Research Article

Marius Hofert*, Amir Memartoluie, David Saunders and Tony Wirjanto

Improved algorithms for computing worst Value-at-Risk

DOI: 10.1515/strm-2015-0028
Received November 29, 2015; revised December 29, 2016; accepted January 11, 2017

Abstract: Numerical challenges inherent in algorithms for computing worst Value-at-Risk in homogeneous portfolios are identified and solutions as well as words of warning concerning their implementation are provided. Furthermore, both conceptual and computational improvements to the Rearrangement Algorithm for approximating worst Value-at-Risk for portfolios with arbitrary marginal loss distributions are given. In particular, a novel Adaptive Rearrangement Algorithm is introduced and investigated. These algorithms are implemented using the R package `qrmtools` and may be of interest in any context in which it is required to find columnwise permutations of a matrix such that the minimal (maximal) row sum is maximized (minimized).

Keywords: Risk aggregation, model uncertainty, Value-at-Risk, Adaptive Rearrangement Algorithm

MSC 2010: 65C60, 62P05

1 Introduction

An integral part of Quantitative Risk Management is to analyze the one-period ahead vector of losses $\boldsymbol{L} = (L_1, \dots, L_d)$, where L_j represents the loss (a random variable) associated with a given business line or risk type with counterparty j, $j \in \{1, \dots, d\}$, over a fixed time horizon. For financial institutions, the *aggregated loss*

$$L^+ = \sum_{j=1}^{d} L_j$$

is of particular interest. A risk measure $\rho(\,\cdot\,)$ is used to map the aggregate position L^+ to $\rho(L^+) \in \mathbb{R}$ for obtaining the amount of capital required to account for future losses over a predetermined time period. As a risk measure, Value-at-Risk (VaR_α) has been widely adopted by the financial industry since the mid nineties. It is defined as the α-quantile of the distribution function F_{L^+} of L^+, i.e.,

$$\mathrm{VaR}_\alpha(L^+) = F_{L^+}^-(\alpha) = \inf\{x \in \mathbb{R} : F_{L^+}(x) \geq \alpha\}, \quad \alpha \in (0, 1),$$

where $F_{L^+}^-$ denotes the quantile function of F_{L^+}; see [11] for more details. There are various methods for estimating the marginal loss distributions $F_1, \dots, F_d$ of $L_1, \dots, L_d$, respectively, but capturing the d-variate dependence structure (i.e., the underlying copula C) of $\boldsymbol{L}$ is often more difficult. This is due to the fact that typically not much is known about C and estimation is often not feasible (e.g., for rare-event losses occurring in different geographic regions). However, partial dependence information is often available (e.g., through knowledge of the variance of the sum). This case (and thus a possibly smaller *dependence uncertainty spread* $\overline{\mathrm{VaR}}_\alpha(L^+) - \underline{\mathrm{VaR}}_\alpha(L^+)$), is studied by [5–8, 21], where the RA also has been shown to be a useful tool.

***Corresponding author: Marius Hofert:** Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada, e-mail: marius.hofert@uwaterloo.ca
Amir Memartoluie: Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada, e-mail: amir.memartoluie@uwaterloo.ca
David Saunders, Tony Wirjanto: Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1, Canada, e-mail: david.saunders@uwaterloo.ca, tony.wirjanto@uwaterloo.ca

In this work we focus on the case where C is unknown and study the problem of computing $\mathrm{VaR}_\alpha(L^+)$ bounds $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$, where

$$\underline{\mathrm{VaR}}_\alpha(L^+) = \inf\{\mathrm{VaR}_\alpha(L^+) : L_1 \sim F_1, \dots, L_d \sim F_d\},$$
$$\overline{\mathrm{VaR}}_\alpha(L^+) = \sup\{\mathrm{VaR}_\alpha(L^+) : L_1 \sim F_1, \dots, L_d \sim F_d\},$$

i.e., $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$ denote the best and the worst $\mathrm{VaR}_\alpha(L^+)$ over all distributions of $\boldsymbol{L}$ with marginals $F_1, \dots, F_d$, respectively. This problem is non-trivial as $\mathrm{VaR}_\alpha(L^+)$ can be superadditive, so, in particular, computing $\mathrm{VaR}_\alpha(L^+)$ in the comonotonic case (of all losses being equal in distribution to $F_1^-(U), \dots, F_d^-(U)$ for $U \sim \mathrm{U}[0,1]$) may not lead to $\overline{\mathrm{VaR}}_\alpha(L^+)$.

The *dependence uncertainty interval* $[\underline{\mathrm{VaR}}_\alpha(L^+), \overline{\mathrm{VaR}}_\alpha(L^+)]$ can be wide, see, e.g., [14], but financial firms are interested in computing it (often in a high dimensional d) to determine their risk capital for L^+ within this range. As we show in this work, even the computations for small values of d (and other moderate parameter choices) require care.

We investigate analytical solutions in the *homogeneous case* (i.e., $F_1 = \dots = F_d$); see [14]. In the general, *inhomogeneous case* (i.e., not all marginals F_j necessarily being equal), we consider and improve the *Rearrangement Algorithm (RA)* of [13] for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$; we mostly focus on $\overline{\mathrm{VaR}}_\alpha(L^+)$. All presented algorithms have been implemented in the R package `qrmtools`; see also the accompanying vignette `VaR_bounds` which provides further results, numerical investigations, diagnostic checks and an application. The results presented in this paper can be reproduced with the package and vignette (and, obviously, other parameter values can be chosen).

Another direction to improve the RA is by considering so-called block rearrangements (thus leading to the so-called *block RA (BRA)*), where blocks of columns (instead of single columns) are rearranged at a time. This idea was introduced in [3] and further applied in [5]. It is related to the notion of Σ-countermonotonicity studied in [22]. Whenever a block of columns can be identified with the property that its row sums are not oppositely ordered with respect to the row sums across the remaining columns, the outcome of the RA can be improved. In general, however, there are many blocks to consider and it is not clear whether in practice this approach always leads to better results (checking all blocks is not feasible for moderate or large dimensions of the underlying matrix). Bernard and McLeish [2] address the issue of non-convergence of the BRA to a global optimum by applying Markov Chain Monte Carlo methods.

The remaining parts of this paper are organized as follows. In Section 2 we highlight and solve numerical challenges that practitioners may face when implementing theoretical solutions for $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case $F_1 = \dots = F_d$. Section 3 presents the main concept underlying the Rearrangement Algorithm for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$, improves the choices of its tuning parameters and investigates its empirical performance under various scenarios. Section 4 then presents a conceptually and numerically improved version of the RA, which we call the Adaptive Rearrangement Algorithm (ARA), for calculating $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$. Section 5 concludes.

2 Known optimal solutions in the homogeneous case and their tractability

For $d = 2$, Embrechts, Puccetti and Rüschendorf [13, Proposition 2] provide an analytical solution for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$ for distribution functions concentrated on $[0, \infty)$ with ultimately decreasing densities (i.e., densities which are decreasing beyond a certain point); the dependence underlying $\overline{\mathrm{VaR}}_\alpha(L^+)$ above the confidence level $\alpha \in (0,1)$ is simply countermonotonicity, see [13, Remark 3]. In particular, a sum which is minimal with respect to the convex ordering is obtained under countermonotonicity and this is sufficient to get sharp VaR_α bounds; see [4, Theorem 2.5] for more details. The latter result also holds for $d \geq 3$ and implies that maximization of VaR_α can be achieved by constructing a sum that is minimal with respect to the convex ordering above the α-quantile. The RA provides a practical way of achieving such a sum. In order to assess its quality from a numerical point of view, we need to know (at least some) optimal solutions to which we can compare the RA algorithm. The paper [12] presents a formula for obtaining $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case; see also [13, Proposition 4] or [14, Proposition 1]. In this section, we address the corresponding numerical aspects and algorithmic improvements. We assume $d \geq 3$ throughout.

2.1 Crude bounds for any $\mathrm{VaR}_\alpha(L^+)$

The following lemma is a consequence of [18, Theorem 2.7] and provides (crude) bounds for $\mathrm{VaR}_\alpha(L^+)$ which are useful for computing initial intervals (see Section 2.2) and for conducting sanity checks. Note that this lemma does not involve any assumptions on the involved marginal loss distributions.

Lemma 2.1 (Crude bounds for $\mathrm{VaR}_\alpha(L^+)$). *Let $L_j \sim F_j$, $j \in \{1, \dots, d\}$. For any $\alpha \in (0, 1)$,*

$$d \min_j F_j^-\Big(\frac{\alpha}{d}\Big) \leq \mathrm{VaR}_\alpha(L^+) \leq d \max_j F_j^-\Big(1 - \frac{1-\alpha}{d}\Big), \tag{1}$$

where F_j^- denotes the quantile function of F_j.

The bounds (1) can be computed with the function `crude_VaR_bounds()` in the R package `qrmtools`.

2.2 The dual bound approach for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

This approach, termed *dual bound approach*, for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case with margin(s) F is presented in [12]; note that there is no corresponding algorithm for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ with this approach. In the remaining part of this subsection we assume that $F(0) = 0$, $F(x) < 1$ for all $x \in [0, \infty)$ and that F is absolutely continuous with an ultimately decreasing density. Let

$$D(s, t) = \frac{d}{s - dt} \int_t^{s-(d-1)t} \bar{F}(x)\,dx \quad \text{and} \quad D(s) = \min_{t \in [0, s/d]} D(s, t), \tag{2}$$

where $\bar{F}$ denotes the *survival function of* F, i.e., $\bar{F}(x) = 1 - F(x)$; $D(s)$ is known as *dual bound*. Note that by l'Hôpital's rule,

$$\lim_{t \uparrow s/d} D(s, t) = d\bar{F}\Big(\frac{s}{d}\Big)$$

(concerning the required differentiability assumptions, the numerator $t \mapsto d \int_t^{s-(d-1)t} \bar{F}(x)\,dx$ is differentiable by absolute continuity of F and the denominator $t \mapsto s - dt$ has derivative $-d \neq 0$).

Numerically, the procedure for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ can now be given as follows.

Algorithm 2.2 (Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ according to the dual bound approach). Proceed as follows.

(1) Specify initial intervals $[s_l, s_u]$ and $[t_l, t_u]$.

(2) Inner root-finding in t: For each considered $s \in [s_l, s_u]$, compute $D(s)$ by finding a $t^* \in [t_l, t_u]$ such that $h(s, t^*) = 0$, where

$$h(s, t) := D(s, t) - (\bar{F}(t) + (d-1)\bar{F}(s - (d-1)t)).$$

Then $D(s) = \bar{F}(t^*) + (d-1)\bar{F}(s - (d-1)t^*)$.

(3) Outer root-finding in s: Repeat Step (2) for $s \in [s_l, s_u]$ until an s^* is found for which $D(s^*) = 1 - \alpha$. Then return $s^* = \overline{\mathrm{VaR}}_\alpha(L^+)$.

Algorithm 2.2 is implemented in the function `VaR_bounds_hom(..., method="dual")` in the R package `qrmtools`; the dual bound D is available via `dual_bound()`. It requires a one-dimensional numerical integration (unless $\bar{F}$ can be integrated analytically) within two nested calls of a root-finding algorithm (`uniroot()` in R). Note that Algorithm 2.2 requires a specification of the two initial intervals $[s_l, s_u]$ and $[t_l, t_u]$.

