Annals of Operations Research (2024) 334:205–240
https://doi.org/10.1007/s10479-022-04994-9

ORIGINAL RESEARCH

Monte carlo within simulated annealing for integral constrained optimizations

Roberto Casarin[1] · Bertrand B. Maillet[2,3] · Anthony Osuntuyi[1]

Accepted: 12 September 2022 / Published online: 30 September 2022

Abstract

For years, Value-at-Risk and Expected Shortfall have been well established measures of market risk and the Basel Committee on Banking Supervision recommends their use when controlling risk. But their computations might be intractable if we do not rely on simplifying assumptions, in particular on distributions of returns. One of the difficulties is linked to the need for Integral Constrained Optimizations. In this article, two new stochastic optimization-based Simulated Annealing algorithms are proposed for addressing problems associated with the use of statistical methods that rely on extremizing a non-necessarily differentiable criterion function, therefore facing the problem of the computation of a non-analytically reducible integral constraint. We first provide an illustrative example when maximizing an integral constrained likelihood for the stress-strength reliability that confirms the effectiveness of the algorithms. Our results indicate no clear difference in convergence, but we favor the use of the problem approximation strategy styled algorithm as it is less expensive in terms of computing time. Second, we run a classical financial problem such as portfolio optimization, showing the potential of our proposed methods in financial applications.

Keywords Finance · Simulated annealing · Markov chain monte carlo · Constrained optimization · Penalty method

✉ Anthony Osuntuyi
ayokunle.osuntuyi@unive.it

Roberto Casarin
r.casarin@unive.it

Bertrand B. Maillet
bmaillet@em-lyon.com

[1] Department of Economics, Ca' Foscari University of Venice, Fondamenta San Giobbe 873, 30121 Venice, Italy

[2] Aim Quant Research Center (Emlyon Business School), Paris, France

[3] University of La Reunion, Saint-Denis, France And Variances, Paris, France

1 Introduction

Risk management and asset allocation are the bulk of portfolio management, and are of great interest for investors, risk managers and regulators. In most related cases, however, no closed-form solutions are available for many risk and asset allocation problems, unless we make further stringent hypotheses. More generally, several statistical problems, such as inferential statistics (Isaacson and Madsen (1976), Dong and Wets (2000), Shapiro et al. (2014)), statistical process control (Feng et al. (2018), Glizer and Turetsky (2019)), and estimation of density function (Cramer (1989)) among others, rely indeed on constrained optimizations. In such cases, classical methods for solving constrained optimization problems often employ the Lagrangian method, which requires the selection of the Lagrange multiplier so that the constraint is satisfied (e.g., Bertsekas (2014) for an illustration).

However, on the one hand, it is reported in the literature that standard optimization methods often break down in many situations: when problems involve non-differentiable functions, large and multi-modal search spaces, non-convex or non-linear programs, multiple objectives, high-dimensional problems, or integral constraints. Charalambous (1978), for example, identifies a number of situations where classical numerical optimization techniques fail and where standard methods produce undesirable results.

On the other hand, Simulated Annealing (SA) algorithms are classically used to solve optimization problems. From a strictly practical point of view, the phenomenon of cooling applied to metals to obtain an optimal temperature share similarities with many financial application optimization problems when solved using SA, as shown by, e.g., Cramer (1989), Woodside-Oriakhi et al. (2011) and Castellano et al. (2021). As stated by Robert and Casella (2013), it appears that the SA algorithm borrowed its name from metallurgy and from physics. Especially in computational physics, the relationship with portfolio optimization is clearer, since the problem solved by SA is an optimization problem, where the function to be minimized is called "energy" and the variance factor which controls the convergence is called "temperature". In more technical terms, the underlying problems are basically the same, and SA algorithms are classically used to solve optimization problems in different areas of operation research such as, for instance: commodity queuing inventories (Shajin et al. (2021)), planning and release (Etgar et al. (1997), Etgar et al. (2017)), payment scheduling (He et al. (2009)), graph partitioning (D'Amico et al. (2002)), energy district design (Bergey et al. (2003)), political district design (Ricca and Simeone (2008)), and more recently for forecasting the relationship between oil and gas (Ftiti et al. (2020)). An area where SA reveals itself to be particularly useful is also risk management, with applications in many fields, e.g., tunnel construction (Ryu et al. (2015)), disaster risk management (Edrisi et al. (2020)) and facility layouts with uncertainty (Tayal and Singh (2018)). In our context of financial applications, it happens that a large number of optimization problems in finance can be solved by applying optimization heuristics, such as SA or Genetic Algorithms (GA), as for instance in Gilli and Schumann (2012). These methods can handle non-convex optimization problems overcoming many difficulties such as multiple local optima and discontinuities in the objective function. For example, Cramer (1989) applies SA to Markowitz (1952) portfolio selection problems with realistic constraints, whilst Woodside-Oriakhi et al. (2011) use SA for imposing a cardinality constraint onto the classical portfolio optimization program. Bick et al. (2013) also employ SA for solving constrained consumption-investment optimization problems, whilst Porth et al. (2016) propose to improve risk diversification combining portfolios of crop insurance policies and an SA approach to find the optimal allocation. More recently, Castellano et al. (2021) provide an approach to systemic risk minimization based on SA.

In this article, attention will be devoted to optimization problems for which: 1) the target function is not necessarily differentiable and 2) the constraints are characterized by non-analytically reducible integrals. It happens that analytical intractability of the integral constraint is a major cause of limits in the application of classical optimization techniques. We address this issue by exploring two alternative numerical optimization strategies. A first strategy consists of solving an auxiliary (approximated) problem, while the other entails the construction of an approximation algorithm. This model approximation approach decomposes further into two single sequential strategies, depending on the type of integral approximation technique employed, i.e. either simulation-based or deterministic-rule. In the latter case, the deterministic approximation approach generally involves the replacement of the non-analytically reducible integral with a numerical quadrature (see Shapiro et al. (2014) and Smith et al. (2015)). In the case of the simulation-based approach, the integral constraint is reformulated as a stochastic constraint and then a Monte Carlo simulation combined with another technique, such as the Sample Average Approximation (SAA), is implemented to approximate the problem. On the basis of these auxiliary problems, simulation-based optimization procedures can then be applied. However, the need to concurrently draw samples from both the decision and auxiliary variable spaces can make the algorithms computationally expensive. To alleviate this problem, methods such as the Group Independent Metropolis Hastings (GIMH) of Beaumont (2003) and the Inhomogeneous Markov Chain (IMC) simulation of de Mello and Bayraksan (2014) may be mobilized. As for the approximation algorithm approach, these entail, however, the replacement of the intractable target function by a (consistent or unbiased) estimator within the Markov Chain Monte Carlo (MCMC) algorithm. This procedure is referred to as the Monte Carlo Within Metropolis (MCWM) - Cf. Andrieu and Roberts (2009).

Before presenting our empirical results obtained in some settings of classical financial applications, the primary goal of this article is to design efficient and reliable simulation-based algorithms for solving integral constrained optimization problems by finding Extended Saddle Points (ESP) of a relevant penalty function reformulation of the integral constrained problem. Two SA algorithms are proposed, both relying on importance sampling strategies. Given a penalty function reformulation of the integral constrained problem, the first algorithm, called the Group Independence Metropolis Hastings Simulated Annealing (GIMHSA), requires the construction of an auxiliary problem upon which the GIMH method is implemented. The second method denominated as the Monte Carlo Within Metropolis Simulated Annealing (MCWMSA), differently, focuses on an approximation of the target (penalty) function used for the MCMC updating step of the MCWM method. We compare standard SA combined with quadrature and the two new algorithms first on a low dimension optimization problem given by the maximization of an integral constrained likelihood for a stress-strength reliability problem. Our empirical trials confirm the effectiveness of the two new algorithms and convergence properties similar to the standard SA. Nevertheless, in a higher dimension, standard SA becomes less effective while our proposed algorithm does not suffer from convergence issues. Our results indicate no clear differences in the convergence of the GIMHSA and MCWMSA (see Fig. 1) but we favor the use of the MCWMSA, since the coding difficulty is lower and it is rather quite less expensive in terms of computing time when the optimization problems are complex.

The SA algorithm we propose extends the Constrained SA (CSA) strategy based on the Extended Saddle Point (ESP) theory developed by Smith and Wong (2017) to an integral constrained setup. In contrast with the exogenously given penalty factor in the SA strategy proposed by Wah et al. (2006) and adopted in Smith et al. (2015), we endogenously obtain the penalty factors by augmenting the decision variables. More precisely, we optimize the penalty function by undertaking a minimization exercize over the decision variable space and maximization over the penalty factor space. This procedure therefore provides, in addition to the probabilistic descents in the problem-decision variable space as in conventional SA (as in Juditsky and Nemirovski (2020)), a probabilistic accent in the penalty factor subspace using a unified method based on a piecewise version of the Metropolis acceptance probability to control descents and ascents; this technique offers the benefit of reducing the risk for the chain to produce poor results. Embedded within the proposed SA-type algorithm is the simultaneous evaluation of the integral in the constraint which is done using a Monte Carlo integration procedure. The effectiveness of the algorithms is first assessed herein by applying them to an inferential problem on reliability in stress-strength with independent exponentiated exponential distributions. Our results in the simulation study here show that our algorithms are robust procedures for solving integral constrained statistical optimization problems.

The article is organized as follows. Section 2 introduces the integral constrained optimization problem and discusses the adopted method of the penalty function. In Sect. 3 we describe the constrained simulated annealing and discuss its implementation. Section 4 is dedicated to the empirical studies of the proposed algorithms, providing 3 applications using 6 sets of datasets used in previous publications. Section 5 concludes, whilst proofs and some robustness tests are left in the Appendix.

2 On integral constrained optimizations

Let $\mathcal{X} = \{x : x \in \mathbb{R}^n\}$ be the solution space (i.e. the set of all feasible solutions) where x are bounded continuous variables. Let $f : \mathcal{X} \to \mathbb{R}$ be a lower bounded objective function defined on the solution space, $\mathcal{X}$. Furthermore, let $g(x) = (g_1(x), \ldots, g_I(x))^T$ represent a vector of I equality constraint functions. Functions $f(\cdot)$ and $g(\cdot)$ are general functions that can be discontinuous and not in closed-form. In our case, one of the equality constraints, say g_1, is assumed to be characterized by an integral function whose solution is not available in closed-form. The goal is to find a global minimum, x^*, subject to the equality constraints[1] $g(x) = 0$ (i.e., $x^* \in \mathcal{X}$ such that $f(x) \geq f(x^*)$, $g(x) = 0$ for all $x \in \mathcal{X}$). We have the following mathematical representation of the problem:

$$\begin{aligned} &\min_{x \in \mathcal{X}} f(x) \\ &\text{subject to } g_i(x) = 0, \quad \forall\, i = 1, \ldots, I, \end{aligned} \tag{1}$$

where $g_1(x) = \displaystyle\int_{\mathcal{Y}} h(x, y)dy - \psi_0$ with ψ_0 a real constant, $\mathcal{Y} \subset \mathbb{R}^m$ and $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ a real-valued measurable function.

Several constraint handling methods have been developed in the literature for dealing with constrained optimization problems (Zhang et al. (2015)). In this article, we adopt the penalty function method because of its relative ease of use and amenability to complex optimization problems. More precisely, we reformulated the integral constrained optimization problem in Eq. 1 into an unconstrained one using the non-differentiable penalty described in Charalam-

[1] Without loss of generality, we only consider a minimization problem with equality constraints, since a maximization problem can be transformed into a minimization problem by negating the objective function while inequality constraints, say, $g_i(x) \leq 0$ can likewise be transformed into an equivalent equality constraint of the form $(g_i(x))^+ = \max\{0, g_i(x)\} = 0$.

bous (1978). Furthermore, by relying on Extended Saddle Points (ESP) theory (Wah et al. (2006)), the ESP solution of the following minmax problem:

$$\min_{x\in\mathcal{X}} \max_{\lambda\in\mathbb{R}_+^r} L(x,\lambda) = f(x) + \left(\sum_{i=1}^{r}(\lambda_i|g_i(x)|)^p\right)^{1/p}, \tag{2}$$

corresponds to a solution of the problem in Eq. 1, where $p \geq 1$, and $\lambda_i > 0$, $i = 1, 2, \ldots, I$, are penalty factors. Accordingly, stochastic optimization algorithms, such as the CSA proposed by Wah et al. (2006) may be applied. Unfortunately, the penalty function, $L(x,\lambda)$, in Eq. 2 cannot be evaluated analytically due to the integral constraint. In practice, researchers approximate the constraint, $g_1(x)$ by a numerical approximant, $g_1^M(x)$, so that $g_1^M(x)\longrightarrow g_1(x)$ $\forall x$ as M goes to infinity. We propose here to apply the Monte Carlo method for numerical integration with a focus on importance sampling. Following this argument, the integral can be written in the form:

$$g_1(x) = \int_{\mathcal{Y}} h(x,y)dy - \psi_0 = \int_{\mathcal{Y}} \frac{h(x,y)}{q(y;x)}q(y;x)dy - \psi_0 = E_q\left[\frac{h(x,Y)}{q(Y;x)}\right] - \psi_0, \tag{3}$$

where $q(y;x)$ is a probability density function on $\mathcal{Y}$, $Y{\sim}q(\cdot;x)$ and $E_q[\cdot]$ denotes the expectation with respect to the importance density q. The expectation can be approximated by the importance sampling estimator which reads:

$$g_1^M(x) = \frac{1}{M}\sum_{i=1}^{M}\frac{h(x,Y_i)}{q(Y_i;x)}, \tag{4}$$

where $Y_i \overset{iid}{\sim} q$, $i = 1, \ldots, M$, is a set of auxiliary variables. Since $g_1^M(x)$ converges to $g_1(x)$ almost surely uniformly in x provided that the $E_{q^M}[|g_1^M(x)|] < \infty$ (see Homem-de-Mello (2008), Feng et al. (2018) and Shapiro et al. (2014)), the approximation can always be made more precise by increasing M.

In light of the above, two alternative implementations of the importance sampling estimator (4) are explored in the construction of stochastic algorithms for estimating the ESP value and solution of the minmax problem in Eq. 2. The first consists of approximating the analytically intractable penalty function required for the computation of the acceptance probability of a simulated annealing type algorithm update, while the second entails optimizing an approximation of the analytically intractable penalty function in minmax problems of Eq. 2.

3 Simulated annealing for integral constrained optimization

In this section, we propose two SA type algorithms, based on the CSA developed by Wah et al. (2006), for solving the integral constrained problem in Eq. 1 by finding an ESP solution to the minmax problem in Eq. 2. We proceed with the construction of these algorithms in the following section.

3.1 Two simulated annealing type algorithms

Let us present the first algorithm proposal, i.e. the MCWMSA.

