# A Gradual Non-Convexification Method for Minimizing VaR

Jiong Xi * Thomas F. Coleman † Yuying Li ‡ Aditya Tayal §

August 7, 2013

**Abstract**

Given a finite set of $m$ scenarios, computing a portfolio with the minimium Value-at-Risk (VaR) is computationally difficult: the portfolio VaR function is non-convex, non-smooth, and has many local minima. Instead of formulating an $n$-asset optimal VaR portfolio problem as minimizing a loss quantile function to determine the asset holding vector $\mathbb{R}^n$, we consider it as a minimization problem in an augmented space $\mathbb{R}^n$, with a linear objective function under a probability constraint. We then propose a new gradual non-convexification penalty method, aiming to reach a global minimum of nonconvex minimization under the probability constraint. A continuously differentiable piecewise quadratic function is used to approximate step functions, the sum of which defines the probabilistic constraint. In an attempt to reach the global minimizer, we solve a sequence of minimization problems indexed by a parameter $\rho_k > 0$ where $-\rho_k$ is the minimum curvature for the probability constraint approximation. As the indexing parameter increases, the approximation function for the probabilistic inequality constraint becomes more non-convex. Furthermore, the solution of the $k$-th optimization problem is used as the starting point of the $(k+1)$-th problem. Our new method has three advantages. Firstly, it is structurally simple. Secondly, it is efficient since each function evaluation requires $O(m)$ work. Thirdly, a gradual non-convexification process is designed to track the global minimum. Both historical and synthetic data are used to illustrate the efficacy of the proposed VaR minimization method. We compare our method with Gaivoronski and Pflug's quantile-based smoothed VaR method in terms of VaR, CPU time, and efficient frontiers. We show that our gradual non-convexification penalty method yields better minimal VaR portfolio. We show that the proposed gradual non-convexification penalty method is computationally much more efficient, especially when the number of scenarios is large.

**Keywords:** VaR, CVaR, portfolio optimization, local minimum, Gradual non-convexification

*David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: j2xi@uwaterloo.ca

†Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: tfcoleman@uwaterloo.ca.

‡David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: yuying@uwaterloo.ca

§David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1, email: amtayal@uwaterloo.ca

# 1 Introduction

Portfolio allocation is a fundamental problem in finance. Markowitz [9] provided the first systematic treatment of the problem, by measuring risk using the standard deviation of asset returns. In the Markowitz portfolio, allocation consists of finding optimal asset weights that minimize the variance of the portfolio return for a given expected return and budget constraint. This turns out to be a straightforward convex quadratic programming problem, which can be solved to obtain globally optimal weights. However, the choice of using variance as the risk measure is largely due to convenience, rather than theoretical or practical justification. In particular, one major drawback is its symmetrical treatment of both upside and downside deviations from the mean—which does not accurately reflect the risk preferences of an investor—especially for skewed or non-normal distributions.

Value-at-Risk (VaR) has emerged as a widely used risk measure to quantitatively aggregate and measure risk across diverse asset types. In the early 1990s, VaR was adopted by J.P Morgan to report firmwide risk. The methodology later became part of the RiskMetrics$^{TM}$ system in 1995. VaR expresses risk in a simple and intuitive way [e.g., see 5]. The $\beta$-quantile of the loss distribution is the Value-at-Risk. For example, 95% VaR is the portfolio loss which is exceeded with 5% probability. In contrast to variance, VaR measures portfolio loss or downside risk, and is appropriate for skewed or asymmetric probability distributions. It provides a common risk measure that can be used across different types of portfolios and has been accepted as a standard for measuring financial risk [e.g., see 6].

Other risk measures have also been proposed. For instance, Conditional Value-at-Risk (CVaR), closely related to VaR, measures the expected loss of a portfolio in the worst $100(1-\beta)$ percent of the cases. CVaR accounts for the full shape of the tail distribution. In addition, the notion of a coherent risk measure has been developed axiomatically [e.g. see 1]. A risk measure is coherent if it satisfies the axioms of translation invariance, subadditivity, positive homogeneity, and monotonicity. While CVaR is a coherent risk measure, VaR is not, since it lacks the subadditivity property (i.e. using VaR, the risk of two portfolios together can be greater than the two risks separately). This is contrary to the diversification principle and consequently has attracted much criticism in using VaR as a risk measure. Indeed, there has been continual debate among both academics and practitioners regarding which risk measure is better. In spite of its shortcomings, VaR continues to be widely used in practice and is required by regulation (i.e. refer to *Amendment to the Capital Accord to Incorporate Market Risks*, 1996; Basel II). Estimating the distribution of losses beyond the VaR point is difficult, and thus VaR can provide a more robust risk measure than CVaR. Additionally, Cont et al. [4] illustrate a conflict between the subadditivity property and robustness of risk measure, and argue in favor of VaR over CVaR in this respect.

Leaving the choice of a risk measure aside, the focus of this paper is on portfolio allocation using VaR as a risk preference. While VaR has traditionally been used for ex post measurement of risk, recently there has been interest in using the measure ex ante for optimal portfolio selection. As Gaivoronski and Pflug [7] point out, for an investor with risk preference expressed in terms of VaR, allocation using other risk preferences, such as variance or even CVaR, can be poor substitutes.

However, adopting VaR as a risk preference in portfolio allocation leads to a difficult optimization problem compared to mean-variance or CVaR optimization. Specifically, the lack of subadditivity results in a non-convex minimization problem, which has many local minima. In addition, under a finite number of scenarios, the objective function is generally non-smooth. Uryasev

and Rockafellar [11] propose an efficient algorithm to minimize CVaR of a portfolio using a linear programming formulation. Since CVaR is an upper bound for VaR, they argue that this algorithm also yields a low VaR. Subsequently, Larsen et al. [8] propose two heuristic algorithms for VaR optimization problem by taking advantage of the efficient CVaR optimization method. The main idea is to start with the minimal CVaR portfolio, using the linear programming approach of Uryasev and Rockafellar [11], and systematically reduce VaR by solving a series of CVaR optimization problems, which are obtained by constraining and discarding scenarios that correspond to large losses. Gaivoronski and Pflug [7] formulate VaR as a quantile of the portfolio loss and use a quantile-based smoothed VaR function to replace the original VaR function in the VaR minimization problem. This smoothed VaR function filters out irregularity of the original VaR function. A smoothing parameter, $\epsilon$, is used to control how accurate the approximation is. However, it is difficult to choose an appropriate value for $\epsilon$, since it can depend on the dataset. Moreover, the approach is computationally expensive. The objective function requires enumerating all possible combinations in the tail distribution; consequently, the number of arithmetic operations necessary for a straightforward evaluation grows exponentially with the number of scenarios, $m$. A more efficient approach to evaluate the objective is proposed by the authors, but still requires $O(m^3)$ work in addition to evaluating each loss function of the portfolio. More recently, Wozabal et al. [12] propose a method to compute an approximate solution of the VaR optimization problem. The computational time of this algorithm also becomes prohibitively long for large data sets. Consequently there is need for a computationally efficient method that can produce high quality solutions for minimizing VaR.

The main objective of this paper is to propose a new gradual non-convexification penalty (GNCP) method for the VaR minimization problem. This gradual non-convexification method is an adaptation of a technique used in image processing, see, e.g., [2]. Instead of using the notion of VaR as a quantile, we introduce an auxiliary variable $\alpha$ and use the definition of $\text{VaR}_\beta$ as the minimum $\alpha$ such that the probability of loss greater than $\alpha$ does not exceed $1-\beta$. The proposed optimization formulation has a linear objective function with a probabilistic constraint which involves a sum of step functions. We use a piecewise quadratic smooth function to approximate the step function and formulate the VaR minimization problem as an optimization problem with a nonlinear constraint. The proposed optimization formulation is conceptually simpler than the quantile-based smoothed VaR method [7], avoiding combinatorial considerations, and results in a computational method which is significantly more efficient. The proposed method requires $O(m)$ work for function evaluation instead of $O(m^3)$ work, as in Gaivoronski and Pflug [7]. Furthermore, the proposed method employs a gradual non-convexification process in an attempt to track the global minimizer. This process is indexed by a parameter $\rho$, where $-\rho$ represents the most negative curvature of the probability constraint approximation. As $\rho$ is gradually increased, more negative curvature is introduced in the probability constraint approximation. The VaR minimization solution is computed by solving a sequence of nonlinearly constrained (approximation to the probability constraint) optimization problems using an exact penalty method. In the process, the solution avoids the difficulty of choosing a single smoothing parameter.

