 + \mathbb{V}^k_{\boldsymbol{\lambda}}} + \epsilon},\\
\delta^{k+1}_{2,\boldsymbol{\lambda}} &= \frac{\nabla^k_{\boldsymbol{\lambda}} L^\beta}{\sqrt{(\nabla^k_{\boldsymbol{\lambda}} L^\beta)^2 + \mathbb{V}^k_{\boldsymbol{\lambda}}} + \epsilon} - \frac{\nabla^*_{\boldsymbol{\lambda}} L^\beta}{\sqrt{(\nabla^*_{\boldsymbol{\lambda}} L^\beta)^2 + \mathbb{V}^*_{\boldsymbol{\lambda}}} + \epsilon},\\
\delta^{k+1}_{3,\boldsymbol{\lambda}} &= \frac{\nabla^*_{\boldsymbol{\lambda}} L^\beta}{\sqrt{(\nabla^*_{\boldsymbol{\lambda}} L^\beta)^2 + \mathbb{V}^*_{\boldsymbol{\lambda}}} + \epsilon} - \Psi(\boldsymbol{\lambda}^k).
\end{aligned}$$

Now, the following Lemma may be stated to prove the convergence properties of the update $\boldsymbol{\lambda}$.
Lemma A.6. Let consider the following continuous time system dynamics of the updates,

$$\dot{\boldsymbol{\lambda}} = \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right] = \Gamma_{\boldsymbol{\lambda}}\left[\frac{\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)}{\sqrt{(\nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k))^2 + \mathbb{V}_{\boldsymbol{\lambda}}(\mathbf{x}^*(\boldsymbol{\lambda}^k), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k), \boldsymbol{\lambda}^k)} + \epsilon}\right], \tag{37}$$

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

$$\boldsymbol{\lambda}^* = \{\boldsymbol{\lambda} \in \mathcal{L} \mid \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right] = 0\}, \tag{38}$$

and the sequences $(\boldsymbol{\lambda}^k)$ converges almost surely to this local maximum given in Equation (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 (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

$$\frac{d\mathcal{L}(\boldsymbol{\lambda})}{d\tau} = \nabla_{\boldsymbol{\lambda}} L^\beta(\mathbf{x}^*(\boldsymbol{\lambda}), \mathbf{t}^*(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}), \boldsymbol{\lambda})^T\, \Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right],$$

which is negative definite (the proof may be done in the exact same way as the one given in Lemma A.4 and is omitted here). Therefore, the function is a Lyapunov function and, by Lyapunov stability theorem [24, 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 [10, Chapter 6] is used.

- The conditions (i) to (v) given in the proof of Lemma A.3 are still satisfied.
- The function $\Gamma_{\boldsymbol{\lambda}}\left[\Psi(\boldsymbol{\lambda})\right]$ is Lipschitz continuous by properties of ∇L^β.
- The random sequence $(\delta^{k+1}_{1,\boldsymbol{\lambda}})$, $(\delta^{k+1}_{2,\boldsymbol{\lambda}})$ and $(\delta^{k+1}_{3,\boldsymbol{\lambda}})$ converges asymptotically to 0 by Lemma A.3, Lemma A.4 and Lemma A.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 [10, 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 (14). This result is formally settled in Theorem 5.1 which is recalled here;

Theorem A.7. Under Assumption 1.3 and Assumption 2, let further assume that the problem given in Equation (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 (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 $j \in [1, m]$

$$c_j(\mathbf{x}^*, \mathbf{t}^*) := t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - (t_j^* + \beta_2 v_j))^+] \leq 0 \quad \text{and} \tag{39}$$

$$\lambda_j^* c_j(\mathbf{x}^*, \mathbf{t}^*) = \lambda_j^* \left(t_j^* + \frac{1}{1-\alpha}\mathbb{E}_{\mathbf{u},\mathbf{v},\boldsymbol{\xi}}[(C(\mathbf{x}^* + \beta_1\mathbf{u}, \boldsymbol{\xi}) - (t_j^* + \beta_2 v_j))^+] \right) = 0. \tag{40}$$