3.1.1 Monte carlo within metropolis simulated annealing (MCWMSA)

The MCWMSA algorithm is based on the combination of the Monte Carlo Within Metropolis (MCWM) of Andrieu and Roberts (2009) and the CSA of Wah et al. (2006). The MCWM is an iterative sampling algorithm where the MCMC uses an approximation of the target function in its execution. The MCWMSA consists of replacing the analytically intractable penalty function, $L(x, \lambda)$, required for the computation of the acceptance probability in the CSA by $L^M(x, \lambda)$ such that:

$$L^M(x,\lambda) = f(x) + \left((\lambda_1 |g_1^M(x)|)^p + \sum_{i=2}^{r} (\lambda_i |g_i(x)|)^p \right)^{1/p}. \tag{5}$$

In the next few lines, we shall present the details of the first Algorithm.

Algorithm 1

Let (x, λ) be a point in the $\mathcal{X} \times \mathbb{R}_+^r$ space and let $G(x, \lambda; x', \lambda')$ be an arbitrary proposal transition function satisfying the condition $G(x, \lambda; x', \lambda') > 0$ if and only if $G(x', \lambda'; x, \lambda) > 0$. Denote the target function corresponding to the minmax problem 2 by:

$$\pi(x,\lambda) \propto \exp\left(-L(x,\lambda)/T_k\right), \tag{6}$$

where $T_k = T_0 a^k$ (as in Wah et al. (2006)) is the temperature at the k-th iterate of the algorithm, T_0 the initial temperature, and $a \in (0, 1)$ the cooling rate. After setting the length of the Markov Chain N and the initial state (x, λ) with $\lambda = 0$, at the kth iteration the MCWMSA consists of the following steps:

1. Let $(x_{k,0}, \lambda_{k,0})$ be the current state and T_k the current temperature, then use the MCWM algorithm to update the Markov Chain $(x_{k,\ell}, \lambda_{k,\ell})$, $\ell = 1, \ldots, N$, as follows:

   1.a Draw $u_{k,\ell} \sim \mathcal{B}er(q)$ where $\mathcal{B}er(\cdot)$ denotes a Bernoulli distribution with q the probability to make a proposal in the solution space and $1 - q$ the probability to make a proposal in the penalty space.

   1.b Draw a trial solution $(x^*_{k,\ell}, \lambda^*_{k,\ell})$ from the proposal distribution $G(x_{k,\ell-1}, \lambda_{k,\ell-1}; \cdot, \cdot)$ given by:

$$G(x_{k,\ell-1}, \lambda_{k,\ell-1}; x^*_{k,\ell}, \lambda^*_{k,\ell}) = \begin{cases} H(x_{k,\ell-1}, x^*_{k,\ell})\delta(\lambda_{k,\ell-1} - \lambda^*_{k,\ell}) & \text{if } u_{k,\ell} = 1 \\ \tilde{H}(\lambda_{k,\ell-1}, \lambda^*_{k,\ell})\delta(x_{k,\ell-1} - x^*_{k,\ell}) & \text{if } u_{k,\ell} = 0, \end{cases} \tag{7}$$

   where $\delta(x - y)$ denotes a Dirac function which takes the value 1 if $x = y$ and 0 otherwise. $H(x, \cdot)$ and $\tilde{H}(\lambda, \cdot)$ are, respectively, the proposal distributions for generating samples for the decision variables and penalty variables. A detailed description of the proposal distributions $H(\cdot, \cdot)$ and $\tilde{H}(\cdot, \cdot)$ is presented in the next Subsection 3.1.3.

   1.c Draw importance samples $y_{k,\ell,1}, \ldots, y_{k,\ell,M}$ from $q(\cdot\,; x_{k,\ell-1})$ and $y^*_{k,\ell,1}, \ldots, y^*_{k,\ell,M}$ from $q(\cdot\,; x^*_{k,\ell})$.
   Then compute:

$$w^M(x^*_{k,\ell}, \lambda^*_{k,\ell}; x_{k,\ell-1}, \lambda_{k,\ell-1}) = \begin{cases} \pi^M(x^*_{k,\ell}, \lambda^*_{k,\ell})G_{l,l-1} & \text{if } u_{k,\ell} = 1 \\ \pi^M(x^*_{k,\ell}, \lambda^*_{k,\ell})^{-1}G_{l,l-1} & \text{if } u_{k,\ell} = 0, \end{cases} \tag{8}$$

where: $G_{l,l-1} = G(x^*_{k,\ell}, \lambda^*_{k,\ell}; x_{k,\ell-1}, \lambda_{k,\ell-1})$, and:

$$w^M(x_{k,\ell-1}, \lambda_{k,\ell-1}; x^*_{k,\ell}, \lambda^*_{k,\ell}) = \begin{cases} \pi^M(x_{k,\ell-1}, \lambda_{k,\ell-1}) G_{l-1,l} & \text{if } u_{k,\ell} = 1 \\ \pi^M(x_{k,\ell-1}, \lambda_{k,\ell-1})^{-1} G_{l-1,l} & \text{if } u_{k,\ell} = 0, \end{cases} \tag{9}$$

where: $G_{l-1,l} = G(x_{k,\ell-1}, \lambda_{k,\ell-1}; , x^*_{k,\ell}, \lambda^*_{k,\ell})$.
Then compute:

$$\pi^M(x_{k,\ell-1}, \lambda_{k,\ell-1}) = \exp(-L^M(x_{k,\ell-1}, \lambda_{k,\ell-1})/T_k),$$

and:

$$\pi^M(x^*_{k,\ell}, \lambda^*_{k,\ell}) = \exp(-L^M(x^*_{k,\ell}, \lambda^*_{k,\ell})/T_k),$$

where:

$$g_1^M(x) = \frac{1}{M}\sum_{i=1}^{M} \frac{h(x_{k,\ell-1}, y_{k,\ell,i})}{q(y_{k,\ell,i}; x_{k,\ell-1})} \text{ and } g_1^M(x^*_{k,\ell}) = \frac{1}{M}\sum_{i=1}^{M} \frac{h(x^*_{k,\ell}, y^*_{k,\ell,i})}{q(y^*_{k,\ell,i}; x^*_{k,\ell})}.$$

Accept the trial solution $(x^*_{k,\ell}, \lambda^*_{k,\ell})$ with probability:

$$A^M_{T_k}(x_{k,\ell-1}, \lambda_{k,\ell-1}; x^*_{k,\ell}, \lambda^*_{k,\ell}) = \min\left\{ \frac{w^M(x^*_{k,\ell}, \lambda^*_{k,\ell}; x_{k,\ell-1}, \lambda_{k,\ell-1})}{w^M(x_{k,\ell-1}, \lambda_{k,\ell-1}; x^*_{k,\ell}, \lambda^*_{k,\ell})}, 1 \right\}, \tag{10}$$

or reject it with probability $(1 - A^M_{T_k}(x_{k,\ell-1}, \lambda_{k,\ell-1}; x^*_{k,\ell}, \lambda^*_{k,\ell}))$.

2. Stop the MCWMSA and keep both the last solution $(x_{k,N}, \lambda_{k,N})$ and the corresponding estimated value of the minmax problem in Eq. 2 if the current temperature T_k is below a threshold value T^*; otherwise, update k to $k+1$, and proceed to step (1) with $(x_{k+1,0}, \lambda_{k+1,0}) = (x_{k,N}, \lambda_{k,N})$.

Algorithm#1 in Appendix C provides a pseudo-code for the MCWMSA.

3.1.2 Group independence metropolis hastings simulated annealing (GIMHSA)

Let us now turn to the presentation of the second proposed algorithm, i.e. the GIMHSA.

The GIMHSA is similar in spirit to the MCWMSA but differs in that the algorithm operates directly on an approximation of the minmax problem in Eq. 2 given by the following:

$$\min_{x\in\mathcal{X}} \max_{\lambda\in\mathbb{R}^r_+} L^M(x,\lambda) = f(x) + \left((\lambda_1 |g_1^M(x)|)^p + \sum_{i=2}^{r} (\lambda_i |g_i(x)|)^p \right)^{1/p}. \tag{11}$$

Since the minmax problem in Eq. 11 depends on the set of random vectors $Y_{1:M} = \{Y_1, \ldots, Y_M\}$, its ESP value, L^{*M}, is an estimator of the ESP value, L^*, of the problem in Eq. 2, and an ESP solution pair (x^{*M}, λ^{*M}) from the set of ESP solutions to problem 11 is an estimator of a ESP solution pair, (x^*, λ^*), of the problem 2.

In the next few lines, we now present in details the second algorithm proposal.

Algorithm 2

Let (x, λ) be a point in the $\mathcal{X} \times \mathbb{R}^r_+$ space and let $y_{1:M} = \{y_1, \ldots, y_M\}$ be a random sample of size M with $y_i \in \mathcal{Y}$ and $Y_i \sim q_x, \ \forall i$. Also, let $G(x, \lambda; x', \lambda')$ be an arbitrary proposal transition function satisfying the condition $G(x, \lambda; x', \lambda') > 0$ if and only if $G(x', \lambda'; x, \lambda) > 0$. Denote the target function corresponding to the minmax problem 11 by:

$$\pi^M(x, \lambda) \propto \exp\left(-L^M(x, \lambda)/T_k\right), \tag{12}$$

where $T_k = T_0 a^k$ is the temperature at the k-th iteration of the algorithm, T_0 the initial temperature, and $a \in (0, 1)$ the cooling rate. After setting the length of the Markov Chain N and the initial state (x, λ) with $\lambda = 0$, the GIMHSA iterates the following steps:

1. Let (x, λ) be the current state, $y_{1:M} = \{y_1, \ldots . y_M\}$ the current collection of importance samples drawn independently from q_x, and T_k the current temperature. Compute:

$$\pi^M(x, \lambda) \propto \exp(-L^M(x, \lambda)/T_k), \quad \text{with} \ \ g_1^M(x) = \frac{1}{M}\sum_{i=1}^{M} \frac{h(x, y_i)}{q_x(y_i)},$$

then use the Metropolis Hastings algorithm to update the Markov Chain N times as follows:

1.a Draw $u \sim \mathcal{B}er(q)$ where q is the probability to make a proposal in the solution space and $1 - q$ the probability to make a proposal in the penalty space.

1.b Draw a trial solution (x^*, λ^*) from the proposal distribution $G(x, \lambda; \cdot, \cdot)$ given by:

$$G(x, \lambda, x^*, \lambda^*) = \begin{cases} H(x, x^*)\delta_\lambda(\lambda^*) & \text{if } u = 1 \\ \tilde{H}(\lambda, \lambda^*)\delta_x(x^*) & \text{if } u = 0, \end{cases} \tag{13}$$

where $H(x, \cdot)$ and $\tilde{H}(\lambda, \cdot)$ are respectively the proposal distributions for generating samples for the decision variables and penalty variables. A detailed description of the proposal distributions $H(\cdot, \cdot)$ and $\tilde{H}(\cdot, \cdot)$ is presented in Sect. 3.1.3.

1.c Draw importance samples $y_1^*, \ldots, y_M^*$ from q_{x^*}. Then compute:

$$w^M(x^*, \lambda^*; x, \lambda) = \begin{cases} \pi^M(x^*, \lambda^*)G(x^*, \lambda^*; x, \lambda) & \text{if } u = 1 \\ \pi^M(x^*, \lambda^*)^{-1}G(x^*, \lambda^*; x, \lambda) & \text{if } u = 0, \end{cases} \tag{14}$$

where:

$$\pi^M(x^*, \lambda^*) \propto \exp(-L^M(x^*, \lambda^*)/T_k), \quad \text{with} \ \ g_1^M(x^*) = \frac{1}{M}\sum_{i=1}^{M} \frac{h(x, y_i^*)}{q_{x^*}(y_i^*)}.$$

Accept the trial solution (x^*, λ^*) and $y_{1:M}^*$ with probability:

$$A_{T_k}^M(x, \lambda; x^*, \lambda^*) = \min\left\{\frac{w^M(x^*, \lambda^*; x, \lambda)}{w^M(x, \lambda; x^*, \lambda^*)}, 1\right\}, \tag{15}$$

or reject it with probability $(1 - A_{T_k}^M(x, \lambda; x^*, \lambda^*))$.

2. Stop the GIMHSA and keep both the last solution (x, λ), $y_{1:M}$ and the corresponding estimated value of the minmax problem in Eq. 11 if the current temperature T_k is less than 10^{-6}; otherwise, update k to $k + 1$, and proceed to step (1).

It is worth noting that the GIMHSA procedure may also be viewed as an approximation of a Metropolis Hastings algorithm with target density $\pi(x, \lambda)$, making it a special case of the MCWMSA algorithm. This dual interpretation of the GIMHSA presents an opportunity for designing algorithms that inherit the potential efficiency of a Markov Chain with a one-step transition kernel (denoted later on P_{GIMHSA}, see sub-Section 3.2 below) while still being able to produce samples from the target function, and not an approximation. A pseudo-code for the GIMHSA is provided in Algorithm#1 in Appendix C.

3.1.3 Proposal distributions $H(x, x')$ and $\tilde{H}(\lambda, \lambda')$

In generating random points, (x', λ'), from the neighbourhood, $\mathcal{N}(x, \lambda)$, of the current point (x, λ) using the proposal distributions $G(x, \lambda; x', \lambda')$, we start off, at each iteration step, by making a decision either to update the decision variables, x, while the penalty factors are kept fixed or as an alternative to update the penalty factor, λ, maintaining the decision variables constant. In our Metropolis Hastings SA version, decision variables, x, need to be updated more frequently than the penalty factors, λ, to enforce the convergence of the algorithm. Thus, we choose large values of the generation ratio, q, for example $q = 20n/(20n+r)$ with n and r being, respectively, the number of decision variables and the number of constraints. Upon doing this, we then draw trial solutions for the decision variables from $H(x, \cdot)$ and the penalty factors from $\tilde{H}(\lambda, \cdot)$. Each of these proposal distributions are described below.
As regards to the proposal distribution $H(x, \cdot)$ for the decision variables, given the challenging task of finding one good proposal distribution in n dimensions for generating x' from the neighbourhood of x, we propose n good proposal distributions for updating each component of x. However, to improve the efficiency of the algorithm, we only make x' to differ from x in the j-th component at each iteration step. Let j be a uniform number in $\{1, 2, \ldots, n\}$ and $\{\sigma_j : 1 \leq j \leq n\}$, a set of parameters where σ_j represents the scale parameter for the proposal distribution of the j-th decision variable. We generate a trial solution for the decision variables as follows:

$$x' = x + (\theta_1 e^{(1)}_{j,1}, \theta_2 e^{(1)}_{j,2}, \ldots, \theta_n e^{(1)}_{j,n})^T, \tag{16}$$

where $\mathbf{e}^{(1)}_j = (e^{(1)}_{j,1}, \ldots, e^{(1)}_{j,n})$ is the j-th column of an $n \times n$ identity matrix and θ_j is Cauchy distributed as such:

$$f_\theta(\theta_j) = \frac{1}{\pi} \frac{\sigma_j}{\sigma_j^2 + \theta_j^2}.$$

In addition, we dynamically update the scale parameter of the Cauchy proposal corresponding to the $j-$th component of the decision variable, σ_j according to the following rule (see Andrieu and Thoms (2008)):

$$\log(\sigma_j^{(k+1)}) = \log(\sigma_j^{(k)}) + \frac{1}{(k+1)}(\overline{A}_{j,T_k} - A),$$

where $\overline{A}_{j,T_k}$ denotes the simple average of the acceptance probabilities associated with the $j-$th decision variable at the k-th temperature step, and A is the target acceptance rate set to 0.44 as in Andrieu and Thoms (2008).