The presentation is organized as follows. In §2, we present the Mean-VaR optimization problem and discuss the computational difficulty in solving this problem. The quantile-based smoothed VaR method of Gaivoronski and Pflug [7] will also be reviewed. In §3, we develop the gradual non-convexification penalty method formulation for portfolio VaR minimization. In §4, we compare our proposed method with the quantile-based smoothed VaR method in terms of the VaR value, CPU time and efficient frontiers. Both historical and synthetic datasets are used in the computational

investigations. We conclude the paper in §5.

## 2 Mean-VaR Optimization Problem

Given a finite set of assets, $i = 1, 2, \ldots, n$, we want to determine the percentage holding for each asset such that the portfolio meets objectives in return and risk. Assuming that the return distribution is jointly normal, the classical Markowitz Mean-Variance optimal portfolio problem can be expressed as

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \mathbf{x}^T Q \mathbf{x} \\ \text{s.t.} & \mathbf{x}^T \mathrm{E}(\mathbf{r}) \geq R \\ & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n \end{array} \tag{2.1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$ is the position of each asset in the portfolio, $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T \in \mathbb{R}^n$ is a random vector of asset returns, and $Q \in \mathbb{R}^{n \times n}$ is the covariance matrix. Thus we minimize the variance of the portfolio, $\mathbf{x}^T Q \mathbf{x}$, while ensuring the expected return of the portfolio, $\mathrm{E}(\mathbf{x}^T \mathbf{r}) = \mathbf{x}^T \mathrm{E}(\mathbf{r})$, meets some target level $R$. The constraint $x_i \geq 0$ means that short selling is not allowed. For many portfolio optimization problems, the return distribution is not normal, possibly due to a short time horizon or inclusion of nonlinear instruments. In these situations, we are mainly interested in minimizing downside risk. In this paper we consider using Value-at-Risk (VaR) as an alternative risk measure.

VaR provides an intuitive way to measure the risk of a portfolio. Given a confidence level $\beta$, $\beta \in (0, 1)$, $\mathrm{VaR}_\beta$ of a portfolio with random loss $L$ is the loss value such that, with $\beta$ confidence, the portfolio loss will not exceed this value. Let $p_L(l)$ be the probability density function $p_L(l)$ for a random loss $L$. Then VaR of the loss $L$ can be expressed as:

$$\mathrm{VaR}_\beta(L) = \min\big\{\alpha \in \mathbb{R} : \int_{l \leq \alpha} p_L(l)\, \mathrm{d}l \geq \beta\big\}. \tag{2.2}$$

Here we consider negative portfolio return as portfolio loss, i.e. $L = f(\mathbf{x}) = -\mathbf{x}^T \mathbf{r}$. Thus, replacing variance with VaR in the Mean-Variance optimization model (2.1), we have the following Mean-VaR optimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \mathrm{VaR}_\beta(-\mathbf{x}^T \mathbf{r}) \\ \text{s.t.} & \mathrm{E}(\mathbf{x}^T \mathbf{r}) \geq R \\ & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n. \end{array} \tag{2.3}$$

In contrast to the Mean-Variance optimization problem (2.1), Mean-VaR optimization problem (2.3) is difficult to solve. When the distribution is given by a finite set of samples, the objective function is non-convex, non-smooth and has many local minima. We note that additional linear constraints can be incorporated in (2.3) without increasing the complexity of the optimization problem. Thus, in order to focus on the main challenge which is introduced by VaR as a risk measure, we remove

the first constraint in problem (2.3) and consider (for simplicity in presentation):

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathrm{VaR}_\beta(-\mathbf{x}^T\mathbf{r}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n. \end{aligned} \tag{2.4}$$

We illustrate difficulties in solving VaR minimization problem (2.4) with a simple example. Using 1,000 daily returns of two stocks, AA and AXP, in the Dow Jones stock index from September 4th, 1991 to August 17th, 1995, we consider the problem of determining a two-asset portfolio which minimizes $\mathrm{VaR}_{0.95}$. The weight of each asset in the portfolio is written as $\mathbf{x} = (x_1, x_2)^T = (w, 1-w)^T$, $w \in [0, 1]$.

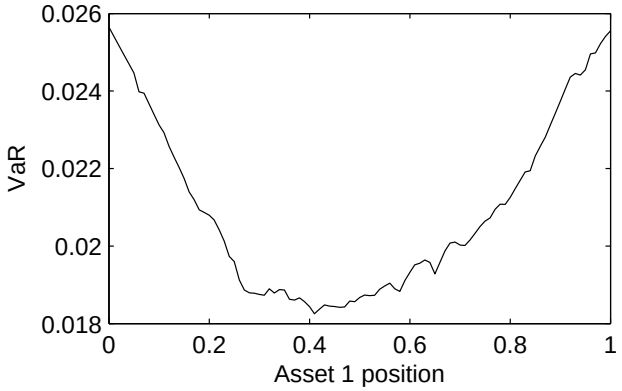



FIGURE 2.1: $\mathrm{VaR}_{0.95}$ *with respect to asset 1 position for a two-asset portfolio*

Figure 2.1 graphs the $\mathrm{VaR}_{0.95}$ of the portfolio with respect to $w$, illustrating the VaR function is non-convex and non-smooth. Thus it is difficult to find a globally optimal set of weights; specifically, there is no known algorithm which runs in polynomial time and can guarantee a global minimizer (i.e. NP hard). In practice, VaR portfolio optimization problem can include thousands of assets with more than tens of thousands of scenarios. Consequently it is imperative for a computational method to scale well with both the number of assets and number of scenarios, while able to find a high quality solution.

Recently, Gaivoronski and Pflug [7] propose a quantile-based smoothed VaR method to tackle problem (2.4). They view the VaR function as a superposition of two components: a global component that represents the macrostructure of the VaR function, and a local component that is responsible for the irregularity of the VaR function. The key idea of their approach is to extract the global component of the VaR function and filter out the local component.

Since the new algorithm we propose shares some similarity with the quantile-based smoothed VaR approach in [7], we briefly discuss it here for motivation and comparison purposes. Following notations in [7], denote:

- $f_i(\mathbf{x}) = -\mathbf{x}^T\mathbf{r}^{(i)}$: the $i$-th sample loss of the portfolio
- $M^i$: the set of all integers from 1 to $m$ excluding $i$, i.e., $M^i = \{1, 2, \ldots, m\} \backslash \{i\}$

- $\Lambda_k^i$: an arbitrary subset of $M^i$ which contains exactly $k$ elements
- $X(\Lambda_k^i)$: a subset of $\mathbb{R}^n$ associated with the set $\Lambda_k^i$ such that

$$X(\Lambda_k^i) = \{\mathbf{x} : f_i(\mathbf{x}) \leq f_j(\mathbf{x}) \text{ for } j \in \Lambda_k^i, f_i(\mathbf{x}) \geq f_j(\mathbf{x}) \text{ for } j \in M^i \backslash \Lambda_k^i\}$$

- $\Theta_i^k$: the family of all different sets $\Lambda_k^i$

When a finite set of scenarios are given, $\text{VaR}_\beta$ can be expressed as a $\beta$-quantile of the portfolio loss, which can be written as a linear combination of all the portfolio losses:

$$\text{VaR}_\beta = G(m - \lfloor \beta m \rfloor, \mathbf{x}), \tag{2.5}$$

where $\lfloor \cdot \rfloor$ is the floor operator and

$$G(k, \mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{i=1}^{m} c_i(\mathbf{x}) f_i(\mathbf{x}) \tag{2.6}$$

$$C(\mathbf{x}) = \sum_{i=1}^{m} c_i(\mathbf{x}) \tag{2.7}$$

$$c_i(\mathbf{x}) = \sum_{\Lambda_k^i \in \Theta_k^i} \mathbb{I}_{X(\Lambda_k^i)}(\mathbf{x}) \tag{2.8}$$

$$\mathbb{I}_{X(\Lambda_k^i)}(\mathbf{x}) = \prod_{j \in \Lambda_k^i} \mathbb{I}_-\big(f_i(\mathbf{x}) - f_j(\mathbf{x})\big) \prod_{j \in M^i \backslash \Lambda_k^i} \mathbb{I}_-\big(f_j(\mathbf{x}) - f_i(\mathbf{x})\big) \tag{2.9}$$

$$\mathbb{I}_-(z) = \begin{cases} 1 & \text{if } z \leq 0 \\ 0 & \text{if } z > 0 \end{cases}. \tag{2.10}$$

The indicator function $\mathbb{I}_{X(\Lambda_k^i)}(\mathbf{x})$ in equation (2.9) is equal to 1 only if the index set $\Lambda_k^i$ is chosen such that all the losses corresponding to the indices in the set are no less than $f_i(\mathbf{x})$ and all the losses corresponding to the indices not in the set are no greater than $f_i(\mathbf{x})$. This means $f_i(\mathbf{x})$ is the $(k+1)$-th largest loss among all the losses. Hence $c_i(\mathbf{x}) > 0$ holds only when the $i$-th loss $f_i(\mathbf{x})$ is $\text{VaR}_\beta$ of the portfolio.