First consider $[t_l, t_u]$. Under the assumptions on F, one can choose $t_l = 0$. For t_u one would like to choose s/d; see the definition of $D(s)$ in (2). However, care has to be exercised as $h(s, s/d) = 0$ for any s and thus the inner root-finding procedure will directly stop when a root is found at $t_u = s/d$. To address this issue, the inner root-finding algorithm in `VaR_bounds_hom(..., method="dual")` fixes `f.upper`, i.e., $h(s, t_u)$ for the considered s, to $-h(s, 0)$ so that a root below $t_u = s/d$ can be detected; note that this is an (inelegant; for lack of a better method) adjustment in the function value, and not in the root-finding interval $[t_l, t_u]$.

Now consider $[s_l, s_u]$, in particular, s_l. If it is chosen too small, the inner root-finding procedure in Step (2) of Algorithm 2.2 will not be able to locate a root. A solution candidate is $\mathrm{VaR}_\alpha(L^+)$ under comonotonicity, so $s_l = d\,\mathrm{VaR}_\alpha(L)$ for $L \sim F$. Given s_l, one can then choose s_u as the maximum of $s_l + 1$ and the upper bound on $\mathrm{VaR}_\alpha(L^+)$ as given in (1), for example; another option for s_u is worst expected shortfall (see later) if it exists, so $d\mathrm{ES}_\alpha(L)$.

On the theoretical side, the following proposition implies that if $\bar{F}$ is strictly convex, so is $D(s, \cdot)$ for fixed s. This shows the uniqueness of the minimum when computing $D(s)$ as in (2). A standard result on convexity of objective value functions then implies that $D(s)$ itself is also convex (and therefore continuous on the interior of its domain); see [23, Proposition 2.22].

Proposition 2.3 (Properties of $D(s, t)$ and $D(s)$). *The following statements hold.*

(1) *$D(s)$ is decreasing. If $\bar{F}$ is strictly decreasing, so is D.*

(2) *If $\bar{F}$ is convex, so is $D(s, t)$.*

Proof. (1) Let $s \geq s'$ and $t' \in [0, \frac{s'}{d}]$ such that $D(s', t') = D(s')$. Define

$$t = \frac{s - (s' - t'd)}{d} = \frac{s - s'}{d} + t'$$

so that $0 \leq t' \leq t \leq \frac{s}{d}$. Let $\kappa = s' - t'd = s - td$. If $\kappa > 0$, noting that $\bar{F}$ is decreasing and that $t' \leq t$, we obtain

$$D(s', t') - D(s, t) = \frac{d}{\kappa}\bigg(\int_{t'}^{t'+\kappa} \bar{F}(x)\,dx - \int_{t}^{t+\kappa} \bar{F}(x)\,dx\bigg) \geq 0,$$

so that $D(s) \leq D(s, t) \leq D(s', t') = D(s')$. If $\kappa = 0$, then

$$D(s') = D\Big(s', \frac{s'}{d}\Big) = d\bar{F}\Big(\frac{s'}{d}\Big) \geq d\bar{F}\Big(\frac{s}{d}\Big) \geq D(s).$$

When $\bar{F}$ is strictly decreasing, the inequalities are strict.

(2) Recall that

$$D(s, t) = \frac{d}{s - td} \int_{t}^{t+(s-td)} \bar{F}(x)\,dx.$$

Using the transformation $z = (x - t)/(s - td)$, we have

$$D(s, t) = d \int_0^1 \bar{F}(sz + t(1 - zd))\,dz.$$

Define $C = \{(s, t) : 0 \leq s < \infty,\ 0 \leq t \leq \frac{s}{d}\}$, and note that C is convex. Furthermore, if $\bar{F}$ is convex, then $D(s, t)$ is jointly convex in s and t on C since for $\lambda \in (0, 1)$,

$$\begin{aligned}
D(\lambda s_1 + (1 - \lambda)s_2, \lambda t_1 + (1 - \lambda)t_2) &= d \int_0^1 \bar{F}((\lambda s_1 + (1 - \lambda)s_2)z + (\lambda t_1 + (1 - \lambda)t_2)(1 - zd))\,dz \\
&= d \int_0^1 \bar{F}(\lambda(s_1 z + t_1(1 - zd)) + (1 - \lambda)(s_2 z + t_2(1 - zd)))\,dz \\
&\leq \int_0^1 \lambda\bar{F}(((s_1 + t_1(1 - zd))) + (1 - \lambda)\bar{F}((s_2 + t_2(1 - zd))))\,dz \\
&= \lambda D(s_1, t_1) + (1 - \lambda)D(s_2, t_2).
\end{aligned}$$

□

2.3 Wang's approach for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$

Theory

The approach mentioned in [14, Proposition 1] is termed *Wang's approach* here. It originates from [25, Corollary 3.7]. For notational simplicity, let us introduce

$$a_c = \alpha + (d-1)c, \quad b_c = 1 - c,$$

for $c \in [0, (1-\alpha)/d]$ (so that $a_c \in [\alpha, 1-(1-\alpha)/d]$ and $b_c \in [1-(1-\alpha)/d, 1])$ and

$$\bar{I}(c) := \frac{1}{b_c - a_c} \int_{a_c}^{b_c} F^-(y)\, dy, \quad c \in \left(0, \frac{1-\alpha}{d}\right],$$

with the assumption that F admits a density which is positive and decreasing on $[\beta, \infty)$ for some $\beta \le F^-(\alpha)$. Then, for $L \sim F$,

$$\overline{\mathrm{VaR}}_\alpha(L^+) = d\,\mathbb{E}[L \mid L \in [F^-(a_c), F^-(b_c)]], \quad \alpha \in [F(\beta), 1), \tag{3}$$

where c (typically depending on d, α) is the smallest number in $(0, (1-\alpha)/d]$ such that

$$\bar{I}(c) \ge \frac{d-1}{d} F^-(a_c) + \frac{1}{d} F^-(b_c). \tag{4}$$

In contrast to what is given in [14], note that $(0, (1-\alpha)/d]$ has to exclude 0 since otherwise, for $\mathrm{Par}(\theta)$ margins with $\theta \in (0, 1]$, c equals 0 and thus, erroneously, $\overline{\mathrm{VaR}}_\alpha(L^+) = \infty$. If F is the distribution function of the $\mathrm{Par}(\theta)$ distribution, then $\bar{I}$ is given by

$$\bar{I}(c) = \begin{cases} \frac{1}{b_c - a_c} \frac{\theta}{1-\theta}((1-b_c)^{1-1/\theta} - (1-a_c)^{1-1/\theta}) - 1, & \text{if } \theta \ne 1, \\ \frac{1}{b_c - a_c} \log(\frac{1-a_c}{1-b_c}) - 1, & \text{if } \theta = 1. \end{cases} \tag{5}$$

The conditional distribution function $F_{L \mid L \in [F^-(a_c), F^-(b_c)]}$ of $L \mid L \in [F^-(a_c), F^-(b_c)]$ is given by

$$F_{L \mid L \in [F^-(a_c), F^-(b_c)]}(x) = \frac{F(x) - a_c}{b_c - a_c}, \quad x \in [F^-(a_c), F^-(b_c)].$$

Using this fact and by substitution, we obtain that, for $\alpha \in [F(\beta), 1)$, (3) becomes

$$\overline{\mathrm{VaR}}_\alpha(L^+) = d \int_{F^-(a_c)}^{F^-(b_c)} x\, dF_{L \mid L \in [F^-(a_c), F^-(b_c)]}(x) = d\, \frac{\int_{F^-(a_c)}^{F^-(b_c)} x\, dF(x)}{b_c - a_c} = d\bar{I}(c). \tag{6}$$

Equation (6) has the advantage of having the integration in $\bar{I}(c)$ over an (at least theoretically) compact interval. Furthermore, finding the smallest c such that (4) holds also involves $\bar{I}(c)$. We thus only need to know the quantile function F^- in order to compute $\overline{\mathrm{VaR}}_\alpha(L^+)$. This leads to the following algorithm.

Algorithm 2.4 (Computing $\overline{\mathrm{VaR}}_\alpha(L^+)$ according to Wang's approach). Proceed as follows.
(1) Specify an initial interval $[c_l, c_u]$ with $0 \le c_l < c_u < (1-\alpha)/d$.
(2) Root-finding in c: Find a $c^* \in [c_l, c_u]$ such that $h(c^*) = 0$, where

$$h(c) := \bar{I}(c) - \left(\frac{d-1}{d} F^-(a_c) + \frac{1}{d} F^-(b_c)\right), \quad c \in \left(0, \frac{1-\alpha}{d}\right].$$

Then return $(d-1)F^-(a_{c^*}) + F^-(b_{c^*})$.

This procedure is implemented in the function `VaR_bounds_hom(..., method="Wang")` in the R package `qrmtools` with numerical integration via R's `integrate()` for computing the integral $\bar{I}(c)$; the function `VaR_bounds_hom(..., method="Wang.Par")` makes use of (5).

The following proposition shows that the root-finding problem in Step (2) of Algorithm 2.4 is well defined in the case of Pareto margins for all $\theta > 0$ (including the infinite-mean case); for other distributions under more restrictive assumptions, see [1].

Proposition 2.5. *Let* $F(x) = 1 - (1+x)^{-\theta}$, $\theta > 0$, *be the distribution function of the* $\operatorname{Par}(\theta)$ *distribution. Then* h *in Step* (2) *of Algorithm 2.4 has a unique root on* $(0, (1-\alpha)/d)$, *for all* $\alpha \in (0,1)$ *and* $d > 2$.

Proof. First consider $\theta \neq 1$. Using (5), one can rewrite $h(c)$ as

$$h(c) = \frac{c^{-1/\theta+1}\frac{\theta}{1-\theta}(1-(\frac{1-\alpha}{c}-(d-1))^{-1/\theta+1})}{1-\alpha-dc} - \frac{(d-1)(\frac{1-\alpha}{c}-(d-1)^{-1/\theta}+1)}{c^{1/\theta}d}.$$

Multiplying with $c^{1/\theta}d$ and rewriting the expression, one sees that $h(c) = 0$ is equivalent to $h_1(x_c) = 0$ where

$$x_c = \frac{1-\alpha}{c} - (d-1) \tag{7}$$

(which is in $(1,\infty)$ for $c \in (0, (1-\alpha)/d)$) and $h_1(x) = d\frac{\theta}{1-\theta}\frac{1-x^{-1/\theta+1}}{x-1} - ((d-1)x^{-1/\theta}+1)$. It is easy to see that $h_1(x) = 0$ if and only if $h_2(x) = 0$, where

$$h_2(x) = \Big(\frac{d}{1-\theta} - 1\Big)x^{-1/\theta+1} - (d-1)x^{-1/\theta} + x - \Big(\frac{d\theta}{1-\theta} + 1\Big), \quad x \in (1,\infty). \tag{8}$$

We are done for $\theta \neq 1$ if we show that h_2 has a unique root on $(1,\infty)$. To this end, note that $h_2(1) = 0$ and $\lim_{x\uparrow\infty} h_2(x) = \infty$. Furthermore,

$$\begin{aligned} h_2'(x) &= \Big(1-\frac{1}{\theta}\Big)\Big(\frac{d}{1-\theta}-1\Big)x^{-1/\theta} + \frac{d-1}{\theta}x^{-1/\theta-1} + 1, \\ h_2''(x) &= \frac{d+\theta-1}{\theta^2}x^{-1/\theta-1} - \Big(\frac{1}{\theta}+1\Big)\frac{d-1}{\theta}x^{-1/\theta-2}. \end{aligned}$$

It is not difficult to check that $h_2''(x) = 0$ if and only if $x = \frac{(d-1)(1+\theta)}{d+\theta-1}$ (which is greater than 1 for $d > 2$). Hence, h_2 can have at most one root. We are done if we find an $x_0 \in (1,\infty)$ such that $h_2(x_0) < 0$, but this is guaranteed by the fact that $\lim_{x\downarrow 1} h_2'(x) = 0$ and $\lim_{x\downarrow 1} h_2''(x) = -(d-2)/\theta < 0$ for $d > 2$.

