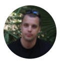
Optimization of Value-at-Risk: Computational Aspects of MIP Formulations

Article *in* Journal of the Operational Research Society · December 2017

DOI: 10.1057/s41274-017-0197-4

3 authors:


Konstantin Pavlikov
University of Southern Denmark
14 PUBLICATIONS **152** CITATIONS
SEE PROFILE

Alexander Veremyev
University of Central Florida
57 PUBLICATIONS **699** CITATIONS
SEE PROFILE

Eduardo Pasiliao
Air Force Research Laboratory
85 PUBLICATIONS **929** CITATIONS
SEE PROFILE

Optimization of Value-at-Risk: Computational Aspects of MIP Formulations

Konstantin Pavlikov[a], Alexander Veremyev[a], Eduardo L. Pasiliao[b]

[a]*Department of Industrial and Systems Engineering*
303 Weil Hall, University of Florida, Gainesville, FL 32611

[b]*Munitions Directorate, Air Force Research Laboratory,*
Eglin AFB, FL 32542

E-mails: `kpavlikov@ufl.edu, averemyev@ufl.edu, eduardo.pasiliao@eglin.af.mil.`

First draft: September 2014, This draft: October 30, 2016

Correspondence should be addressed to: Konstantin Pavlikov

Abstract

Optimization of Value-at-Risk is an important problem both from theoretical and practical standpoints. It can be represented through a class of chance-constrained optimization problems, which are generally hard to solve. Mixed integer problem formulations with big M constants is a standard way to approach such problems, where tightness of these constants is a crucial factor for good performance of a solver. This study aims to improve the tightness of existing big Ms by explicitly incorporating bounds on the optimal value of VaR into the problem formulation. Moreover, the lower bound is demonstrated to play an important role in obtaining tight big M constants and a procedure to lift this bound is discussed. Finally, a "two-stage" solution approach is proposed, where the first stage solely deals with tightening the bounds, and the seconds stage employs improved bounds to redefine big Ms and solves the problem to optimality. Numerical experiments suggest that proposed solution methods can decrease solution time by up to 75% compared to the most recent benchmark, which may allow to handle larger problem instances while using the same hardware.

Keywords: Chance-Constrained Programming, Percentile Minimization, Combinatorial Optimization, Value-at-Risk, Conditional Value-at-Risk

---

1. Introduction

Optimization problems involving Value-at-Risk (VaR) can be divided into two main categories: with Value-at-Risk in the objective and Value-at-Risk in the constraint. The latter problem is known to fall into a widely studied area of chance-constrained problems, Nemirovski and Shapiro (2006). The former problem received considerably less attention in the literature and is the main focus of this paper. While problems with VaR in the objective can also be interpreted as special cases of chance-constrained problems, we find it useful to separate these cases and explicitly employ some special advantages of doing so in the optimization process.

The Value-at-Risk first appeared as the risk measure in various areas of risk management of financial institutions, see Duffie and Pan (1997). It remains an important risk management tool in the most recent Basel III framework on capital regulation and market risk, Basel Committee on Banking Supervision (2010). As a risk measure, it answers the question of the maximum possible loss of a financial institution that is not exceeded with a specified probability and over a specified period of time. Hence, the rationale behind minimization of VaR is straightforward: if there is a threshold that is not exceeded by random loss with probability at least α, then with all other settings kept the same, one would prefer this threshold to be as small as possible. In a simplest discrete setting, the VaR minimization problem reduces to the minimization of the k-th largest loss in a set of ordered loss scenarios. Currently, applications of Value-at-Risk have grown in operations research literature far beyond finance to other areas where there is a need to make decisions in stochastic settings. For instance, VaR optimization has been applied in the context of portfolio allocation (Pflug, 2000; Gaivoronski and Pflug, 2005; Benati and Rizzi, 2007; Natarajan et al., 2008; Goh et al., 2012), in the facility location modeling ($\alpha-$reliable minimax regret problem Daskin et al. (1997); Chen et al. (2006)), in classification problems via support vector machine (Tsyurmasto et al., 2014), and even in regression analysis (least median of squares regression problem Rousseeuw (1984)). Thus, efficient general approaches for VaR optimization can positively impact a number of different application areas.

Mathematical programs involving Value-at-Risk generally are challenging problems. One of the main challenges of VaR minimization is the nonconvexity of the objective function, see for instance Gaivoronski and Pflug (2005) and Daníelsson et al. (2013) for specific examples. At the same time, feasible regions of such problems become unions of polyhedrons, which are generally nonconvex (Nemirovski and Shapiro, 2006; Qiu et al., 2014). As a result, it is known that chance-constrained problems are generally NP-hard, see for instance Benati and Rizzi (2007). Therefore, a large body of literature deals with heuristic solutions to these problems. To name a few, a number of optimization approaches is based on the approximations of the Value-at-Risk by the Conditional Value-at-Risk, (Larsen et al., 2002; Mausser and Romanko, 2014; Romanko and Mausser, 2016), which has the desired convexity property (Rockafellar and Uryasev, 2002), whereas Gaivoronski and Pflug (2005) proposed a smoothing method that aimed to filter out local nonconvexities. Exact solution approaches usually involve mixed integer programming with big M constants. As it is usual in such problems, these constants should be defined as big enough while at the same time as small as possible for computational tractability of problems. Recently, Feng et al. (2015) proposed a method for defining a set of big M constants, which significantly improved the efficiency of solving such problems. Moreover, Feng et al. (2015) applied results of Luedtke (2014) for general chance-constrained optimization problems to the particular case of VaR optimization problem and compared a number of solution approaches from computational point of view. Despite the advanced theoretical results obtained in Luedtke (2014), the authors concluded that MIP optimization with big M constants remained the most practical way for dealing with VaR minimization problems. Therefore, we treat such approach as the current benchmark for solving such problems to optimality and explore further ways to define even tighter MIP formulations with big M constants. Specifically, this paper incorporates bounds on the optimal value, lower and upper, into the optimization process and shows how simple considerations allow to significantly reduce the solution time compared to the benchmark. These bounds are relatively easy to obtain. For instance, an upper bound to the optimal value is provided by any heuristic procedure discussed in the literature, and the lower bound can be obtained by relaxing the corresponding MIP formulation. The objectives of this paper are outlined below:

- Introduce the problem in a very general manner that is suitable for a wide range of different applications.
- Tie together related but frequently isolated research efforts on heuristic algorithms and on lower bounds on the optimal value by showing how better bounds can be further

combined in a tighter MIP formulation and / or in a formulation with reduced number of binary variables and constraints.

- Demonstrate the importance of having accurate bounds on an optimal solution for obtaining tighter linear relaxations of the problem.
- Increase the efficiency of benchmarking of newly developing algorithms, allowing them to be benchmarked against larger instances solved to optimality.

2. Mixed Integer Programming Approach to VaR Minimization

Suppose there is a random variable of losses $\mathcal{L}$ that can be represented by some $L_1, \ldots, L_Q$ loss scenarios that occur with probabilities $p_j > 0$. Losses L_j , $j = 1, \ldots, Q$ are some linear functions of scenario information S_j and decision variables $\mathbf{x} \in X$, i.e., $L_j(\mathbf{x}) = f(S_j\,, \mathbf{x})$. In the settings of this paper, scenario information S_j is assumed to be represented by some vector. In addition, we assume that X is a compact convex set that can be represented via a set of linear constraints on $\mathbf{x}$. Alternatively, X might be a finite space of discrete decision variables, or a combination of both. Because of this assumption on X, we can define finite upper and lower bounds for loss L_j :

$$\overline{L}_j = \max_{\mathbf{x}\in X} f(S_j\,, \mathbf{x})\,, \qquad j = 1, \ldots, Q\,, \tag{2.1}$$

$$\underline{L}_j = \min_{\mathbf{x}\in X} f(S_j\,, \mathbf{x})\,, \qquad j = 1, \ldots, Q\,. \tag{2.2}$$

While obtaining the above bounds require solution of optimization problems that might involve binary or integer variables, we assume that these problems are relatively easy to solve. In addition, we would assume throughout this study that

$$\underline{L}_j < \overline{L}_j\,, \qquad j = 1, \ldots, Q\,. \tag{2.3}$$

By definition, the Value-at-Risk with confidence level $\alpha \in (0, 1]$ of the random variable representing loss $\mathcal{L}$ is defined as

$$\mathsf{VaR}_\alpha(\mathcal{L}) = \mathsf{inf}\{l \in \mathbb{R} : \mathbb{P}(\mathcal{L} > l) \le 1 - \alpha\}. \tag{2.4}$$

In terms of notation, we will use $\mathsf{VaR}_\alpha(\xi)$ or $\mathsf{VaR}_\alpha(\xi_1, \ldots, \xi_Q)$ interchangeably to denote the Value-at-Risk of a random variable ξ that takes values $\{\xi_1, \ldots, \xi_Q\}$ with probabilities $\{p_1, \ldots, p_Q\}$. However, sometimes $\{\xi_1, \ldots, \xi_Q\}$ will just mean a set of numbers, and corresponding random variable ξ can be constructed artificially. Before we proceed, let us list some important properties of the Value-at-Risk that would be used later in the paper.
Property 1. Notice that the inf in the definition (2.4) is attained at some outcome of ξ, i.e.,

$$\mathsf{VaR}_\alpha(\xi) = \xi_j \ \text{ for some } j \in \{1, \ldots, Q\}\,. \tag{2.5}$$

For $\alpha \in (0, 1)$, Pflug (2000) stated the following identity:

$$\mathsf{VaR}_\alpha(\xi) = -\mathsf{VaR}_{1-\alpha}(-\xi)\,. \tag{2.6}$$

While this identity is certainly true for random variables with probability densities, in a more general case that includes ξ with a discrete distribution, (2.6) is not necessary correct. In that

case this identity should be restated as follows.
Property 2.

$$\mathsf{VaR}_\alpha(\xi) \geq -\mathsf{VaR}_{1-\alpha+\epsilon}(-\xi)\,, \tag{2.7}$$
$$\mathsf{VaR}_\alpha(\xi) \leq -\mathsf{VaR}_{1-\alpha}(-\xi)\,, \tag{2.8}$$

where (2.7) holds true for any $\xi > 0$. However, we will be interested further in bounding $\mathsf{VaR}_\alpha(\xi)$ from below, hence ϵ needs to be as small as possible. In our settings, it can be defined to be less than any jump of the cumulative distribution function of the random variable ξ, for example as follows:

$$\epsilon = \frac{1}{2}\min_j \; p_j\,. \tag{2.9}$$

See an example demonstrating violation of (2.6) and a proof of statements (2.7) – (2.8) in the Appendix.