In generating the trial point λ' from $\tilde{H}(\lambda, \cdot)$, we follow a similar strategy to that of the decision variables. More precisely, we generate λ' so that it differs from λ only in the j-th component where j is a uniform random number on the set $\{1, 2, \ldots, I\}$. A trial point for λ'_j is drawn from a uniform distribution over the interval $(a_j^{(\lambda)}, b_j^{(\lambda)})$ with $a_j^{(\lambda)} = \max\{\lambda_j -$

$w_j^{(\lambda)} h_j(z), 0\}$, $b_j^{(\lambda)} = \min\{\lambda_j + w_j^{(\lambda)} h_j(z), 10,000\}$ and $w_j^{(\lambda)}$ represent a scale parameter that controls the width of the search space. Note that the neighbourhood of λ_j from which λ'_j is generated is adjusted according to the degree of constraint violation. Thus, whenever $g_j(x) = 0$ is satisfied, λ_j does not need to be updated. Furthermore, we adjust the scale parameter, $w_j^{(\lambda)}$, according to the following rule:

$$w_j^{(\lambda)} = \begin{cases} \eta_0 w_j^{(\lambda)} & \text{if } g_j(x) > \tau_0 T \\ \eta_1 w_j^{(\lambda)} & \text{if } g_j(x) < \tau_1 T \\ \text{unchanged} & \text{otherwise,} \end{cases} \tag{17}$$

where we set, experimentally, $\eta_0 = 1.25$, $\eta_1 = 0.95$, $\tau_0 = 1.0$ and $\tau_1 = 0.01$ so that the MH chain is not moving too fast in the penalty subspace. That is, when $g_j(x)$ is reduced too quickly (i.e. $g_j(x) < \tau_1 T$ is satisfied), $g_j(x)$ is over-weighted, leading to a possibly poor objective value or to some difficulties in satisfying other under-weighted constraints. Hence, we reduce λ_j's neighborhood. In contrast, if $g_j(x)$ is reduced too slowly (i.e. $g_j(x) > \tau_0 T$ is satisfied), we enlarge λ_j's neighborhood in order to improve its chance of satisfaction. Note that w_j^{λ} is adjusted using T as a reference because constraint violations are expected to decrease when T decreases.

3.2 Asymptotic convergence of MCWMSA and GIMSA

In this section, we show the asymptotic convergence of both the MCWMSA and GIMSA to a constrained global minimum in a discrete constrained optimization problem.

3.2.1 Asymptotic convergence of MCWMSA

The MCWMSA algorithm (Algorithm 1) describes a Markov Chain (x_i, λ_i), $i = 1, 2, \ldots$, in the state vector (x, λ) since the auxiliary variables y are drawn independently at each iteration. Modeling this chain as an Inhomogeneous Markov Chain that consists of a sequence of Homogeneous Markov Chains of a finite length, each at a specific temperature in a cooling schedule. The one-step transition kernel, $P_{MCWMSA}(x, \lambda; x', \lambda')$, of the MCWMSA algorithm from (x, λ) to (x', λ') is:

$$\begin{aligned} P_{MCWMSA}(x, \lambda; x', \lambda') &= \int_{\mathcal{Y}^M \times \mathcal{Y}^M} G(x, \lambda; x', \lambda') A(x, \lambda, y; x', \lambda', y') dy dy' \\ &+ \delta_{(x', \lambda')}(x, \lambda) \left(1 - \int_{\mathcal{N}(x,\lambda)} \int_{\mathcal{Y}^M \times \mathcal{Y}^M} A(x, \lambda, y; x', \lambda', y') dy dy' G(x, \lambda; dx', d\lambda') \right), \end{aligned} \tag{18}$$

where:

$$A(x, \lambda, y; x', \lambda', y') = A_T^M(x, \lambda, y; x', \lambda', y') q_x^M(y) q_{x'}^M(y'),$$

and $\mathcal{N}(x, \lambda)$ denotes the neighborhood of (x, λ), q_x^M the joint probability distribution of the importance densities and $G(x, \lambda; x', \lambda')$, the proposal distribution of $(x', \lambda') \in \mathcal{N}(x, \lambda)$ conditional on (x, λ) which satisfies:

$$G(x, \lambda; x', \lambda') = G(x', \lambda'; x, \lambda) \text{ and } 0 \le G(x, \lambda; x', \lambda') \le 1, \tag{19}$$

with:

$$\int_{\mathcal{N}(x,\lambda)} G(x, \lambda; dx', d\lambda') = 1. \tag{20}$$

Since the acceptance probability in the MCWMSA depends upon an approximation of the target function, the invariant distribution of P_{MCWMSA} is not $\pi(x, \lambda)$ and therefore MCWMSA will not produce samples from $\pi(x, \lambda)$ even in a steady state. However, by relying on the ergodicity of a one-step transition kernel, P_{MCWM}, of an MCWM sampling procedure, Andrieu and Roberts (2009) show that these samples are distributed asymptotically according to an approximation of $\pi(x, \lambda)$, which should be all more precise, as the sample size grows large. In view of this, it suffices to show that the Markov Chain is ergodic.

Theorem 1 *Let* (x, λ) *denote the state of the inhomogeneous Algorithm 1. The couple* (x, λ) *is strongly ergodic if the annealing schedule* $\{T_k : k = 0, 1, \dots\}$ *satisfies:*

$$T_k \geq \frac{N \Delta_{L^M}}{\log(k+1)}, \tag{21}$$

where:

$$\begin{cases} T_k > T_{k+1} \textit{ and} \lim_{k \to \infty} T_k = 0 \\ \Delta_{L^M} = 2 \max_{(x',\lambda') \in \mathcal{N}(x,\lambda)} \{|L^M(x', \lambda') - L^M(x, \lambda)|\}. \end{cases}$$

Proof See Appendix A. □

3.2.2 Asymptotic convergence of GIMHSA

In this case, the sequence (x_i, λ_i), $i = 1, 2, \dots$, is no longer a Markov Chain, but rather the triplet (x_i, λ_i, y_i), $i = 1, 2, \dots$, is. The GIMHSA defines a Markov Chain with a one-step transition kernel, $P_{GIMHSA}(x, \lambda, y; x', \lambda', y')$, given by:

$$\begin{aligned} P_{GIMHSA}(x, \lambda, y; x', \lambda', y') &= G(x, \lambda; x', \lambda') A_T^M(x, \lambda; x', \lambda') q_{x'}^M(y') \\ &+ \delta_{(x',\lambda',y')}(x, \lambda, y) \left(1 - \int_{\mathcal{N}(x,\lambda)} \int_{\mathcal{Y}^M} A(x, \lambda, y; x', \lambda', y') dy' G(x, \lambda; dx', d\lambda')\right), \end{aligned} \tag{22}$$

where:

$$A(x, \lambda, y; x', \lambda', y') = A_T^M(x, \lambda; x', \lambda') q_{x'}^M(y').$$

Since there exists an integer M so that $P_{GIMHSA}^M(x, \lambda, y; x', \lambda', y') > 0$, for all (x, λ, y), (x', λ', y') the chain is irreducible. The chain is also aperiodic, thus the samples (x_i, λ_i) produced are distributed in the limit as $i \to \infty$ according to the target function $\pi^M(x, \lambda)$. The GIMH can be viewed as a special case of the MCWM sampling strategy, thus the irreducible and aperiodic property of the GIMHSA sampling scheme can easily be deduced from the weak ergodicity property of the MCWMSA algorithm.

4 Numerical illustrations

In this section, we illustrate the efficiency of our algorithms by first conducting an inferential exercize on reliability in stress-strength problems and, second, optimization exercizes for financial portfolio allocation problems.

To begin with, we make explicit all the necessary parameters required in the successful implementation of the SA algorithms. These include the parameters constituting the cooling schedule, i.e. the initial temperature (T_0), the cooling function, the Metropolis loop length (N_T), and the stopping condition. The logarithmic cooling schedule is a well known schedule that guarantees convergence. Nonetheless, this requires infinitely slow cooling. In lieu, most users prefer the geometric cooling proposed by Kirkpatrick et al. (1983), as such:

$$T_k = aT_{k-1}, \quad k = 1, 2, \ldots, M,$$

where T_0 and a, respectively, denote the initial temperature and the cooling rate. Consistent with the literature, the cooling rate, $0.8 < a < 0.99$, which has been shown to be quite effective across many applications (e.g., Stander and Silverman (1994), Wah et al. (2006)) is set to 0.95 in this article. Results of robustness checks are made available in Appendix B. The closer the value of the cooling factor a is to 1, the slower the rate of decay. As for the initial temperature, T_0, we choose a value to ensure that around 60% of initial uphill moves are accepted (as in Rayward-Smith et al. (1996)), or similarly to achieve an acceptance probability of about 0.90 irregardless of the value of the cost function (e.g., Rutenbar (1989), Bohachevsky et al. (1986)). To ensure that the simulated annealing spends sufficient time to thoroughly explore local optimal at each temperature, we fixed a priori the number of iterations, N_T, to $1,000$ (as in Fisk et al. (2022)). The process is terminated at a temperature, T^*, low enough that useful improvements can be expected anymore. Arguably, higher values of T^* would downgrade the precision of the output; on the contrary, smaller values would increase the accuracy of the numerical solution, but at the cost of a larger computing time. On this basis, we fixed the terminating temperature to 10^{-6} (see van Laarhoven et al. (1992) and Fisk et al. (2022)).

4.1 First application: reproduction of inference and reliability in a stress-strength problem

For the sake of a better demonstration of the two new algorithms, we first employ a low dimension optimization problem given by the maximization of an integral constrained likelihood for a stress-strength reliability problem. We consider the inferential exercize on reliability in stress-strength problems with independent Exponentiated Exponential (EE) distributions (as in Gupta and Kundu (2001)) studied by Smith et al. (2015). This problem fits the form of Eq. 1 and reads:

$$\begin{aligned} &\max_{\boldsymbol{\phi}\in\mathcal{X}} \; l(\boldsymbol{\phi}; x, y) \\ &\text{subject to } \mathcal{R} - \psi_0 = 0, \end{aligned} \tag{23}$$

where $l(\cdot)$ is the Log-likelihood, ϕ the parameter vector, x and y are, respectively, outcomes of random variables X and Y, and ψ_0 the target reliability value.
Given two random samples $x = \{x_1, \ldots, x_n\}$ and $y = \{y_1, \ldots, y_m\}$ drawn, respectively, from two independent EE distributions[2], $F_X(\cdot)$ and $F_Y(\cdot)$, $l(\boldsymbol{\phi}; x, y)$ in Eq. 23 is given by:

[2] We denote the EE distribution with $EE(\alpha, \beta)$ where α and β are, respectively, the shape and scale parameters. The respective distribution function, $F_X(\cdot)$, may be written following the definition by Gupta and Kundu (2001) as:

$$F_X(x; \alpha, \beta) = (1 - \exp(-\beta x))^{\alpha}, \quad \alpha > 0, \beta > 0, x > 0.$$

$$
\begin{aligned}
l(\boldsymbol{\phi}; x, y) &= l\left(\alpha_1, \beta_1, \alpha_2, \beta_2; x, y\right) \\
&= n \log \alpha_1 + n \log \beta_1 + (\alpha_1 - 1) \sum_{i=1}^{n} \log\left(1 - e^{-\beta_1 x_i}\right) - \beta_1 \sum_{i=1}^{n} x_i \\
&\quad + m \log \alpha_2 + m \log \beta_2 + (\alpha_2 - 1) \sum_{j=1}^{m} \log\left(1 - e^{-\beta_2 y_j}\right) - \beta_2 \sum_{j=1}^{m} y_j,
\end{aligned}
\tag{24}
$$

and is the Log-likelihood function of the parameter vector (decision variables) $\boldsymbol{\phi} = (\alpha_1, \beta_1, \alpha_2, \beta_2)$ of two independent exponentiated exponential (EE) distributions.

It is important to note that even though this problem is only four-dimensional, it is still a challenging problem. The Log-likelihood, $l(\cdot)$, fails a number of commonly assumed properties of Log-likelihood functions which provide support and guarantee the parameter estimation results obtained when classical optimization techniques are applied. For instance, the Log-likelihood of the exponentiated exponential distribution is neither a concave, nor even a quasi- or pseudo-concave function; there exists a small region where the gradient of $l(\cdot)$ vanishes with a negative definite Hessian matrix; elsewhere, a frequent change in the sign of the determinant of the Hessian matrix is observed (see Smith et al. (2015) for a detailed discussion). These observations therefore suggest that extreme care has to be taken should a derivative-based technique be opted for.

In problem 23, $\mathcal{R}$ denotes the reliability between two EE distributions. Within the reliability framework, the stress-strength model describes the life of a component which has a random strength X that is subjected to a random stress Y. The component is adjudged to fail immediately the stress surpasses the strength, otherwise the component is considered to function properly (i.e. $X > Y$). Therefore, $\mathcal{R} = Pr(Y < X)$ is a suitable measure for quantifying reliability. Applications of this concept can be found in many areas, especially in engineering concepts such as structures, deterioration of rocket motors, static fatigue of ceramic components, fatigue failure of aircraft structures, and the aging of concrete pressure vessels. In our case, given that both stress, Y, and strength, X, are described by independent EE distributions, then the reliability $\mathcal{R}$ of the random variables can be written as (see Kundu and Gupta (2005)):

$$
\mathcal{R} = \beta_1 \alpha_1 \int_0^{\infty} \exp(-\beta_1 x) \left(1 - \exp\left(-\beta_1 x\right)\right)^{\alpha_1 - 1} \left(1 - \exp\left(-\beta_2 x\right)\right)^{\alpha_2} dx. \tag{25}
$$

Except in a few exceptional cases, the integral in Eq. 25 is known not to be analytically reducible (see Nadarajah (2011)). In addition, $\mathcal{R}$ has been shown to be homogeneous of degree 0 in (β_1, β_2), upon the introduction of a change of variable.[3] This observation suggests that the contours of $\mathcal{R}$ are constant along the line in the parameter space satisfying $\beta_2 = c\beta_1$, with c being a real constant.