The proof for $\theta = 1$ works similarly; in this case, h_2 is given by

$$h_2(x) = x^2 + x(-d\log(x) + d - 2) - (d-1), \quad x \in (1,\infty), \tag{9}$$

and the unique point of inflection of h_2 is $x = d/2$. □

Practice

Let us now focus on the case of $\operatorname{Par}(\theta)$ margins (see `VaR_bounds_hom(..., method="Wang.Par")`) and, in particular, how to choose the initial interval $[c_l, c_u]$ in Algorithm 2.4. We first consider c_l. Note that $\bar{I}$ satisfies $\bar{I}(0) = \frac{1}{1-\alpha}\int_\alpha^1 F^-(y)\,dy = \mathrm{ES}_\alpha(L)$, i.e., $\bar{I}(0)$ is the *expected shortfall* of $L \sim F$ at confidence level α. If L has a finite first moment, then $\bar{I}(0)$ is finite. Therefore, $h(0)$ is finite (if $F^-(1) < \infty$) or $-\infty$ (if $F^-(1) = \infty$). Either way, one can take $c_l = 0$. However, if $L \sim F$ has an infinite first moment (see, e.g., [16] or [9] for situations in which this can happen), then $\bar{I}(0) = \infty$ and $F^-(1) = \infty$, so $h(0)$ is not well defined; this happens, e.g., if F is $\operatorname{Par}(\theta)$ with $\theta \in (0,1]$. In such a case, we are forced to choose $c_l \in (0, (1-\alpha)/d)$; see the following proposition for how this can be done *theoretically*. Concerning c_u, note that l'Hôpital's rule implies that $\bar{I}(c_u) = F^-(1-(1-\alpha)/d)$ and thus $h((1-\alpha)/d) = 0$. We thus have a similar problem (a root at the upper endpoint of the initial interval) to the computation of the dual bound. However, here we can construct a suitable $c_u < (1-\alpha)/d$; see the following proposition.

Proposition 2.6 (Computing c_l, c_u for $\operatorname{Par}(\theta)$ risks). *Let* F *be the distribution function of a* $\operatorname{Par}(\theta)$ *distribution,* $\theta > 0$. *Then* c_l *and* c_u *in Algorithm 2.4 can be chosen as*

$$c_l = \begin{cases} \frac{(1-\theta)(1-\alpha)}{d}, & \textit{if } \theta \in (0,1), \\ \frac{1-\alpha}{(d+1)^{\frac{e}{e-1}}+d-1}, & \textit{if } \theta = 1, \\ \frac{1-\alpha}{(d/(\theta-1)+1)^\theta+d-1}, & \textit{if } \theta \in (1,\infty), \end{cases} \qquad c_u = \begin{cases} \frac{(1-\alpha)(d-1+\theta)}{(d-1)(2\theta+d)}, & \textit{if } \theta \neq 1, \\ \frac{1-\alpha}{3d/2-1}, & \textit{if } \theta = 1. \end{cases}$$

Proof. First consider c_l and $\theta \neq 1$. Instead of h, formulas (7) and (8) allow us to study

$$h_2(x) = \Big(\frac{d}{1-\theta} - 1\Big)x^{-1/\theta+1} - (d-1)x^{-1/\theta} + x - \Big(\frac{d\theta}{1-\theta} + 1\Big), \quad x \in [1, \infty).$$

Consider the two cases $\theta \in (0, 1)$ and $\theta \in (1, \infty)$ separately. If $\theta \in (0, 1)$, then $d/(1-\theta) - 1 > d - 1 \geq 0$ and $x^{-1/\theta+1} \geq x^{-1/\theta}$ for all $x \geq 1$, so

$$h_2(x) \geq \Big(\Big(\frac{d}{1-\theta} - 1\Big) - (d-1)\Big)x^{-1/\theta} + x - \Big(\frac{d\theta}{1-\theta} + 1\Big) \geq x - \Big(\frac{d\theta}{1-\theta} + 1\Big)$$

which is 0 if and only if $x = d\theta/(1-\theta) + 1$. Setting this equal to x_c (defined in (7)) and solving for c leads to the c_l as provided. If $\theta \in (1, \infty)$, then using $x^{-1/\theta} \leq 1$ leads to $h_2(x) \geq (d/(1-\theta) - 1)x^{-1/\theta+1} + x$ which is 0 for $x \geq 1$ if and only if $x = (d/(\theta - 1) + 1)^{\theta}$. Setting this equal to x_c and solving for c leads to the c_l as required.

Now consider $\theta = 1$. As before, we can consider (7) and (9). By using the fact that $\log x \leq x^{1/e}$ and $x \geq -x^{1+1/e}$ for $x \in [1, \infty)$, we obtain $h_2(x) \geq x^2 + x(-dx^{1/e} + d - 2) - (d-1) \geq x^2 - (d+1)x^{1+1/e}$ which is 0 if and only if $x = (d+1)^{e/(e-1)}$. Setting this equal to x_c and solving for c leads to the c_l as required.

Lastly consider c_u. It is easy to see that the inflection point of h_2 provides a lower bound x_c on the root of h_2. As derived in the proof of Proposition 2.5, the point of inflection is $x = x_c := (d-1)(1+\theta)/(d+\theta-1)$ for $\theta \neq 1$ and $x = d/2$ for $\theta = 1$. Solving $x_c = (1-\alpha)/c - (d-1)$ for c then leads to c_u as stated. □

In the following example, we briefly address several numerical hurdles we had to overcome when implementing `VaR_bounds_hom(..., method="Wang.Par")`; see the vignette `VaR_bounds` for more details.

Example 2.7 (Comparison of the approaches for Par(θ) risks). For obtaining numerically reliable results, one has to be careful when computing the root of h for $c \in (0, (1-\alpha)/d)$. As an example, consider Par(θ) risks and the confidence level $\alpha = 0.99$. Figure 1 compares Wang's approach (using numerical integration; see `VaR_bounds_hom(...,method="Wang")`), Wang's approach (with an analytical formula for the integral $\bar{I}(c)$ but `uniroot()`'s default tolerance; see the vignette `VaR_bounds`), Wang's approach (with an analytical formula for the integral $\bar{I}(c)$ and auxiliary function h transformed to $(1, \infty)$; see the vignette `VaR_bounds`), Wang's approach (with analytical formula for the integral $\bar{I}(c)$, smaller `uniroot()` tolerance and adjusted initial interval; see `VaR_bounds_hom(..., method="Wang.Par")`), and the lower and upper bounds obtained from the RA (with absolute tolerance 0); see Section 3 and `RA()`. All of the results are divided by the values obtained from the dual bound approach to facilitate comparison.

As can be seen, choosing a smaller root-finding tolerance is crucial. Figure 1 shows what could occur if this is not done (our procedure chooses MATLAB's default $2.2204 \cdot 10^{-16}$ instead of the much larger `uniroot()` default `.Machine$double.eps^0.25`). Furthermore, it turned out to be required to adjust the *theoretically valid* initial interval described in Proposition 2.6 in order to guarantee that h is *numerically* of opposite sign at the interval end points. In particular, `VaR_bounds_hom(..., method="Wang.Par")` chooses $c_l/2$ as a lower end point (with c_l as in Proposition 2.6) in the case $\theta \neq 1$.

These problems are described in detail in Section 1.4 of the vignette `VaR_bounds`, where we also show that transforming the auxiliary function h to a root-finding problem on $(1, \infty)$ as described in the proof of Proposition 2.6 does not only require a smaller root-finding tolerance but also an extended initial interval and, furthermore, it faces a cancellation problem (which can be solved, though); see also the left-hand side of Figure 3, where we compare this approach to `VaR_bounds_hom(..., method="Wang.Par")` *after* fixing these numerical issues.

In short, one should exercise caution in implementing analytical solutions for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ or $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the homogeneous case with Par(θ) (and most likely also other) margins.

Let us again stress how important the initial interval $[c_l, c_u]$ is. One could be tempted to simply choose $c_u = (1-\alpha)/d$ and force the auxiliary function h to be of opposite sign at c_u, e.g., by setting $h(c_u)$ to `.Machine$double.xmin`, a positive but small number. Figure 2 (left-hand side) shows how Figure 1 looks like in this case (here it is standardized with respect to the upper bound obtained from the RA). Figure 2 (right-hand side) shows the implied $\overline{\mathrm{VaR}}_\alpha(L^+)$. In particular, the computed $\overline{\mathrm{VaR}}_\alpha(L^+)$ values are not monotone in α anymore (and thus not correct).

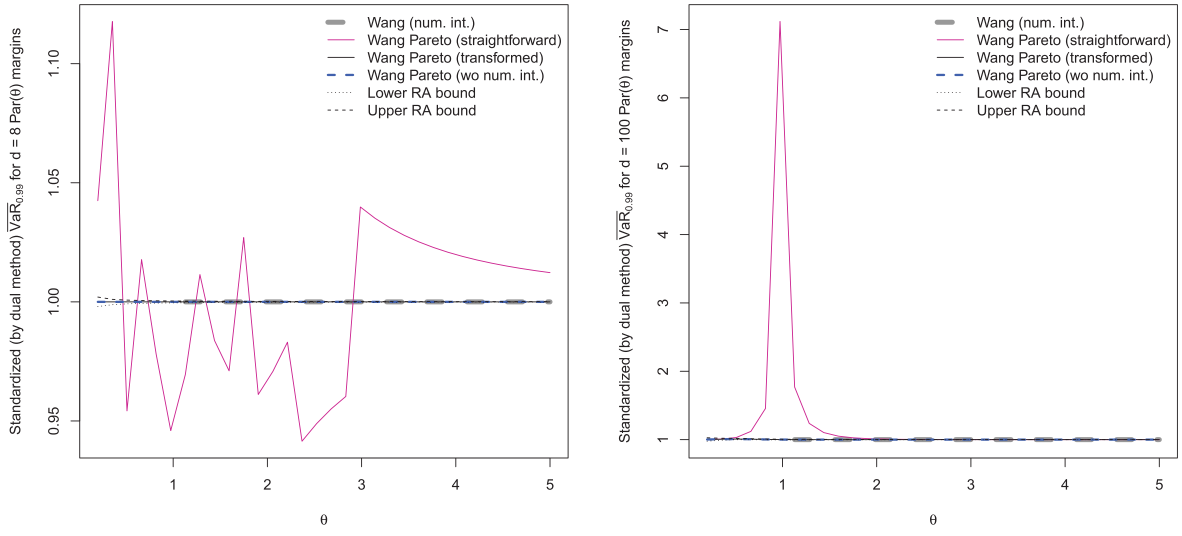

Figure 1. Comparisons of Wang's approach (using numerical integration; see `VaR_bounds_hom(..., method="Wang")`), Wang's approach (with an analytical formula for the integral $\overline{I}(c)$ but `uniroot()`'s default tolerance; see the vignette `VaR_bounds`), Wang's approach (with an analytical formula for the integral $\overline{I}(c)$ and auxiliary function h transformed to $(1, \infty)$; see the vignette `VaR_bounds`), Wang's approach (with analytical formula for the integral $\overline{I}(c)$, smaller `uniroot()` tolerance and adjusted initial interval; see `VaR_bounds_hom(..., method="Wang.Par")`), and the lower and upper bounds obtained from the RA; all of the results are divided by the values obtained from the dual bound approach to facilitate comparison. The left-hand side shows the case with $d = 8$, the right-hand side with $d = 100$.

