An Algorithm for Sequential Tail Value at Risk for Path-Independent Payoffs in a Binomial Tree

Berend Roorda*

Revised version: February 19, 2010

Abstract

We present an algorithm that determines Sequential Tail Value at Risk (STVaR) for path-independent payoffs in a binomial tree. STVaR is a dynamic version of Tail-Value-at-Risk (TVaR) characterized by the property that risk levels at any moment must be in the range of risk levels later on. The algorithm consists of a finite sequence of backward recursions that is guaranteed to arrive at the solution of the corresponding dynamic optimization problem. The algorithm makes concrete how STVaR differs from TVaR over the remaining horizon, and from recursive TVaR, which amounts to Dynamic Programming. Algorithmic aspects are compared with the cutting-plane method. Time consistency and comonotonicity properties are illustrated by applying the algorithm on elementary examples.

Keywords: Value at Risk, Tail Value at Risk, Dynamic Risk Measures, Time Consistency, Dynamic Programming, Path Dependency, Cutting Plane Method

1 Introduction

A wide range of problems in applied science involve the optimization of a performance criterion under risk limits that guarantee a desired or required level of safety. In finance, the dominant approach to express risk limits is in terms of Value-at-Risk (VaR) (Morgan J. P. Inc 1996; Jorion 1997; Duffie and Pan 1997), being the maximum loss over a give time horizon at a certain confidence level. The key to its success is that it expresses risk as a monetary value with a transparent interpretation, which is very helpful in comparing and aggregating risks originating from different sources. The dominance of VaR in the financial

*B. Roorda, FELab and School of Management and Governance, University of Twente, P.O. Box 217, 7500 AE, Enschede, the Netherlands. Phone: +3153-4894383. E-mail: b.roorda@utwente.nl. Research supported in part by Netspar

industry is apparent from its central role in the world-wide regulation of banks for all risk categories, including operational risk (BIS 2006). Although VaR inherently has a financial flavor, already in the name itself, it is applicable in a non-financial context as well, if all types of risk under consideration can be quantified in a common unit of value loss, see e.g. Tapiero (2003) for an application in inventory control.

A well-known shortcoming of using VaR as risk limit is that it stimulates concentration of risk, because it is insensitive for the actual level of the worst losses that can be ignored under a given confidence level. In the financial industry, with an abundance of opportunities to exploit any loophole at large scale, this aspect can be really harmful, and has led to considerable interest in TVaR as an alternative, also called Average VaR, Conditional VaR, or Expected Shortfall (Artner et al. 1999; Szegö 2002; Föllmer and Schied 2004; McNeil et al. 2005; Pflug 2007). TVaR measures the expected loss on the probability mass that is ignored in VaR, thus avoiding the anomalies in VaR as risk limit.

We refer to Rockafellar and Uryasev (2002) for a fundamental result that links portfolio optimization under TVaR constraints to Linear Programming. A strong motivation for working with this type of constraints is the connection with optimization under second order stochastic dominance constraints, see e.g. Fishburn (1964), Föllmer and Schied (2004), Dentcheva and Ruszczyński (2003). Considerable effort has been put in efficient algorithms that can cope with the huge amount of restrictions in the corresponding LP-problems, in particular by means of cutting-plane methods, see Klein Haneveld and van der Vlerk (2006), Künzi-Bay and Mayer (2006), Rudolf and Ruszczyński (2008), Luedtke (2008), Fábián et al. (2009).

The aim of this paper, however, is not primarily related to the controversy VaR vs. TVaR (we take our starting point in TVaR, and indicate how to derive a corresponding VaR-version), nor to optimizing performance under VaR-like restrictions (we only compute the outcome of these constraints for a given position).

Our primary focus is *the dynamics of risk measurement itself.* Our findings suggest that the evaluation of dynamic risk measures requires a new class of algorithms, more complex than Dynamic Programming, yet with sufficient structure to maintain some weaker, iterative form of backward recursive evaluation.