The proof of the inequality given in Equation (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 (38). Thus, the inequality given in Equation (39) holds. To show the result given in Equation (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 (38) and thus the equality in Equation (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 (14). □

B Analytical problems description

Here are the list of analytical problems considered in Section 6.1.
Steel column problem [59]

- 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 [59]: $\mathbf{x}^* = (257.7806, 13.5335, 100)$ 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 [59]

- 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 [59]: $x^* = [5.9188, 181.2849, 210.6114, 6.2253]$ 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 [59]

- 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) + 1.78(x_5 + \xi_5) \\
&\quad + 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, \\
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 [59]: $x^* = (0.7872, 1.35, 0.6887, 1.5, 1.0706, 1.2, 0.7284)$ 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 [13]

- 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 [59]: $x^* = (3.5765, 0.7, 17.0, 7.3, 7.7541, 3.3652, 5.3017)$ 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).

C Detailed numerical results

This section details the numerical results of Section 6.3 and Section 6.4. 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 6.3.

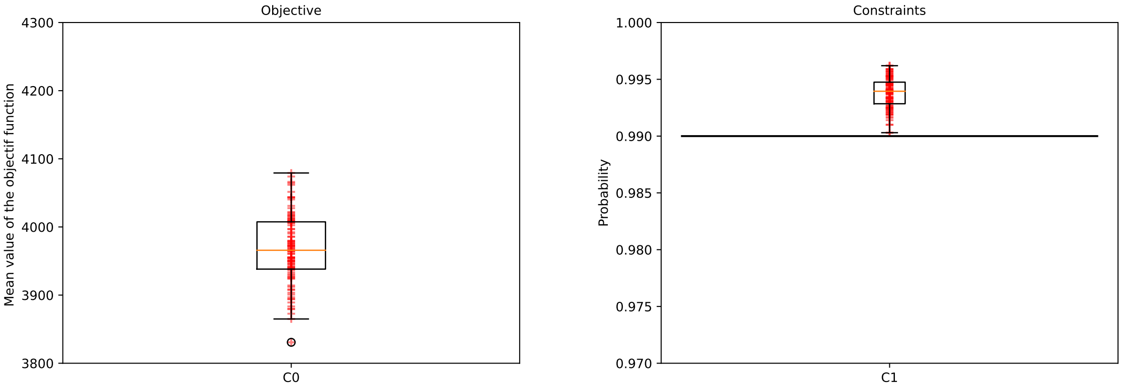

Figure 1: Detail result for Steel Column Design problem with classical Gaussian gradient approximation

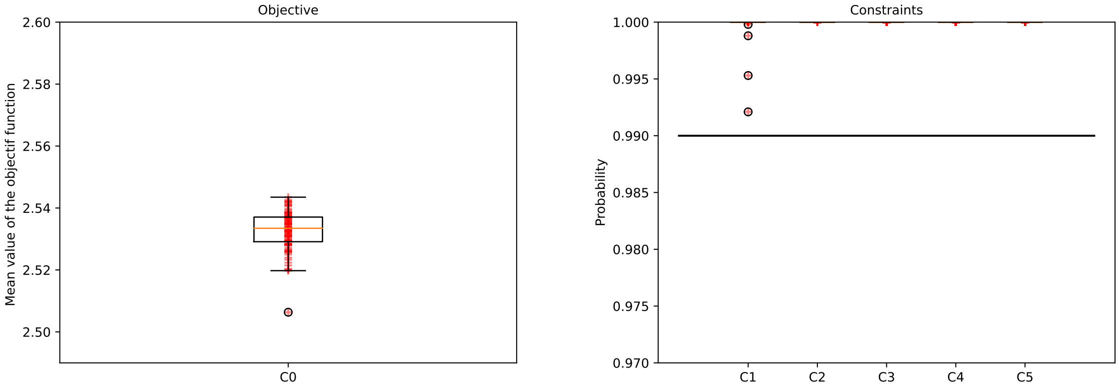

Figure 2: Detail result for Welded Beam Design problem with classical Gaussian gradient approximation