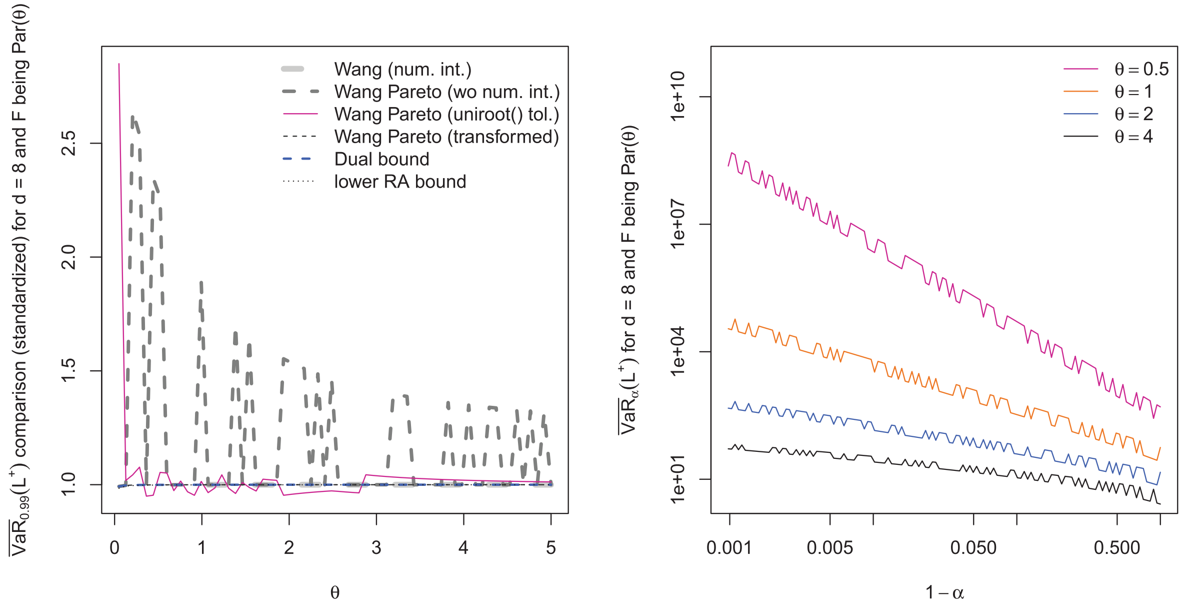

Figure 2. Comparison of various methods for computing $\overline{\mathrm{VaR}}_{0.99}(L^+)$ (left-hand side) and $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ as a function of α (right-hand side) for Par(θ) risks with $h((1-\alpha)/d)$ being naively adjusted to `.Machine$double.xmin`.

After carefully addressing the numerical issues, we can now consider $\underline{\mathrm{VaR}}_{\alpha}(L^+)$ and $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ from a different perspective. The right-hand side of Figure 3 shows $\underline{\mathrm{VaR}}_{\alpha}(L^+)$ and $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ as functions of the dimension d. The linearity of $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ in the log-log scale suggests that $\overline{\mathrm{VaR}}_{\alpha}(L^+)$ is actually a power function in d. To the best of our knowledge, this is not known (nor theoretically justified) yet.

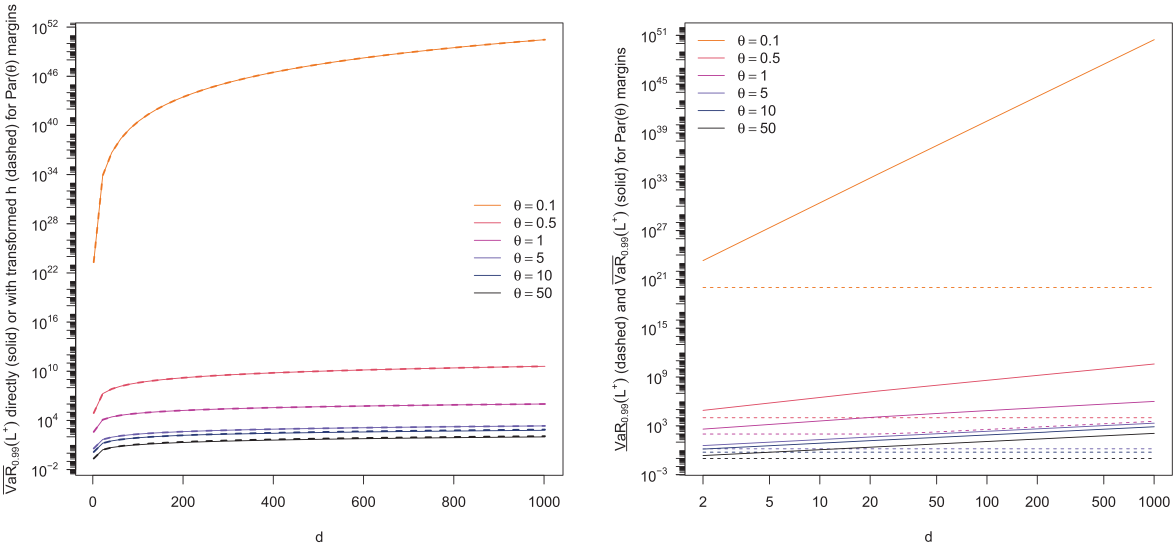

Figure 3. A comparison of $\overline{\mathrm{VaR}}_\alpha(L^+)$ computed with `VaR_bounds_hom(..., method="Wang.Par")` (solid line) and with the approach based on transforming the auxiliary function h to a root-finding problem on $(1, \infty)$ (dashed line) as described in the proof of Proposition 2.6 (left-hand side). $\underline{\mathrm{VaR}}_\alpha(L^+)$ (dashed line) and $\overline{\mathrm{VaR}}_\alpha(L^+)$ (solid line) as functions of d on log-log scale (right-hand side).

3 The Rearrangement Algorithm

We now consider the RA for computing $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$ in the inhomogeneous case; as before, we mainly focus on $\overline{\mathrm{VaR}}_\alpha(L^+)$ here. The algorithm was proposed by [19]; see also [13]. The idea underlying the RA and the numerical approximation for calculating $\underline{\mathrm{VaR}}_\alpha(L^+)$ and $\overline{\mathrm{VaR}}_\alpha(L^+)$ dates back to [24]. Note that the RA tries to find solutions to *maximin* (for $\overline{\mathrm{VaR}}_\alpha(L^+)$) and *minimax* (for $\underline{\mathrm{VaR}}_\alpha(L^+)$) *problems* and is thus of interest in a wider context, e.g., Operations Research; an early reference for this is [10]. In the following subsections we look at how the RA works and analyze its performance using various test cases.

3.1 How the Rearrangement Algorithm works

The RA can be applied to approximate the best Value-at-Risk $\underline{\mathrm{VaR}}_\alpha(L^+)$ or the worst Value-at-Risk $\overline{\mathrm{VaR}}_\alpha(L^+)$ for any set of marginals F_j, $j \in \{1, \dots, d\}$. In what follows our focus is mainly on $\overline{\mathrm{VaR}}_\alpha(L^+)$; our implementation `RA()` in the R package `qrmtools` also addresses $\underline{\mathrm{VaR}}_\alpha(L^+)$. To understand the algorithm, note that two columns $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^N$ are called *oppositely ordered* if for all $i, j \in \{1, \dots, N\}$ we have $(a_i - a_j)(b_i - b_j) \le 0$. Given a number N of discretization points of the marginal quantile functions $F_1^-, \dots, F_d^-$ above α (see Steps (2) (a) and (3) (a) of Algorithm 3.1 below), the RA constructs two (N, d)-matrices, denoted by $\underline{X}^\alpha$ and $\overline{X}^\alpha$, respectively; the first matrix aims at constructing an approximation of $\overline{\mathrm{VaR}}_\alpha(L^+)$ from below, the second matrix is used to construct an approximation of $\overline{\mathrm{VaR}}_\alpha(L^+)$ from above. Separately for each of these matrices, the RA iterates over its columns and rearranges each of them so that it is oppositely ordered to the sum of all other columns. This is repeated until the minimal row sum

$$s(X) = \min_{1 \le i \le N} \sum_{1 \le j \le d} x_{ij}$$

(for $X = (x_{ij})$ being one of the said (N, d)-matrices) changes by less than a given *(convergence) tolerance* $\varepsilon \ge 0$. The RA for $\overline{\mathrm{VaR}}_\alpha(L^+)$ thus aims at approximating the maximin problem. The algorithm reduces the variance of the conditional distribution of $L^+ | L^+ > F_{L^+}^-(\alpha)$ so that it can concentrate more of the $1 - \alpha$ mass of F_{L^+} in its tail. This pushes $\mathrm{VaR}_\alpha(L^+)$ further up. As [13] state, one then typically ends up with two matrices whose minimal row sums are close to each other and roughly equal to $\overline{\mathrm{VaR}}_\alpha(L^+)$. Note that if one iteration over all columns of one of the matrices does not lead to any change in that matrix, then each column of the matrix is oppositely ordered to the sum of all others and thus there is also no change in the minimal row sum (but the converse is not necessarily true; see below).

The version of the RA given below slightly differs from the one stated in [13]; e.g., concerning dealing with infinite quantiles, which, due to the efficiency of the implementation and in particular the sorting algorithm used, cannot be handled by `rearrange()` at this point (see the source code for an explanation). The actual implementation, see `RA()` and the underlying workhorse `rearrange()`, additionally provides more information about the computed quantities. This includes, e.g., a parameter `max.ra` determining the maximal number of columns to rearrange or the choice ε =`(abstol=)NULL` to iterate until each column is oppositely ordered to the sum of all others. The latter is typically (by far) not implied by $\varepsilon = 0$, but does not result in higher accuracy (see, e.g., the application discussed in the vignette `VaR_bounds`) and is typically very time-consuming (hence the introduction of `max.ra` in our implementation).

Algorithm 3.1 (RA for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$). Proceed as follows.

(1) Fix a confidence level $\alpha \in (0, 1)$, marginal quantile functions $F_1^-, \dots, F_d^-$, an integer $N \in \mathbb{N}$ and the desired (absolute) convergence tolerance $\varepsilon \geq 0$.

(2) Compute the lower bound:
  - (a) Define the matrix $\underline{X}^\alpha = (\underline{x}_{ij}^\alpha)$ for $\underline{x}_{ij}^\alpha = F_j^-\big(\alpha + \frac{(1-\alpha)(i-1)}{N}\big)$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$.
  - (b) Permute randomly the elements in each column of $\underline{X}^\alpha$.
  - (c) Set $\underline{Y}^\alpha = \underline{X}^\alpha$. For $1 \leq j \leq d$, rearrange the jth column of the matrix $\underline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns.
  - (d) While $s(\underline{Y}^\alpha) - s(\underline{X}^\alpha) > \varepsilon$, set $\underline{X}^\alpha$ to $\underline{Y}^\alpha$ and repeat Step (2) (c).
  - (e) Set $\underline{s}_N = s(\underline{Y}^\alpha)$.