In fact, it is surprisingly difficult to extend a static notion of risk to a multiperiod setting, without violating certain compelling rules for the consistency of risk levels over time. The literature on dynamic risk measures and their time consistency properties is rapidly growing, but here we just briefly sketch the situation for VaR and TVaR. Straightforward extensions, such as TVaR over the remaining horizon, are severely time-inconsistent, in the sense that initial risk levels may decrease with probability one in the next period (Artzner et al. 2007; see also the discussion in Roorda and Schumacher (2007), henceforth RS07). An obvious way to avoid time inconsistency is to adhere to a backward recursive definition, corresponding to so-called (strongly) time consistent risk measures, satisfying (10.1), but for TVaR this leads to accumulation of conservatism, as explained in RS07. In continuous time such strongly time consistent versions do not even exist, cf. Kupper and Schachermayer (2009) and Delbaen (2006).

Sequential Tail-Value-at-Risk (STVaR) has been introduced in RS07 as a weakly time consistent dynamic version of static TVaR. On the one hand, it avoids the type of time inconsistency as indicated above, by imposing so-called *sequential consistency*, which is the property that risk levels should never increase or decrease *for sure*, as expressed in (10.2). In fact, STVaR is the most conservative risk measure with this property that is dominated by TVaR over the entire (and remaining) horizon. On the other hand, accumulation of conservatism is avoided by deliberately giving up the backward recursive structure corresponding to strong time consistency. We remark that STVaR does not involve any extra parameters, besides the confidence level, unlike the proposal for multiperiod TVaR in Pflug and Römisch (2007).

We present an algorithm for computing STVaR in a binomial tree model, for a path-independent payoff. In RS07 it already has been shown that the optimization related to STVaR amounts to a Linear Programming problem, but a straightforward LP-implementation is infeasible already at a moderate scale, due to the path dependency inherent in STVaR.

The algorithm presented here exploits more specific features of the problem, that allow for a solution by a finite sequence of backward recursions, despite the fact that it is, for reasons indicated above, *not* strongly time consistent and hence does not follow the standard backward recursive scheme of Dynamic Programming. A simple example is used to illustrate the working of the algorithm, and to indicate the contrast with cutting-plane methods.

The paper is organized as follows. In Section 2 we repeat the definition of STVaR, and reformulate it as an optimization problem over admissible weighting functions. Section 3 introduces the notion of τ-path independency. The algorithm is described in Section 4 (outline) and 5 (implementation). The proof of correctness can be found in Section 7, after an explanation how the output of the algorithm should be interpreted in terms of weighting functions. In Section 8 the working of the algorithm is further explained by an example. A comparison with cutting-plane methods is made in Section 9. Time consistency aspects are discussed in Section 10, and conclusions follow in Section 11. The Appendix contains two proofs.

1.1 Notation

Throughout the paper we work with a standard recombining binomial tree with T steps. The root of the tree is denoted as $\mathbf{0}$. Each node ν of depth less than T has child nodes νu and νd; these links will be depicted in figures by resp. an up- and a down-branch. Because the tree recombines, $(\nu u)d = (\nu d)u$. The set of nodes of depth t, henceforth referred to as nodes *at time* t, are denoted by N_t; in particular $N_0 = \{\mathbf{0}\}$ and N_T is the set of $T+1$ end-nodes (or leaves). Further, $N = \cup_{t=0,\ldots,T} N_t$ is the set of all nodes, and $N' := N \setminus N_T$ denotes the set of pre-final (or internal) nodes. The subtree with root ν is indicated as $S(\nu)$. A *path* is a sequence of connected nodes. A *full* path starts in the root and ends in N_T.

We work with an encompassing probability space $(\Omega, \mathcal{F}, P)$, with outcome space Ω identified with full paths in the binomial tree, i.e., $\Omega = \{(\mathbf{0}, \nu_1, \ldots, \nu_T) \,|\, \nu_i \in N_i, \nu_{i+1} = \nu_i u \text{ or } \nu_{i+1} = \nu_i d\}$, and $\mathcal{F}$ is the collection of all subsets of Ω. Assuming a fixed probability $p \in (0,1)$ for an up-branch, P is defined by $P(\omega) = p^k(1-p)^{T-k}$ with k the number of up-branches in the path ω.

For a given $\omega = (\mathbf{0}, \nu_1, \ldots, \nu_T) \in \Omega$, let the corresponding partial path starting in s and ending in t be denoted by $\omega_{[s,t]} := (\nu_s, \ldots, \nu_t)$. For the single node $\omega_{[t,t]}$ we use the notation ω_t. We define $F(\omega_{[0,t]}) := \{\omega' \in \Omega \,|\, \omega'_{[0,t]} = \omega_{[0,t]}\}$, and $\mathcal{F}_t$ is the sub-σ-algebra generated by these sets, representing the available information at time t. $F(\nu)$ denotes the set of paths on the subtree $F(\nu)$, and the previous definitions extend in the obvious way.