[3] Smith et al. (2015) consider the change of variable $z = \beta_1 x$ to obtain:

$$
\mathcal{R} = \alpha_1 \int_0^{\infty} \exp(-\beta_1 z) \left(1 - \exp\left(-\beta_1 z\right)\right)^{\alpha_1 - 1} \left(1 - \exp\left(-(\beta_2/\beta_1) z\right)\right)^{\alpha_2} dz,
$$

while Nadarajah (2011) poses the transformation $z = \exp(-\beta_1 x)$ and obtains:

$$
\mathcal{R} = \alpha_1 \int_0^{1} (1 - z)^{\alpha_1 - 1} \left(1 - z^{\beta_2/\beta_1}\right)^{\alpha_2} dz.
$$

Table 1 Dataset#1: Two Simulated Datasets of Size 10, each Drawn from Two Exponentiated Exponential (EE) Distributions with $\psi_0 = 0.1$, such as $EE(2, 3)$ and $EE(5, 1.6015)$

Variables	Observations					Sample Sizes
X	0.4977	0.0781	0.3827	0.2694	0.4125	10
	0.6414	0.2669	0.1978	0.1968	0.2397	
Y	1.7057	1.0191	0.5899	0.9031	0.9207	10
	1.9481	2.1290	0.8109	1.6463	1.9842	

*Source: Table 1, page 4,045, in Smith et al. (2015)

While noting that the probability density function, $f_X(\cdot)$, of the EE distribution, $EE(\alpha_1, \beta_1)$, can be expressed as:

$$f_X(x) = \beta_1\alpha_1 \exp(-\beta_1 x)\left(1 - \exp\left(-\beta_1 x\right)\right)^{\alpha_1 - 1}, \quad \alpha_1 > 0, \beta_1 > 0, x > 0, \tag{26}$$

then the integral in Eq. 25, $\mathcal{R}$, can be written as:

$$\mathcal{R} = E\left[(1 - e^{-\beta_2 X})^{\alpha_2}\right].$$

We sample directly from the distribution:

$$q_\phi(v) = \alpha\beta_1 \exp(-\beta_1 v)\left(1 - \exp\left(-\beta_1 v\right)\right)^{\alpha_1 - 1},$$

with samples ($M = 10,000$ in Eq. 4) drawn by using:

$$V = -\frac{1}{\beta_1}\log\left(1 - F^{1/\alpha_1}\right), \quad F \sim U(0, 1),$$

during the execution of our algorithms.

Smith and Wong (2017) present the difficulties (i.e. numerical accuracy issues, poor performances when the value of ψ_0 is close to the boundary) involved in using the Lagrange multiplier approach to solve the integral constrained problem 1. They also describe an application of simulated algorithms to integral constrained problems. Ours defer from theirs in a number of ways. First, while Smith and Wong (2017) optimized sequentially the penalty function, by gradually increasing the penalty value until convergence is reached, the decision variables as well as the penalty values are simultaneously optimized within our Monte Carlo styled strategy. Our strategy when compared to the sequential optimization procedure adopted by Smith and Wong (2017), has the potential of considerably reducing the computational time required for the optimization exercize. Second, the Smith and Wong (2017) approach is based on optimizing an approximation of the penalty function while we offer two ways of solving the problem, i.e. either by an approximated problem or by an approximation algorithm. We shall see in the following, that even with a small sample, our simulated annealing algorithms perform efficiently and effectively in estimating the parameters of the integral constrained likelihood problem.

Finally, we illustrate the application of our algorithms by using the simulated data generated in Smith et al. (2015) reported in Table 1 below.

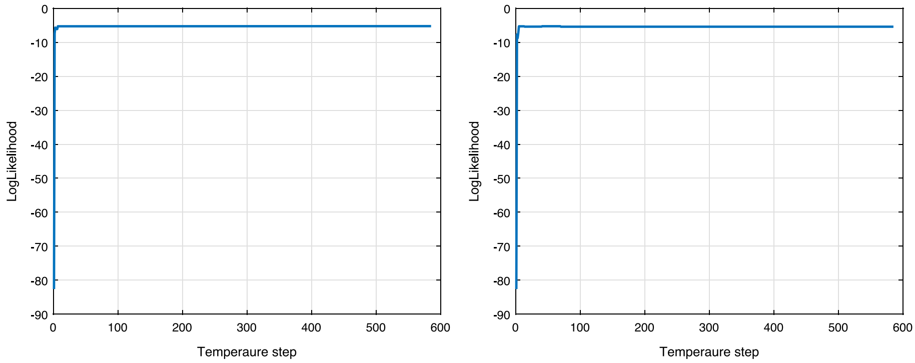


Fig. 1 A Trajectory of the Constrained Log-likelihood Functions obtained from the use of MCWMSA (left figure) and GIMHSA (right figure) Algorithms for solving the Integral Constrained Problem

Table 2 Parameter Estimates for Dataset#1, comparing CSA of Smith et al. (2015), MCWMSA and GIMHSA Algorithms when solving the Integral Constrained Log-likelihood in Eq. 23 using Dataset#1 as reported in Table 1, with $a = 0.95$, $N = 1,000$, $T_0 = 10$ and $T^* = 10^{-6}$

		α_1	β_1	α_2	β_2	$\mathcal{R}$	Log-lik.
CSA[a]		3.6028	5.2707	3.2018	1.5700	0.0989	−5.1659
MCWMSA	Mean	3.4424	5.1485	3.3894	1.6276	0.1000	−5.1984
	Std.	0.3204	0.3604	0.3737	0.1036	0.0001	0.0286
GIMHSA	Mean	3.4308	5.0533	3.6478	1.6691	0.1000	−5.2567
	Std.	0.5324	0.6417	0.8315	0.1713	0.0002	0.0648

[a] Source: The second line in this table comes from Table 2 third line (“Constrained Penalty”), page 4,045 in Smith et al. (2015). Computations by the authors

The data is generated by drawing random samples of size 10 each from the two EE distributions, $EE(2, 3)$ and $EE(5, 1.6015)$, with $\psi_0 = 0.1$ as the corresponding value on reliability[4], $\mathcal{R}$.

Given the above set-up, we executed our stochastic optimization algorithms 10 times each on a penalty function representation of the integral constrained problem 23. Figure 1 displays the convergence towards the optimal Log-likelihood obtained by Smith et al. (2015) of a run, each, of our two algorithms. From this figure there seems to be no difference in the rate of convergence of the algorithms. In contrast with the GIMHSA algorithm, the MCWMSA algorithm requires more time in its execution largely because of the need to compute the penalty function twice at each iteration step (see Eq. 5). In Table 2, we report the parameter estimates obtained by Smith et al. (2015) alongside the statistical results (i.e. mean and standard deviation) obtained from 10 independent runs of our algorithms.

The results in Table 2 indicate that on average the GIMHSA algorithm outperforms other methods as it is the algorithm with the lowest constraint violation (i.e. $\mathcal{R} - 0.1$) estimate.[5]

[4] All results are obtained using MatLab (version R2019a) on a laptop with a Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz and 8GB RAM. The codes used in this article are available upon request from the corresponding author.

[5] In our robustness checks we vary the value of main parameters of the algorithms, namely: a, N and T^*, obtaining similar results in terms of efficiency and computational time (Cf. Appendix B).

Table 3 Reported Characteristics of the Methods in terms of Performances of the Algorithms on the Dataset#1 of Smith et al. (2015)

Methods: (Application#1, Dataset#1)	Elapsed time (sec.)	Number of iterations	Number of variables	Coding complexity
MCWMSA	1,373.30	203,000	15	9,732
GIMHSA	988.82	203,000	14	9,581

Source: Dataset#1 in Tables 1 and 2 of Smith et al. (2015). The algorithm complexity is approached here by the number of variables per iteration i.e. $((1+\eta)(n+I)+m+3)$ for GIMHSA and $((1+\eta)(n+I)+2m+3)$ for both the MCWMSA and GIMHSA (Number of variables) and the number of characters in the codes (Coding complexity)

4.2 Second application: portfolio optimizations in a simple gaussian world

As a second illustration, we implement our algorithms to a set of portfolio optimization problems with positivity constraint on the weights, namely the Markowitz (1952) mean-variance portfolio optimization with positive constraint on the weights.

The Markowitz mean-variance problem with short-selling constraints is taken up in the section to demonstrate the effectiveness of our algorithms. Using this problem as a benchmark, we are able to assess the quality of the solutions of our algorithms, since an optimal portfolio to the Markowitz mean-variance problem can easily be obtained using, for example, Quadratic Programming (QP).

Suppose we have N assets to choose from, let $\mathbf{W}^T = (w_1, \ldots, w_N)$ be the transposed vector of weights with $w_i \in \mathbb{R}$ being the weight of the i-th asset in the portfolio $\mathbf{X}^T = (x_1, \ldots, x_N)$. Also, let $\mathbf{r}^T = (r_1, \ldots, r_N)$ be a transposed real random vector with r_i, $i = 1, \ldots, N$, the return on asset i. Furthermore, we define $\mathbf{r}$ by the following affine transformation:

$$\mathbf{r} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon} + \boldsymbol{\mu}, \tag{27}$$

where $\boldsymbol{\Sigma} \in \mathrm{R}^{N\times N}$ is a non-singular, positive definite matrix (called *precision matrix*), $\boldsymbol{\mu} \in \mathrm{R}^N$ a real vector and $\boldsymbol{\epsilon}$ a real random vector. Following these assumptions, the random variable $R \in \mathbb{R}$ representing the return on the whole portfolio can be expressed as:

$$R = \mathbf{W}^T\mathbf{r}, \tag{28}$$

with the corresponding expected value ($E[R]$) and variance ($var(R)$) given as:

$$\begin{aligned} E[R] &= \mathbf{W}^T(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}E[\boldsymbol{\epsilon}]) \\ &= \mathbf{W}^T\left(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\int \boldsymbol{\epsilon} f(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}\right), \end{aligned} \tag{29}$$

and:

$$\begin{aligned} var(R) &= \mathbf{W}^T(\boldsymbol{\Sigma}^{1/2})^T var(\boldsymbol{\epsilon})\boldsymbol{\Sigma}^{1/2}\mathbf{W} \\ &= \mathbf{W}^T(\boldsymbol{\Sigma}^{1/2})^T\left(\int \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T f(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} - E[\boldsymbol{\epsilon}](E[\boldsymbol{\epsilon}])^T\right)\boldsymbol{\Sigma}^{1/2}\mathbf{W}, \end{aligned} \tag{30}$$

where $f(\cdot)$ denotes the probability density function of $\boldsymbol{\epsilon}$. Under certain distributional assumptions on $\boldsymbol{\epsilon}$, solutions to the integrals in Eq. (29) and (30) can be obtained in closed-forms.

Subject to the above assumptions and given a desired level of expected portfolio returns, R_{exp}, the Makowitz mean-variance portfolio selection problem can be summarized as follows:

$$\min_{\mathbf{W}\in\mathbb{R}^N} \quad \mathbf{W}^T var[\mathbf{r}]\mathbf{W}$$
$$\text{subject to} \begin{cases} \mathbf{W}^T E[\mathbf{r}] = R_{exp} \\ \sum_{i=1}^N w_i = 1 \\ w_i \geq 0, i = 1, \ldots, N, \end{cases} \tag{31}$$

where $\mathbf{W}^T = (w_1, \ldots, w_N)$.

Given the multivariate normal assumption on $\boldsymbol{\epsilon}$ in the simple Gaussian world of this first application, we numerically evaluate the portfolio expected return and variance following a Monte Carlo procedure. Let $\boldsymbol{\epsilon}_i$, $i = 1, 2, \ldots, M$ be a sample of size M drawn from the multivariate normal distribution, $f(\cdot)$, then the integrals in the definition of portfolio expected returns and variance in Eqs. (29) and (30) are evaluated using the empirical counterparts as follows:

$$\int \boldsymbol{\epsilon} f(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} = E[\boldsymbol{\epsilon}] \approxeq \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\epsilon}_i,$$

and:

$$\int \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T f(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} = E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] \approxeq \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^T.$$

The consistency of the Monte Carlo approximation, for $M \longrightarrow \infty$, is guaranteed by the strong law of large numbers (see Robert and Casella (2013), Sect. 3.3, for further details).

We explore some of the properties of the portfolio optimization problem and propose a slight variation of the sampling procedure described in Section 3.1.3 by limiting our algorithms to the consideration of solutions that strictly satisfy the return and the budget constraints. More precisely, we follow the proposal by Crama and Schyns (2003) who state that in problem 31 given a portfolio $\mathbf{W}$, the neighborhood of the current $\mathbf{W}$ contains all solutions $\mathbf{W}^*$ with the following property: there exist three assets, labeled i_1, i_2 and i_3, without loss of generality[6], so that:

$$\begin{cases} w_{i_1}^* = w_{i_1} - S, \\ w_{i_2}^* = w_{i_2} + S\alpha, \\ w_{i_3}^* = w_{i_3} + S(1-\alpha), \\ w_i^* = w_i, \quad \forall\, i \in \{1, \ldots, N\} - \{i_1, i_2, i_3\}, \end{cases} \tag{32}$$

where S is a Cauchy distributed random variable and α a tuning parameter set equal to:

$$\alpha = \frac{r_{i_1} - r_{i_3}}{r_{i_2} - r_{i_3}},$$

where r_{i_1}, r_{i_2} and r_{i_3} corresponds to the returns on the three randomly selected assets whose weights are to be modified.

Furthermore, in the presence of a minimum ($\underline{w}_i$) and a maximum ($\overline{w}_i$) restriction on the proportion of total investment that can be held in asset i ($i = 1, 2, \ldots, N$), then a feasible interval of variation for S to ensure that a move from the current solution W to W^* satisfies the floor and ceiling restrictions can be constructed. Suppose the three assets (say, i_1, i_2 and

[6] We thank the first referee for noticing that the choice of the randomly chosen assets has no importance (since it can be generalized for all triplets of assets, and, more generally, to all sets of assets).

Table 4 Reported Characteristics of the Methods in terms of Differences in Volatility (by Level of Expected Return) versus the QP and Main Reported Performance of the Algorithms for Three Arbitrary Levels of Expected Returns (for each considered market)

Methods:	Obj. Function (Volatility)	Elapsed time (*sec.*)	Number of iterations	Number of variables	Coding complexity
Panel A: US Stock Market (Application#2, Dataset#2)					
Quadratic Prog.	14%	0.03		29	384
MCWMSA	14%	62.95	254,500	94	14,364
GIMHSA	14%	62.95	254,500	94	14,364
Panel B: French Stock Market (Application#2, Dataset#3)					
Quadratic Prog.	17%	0.05		39	384
MCWMSA	17%	58.98	203,000	124	14,364
GIMHSA	17%	58.98	203,000	124	14,364
Panel C: Chinese Stock Market (Application#2, Dataset#4)					
Quadratic Prog.	17%	1.65		40	384
MCWMSA	17%	69.71	203,000	127	14,364
GIMHSA	17%	69.71	203,000	127	14,364

Source. Daily Returns on US, Chinese and French Stock Markets (see Costola et al. (2022)). Are reported here the Markowitz Mean-variance Efficient frontiers differences in terms of Expected Returns obtained using Monte Carlo Within Metropolis Simulated Algorithm (MCWMSA), and Group Independent Metropolis Hastings Simulated Annealing (GIMHSA), versus the benchmark method namely the Quadratic programming (Quadratic Prog.). Elapsed time column represents the total time to reach such a result. The algorithm complexity is approached here by the number of variables per iteration i.e. $((1+\eta)(n+I)+m+3)$ for GIMHSA and $((1+\eta)(n+I)+2m+3)$ for both the MCWMSA and GIMHSA (Number of variables) and the number of characters in the codes (Coding complexity)

i_3) to be modified for a move from W to W^* are known, then we arrive at the following interval of variation for S:

$$\begin{cases} w_{i_1} - \overline{w}_{i_1} < S < w_{i_1} - \underline{w}_{i_1} \\ \underline{w}_{i_2} - w_{i_2} < S\alpha < \overline{w}_{i_2} - w_{i_2} \\ \underline{w}_{i_3} - w_{i_3} < S(1-\alpha) < \overline{w}_{i_3} - w_{i_3}, \end{cases} \tag{33}$$

by combining the ceiling and floor restrictions with Eq. (32).