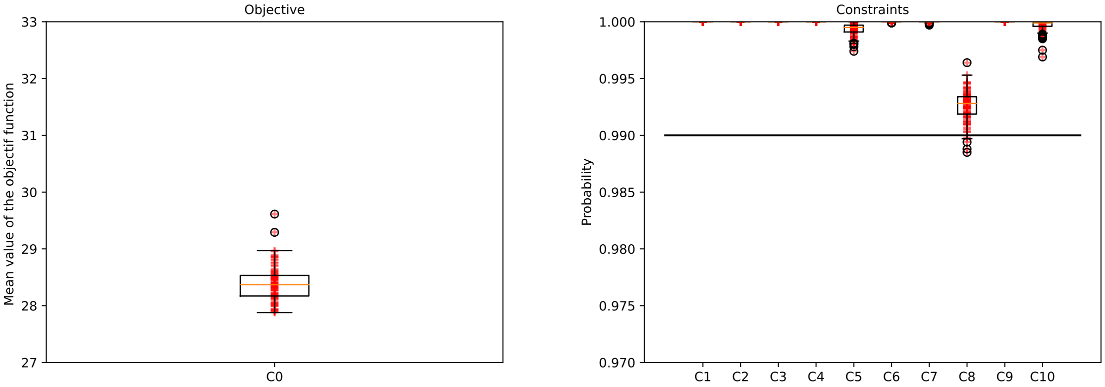

Figure 3: Detail result for Vehicle Side Impact problem with classical Gaussian gradient approximation

Here are the results for Section 6.4.

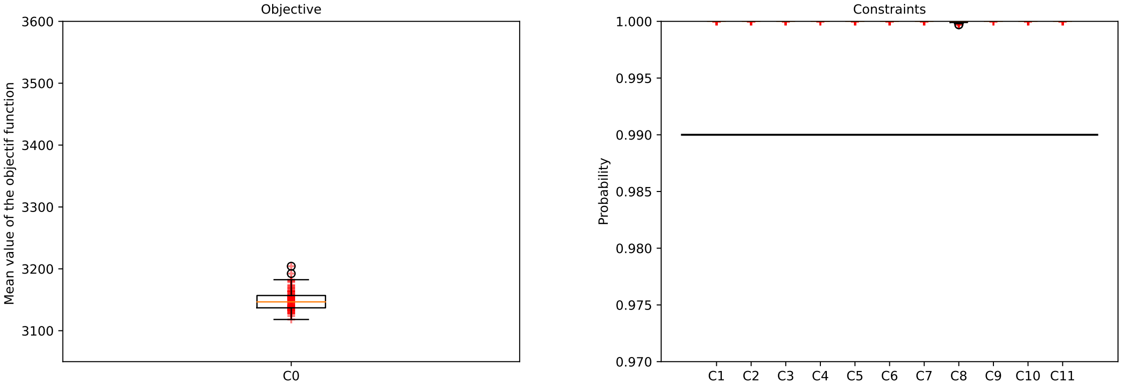

Figure 4: Detail result for Speed Reducer Design problem with classical Gaussian gradient approximation