Another set of properties is a subset of important properties of general risk measures listed in Artzner et al. (1999). We list here only those applicable to VaR and which will be used further in the paper.
Property 3. Monotonicity. Let ξ_1 and ξ_2 denote two random variables representing losses. The VaR is monotone risk measure, i.e., if $\xi_1 \leq \xi_2$ with probability one, then

$$\mathsf{VaR}_\alpha(\xi_1) \leq \mathsf{VaR}_\alpha(\xi_2)\,. \tag{2.10}$$

Property 4. Translation invariance. For any constant $C \in \mathbb{R}$,

$$\mathsf{VaR}_\alpha(C + \xi) = C + \mathsf{VaR}_\alpha(\xi)\,. \tag{2.11}$$

The problem we are focusing on can be expressed as follows:

$$\min_{\mathbf{x}\in X} \; \mathsf{VaR}_\alpha(\mathcal{L}(\mathbf{x}))\,, \qquad \alpha \in (0,1)\,. \tag{2.12}$$

Let us introduce the variable l that is supposed to be equal to the VaR of $L_1(\mathbf{x}), \ldots, L_Q(\mathbf{x})$. Then, (2.12) can be represented as the following chance-constrained optimization problem:

$$\min_{\mathbf{x}\in X} \; l \tag{2.13}$$
subject to
$$\mathbb{P}(\mathcal{L}(\mathbf{x}) > l) \leq 1 - \alpha\,. \tag{2.14}$$

In order to determine $\mathbb{P}(\mathcal{L}(\mathbf{x}) > l)$, we introduce a set of indicator variables $z_j \in \{0, 1\}$, $j = 1, \ldots, Q$, such that

$$z_j = 1 \iff L_j(\mathbf{x}) > l\,, \tag{2.15}$$
$$z_j = 0 \iff L_j(\mathbf{x}) \leq l\,. \tag{2.16}$$

The following MIP formulation of the VaR minimization problem has appeared in the literature:

$$\min_{\mathbf{x}\in X} \quad l \tag{2.17}$$

subject to

$$Mz_j \geq L_j(\mathbf{x}) - l\,, \qquad j = 1, \ldots, Q\,, \tag{2.18}$$

$$\sum_{j=1}^{Q} p_j z_j \leq 1 - \alpha\,, \tag{2.19}$$

$$z_j \in \{0, 1\}\,, \qquad j = 1, \ldots, Q\,, \tag{2.20}$$

where M is a sufficiently large positive constant. For instance, the straightforward approach to define the constant M is as follows:

$$M = \max_{j \in \{1,\ldots,Q\}} \overline{L}_j - \min_{j \in \{1,\ldots,Q\}} \underline{L}_j\,. \tag{2.21}$$

Alternatively, the single constant M can be replaced by a set of constants M_j, $j = 1, \ldots, Q$, with each M_j being special for every particular scenario. For example, as follows:

$$M_j = \overline{L}_j - \min_{j \in \{1,\ldots,Q\}} \underline{L}_j\,, \qquad j = 1, \ldots, Q\,. \tag{2.22}$$

Note that the formulation (2.17) – (2.20) should only be applied when $1 - \alpha \geq \min_j \; p_j$, otherwise the corresponding $\mathsf{VaR}_\alpha(\mathcal{L}(\mathbf{x})) = \max_j \; L_j(\mathbf{x})$ and there exists a trivial linear programming minimax reformulation for that case.

In many papers, such as Daskin et al. (1997); Chen et al. (2006); Natarajan et al. (2008); Goh et al. (2012); Mausser and Romanko (2014); Feng et al. (2015), formulation (2.17) – (2.20), possibly with slight modifications, appeared as the standard approach to solve this problem, against which alternative approaches or heuristics had been tested. Further effort was made in a recent paper by Feng et al. (2015), where the authors described a number of decomposition algorithms for solving this problem. Besides that, the authors proposed a set of tighter constants $\widetilde{M}_j$, specific to the data in every particular scenario j and concluded that the big M approach remained the most practical way of solving such problems. The following procedure for $\widetilde{M}_j$ calculation was suggested in Feng et al. (2015). First, the maximum loss in scenario j in excess of the loss in scenario t is calculated:

$$d_t(j) = \max_{\mathbf{x} \in X} \; L_j(\mathbf{x}) - L_t(\mathbf{x})\,, \qquad t = 1, \ldots, Q\,. \tag{2.23}$$

With that, the following definition of M_j is proposed then:

$$\widetilde{M}_j = \mathsf{VaR}_{1-\alpha+\epsilon}\left(d_1(j), \ldots, d_Q(j)\right)\,, \quad j = 1, \ldots, Q\,, \tag{2.24}$$

where ϵ is defined according to (2.9). This result is quite interesting so we would like to revise the argument behind it; see also Propositions 3.2 – 3.3 in Feng et al. (2015). To show (2.24), we need to use Property 1, saying that any feasible l to (2.17) – (2.20) is in fact equal to the value of loss in some scenario t corresponding to decision variables $\mathbf{x}$, i.e., $l = L_t(\mathbf{x})$. Then, the constant $\widetilde{M}_j$ should be defined in a way that $\forall\, \mathbf{x} \in X$:

$$L_j(\mathbf{x}) - L_t(\mathbf{x}) \leq \widetilde{M}_j\,. \tag{2.25}$$

Given that, consider the following random variable $L_j(\mathbf{x}) - \mathcal{L}(\mathbf{x})$. Note that when $\mathbf{x}$ is fixed, then $L_j(\mathbf{x})$ is just a constant equal to the value of loss in scenario j. The Value-at-Risk with confidence level $1-\alpha+\epsilon$ of this random variable can be represented as

$$\begin{aligned}&\mathsf{VaR}_{1-\alpha+\epsilon}\left(L_j(\mathbf{x})-\mathcal{L}(\mathbf{x})\right)=L_j(\mathbf{x})+\mathsf{VaR}_{1-\alpha+\epsilon}\left(-\mathcal{L}(\mathbf{x})\right)\geq\\&L_j(\mathbf{x})-\mathsf{VaR}_{\alpha}\left(\mathcal{L}(\mathbf{x})\right),\end{aligned}\tag{2.26}$$

where we used Property 2 in the last transition and Property 4 in the first transition. At the same time,

$$\begin{aligned}&\mathsf{VaR}_{1-\alpha+\epsilon}\left(L_j(\mathbf{x})-\mathcal{L}(\mathbf{x})\right)=\mathsf{VaR}_{1-\alpha+\epsilon}\left(L_j(\mathbf{x})-L_1(\mathbf{x}),\ldots,L_j(\mathbf{x})-L_Q(\mathbf{x})\right)\leq\\&\mathsf{VaR}_{1-\alpha+\epsilon}\left(\max_{\mathbf{x}\in X}\,L_j(\mathbf{x})-L_1(\mathbf{x}),\ldots,\max_{\mathbf{x}\in X}\,L_j(\mathbf{x})-L_Q(\mathbf{x})\right)=\\&\mathsf{VaR}_{1-\alpha+\epsilon}\left(d_1(j),\ldots,d_Q(j)\right)=\widetilde{M}_j\,,\end{aligned}\tag{2.27}$$

where we used the monotonicity Property 3. Combining (2.26) and (2.27), we obtain that $\forall\,\mathbf{x}\in X:$

$$L_j(\mathbf{x})-\mathsf{VaR}_{\alpha}\left(\mathcal{L}(\mathbf{x})\right)\leq\widetilde{M}_j\,.\tag{2.28}$$

These constants will be used in the formulation (2.17) – (2.20), where a single M is replaced by a set of these constants. The authors concluded that this choice of $\widetilde{M}_j$ remained the most practical and efficient way of solving this problem. We do not call these constants as big Ms, since the definition (2.24) does not prevent them from being negative or zero. However, a nonpositive $\widetilde{M}_j$ in the constraint (2.18) should necessary imply that $L_j(\mathbf{x})\leq l$, which is a special case when the loss does not exceed the VaR and so that scenario can be safely removed from the optimization problem formulation. In the next section we will describe another way to define positive big M constants based on a different consideration.

3. Formulation Improvements

3.1. Using the Bounds on the Optimal Value

Further improvements of the problem formulation can be done by defining tighter constants M_j. However, before we proceed, we need to address situations when a scenario does not need defining a constant M_j. Recall that this constant should constraint the corresponding variable z_j to one when the loss $L_j(\mathbf{x})$ is greater that l. What if, for instance, we know that the optimal l is such that the loss in scenario j will always be greater than l. Alternatively, what if we can guess that the loss $L_j(\mathbf{x})$ can never exceed l; how can we incorporate this knowledge in a more accurate problem formulation? Such situations are possible when bounds on the optimal solution are available. Let l^* denote the optimal objective value of the optimization problem (2.17) – (2.20). Bounds on the value of l^* will be denoted as follows:

$$\underline{l}\leq l^*\leq\overline{l}\,,\qquad\underline{l}<\overline{l}\,.\tag{3.1}$$

With this information, we classify all the scenarios into four possible categories depending on the relations between upper and lower bounds on the optimal VaR and corresponding bounds on the loss in every particular scenario, $\overline{L}_j$ and $\underline{L}_j$. Let $j\in\{1,\ldots,Q\}$ and the following cases are possible:

0. $\overline{L}_j\leq\underline{l}$. In such case, for any $\mathbf{x}\in X$, the loss $L_j(\mathbf{x})$ will be less or equal than l, necessary resulting in the constraint

$$z_j=0\,.\tag{3.2}$$

1. $\bar{l} < \underline{L}_j$. In this case, for any $\mathbf{x} \in X$, the loss $L_j(\mathbf{x})$ will be greater than l, necessary resulting in the constraint
$$z_j = 1\,. \tag{3.3}$$
2. $\bar{l} > \underline{L}_j$ and $\overline{L}_j > \underline{l}$. Consider the loss expression $L_j(\mathbf{x}) - l$; it will be bounded from the above and below as follows:
$$\underline{L}_j - \bar{l} \leq L_j(\mathbf{x}) - l \leq \overline{L}_j - \underline{l}\,. \tag{3.4}$$
Note that the right side of the above inequality is strictly positive, and the left side is strictly negative because of the conditions for this case of classification. Therefore, if $L_j(\mathbf{x}) - l > 0$, then the corresponding z_j should be constrained to one, which can be expressed like this:
$$z_j \geq \frac{L_j(\mathbf{x}) - l}{\overline{L}_j - \underline{l}}\,. \tag{3.5}$$
At the same time, when $L_j(\mathbf{x}) - l \leq 0$, the corresponding z_j should equal to zero, and this requirement is constrained as follows:
$$z_j \leq 1 + \frac{L_j(\mathbf{x}) - l}{\bar{l} - \underline{L}_j}\,. \tag{3.6}$$
Notice that there is no conflict between (3.5) and (3.6): constraint (3.5) does not restrict corresponding z_j when $L_j(\mathbf{x}) - l \leq 0$ and (3.6) does not restrict z_j when $L_j(\mathbf{x}) - l > 0$.
3. $\bar{l} = \underline{L}_j$. In this case, for any $\mathbf{x} \in X$, the loss $L_j(\mathbf{x})$ will generally be greater than l, however, the equality is possible. Therefore, the variable z_j should be constrained to one, unless $L_j(\mathbf{x}) = \underline{L}_j$ and $l = \bar{l}$ which implies $L_j(\mathbf{x}) = l$ and that z_j should be equal to zero according to the definition of VaR. Thus, $z_j = 1$ whenever $L_j(\mathbf{x}) - \underline{L}_j > 0$ or $\bar{l} - l > 0$:
$$z_j \geq \frac{L_j(\mathbf{x}) - \underline{L}_j}{\overline{L}_j - \underline{L}_j}\,, \tag{3.7}$$
$$z_j \geq \frac{\bar{l} - l}{\bar{l} - \underline{l}}\,. \tag{3.8}$$

Thus, the Value-at-Risk minimization problem possesses a very interesting property: definition of constants M_j can be directly related to the question about the lower bound on the optimal VaR. In other words, if we can find a tight lower bound on the optimal VaR, then we can set smaller big M constants. Therefore, we suggest attacking the problem of setting tight big M constants from another end: how to establish a high-quality lower bound on the optimal solution first. We will address this question in Section 4.2. For now, assuming that the problem already has some upper and lower bounds, the problem formulation can be improved using these bounds as follows. Introduce the explicit notations for the lists of scenarios falling in every of four categories: $\mathcal{Q}_0 = \{j \in \{1,\ldots,Q\} \mid \overline{L}_j \leq \underline{l}\}$, $\mathcal{Q}_1 = \{j \in \{1,\ldots,Q\} \mid \bar{l} < \underline{L}_j\}$, $\mathcal{Q}_3 = \{j \in \{1,\ldots,Q\} \mid \bar{l} = \underline{L}_j\}$, and $\mathcal{Q}_2 = \{1,\ldots,Q\} \backslash \{\mathcal{Q}_1 \cup \mathcal{Q}_2 \cup \mathcal{Q}_3\}$. Note that we would prefer as many scenarios to be classified either to $\mathcal{Q}_0$, $\mathcal{Q}_1$ or $\mathcal{Q}_3$ since that essentially reduces the dimension of the problem, i.e., the number of binary variables and constraints. Still, in the least desirable case of this scenario classification, we end up with $\mathcal{Q}_2 = \{1,\ldots,Q\}$.

The scenario breakdown into $\mathcal{Q}_0$, $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ illustrates the importance of being able to find a high-quality heuristic solution, $\bar{l}$, and the lower bound $\underline{l}$, for obtaining an optimal solution. Specifically, the smaller is $\bar{l}$, the larger number of scenarios can possibly fall into the set $\mathcal{Q}_1$, thus tightening the relaxation of the problem formulation. At the same time, the larger is $\underline{l}$, the tighter constant can be obtained in constraint (3.5), again making tighter the relaxation of the problem. Moreover, with larger $\underline{l}$, more scenarios are possible to fall into the $\mathcal{Q}_0$ set, thus reducing the dimension of the problem formulation. These observations constitute the main advantage of VaR optimization problems versus VaR-constrained ones: both bounds on the optimal objective directly impact the quality of the MI problem formulation.

In addition, the existence of the upper bound $\bar{l}$ based on a heuristic solution procedure not only provides the bound, but also guarantees that the problem is feasible with the objective at most $\bar{l}$. In other words, we know that there exists a feasible solution $\mathbf{x}_0 \in X$, such that the corresponding Value-at-Risk of the losses equals $\bar{l}$. Therefore, the solver should be started from the "warm" feasible solution $\mathbf{x}_0$ for the best efficiency of the problem formulation (through the variable attribute *Start* in Gurobi Optimization (2015)).

Remark 2. While the constraint (3.6) does not improve the relaxation of the problem formulation, for some problems and some commercial MIP solvers, it makes the solver run faster. Most probably, this happens due to a different branching strategy employed by the solver for problems with two constraints. Still, in this study, this constraint will be omitted throughout the paper. In other words, the following problem formulation is also valid and will be used further:

$$\min_{\mathbf{x}\in X} \quad l \tag{3.9}$$

subject to

$$z_j \geq \frac{L_j(\mathbf{x}) - l}{\bar{L}_j - \underline{l}}\,, \qquad j \in \mathcal{Q}_2\,, \tag{3.10}$$

$$z_j \geq \frac{L_j(\mathbf{x}) - \underline{L}_j}{\bar{L}_j - \underline{L}_j}\,, \qquad j \in \mathcal{Q}_3\,, \tag{3.11}$$

$$z_j \geq \frac{\bar{l} - l}{\bar{l} - \underline{l}}\,, \qquad j \in \mathcal{Q}_3\,, \tag{3.12}$$

$$\sum_{j\in\mathcal{Q}_2\cup\mathcal{Q}_3} p_j z_j \leq 1 - \alpha - \sum_{j\in\mathcal{Q}_1} p_j\,, \tag{3.13}$$

$$\underline{l} \leq l \leq \bar{l}\,, \tag{3.14}$$

$$z_j \in \{0,1\}\,, \qquad j \in \mathcal{Q}_2 \cup \mathcal{Q}_3\,. \tag{3.15}$$

3.2. Valid Inequalities

The data obtained during the calculation of $\widetilde{M}_j$ values allows to establish a set of valid inequalities strengthening the linear relaxation of the corresponding mixed integer problem formulation. Specifically, recall that calculation of a single $\widetilde{M}_j$ requires first to find a set of values $d_t(j)$:

$$d_t(j) = \max_{\mathbf{x}\in X} \;\; L_j(\mathbf{x}) - L_t(\mathbf{x})\,, \qquad t = 1,\ldots,Q\,. \tag{3.16}$$

However, once $d_t(j) \leq 0$, then we can immediately conclude that $\forall \mathbf{x} \in X$:

$$L_j(\mathbf{x}) - L_t(\mathbf{x}) \leq 0 \implies L_j(\mathbf{x}) \leq L_t(\mathbf{x})\,.$$

Hence, according to the definition of z_j value (2.15) – (2.16), in such case z_j can not be equal to one without z_t equal to one. At the same time, z_t can not be equal to zero without z_j equal to zero. Let $E = \big\{(j\,,\,t) \;\big|\; j \neq t\,,\; d_t(j) \leq 0\,,\; j = 1,\ldots,Q\,,\, t = 1,\ldots,Q\big\}$. Therefore, we obtain the following set of strengthening inequalities:

$$z_j \leq z_t\,, \qquad\qquad (j\,,\,t) \in E\,. \tag{3.17}$$

The strengthening property of such inequalities comes from the observation that any linear relaxation problem can no longer have an optimal solution such that $z_t < z_j$. In Section 5 we run a separate set of experiments to demonstrate the efficiency of these inequalities.

4. Bounds on the Optimal Value

4.1. Upper Bound

As we pointed out in the introduction, there are a lot of heuristic approaches and any of them can provide a valid upper bound and be used in the MIP formulation. For instance, the simple and straightforward upper bound comes as a byproduct of the solution of the CVaR minimization problem (Rockafellar and Uryasev, 2000, 2002) and can be obtained as follows:

$$\bar{l} = \mathsf{arg}_c \left(\min_{\mathbf{x} \in X,\, c} \quad c + \frac{1}{1-\alpha} \sum_{j=1}^{Q} p_j h_j \right) \tag{4.1}$$

subject to

$$h_j \geq L_j(\mathbf{x}) - c\,, \qquad j = 1,\ldots,Q\,, \tag{4.2}$$

$$h_j \geq 0\,, \qquad j = 1,\ldots,Q\,. \tag{4.3}$$

As reported in some papers, e.g., Chen et al. (2006), such bound can sometimes be optimal to the VaR optimization problem as well. As noted in Pang and Leyffer (2004), the set $\mathsf{arg}_c \left(\min_{\mathbf{x} \in X,\, c} \quad c + \frac{1}{1-\alpha} \sum_{j=1}^{Q} p_j h_j \right)$ is likely to be an interval and since by definition we refer to VaR as the left side of this interval, a more accurate formulation of this bound should be written as follows:

$$\bar{l} = \min_{\mathbf{x} \in X} \quad c \tag{4.4}$$

subject to

$$c + \frac{1}{1-\alpha} \sum_{i=1}^{Q} p_j h_j \leq \min_{\mathbf{x} \in X} \mathsf{CVaR}_\alpha(L_1(\mathbf{x}), \ldots, L_Q(\mathbf{x}))\,, \tag{4.5}$$

$$h_j \geq L_j(\mathbf{x}) - c\,, \qquad j = 1,\ldots,Q\,, \tag{4.6}$$

$$h_j \geq 0\,, \qquad j = 1,\ldots,Q\,. \tag{4.7}$$

While the upper bound obtained through (4.4) – (4.7) is straightforward to find, it is not necessary the most efficient one. In our computational experiments we implement and employ a procedure based on one of two heuristic methods described in Larsen et al. (2002). This procedure was observed to be superior to the solution of (4.4) – (4.7). We briefly outline it below. The procedure iteratively minimizes Conditional Value-at-Risk with different values of the confidence level; first iteration starts with the confidence level α. Then, the VaR of an optimal solution $\mathbf{x}_0$ serves as an initial upper bound. In addition, the index $j_0 \in \{1, \ldots, Q\}$ with maximum loss $L_{j_0}(\mathbf{x}_0) = \operatorname*{argmax}_j L_j(\mathbf{x}_0)$ is obtained and the next CVaR minimization problem does not include scenario j_0 and solves the CVaR minimization problem with the confidence level $\dfrac{\alpha}{1 - p_{j_0}}$ and so on. The formal description of this procedure is as follows:

- Step 0. $\mathbf{x}_0 = \operatorname*{argmin}_{\mathbf{x}\in X} \mathsf{CVaR}_\alpha(\mathcal{L}(\mathbf{x}))$, $j_0 = \operatorname*{argmax}_{j\in\{1,\ldots,Q\}} L_j(\mathbf{x}_0)$.
- Step 1. $\mathbf{x}_1 = \operatorname*{argmin}_{\mathbf{x}\in X} \mathsf{CVaR}_{\frac{\alpha}{1-p_{j_0}}} \left(L_j(\mathbf{x}) \,\middle|\, j \in \{1, \ldots, Q\}\backslash\{j_0\}\right)$,
  $j_1 = \operatorname*{argmax}_{j\in\{1,\ldots,Q\}\backslash\{j_0\}} L_j(\mathbf{x}_1)$.
- Step 2. $\mathbf{x}_2 = \operatorname*{argmin}_{\mathbf{x}\in X} \mathsf{CVaR}_{\frac{\alpha}{1-p_{j_0}-p_{j_1}}} \left(L_j(\mathbf{x}) \,\middle|\, j \in \{1, \ldots, Q\}\backslash\{j_0\,, j_1\}\right)$,
  $j_2 = \operatorname*{argmax}_{j\in\{1,\ldots,Q\}\backslash\{j_0\,,j_1\}} L_j(\mathbf{x}_2)$.