To empirically assess our algorithms on the mean-variance optimization problem, we next consider three databases (US, Chinese, and French stock markets) covering different economic sectors and used in Costola et al. (2022). The daily prices on US, French, and Chinese Stock Markets covers, respectively, the period 19/3/2008 to 31/12/2019, 6/4/2006 to 4/10/2019 and 18/8/2008 to 31/12/2019. Our algorithm finds the exact optimal risk for all values of the targeted expected returns. The algorithm requires between 58 and 70 seconds per portfolio of between 29 and 40 securities within a standard setting (see Table 4). The efficient frontier for these instances are plotted in Fig. (2).

The steps for generating the efficient frontiers in Fig. 2 are briefly presented in Algorithm#1 in Appendix C, following Calvo et al. (2012).

The solutions obtained with the two algorithms are very close to the ones resulting from QP for the three stock markets considered, as simple eyeball analyses of Figs. 2 and 3 reveal. The difference in the optimal portfolio variance obtained with QP and SA for a given targeted

Fig. 2 Efficient Frontiers in the US, Chinese and French Equity Markets. Note. Are plotted here the Markowitz Mean-variance Efficient frontiers obtained using: Quadratic programming (Quadratic Prog.), Monte Carlo Within Metropolis Simulated Algorithm (MCWMSA), and Group Independent Metropolis Hastings Simulated Annealing (GIMHSA). See Table 4 for the performances of the various algorithms with some precise examples

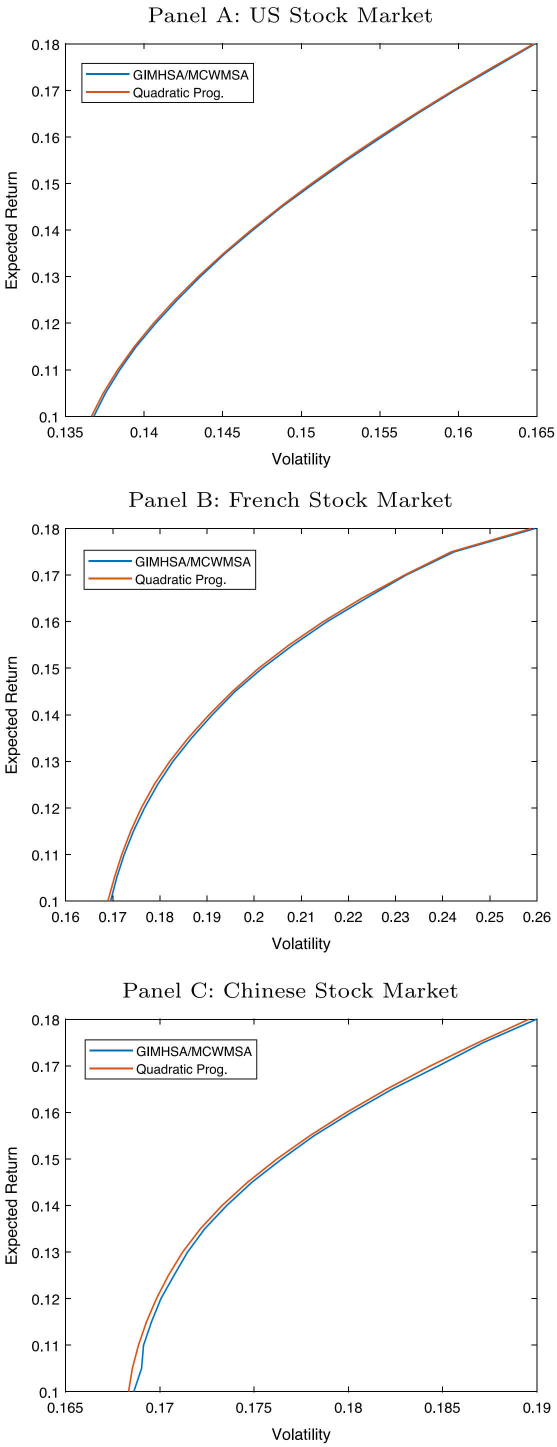
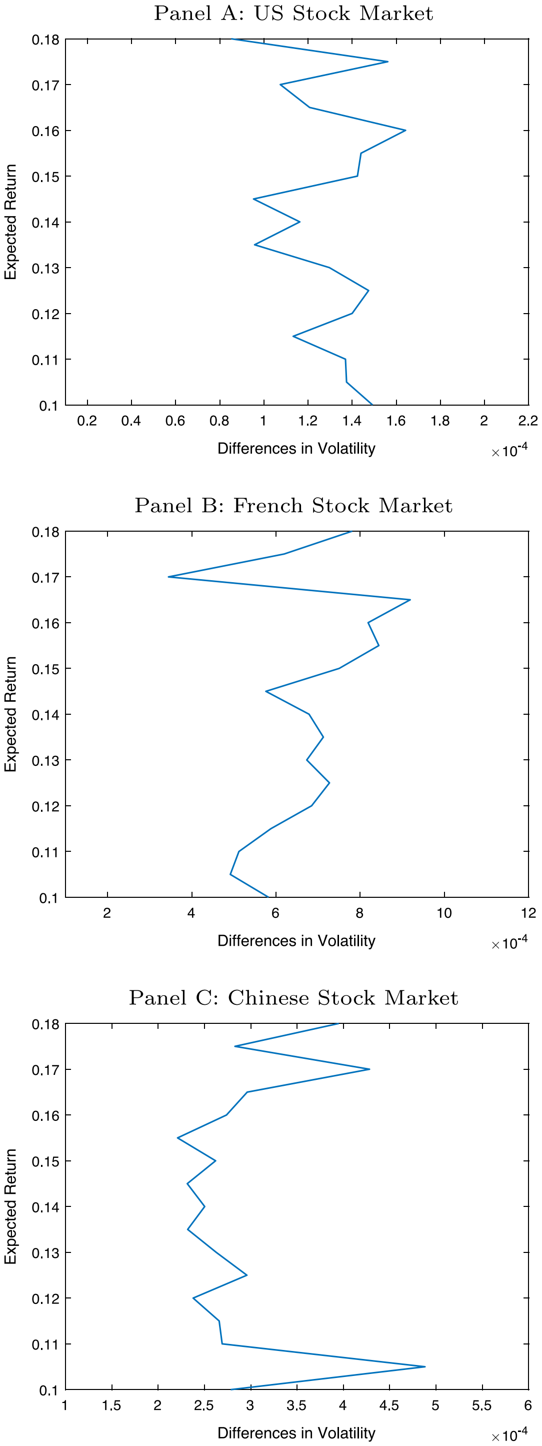


Fig. 3 Differences in Volatilities between GIMHSA/MCWMSA and Quadratic Programming for Each Level of Return and Different Markets



expected return is of the order of 10^{-4} (see Fig. 3), which is negligible for the applications of SA to asset allocation problem.

4.3 Third application: mean-expected shortfall problems in a non-gaussian world

As a third illustration, we implement our algorithms in the case of the Mean-Expected Shortfall problem (as in Casarin and Billio (2007)), when the data are Non-Gaussian and the constraint a true integral, for which few approaches have been developed in the literature to the best of our knowledge. As previously, the return vector $\mathbf{r}^T = (r_1, \ldots, r_N)$ with r_i, $i = 1, \ldots, N$, being the return on asset i, is defined by the affine transformation in Eq. 27 and the corresponding portfolio return, $R \in \mathbb{R}$, is given in Eq. 28.

Considering the Expected Shortfall of the portfolio, $ES_\alpha(R)$, as the measure of risk, the goal is to maximize the $E[R]$ subject to $ES_\alpha(R)$ among several other constraints. This problem fits the form of Eq. 1, so that, for $\alpha \in [0, 1]$:

$$\max_{\mathbf{W}\in\mathbb{R}^N} \mathbf{W}^T E[\mathbf{r}]$$
$$\text{subject to} \begin{cases} ES_\alpha(R) \le \underline{R} \\ \sum_{i=1}^N w_i = 1 \\ w_i \ge 0, \forall i = 1, \ldots, N, \end{cases} \tag{34}$$

where:

$$ES_\alpha(R) = \frac{1}{\alpha}\int_0^\alpha VaR_u(R)du,$$

and $\underline{R}$ a return threshold, and with:

$$\begin{aligned} VaR_u(R) &= \min\{x \in \mathrm{R} : Pr(x + R < 0) \le u\} \\ &= \min\{x \in \mathrm{R} : Pr(-R > x) \le u\} \\ &= \min\{x \in \mathrm{R} : Pr(L \le x) \ge 1 - u\} \\ &= F_L^{-1}(1 - u), \end{aligned}$$

where $F_L(x) = Pr(L \le x)$.

Except for some ad hoc densities, the exact formula for the ES is not available, which often renders the portfolio selection problem a challenging exercize to undertake.

In our case, suppose the vector of returns on the assets satisfies the affine transformation defined by Eq. 27, then the Expected Shortfall on the portfolio return leads to:

$$ES_\alpha(R) = -\mathbf{W}^T\boldsymbol{\mu} + \frac{1}{\alpha}\int_0^\alpha F^{-1}_{-\mathbf{W}^T\Sigma^{1/2}\boldsymbol{\epsilon}}(1 - u)du.$$

For illustrative purposes, we now assume that the vector of asset returns is characterized by a class of Multivariate Skew Student-t distributions which accounts for fat tails and skewness, in the more realistic Non-Gaussian world of this second application. More precisely, we assume that the random vector $\boldsymbol{\epsilon}^T = (\epsilon_1, \ldots, \epsilon_N)$ in Eq. 27 follows a Multivariate Skew Student-t distribution, $Sk\mathcal{T}_N(\boldsymbol{\epsilon}; 0, I_N, \nu, \gamma)$ with $\nu = \{\nu_1, \ldots, \nu_N\}$ the vector of degrees of freedom and $\gamma = \{\gamma_1, \ldots, \gamma_N\}$ the vector of parameters that control the skewness of the distribution. The components ϵ_i of $\boldsymbol{\epsilon}$, $i = 1, 2. \ldots, N$, are independent and Skew Student-t distributed with density:

$$g(\epsilon_i;\nu_i,\gamma_i)=\frac{2}{\gamma_i+1/\gamma_i}\left[h_{\nu_i}\left(\frac{\epsilon_i}{\gamma_i}\right)\mathrm{I}_{[0,+\infty)}(\epsilon_i)+h_{\nu_i}\,(u_i\gamma_i)\,\mathrm{I}_{(-\infty,0]}(\epsilon_i)\right], \tag{35}$$

and:

$$h_{\nu_i}(x)=\frac{\Gamma((\nu_i+1)/2)}{\Gamma(\nu_i/2)\sqrt{(\pi\nu_i)}}\left(1+\frac{x^2}{\nu_i}\right)^{-(\nu_i+1)/2}\mathrm{I}_{(-\infty,\infty)}(x)\ \text{ with }\ \nu_i>2, \tag{36}$$

the density of a univariate Symmetric Student-t distribution. The mean and variance of ϵ_i are respectively given as follows:

$$E[\epsilon_i]=\frac{\gamma_i^2-1}{\gamma_i}\frac{\Gamma((\nu_i-1)/2)}{\Gamma(\nu_i/2)}\left(\frac{\nu_i}{\pi}\right)^{1/2}, \tag{37}$$

and:

$$V(\epsilon_i)=\frac{\gamma_i^4-\gamma_i^2+1}{\gamma_i^2}\left(\frac{\nu_i}{\nu_i-2}-\frac{\nu_i}{\pi}\left(\frac{\Gamma((\nu_i-1)/2)}{\Gamma(\nu_i/2)}\right)^2\right)+\frac{\nu_i}{\pi}\left(\frac{\Gamma((\nu_i-1)/2)}{\Gamma(\nu_i/2)}\right)^2. \tag{38}$$

Subject to the above assumptions, it can easily be shown that **r** follows a Multivariate Skew Student-t distribution, $Sk\mathcal{T}_N(\mathbf{r};\boldsymbol{\mu},\boldsymbol{\Sigma},\nu,\gamma)$, with a Variance-CoVariance of returns matrix given by:

$$V(\mathbf{r})=\boldsymbol{\Sigma}^{1/2}V(\boldsymbol{\epsilon})\boldsymbol{\Sigma}^{1/2}=\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Xi}(\nu,\gamma)\boldsymbol{\Sigma}^{1/2}, \tag{39}$$

where $\boldsymbol{\Xi}(\nu,\gamma)$ is a diagonal matrix parametrized in ν and γ, with elements $\xi_i=V(\epsilon_i)$ on the main diagonal.

To simulate asset returns from a Skew Student-t distribution with ν degrees of freedom and skewness γ, we use the following stochastic representation. Let Z have a Bernoulli distribution with parameter $1/(\nu^2+1)$ and X a Student-t distribution with ν degrees of freedom, then we have:

$$Y=Z|X|\gamma-(1-Z)|X|\frac{1}{\gamma}. \tag{40}$$

To facilitate the implementation of our optimization strategy, we embed the following Monte Carlo evaluation of ES within our optimization scheme as briefly presented here below.