(3) Compute the upper bound:
  - (a) Define the matrix $\overline{X}^\alpha = (\overline{x}_{ij}^\alpha)$ for $\overline{x}_{ij}^\alpha = F_j^-\big(\alpha + \frac{(1-\alpha)i}{N}\big)$, $i \in \{1, \dots, N\}$, $j \in \{1, \dots, d\}$. If (for $i = N$ and) for any $j \in \{1, \dots, d\}$, $F_j^-(1) = \infty$, adjust it to $F_j^-\big(\alpha + \frac{(1-\alpha)(N-1/2)}{N}\big)$.
  - (b) Permute randomly the elements in each column of $\overline{X}^\alpha$.
  - (c) Set $\overline{Y}^\alpha = \overline{X}^\alpha$. For $1 \leq j \leq d$, rearrange the jth column of the matrix $\overline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns.
  - (d) While $s(\overline{Y}^\alpha) - s(\overline{X}^\alpha) > \varepsilon$, set $\overline{X}^\alpha$ to $\overline{Y}^\alpha$ and repeat Step (3) (c).
  - (e) Set $\overline{s}_N = s(\overline{Y}^\alpha)$.

(4) Return $(\underline{s}_N, \overline{s}_N)$.

As mentioned earlier, the main feature of the RA is to iterate over all columns and oppositely order each of them with respect to the sum of all others (see Steps (2) (c) and (3) (c)). This procedure reduces the variance of the row sums with each rearrangement. Note that it does not necessarily reach an optimal solution of the maximin problem (see, e.g., [15, Lemma 6] for a counter-example) and thus the convergence $|\overline{s}_N - \underline{s}_N| \to 0$ is not guaranteed in this approach. The randomization of the initial input in Steps (2) (b) and (3) (b) aims at avoiding these situations and also helps reducing the run time due to avoiding the worst-case sorting. Finally, let us remark that the actual outcome of the algorithm may depend on the underlying sorting algorithm used; see our study in Section 2.3 of the vignette `VaR_bounds` concerning the possible rearranged output matrices and the influence of the underlying sorting algorithm on the outcome.

3.2 Conceptual and numerical improvements

Besides the confidence level α and the marginal quantile functions $F_1^-, \dots, F_d^-$, RA relies on the inputs $N \in \mathbb{N}$ and $\varepsilon \geq 0$. Concerning N, it needs to be “sufficiently large”, but a practitioner is left to make a choice about this value.

Another issue is the use of the *absolute* tolerance ε in the algorithm. There are two problems. The first problem is that it is more natural to use a relative instead of an absolute tolerance. Without (roughly) knowing the minimal row sum, a pre-specified absolute tolerance does not guarantee that the change in the minimal row sum from $\underline{X}^\alpha$ to $\underline{Y}^\alpha$ is of the right order (and this order depends at least on d and the chosen quantile functions). If ε is chosen to be too large, the computed bounds $\underline{s}_N$ and $\overline{s}_N$ would carry too much uncertainty, whereas if it is selected to be too small, an unnecessarily long run time results; the latter seems to be the case for [13, Table 3], where the chosen $\varepsilon = 0.1$ is roughly 0.000004 % of the computed $\overline{\mathrm{VaR}}_{0.99}(L^+)$.

The second problem is that the absolute tolerance ε is only used to check *individual* "convergence" of $\underline{s}_N$ and of $\overline{s}_N$. It does not guarantee that $\underline{s}_N$ and $\overline{s}_N$ are sufficiently close to provide a reasonable approximation to $\overline{\mathrm{VaR}}_\alpha(L^+)$. From a computational point of view, this should still be checked. Also, an implementation should return convergence and other useful information, e.g., the *relative rearrangement range* $|(\overline{s}_N - \underline{s}_N)/\overline{s}_N|$, the actual individual absolute tolerances reached when computing $\underline{s}_N$ and $\overline{s}_N$, the number of column rearrangements used, logical variables indicating whether the individual absolute tolerances have been reached, the number of column rearrangements for $\underline{s}_N$ and $\overline{s}_N$, the row sums computed after each column rearrangement, the constructed input matrices $\underline{X}^\alpha$ and $\overline{X}^\alpha$, and the corresponding rearranged, final matrices $\underline{Y}^\alpha$ and $\overline{Y}^\alpha$; see `RA()` in the R package `qrmtools` for such information.

A further problem is that the RA iterates over all d columns before checking the termination conditions; see Steps (2) (c) and (3) (c) of Algorithm 3.1. Our underlying workhorse `rearrange()` keeps track of the column rearrangements of the last d considered columns and can thus terminate after rearranging any column (not only the last one); see also Algorithm 4.1 below. This "early termination" criterion saves run time (note that the tracking overhead is rather minimal). We advise the interested reader to have a look at the source code of `rearrange()` for further numerical and run-time improvements (fast accessing of columns via lists; avoiding having to compute the row sums over all but the current column; an extended tracing feature), some of which are mentioned in the vignette `VaR_bounds`.

3.3 Empirical performance under various scenarios

In order to empirically investigate the performance of the RA, we consider two studies, each of which addresses four cases; we thus consider eight scenarios. In terms of studies, we consider the following:
Study (1) (N running, d fixed): $N \in \{2^7, 2^8, \dots, 2^{17}\}$ and $d = 20$.
Study (2) (N fixed, d running): $N = 2^8 = 256$ and $d \in \{2^2, 2^3, \dots, 2^{10}\}$.
These choices allow us to investigate the impact of the upper tail discretization parameter N (in Study (1)) and the impact of the number of risk factors d (in Study (2)) on the performance of the RA. In terms of cases, we consider the following different marginal tail behaviors based on the Pareto distribution function $F_j(x) = 1 - (1 + x)^{-\theta_j}$:
Case HH: $\theta_1, \dots, \theta_d$ form an equidistant sequence from 0.6 to 0.4; this case represents a portfolio with all marginal loss distributions being heavy-tailed (slightly increasing in heavy-tailedness).
Case LH: $\theta_1, \dots, \theta_d$ form an equidistant sequence from 1.5 to 0.5; this case represents a portfolio with marginals with different tail behaviors ranging from comparably light-tailed to very heavy-tailed distributions.
Case LL: $\theta_1, \dots, \theta_d$ form an equidistant sequence from 1.6 to 1.4; this case represents a portfolio with all marginal loss distributions being comparably light-tailed.
Case LH$_1$: $\theta_1, \dots, \theta_{d-1}$ are chosen as in Case LL and $\theta_d = 0.5$; this case represents a portfolio all marginal loss distributions being light-tailed except for the last.
To keep the studies tractable, we focus on the confidence level $\alpha = 0.99$ and the absolute convergence tolerance $\varepsilon = 0$ in all scenarios. Furthermore, we consider $B = 100$ replicated simulation runs in order to provide empirical 95 % confidence intervals for the estimated quantities; note that some of them are so tight that they are barely visible in the figures presented below. The B replications only differ due to different permutations of the columns in Steps (2) (b) and (3) (b) of Algorithm 3.1, everything else is deterministic; this allows us to study the effect of these (initial) randomization steps on the (convergence) results of the RA. Concerning the hardware used, all results were produced on an AMD 3.2 GHz Phenom II X4 955 processor with 8 GB RAM. Note that we only present detailed figures for the results of Study (1); the figures related to Study (2) can be obtained from the authors upon request.

Results of Study (1) (*N* running, *d* fixed)

The simulation results for Study (1) can be summarized as follows:

- As can be seen in Figure 4, the means over all B computed $\underline{s}_N$ and $\overline{s}_N$ converge as N increases.
- Figure 5 indicates that as N increases, so does the mean of the elapsed time (as to be expected). Overall, run time does not drastically depend on the case for our choices of Pareto margins, which is a good feature.
- Figure 6 shows that the upper limit of the bootstrapped 95 % confidence interval for the number of column rearrangements rarely exceeds $10d$ as N increases; this will be used as a default number of column rearrangements required in the ARA (see later).
- Figure 7 indicates that the rate at which the number of oppositely ordered columns (based on the final rearranged $\underline{Y}^\alpha$ and $\overline{Y}^\alpha$) decreases depends on the characteristics of the marginal distributions involved, i.e., the input matrix X. The number of oppositely ordered columns seems particularly small (for large N) in Case LL, where essentially only the last column is oppositely ordered to the sum of all others.

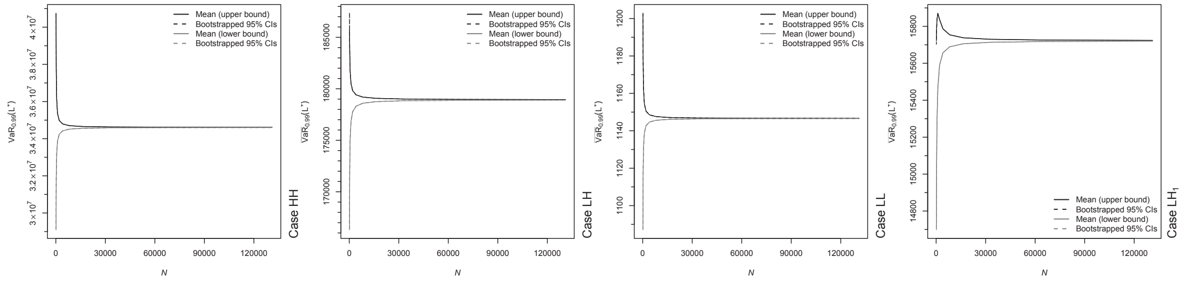

Figure 4. Study (1): $\overline{\mathrm{VaR}}_{0.99}$ bounds $\underline{s}_N$ and $\overline{s}_N$ for Cases HH, LH, LL and LH$_1$ (from left to right).

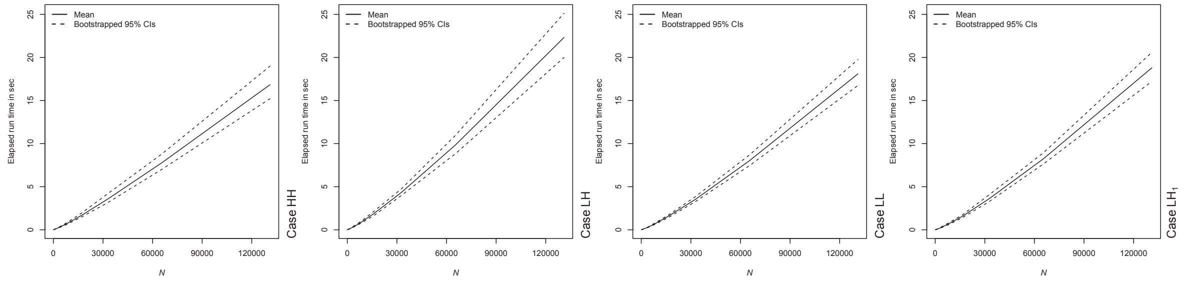

Figure 5. Study (1): Run times (in s) for Cases HH, LH, LL and LH$_1$ (from left to right).