The step indicator function (2.10) is not continuous at $z = 0$. [7] introduce a twice continuously differentiable function $\varphi_\epsilon(z)$ below to approximate (2.10):

$$\varphi_\epsilon(z) = \begin{cases} 1 & \text{if } z \leq 0 \\ 1 - \frac{16}{3\epsilon^3} z^3 & \text{if } 0 \leq z \leq \frac{\epsilon}{4} \\ \frac{5}{6} + \frac{2}{\epsilon} z - \frac{8}{\epsilon^2} z^2 + \frac{16}{3\epsilon^3} z^3 & \text{if } \frac{\epsilon}{4} \leq z \leq \frac{3\epsilon}{4} \\ \frac{16}{3} - \frac{16}{\epsilon} z + \frac{16}{\epsilon^2} z^2 - \frac{16}{3\epsilon^3} z^3 & \text{if } \frac{3\epsilon}{4} \leq z \leq \epsilon \\ 0 & \text{if } z \geq \epsilon\,, \end{cases} \tag{2.11}$$

where $\epsilon$ controls the amount of smoothing. Then the $\epsilon$-approximate function $G_\epsilon(k, \mathbf{x})$ is:

$$G_\epsilon(k, \mathbf{x}) = \frac{1}{C_\epsilon(\mathbf{x})} \sum_{i=1}^{m} c_i^\epsilon(\mathbf{x}) f_i(\mathbf{x}), \tag{2.12}$$

where

$$C_\epsilon(\mathbf{x}) = \sum_{i=1}^{m} c_i^\epsilon(\mathbf{x}), \tag{2.13}$$

$$c_i^\epsilon(\mathbf{x}) = \sum_{\Lambda_k^i \in \Theta_k^i} \prod_{j \in \Lambda_k^i} \varphi_\epsilon\big(f_i(\mathbf{x}) - f_j(\mathbf{x})\big) \prod_{j \in M^i \setminus \Lambda_k^i} \varphi_\epsilon\big(f_j(\mathbf{x}) - f_i(\mathbf{x})\big), \tag{2.14}$$

is twice continuously differentiable for all $\epsilon > 0$, assuming $\{f_i(\mathbf{x})\}$ are twice continuously differentiable. In addition, $G_\epsilon(k, \mathbf{x}) \to G(k, \mathbf{x})$ as $\epsilon \to 0$ for any fixed $\mathbf{x}$.

Replacing $\text{VaR}_\beta$ with the quantile-based smoothed VaR function, $G_\epsilon(m - \lfloor \beta m \rfloor, \mathbf{x})$, in equation (2.4), it yields the following minimization problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & G_\epsilon(m - \lfloor \beta m \rfloor, \mathbf{x}) \\ \text{s.t.} & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n. \end{array} \tag{2.15}$$

This smooth nonconvex minimization problem can be solved by a nonlinear programming solver. For example, [7] solve the approximate optimization problem (2.15) using the **fmincon** subroutine from MATLAB Optimization Toolbox.

The quantile-based smoothed VaR minimization problem (2.15) is a reasonable approximation method for the original VaR minimization problem (2.4). However, this method is computationally expensive. From equation (2.12)–(2.14), it takes exponential time to evaluate the function $G_\epsilon(k, \mathbf{x})$ since it needs to enumerate all the possible combinations of choosing $k$ numbers from $m$ indices. Gaivoronski and Pflug [7] show that, when a suitable parameter $\epsilon$ is chosen for the smooth approximation function in (2.11), the computational time complexity to evaluate the approximation function $G_\epsilon(k, \mathbf{x})$ can be reduced to $O(m^3)$ where $m$ is the number of return samples. In general, since a nonlinear optimization algorithm is iterative, it is necessary to evaluate $G_\epsilon(k, \mathbf{x})$ many times during the optimization procedure. This is costly, especially when the number of scenarios, $m$, is large. Furthermore, an effective nonlinear optimization algorithm typically requires computation of the first and preferably the second order derivatives of the objective and constraint functions. Although the smoothed VaR function $G_\epsilon(k, \mathbf{x})$ is twice continuously differentiable, the analytic form of the gradient and Hessian are complicated, and computation would be very costly as the function evaluation already requires $O(m^3)$ work. In addition, it is difficult to select an appropiate value for the smoothing parameter $\epsilon$, since it can depend on the particular dataset. This will be illustrated using computational examples in §4. As the size of the problem increases, choosing an appropriate parameter becomes even more difficult.

# 3 A Gradual Non-Convexification Method for VaR Minimization

Portfolio VaR minimization problem (2.4) is an $n$-dimensional minimization problem with the objective function defined by a quantile definition of VaR given in (2.5). Using the definition of VaR given in (2.2), an alternative formulation for VaR minimization in $(n+1)$-variables can be

stated as,

$$\begin{array}{lll} \min\limits_{\mathbf{x}} & \min\limits_{\int_{-\mathbf{x}^T\mathbf{r}\leq\alpha} p(\mathbf{r})d\mathbf{r}\geq\beta} & \alpha \\ & \text{s.t.} & \sum\limits_{i=1}^{n} x_i = 1 \\ & & x_i \geq 0 \qquad i = 1, 2, \ldots, n, \end{array} \tag{3.1}$$

which is equivalent to,

$$\begin{array}{ll} \min\limits_{\mathbf{x},\alpha} & \alpha \\ \text{s.t.} & \int_{-\mathbf{x}^T\mathbf{r}\leq\alpha} p(\mathbf{r})d\mathbf{r} \leq 1-\beta \\ & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n. \end{array} \tag{3.2}$$

Problem (3.2) uses a probabilistic constraint to express VaR in the extended space. With the new formulation (3.2), the objective and constraint functions become simpler and combinatorial complexity in evaluating VaR function is avoided. We note the CVaR optimization method proposed in Uryasev and Rockafellar [11] use a similar probabilistic definition of VaR.

Suppose we are given $m$ equally probable return samples for $n$ assets. Assume that $\mathbf{r}^{(i)} \in \mathbb{R}^n$ denotes the $i$-th sample return of assets in the portfolio, $1 \leq i \leq m$. Let $\mathbb{I}_+(z)$ denote a step indicator function defined by the following equation:

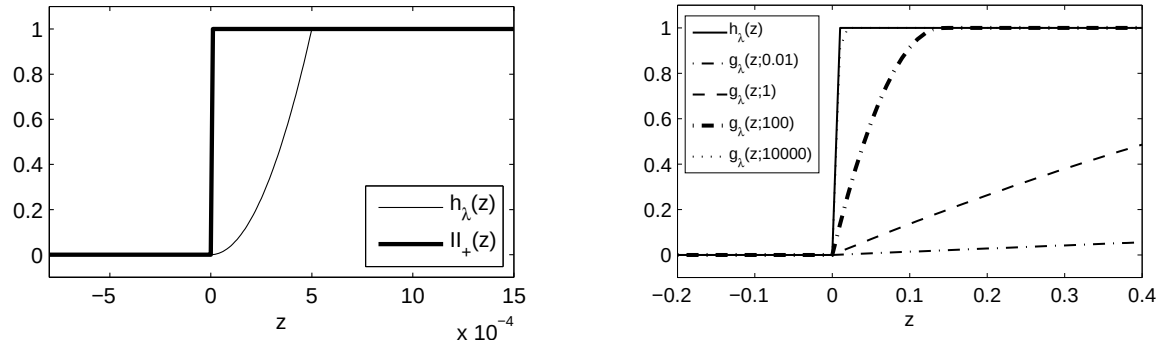
$$\mathbb{I}_+(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise .} \end{cases} \tag{3.3}$$

Then problem (3.2) becomes:

$$\begin{array}{ll} \min\limits_{\mathbf{x},\alpha} & \alpha \\ \text{s.t.} & \sum\limits_{i=1}^{m} \mathbb{I}_+(-\mathbf{x}^T\mathbf{r}^{(i)} - \alpha) \leq (1-\beta)m \\ & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \qquad i = 1, 2, \ldots, n. \end{array} \tag{3.4}$$

In Problem (3.4), the objective function $\alpha \in \mathbb{R}$ is an auxiliary variable. If the $i$-th loss, $-\mathbf{x}^T\mathbf{r}^{(i)}$, is greater than $\alpha$, the step function $\mathbb{I}_+(-\mathbf{x}^T\mathbf{r}^{(i)} - \alpha)$ equals 1, otherwise, it equals 0. Therefore the left-hand side of the first constraint in (3.4) is the number of sample losses that are greater than $\alpha$. This constraint guarantees that the number of sample losses that are greater than $\alpha$ cannot exceed $(1-\beta)m$. Hence, the minimum $\alpha$ value satisfying the probability constraint yields $\text{VaR}_\beta$ of a portfolio.