  ...

  Repeat until step k when $\sum_{j=0}^{k} p_j \geq 1 - \alpha$ and exit with heuristic objective value $\bar{l} = \min_k \mathsf{VaR}_\alpha\left(\mathcal{L}(\mathbf{x}_k)\right)$ and a corresponding heuristic solution $\mathbf{x_t} : t \in \operatorname*{argmin}_k \mathsf{VaR}_\alpha\left(\mathcal{L}(\mathbf{x}_k)\right)$.

4.2. Lower Bound

Obtaining the lower bound is an iterative process. We start from a poor initial bound and that allows us to apply the improved problem reformulation described in Section 3. After that, the bound is improved iteratively. The initial lower bound on VaR is based on the concept of monotonicity property of VaR as a risk measure. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ denote two random variables representing losses. A risk measure $\mathcal{R}$ is called monotone (Artzner et al., 1999), if given that $\mathcal{L}_1 \leq \mathcal{L}_2$ a.s., then

$$\mathcal{R}(\mathcal{L}_1) \leq \mathcal{R}(\mathcal{L}_2)\,.$$

Notice that by definition the following inequalities hold $\forall\, \mathbf{x} \in X$:

$$L_j(\mathbf{x}) \geq \underline{L}_j\,, \qquad j = 1, \ldots, Q\,.$$

Therefore, due to the monotonicity property of VaR as the risk measure,

$$\mathsf{VaR}_\alpha\left(L_1(\mathbf{x}), \ldots, L_Q(\mathbf{x})\right) \geq \mathsf{VaR}_\alpha\left(\underline{L}_1, \ldots, \underline{L}_Q\right) = \underline{l}^0. \tag{4.8}$$

This initial bound $\underline{l}^0$ is very important since it is based just on the problem data and does not require solving any relaxation problem. Moreover, in many cases this bound is better than the bound based on the relaxation of the straightforward problem formulation (2.17) – (2.20). Most importantly, the initial lower bound also allows us to create a more accurate problem formulation (3.9) – (3.15) and then relax it.

The idea behind the iterative improvement of the lower bound is similar to one presented in Qiu et al. (2014). In order to obtain a potentially tighter lower bound, we use $\bar{l}$ and $\underline{l}^0$ to classify scenarios into $\mathcal{Q}_0^0\,, \mathcal{Q}_1\,, \mathcal{Q}_2^0\,, \mathcal{Q}_3$ classes and solve the relaxation of the (3.9) – (3.15) problem:

$$\underline{l}^1 = \min_{\mathbf{x}\in X} \quad l \tag{4.9}$$

subject to

$$z_j \geq \frac{L_j(\mathbf{x}) - l}{\overline{L}_j - \underline{l}^0}\,, \qquad j \in \mathcal{Q}_2^0\,, \tag{4.10}$$

$$z_j \geq \frac{L_j(\mathbf{x}) - \underline{L}_j}{\overline{L}_j - \underline{L}_j}\,, \qquad j \in \mathcal{Q}_3\,, \tag{4.11}$$

$$z_j \geq \frac{\overline{l} - l}{\overline{l} - \underline{l}^0}\,, \qquad j \in \mathcal{Q}_3\,, \tag{4.12}$$

$$\sum_{j \in \mathcal{Q}_2^0 \cup \mathcal{Q}_3} p_j z_j \leq 1 - \alpha - \sum_{j \in \mathcal{Q}_1} p_j\,, \tag{4.13}$$

$$\underline{l}^0 \leq l \leq \overline{l}\,, \tag{4.14}$$

$$z_j \in [0,1]\,, \qquad j \in \mathcal{Q}_2^0 \cup \mathcal{Q}_3\,. \tag{4.15}$$

Note that the term “relaxation” relates only to the relaxation of binary variables z_j that have to deal with the definition of the Value-at-Risk. Therefore, the relaxation that is solved might in fact be a mixed integer problem if $\mathbf{x}$ includes such variables. As before, we assume that this problem is relatively easy to solve, comparing to the Value-at-Risk minimization problem itself.

Clearly, $\underline{l}^1 \geq \underline{l}^0$, and if $\underline{l}^1 > \underline{l}^0$, we can tighten the set of constants $\overline{L}_j - \underline{l}^0 \to \overline{L}_j - \underline{l}^1$ and redefine classes $\mathcal{Q}_0^1$, $\mathcal{Q}_2^1$ based on the new lower bound $\underline{l}^1$ (only classes $\mathcal{Q}_0$ and $\mathcal{Q}_2$ require redefining at each iteration, since other two classes rely solely on the upper bound $\overline{l}$ that remains unchanged during the procedure). After that, we solve yet another relaxation of problem (3.9) – (3.15) using a tighter set of constants, in order to obtain a better lower bound $\underline{l}^2$. At each iteration, we obtain a new lower bound on l^* and this process will converge as a sequence of nondecreasing numbers bounded from the above (for instance, by $\overline{l}$). The process can be stopped when the difference between consecutive bounds becomes smaller than a predefined tolerance $\epsilon > 0$.

Note that the constant $\overline{L}_j - \underline{l}^k$ can now be compared with the value $\widetilde{M}_j$ defined by (2.24). These two constants are obtained based on rather different considerations: the $M_j = \overline{L}_j - \underline{l}^k$ assumes the variable l is an independent variable, while the approach behind (2.24) explicitly recognizes it to be one of $\{L_1(\mathbf{x}), \dots, L_Q(\mathbf{x})\}$, which is why corresponding $\widetilde{M}_j$ potentially can be more accurate and tighter. Indeed, in many practical cases, $\widetilde{M}_j < \overline{L}_j - \underline{l}^k$. Therefore, the iterative lower bound lifting procedure and constants (2.24) can be combined in a more efficient hybrid approach to initiate the second iterative lifting procedure. First, we identify the set of scenarios with nonpositive $\widetilde{M}_j$ that can be safely removed from the model since corresponding variables z_j are necessary set to zero. Let

$$\mathcal{M}^+ = \{j \in \{1, \dots, Q\} : \widetilde{M}_j > 0\}\,, \tag{4.16}$$

$$\mathcal{M}^- = \{j \in \{1, \dots, Q\} : \widetilde{M}_j \leq 0\}\,. \tag{4.17}$$

The next $k{+}1^{th}$ lower bound will be obtained by solving an improved relaxation problem, where we remove scenarios $\mathcal{M}^-$, and use the best of two constants, $\widetilde{M}_j$ or $\overline{L}_j - \underline{l}^k$, $j \in \left\{\mathcal{Q}_2^k \cup \mathcal{Q}_3\right\} \cap \mathcal{M}^+$ to find a better lower bound:

$$\underline{l}^{k+1} = \min_{\mathbf{x} \in X} \quad l \tag{4.18}$$

subject to

$$z_j \geq \frac{L_j(\mathbf{x}) - l}{\min\left(\overline{L}_j - \underline{l}^k,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_2^k \cap \mathcal{M}^+\,, \tag{4.19}$$

$$z_j \geq \frac{L_j(\mathbf{x}) - \underline{L}_j}{\min\left(\overline{L}_j - \underline{L}_j\,,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_3 \cap \mathcal{M}^+\,, \tag{4.20}$$

$$z_j \geq \frac{\bar{l} - l}{\min\left(\bar{l} - \underline{l}^k,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_3 \cap \mathcal{M}^+\,, \tag{4.21}$$

$$\sum_{j \in \left\{\mathcal{Q}_2^k \cup \mathcal{Q}_3\right\} \cap \mathcal{M}^+} p_j z_j \leq 1 - \alpha - \sum_{j \in \mathcal{Q}_1} p_j\,, \tag{4.22}$$

$$\underline{l}^k \leq l \leq \bar{l}\,, \tag{4.23}$$

$$z_j \in [0, 1]\,, \qquad j \in \left\{\mathcal{Q}_2^k \cup \mathcal{Q}_3\right\} \cap \mathcal{M}^+\,. \tag{4.24}$$

Consequently, the new iterative procedure can be started if $\underline{l}^{k+1} - \underline{l}^k \geq \epsilon$. The value which the second iterative process converges to is called the final lower bound and will be referred to as $\underline{l}$. In the next section, we demonstrate possible advancements we can achieve, i.e., the number of scenarios that can be removed from the problem and compare the quality of big M constants obtained by two approaches. Note that the above relaxation problem can be strengthened by the set of additional inequalities (3.17) that can now be simplified to the following:

$$z_j \leq z_t\,, \qquad (j\,,\, t) \in E\,;\ j\,,\, t \in \left\{\mathcal{Q}_2^k \cup \mathcal{Q}_3\right\} \cap \mathcal{M}^+. \tag{4.25}$$

Another observation can be made here. The set $\mathcal{Q}_3 \cap \mathcal{M}^+$ should generally coincide with $\mathcal{Q}_3$. Otherwise, this results in a criterion that the heuristic procedure that has been employed to obtain $\bar{l}$ in fact optimally solved the problem. The following proposition formalizes this statement.