1. Draw $\boldsymbol{\epsilon}_j^T=(\epsilon_{j1},\epsilon_{j2},\ldots,\epsilon_{jN})\sim Sk\mathcal{T}_N(\boldsymbol{\epsilon}_j;0,\mathrm{I}_N,\nu,\gamma)$, for $j=1,\ldots,M$.
2. Compute $-\mathbf{W}^T\boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_j$ for $j=1,2,\ldots,M$.
3. Denote $L_j=-\mathbf{W}^T\boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}_j$. If we order the sample $(L_1,\ldots,L_M)$ so that $L_{1,M}\geq L_{2,M}\geq\cdots\geq L_{M,M}$ and:

$$\begin{aligned}P(L\leq x)&=\int\mathrm{I}_{\{L\leq x\}}f_L(l)dl=E[\mathrm{I}_{\{L\leq x\}}]\\&\approxeq\frac{1}{M}\sum_{j=1}^{M}\mathrm{I}_{\{L_j\leq x\}},\end{aligned}$$

that is:

$$\hat{P}(L\leq x)=\frac{1}{M}\sum_{i=1}^{M}\mathrm{I}_{\{L_{i,M}\leq x\}}.$$

Table 5 Data Characteristics of Dataset#5 and Application#3 (Long-term Mean-Expected Shortfall Optimizations)

Asset Classes	Return	Std..	Correlation Matrix		
			Cash	Stocks	Bonds
Cash	6.8%	2.3%	1	0.01	0.18
Stocks	17.0%	14.7%		1	0.73
Bonds	12.3%	10.5%			1

P^a Source: Casarin and Billio (2007)

4. Evaluate:

$$ES_\alpha(R) = -\mathbf{W}^T\boldsymbol{\mu} + \frac{1}{\alpha}\int_0^\alpha \sum_{i=1}^{M-1}\left(L_{i,M}\mathrm{I}_{(\frac{i-1}{M},\frac{i}{M}]}(u) + L_{M,M}\mathrm{I}_{(\frac{M-1}{M},1]}(u)\right)du$$
$$= -\mathbf{W}^T\boldsymbol{\mu} + \frac{1}{M\alpha}\left(\sum_{i=1}^{\lfloor M\alpha\rfloor} L_{i,M} + (M\alpha - \lfloor M\alpha\rfloor)\, L_{\lfloor M\alpha\rfloor+1,M}\right), \tag{41}$$

where $\lfloor x\rfloor$ and $\mathrm{I}_{(a,b)}$ denote the greatest integer function and the indicator function on the interval (a, b), respectively.

As a first step in solving the portfolio optimization in Eq. 34, we consider a revised problem by exploring the convexity property of Expected Shortfall, i.e.:

$$ES_\alpha(R) \leq \sum_i w_i ES_\alpha(r_i),$$

and solve the following problem:

$$\max_{\mathbf{W}\in\mathbb{R}^N} \ \mathbf{W}^T E[\mathbf{r}]$$
$$\text{subject to} \begin{cases} \sum_i w_i ES_\alpha(r_i) \leq \underline{R} \\ \sum_{i=1}^N w_i = 1 \\ w_i \geq 0, \forall i = 1,\ldots,N. \end{cases} \tag{42}$$

A pseudo-code for the Expected Shortfall efficient frontiers in Fig. 4 is given in Algorithm#1 in Appendix C.

In-view of the above set-up we next illustrate further the effectiveness of our SA type algorithms in two settings. The first exercize studies the effect of heavy tails and skewness of the asset distribution on the ES efficient frontier within a strategic allocation problem. The second one shows the effect of model specification errors on the ES efficient frontier.

The first exercize considers a strategic asset allocation problem similar to the one in Casarin and Billio (2007). Strategic allocation decisions refer to long-term investments and usually involve a small number of assets, often represented as the benchmark of different asset classes. Long-term investors can mitigate the adverse effect arising from their exposure to the increasing volatility of the financial markets by imposing a constraint on their portfolio's downside risk.

Table 5 presents the characteristics of the three asset classes used in Casarin and Billio (2007). Figure 4 displays the Mean-Expected Shortfall Efficient frontiers under different parameter settings of the Skew Student-t distribution.

The mean and scale parameters are the same across frontiers while the skewness and kurtosis vary across the frontiers. Our numerical results show that, for a given expected

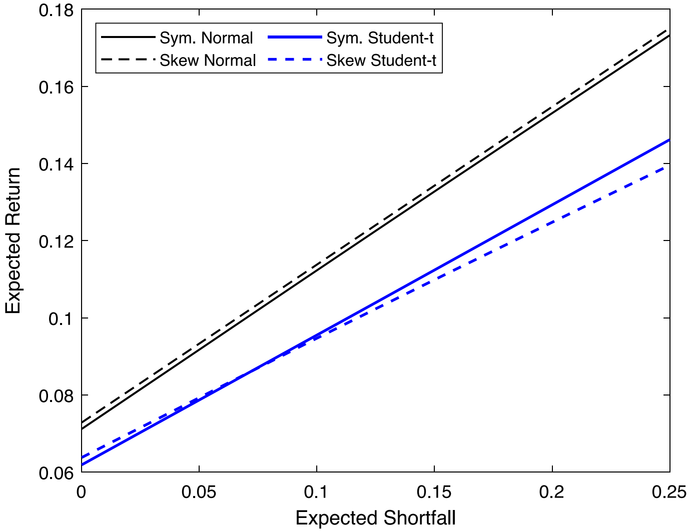


Fig. 4 Mean-Expected Shortfall Efficient Frontiers according to various Distributional Assumptions for the Returns for Dataset#5 and Application#3

Table 6 Data Characteristics of Dataset#6 and Application#3 (Long-term Mean-Expected Shortfall Optimization)

Asset Classes	Mean Returns	Std.	Correlation Matrix		
			Cash	Stocks	Bonds
Cash	−0.66%	0.89%	1	0.02	−0.02
Stocks	4.23%	16.85%		1	0.01
Bonds	1.11%	4.47%			1

[a] Source: see Boucher et al. (2013)

return level, the portfolio risk of normally distributed asset returns is lower than those of heavy-tailed asset returns (see Symmetric Normal and Symmetric Student-t frontiers in Fig. 4). For both normal and heavy tail distributions, the presence of (left) asymmetry in the distribution implies a larger risk when compared to the case of symmetric distribution (see Skew Normal and Skew Student-t frontiers in Fig. 4).

In the second exercize dedicated to Mean-Expected Shortfall optimization, we consider the asset classes used in the long-term global asset allocation with model risk studied in Boucher et al. (2013). The asset classes are represented by cash, stock and bond markets.[7] The main characteristics in terms of means and variance-covariances are presented in Table 6.

We first assume the investor is using a non-correctly specified model (a Normal distribution) for the returns, plugged into the ES optimization problem. Then, we evaluate the actual (realized) expected return and risk using the best flexible model (that is a Skew Student-t dis-

[7] We follow Boucher et al. (2013) to build up our database: the “Equity” asset class is represented by a composite index of 95% of the MSCI Europe and 5% of the MSCI World; the “Bonds” asset class comes from the x-trackers II iBoxx Sovereigns Eurozone AAA UCITS 1C index (denoted as XBAT index); the “Money Market” is represented by the Euro Overnight Index Average (EONIA).

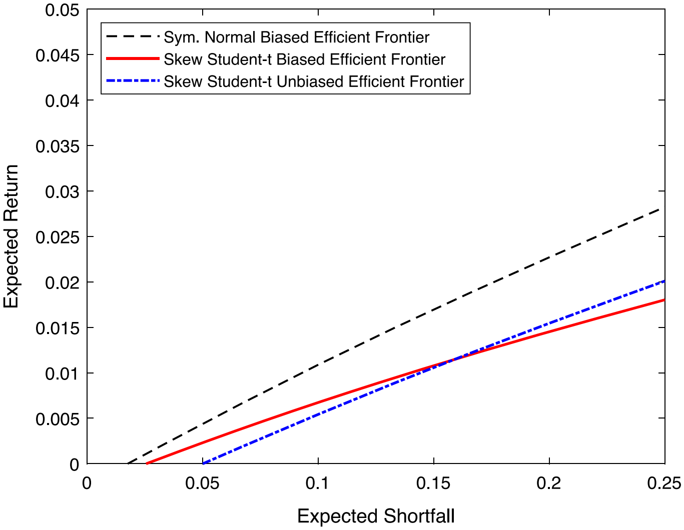


Fig. 5 Mean-Expected Shortfall Efficient Frontiers according to various Estimation Error Assumptions for the Returns for Dataset#6 and Application#3. Note. Are plotted here three Mean-ES efficient frontiers. The two first ones are obtained by using the incorrect return distribution (Symmetric Normal mean and ES in dashed black: "Sym. Normal Biased Efficient Frontier"), and the correctly estimated distribution (Skew Student-t mean and ES in dotted-dashed blue: "Skew Student-t Unbiased Efficient Frontier"). The third one is obtained using the optimal weights corresponding to the incorrect allocations (Symmetric Normal) all together with the mean and ES under the Skew Student-t assumption (solid red line: "Skew Student-t Biased Efficient Frontier"), where degrees of freedom and skewness parameters are estimated on the data

tribution) fitted on the real market data on the entire sample. A pseudo-code for the Expected Shortfall efficient frontiers with estimation errors (Fig. 5) is provided in Algorithm#1 in Appendix C.

Finally, we compare the two ex post biased frontiers with the optimal one obtained from the problem 42 with the best flexible model. In Fig. 5, the Expected Return and ES in the Normal case differ from the Skew Student-t since the Normal distribution assumption for the entire sample period is too restrictive and suffers from an estimation bias. As expected the Symmetric Normal frontier is above the two Skew Student-t frontiers (because of the better fitting of asymmetry and tails provided by the Skew Student-t). We shall also note that asymmetry and tail estimates are crucial for a correct evaluation of the frontier since the differences of the two Skew Student-t frontiers are small, while the differences between the Symmetric Normal and the Skew Student-t ones are large.

In this case (Dataset#6 and Application#3), as in all our previous exercizes (Dataset#1-#5 and Application#1-#3), the proposed MCWSA and GIMHSA algorithms reach convergence and are able to efficiently provide optimal solutions within a reasonable and realistic computational time for a practitioner.

5 Conclusion

The problem of designing efficient algorithms for optimizing a non-differentiable function subject to an integral constraint whose solution is not available in a closed-form is an important one, in particular in Operational Research and more specifically in Finance. In this article, we propose two new stochastic optimization algorithms referred to as Monte Carlo within Metropolis Simulated Annealing (MCWMSA) and Group Independent Metropolis Hastings Simulated Annealing (GIMHSA), which both rely on the use of importance sampling. The latter employed Monte Carlo simulations in approximating the analytically intractable integral constrained problem, before using a Simulated Annealing-type algorithm, while the former used the Monte Carlo strategy in constructing an approximate algorithm. We were first able to prove some convergence properties of the two algorithms. We show that the sequence of samples generated by MCWMSA and GIMHSA is an ergodic Markov Chain process and the limiting distribution is the true target function.

We study first the performance of our algorithms by applying them to the problem of maximizing the integral constrained likelihood for the stress-strength reliability with independent exponentiated exponential distributions, as discussed in Smith et al. (2015). Our results confirm the effectiveness of our algorithms in solving the integral constrained problem. It also happens that GIMHSA gives the best results in the sense that it produces the lowest constraint violations.

As a complement, we secondly provide examples of potential uses in finance. A first simple benchmark show that they provide accurate approximations of the true solutions in the case of the classical portfolio optimization problem of Markowitz (1952), in which the constrained integral simply concerns the constrained expected return. We further confirm this basic result by numerically solving the positively constrained Expected Return / Expected Shortfall strategic asset allocation problems (as in Casarin and Billio (2007)). In the this case, efficient results are obtained. We also complement this later result using the dataset used in Boucher et al. (2013) for the sake of robustness and show the impact of some potential estimation errors on the estimated frontiers. Our results finally show that the two proposed algorithms quickly converge to practical optimal solutions, within a decent and operational computational time for risk and asset managers.

We can imagine several extensions of this work. On the algorithmic added-value of the article, we may think about parallel GIMSA with and without interactions. Also, application of the proposed algorithms to mixed-integer programming problems, such as the portfolio selection problem with or without a cardinality constraint (see Woodside-Oriakhi et al. (2011)) will be consider in further studies.

Regarding financial applications, competing with existing numerical approximations for VaR and ES when Cumulative Density Function analytical expressions are unknown, will be a decent challenge (see, for instance, Azzalini and Valle (1996) Skew Normal distribution). To complement this, the recently proposed joint approach for estimating VaR and ES (see Wang et al. (2018); Meng and Taylor (2020)) might be a target of interest for testing further our proposed improved approaches. Finally, tackling the full constrained problem of portfolio optimization in a dynamic setting would be beneficial for investors (see Costola et al. (2022)).

Acknowledgements We thank Jean-Luc Prigent for positive comments and encouragement on the preliminary draft of this article, as well as participants to the FEM2021 conference (Paris, on-line, June 2021), as well as the Editor in charge and two anonymous referees who significantly contributed to the improvement of the article. Roberto Casarin and Anthony Osuntuyi gratefully acknowledge the funding received from the EURO Horizon 2020 project on Energy Efficient Mortgages Action Plan (EeMAP), grant agreement No 746205 and the VERA project on economic and financial cycles under the grant number No. 479/2018 Prot. 33287 -VII / 16 of 08.06.2018. The usual disclaimers apply.

Appendix A proof to theorem 1: ergodicity of MCWMSA

The proof for Theorem 1 follows standard arguments for the convergence of Inhomogeneous Markov Chains (e.g., Isaacson and Madsen (1976) and Seneta (1981) for related results). More precisely, we first form a lower bound on the probability of reaching any solution from any local minimum, and then show that this bound does not approach zero too quickly. Let G be the generation probability that satisfies 19.

a. Let:

$$\Delta_G = \min_{\substack{(x',\lambda')\in\mathcal{N}(x,\lambda),\\ x,\lambda\in\mathcal{S}}} G(x,\lambda;x',\lambda')$$

and:

$$\Delta_{L^M} = 2 \max_{\substack{(x',\lambda')\in\mathcal{N}(x,\lambda),\\ (x'\lambda')\in\mathcal{S},y,y'\in\mathcal{Y}^M}} \{|L^M(x',\lambda') - L^M(x,\lambda)|\},$$

be, respectively, the value of the smallest one-step solution generation probabilities and the size of the largest potential jump in the value of the auxiliary penalty function between any pair of trial solutions. $\mathcal{S} = \mathcal{X} \times \mathbb{R}^r_+$ denotes the search space while $\mathcal{N}(x,\lambda)$ represents the neighborhood of (x,λ). If $(x,\lambda) \neq (x',\lambda')$, then for all $(x,\lambda) \in \mathcal{S}$ and $(x',\lambda') \in \mathcal{N}(x,\lambda)$, we have:

$$\begin{aligned}
P_{MCWMSA}(x,\lambda;x',\lambda') &= \int_{\mathcal{Y}^M}\int_{\mathcal{Y}^M} G(x,\lambda;x',\lambda')A^M_{T_k}(x,\lambda;x',\lambda')q^M_x(y)q^M_{x'}(y')dydy' \\
&= G(x,\lambda;x',\lambda')\int_{\mathcal{Y}^M}\int_{\mathcal{Y}^M} A^M_{T_k}(x,\lambda;x',\lambda')q^M_x(y)q^M_{x'}(y')dydy' \\
&= \begin{cases} G(x,\lambda;x',\lambda')\int_{\mathcal{Y}^M}\int_{\mathcal{Y}^M} \min\left\{\exp\left(-\frac{(L^M(x',\lambda')-L^M(x,\lambda))}{T}\right),1\right\} q^M_x(y)q^M_{x'}(y')dydy', & \text{if } u=1 \\ G(x,\lambda;x',\lambda')\int_{\mathcal{Y}^M}\int_{\mathcal{Y}^M} \min\left\{\exp\left(-\frac{(L^M(x,\lambda)-L^M(x',\lambda'))}{T}\right),1\right\} q^M_x(y)q^M_{x'}(y')dydy', & \text{if } u=0 \end{cases} \\
&\geq G(x,\lambda;x',\lambda')\exp\left(-\frac{\Delta_{L^M}}{T_k}\right)\int_{\mathcal{Y}^M}\int_{\mathcal{Y}^M} q^M_x(y)q^M_{x'}(y')dydy' \\
&\geq \Delta_G \exp\left(-\frac{\Delta_{L^M}}{T_k}\right).
\end{aligned} \tag{A.1}$$

b. For any λ, let:

$$\Delta^M_{min} = \min_{\substack{(x',\lambda')\in\mathcal{N}(x,\lambda),\\ A^M_{T_k}<1,y,y'\in\mathcal{Y}^M}} \left(L^M(x',\lambda') - L^M(x,\lambda)\right) \geq 0,$$

be the size of the smallest positive jump in the value of the objective function between any solution pair.