In contrast to the smoothed quantile approach, both the objective function and the constraint in (3.4) can be evaluated in linear time. Formulation (3.4) has a linear objective function with a discontinuous and nonconvex probabilistic inequality constraint. The challenge in solving (3.4) is due to this constraint, which we address by using a sequence of approximate functions similar to [2], as described below.




(a) $h_\lambda(z)$ and $\mathbb{I}_+(z)$ (b) $h_\lambda(z)$ and $g_\lambda(z;\rho)$

FIGURE 3.1: *Step function and its approximations ( $\lambda = 4 \times 10^6$)*

.

We first use the following function, $h_\lambda(z)$, to obtain a continuous approximation of the step function $\mathbb{I}_+(z)$:

$$h_\lambda(z) = \begin{cases} 1 & \text{if } z > \frac{1}{\sqrt{\lambda}} \\ \lambda z^2 & \text{if } 0 < z \leq \frac{1}{\sqrt{\lambda}} \\ 0 & \text{if } z \leq 0 \end{cases}. \tag{3.5}$$

where $\lambda > 0$ is a constant. The functions $h_\lambda(z)$ and $\mathbb{I}_+(z)$ only differ in the interval $[0, \frac{1}{\sqrt{\lambda}}]$. In order to be a good approximation for $\mathbb{I}_+(z)$, the constant $\lambda$ is chosen to be sufficiently large. In our computational results, $\lambda$ is approximately $10^6$.

Subplot (a) in Figure 3.1 graphs both $\mathbb{I}_+(z)$ and $h_\lambda(z)$. The thick line is the plot of the step function $\mathbb{I}_+(z)$. The thin line is a plot of the approximation function $h_\lambda(z)$ when $\lambda = 4 \times 10^6$. We can see that $h_\lambda(z)$ is equal to $\mathbb{I}_+(z)$ outside the interval $[0, \frac{1}{\sqrt{\lambda}}]$, while inside the interval, we use a quadratic function, which is equal to 0 at $z = 0$ and is equal to 1 at $z = \frac{1}{\sqrt{\lambda}}$, to approximate $\mathbb{I}_+(z)$. Hence for any fixed $\lambda \in \mathbb{R}^+$, the function $h_\lambda(z)$ is continuous over $z \in \mathbb{R}$.

The function $h_\lambda(z)$ is continuous but not differentiable at $z = \frac{1}{\sqrt{\lambda}}$. We now define a new function $g_\lambda(z;\rho)$, indexed by a parameter $\rho > 0$, to approximate $h_\lambda(z)$ as follows:

$$g_\lambda(z;\rho) = \begin{cases} 1 & \text{if } z \geq \gamma \\ 1 - \frac{\rho}{2}(z-\gamma)^2 & \text{if } \kappa \leq z \leq \gamma \\ \lambda z^2 & \text{if } 0 \leq z \leq \kappa \\ 0 & \text{if } z \leq 0 \end{cases} \tag{3.6}$$

where

$$\gamma = \sqrt{\frac{2}{\rho} + \frac{1}{\lambda}}, \quad \kappa = \frac{1}{\lambda\gamma}, \tag{3.7}$$

$\rho > 0$ is a parameter of the approximation. The function $g_\lambda(z;\rho)$ is a piecewise quadratic function. Inside each interval, $(-\infty, 0)$, $(0, \kappa)$, $(\kappa, \gamma)$, and $(\gamma, \infty)$, $g_\lambda(z;\rho)$ is differentiable. For example, $\gamma = 0.4472$, $\kappa = 5.59 \times 10^{-7}$ when $\rho = 10^7$ and $\gamma = 6.7 \times 10^{-4}$, $\kappa = 3.7 \times 10^{-4}$. Let $g'_+(z;\rho)$

and $g'_-(z;\rho)$ denote the right and the left limit of the function $g'(z;\rho)$ respectively. When $z=0$, $g'_+(0;\rho) = 2\lambda z|_{z=0} = 0 = g'_-(0;\rho)$. When $z=\gamma$, $g'_-(\gamma;\rho) = -\rho(z-\gamma)|_{z=\gamma} = 0 = g'_+(\gamma;\rho)$. When $z=\kappa$, $g'_-(\kappa;\rho) = 2\lambda z|_{z=\kappa} = 2\lambda\kappa = \frac{2}{\gamma}$, $g'_+(\kappa;\rho) = -\rho(z-\gamma)|_{z=\kappa} = -\rho(\kappa-\gamma) = -\frac{2}{\gamma^2-\frac{1}{\lambda}}(\frac{1}{\lambda\gamma}-\gamma) = \frac{2}{\gamma}$, hence $g'_-(\kappa;\rho) = g'_+(\kappa;\rho)$. Consequently, at the endpoints of the interval, $z=0$, $z=\kappa$, and $z=\gamma$, $g_\lambda(z;\rho)$ is also differentiable and the derivative is continuous. Therefore, for any fixed $\rho\in\mathbb{R}^+$, the function $g_\lambda(z;\rho)$ is continuously differentiable at any point $z\in\mathbb{R}$.

Problem (3.4) is nonconvex since the step function $\mathbb{I}_+(z)$ in the probabilistic constraint is nonconvex. The approximation $g_\lambda(z;\rho)$ is also nonconvex, due to the piece in $[\kappa,\gamma]$, which is a concave quadratic with negative curvature $-\rho$. To solve (3.4), we use a gradual nonconvexification process [e.g. see 3] in attempt to track a global VaR optimal portfolio.

Subplot (b) in Figure 3.1 shows plots of $h_\lambda(z)$ and $g_\lambda(z;\rho)$ for $\rho = 0.01, 1, 100, 10000$ and $\lambda = 4\times 10^6$. We see that as $\rho\to\infty$, $g_\lambda(z;\rho)$ approaches to $h_\lambda(z)$. In Figure 3.1(b), $g_\lambda(z;10000)$ is very close to $h_\lambda(z)$ and they only differ in a very small interval $[\kappa,\gamma]$ to the right of $z=0$. It is also straightforward to verify that $\kappa\to 0$ and $\gamma\to+\infty$ as $\rho\to 0$, $\kappa\to\frac{1}{\sqrt{\lambda}}$ and $\gamma\to\frac{1}{\sqrt{\lambda}}$ as $\rho\to+\infty$. Indeed, when $\rho$ is very small, the interval $[\kappa,\gamma]$ occupies almost the whole range of $\mathbb{R}^+$ and the curvature of $g_\lambda(z;\rho)$ is close to zero. Thus $g_\lambda(z;\rho)$ is nearly a linear (convex) function in this interval. As $\rho$ increases, the length of the interval $[\kappa,\gamma]$ shrinks towards zero and $g_\lambda(z;\rho)$ approaches $h_\lambda(z)$.

Replacing $\mathbb{I}_+(z)$ with $g_\lambda(z;\rho)$ in problem (3.4), we obtain the following approximate VaR minimization problem:

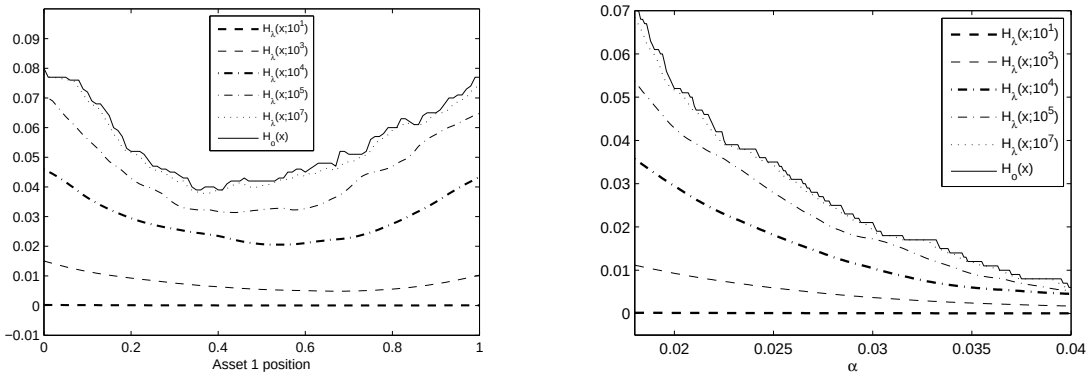
$$\begin{array}{ll}\min\limits_{\mathbf{x},\alpha} & \alpha \\ \text{s.t.} & H_\lambda(\mathbf{x};\rho)\le(1-\beta) \\ & \sum\limits_{i=1}^{n} x_i = 1 \\ & x_i\ge 0 \qquad i=1,2,\ldots,n,\end{array} \tag{3.8}$$

where

$$H_\lambda(\mathbf{x};\rho) = \frac{1}{m}\sum_{i=1}^{m} g_\lambda(-\mathbf{x}^T\mathbf{r}^{(i)}-\alpha;\rho). \tag{3.9}$$

When the parameter $\rho$ is small, problem (3.8) is close to a (convex) linear programming problem. As we gradually increase $\rho$, problem (3.8) gradually becomes increasingly more non-convex. The solution to (3.8), when $\rho$ is sufficiently large, yields an accurate approximation to the optimal $\text{VaR}_\beta$ portfolio.