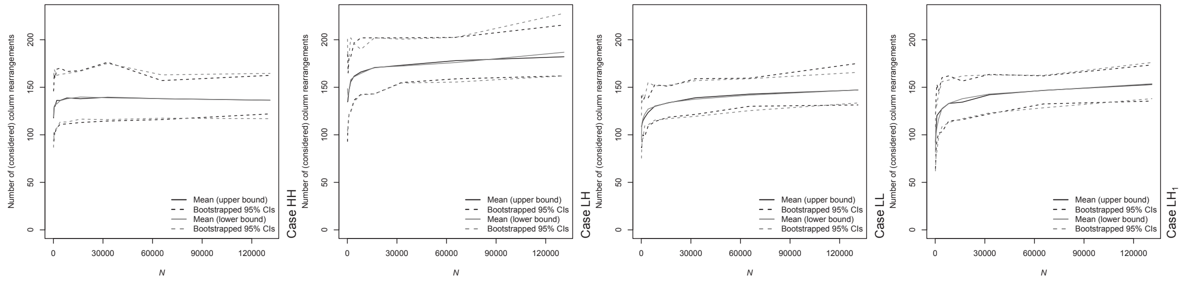

Figure 6. Study (1): Number of rearranged columns for Cases HH, LH, LL and LH$_1$ (from left to right).

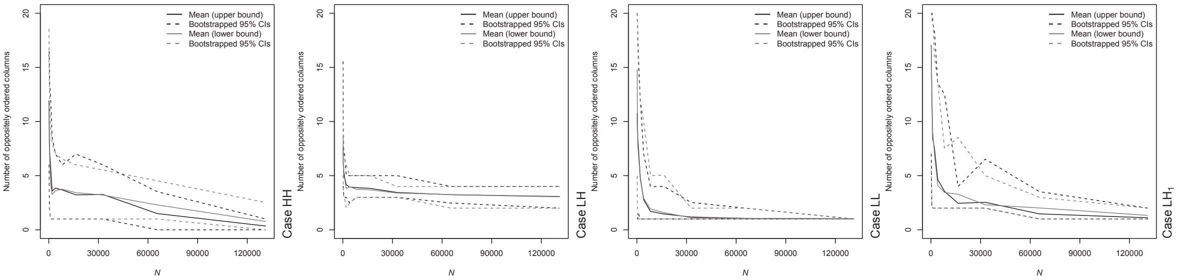

Figure 7. Study (1): Number of oppositely ordered columns (of $\underline{Y}^\alpha$ and $\overline{Y}^\alpha$) for Cases HH, LH, LL and LH$_1$ (from left to right).

Results of Study (2) (*N* fixed, *d* running)

In Study (2) we are interested in analyzing the impact of the number of risk factors d on portfolios which exhibit different marginal tail behaviors. The simulation results can be summarized as follows:

- The means over all computed $\underline{s}_N$ and $\overline{s}_N$ as functions of d diverge from one another; especially in the case where more marginal distributions are heavy-tailed. This is due to the fact that we have kept N the same for all cases in Study (2).
- Similar to what we have seen in Study (1), the mean of the run time of the RA increases as the number of risk factors increases, with Case LL in Study (2) having the smallest run time on average.
- As in Study (1), the mean of the number of rearranged columns is typically below $10d$.
- The number of oppositely ordered columns (of $\underline{Y}^\alpha$ and $\overline{Y}^\alpha$) increases as d increases; note that this finding does not contradict what we have seen in Study (1) as we have kept N constant here.

4 The Adaptive Rearrangement Algorithm

4.1 How the Adaptive Rearrangement Algorithm works

In this subsection we present an adaptive version of the RA, termed the *Adaptive Rearrangement Algorithm (ARA)*. This algorithm for computing the bounds $\underline{s}_N$ and $\overline{s}_N$ for $\underline{\mathrm{VaR}}_\alpha(L^+)$ or $\overline{\mathrm{VaR}}_\alpha(L^+)$ (as before, we focus on the latter) provides an algorithmically improved version of the RA, has more meaningful tuning parameters and adaptively chooses the number of discretization points N. The ARA is implemented in the `R` package `qrmtools`, see the function `ARA()`. Similar to our `RA()` implementation, `ARA()` also relies on the workhorse `rearrange()` and returns much more information (and can also compute $\underline{\mathrm{VaR}}_\alpha(L^+)$), but the essential part of the algorithm is given below.

Algorithm 4.1 (ARA for computing $\overline{\mathrm{VaR}}_\alpha(L^+)$). Proceed as follows.

(1) Fix a confidence level $\alpha \in (0,1)$, marginal quantile functions $F_1^-, \dots, F_d^-$, an integer vector $\boldsymbol{K} \in \mathbb{N}^l$, $l \in \mathbb{N}$, (containing the numbers of discretization points which are adaptively used), a bivariate vector of relative convergence tolerances $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2)$ (containing the individual relative tolerance ε_1 and the joint relative tolerance ε_2; see below) and the maximal number of iterations used for each $k \in \boldsymbol{K}$.

(2) For $N = 2^k$, $k \in \boldsymbol{K}$, do:

    (a) Compute the lower bound:

        (i) Define the matrix $\underline{X}^\alpha = (\underline{x}_{ij}^\alpha)$ for

$$\underline{x}_{ij}^\alpha = F_j^-\Big(\alpha + \frac{(1-\alpha)(i-1)}{N}\Big), \quad i \in \{1, \dots, N\},\ j \in \{1, \dots, d\}.$$

        (ii) Permute randomly the elements in each column of $\underline{X}^\alpha$.

        (iii) Set $\underline{Y}^\alpha = \underline{X}^\alpha$. For $j \in \{1, 2, \dots, d, 1, 2, \dots, d, \dots\}$, rearrange the jth column of the matrix $\underline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. After having rearranged at least $d + 1$ columns, set, after every column rearrangement, the matrix $\underline{X}^\alpha$ to the matrix $\underline{Y}^\alpha$ from d rearrangement steps earlier and stop if the maximal number of column rearrangements is reached or if

$$\left|\frac{s(\underline{Y}^\alpha) - s(\underline{X}^\alpha)}{s(\underline{X}^\alpha)}\right| \leq \varepsilon_1. \tag{10}$$

(iv) Set $\underline{s}_N = s(\underline{Y}^\alpha)$.

(b) Compute the upper bound:

(i) Define the matrix $\overline{X}^\alpha = (\overline{x}^\alpha_{ij})$ for

$$\overline{x}^\alpha_{ij} = F_j^-\Big(\alpha + \frac{(1-\alpha)i}{N}\Big), \quad i \in \{1, \dots, N\},\ j \in \{1, \dots, d\}.$$

If (for $i = N$ and) for any $j \in \{1, \dots, d\}$, $F_j^-(1) = \infty$, adjust it to $F_j^-(\alpha + \frac{(1-\alpha)(N-1/2)}{N})$.

(ii) Permute randomly the elements in each column of $\overline{X}^\alpha$.

(iii) Set $\overline{Y}^\alpha = \overline{X}^\alpha$. For $j \in \{1, 2, \dots, d, 1, 2, \dots, d, \dots\}$, rearrange the jth column of the matrix $\overline{Y}^\alpha$ so that it becomes oppositely ordered to the sum of all other columns. After having rearranged at least $d + 1$ columns, set, after every column rearrangement, the matrix $\overline{X}^\alpha$ to the matrix $\overline{Y}^\alpha$ from d rearrangement steps earlier and stop if the maximal number of column rearrangements is reached or if

$$\left|\frac{s(\overline{Y}^\alpha) - s(\overline{X}^\alpha)}{s(\overline{X}^\alpha)}\right| \leq \varepsilon_1. \tag{11}$$

(iv) Set $\overline{s}_N = s(\overline{Y}^\alpha)$.

(c) Determine convergence based on both the individual and the joint relative convergence tolerances: If (10) and (11) hold, and if

$$\left|\frac{\overline{s}_N - \underline{s}_N}{\overline{s}_N}\right| \leq \varepsilon_2,$$

break.

(3) Return $(\underline{s}_N,\ \overline{s}_N)$.

Concerning the choices of tuning parameters in Algorithm 4.1, note that if $\boldsymbol{K} = (k = \log_2 N)$, i.e., if we have a single number of discretization points, then the ARA reduces to the RA but uses the improved relative instead of absolute tolerances and not only checks what we termed the *individual (convergence) tolerance*, i.e., the tolerance ε_1 for checking "convergence" of $\underline{s}_N$ and of $\overline{s}_N$ individually, but also the *joint (convergence tolerance)*, i.e., the relative tolerance ε_2 between $\underline{s}_N$ and $\overline{s}_N$; furthermore, termination is checked after each column rearrangement (which is also done by our implementation `RA()`). As our simulation studies in Section 3.3 suggest, useful (conservative) defaults for $\boldsymbol{K}$ and the maximal number of column rearrangements are $\boldsymbol{K} = (8, 9, \dots, 19)$ and $10d$, respectively. Given the ubiquitous model risk and the (often) rather large values of $\overline{\mathrm{VaR}}_\alpha(L^+)$ (especially in heavy-tailed of test cases), a conservative choice for the relative tolerance $\boldsymbol{\varepsilon}$ may be $\boldsymbol{\varepsilon} = (0,\ 0.01)$; obviously, all these values can be freely chosen in the actual implementation of `ARA()`.

4.2 Empirical performance under various scenarios

In this subsection, we shall empirically investigate the performance of the ARA. To this end, we consider $d \in \{20, 100\}$ risk factors, paired with Cases HH, LH, LL, LH_1 as in Section 3.3. The considered relative joint tolerances are 0.5 %, 1 % and 2 % and we investigate the results for the individual relative tolerances 0 % and 0.1 %. Therefore the performance of ARA is investigated in 48 different scenarios. As before, the results shown are based on $B = 100$ simulations and we investigate the $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ and $\overline{s}_N$, the N *used* on the final column rearrangement of ARA (i.e., the $N = 2^k$, $k \in \boldsymbol{K}$ for which the algorithm terminates), the run time (in s), the number of column rearrangements (measured for the N used on termination of the algorithm) and the number of oppositely ordered columns after termination of the algorithm.

Results

We first consider the results for the individual relative tolerance fixed at $\varepsilon_1 = 0$.

Our findings (see Figures 8–11) indicate that:

- Although for both $d = 20$ and $d = 100$ the length of the confidence intervals for $\overline{\mathrm{VaR}}_{0.99}(L^+)$ can be checked to be increasing as the joint relative tolerance ε_2 gets larger, the mean and lower and upper confidence bounds remain fairly close to each other. More importantly for a fixed individual relative tolerance, as ε_2 increases, we do not observe a drastic shift in both lower and upper bounds for the mean across different examples.
- An important observation about the N in the ARA is that in virtually all examples, the 95 % confidence interval remains the same; this fact can be leveraged in practice for portfolios which exhibit the same marginal tail behavior to reduce the run time of the ARA.
- Across all of the 24 scenarios, doubling the joint relative tolerance reduces the run time (measured in s) by more than 50 %.
- The number of column rearrangements for the N used remains below $10d$.
- Finally, as Figures 8–11 reveal, randomizing the input matrix X has a minimal impact on various outputs of the ARA. However, this randomization has an interesting effect on the run time in that it seems to avoid the worst case in which sorting of a lot of numbers is required when oppositely ordering the columns causing the algorithm to take quite a bit longer.

Concerning the effect of the choice $\varepsilon_1 = 0.001$, the findings are overall very similar. Note, however, that Figure 13 (based on the twenty constituents of the SMI from 2011-09-12 to 2012-03-28) which visualizes the inaccuracy possible appearing if ε_1 is chosen too large; hence our default $\varepsilon_1 = 0$ of `RA()` and `ARA()`.