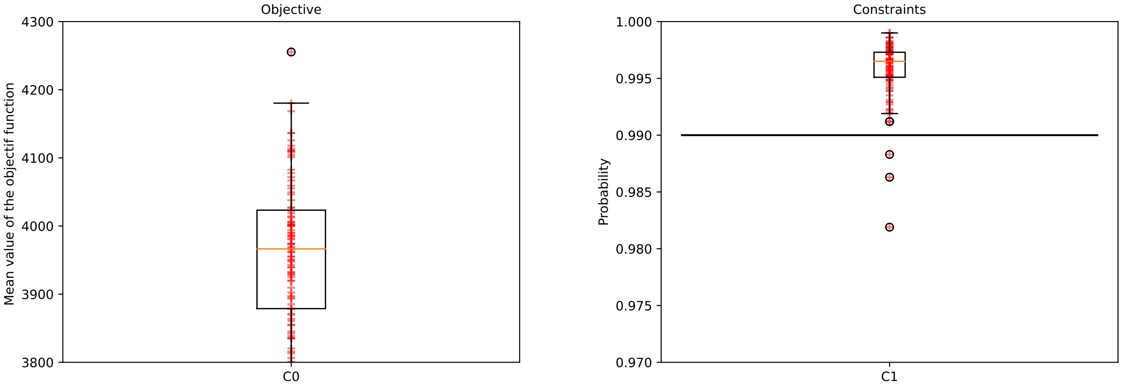

Figure 5: Detail result for Steel Column Design problem with truncated Gaussian gradient approximation

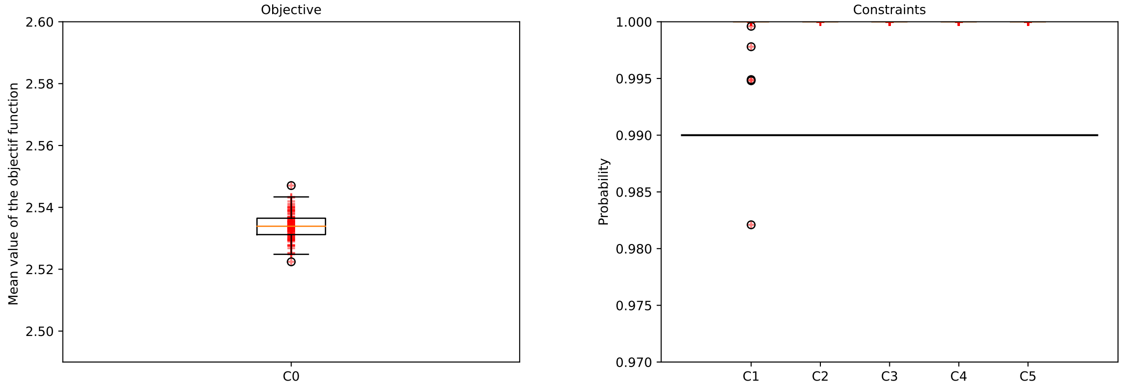

Figure 6: Detail result for Welded Beam Design problem with truncated Gaussian gradient approximation

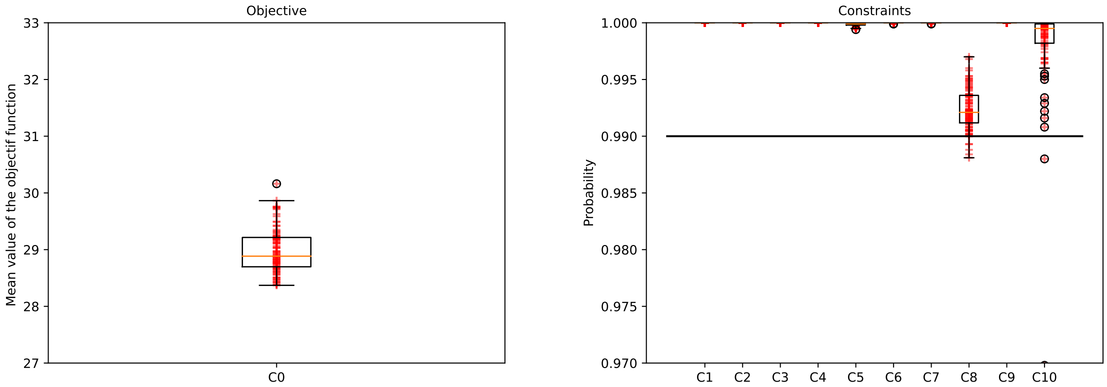

Figure 7: Detail result for Vehicle Side Impact problem with truncated Gaussian gradient approximation