To illustrate, we use the same 1000 historical samples for the two stock allocation example in §2. Consider the nonconvex function $H_\lambda(\mathbf{x};\rho) = \frac{1}{m}\sum_{i=1}^m g_\lambda(-\mathbf{x}^T\mathbf{r}^{(i)}-\alpha;\rho)$ defining the probabilistic constraint in problem (3.8). Figure 3.2 graphs $H_\lambda(\mathbf{x};\rho)$, for different values of $\rho$, as a function of $x_1$ when $x_2 = 1-x_1$ corresponding to the two asset allocation example ($\alpha$ is set to 0.02). The thick dashed line at the bottom is $H_\lambda(\mathbf{x};10^1)$; it is almost a straight line. The thin dashed line curve above it is $H_\lambda(\mathbf{x};10^3)$; it is smooth and visually convex. The thick dash-dot line in the middle, $H_\lambda(\mathbf{x};10^4)$, becomes more non-convex and has multiple local minima. The dash-dot and dotted lines above, corresponding to $H_\lambda(\mathbf{x};10^5)$ and $H_\lambda(\mathbf{x};10^7)$, respectively, exhibit a greater number of local minima. The original function, denoted as $H_o(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \mathbb{I}_+(-\mathbf{x}^T\mathbf{r}^{(i)}-\alpha)$, is the solid line curve at the top in Figure 3.2. We can see that $H_\lambda(\mathbf{x};10^7)$ is very close to the original function $H_o(\mathbf{x})$. The dependence relationship of the function $H_\lambda(\mathbf{x};\rho)$ with respective to $\alpha$ is illustrated in




(a) $H_\lambda(\mathbf{x};\rho)$ with respect to $x_1$ (b) $H_\lambda(\mathbf{x};\rho)$ with respect to $\alpha$

FIGURE 3.2: *Plots of $H_o(\mathbf{x})$ and its approximations ($\lambda = 4 \times 10^6$, $\alpha = 0.02$), $\rho = 10^1, 10^3, 10^4, 10^5, 10^7$*

.

subplot (b) in Figure 3.2. We make similar observations as in subplot (a): $H_\lambda(\mathbf{x};\rho)$, as a function of $\alpha$, approaches the original non-smooth function as $\rho$ increases.

Figure 3.2 illustrates that $H_o(\mathbf{x})$ function can be viewed as a global convex curve superimposed with many small variations which create multiple local minimizers. The global minimizer is in a neighborhood of the minimizer of this globally convex curve. When $\rho$ is small, the function $H_\lambda(\mathbf{x})$ approximates $H_o(\mathbf{x})$ from a global perspective. In this case, since it is close to a convex problem, a computed solution is much more likely to be a global minimizer of Problem (3.8). This yields a candidate solution for optimal VaR portfolio from a crude but global perspective, as the function is regarded as a composition of a smooth curve plus noisy variations. The approximation can then be refined and a more accurate approximation to the VaR optimization problem can be obtained using this candidate as a starting point. As we successively refine the approximation, the solution attempts to track the global minimizer.

More formally, let $\rho_0 > 0$ be an initial small number, e.g., $\rho_0 = 10^{-4}$. We compute an optimal solution $x_0^*$ for problem (3.8) corresponding to $\rho_0$. In general, assume we have computed $x_k^*$ as an optimal solution to (3.8) with parameter $\rho_k$. Now let $\rho_{k+1} = c\rho_k$ where $c > 1$. Using $x_k^*$ as a starting point, we solve (3.8) with the parameter $\rho_{k+1}$ and obtain a solution $x_{k+1}^*$. We repeat this process until the interval $[\kappa, \gamma]$ is sufficiently small and there is no loss $-\mathbf{x}^T\mathbf{r}^{(i)}$ such that $z = -\mathbf{x}^T\mathbf{r}^{(i)} - \alpha$ falls in this interval. This criteria is reasonable since $\mathcal{I} = \{\ i | -\mathbf{x}_k^T\mathbf{r}^{(i)} - \alpha_k \in [\kappa, \gamma], i = 1, 2, \ldots, m\}$ is empty implies that the current optimal solution is also optimal for the optimization problem that uses $h_\lambda(z)$ instead of $g_\lambda(z;\rho_k)$ in (3.4), which follows from the fact that $g_\lambda(z;\rho_k)$ and $h_\lambda(z)$ only differ in the interval $[\kappa_k, \gamma_k]$. Algorithm 1 describes the computational process in detail. We refer to this algorithm as a gradual non-convexification penalty (GNCP) method for minimizing VaR. The computational complexity of function and constraint evaluation for this method is $O(m)$, where $m$ is the number of scenarios.

Problem (3.8) can be solved using an optimization algorithm for nonconvex programming. In our implementation, we use an exact penalty method to handle the nonconvex constraint. Specifically

**Algorithm 1:** Gradual Non-Convexification Penalty Method for Minimizing VaR

**Input**: $m$ sample returns for $n$ assets, $\mathbf{r}^{(i)} \in \mathbb{R}^n, i = 1, 2, \ldots, m$; starting point $(\alpha_0, \mathbf{x}_0)$, a resolution parameter $\lambda > 0$, and penalty parameter $\mu > 0$

**Output**: A computed solution $\mathbf{x}$ for VaR minimization problem (3.4)

**begin**

$k \longleftarrow 0$ ;

$\rho_k \longleftarrow 1 \times 10^{-5}$ ;

$\mathcal{I} \longleftarrow \{\ i | -\mathbf{x}_k^T \mathbf{r}^{(i)} - \alpha_k \in [\kappa, \gamma], i = 1, 2, \ldots, m\}$ ;

**while** $\mathcal{I}$ *is not empty* **or** $|\gamma_k - \kappa_k| > 1 \times 10^{-4}$ **do**

$k \longleftarrow k + 1$ ;

$\rho_k \longleftarrow 10\rho_{k-1}$ ;

solve problem (3.8) using the exact penalty method (3.10) with $\rho = \rho_k$ and starting point $(\alpha_{k-1}, \mathbf{x}_{k-1})$ to obtain the computed solution $(\alpha_k, \mathbf{x}_k)$ ;

**end**

**return** $\mathbf{x}_k$ **;**

**end**

the nonlinearly constrained problem (3.8) is solved using an exact penalty:

$$\min_{\mathbf{x}\in\Omega,\alpha} \quad \alpha + \mu \cdot \max\left(H_\lambda\left(\mathbf{x};\rho\right) - (1-\beta), 0\right) \tag{3.10}$$

$$\text{where} \qquad \Omega = \{\mathbf{x} \in \mathbb{R}^n | \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \text{ for } i = 1, 2, \ldots, n\}. \tag{3.11}$$

Here $\mu$ is a sufficiently large exact penalty parameter. When the probabilistic constraint in problem (3.8) is satisfied, the penalty term does not have any impact on the objective function in (3.10). On the other hand, if the constraint is not satisfied, the penalty term $\mu \cdot \max\left(H_\lambda\left(\mathbf{x};\rho\right) - (1-\beta), 0\right)$ becomes dominant and its minimization enforces satisfaction of the constraint.

Note, the parameter $\rho$ in $g_\lambda(z;\rho)$ has a similar effect as the parameter $\epsilon$ in the smoothed quantile method of Gaivoronski and Pflug [7]. However, in Algorithm 1, $\rho$ is automatically adjusted at each iteration to track a minimizer, thereby avoiding the difficultly in selecting a fixed value for the parameter.

Since the gradual non-convexification method solves a sequence of optimization problems, a natural concern is the computational cost. We note that consecutive problems are similar due to the fact that $\rho$ is updated gradually. Thus we exploit this property and use a warm start when solving a subsequent problem. Typically, each problem requires only a few iterations for the optimization subroutine.

# 4 Computational Comparisons

In this section, we present computational results to compare the proposed gradual non-convexification penalty (GNCP) method for minimizing VaR with the quantile-based smoothed VaR (QBSVaR) method of Gaivoronski and Pflug [7]. We compare the results in terms of the computed VaR of the optimal portfolio, the CPU time, and the efficient frontiers. Both historical and synthetic data are

used for computational comparisons. All the tests are done in Matlab on a 2.9GHz PC that has an AMD Athlon 64 X2 Dual CPU and 4 GB RAM.