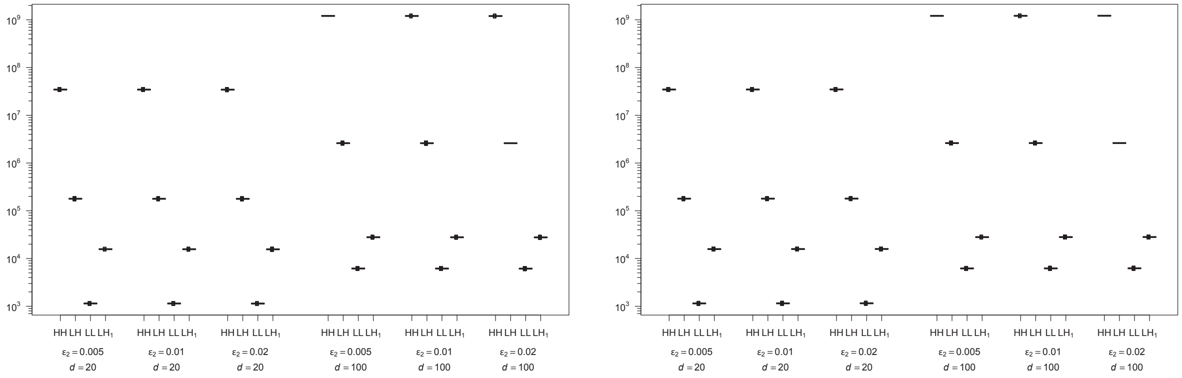

Figure 8. Boxplots of the lower and upper $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left-hand side) and $\overline{s}_N$ (right-hand side) computed with the ARA for `itol=0` based on $B = 100$ replications.

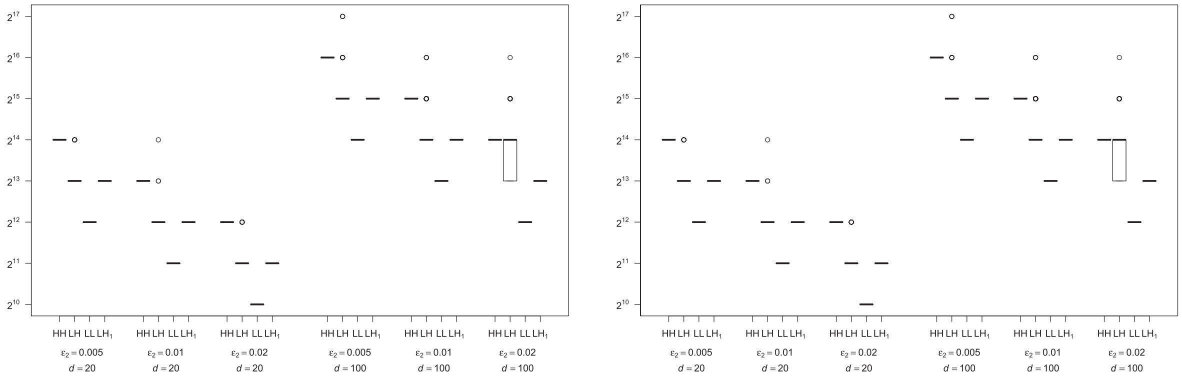

Figure 9. Boxplots of the actual $N = 2^k$ used for computing the lower and upper $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left-hand side) and $\overline{s}_N$ (right-hand side) with the ARA for `itol=0` based on $B = 100$ replications.

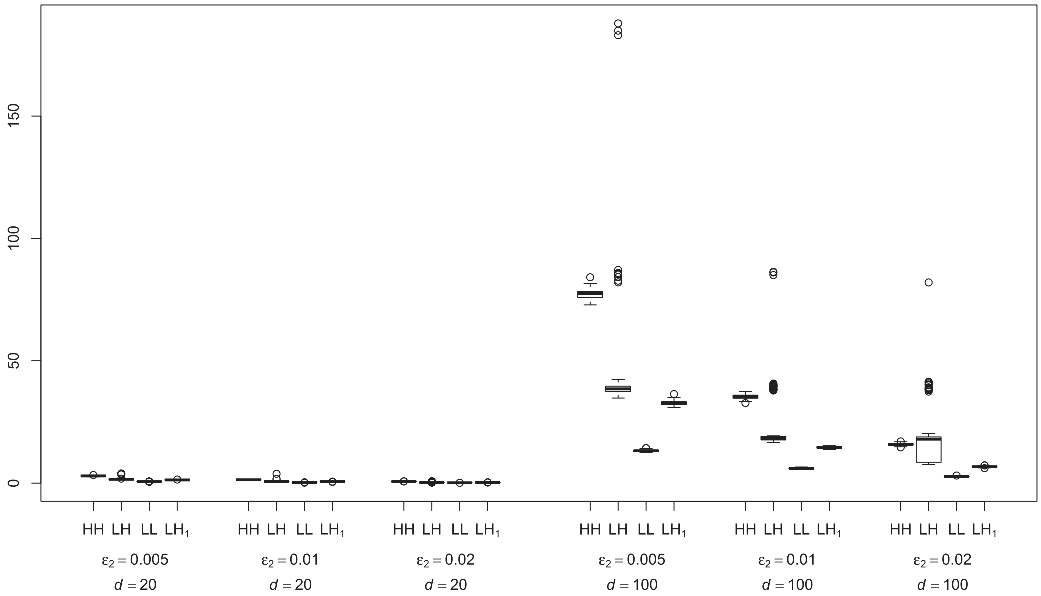

Figure 10. Boxplots of the run time (in s) for computing the lower and upper $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left-hand side) and $\overline{s}_N$ (right-hand side) with the ARA for itol=0 based on $B = 100$ replications.

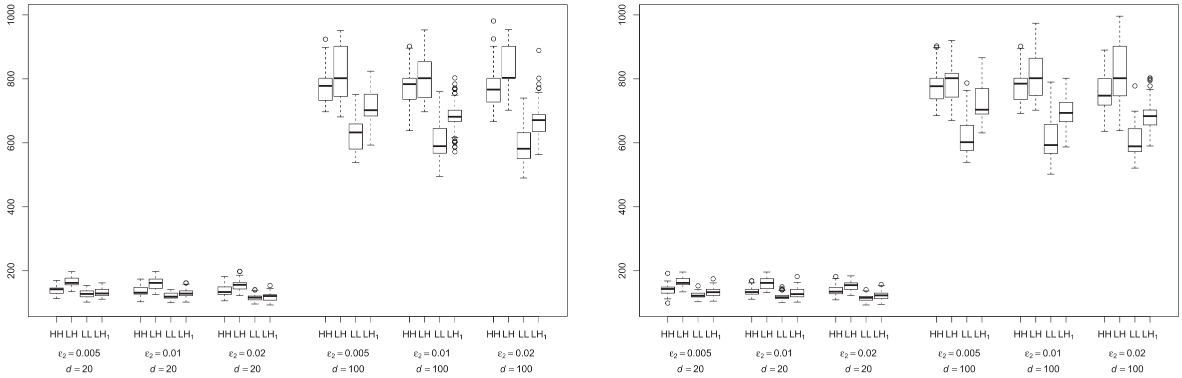

Figure 11. Boxplots of the number of column rearrangements (measured for the N used) for computing the lower and upper $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left-hand side) and $\overline{s}_N$ (right-hand side) with the ARA for itol=0 based on $B = 100$ replications.

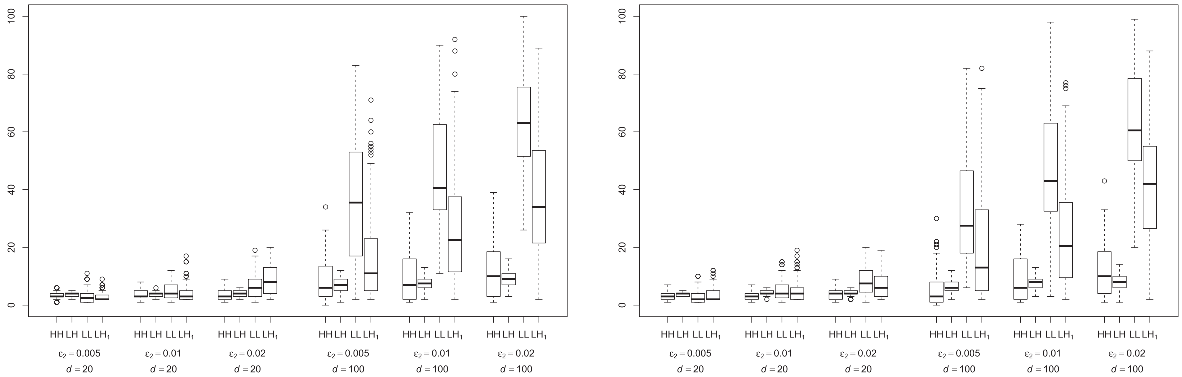

Figure 12. Boxplots of the number of oppositely ordered columns for computing the lower and upper $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ (left-hand side) and $\overline{s}_N$ (right-hand side) with the ARA for itol=0 based on $B = 100$ replications.

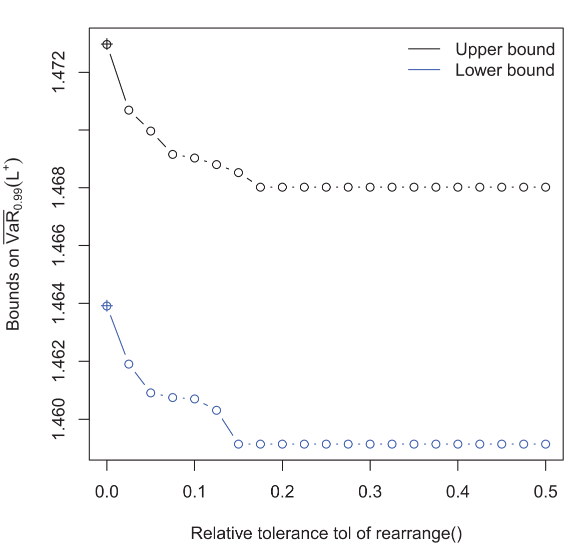

Figure 13. Computed $\overline{\mathrm{VaR}}_{0.99}(L^+)$ bounds $\underline{s}_N$ and $\overline{s}_N$ (with `rearrange()`) depending on the chosen relative tolerance ε_1 (the crosses "+" indicate the values for ε_1 = `NULL`) for an application to SMI constituents data from 2011-09-12 to 2012-03-28; see the vignette `VaR_bounds` for more details.

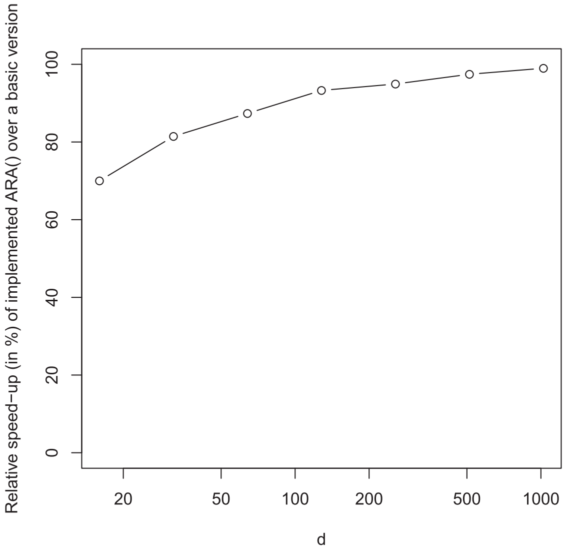

Figure 14. Relative speed-up (in %) of the actual implementation of `ARA()` over a basic version as given in the vignette `VaR_bounds`.