Proposition 4.1. *Suppose $\{\mathcal{Q}_3 \cap \mathcal{M}^+\} \neq \mathcal{Q}_3$. Let $j \in \mathcal{Q}_3$, $j \notin \{\mathcal{Q}_3 \cap \mathcal{M}^+\}$. Then, the optimal value of the VaR optimization problem is obtained: $l^* = \bar{l}$.*

Proof. The condition $j \in \mathcal{Q}_3\,,\, j \notin \{\mathcal{Q}_3 \cap \mathcal{M}^+\}$ implies that $j \in \mathcal{M}^-$. We obtain then:

$$j \in \mathcal{Q}_3 \implies \bar{l} = \underline{L}_j \implies l^* \leq \underline{L}_j\,, \tag{4.26}$$

$$j \in \mathcal{M}^- \implies L_j(\mathbf{x}) \leq l^* \implies \underline{L}_j \leq l^*\,. \tag{4.27}$$

Combining (4.26) and (4.27), the statement of the proposition follows. □

5. Computational Experiments

The objective of this computational study is to demonstrate that the simple improvements to the problem formulation can significantly speed up the solution time. The portfolio optimization problem is selected to be the underlying problem to test different VaR minimization approaches. Mainly, it was selected due to its simplicity. We describe the problem first.

Suppose there is a set of n financial instruments, having some joint distribution of returns. We assume that the joint distribution of returns is discrete and is represented by a set of Q scenarios, which are assumed to be equally probable, i.e., $p_j = 1/Q\,,\ j = 1, \ldots, Q$. Thus, the underlying data for an experiment can be represented by a matrix of instrument returns $R = \{r_i^j\} \in \mathbb{R}^{Q\times n}$. A portfolio of these instruments can be created: let x_i be a fraction of the budget to be allocated in the instrument i. We would assume no short positions, therefore, the decision space is a compact convex set, $X = \{\mathbf{x} \in \mathbb{R}^n \Big| \sum_{i=1}^{n} x_i = 1\,,\ x_i \geq 0\,,\ i = 1,\dots,n\}$. For every scenario j, the loss $L_j(\mathbf{x})$ is then defined as follows:

$$L_j(\mathbf{x}) = -\sum_{i=1}^{n} r_i^j x_i\,, \qquad j = 1,\dots,Q\,. \tag{5.1}$$

Substituting the above definition of loss into the VaR optimization formulation (3.9) – (3.14), we obtain the minimum VaR portfolio optimization problem. In some papers, i.e., Goh et al. (2012), this problem is defined with an additional constraint on the minimum expected return of the desired portfolio:

$$\sum_{j=1}^{Q} p_j \sum_{i=1}^{n} r_i^j x_i \geq r\,. \tag{5.2}$$

In our experiments, we consider this problem without such a constraint, mainly due to ambiguity of definition of constant r in such artificially created problems. Here is another reason: consider two decision spaces, X and $X_1 = \{\mathbf{x} \in X \Big| A\mathbf{x} \leq \mathbf{b}\}$, where $A\mathbf{x} \leq \mathbf{b}$ may correspond to (5.2) or be something different. Recall the definitions of lower and upper bounds on the loss in scenario j:

$$\overline{L}_j = \max_{\mathbf{x}\in X_1} f(S_j,\mathbf{x}) \leq \max_{\mathbf{x}\in X} f(S_j,\mathbf{x})\,, \tag{5.3}$$

$$\underline{L}_j = \min_{\mathbf{x}\in X_1} f(S_j,\mathbf{x}) \geq \min_{\mathbf{x}\in X} f(S_j,\mathbf{x})\,. \tag{5.4}$$

These definitions show that the loss bounds do depend on the solution space and are easy to be manipulated. Moreover, it is clear that imposing additional constraints to the decision space, we obtain a problem with a smaller decision space to begin with, and in an extreme case this may result in a significant simplification of the VaR minimization problem, only due to the way how we defined the problem. We would like to avoid that possibility, testing problem formulations under the most general conditions.

Note that upper and lower bounds, (2.2) and (2.1), for this portfolio optimization problem are extremely easy to compute using the matrix of returns:

$$\overline{L}_j = \max_i \ -r_i^j\,, \qquad\qquad j = 1,\dots,Q\,,$$

$$\underline{L}_j = \min_i \ -r_i^j\,, \qquad\qquad j = 1,\dots,Q\,.$$

The data employed in the computational study represent daily stock prices for ten aerospace companies from January 1988 through October 1991, obtained from the KEEL dataset repository (can be downloaded from http://sci2s.ugr.es/keel/dataset.php?cod=77, Alcalá et al. (2010)). The dataset contains 950 vectors of daily stock prices (949 vectors of daily returns) that were split to generate the following three test instances:

- Dataset 1: contains the first 475 scenarios of returns of the original dataset.
- Dataset 2: contains the remaining 474 scenarios of returns of the original dataset.

- Dataset 3: contains all 949 scenarios of returns.

The computational results presented below were obtained using a machine equipped with Windows 8.1x64 operating system, Intel Core(TM) i5-4200M CPU 2.5GHz processor, 6GB RAM, and Gurobi 6.5.1 64-bit solver, Gurobi Optimization (2015). Optimization models were implemented using 64-bit Python 3.5.1. Before we proceed, we demonstrate separately the efficiency of the iterative lower bound lifting procedure proposed in Section 4.2. Tables 1 and 3 provide few examples of how the procedure can improve the lower bound for problem instances based on Datasets 1 and 2, respectively, with or without using the heuristic. In the majority of presented cases, the iterative procedure leads to a remarkable increase of the lower bound. The improvement of the lower bound is best to analyze comparing the bound value $\underline{l}^1$, which is the lower bound after solving the relaxation problem just once and the bound obtained after k iterations. Moreover, the benefit of having a good heuristic procedure allowing some scenarios to be classified as $\mathcal{Q}_1$ for obtaining a better lower bound is observable as well. As we can see, such preprocessing requires rather neglidgeable amount of time, especially compared to the potential time improvement of the MI problem execution. Finally, Tables 2 and 4 illustrate how the results of the second lower bound lifting procedure can be improved if the strengthening constraints (4.25) are incorporated in the corresponding relaxation problem formulation.

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	170/250		220/300		270/350		340/400		390/450	
$\bar{l}$	1.240	–	2.419	–	3.174	–	5.947	–	6.162	–
$\underline{l}^0$	−12.552	−12.552	−11.494	−11.494	−9.434	−9.434	−6.061	−6.061	−5.714	−5.714
$\underline{l}^1$	−10.288	−10.949	−8.369	−8.937	−6.553	−6.858	−3.175	−3.319	−2.416	−2.551
$\underline{l}^2$	−9.524	−10.365	−7.445	−8.125	−5.777	−6.124	−2.575	−2.735	−1.771	−1.922
$\underline{l}^3$	−9.262	−10.152	−7.167	−7.866	−5.563	−5.911	−2.445	−2.606	−1.638	−1.789
$\underline{l}^4$	−9.173	−10.074	−7.081	−7.783	−5.502	−5.850	−2.416	−2.578	−1.610	−1.761
$\underline{l}^5$	−9.142	−10.045	−7.055	−7.756	−5.484	−5.832	−2.410	−2.571	−1.605	−1.755
$\underline{l}^6$	−9.132	−10.035	−7.047	−7.747	−5.479	−5.826	−2.408	−2.570	−1.603	−1.753
$\underline{l}^7$	−9.128	−10.031	−7.044	−7.744	−5.478	−5.825	–	−2.569	–	–
$\underline{l}^8$	−9.127	−10.029	−7.043	−7.743	−5.477	−5.824	–	–	–	–
$\underline{l}^9$	−9.126	–	–	–	–	–	–	–	–	–
$\underline{l}^{10}$	−6.996	−7.429	−5.300	−5.673	−3.942	−4.132	−1.214	−1.313	−0.445	−0.545
$\underline{l}^{11}$	−6.970	−7.411	−5.272	−5.650	−3.922	−4.112	−1.205	−1.306	−0.441	−0.543
$\underline{l}^{12}$	−6.969	−7.410	−5.271	−5.649	–	–	–	–	–	–
$\lvert\mathcal{Q}_1\rvert$	11	0	10	0	7	0	3	0	3	0
$\lvert\mathcal{Q}_0^{12}\rvert$	0	0	3	0	5	5	16	16	17	17
$\lvert\mathcal{M}^-\rvert$	7	7	12	12	20	20	35	35	42	42
$\lvert\mathcal{Q}_0^{12}\cup\mathcal{M}^-\rvert$	7	7	12	12	20	20	35	35	42	42
$\#j$	8	6	16	15	15	15	16	15	12	11
CPU (sec.)	5.76	3.36	8.74	5.06	12.35	7.73	16.10	11.30	21.30	15.89

Table 1: Iterative tightening of the lower bound, tested using problem instances based on Dataset 1. For every problem instance, results for two iterative procedures are presented: the left side starts from the heuristic procedure and uses the corresponding upper bound for scenario classification and the right side presents lower bound iteration without using that upper bound. Iterations 1 to 9 correspond to the first lifting procedure and iterations 10 to 12 represent results of the second procedure. Both iterative lower bound lifting procedures stop when the difference between bounds in two subsequent iterations becomes smaller than 10^{-7}. $\#j := \left|j \in \{\mathcal{Q}_2^{12} \cup \mathcal{Q}_3\} \cap \mathcal{M}^+ : 0 < \overline{L}_j - \underline{l}^{12} < \widetilde{M}_j\right|$ denotes the number of scenarios where the big M value $L_j - \underline{l}^{12}$ based on the best lower bound on the optimal value is smaller than $\widetilde{M}_j$. Upper and lower bounds are scaled with coefficient 10^3 for ease of presentation.

Another important question is to compare the efficiency of the preprocessing stage described in Tables 1 and 3 with the MIP solver capabilities within an equivalent amount of CPU time.

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	170/250		220/300		270/350		340/400		390/450	
$\bar{l}$	1.240	–	2.419	–	3.174	–	5.947	–	6.162	–
$\underline{l}^{10}$	−6.937	−7.276	−5.256	−5.554	−3.893	−4.053	−1.192	−1.262	−0.431	−0.498
$\underline{l}^{11}$	−6.903	−7.248	−5.218	−5.526	−3.867	−4.027	−1.180	−1.252	−0.427	−0.495
$\underline{l}^{12}$	−6.901	−7.247	−5.216	−5.525	−3.866	−4.026	–	–	–	–
$\lvert\mathcal{Q}_1\rvert$	11	0	10	0	7	0	3	0	3	0
$\lvert\mathcal{Q}_0^{12}\rvert$	0	0	3	2	5	5	16	16	17	17
$\lvert\mathcal{M}^-\rvert$	7	7	12	12	20	20	35	35	42	42
$\lvert\mathcal{Q}_0^{12}\cup\mathcal{M}^-\rvert$	7	7	12	12	20	20	35	35	42	42
$\# j$	9	6	16	16	15	16	16	15	13	12

Table 2: Addendum to Table 1. Results for the second iterative lower bound lifting procedure are presented where relaxation problems are strengthened by a set of constraints (4.25).