The historical data set, denoted as DS1, consists of 10 year historical daily returns (2513 samples) of 30 stocks in Dow Jones stock index from September 4th, 1991 to September 4th 2001. The names of 30 stocks are listed in Table 4.1.

| | | | | | |
|---|---|---|---|---|---|
| 1 | AA | 11 | HD | 21 | MMM |
| 2 | AXP | 12 | HON | 22 | MO |
| 3 | BA | 13 | HWP | 23 | MRK |
| 4 | C | 14 | IBM | 24 | MSFT |
| 5 | CAT | 15 | INTC | 25 | PG |
| 6 | DD | 16 | IP | 26 | SBC |
| 7 | DIS | 17 | JNJ | 27 | T |
| 8 | EK | 18 | JPM | 28 | UTX |
| 9 | GE | 19 | KO | 29 | WMT |
| 10 | GM | 20 | MCD | 30 | XOM |

TABLE 4.1: *Stocks in data set 1*

The synthetic data set, denoted as DS2, is generated from correlated multivariate jump models with random model parameters. This data set consists of 100000 returns of 1000 assets. The return of each asset follows a Merton's jump model [10]. We use this large data set to compare computational efficiency of GNCP method and QBSVaR method. Figure 4.1 shows a return distribution of an asset in DS2, we can see that there is a bump on the left of the return distribution, representing a significant probability of unusually large losses.

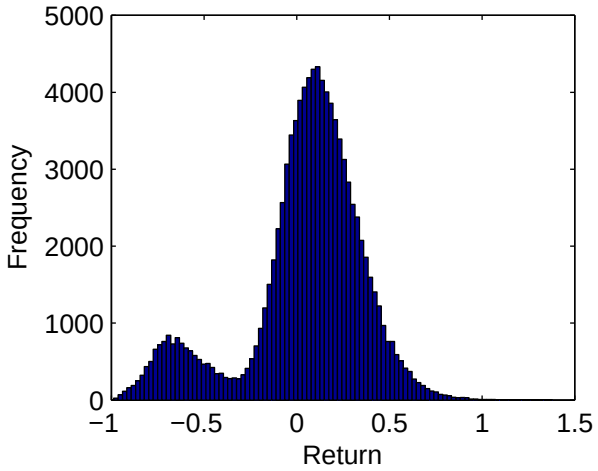



FIGURE 4.1: *Return distribution of an asset in DS2*

.

Firstly, we illustrate the effect of starting points of our GNCP method. For all the results reported in this paper, we set the accuracy parameter $\lambda = 4 \times 10^6$. Using the historical data

set DS1, we compare computed optimal VaR from GNCP using the equal weight starting point $\mathbf{x}_{ew} = \frac{1}{n}e$, where $e$ is a $n$-component vector of ones, with VaR obtained using the starting point $(\mathbf{x}^*_{\text{CVaR}})$, which is the optimal CVaR portfolio from the CVaR minimization method [11]. We set $\alpha_0 = 0$ in each computation.

Table 4.2 shows $\text{VaR}_{0.95}$ and $\text{VaR}_{0.9}$ for different number of assets, $n$, and different number of returns, $m$. We can see that the VaR of the portfolio corresponding to two starting points are not significantly different. This illustrates that GNCP method is relatively insensitive to the starting point in terms of the VaR objective value. In addition to the VaR value[1], the relative difference between the computed portfolios will be an additional important property. However a careful investigation of this issue concerns the stability of the VaR optimization, which is outside the scope of this paper.

| m | n | $\text{VaR}_{0.95}$ | | $\text{VaR}_{0.9}$ | |
|---|---|---|---|---|---|
| | | $\frac{1}{n}$ | $\mathbf{x}^*_{\text{CVaR}}$ | $\frac{1}{n}$ | $\mathbf{x}^*_{\text{CVaR}}$ |
| 300 | 10 | 0.01061 | 0.010958 | 0.0079865 | 0.0082694 |
| 300 | 20 | 0.0087491 | 0.0087733 | 0.0072237 | 0.0073217 |
| 300 | 30 | 0.0074684 | 0.0074303 | 0.0051924 | 0.0053269 |
| 500 | 10 | 0.010772 | 0.010086 | 0.0078666 | 0.0078336 |
| 500 | 20 | 0.0087856 | 0.0090869 | 0.0072434 | 0.0080066 |
| 500 | 30 | 0.0075459 | 0.0080684 | 0.0056612 | 0.0058597 |
| 1000 | 10 | 0.010478 | 0.010247 | 0.0076282 | 0.0075285 |
| 1000 | 20 | 0.0089188 | 0.0087112 | 0.0071986 | 0.0068093 |
| 1000 | 30 | 0.0075692 | 0.0076763 | 0.0056484 | 0.0056494 |
| 2000 | 10 | 0.013762 | 0.013913 | 0.010026 | 0.0099563 |
| 2000 | 20 | 0.012347 | 0.011696 | 0.0086397 | 0.0086409 |
| 2000 | 30 | 0.01039 | 0.01044 | 0.0075813 | 0.0076355 |

TABLE 4.2: *Effect of starting points for GNCP method*

Next we compare GNCP with QBSVaR in terms of the VaR of the computed optimal portfolio using both historical and synthetic data sets. For all subsequent results, we use the starting point $\alpha_0 = 0$ and $\mathbf{x}_{ew}$. We note that $\rho$ is usually around $10^8 \sim 10^9$ when GNCP terminates.

We report VaR of the computed minimal VaR portfolio for different number of assets, $n$, and different number of returns, $m$. In order to access improvement obtained using GNCP, we report the relative difference defined as follows:

$$RD^Q_{VaR} = \frac{\text{VaR}_{\text{QBSVaR}} - \text{VaR}_{\text{GNCP}}}{|\text{VaR}_{\text{GNCP}}|} 100\%. \tag{4.1}$$

Positive relative difference indicates degradation of QBSVaR method compared to GNCP method.

Table 4.3 shows $\text{VaR}_{0.95}$ and the relative difference on historical data for different number of assets, $n$, and different number of sample returns, $m$. For QBSVaR method, we report results for four different values, 0.01, 0.001, 0.0001, and 0.00001, for the smooth parameter $\epsilon$. Results in Table 4.3 demonstrates that performance of of QBSVaR is sensitive to the smoothing parameter $\epsilon$. In

[1]We thank an anonymous referee for the comment regarding this.

| m | n | $\text{VaR}_{\text{GNCP}}$ | $RD^{Q}_{VaR}$ | | | |
|---|---|---|---|---|---|---|
| | | | $\epsilon = 0.01$ | $\epsilon = 0.001$ | $\epsilon = 0.0001$ | $\epsilon = 0.00001$ |
| 300 | 10 | 0.01061 | 24.27% | 6.66% | 8.03% | 7.82% |
| 300 | 20 | 0.0087491 | 24.45% | 6.17% | 4.17% | 8.43% |
| 300 | 30 | 0.0074684 | 35.96% | 15.11% | 12.83% | 17.78% |
| 500 | 10 | 0.010772 | 4.91% | 0.05% | -5.70% | 1.58% |
| 500 | 20 | 0.0087856 | 13.06% | 6.27% | 2.77% | 2.63% |
| 500 | 30 | 0.0075459 | 20.71% | 13.47% | 10.24% | 12.28% |
| 1000 | 10 | 0.010478 | 9.76% | 13.56% | 3.22% | 3.99% |
| 1000 | 20 | 0.0089188 | 9.24% | 14.65% | 4.61% | 7.45% |
| 1000 | 30 | 0.0075692 | 18.93% | 21.03% | 14.17% | 19.47% |
| 2000 | 10 | 0.013762 | 7.06% | 5.27% | 4.56% | 4.40% |
| 2000 | 20 | 0.012347 | 4.73% | 4.41% | 10.26% | 8.16% |
| 2000 | 30 | 0.01039 | 10.31% | 11.31% | 12.27% | 9.96% |