If $(x', \lambda') = (x, \lambda)$, we have:

$$\begin{aligned}
&P_{MCWMSA}(x, \lambda; x', \lambda') \\
&\quad = 1 - \int_{\mathcal{N}(x,\lambda)} \int_{\mathcal{Y}^M} \int_{\mathcal{Y}^M} A^M_{T_k}(x, \lambda; x', \lambda') q^M_x(y) q^M_{x'}(y') dy dy' G(x, \lambda; dx', d\lambda') \\
&\quad = \int_{\mathcal{N}(x,\lambda), A^M_{T_k} < 1} \int_{\mathcal{Y}^M} \int_{\mathcal{Y}^M} \left(1 - A^M_{T_k}(x, \lambda; x', \lambda')\right) q^M_x(y) q^M_{x'}(y') dy dy' G(x, \lambda; dx', d\lambda') \\
&\quad \geq \int_{\mathcal{N}(x,\lambda), A^M_{T_k} < 1} \int_{\mathcal{Y}^M} \int_{\mathcal{Y}^M} \Delta_G (1 - \exp(-\Delta^M_{min}/T_k)) q^M_x(y) q^M_{x'}(y') dy dy' dx' d\lambda'. \qquad \text{(A.2)}
\end{aligned}$$

Since, T_k is a decreasing sequence, it is always possible to find a $k_0 > 0$ so that for all $k \geq k_0$, $(1 - \exp(-\Delta^M_{min}/T_k)) \geq \exp(-\Delta_{L^M}/T_k)$.
Thus:

$$P_{MCWMSA}(x, \lambda; x, \lambda) \geq \exp(-\Delta_{L^M}/T_k).$$

c. Based on the result obtained above, for all $(x, \lambda), (x', \lambda') \in \mathcal{S}$ and $k \geq k_0$, the N-step transition probability from $(x, \lambda) = (x_0, \lambda_0)$ to $(x', \lambda') = (x_N, \lambda_N)$ satisfies the following:

$$P^N_{MCWMSA}(x, \lambda; x', \lambda') = \prod_{i=0}^{N-1} P_{MCWMSA}(x_i, \lambda_i; x_{i+1}, \lambda_{i+1}) \geq \left(\Delta_G \exp(-\Delta_{L^M}/T_k)\right)^N .$$

Let $\tau_1(P)$ denote the coefficient of ergodicity for the transition probability matrix P, defined by:

$$\tau_1\left(P^N_{MCWMSA}\right) = 1 - \min_{(x,\lambda),(x',\lambda') \in \mathcal{S}} \sum_{(x'',\lambda'') \in \mathcal{S}} d(x, \lambda, x', \lambda', x'', \lambda'')$$

where $d(x, \lambda, x', \lambda', x'', \lambda'') = \min\left\{P^N_{MCWMSA}(x, \lambda; x'', \lambda''), P^N_{MCWMSA}(x', \lambda'; x'', \lambda'')\right\}$
An Inhomogeneous Markov Chain is weakly ergodic if and only if there is a strictly increasing sequence of positive numbers so that:

$$\sum_{k=0} \left(1 - \tau_1(P^N_{MCWMSA})\right) = \infty, \qquad \text{(A.3)}$$

using the bound on $P^N_{MCWMSA}(x, \lambda; x', \lambda')$, we find that:

$$\begin{aligned}
1 - \tau_1(P^N_{MCWMSA}) &= \min_{(x,\lambda),(x',\lambda') \in \mathcal{S}} \sum_{(x'',\lambda'') \in \mathcal{S}} d(x, \lambda, x', \lambda', x'', \lambda'') \\
&\geq \min_{(x,\lambda),(x',\lambda') \in \mathcal{S}} \min_{(x'',\lambda'') \in \mathcal{S}} \left\{P^N_{MCWMSA}(x, \lambda; x'', \lambda''), P^N_{MCWMSA}(x', \lambda'; x'', \lambda'')\right\} \\
&\geq \left(\Delta_G \exp(-\Delta_{L^M}/T_k)\right)^N = \Delta^N_G \exp(-N\Delta_{L^M}/T_k). \qquad \text{(A.4)}
\end{aligned}$$

Substituting $T_k \geq \dfrac{N\Delta_{L^M}}{\log(k+1)}$, $k \geq 0$ in Eq. A.3 gives:

$$\begin{aligned}
\sum_{k=0} \left(1 - \tau_1(P^N_{MCWMSA})\right) &\geq \sum_{k=k_0} \Delta^N_G \exp(-N\Delta_{L^M}/T_k) \\
&\geq \Delta^N_G \sum_{k=k_0} \frac{1}{k+1} = \infty.
\end{aligned} \qquad \text{(A.5)}$$

This implies that the Markov Chain, $(x, \lambda, x', \lambda')$, is **weakly ergodic**.

d. Furthermore, since the acceptance probability function $A^{M}_{T_K}$ is an exponential rational function in $1/T_k$ and belongs to the closed class of asymptotically monotone functions (Anily and Federgruen (1987), Proposition 1) then the sequence of (x_n, λ_n) converges in distribution (Anily and Federgruen (1987), Corollary 1). This implies that the Markov Chain is **strongly ergodic**. □

Appendix B robustness checks

This section provides the robustness checks of the results in Tables 2 and 3 with respect the main parameters of the algorithms, namely a, N and T^* (boldface values), showing similar results in terms of efficiency, computational time and GIMHSA dominance.

Table 7 Parameter Estimates for Dataset#1, comparing CSA of Smith et al. (2015), MCWMSA and GIMHSA Algorithms when solving the Integral Constrained Log-likelihood in Eq. 23 using Dataset#1 as reported in Table 1, with $a = \mathbf{0.80}$, $N = \mathbf{250}$, $T_0 = 10$ and $T^* = 10^{-6}$

		α_1	β_1	α_2	β_2	$\mathcal{R}$	Log-lik.
SA Constrained with penalty[a]		3.6028	5.2707	3.2018	1.5700	0.0989	−5.1659
MCWMSA	Mean	3.5234	5.0515	4.1315	1.7729	0.0999	−5.4657
	Std.	0.9969	1.0508	1.6336	0.3428	0.0005	0.3189
GIMHSA	Mean	3.3610	4.7619	4.3054	1.6386	0.0997	−6.6136
	Std.	1.6862	1.8144	2.3100	0.4620	0.0004	2.7570

[a] Source: The second line in this table comes from Table 2 third line ("Constrained Penalty"), page 4,045 in Smith et al. (2015). Computations by the authors

Table 8 Reported Characteristics of the Methods in terms of Performances of the Algorithms on the Dataset#1 of Smith et al. (2015) with $a = \mathbf{0.80}$, $N = \mathbf{250}$, $T_0 = 10$ and $T^* = 10^{-6}$

Methods: (Application#1, Dataset#1)	Elapsed time (sec.)	Number of iterations	Number of variables	Coding complexity
MCWMSA	53.6175	18,500	2	256
GIMHSA	40.1336	18,500	2	256

Source: Dataset#1 in Tables 1 and 2 of Smith et al. (2015). The coding complexity is approached her by the number of characters of the programs leading to final results

Table 9 Parameter Estimates for Dataset#1, comparing CSA of Smith et al. (2015), MCWMSA and GIMHSA Algorithms when solving the Integral Constrained Log-likelihood in Eq. 23 using Dataset#1 as reported in Table 1, with $a = 0.95$, $N = \mathbf{250}$, $T_0 = 10$ and $T^* = \mathbf{10^{-4}}$

		α_1	β_1	α_2	β_2	$\mathcal{R}$	Log-lik.
CSA with penalty[a]		3.6028	5.2707	3.2018	1.5700	0.0989	−5.1659
MCWMSA	Mean	3.5015	5.0927	3.7010	1.7165	0.1000	−5.2647
	Std.	0.5040	0.5041	0.6256	0.1594	0.0001	0.0964
GIMHSA	Mean	3.7916	5.3802	3.6220	1.6974	0.1000	−5.3448
	Std.	0.5905	0.8248	0.8069	0.2008	0.0002	0.1212

[a] Source: The second line in this table comes from Table 2 third line ("Constrained Penalty"), page 4,045 in Smith et al. (2015). Computations by the authors

Table 10 Reported Characteristics of the Methods in terms of Performances of the Algorithms on the Dataset#1 of Smith et al. (2015) with $a = 0.95$, $N = \mathbf{500}$, $T_0 = 10$ and $T^* = \mathbf{10^{-4}}$

Methods: (Application# 1, Dataset#1# 1)	Elapsed time (sec.)	Number of iterations	Number of variables	Coding complexity
MCWMSA	229.1966	113,000	2	256
GIMHSA	169.4235	113,000	2	256

Source: Dataset#1 in Tables 1 and 2 of Smith et al. (2015). The coding complexity is approached here by the number of characters of the programs leading to final results

Table 11 Parameter Estimates for Dataset#1, comparing CSA of Smith et al. (2015), MCWMSA and GIMHSA Algorithms when solving the Integral Constrained Log-likelihood in Eq. 23 using Dataset#1 as reported in Table 1, with $a = 0.95$, $N = \mathbf{250}$, $T_0 = 10$ and $T^* = \mathbf{10^{-4}}$

		α_1	β_1	α_2	β_2	$\mathcal{R}$	Log-lik.
CSA with penalty[a]		3.6028	5.2707	3.2018	1.5700	0.0989	−5.1659
MCWMSA	Mean	3.6726	5.5130	3.4265	1.5122	0.1000	−7.1298
	Std.	0.7621	1.1537	1.7526	0.6076	0.0004	5.2893
GIMHSA	Mean	3.4124	5.1838	3.8654	1.5924	0.1001	−6.1172
	Std.	1.0364	1.4316	1.8698	0.5675	0.0003	1.9353

[a] Source: The second line in this table comes from Table 2 third line ("Constrained Penalty"), page 4,045 in Smith et al. (2015). Computations by the authors

Table 12 Reported Characteristics of the Methods in terms of Performances of the Algorithms on the Dataset#1 of Smith et al. (2015) for $a = 0.95$, $N = \mathbf{250}$, $T_0 = 10$ and $T^* = \mathbf{10^{-4}}$

Methods: (Application#1, Dataset#1)	Elapsed time (sec.)	Number of iterations	Number of variables	Coding complexity
MCWMSA	92.3520	56,500	2	256
GIMHSA	67.9437	56,500	2	256

sparaSource: Dataset#1 in Table 1 and of Smith et al. (2015). The coding complexity is approached here by the number of characters of the programs leading to final results

Appendix C Pseudo-codes

```
input : Set starting point $z = (x, \lambda)$ and initialize $\lambda = 0$.
input : Set starting $T = T_0$ and cooling rate $0 < \alpha < 1$.
input : Set $N$ - the number of trials per temperature.
while stopping condition is not satisfied do
    for $\ell = 1 \to N$ do
        Generate $u \sim \mathcal{B}\text{er}(q)$
        if $u = 1$ then
            Generate trial points $(x^*, \lambda^*) \in \mathcal{N}(x, \lambda)$
            from $G(x, \lambda; x^*, \lambda^*) = H(x, x^*)\delta_\lambda(\lambda^*)$ (See Eq. 7).
            Draw importance samples $y_1, y_2, \ldots, y_M$ from $q_x$ and $y_1^*, y_2^*, \ldots, y_M^*$
              from $q_{x^*}$.
            Compute:
            $w^M(x, \lambda; x^*, \lambda^*) = \pi^M(x, \lambda)G(x, \lambda; x^*, \lambda^*)$,
            $w^M(x^*, \lambda^*; x, \lambda) = \pi^M(x^*, \lambda^*)G(x^*, \lambda^*; x, \lambda)$,
            and acceptance probability $A_T^M(x, \lambda; , x^*, \lambda^*)$.
        else
            Generate trial points $(x^*, \lambda^*) \in \mathcal{N}(x, \lambda)$
            from $G(x, \lambda; x^*, \lambda^*) = \tilde{H}(\lambda, \lambda^*)\delta_x(x^*)$(See Eq. 7).
            Draw $y_1^*, y_2^*, \ldots, y_M^*$ from $q_{x^*}$.
            Compute:
            $w^M(x, \lambda; x^*, \lambda^*) = \pi^M(x, \lambda)^{-1}G(x, \lambda; x^*, \lambda^*)$,
            $w^M(x^*, \lambda^*; x, \lambda) = \pi^M(x^*, \lambda^*)^{-1}G(x^*, \lambda^*; x, \lambda)$,
            and acceptance probability $A_T^M(x, \lambda; , x^*, \lambda^*)$.
        end
        Generate $(rand \sim \mathcal{U}(0, 1))$.
        if $A_T^M(x, \lambda; , x^*, \lambda^*) > rand$ then
            accept the trial solution $(x^*, \lambda^*)$.
        end
    end
    Reduce temperature by $T = aT$.
end
```

Algorithm 1: Pseudo-code for the MCWMSA

```
input : Set starting point $z = (x, \lambda)$ and initialize $\lambda = 0$.
input : Set starting $T = T_0$ and cooling rate $0 < \alpha < 1$.
input : Set $N_T$ as the number of trials per temperature.
while stopping condition is not satisfied do
    for $k = 1 \to N_T$ do
        Generate $u \sim \mathcal{B}\mathrm{er}(q)$.
        if $u = 1$ then
            Generate trial points $(x^*, \lambda^*) \in \mathcal{N}(x, \lambda)$
            from $G(x, \lambda; x^*, \lambda^*) = H(x, x^*)\delta_\lambda(\lambda^*)$.
            Draw $y_1^*, y_2^*, \ldots, y_M^*$ from $q_{x^*}$.
            Compute:
            $w^M(x, \lambda; x^*, \lambda^*) = \pi^M(x, \lambda)G(x, \lambda; x^*, \lambda^*)$,
            $w^M(x^*, \lambda^*; x, \lambda) = \pi^M(x^*, \lambda^*)G(x^*, \lambda^*; x, \lambda)$,
            and acceptance probability $A_T^M(x, \lambda; , x^*, \lambda^*)$.
        else
            Generate trial points $(x^*, \lambda^*) \in \mathcal{N}(x, \lambda)$
            from $G(x, \lambda; x^*, \lambda^*) = \tilde{H}(\lambda, \lambda^*)\delta_x(x^*)$.
            Draw $y_1^*, y_2^*, \ldots, y_M^*$ from $q_{x^*}$.
            Compute:
            $w^M(x, \lambda; x^*, \lambda^*) = \pi^M(x, \lambda)^{-1}G(x, \lambda; x^*, \lambda^*)$,
            $w^M(x^*, \lambda^*; x, \lambda) = \pi^M(x^*, \lambda^*)^{-1}G(x^*, \lambda^*; x, \lambda)$,
            and acceptance probability $A_T^M(x, \lambda; , x^*, \lambda^*)$.
        end
        Generate $(rand \sim \mathcal{U}(0, 1))$.
        if $A_T^M(x, \lambda; , x^*, \lambda^*) > rand$ then
            accept the trial solution $(x^*, \lambda^*)$ and the auxiliary variables
              $y_1^*, y_2^*, \ldots, y_M^*$.
        end
    end
    Reduce temperature by $T = aT$.
end
```