A stopping time is a function $\tau : \Omega \to \{0, \ldots, T\}$ that is measurable with respect to $\mathcal{F}_t$ for all t, i.e., in obvious notation, with τ a function of $\omega_{[0,\tau]}$. The space of partial paths stopped at τ is denoted by $\Omega_\tau := \{\omega_{[0,\tau]} \,|\, \omega \in \Omega\}$, and $\mathcal{F}_\tau$ is generated by the collection $\{F(\omega_{[0,\tau]})\}_{\omega \in \Omega}$, representing the available information at time τ. $N_\tau := \{\omega_\tau \,|\, \omega \in \Omega\}$ is the collection of leaves in Ω_τ. The set $N_{<\tau} := \{\omega_t \,|\, \omega \in \Omega, t < \tau(\omega)\}$ consists of all (possibly) pre-final nodes in Ω_τ, and $N_{\leq\tau} := N_{<\tau} \cup N_\tau$. Further, for a function h defined on N_τ, we define the $\mathcal{F}_\tau$-measurable random variable $h^\tau : \omega \mapsto h(\omega_\tau)$.

We mainly restrict the attention to stopping times that amount to reaching a certain subset of nodes $S \subseteq N$ (or an end-node in N_T) for the first time, denoted as $\tau(S)$. Such stopping times have $N_{<\tau}$ disjoint from S, and $N_\tau \subseteq S \cup N_T$. This inclusion is strict when the last union contains nodes that cannot be reached without crossing S earlier.

In the notation that follows, notice that deterministic time is a special case of a stopping time τ. $\mathcal{X}$ denotes the space of real random variables on Ω, and $\mathcal{X}_\tau$ its restriction to $\mathcal{F}_\tau$-measurable variables. Elements of $\mathcal{X}_\tau$ are sometimes identified with random variables on Ω_τ, in the obvious way. Expected values are taken under P, and we use $E_\tau[X]$ as an abbreviation for $E[X|\mathcal{F}_\tau]$. We also use the notation $E_\nu[X]$, but this involves some subtleties that are explained in the next section. Further, the $\mathcal{F}_\tau$-conditional maximum of X is denoted as

$|X|_\tau := \min\{c_\tau : \Omega \to \mathbb{R} \,|\, c_\tau \in \mathcal{X}_\tau, X - c_\tau \le 0\}$, where it is understood that inequalities hold for all ω. $X \in \mathcal{X}$ is called path independent (on Ω) if $X(\omega) = x(\omega_T)$ for some function $x : N_T \to \mathbb{R}$.

2 Sequentially consistent TVaR

Let be given a binomial tree model over T periods with probability p for an up-branch, and a path-independent payoff $X \in \mathcal{X}$. STVaR at level $\alpha \in (0, 1]$ is defined as (cf. RS07)

$$\mathrm{STVaR}_\alpha(X) = \inf_{Z \in \mathcal{Z}} E[ZX] \tag{2.1}$$

with

$$\mathcal{Z} = \{Z : \Omega \to \mathbb{R} \,|\, E[Z] = 1 \text{ and } 0 \le Z \le \alpha^{-1} Z_t \ \text{ for } t = 0, \dots, T\} \tag{2.2}$$

writing Z_t for $E_t[Z]$. In particular, $0 \le Z \le \alpha^{-1}$ in $\mathcal{Z}$. It turns out to be convenient to rewrite this as

$$\mathrm{STVaR}_\alpha(X) = \inf_{W \in \mathcal{W}} E[WX]/E[W] \tag{2.3}$$

with $\mathcal{W}$ the *set of admissible weighting functions*, given by

$$\mathcal{W} = \{W : \Omega \to [0, 1] \,|\, |W| = 1 \text{ and } E_t[W] \ge \alpha |W|_t \ \text{ for } t = 0, \dots, T\}. \tag{2.4}$$

The equivalence of both formulations follows readily from taking $W = Z/|Z|$ for a given Z in $\mathcal{Z}$, or, conversely, $Z = W/E[W]$ for a given $W \in \mathcal{W}$.