TABLE 4.3: *VaR of minimizing VaR portfolio using GNCP method and the relative difference compared to the QBSVaR method on historical data, $\beta = 95\%$*

| m | n | $\text{VaR}_{\text{GNCP}}$ | $RD^{Q}_{VaR}$ | | | |
|---|---|---|---|---|---|---|
| | | | $\epsilon = 0.01$ | $\epsilon = 0.001$ | $\epsilon = 0.0001$ | $\epsilon = 0.00001$ |
| 300 | 10 | 0.0079865 | 17.45% | 19.39% | 4.17% | 8.56% |
| 300 | 20 | 0.0072237 | 18.54% | 9.79% | 3.92% | 11.94% |
| 300 | 30 | 0.0051924 | 54.74% | 28.17% | 14.92% | 29.28% |
| 500 | 10 | 0.0078666 | 11.05% | 3.41% | 2.08% | 2.49% |
| 500 | 20 | 0.0072434 | 13.15% | -1.98% | 5.16% | 6.59% |
| 500 | 30 | 0.0056612 | 25.58% | 9.30% | 7.53% | 4.64% |
| 1000 | 10 | 0.0076282 | 10.54% | 5.53% | 3.52% | 8.84% |
| 1000 | 20 | 0.0071986 | 2.45% | 1.29% | 1.76% | 0.45% |
| 1000 | 30 | 0.0056484 | 16.77% | 10.68% | 10.55% | 10.20% |
| 2000 | 10 | 0.010026 | 6.81% | 3.20% | 4.81% | 3.32% |
| 2000 | 20 | 0.0086397 | 6.22% | 2.11% | 6.28% | 7.44% |
| 2000 | 30 | 0.0075813 | 9.50% | 9.91% | 9.72% | 8.96% |

TABLE 4.4: *VaR of minimizing VaR portfolio using GNCP method and the relative difference compared to the QBSVaR method on historical data, $\beta = 90\%$*

addition, on this historical data set at least, GNCP outperform QBSVaR in achieving smaller VaRs in all cases with a single exception when $m = 500$, $n = 10$ and $\epsilon = 0.0001$ for QBSVaR. We also report $\text{VaR}_{0.90}$ and the relative difference on data set DS1 in Table 4.4.

Table 4.5 and 4.6 report VaR obtained using the GNCP method and the relative difference performance measure with $\beta = 95\%$ on the synthetic data set DS2. For simplicity, we report results for $\epsilon = 0.001$ in QBSVaR, as this seems to generate best results for QBSVaR. Table 4.5 compares performance as the number of scenarios increases; the entries for QBSVaR when $m = 30000$ or

| m | $n = 30$ | | $n = 50$ | | $n = 100$ | |
|---|---|---|---|---|---|---|
| | $\text{VaR}_{\text{GNCP}}$ | $RD^Q_{VaR}$ | $\text{VaR}_{\text{GNCP}}$ | $RD^Q_{VaR}$ | $\text{VaR}_{\text{GNCP}}$ | $RD^Q_{VaR}$ |
| 300 | 0.066506 | 10.48% | 0.059168 | -29.25% | 0.020184 | 33.05% |
| 1000 | 0.086351 | 4.16% | 0.070866 | 0.68% | 0.045326 | 16.14% |
| 3000 | 0.094539 | -0.36% | 0.076082 | 6.29% | 0.056926 | 16.04% |
| 10000 | 0.094991 | 2.77% | 0.081098 | 3.93% | 0.069305 | 4.01% |

TABLE 4.5: *VaR of minimizing VaR portfolio using GNCP method and the relative difference compared to the QBSVaR method on synthetic data,* $\beta = 95\%$, $\epsilon = 0.001$

| m | n | $\text{VaR}_{\text{GNCP}}$ | $RD^Q_{VaR}$ | | |
|---|---|---|---|---|---|
| | | | $\epsilon = 0.01$ | $\epsilon = 0.001$ | $\epsilon = 0.0001$ |
| 1000 | 500 | 0.014584 | 91.15% | 198.20 % | 220.87% |
| 2000 | 500 | 0.026121 | 51.27% | 82.58% | 107.37% |
| 2000 | 1000 | 0.034729 | 55.02% | 16.37% | 55.43% |
| 5000 | 500 | 0.033475 | 90.65% | 56.36% | 85.08% |
| 5000 | 1000 | 0.040264 | - | 24.30% | 18.71% |

TABLE 4.6: *VaR of minimizing VaR portfolio using GNCP method and the relative difference compared to the QBSVaR method on synthetic data when n is large,* $\beta = 95\%$

$m = 100000$ are blank, since the computation did not complete in 3 days. Table 4.6 compares performance as the number of assets increases. We observe that GNCP continue to outperform QBSVaR and the improvement of GNCP over QBSVaR becomes more significant as the number of assets increase (with a 220% maximum improvement ratio).

Finally, we compare CPU time required by GNCP and QBSVaR. Figure 4.2 shows the CPU time (in seconds) for evaluating the smoothed probabilistic constraint $H_\lambda(\mathbf{x}; \rho)$ in the GNCP method and the QBSVaR function $G_\epsilon(k, \mathbf{x})$ in the QBSVaR method for different number of returns, $m$, when $n = 100$ and $\beta = 95\%$. We can see that the CPU time for evaluating $H_\lambda(\mathbf{x}; \rho)$ grows very slowly with respect to the number of samples, $m$. It is linear (with very small slope) with respect to $m$. However, the CPU time to evaluate $G_\epsilon(k, \mathbf{x})$ grows much faster than that to evaluate $H_\lambda(\mathbf{x}; \rho)$. Data set DS2 is used for this test since we need to see the running time of both methods as the problem size becomes large. We fix $\mathbf{x} = \frac{1}{n}e$for both functions. For the smoothed probabilistic constraint, we choose $\alpha = 0.01$, $\lambda = 4 \times 10^6$, and $\rho = 100$. For the QBSVaR function, we choose $\epsilon = 0.001$.

We now compare CPU time for computing optimal VaR portfolio for different number of assets, $n$, and different number of returns, $m$. Table 4.7 reports the CPU time (in minutes) for both methods on the synthetic data set DS2 with $\beta = 95\%$. For the QBSVaR method, we choose $\epsilon = 0.01$ for the smooth parameter. We do not report the CPU time for QBSVaR method when $m = 30000$ or $m = 100000$ since it takes more than 3 days to obtain the VaR optimal portfolio using QBSVaR method. We also report the CPU time when $n$ is large in Table 4.8. We compute the relative time ($RT$), which is defined in equation (4.2), to provide a more clear comparison between

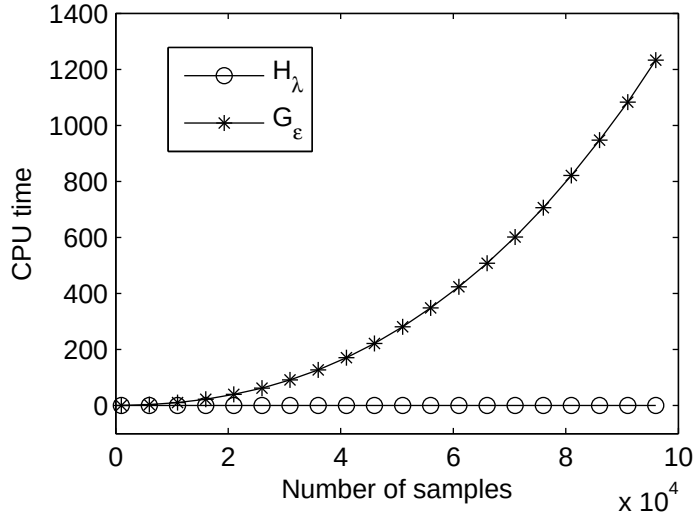



FIGURE 4.2: *CPU time for function evaluation,* $n = 100$, $\beta = 95\%$, $\epsilon = 0.001$

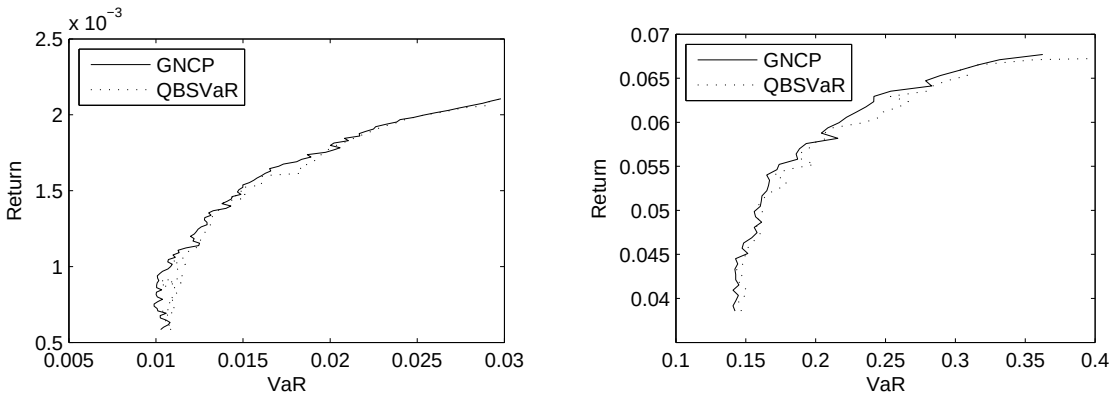
.