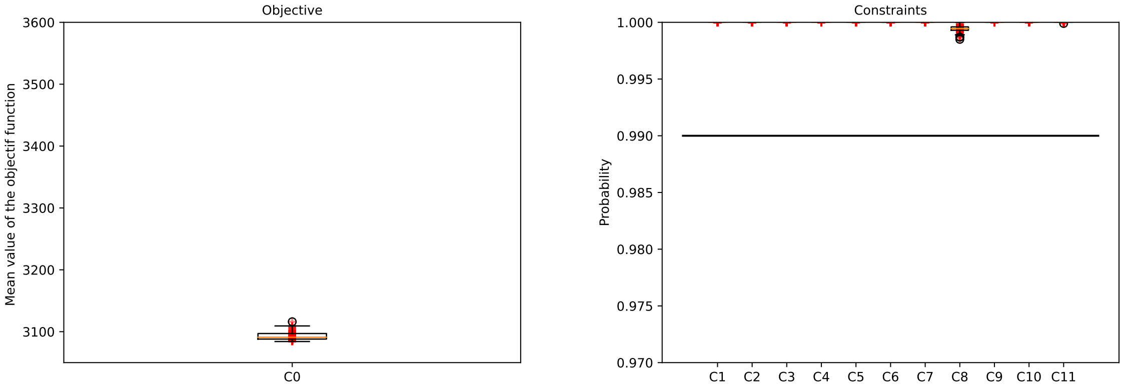

Figure 8: Detail result for Speed Reducer Design problem with truncated Gaussian gradient approximation

References

[1] Alarie, S., Audet, C., Gheribi, A.E., Kokkolaras, M., Le Digabel, S.: Two decades of blackbox optimization applications. EURO Journal on Computational Optimization 9, 100011 (2021). URL http://dx.doi.org/10.1016/j.ejco.2021.100011

[2] Amri, R.E., Riche, R.L., Helbert, C., Blanchet-Scalliet, C., Da Veiga, S.: A sampling criterion for constrained bayesian optimization with uncertainties. arXiv preprint arXiv:2103.05706 (2021)

[3] de Angelis, M., Patelli, E., Beer, M.: Advanced line sampling for efficient robust reliability analysis. Structural Safety 52, 170–182 (2015). URL http://dx.doi.org/10.1016/j.strusafe.2014.10.002

[4] Aoues, Y., Chateauneuf, A.: Benchmark study of numerical methods for reliability-based design optimization. Structural and Multidisciplinary Optimization 41, 277–294 (2010). URL https://doi.org/10.1007/s00158-009-0412-2

[5] Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. In: Risk Management, pp. 145–175. Cambridge University Press (2002). URL https://doi.org/10.1017%2Fcbo9780511615337.007

[6] Au, S., Ching, J., Beck, J.: Application of subset simulation methods to reliability benchmark problems. Structural Safety 29(3), 183–193 (2007). URL https://doi.org/10.1016%2Fj.strusafe.2006.07.008

[7] Audet, C., Bigeon, J., Couderc, R., Kokkolaras, M.: Sequential stochastic blackbox optimization with zeroth-order gradient estimators. AIMS Mathematics 8(11), 25922–25956 (2023). URL https://www.aimspress.com/article/doi/10.3934/math.20231321

[8] Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, Cham, Switzerland (2017). URL https://dx.doi.org/10.1007/978-3-319-68913-5

[9] Bhatnagar, S., Prasad, H., Prashanth, L.: Stochastic Recursive Algorithms for Optimization. Springer London (2013). URL http://dx.doi.org/10.1007/978-1-4471-4285-0