For the interpretation of W, notice that Z and W only differ in scaling. Where Z represents a relative density, having unit expected value, W is a scaled version of Z so that its maximum is 1. Similarly, Z/Z_t is the conditional relative density, while $W^t := W/|W|_t$ is the same object, down-scaled to maximum value 1. To avoid ambiguity where $|W|_t = 0$, we set $W^t(\omega) = 1$ if $|W|_t(\omega) = 0$, similar to the common conventions in terms of Z. We remark that in the algorithm we will also use an alternative convention when that turns out to be more practical. Regardless the convention that will be used, the following decomposition holds true:

$$W = |W|_t W^t. \tag{2.5}$$

Intuitively, $W(\omega)$ can be seen as the survival probability of a path ω, in the sense that the contribution of a path ω to the expectation in the numerator of (2.3) equals $E[1_\omega X(\omega)]$ times the 'probability' $W(\omega)$ to 'survive' the selection of contributing paths; W^t has a similar interpretation, conditioned on survival at till time t, whereas $|W|_t$ can be interpreted as the (unconditional) probability of this condition.

3 τ-path independency

The optimal weighting function is heavily path-dependent in general, even though the position X itself is not. A full specification of W per path ω is simply not feasible for large T, because of the 2^T different paths, each involving T linear restrictions, one in each prefinal node. It is hence critical to avoid irrelevant path dependencies. In our approach the following notion plays a central role.

Definition 3.1 A random variable $Y \in \mathcal{X}$ is called τ-path independent, with respect to a given stopping time τ, if for all pairs $\omega, \omega' \in \Omega$ with $\omega_{[\tau,T]} = \omega'_{[\tau,T]}$, $Y(\omega) = Y(\omega')$.

This notion is increasingly restrictive in τ; for $\tau = 0$ the condition is void, while for $\tau = T$ it amounts to ordinary path independency.

The STVaR-algorithm is based on two findings. Firstly, as we will prove in Section 7, attention can be restricted to τ-path independent weighting schemes, for a decreasing sequence of stopping times as determined by the algorithm. Secondly, the essential features of such weighting schemes can be represented by functions on N_τ, thus avoiding a state space of exponential magnitude in T.

To this end, we use the following notation and terminology related to a given τ-path independent weighting scheme W. For $\nu \in N'$, let $\mathcal{W}^\nu$ denote the set of admissible weighting schemes on $F(\nu)$, the set of all paths in the subtree $S(\nu)$, where admissibility is defined entirely analogous to (2.4). For $\nu \in N_{\leq\tau} \cap N_t$, we define

$$|W|_\nu := \max\{W(\omega) \,|\, \omega_t = \nu \text{ and } t \leq \tau(\omega)\},$$

and W^ν as the normalized restriction of W on $F(\nu)$,

$$W^\nu(\omega_{[t,T]}) = \begin{cases} W(\omega)/|W|_\nu & \text{if } |W|_\nu > 0 \\ V & \text{if } |W|_\nu = 0 \end{cases} \tag{3.1}$$

where $V \in W^\nu$ is determined by a suitable convention, e.q. $V \equiv 1$. Notice that W^ν is well defined because W is τ-path independent, and that $|W^\nu| = 1$. Clearly $W^\nu \in \mathcal{W}^\nu$ if $W \in \mathcal{W}$. We remark that the dependency of $|W|_\nu$ on τ is not made explicit in the notation, but we will take care that this does not cause confusion when several stopping times are involved. Analogous to (2.5) we can write

$$W = |W|_\tau W^\tau, \tag{3.2}$$

so W is fully specified by the collection $\{W^\nu\}_{\nu \in N_\tau}$, and a function $w : \nu \mapsto |W|_\nu$ for $\nu \in N_\tau$. In the algorithm, w is represented as an indicator function of a set of nodes, while for the collection of conditional measures it only keeps track of the following triple of functions on $N_{\leq\tau}$,

$$\begin{aligned} y(\nu) &= E_\nu[W^\nu] && \text{'the probability mass in } \nu \text{ (under } W\text{)'} \\ g(\nu) &= E_\nu[W^\nu X] && \text{'the raw level in } \nu \text{ (under } W\text{)'} \\ f(\nu) &= g(\nu)/y(\nu) && \text{'the level in } \nu \text{ (under } W\text{)'} \end{aligned} \tag{3.3}$$

We remark that all these values can be derived from the three functions w, y^τ and g^τ on N_τ. Notice that $f(\mathbf{0}) = E[WX]/E[W]$ evaluates the STVaR criterion for W and a given position X. Admissibility of W can be characterized as follows.