| m | CPU Time (Minutes) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $n = 30$ | | | $n = 50$ | | | $n = 100$ | | |
| | GNCP | QBSVaR | RT | GNCP | QBSVaR | RT | GNCP | QBSVaR | RT |
| 300 | 0.19 | 1.83 | 9.64 | 0.18 | 3.05 | 16.76 | 0.30 | 6.22 | 20.72 |
| 1000 | 0.22 | 9.41 | 42.99 | 0.32 | 15.83 | 49.07 | 0.66 | 32.34 | 49.19 |
| 3000 | 0.53 | 56.04 | 104.92 | 0.76 | 95.70 | 126.67 | 1.40 | 190.10 | 135.37 |
| 10000 | 1.14 | 506.38 | 442.84 | 1.93 | 865.88 | 448.95 | 3.35 | 1789.40 | 534.26 |
| 30000 | 3.05 | - | - | 4.94 | - | - | 11.20 | - | - |
| 100000 | 9.82 | - | - | 16.54 | - | - | 34.41 | - | - |

TABLE 4.7: *CPU time of running GNCP method and QBSVaR method on synthetic data and relative time,* $\beta = 95\%$, $\epsilon = 0.001$

| m | n | CPU Time (Minutes) | | | | RT, $\epsilon = 0.001$ |
|---|---|---|---|---|---|---|
| | | GNCP | QBSVaR | | | |
| | | | $\epsilon = 0.01$ | $\epsilon = 0.001$ | $\epsilon = 0.0001$ | |
| 1000 | 500 | 8.41 | 293.23 | 163.53 | 140.34 | 19.46 |
| 2000 | 500 | 10.25 | 1225.59 | 483.75 | 418.31 | 47.18 |
| 2000 | 1000 | 51.70 | 2369.37 | 986.47 | 847.74 | 19.08 |
| 5000 | 500 | 17.04 | 11218.49 | 2396.22 | 2086.44 | 140.61 |
| 5000 | 1000 | 73.61 | - | 4217.67 | 4854.85 | 57.30 |

TABLE 4.8: *CPU time of running GNCP method and QBSVaR method on synthetic data and relative time when* $n$ *is large,* $\beta = 95\%$




(a) historical data

(b) synthetic data

FIGURE 4.3: *Comparisons in efficient frontiers:* $m = 500$, $n = 10$, $\beta = 95\%$

.

the two methods.

$$RT = \frac{Time_{QBSVaR}}{Time_{GNCP}} \tag{4.2}$$

We make following observations based on Table 4.7 and Table 4.8.

- QBSVaR takes less time to solve the VaR minimization problem (2.4) when the smooth parameter $\epsilon$ becomes smaller. According to Gaivoronski and Pflug [7], when $\epsilon$ is small, the number of losses that satisfies the constraint $|G(k, \mathbf{x}) - f_i(\mathbf{x})| \leq \epsilon$ is small and this reduces the time to evaluate the QBSVaR function $G_\epsilon(k, \mathbf{x})$.

- For a fixed number of assets, $n$, the CPU time of GNCP grows much slower than QBSVaR as the number of scenarios, $m$, increases. With 10000 scenarios, GNCP is more than 400 times faster than QBSVaR. Moreover, as the number of scenarios is further increased QBSVaR takes more than 3 days to complete, while GNCP can compute the optimal VaR portfolio in a reasonable amount of time.

Finally, we compare VaR frontiers obtained using both GNCP and QBSVaR methods. For different levels of expected return target $R$, we solve problem (2.3) and obtain the corresponding VaR of the optimal portfolio.

Subplots (a) and (b) in Figure 4.3 compare the VaR-Return frontiers of the samples for the historical data set DS1 and the synthetic data set DS2 respectively ($m = 500$, $n = 10$ and $\beta = 95\%$). The solid line is the VaR-Return frontier obtained using GNCP. The dotted line is the VaR-Return frontier we obtain using QBSVaR. From Figure 4.3, it can be observed that the efficient frontier from GNCP dominates that from QBSVaR. For a fixed return $R$, the VaR using GNCP is smaller than that obtained using QBSVaR. We note that these two methods converge at the right end of the frontier. This is reasonable since the right end of the frontier corresponds to the return maximization and VaR minimization becomes irrelevant.

# 5 Concluding Remarks

Given a finite set of return samples, VaR minimization problem is computationally difficult to solve: VaR function is non-convex, non-smooth, and has many local minima. Recently Gaivoronski and Pflug [7] propose a method that minimizes a quantile-based smoothed VaR (QBSVaR) approximation function. They express the quantile as a linear combination of loss functions using a product of indicator functions as coefficients. A smooth differentiable function is obtained by approximating the step function using a cubic spline. This approximation is complicated and expensive to evaluate since it involves enumerating all possible combinations of choosing $k$ tail events from $m$ scenarios. Implemented efficiently, this method still requires $O(m^3)$ time to evaluate. Optimization routines typically require many function, derivative and Hessian evaluations. Thus this method becomes impractical for large sample sizes. In addition, a smoothing parameter, $\epsilon$, is used to control how accurate the approximation is and quality of the computed solution depends on the right choice of $\epsilon$, which can be problem dependent and difficult to determine.

In this paper, we proposed a gradual non-convexification penalty (GNCP) method for VaR minimization. We formulate the problem in an augmented space consisting of portfolio weights and an auxiliary variable $\alpha$; $\text{VaR}_\beta$ is the minimum $\alpha$ value which satisfies the requirement that the probability of loss greater than $\alpha$ does not exceed $1-\beta$. The resulting optimization problem has a linear objective with a probabilistic constraint expressed using a sum of step indicator functions. Consequently evaluating the objective and the constraint function requires only $O(m)$ work.

We approximate the step indicator function using a continuously differentiable function, indexed by a parameter $\rho \geq 0$ and containing negative curvature $-\rho$. When $\rho \to 0$, the approximation to the (nonconvex) probability constraint function approaches a linear function, resulting in a nearly convex minimization problem. When $\rho$ is small, the approximation captures the global characteristics of the VaR minimization problem, and a globally optimal solution is likely to be obtained. As $\rho$ increases, more negative curvature is gradually introduced in the approximation and the problem becomes gradually nonconvex. As the parameter goes to infinity, the approximation approaches the original VaR minimization problem. We solve a sequence of problems indexed by a monotonically increasing sequence $\{\rho_k\}$, in which we use the solution indexed by $\rho_k$ as a starting point for the subsequent problem indexed by $\rho_{k+1}$ (i.e. warm start), thus reducing individual optimization times.

Both historical and synthetic data are used to evaluate performance of the proposed GNCP method and the QBSVaR method. Our computational results demonstrate that the GNCP method generally provides a better minimizer than the QBSVaR method, and the VaR-return frontier of the GNCP method dominates that of the QBSVaR method. In addition, GNCP requires less time, with a significantly more appealing asymptotic run time as both number of scenarios or assets are increased.

In the future, we intend to investigate condition of VaR risk measure in portfolio optimization problem as well as the stability of the proposed optimization algorithm.

# References

[1] Artzner, P., F. Delbaen, J.-M. Eber, and D.Heath (1999). Coherent measures of risk. *Mathematical Finance* 9(3), 203–228.

[2] Blake, A. and A. Zisserman (1987). *Visual Reconstruction*. Cambridge.

[3] Coleman, T. F., J. Henninger, and Y. Li (2006). Minimizing tracking error while restricting the number of assets. *Journal of Risk* 8, 33–56.

[4] Cont, R., R. Deguest, and G. Scandolo (2010). Robustness and sensitivity analysis of risk measure procedures. *Quantitative Finance* 10, 593–606.

[5] Dowd, K. (2005). *Measuring Market Risk*. John Wiley & Sons, Ltd, 2nd edition.

[6] Duffie, D. and J. Pan (1997). An overview of Value at Risk. *Journal of Derivatives* 4, 9–49.

[7] Gaivoronski, A. A. and G. Pflug (2005). Value-at-Risk in portfolio optimization: Properties and computational approach. *Journal of Risk* 7(2), 1–31.

[8] Larsen, N., H.Mausser, and S. Uryasev (2002). Algorithm for optimization of Value-at-Risk. In P. Pardalos and V. Tsitsiringos, eds., *Financial Engineering, e-Commerce and Supply Chain*, pp. 129–157. Kluwer Academic Publishers.

[9] Markowitz, H. (1952). Portfolio selection. *Journal of Finance* 7(1), 77–91.

[10] Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3, 125–144.

[11] Uryasev, S. and R. Rockafellar (2000). Optimization of conditional Value-at-Risk. *The Journal of Risk* pp. 21–41.

[12] Wozabal, D., R. H. Hochreiter, and G. C. Pflug (2010). A difference of convex formulation of Value-at-Risk constrained optimization. *Optimization* 59, 377–400.