[10] Borkar, V.S.: Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press ; Hindustan Book Agency, Cambridge, UK : New York (2008)

[11] Chaudhuri, A., Kramer, B., Willcox, K.E.: Information reuse for importance sampling in reliability-based design optimization. Reliability Engineering & System Safety 201, 106853 (2020). URL https://doi.org/10.1016%2Fj.ress.2020.106853

[12] Chen, X., Liu, S., Xu, K., Li, X., Lin, X., Hong, M., Cox, D.: Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. Advances in Neural Information Processing Systems 32 (2019)

[13] Chen, Z., Qiu, H., Gao, L., Li, P.: An optimal shifting vector approach for efficient probabilistic design. Structural and Multidisciplinary Optimization 47(6), 905–920 (2013). URL https://doi.org/10.1007%2Fs00158-012-0873-6

[14] Cheng, G., Xu, L., Jiang, L.: A sequential approximate programming strategy for reliability-based structural optimization. Computers & Structures 84(21), 1353–1367 (2006). URL https://doi.org/10.1016%2Fj.compstruc.2006.03.006

[15] Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. The Journal of Machine Learning Research 18(1), 6070–6120 (2017)

[16] Cizelj, L., Mavko, B., Riesch-Oppermann, H.: Application of first and second order reliability methods in the safety assessment of cracked steam generator tubing. Nuclear Engineering and Design 147(3), 359–368 (1994). URL https://doi.org/10.1016%2F0029-5493%2894%2990218-6

[17] Curtis, F.E., Scheinberg, K.: Adaptive stochastic optimization: A framework for analyzing stochastic optimization algorithms. IEEE Signal Processing Magazine 37(5), 32–42 (2020). URL https://doi.org/10.1109%2Fmsp.2020.3003539

[18] Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A., Padmanabhan, D.: Reliability-based optimization using evolutionary algorithms. IEEE Transactions on Evolutionary Computation 13(5), 1054–1074 (2009). URL https://doi.org/10.1109%2Ftevc.2009.2014361

[19] Du, X., Chen, W.: Sequential optimization and reliability assessment method for efficient probabilistic design. Journal of Mechanical Design 126(2), 225–233 (2004). URL https://doi.org/10.1115%2F1.1649968

[20] Dubois, D.: Possibility theory, probability theory and multiple-valued logics: A clarification. In: Computational Intelligence. Theory and Applications, pp. 228–228. Springer Berlin Heidelberg (2001). URL https://doi.org/10.1007%2F3-540-45493-4_26

[21] Eldred, M., Swiler, L., Tang, G.: Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation. Reliability Engineering & System Safety 96(9), 1092–1113 (2011). URL https://doi.org/10.1016%2Fj.ress.2010.11.010

[22] Ghadimi, S., Lan, G.: Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization 23(4), 2341–2368 (2013). URL https://doi.org/10.1137/120880811

[23] Heinkenschloss, M., Kramer, B., Takhtaganov, T., Willcox, K.: Conditional-value-at-risk estimation via reduced-order models. SIAM/ASA Journal on Uncertainty Quantification 6(4), 1395–1423 (2018). URL https://doi.org/10.1137%2F17m1160069

[24] Khalil, H.: Nonlinear Systems. Pearson, 3rd edition, Prentice hall Upper Saddle River (2001)

[25] Kokkolaras, M., Mourelatos, Z.P., Papalambros, P.Y.: Impact of uncertainty quantification on design: an engine optimisation case study. International Journal of Reliability and Safety 1 (2006). DOI https://doi.org/10.1504/IJRS.2006.010786

[26] Le Digabel, S., Wild, S.M.: A taxonomy of constraints in simulation-based optimization. To appear in Optimization and Engineering (2023)

[27] Lebrun, R., Dutfoy, A.: A generalization of the nataf transformation to distributions with elliptical copula. Probabilistic Engineering Mechanics 24(2), 172–178 (2009). URL https://doi.org/10.1016%2Fj.probengmech.2008.05.001