4.3 A comparison with an asymptotic result for large *d*

Another interesting practical question not addressed so far is whether, given existence of the first moments of all marginal distributions, asymptotic sharp bounds by worst expected shortfall $\overline{\mathrm{ES}}_\alpha(L^+)$ can replace the need of a computational tool such as the (A)RA. Since expected shortfall is subadditive and comonotone additive, the worst expected shortfall $\overline{\mathrm{ES}}_\alpha(L^+)$ is attained for the sum of the marginal expected shortfalls. Under suitable conditions, it is known that

$$\lim_{d\uparrow\infty} \frac{\overline{\mathrm{ES}}_\alpha(L^+)}{\overline{\mathrm{VaR}}_\alpha(L^+)} = 1;$$

see [20] or [17, Proposition 8.36].

Figure 15 shows $d \mapsto \overline{\mathrm{ES}}_\alpha(L^+)/\,\overline{\mathrm{VaR}}_\alpha(L^+)$ for different confidence levels α under Cases $\mathrm{HH}^{\mathrm{shifted}}$, $\mathrm{LH}^{\mathrm{shifted}}$, $\mathrm{LL}^{\mathrm{shifted}}$ and $\mathrm{LH}_1^{\mathrm{shifted}}$ which simply correspond to Cases HH, LH, LL and LH_1 where each marginal θ_j is increased by one; this is done to ensure that the first moment of all considered marginal distributions (and thus expected shortfall) exists. Worst Value-at-Risk $\overline{\mathrm{VaR}}_\alpha(L^+)$ is computed as the mean of the bounds $\underline{s}_N$ and $\overline{s}_N$ obtained from the ARA (with default values). Figure 15 also displays 2 % bounds around 1.

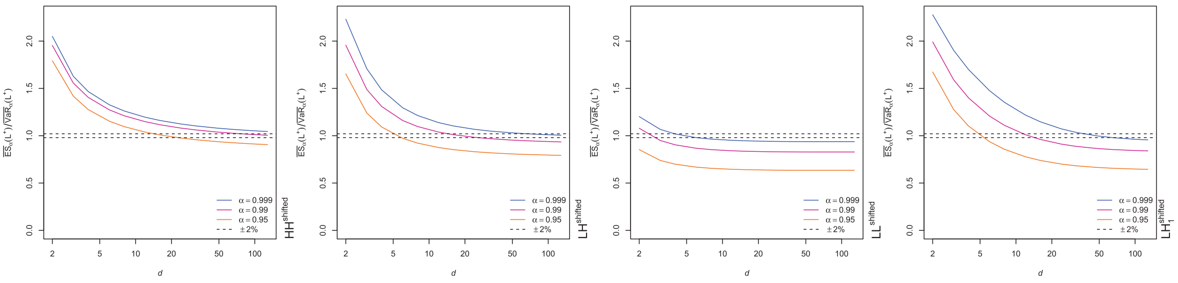

Figure 15. Worst expected shortfall $\overline{\mathrm{ES}}_\alpha(L^+)$ over worst Value-at-Risk $\overline{\mathrm{VaR}}_\alpha(L^+)$ as functions of the dimension d for different confidence levels α. The considered cases $\mathrm{HH}^{\mathrm{shifted}}$, $\mathrm{LH}^{\mathrm{shifted}}$, $\mathrm{LL}^{\mathrm{shifted}}$ and $\mathrm{LH}_1^{\mathrm{shifted}}$ (from left to right) are as before, but each parameter is increased by 1 so that $\overline{\mathrm{ES}}_\alpha(L^+)$ exists.

As can be seen from Figure 15, the convergence rate of $\overline{\mathrm{VaR}}_\alpha(L^+)$ to $\overline{\mathrm{ES}}_\alpha(L^+)$ for large d seems to largely depend on the chosen confidence level α. Only for very large $\alpha \in (0, 1)$ (as required for credit risk and operational risk within the Basel II framework), the convergence rate seems to be acceptable.

5 Conclusion

This paper presents two contributions to the computation of Value-at-Risk bounds for a sum of losses with given marginals in the context of Quantitative Risk Management.

First, we considered the homogeneous case (i.e., all margins being equal) and addressed two approaches for computing Value-at-Risk bounds. We identified and overcame several numerical and computational hurdles in their implementation and addressed them using the R package qrmtools including the vignette VaR_bounds. We covered several numerical aspects such as how to compute initial intervals for the root-finding procedures involved or that care has to be taken when choosing the tolerance of the root-finding procedures; a particular example which highlights the *numerical challenges* when computing Value-at-Risk bounds in general is the case of equal Pareto margins for which we also showed uniqueness of the root even in the infinite-mean case.

Overall, there is an important difference between implementing a specific model (say, with Par(2) margins) where initial intervals can be guessed or found by trial and error and the proper implementation of a result such as [12] in the form of an black-box algorithm; see the source code of qrmtools for the work required to go in this direction and the technical details not presented here.

Second, we considered the inhomogeneous case. We first investigated the Rearrangement Algorithm, which, by now, has been widely adopted by the industry (see also https://www.sites.google.com/site/rearrangementalgorithm/). Its tuning parameters were shown to impact the algorithm's performance and thus need to be chosen with care. We therefore presented an improved version of the Rearrangement Algorithm termed the Adaptive Rearrangement Algorithm. The latter improves the former in that it addresses the aforementioned tuning parameters and improves on the underlying algorithmic design. The number of discretization points is chosen automatically in an adaptive way (hence the name of the algorithm). The absolute convergence tolerance is replaced by two relative convergence tolerances. Since they are relative tolerances, their choice is more intuitive. The first relative tolerance is used to determine the individual convergence of the lower bound $\underline{s}_N$ and the upper bound $\overline{s}_N$ for worst (or best) Value-at-Risk. The second relative tolerance is introduced to control how far apart $\underline{s}_N$ and $\overline{s}_N$ are. The Adaptive Rearrangement algorithm has been implemented in the R package qrmtools, together with conservative defaults. The implementation contains several other improvements as well (e.g., fast accessing of columns via lists; avoiding having to compute the row sums over all but the current column; an extended tracing feature). Both the algorithmic and the numerical improvements may be of interest in any context in which it is required to find columnwise permutations of a matrix such that the minimal (maximal) row sum is maximized (minimized).

There are several interesting questions left to be investigated. Besides the evaluation of the numerical complexity of the algorithm, the theoretical convergence properties of the (Adaptive) Rearrangement Algorithm remain an open problem. Also, it remains unclear how the rows and columns of the input matrices for the (A)RA can be set up in an initialization such that the run time is minimal. Another question is whether one can reduce run time when using the rearranged matrix from the case where $N = 2^{k-1}$ to construct the initial matrix for the case where $N = 2^k$. It remains an open question whether the overhead of building such initial matrices is compensated by the resulting reduced run time; at the moment, randomization seems difficult to beat in this regard.

Acknowledgment: We would like to thank Giovanni Puccetti (University of Milano), Ruodu Wang (University of Waterloo), Kurt Hornik (Vienna University of Economics and Business) and the reviewers for valuable comments and feedback on the paper. We would also like to thank Robert Stelzer (Editor-in-Chief) and the Associate Editor for handling our paper. All remaining errors and omissions are ours.

References

[1] C. Bernard, X. Jiang and R. Wang, Risk aggregation with dependence uncertainty, *Insurance Math. Econom.* **54** (2014), 93–108.

[2] C. Bernard and D. McLeish, Algorithms for finding copulas minimizing convex functions of sums, preprint (2015), http://arxiv.org/abs/1502.02130.

[3] C. Bernard, L. Rüschendorf and S. Vanduffel, Value-at-risk bounds with variance constraints, preprint (2013), http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2342068.

[4] C. Bernard, L. Rüschendorf and S. Vanduffel, Value-at-risk bounds with variance constraints, preprint (2015), http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2342068.

[5] C. Bernard, L. Rüschendorf, S. Vanduffel and J. Yao, How robust is the value-at-risk of credit risk portfolios?, *Eur. J. Finance* **23** (2017), no. 6, 507–534.

[6] C. Bernard and S. Vanduffel, A new approach to assessing model risk in high dimensions, *J. Banking Finance* **58** (2015), 166–178.

[7] C. Bernard and S. Vanduffel, Quantile of a mixture with application to model risk assessment, *Dependence Model.* **3** (2015), 172–181.

[8] V. Bignozzi, G. Puccetti and L. Rüschendorf, Reducing model risk via positive and negative dependence assumptions, *Insurance Math. Econom.* **61** (2015), no. 1, 17–26.

[9] V. Chavez-Demoulin, P. Embrechts and M. Hofert, An extreme value approach for modeling operational risk losses depending on covariates, *J. Risk Insurance* **83** (2016), no. 3, 735–776.

[10] E. G. Coffman and M. Yannakakis, Permuting elements within columns of a matrix in order to minimize maximum row sum, *Math. Oper. Res.* **9** (1984), no. 3, 384–390.

[11] P. Embrechts and M. Hofert, A note on generalized inverses, *Math. Oper. Res.* **77** (2013), no. 3, 423–432.

[12] P. Embrechts and G. Puccetti, Bounds for functions of dependent risks, *Finance Stoch.* **10** (2006), no. 3, 341–352.

[13] P. Embrechts, G. Puccetti and L. Rüschendorf, Model uncertainty and VaR aggregation, *J. Banking Finance* **37** (2013), no. 8, 2750–2764.

[14] P. Embrechts, G. Puccetti, L. Rüschendorf, R. Wang and A. Beleraj, An academic response to Basel 3.5, *Risks* **2** (2014), no. 1, 25–48.

[15] U.-U. Haus, Bounding stochastic dependence, complete mixability of matrices, and multidimensional bottleneck assignment problems, preprint (2014), http://arxiv.org/abs/1407.6475.

[16] M. Hofert and M. V. Wüthrich, Statistical review of nuclear power accidents, *Asia-Pac. J. Risk Insurance* **7** (2012), DOI 10.1515/2153-3792.1157.

[17] A. J. McNeil, R. Frey and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques, Tools*, 2nd ed., Princeton University Press, Princeton, 2015.

[18] G. Puccetti and L. Rüschendorf, Bounds for joint portfolios of dependent risks, *Stat. Risk Model.* **29** (2012), no. 2, 107–132.

[19] G. Puccetti and L. Rüschendorf, Computation of sharp bounds on the distribution of a function of dependent risks, *J. Comput. Appl. Math.* **236** (2012), no. 7, 1833–1840.

[20] G. Puccetti and L. Rüschendorf, Asymptotic equivalence of conservative VaR- and ES-based capital charges, *J. Risk* **16** (2014), no. 3, 3–22.

[21] G. Puccetti, L. Rüschendorf, D. Small and S. Vanduffel, Reduction of value-at-risk bounds via independence and variance information, *Scand. Actuar. J.* (2015), DOI 10.1080/03461238.2015.1119717.

[22] G. Puccetti and R. Wang, Extremal dependence concepts, *Stat. Sci.* **30** (2015), no. 4, 485–517.

[23] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, Berlin, 1998.

[24] L. Rüschendorf, On the multidimensional assignment problem, *Methods OR* **47** (1983), 107–113.

[25] R. Wang, L. Peng and J. Yang, Bounds for the sum of dependent risks and worst value-at-risk with monotone marginal densities, *Finance Stoch.* **17** (2013), no. 2, 395–417.