Algorithm 2: Pseudo-code for the GIMHSA

```
input : Create an initial grid of targeted expected return $R_{exp}$, say:
        $(R_{exp,1}, \ldots, R_{exp,K})$.
input : Compute $var(\mathbf{r})$ and $E[\mathbf{r}]$ using Eq. 29 and 30.
for $k = 1 \to K$ do
    For each $R_{exp,k}$ compute the minimum variance portfolio weights $\mathbf{W}_k$ by
      solving Eq. 31 and compute:
        $$\sigma_k = \sqrt{\mathbf{W}_k^T var[\mathbf{r}]\mathbf{W}_k}.$$
end
Interpolate the pairs $(\sigma_k, R_{exp,k})$ by any standard interpolation technique to obtain
  a continuous curve.
```

Algorithm 3: Pseudo-code for the Markowitz Efficient Frontiers in Fig. 2

```
input : Create an initial grid of targeted expected return $\underline{R}$, say: $(\underline{R}_1, \dots, \underline{R}_K)$.
input : Compute $E[\mathbf{r}]$ and $ES_\alpha(\mathbf{r}) = (ES_\alpha(r_1), \dots, ES_\alpha(r_N)$ using Eq. 30 and 41.
for $k = 1 \to K$ do
    For each $\underline{R}_k$ compute the maximum expected return portfolio weights $\mathbf{W}_k$ by
      solving Eq. 42 and compute:
          $\mu_k = \mathbf{W}_k^T E[\mathbf{r}],$
      and:
          $ES_\alpha(\underline{R}_k) = \mathbf{W}_k^T ES_\alpha(\mathbf{r}).$
end
Interpolate the pairs $(ES_\alpha(\underline{R}_k), \mu_k)$ by any standard interpolation technique to
  obtain a continuous curve.
```

Algorithm 4: Pseudo-code for the Expected Shortfall Efficient Frontiers for Fig. 4

```
input : Create an initial grid of targeted expected return $\underline{R}$, say: $(\underline{R}_1, \dots, \underline{R}_K)$.
input : Compute $E[\mathbf{r}]$ and $ES_\alpha(\mathbf{r}) = (ES_\alpha(r_1), \dots, ES_\alpha(r_N))$ using Eq. 30 and
        41 under the Symmetric Normal assumption.
input : Compute $E^*[\mathbf{r}]$ and $ES^*_\alpha(\mathbf{r}) = (ES^*_\alpha(r_1), \dots, ES^*_\alpha(r_N))$ using Eq. 30 and
        41 under the Skew Student-t assumption with the best estimated degrees
        of freedom and asymmetry parameters.
for $k = 1 \to K$ do
    For each $\underline{R}_k$, compute the maximum expected return portfolio weights $\mathbf{W}_k$ by
      solving Eq. 42 with $[\mathbf{r}]$ and $ES_\alpha(\mathbf{r})$ and compute the biased expected return
      and risk:
          $\mu^*_k = \mathbf{W}_k^T E^*[\mathbf{r}],$
      and:
          $ES^*_\alpha(\underline{R}_k) = \mathbf{W}_k^T ES^*_\alpha(\mathbf{r}).$
end
Interpolate the biased pairs $(ES^*_\alpha(\underline{R}_k), \mu^*_k)$ by any standard interpolation technique
  to obtain a continuous curve.
```

Algorithm 5: Pseudo-code for the Biased Expected Shortfall Efficient Frontiers for Fig. 5

References

Andrieu, C., & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics, 37*(2), 697–725. https://doi.org/10.1214/07-aos574

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing, 18*(4), 343–373.

Anily, S., & Federgruen, A. (1987). Ergodicity in parametric nonstationary Markov chains: An application to simulated annealing methods. *Operations Research, 35*(6), 867–874. https://doi.org/10.1287/opre.35.6.867

Azzalini, A., & Valle, A.D. (1996). The multivariate skew-normal distribution. Biometrika 83(4):715–726, http://www.jstor.org/stable/2337278.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics, 164*(3), 1139–1160. https://doi.org/10.1093/genetics/164.3.1139

Bergey, P. K., Ragsdale, C. T., & Hoskote, M. (2003). A simulated annealing genetic algorithm for the electrical power districting problem. *Annals of Operations Research, 121*(1), 33–55. https://doi.org/10.1023/A:1023347000978

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Cambridge: Academic press.

Bick, B., Kraft, H., & Munk, C. (2013). Solving constrained consumption-investment problems by simulation of artificial market strategies. *Management Science, 59*(2), 485–503. https://doi.org/10.1287/mnsc.1120.1623

Bohachevsky, I. O., Johnson, M. E., & Stein, M. L. (1986). Generalized simulated annealing for function optimization. *Technometrics, 28*(3), 209–217. https://doi.org/10.1080/00401706.1986.10488128

Boucher, C., Jannin, G., Kouontchou, P., & Maillet, B. (2013). An economic evaluation of model risk in long-term asset allocations. *Review of International Economics, 21*(3), 475–491. https://doi.org/10.1111/roie.12049

Calvo, C., Ivorra, C., & Liern, V. (2012). On the computation of the efficient frontier of the portfolio selection problem. *Journal of Applied Mathematics, 2012*, 1–25. https://doi.org/10.1155/2012/105616

Casarin, R., & Billio, M. (2007). Stochastic optimization for allocation problems with shortfall risk constraints. *Applied Stochastic Models in Business and Industry, 23*(3), 247–271. https://doi.org/10.1002/asmb.671

Castellano, R., Cerqueti, R., Clemente, G. P., & Grassi, R. (2021). An optimization model for minimizing systemic risk. *Mathematics and Financial Economics, 15*(1), 103–129. https://doi.org/10.1007/s11579-020-00279-6

Charalambous, C. (1978). A lower bound for the controlling parameters of the exact penalty functions. *Mathematical programming, 15*(1), 278–290. https://doi.org/10.1007/BF01609033

Costola, M., Maillet, B., & Yuan, Z. (2022). Mean–variance efficient large portfolios: A simple machine learning heuristic technique based on the two-fund separation theorem. *Annals of Operations Research*. https://doi.org/10.1007/s10479-022-04881-3

Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research, 150*(3), 546–571. https://doi.org/10.1016/S0377-2217(02)00784-1

Cramer, J.S. (1989). Econometric applications of maximum likelihood methods. CUP Archive.

D'Amico, S. J., Wang, S. J., Batta, R., & Rump, C. M. (2002). A simulated annealing approach to police district design. *Computers and Operations Research, 29*(6), 667–684. https://doi.org/10.1016/S0305-0548(01)00056-9

Dong, M. X., & Wets, R. J. B. (2000). Estimating density functions: A constrained maximum likelihood approach. *Journal of Nonparametric Statistics, 12*(4), 549–595. https://doi.org/10.1080/10485250008832822

Edrisi, A., Nadi, A., & Askari, M. (2020). Optimization-based planning to assess the level of disaster in the city. *International Journal of Human Capital in Urban Management, 5*(3), 199–206. https://doi.org/10.22034/IJHCUM.2020.03.02

Etgar, R., Shtub, A., & LeBlanc, L. J. (1997). Scheduling projects to maximize net present value - the case of time-dependent, contingent cash flows. *European Journal of Operational Research, 96*(1), 90–96. https://doi.org/10.1016/0377-2217(95)00382-7

Etgar, R., Gelbard, R., & Cohen, Y. (2017). Optimizing version release dates of research and development long-term processes. *European Journal of Operational Research, 259*(2), 642–653. https://doi.org/10.1016/j.ejor.2016.10.029

Feng, M. B., Maggiar, A., Staum, J., & Wachter, A. (2018). Uniform convergence of sample average approximation with adaptive importance sampling. In 2018 Winter Simulation Conference (WSC). *IEEE*. https://doi.org/10.1109/wsc.2018.8632370

Fisk, C. L., Harring, J. R., Shen, Z., Leite, W., Suen, K. Y., & Marcoulides, K. M. (2022). Using simulated annealing to investigate sensitivity of SEM to external model misspecification. *Educational and Psychological Measurement*. https://doi.org/10.1177/00131644211073121

Ftiti, Z., Tissaoui, K., & Boubaker, S. (2020). On the relationship between oil and gas markets: A new forecasting framework based on a machine learning approach. *Annals of Operations Research, 313*(2), 915–943. https://doi.org/10.1007/s10479-020-03652-2

Gilli, M., & Schumann, E. (2012). Heuristic optimisation in financial modelling. *Annals of Operations Research, 193*, 129–158. https://doi.org/10.1007/s10479-011-0862-y

Glizer, V.Y., & Turetsky, V. (2019). Robust optimization of statistical process control. In: 2019 27th Mediterranean Conference on Control and Automation (MED), IEEE, https://doi.org/10.1109/med.2019.8798539.

Gupta, R. D., & Kundu, D. (2001). Exponentiated exponential family: An alternative to gamma and Weibull distributions. *Biometrical Journal: Journal of Mathematical Methods in Biosciences, 43*(1), 117–130. https://doi.org/10.1002/1521-4036(200102)43:1<117::AID-BIMJ117>3.0.CO;2-R

He, Z., Wang, N., Jia, T., & Xu, Y. (2009). Simulated annealing and tabu search for multi-mode project payment scheduling. *European Journal of Operational Research, 198*(3), 688–696.

Homem-de-Mello, T. (2008). On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling. *SIAM Journal on Optimization, 19*(2), 524–551. https://doi.org/10.1137/060657418

Isaacson, D. L., & Madsen, R. W. (1976). *Markov chains Theory and Applications*. New York: Wiley.

Juditsky, A., & Nemirovski, A. (2020). Statistical Inference via Convex Optimization. *Princeton University Press*. https://doi.org/10.2307/j.ctvqsdxqd

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Kundu, D., & Gupta, R. D. (2005). Estimation of $P[Y<X]$ for generalized exponential distribution. *Metrika, 61*(3), 291–308. https://doi.org/10.1007/s001840400345

van Laarhoven, P. J. M., Aarts, E. H. L., & Lenstra, J. K. (1992). Job shop scheduling by simulated annealing. *Operations Research, 40*(1), 113–125. https://doi.org/10.1287/opre.40.1.113

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

de Mello, T. H., & Bayraksan, G. (2014). Monte Carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science, 19*(1), 56–85. https://doi.org/10.1016/j.sorms.2014.05.001

Meng, X., & Taylor, J. W. (2020). Estimating value-at-risk and expected shortfall using the intraday low and range data. *European Journal of Operational Research, 280*(1), 191–202. https://doi.org/10.1016/j.ejor.2019.07.011

Nadarajah, S. (2011). The exponentiated exponential distribution: A survey. *AStA Advances in Statistical Analysis, 95*(3), 219–251. https://doi.org/10.1007/s10182-011-0154-5

Porth, L., Boyd, M., & Pai, J. (2016). Reducing risk through pooling and selective reinsurance using simulated annealing: An example from crop insurance. *The Geneva Risk and Insurance Review, 41*(2), 163–191. https://doi.org/10.1057/s10713-016-0013-0

Rayward-Smith, V., Osman, I., Reeves, C., & Simth, G. (1996). *Modern Heuristic Search Methods*. New York: Wiley.

Ricca, F., & Simeone, B. (2008). Local search algorithms for political districting. *European Journal of Operational Research, 189*(3), 1409–1426. https://doi.org/10.1016/j.ejor.2006.08.065

Robert, C., & Casella, G. (2013). *Monte Carlo statistical methods*. Berlin: Springer.

Rutenbar, R. (1989). Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine, 5*(1), 19–26. https://doi.org/10.1109/101.17235

Ryu, D. W., Kim, J. I., Suh, S., & Suh, W. (2015). Evaluating risks using simulated annealing and building information modeling. *Applied Mathematical Modelling, 39*(19), 5925–5935. https://doi.org/10.1016/j.apm.2015.04.024

Seneta, E. (1981). *Non-negative Matrices and Markov Chains*. New York: Springer. https://doi.org/10.1007/0-387-32792-4

Shajin, D., Jacob, J., & Krishnamoorthy, A. (2021). On a queueing inventory problem with necessary and optional inventories. *Annals of Operations Research*. https://doi.org/10.1007/s10479-021-03975-8

Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2014). *Lectures on stochastic programming: Modeling and theory*. US: SIAM.

Smith, B., & Wong, A. (2017). Simulated annealing of constrained statistical functions. In: Computational Optimization in Engineering - Paradigms and Applications, InTech, https://doi.org/10.5772/66069.

Smith, B., Wang, S., Wong, A., & Zhou, X. (2015). A penalized likelihood approach to parameter estimation with integral reliability constraints. *Entropy, 17*(6), 4040–4063. https://doi.org/10.3390/e17064040

Stander, J., & Silverman, B. W. (1994). Temperature schedules for simulated annealing. *Statistics and Computing, 4*(1), 21–32. https://doi.org/10.1007/bf00143921

Tayal, A., & Singh, S. (2018). Integrating big data analytic and hybrid firefly-chaotic simulated annealing approach for facility layout problem. *Annals of Operations Research, 270*(3), 489–514. https://doi.org/10.1007/s10479-016-2237-x

Wah, B. W., Chen, Y., & Wang, T. (2006). Simulated annealing with asymptotic convergence for nonlinear constrained optimization. *Journal of Global Optimization, 39*(1), 1–37. https://doi.org/10.1007/s10898-006-9107-z

Wang, H., Hu, Z., Sun, Y., Su, Q., & Xia, X. (2018). Modified backtracking search optimization algorithm inspired by simulated annealing for constrained engineering optimization problems. *Computational Intelligence and Neuroscience, 2018*, 1–27. https://doi.org/10.1155/2018/9167414

Woodside-Oriakhi, M., Lucas, C., & Beasley, J. (2011). Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research, 213*(3), 538–550. https://doi.org/10.1016/j.ejor.2011.03.030

Zhang, C., Lin, Q., Gao, L., & Li, X. (2015). Backtracking search algorithm with three constraint handling methods for constrained optimization problems. *Expert Systems with Applications, 42*(21), 7831–7845. https://doi.org/10.1016/j.eswa.2015.05.050

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.