[28] Lebrun, R., Dutfoy, A.: An innovating analysis of the nataf transformation from the copula viewpoint. Probabilistic Engineering Mechanics 24(3), 312–320 (2009). URL https://doi.org/10.1016%2Fj.probengmech.2008.08.001

[29] Li, J., Xiu, D.: Evaluation of failure probability via surrogate models. Journal of Computational Physics 229(23), 8966–8980 (2010). URL https://doi.org/10.1016%2Fj.jcp.2010.08.022

[30] Li, W., Li, C., Gao, L., Xiao, M.: Risk-based design optimization under hybrid uncertainties. Engineering with Computers 38(3), 2037–2049 (2020). URL https://doi.org/10.1007%2Fs00366-020-01196-4

[31] Li, W., Xiao, M., Garg, A., Gao, L.: A new approach to solve uncertain multidisciplinary design optimization based on conditional value at risk. IEEE Transactions on Automation Science and Engineering 18(1), 356–368 (2021). URL https://doi.org/10.1109%2Ftase.2020.2999380

[32] Liang, J., Mourelatos, Z.P., Nikolaidis, E.: A single-loop approach for system reliability-based design optimization. In: Volume 1: 32nd Design Automation Conference, Parts A and B. ASMEDC (2006). URL https://doi.org/10.1115%2Fdetc2006-99240

[33] Liu, Z.G., Liu, Y., Dezert, J., Cuzzolin, F.: Evidence combination based on credal belief redistribution for pattern classification. IEEE Transactions on Fuzzy Systems 28(4), 618–631 (2020). URL https://doi.org/10.1109%2Ftfuzz.2019.2911915

[34] Meng, F., Sun, J., Goh, M.: A smoothing sample average approximation method for stochastic optimization problems with CVaR risk measure. Computational Optimization and Applications 50(2), 379–401 (2010). URL https://doi.org/10.1007%2Fs10589-010-9328-4

[35] Meng, Z., Pang, Y., Pu, Y., Wang, X.: 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 363, 112886 (2020). URL https://doi.org/10.1016%2Fj.cma.2020.112886

[36] Meng, Z., Zhou, H.: New target performance approach for a super parametric convex model of non-probabilistic reliability-based design optimization. Computer methods in applied mechanics and engineering 339, 644–662 (2018)

[37] Menhorn, F., Augustin, F., Bungartz, H.J., Marzouk, Y.M.: A trust-region method for derivative-free nonlinear constrained stochastic optimization. arXiv preprint arXiv:1703.04156 (2017)

[38] Nannapaneni, S., Mahadevan, S.: Reliability analysis under epistemic uncertainty. Reliability Engineering & System Safety 155, 9–20 (2016). URL https://doi.org/10.1016%2Fj.ress.2016.06.005

[39] Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. Foundations of Computational Mathematics 17(2), 527–566 (2015). URL https://doi.org/10.1007/s10208-015-9296-2

[40] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM (2017). URL https://doi.org/10.1145/3052973.3053009

[41] Peherstorfer, B., Kramer, B., Willcox, K.: Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation. SIAM/ASA Journal on Uncertainty Quantification 6(2), 737–761 (2018). URL https://doi.org/10.1137%2F17m1122992

[42] Prashanth, L.A.: Policy gradients for CVaR-constrained MDPs. In: Lecture Notes in Computer Science, pp. 155–169. Springer International Publishing (2014). URL https://doi.org/10.1007%2F978-3-319-11662-4_12

[43] Rocchetta, R., Broggi, M., Patelli, E.: Do we have enough data? robust reliability via uncertainty quantification. Applied Mathematical Modelling 54, 710–721 (2018). URL https://doi.org/10.1016%2Fj.apm.2017.10.020