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	185/250		240/300		290/350		350/400		400/450	
$\bar{l}$	4.097	–	5.936	–	7.658	–	9.75	–	9.878	–
$\underline{l}^{0}$	−15.021	−15.021	−13.187	−13.187	−11.905	−11.905	−3.401	−9.434	−8.639	−8.639
$\underline{l}^{1}$	−10.328	−10.729	−7.792	−8.122	−6.213	−6.528	−2.233	−3.532	−2.459	−2.586
$\underline{l}^{2}$	−9.061	−9.507	−6.508	−6.859	−4.931	−5.264	−1.986	−2.377	−1.319	−1.452
$\underline{l}^{3}$	−8.700	−9.155	−6.185	−6.532	−4.623	−4.951	−1.933	−2.129	−1.085	−1.223
$\underline{l}^{4}$	−8.597	−9.050	−6.101	−6.446	−4.548	−4.872	−1.921	−2.076	−1.037	−1.174
$\underline{l}^{5}$	−8.567	−9.018	−6.079	−6.423	−4.529	−4.852	−1.919	−2.064	−1.027	−1.163
$\underline{l}^{6}$	−8.558	−9.009	−6.074	−6.417	−4.524	−4.846	−1.918	−2.062	−1.024	−1.161
$\underline{l}^{7}$	−8.556	−9.006	−6.072	−6.415	−4.523	−4.845	–	−2.061	–	–
$\underline{l}^{8}$	−8.555	−9.005	–	−6.415	–	–	–	–	–	–
$\underline{l}^{9}$	–	−9.004	–	–	–	–	–	–	–	–
$\underline{l}^{10}$	−6.753	−7.025	−4.446	−4.679	−3.096	−3.305	−0.713	−0.808	0.153	0.059
$\underline{l}^{11}$	−6.713	−6.981	−4.412	−4.640	−3.061	−3.267	−0.677	−0.772	0.190	0.094
$\underline{l}^{12}$	−6.712	−6.979	−4.410	−4.639	−3.059	−3.265	−0.676	−0.771	0.192	0.096
$\lvert\mathcal{Q}_1\rvert$	6	0	5	0	5	0	2	0	2	0
$\lvert\mathcal{Q}_0^{12}\rvert$	2	2	2	2	2	2	3	3	7	7
$\lvert\mathcal{M}^-\rvert$	2	2	2	2	5	5	7	7	10	10
$\lvert\mathcal{Q}_0^{12}\cup\mathcal{M}^-\rvert$	3	3	3	3	5	5	7	7	11	11
$\# j$	16	16	22	19	37	31	34	32	60	58
CPU (sec.)	6.47	4.28	8.02	5.16	11.53	7.96	15.20	11.36	21.07	16.01

Table 3: Iterative tightening of the lower bound, tested using problem instances based on Dataset 2. For every problem instance, results for two iterative procedures are presented: the left side starts from the heuristic procedure and uses the corresponding upper bound for scenario classification and the right side presents lower bound iteration without using that upper bound. Iterations 1 to 9 correspond to the first lifting procedure and iterations 10 to 12 represent results of the second procedure. Both iterative lower bound lifting procedures stop when the difference between bounds in two subsequent iterations becomes smaller than 10^{-7}. $\# j := \left| j \in \{\mathcal{Q}_2^{12} \cup \mathcal{Q}_3\} \cap \mathcal{M}^+ : 0 < \overline{L}_j - \underline{l}^{12} < \widetilde{M}_j \right|$ denotes the number of scenarios where the big M value $L_j - \underline{l}^{12}$ based on the best lower bound on the optimal value is smaller than $\widetilde{M}_j$. Upper and lower bounds are scaled with coefficient 10^3 for ease of presentation.

Specifically, finding a heuristic solution and running two iterative lower bound lifting procedures takes time and provides the bounds on the optimal VaR to construct a more efficient MIP formulation. Instead, the same time could simply be used to run the MIP formulation (2.17) – (2.20) with some values of M_j for the same amount of time that preprocessing takes, and obtain bounds on the optimal VaR as well. Table 5 compares the quality of bounds obtained by two approaches and provides the number of solved relaxations (or CVaR minimization problems) in both cases. It is observable that the MIP solver generally manages to find a better lower

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	185/250		240/300		290/350		350/400		400/450	
$\bar{l}$	4.097	–	5.936	–	7.658	–	9.75	–	9.878	–
$\underline{l}^{10}$	−6.736	−6.981	−4.424	−4.644	−3.090	−3.299	−0.713	−0.808	0.153	0.060
$\underline{l}^{11}$	−6.691	−6.930	−4.390	−4.604	−3.053	−3.257	−0.677	−0.769	0.190	0.099
$\underline{l}^{12}$	−6.689	−6.928	−4.389	−4.602	−3.052	−3.255	−0.676	−0.768	0.192	0.102
$\underline{l}^{13}$	−6.688	–	–	–	–	–	–	–	–	–
$\lvert\mathcal{Q}_1\rvert$	6	0	5	0	5	0	2	0	2	0
$\lvert\mathcal{Q}_0^{13}\rvert$	2	2	2	2	2	2	3	3	7	7
$\lvert\mathcal{M}^-\rvert$	2	2	2	2	5	5	7	7	10	10
$\lvert\mathcal{Q}_0^{13}\cup\mathcal{M}^-\rvert$	3	3	3	3	5	5	7	7	11	11
$\# j$	16	16	23	19	37	32	34	32	60	58

Table 4: Addendum to Table 3. Results for the second iterative lower bound lifting procedure are presented where relaxation problems are strengthened by a set of constraints (4.25).

Dataset	Q	α	Preprocessing				Gurobi			
			CPU	$\bar{l}$	$\underline{l}$	# nodes	CPU	$\bar{l}$	$\underline{l}$	# nodes
1	200	120/200	3.56	−0.337	−10.277	97	3.73	0.050	−9.411	1,167
	250	170/250	5.86	1.240	−6.901	95	6.06	1.948	−5.781	2,287
	300	220/300	8.53	2.419	−5.216	95	8.79	2.910	−4.565	1,140
	350	270/350	12.12	3.174	−3.866	94	12.43	3.060	−3.280	741
	400	340/400	15.46	5.947	−1.180	72	15.83	5.993	−2.285	750
	450	390/450	21.14	6.162	−0.427	71	21.56	6.472	0.606	743
	475	410/475	24.95	6.389	−0.413	76	25.37	6.363	0.372	739
	475	450/475	21.06	11.117	4.889	37	21.38	10.203	7.238	2,159
2	200	140/200	3.21	3.006	−8.137	75	3.35	2.700	−6.152	2,015
	250	185/250	5.38	4.097	−6.688	80	5.54	4.058	−4.754	2,258
	300	240/300	7.91	5.936	−4.389	74	8.14	6.407	−2.848	1,276
	350	290/350	11.48	7.658	−3.052	73	11.72	7.186	−1.511	1,620
	400	350/400	16.01	9.750	−0.676	63	16.31	8.677	1.490	1,795
	450	400/450	20.35	9.878	0.192	64	20.72	9.715	2.374	1,326
	474	420/474	23.62	10.111	0.069	66	24.02	9.356	1.926	1,502
	474	450/474	20.83	14.054	5.644	37	21.17	12.985	8.050	1,071

Table 5: Comparison of the bounds after the preprocessing stage (heuristic plus lower bound lifting) versus the MIP solver for an equivalent CPU time. Computational results for problem instances based on Datasets 1, 2 with $n = 10$ are presented.

bound within the same time, at the cost of solving a much greater number of relaxations. Yet, the heuristic procedure can sometimes provide a better upper bound. As a result of these observations, we note that the MIP solver solely can successfully replace the preprocessing procedure and be used to set the initial bounds and formulate the problem. This conclusion results in a corresponding solution method, called the "two-stage" solution approach:

- The "two-stage" approach splits the solution process in two stages. In the first stage we start from some MIP formulation of the VaR minimization problem and launch the solver only for a fixed number of nodes in the branch-and-bound tree, say N. When the the first stage finishes, we may obtain a new feasible solution providing a new upper bound, together with a new lower bound on the optimal VaR. With these new bounds, the scenario classification procedure is performed; big M constants of (4.9) – (4.15) can potentially be tightened and the obtained lower bound can be lifted via the iterative procedure. Then, the tightened problem (4.9) – (4.15), potentially with a reduced number of scenarios and corresponding binary variables, is warm started from the solution obtained at the end of the first stage, and is being run to optimality. When we present the solution time of this approach, we count the overall time that took us to run the preprocessing and both MIP stages.

In this section, the following three problem formulations are identified and will be compared; the "two-stage" approach will further be applied to all of them:

- M1 (benchmark). The problem formulation (2.17) – (2.20) with the set of constants $\widetilde{M}_j$, calculated according to Feng et al. (2015).
- M2. This approach requires preprocessing. First, the heuristic solution procedure described in Section 4.1 is launched to obtain an upper bound $\bar{l}$. Second, two iterative lifting procedures from Section 4.2 are executed to obtain a lower bound. Big M_j for every scenario $j \in \mathcal{M}^+$ equals to the minimum of $\widetilde{M}_j$ and $\overline{L}_j - \underline{l}$; the resulting MIP formulation is the non-relaxed version of (4.9) – (4.15):

$$\min_{\mathbf{x}\in X} \quad l \tag{5.5}$$

subject to

$$z_j \geq \frac{L_j(\mathbf{x}) - l}{\min\left(\overline{L}_j - \underline{l}\,,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_2 \cap \mathcal{M}^+\,, \tag{5.6}$$

$$z_j \geq \frac{L_j(\mathbf{x}) - \underline{L}_j}{\min\left(\overline{L}_j - \underline{L}_j\,,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_3 \cap \mathcal{M}^+\,, \tag{5.7}$$

$$z_j \geq \frac{\bar{l} - l}{\min\left(\bar{l} - \underline{l}\,,\, \widetilde{M}_j\right)}\,, \qquad j \in \mathcal{Q}_3 \cap \mathcal{M}^+\,, \tag{5.8}$$

$$\sum_{j\in\{\mathcal{Q}_2\cup\mathcal{Q}_3\}\cap\mathcal{M}^+} p_j z_j \leq 1 - \alpha - \sum_{j\in\mathcal{Q}_1} p_j\,, \tag{5.9}$$

$$\underline{l} \leq l \leq \bar{l}\,, \tag{5.10}$$

$$z_j \in \{0,1\}\,, \qquad j \in \{\mathcal{Q}_2 \cup \mathcal{Q}_3\} \cap \mathcal{M}^+\,. \tag{5.11}$$

  Finally, the problem formulation should be "warm" started from the heuristic solution corresponding to the obtained upper bound $\bar{l}$.

- M3. The third approach required the same preprocessing as M2, where the problem formulation (5.5) – (5.11) is strengthened by a set of additional constraints (4.25). Note that the size of constraint pool (4.25) can be substantially larger than the number of scenarios and in order to decrease the computational effort due to increased size of relaxation problem, we move that set of constraints to the pool of *Lazy* constraints with parameter 3 (constraint attribute *Lazy* in Gurobi Optimization (2015)). Specifying a constraint as a *Lazy* one implies that it is not incorporated in the model until a feasible solution is found and violates it; when the corresponding parameter is set to 3, incorporation of such a constraint is more aggressive and occurs even when violation happens at the relaxation stage. Such approach ensures than only those valid inequalities that improve the quality of relaxation are included in the model.

Two-stage solution modifications of the above described formulations will be denoted by M1′, M2′ and M3′.

Prior to proceeding to computational experiments comparing performance of M1 – M3 methods and their two-stage counterparts, we evaluate the results of the "first stage" of the two-stage approach. Let the first stage be considered for the M3 formulation. Specifically, we are interested in the number of scenarios that can be classified as $\mathcal{Q}_0$ and $\mathcal{Q}_1$ after the first stage of M3 (solving M3 for only N nodes of the branch-and-bound tree), together with the number of scenarios where big M can be improved compared to (2.24). Tables 6 and 7 reveal that roughly $5-13\%$ of scenarios (and corresponding binary variables) can be removed from the problem formulation in the end of the first stage, while big Ms for over one third of remaining scenarios can be tightened for the final optimization in the second stage. Most importantly, such results were obtained within two minutes of computer time, which is an acceptable time limit taking into account the overall time required to solve the considered problems to optimality.

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	170/250		220/300		270/350		340/400		390/450	
$\bar{l}$	1.160	1.183	2.166	2.195	3.052	2.989	5.640	5.694	6.028	6.096
$\underline{l}^{13}$	−2.259	−2.186	−0.991	−1.317	0.128	−0.011	3.559	3.044	3.520	3.871
$\lvert\mathcal{Q}_1\rvert$	11	11	10	10	7	8	4	4	3	3
$\lvert\mathcal{Q}_0^{13}\rvert$	7	11	15	15	32	16	48	45	54	55
$\lvert\mathcal{M}^-\rvert$	7	7	12	12	20	20	35	35	42	42
$\lvert\mathcal{Q}_0^{13}\cup\mathcal{M}^-\rvert$	12	15	20	20	33	24	50	48	59	59
$\# j$	105	107	109	105	149	113	156	146	157	168
CPU (sec.)	24.5	24.5	39.0	30.6	39.7	44.7	51.6	49.4	65.0	50.1

Table 6: First stage results of the M3 solution procedure, tested using problem instances based on Dataset 1. $N = 100,000$ nodes in all experiments. Upper and lower bounds are scaled with a coefficient 10^3 for ease of presentation. $\# j := \left| j \in \{\mathcal{Q}_2^{13} \cup \mathcal{Q}_3\} \cap \mathcal{M}^+ : 0 < \overline{L}_j - \underline{l}^{13} < \widetilde{M_j}\right|$ denotes the number of scenarios where the big M value $\underset{\sim}{L_j - \underline{l}^{13}}$ based on the best lower bound on the optimal value after the first stage is smaller than $\widetilde{M_j}$.

Q	250		300		350		400		450	
n	10		10		10		10		10	
α	185/250		240/300		290/350		350/400		400/450	
$\bar{l}$	3.444	3.386	5.182	5.523	6.309	6.861	8.554	8.522	9.141	9.141
$\underline{l}^{14}$	−0.866	−0.943	1.783	1.813	2.809	3.191	5.239	5.015	5.657	5.674
$\lvert\mathcal{Q}_1\rvert$	8	8	6	6	5	5	2	2	2	2
$\lvert\mathcal{Q}_0^{14}\rvert$	3	3	4	4	8	11	18	16	25	25
$\lvert\mathcal{M}^-\rvert$	2	2	2	2	5	5	7	7	10	10
$\lvert\mathcal{Q}_0^{14}\cup\mathcal{M}^-\rvert$	3	3	4	4	9	11	18	16	26	26
$\# j$	113	111	146	146	155	169	182	178	190	191
CPU (sec.)	36.6	24.9	50.1	31.4	56.0	37.9	48.7	43.1	72.6	67.8

Table 7: First stage results of the M3 solution procedure, tested using problem instances based on Dataset 2. $N = 100,000$ nodes in all experiments. Upper and lower bounds are scaled with a coefficient 10^3 for ease of presentation. $\# j := \left| j \in \{\mathcal{Q}_2^{14} \cup \mathcal{Q}_3\} \cap \mathcal{M}^+ : 0 < \overline{L}_j - \underline{l}^{14} < \widetilde{M_j}\right|$ denotes the number of scenarios where the big M value $\underset{\sim}{L_j - \underline{l}^{14}}$ based on the best lower bound on the optimal value after the first stage is smaller than $\widetilde{M_j}$.

Tables 8, 9 and 10 present overall computational results comparing CPU times of M1 – M3 solution approaches and their two-stage counterparts. The tables demonstrate that an extra effort in defining better bounds on the optimal value of VaR in the form of preprocessing and the first stage of the two-stage approach can significantly decrease the overall solution time of hard problem instances, by up to 75% compared to the benchmark approach M1. Moreover, this observation encourages further research efforts in the direction of improving the bounds on the optimal VaR, possibly based on different considerations and under more specific assumptions on the decision space X. The test instances considered in Tables 8 – 10 may look much smaller in size than those in the previous literature, e.g., Feng et al. (2015), however, we considered much denser matrix of returns, with 89.4% nonzero elements (this estimate corresponds to the entire dataset, Dataset 3), compared to the matrices in Feng et al. (2015) with only $8-30\%$ of nonzero elements. Moreover, we considered VaR optimization problems with a broader range of confidence levels α, not only 99.5% that is commonly used in financial industry and employed in Feng et al. (2015). For example, for the least median of squares regression problem (Rousseeuw, 1984) there is a need to minimize Value-at-Risk with the 50% confidence level. These settings, i.e., dense return matrices and smaller confidence levels, generally make the problem more difficult to solve (Qiu et al., 2014) and correspond to the objective of our study: demonstrate the effectiveness of few simple improvements to the standard solution approach on an arbitrary set of test cases, rather than attempting to demonstrate solutions of certain real-life financial industry problems.

Q	N	α	Obj	CPU (sec.)					
				M1	M1′	M2	M2′	M3	M3′
200	15,000	120/200	−1.050	609.7	1,959.7	2,857.2	1,020.7	1,158.9	779.3
250	20,000	170/250	1.137	3,239.5	1,748.0	3,870.4	1,829.1	2,992.2	1,686.4
300	20,000	220/300	1.963	3,370.0	1,101.2	1,927.0	1,611.9	2,487.9	829.0
350	20,000	270/350	2.896	2,795.4	2,334.1	4,341.0	2,438.7	3,732.2	2,118.9
400	50,000	340/400	5.631	2,172.4	1,354.4	2,448.3	2,149.5	2,238.2	1,855.1
450	50,000	390/450	6.028	4,346.7	2,234.2	2,746.6	1,554.9	1,973.3	1,196.5
475	100,000	410/475	6.156	7,622.8	5,635.3	5,663.1	5,148.0	4,473.9	3,036.6
475	15,000	450/475	10.203	31.1	38.7	33.7	49.5	32.5	41.3

Table 8: Computational results for problem instances based on Dataset 1 with $n = 10$ are presented. N denotes the number of nodes in the first stage of M1′, M2′ and M3′. Objective values are scaled with coefficient 10^3 for ease of presentation.

Q	N	α	Obj	CPU (sec.)					
				M1	M1′	M2	M2′	M3	M3′
200	50,000	140/200	1.951	2,299.1	1,588.3	3,424.6	1,605.2	2,373.9	1,505.4
250	50,000	185/250	3.278	4,099.2	2,464.9	4,587.3	1,790.8	2,079.9	1,431.4
300	50,000	240/300	5.182	3,104.3	1,850.4	1,853.2	1,621.3	1,942.4	1,209.6
350	50,000	290/350	6.309	2,400.5	1,603.5	1,998.5	1,521.4	1,709.4	1,771.5
400	50,000	350/400	8.322	3,118.8	1,460.4	2,433.3	1,619.9	1,473.0	996.1
450	100,000	400/450	9.006	4,788.4	3,821.7	4,192.3	3,078.0	3,293.0	2,497.8
474	200,000	420/474	8.963	8,789.7	8,775.1	9,251.2	8,139.9	6,824.0	4,311.9
474	10,000	450/474	12.899	44.9	81.0	72.0	67.2	54.5	84.0

Table 9: Computational results for problem instances based on Dataset 2 with $n = 10$ are presented. N denotes the number of nodes in the first stage of M1′, M2′ and M3′. Objective values are scaled with coefficient 10^3 for ease of presentation.