[44] Rocchetta, R., Crespo, L.G.: A scenario optimization approach to reliability-based and risk-based design: Soft-constrained modulation of failure probability bounds. Reliability Engineering & System Safety 216, 107900 (2021). URL https://doi.org/10.1016%2Fj.ress.2021.107900

[45] Rockafellar, R., Royset, J.: On buffered failure probability in design and optimization of structures. Reliability Engineering & System Safety 95(5), 499–510 (2010). URL https://doi.org/10.1016%2Fj.ress.2010.01.001

[46] Rockafellar, R.T., Royset, J.O.: Random variables, monotone relations, and convex analysis. Mathematical Programming 148(1-2), 297–331 (2014). URL https://doi.org/10.1007%2Fs10107-014-0801-1

[47] Rockafellar, R.T., Royset, J.O.: Engineering decisions under risk averseness. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 1(2) (2015). URL https://doi.org/10.1061%2Fajrua6.0000816

[48] Rockafellar, R.T., Royset, J.O.: Risk measures in engineering design under uncertainty. In: Proc. International Conf. on Applications of Statistics and Probability in Civil Engineering (2015)

[49] Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. The Journal of Risk 2(3), 21–41 (2000). URL https://doi.org/10.21314%2Fjor.2000.038

[50] Rubinstein, R.Y. (ed.): Simulation and the Monte Carlo Method. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA (1981). URL http://doi.wiley.com/10.1002/9780470316511

[51] de S. Motta, R., Afonso, S.M.B.: An efficient procedure for structural reliability-based robust design optimization. Structural and Multidisciplinary Optimization 54(3), 511–530 (2016). URL https://doi.org/10.1007%2Fs00158-016-1418-1

[52] Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976). URL https://doi.org/10.1515%2F9780691214696

[53] Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on Stochastic Programming: Modeling and Theory, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2021). URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976595

[54] Soma, T., Yoshida, Y.: Statistical learning with conditional value at risk. arXiv preprint arXiv:2002.05826 (2020)

[55] Tamar, A., Glassner, Y., Mannor, S.: Optimizing the CVaR via sampling. Proceedings of the AAAI Conference on Artificial Intelligence 29(1) (2015). URL https://doi.org/10.1609%2Faaai.v29i1.9561

[56] Wang, L., Ma, Y., Yang, Y., Wang, X.: Structural design optimization based on hybrid time-variant reliability measure under non-probabilistic convex uncertainties. Applied Mathematical Modelling 69, 330–354 (2019). URL https://doi.org/10.1016%2Fj.apm.2018.12.019

[57] Xie, W.: On distributionally robust chance constrained programs with wasserstein distance. Mathematical Programming 186(1–2), 115–155 (2019). URL https://doi.org/10.1007%2Fs10107-019-01445-5

[58] Xu, Y., Wang, P.: CVaR formulation of reliability-based design problems considering the risk of extreme failure events. In: 2021 Annual Reliability and Maintainability Symposium (RAMS). IEEE (2021). URL https://doi.org/10.1109%2Frams48097.2021.9605753

[59] Yang, M., Zhang, D., Han, X.: Enriched single-loop approach for reliability-based design optimization of complex nonlinear problems. Engineering with Computers 38(3), 2431–2449 (2020). URL https://doi.org/10.1007%2Fs00366-020-01198-2

[60] Yuan, X., Lu, Z.: Efficient approach for reliability-based optimization based on weighted importance sampling approach. Reliability Engineering & System Safety 132, 107–114 (2014). URL https://doi.org/10.1016%2Fj.ress.2014.06.015

[61] Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1(1), 3–28 (1978). URL https://doi.org/10.1016%2F0165-0114%2878%2990029-5

[62] Zhu, H., Hale, J., Zhou, E.: Simulation optimization of risk measures with adaptive risk levels. Journal of Global Optimization 70(4), 783–809 (2018). URL http://link.springer.com/10.1007/s10898-017-0588-8