6. Conclusion

This paper provides several improvements to define tight MIP formulation of the Value-at-Risk minimization problem. The proposed improvements are rather general, implying that results are useful for a wide range of applications. The main idea is the explicit usage of the

Q	N	α	Obj	CPU (sec.)					
				M1	M1$'$	M2	M2$'$	M3	M3$'$
949	800,000	902/949	1.267	24,176.5	20,411.8	15,036.8	14,042.5	15,461.4	8,584.0
	100,000	903/949	1.277	12,841.4	10,370.6	12,842.4	7,360.5	8,846.9	5,694.3
	200,000	904/949	1.291	18,028.2	19,771.2	14,758.2	9,453.8	10,092.6	6,175.9
	100,000	905/949	1.301	8,368.5	11,485.1	12,417.1	10,005.6	9,128.9	5,119.6
	50,000	910/949	1.336	2,370.1	1,824.0	3,773.1	1,936.5	2,669.1	1,533.6
	20,000	915/949	1.366	408.3	462.7	543.4	503.4	422.1	315.5
	10,000	920/949	1.429	211.0	226.2	265.3	242.6	232.0	233.3
	10,000	925/949	1.506	154.8	172.6	163.7	187.2	166.2	183.4

Table 10: Computational results for problem instances based on Dataset 3 with $n = 10$ are presented. N denotes the number of nodes in the first stage of M1$'$, M2$'$ and M3$'$. Objective values are scaled with coefficient 10^2 for ease of presentation.

bounds on the optimal value of VaR in the MIP formulation. For instance, a large body of existing literature provides heuristic solution approaches, each of which can serve as a feasible upper bound. A good upper bound can potentially result in a dimension reduction of the problem by constraining some binary variables to one, improving the quality of the linear relaxation of the problem. The lower bound on optimal VaR is shown to be an important factor in defining tight big M constants and the iterative procedure to tighten this bound is proposed. Moreover, the standard MIP solver serves as the final tool to tighten both upper and lower bounds. Specifically, the paper proposes a new solution approach that runs the MIP solver in two stages: the first stage launches the solver for a pre-specified number of branch-and-bound nodes only, after what the solver is stopped and new bounds with the best solution are recorded; the second stage employs new bounds to tighten the MI problem formulation, and uses the best feasible solution from the first stage as a “warm” start for the second stage that runs solver to optimality.

Numerical experiments show that solution time of VaR minimization problems can be reduced by up to 75% compared to the benchmark approach, which suggests that larger problem instances can become tractable while using the same computer hardware.

7. Appendix

Consider first the following example illustrating a case where the statement (2.6) does not necessary hold. Let ξ be a random variable taking values

$$(0, 1, 2, 3) \text{ with probabilities } (0.1, 0.3, 0.2, 0.4), \text{ respectively.}$$

Now, a counterexample to (2.6) would be $\mathsf{VaR}_{0.6}(\xi) = 2$ with $\mathsf{VaR}_{0.4}(-\xi) = -3$. At the same time, there are cases when (2.6) holds even for discrete distributions: $\mathsf{VaR}_{0.5}(\xi) = 2$ with $\mathsf{VaR}_{0.5}(-\xi) = -2$. Therefore, we formulate the following proposition that still provides a helpful relation between $\mathsf{VaR}_\alpha(\xi)$ and $\mathsf{VaR}_{1-\alpha}(-\xi)$ for a general ξ.

Proposition 7.1. *Let ξ be a random variable of losses. Let $\epsilon > 0$, then for $\alpha \in (0, 1]$:*

$$\mathsf{VaR}_\alpha(\xi) \geq -\mathsf{VaR}_{1-\alpha+\epsilon}(-\xi)\,, \tag{7.1}$$
$$\mathsf{VaR}_\alpha(\xi) \leq -\mathsf{VaR}_{1-\alpha}(-\xi)\,. \tag{7.2}$$

Proof. We start from the (7.1) statement. By definition,

$$\begin{aligned}&-\mathsf{VaR}_{1-\alpha+\epsilon}(-\xi) = -\inf\{l \in \mathbb{R} : \mathbb{P}(-\xi > l) \le \alpha - \epsilon\} = \\ &-\inf\{l \in \mathbb{R} : \mathbb{P}(\xi < -l) \le \alpha - \epsilon\} = \\ &-\inf\{-l \in \mathbb{R} : \mathbb{P}(\xi < l) \le \alpha - \epsilon\} = \\ &\ \sup\{l \in \mathbb{R} : \mathbb{P}(\xi < l) \le \alpha - \epsilon\}\,. \end{aligned} \tag{7.3}$$

Recall that

$$\mathsf{VaR}_\alpha(\xi) = \inf\{l \in \mathbb{R} : \mathbb{P}(\xi > l) \le 1 - \alpha\}\,.$$

Consider any $l_1 : \mathbb{P}(\xi < l_1) \le \alpha - \epsilon$ and any $l_2 : \mathbb{P}(\xi > l_2) \le 1 - \alpha$. Suppose that $l_2 < l_1$. Then,

$$\alpha - \epsilon \ge \mathbb{P}(\xi < l_1) \ge \mathbb{P}(\xi \le l_2) = 1 - \mathbb{P}(\xi > l_2) \ge \alpha\,,$$

which is an impossible inequality and that is why $l_1 \le l_2$. Therefore,

$$\sup\{l_1 \in \mathbb{R} : \mathbb{P}(\xi < l_1) \le \alpha - \epsilon\} \le \inf\{l_2 \in \mathbb{R} : \mathbb{P}(\xi > l_2) \le 1 - \alpha\}\,,$$

and the proof of (7.1) is completed. Now, to prove (7.2) let

$$a = -\mathsf{VaR}_{1-\alpha}(-\xi) = \sup\{l \in \mathbb{R} : \mathbb{P}(\xi < l) \le \alpha\}.$$

We can state that $\mathbb{P}(\xi \le a) \ge \alpha$. Indeed, suppose that $\mathbb{P}(\xi \le a) < \alpha$; then, since the cumulative distribution function $F(x) = \mathbb{P}(\xi \le x)$ of random variable ξ is right continuous, there exists an $\epsilon > 0$, such that $\mathbb{P}(\xi \le a + \epsilon) < \alpha$ and implying $\mathbb{P}(\xi < a + \epsilon) \le \alpha$, which contradicts the definition of a. Moreover, the Value-at-Risk can equivalently be defined as $\mathsf{VaR}_\alpha(\xi) = \inf\{l \in \mathbb{R} : \mathbb{P}(\xi \le l) \ge \alpha\}$, which establishes inequality $\mathsf{VaR}_\alpha(\xi) \le a$ and finishes the proof of the proposition. $\square$

8. Acknowledgements

The authors would like to thank Prof. O. A. Prokopyev, University of Pittsburgh, for his comments and suggestions during this study. We would also like to thank Giacomo Benincasa, Florida Institute for Human and Machine Cognition, for significant help with running computational study. This material is based upon work supported by the AFRL Mathematical Modeling and Optimization Institute.

9. References

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2010. Keel Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17 (2-3), 255–287.

Artzner, P., Delbaen, F., Eber, J.-M., Heath, D., 1999. Coherent Measures of Risk. Mathematical Finance 9 (3), 203–228.

Basel Committee on Banking Supervision, 2010. Basel III: A Global Regulatory Framework for more Resilient Banks and Banking Systems. Bank for International Settlements.

Benati, S., Rizzi, R., 2007. A Mixed Integer Linear Programming Formulation of the Optimal Mean/Value-at-Risk Portfolio Problem. European Journal of Operational Research 176 (1), 423–434.

Chen, G., Daskin, M. S., Shen, Z.-J. M., Uryasev, S., 2006. The α-Reliable Mean-Excess Regret Model for Stochastic Facility Location Modeling. Naval Research Logistics (NRL) 53 (7), 617–626.

Daníelsson, J., Jorgensen, B. N., Samorodnitsky, G., Sarma, M., de Vries, C. G., 2013. Fat Tails, VaR and Subadditivity. Journal of Econometrics 172 (2), 283–291.

Daskin, M. S., Hesse, S. M., Revelle, C. S., 1997. α-Reliable p-Minimax Regret: a New Model for Strategic Facility Location Modeling. Location Science 5 (4), 227–246.

Duffie, D., Pan, J., 1997. An Overview of Value at Risk. The Journal of Derivatives 4 (3), 7–49.

Feng, M., Wächter, A., Staum, J., 2015. Practical Algorithms for Value-at-Risk Portfolio Optimization Problems. Quantitative Finance Letters 3 (1), 1–9.

Gaivoronski, A. A., Pflug, G., 2005. Value-at-Risk in Portfolio Optimization: Properties and Computational Approach. Journal of Risk 7 (2), 1–31.

Goh, J. W., Lim, K. G., Sim, M., Zhang, W., 2012. Portfolio Value-at-Risk Optimization for Asymmetrically Distributed Asset Returns. European Journal of Operational Research 221 (2), 397–406.

Gurobi Optimization, I., 2015. Gurobi Optimizer Reference Manual.
URL http://www.gurobi.com

Larsen, N., Mausser, H., Uryasev, S., 2002. Algorithms for Optimization of Value-at-Risk. In: Financial Engineering, E-commerce and Supply Chain. Springer, pp. 19–46.

Luedtke, J., 2014. A Branch-and-Cut Decomposition Algorithm for Solving Chance-Constrained Mathematical Programs With Finite Support. Mathematical Programming 146 (1-2), 219–244.

Mausser, H., Romanko, O., 2014. CVaR Proxies for Minimizing Scenario-based Value-at-Risk. Journal of Industrial and Management Optimization 10 (4), 1109–1127.

Natarajan, K., Pachamanova, D., Sim, M., 2008. Incorporating Asymmetric Distributional Information in Robust Value-at-Risk Optimization. Management Science 54 (3), 573–585.

Nemirovski, A., Shapiro, A., 2006. Convex Approximations of Chance Constrained Programs. SIAM Journal on Optimization 17 (4), 969–996.

Pang, J.-S., Leyffer, S., 2004. On the Global Minimization of the Value-at-Risk. Optimization Methods and Software 19 (5), 611–631.

Pflug, G. C., 2000. Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk. In: Probabilistic Constrained Optimization. Springer, pp. 272–281.

Qiu, F., Ahmed, S., Dey, S. S., Wolsey, L. A., 2014. Covering Linear Programming With Violations. INFORMS Journal on Computing 26 (3), 531–546.

Rockafellar, R. T., Uryasev, S., 2000. Optimization of Conditional Value-at-Risk. Journal of Risk 2 (3), 21–41.

Rockafellar, R. T., Uryasev, S., 2002. Conditional Value-at-Risk for General Loss Distributions. Journal of Banking & Finance 26 (7), 1443–1471.

Romanko, O., Mausser, H., 2016. Robust Scenario-based Value-at-Risk Optimization. Annals of Operations Research 237 (1-2), 203–218.

Rousseeuw, P. J., 1984. Least Median of Squares Regression. Journal of the American Statistical Association 79 (388), 871–880.

Tsyurmasto, P., Zabarankin, M., Uryasev, S., 2014. Value-at-Risk Support Vector Machine: Stability to Outliers. Journal of Combinatorial Optimization 28 (1), 218–232.