Lemma 3.2 *Let be given a τ-path independent weighting scheme $W : \Omega \to [0,1]$ with the corresponding triple of functions (y, g, f) on $N_{\leq\tau}$ as defined above. W is admissible if and only if $|W| = 1$, $W^\nu \in \mathcal{W}^\nu$ for all $\nu \in N_\tau$, and*

$$y(\nu) \geq \alpha \textit{ for all } \nu \in N_{\leq\tau}. \tag{3.4}$$

Proof It is clear that the conditions are necessary; recall that in case $|W|_\nu = 0$, by convention still $W^\nu \in \mathcal{W}^\nu$, and hence $y(\nu) \geq \alpha$ also in that case. That the conditions are sufficient for admissibility of W, see (2.4), follows directly from $E_t[W](\omega) = y(\nu)|W|_\nu$ for all $\omega \in \Omega$ with $\nu = \omega_t$ and $t \leq \tau(\omega)$. □

For the intuition we remark that ordinary Tail Value at Risk, over [0,T] as a single period, only imposes the restriction (3.4) for $\nu = \mathbf{0}$, which always allows for a path independent optimal weighting function. However, it lacks the property of sequential consistency, as is illustrated by Example 10.1.

4 Outline of the algorithm

The algorithm determines a finite sequence of decreasing admissible weighting functions $W_{(0)}, \ldots W_{(K)} =: W^*$ so that W^* is a solution of (2.3). It starts with taking $W = W_{(0)} \equiv 1$, corresponding to initial probability mass one in all nodes, and computing $g(\mathbf{0}) = f(\mathbf{0}) = E[WX] = E[X]$. If $\alpha = 1$, the algorithm is already finished, so we assume that $\alpha < 1$.

The main idea behind the algorithm is simple: in each loop it maximally reduces weights of paths leading to nodes with maximum level, and it stops, roughly speaking, when the probability mass at the root has been decreased to α.

In the first step it hence maximally reduces the weight of those nodes $\nu \in N'$ that have maximum level $f(\nu) = M := \max\{X(\omega) \,|\, \omega \in \Omega\}$, which are called M-nodes. This reduces the probability mass in nodes on paths to M-nodes (not in the M-nodes themselves). Notice that also nodes ν before T can be M-nodes (under $W \equiv 1$), namely if $X = M$ on the entire subtree $S(\nu)$, cf. (3.3).

If in every pre-final node $\nu \in N'$, $P(X = M|\nu) \leq 1 - \alpha$, we can simply annihilate all weights for paths to M-nodes, i.e., set $W = 1_{X<M}$. If not, we backward recursively construct a weighting function $W \in \mathcal{W}$ that corresponds to maximal reduction at rate M in each node, respecting the STVaR condition (3.4). This typically involves weights between 0 and 1 for some paths to M-nodes, as is illustrated by the example in Section 8. Nodes that arrive at minimum probability mass α by this construction are called STVaR-nodes.

The algorithm can be stopped at this point if $y(\mathbf{0})(= E[W]) = \alpha$, i.e., if $\mathbf{0}$ itself has become an STVaR node. Then $f(\mathbf{0})(= E[WX]/E[X]) = \mathrm{STVaR}_\alpha(X)$. This also holds if the root itself has become an M-node, so if $f(\mathbf{0}) = M$, which can only happen in the first loop if X is the constant M.

For the next loop, all nodes with minimal probability mass α (if any) are collected in the set $\mathcal{S}$, and all M-nodes in Ex. These are considered as stopping nodes, and the corresponding stopping time $\tau(\mathrm{Ex} \cup \mathcal{S})$ replaces the role of T. This is justified because the constructed weighting scheme W is τ-path independent. It turns out that W^ν is an STVaR solution on the subtree $S(\nu)$, not only for $\nu \in \mathcal{S}$, but also for $\nu \in \mathrm{Ex}$, even though the probability mass in the root ν may be larger than α. Then we again perform maximal reduction, similarly as before, but now in Ω_τ, of probability mass at the maximum possible rate M, which is now decreased to $M = \max\{f(\nu) \,|\, \nu \in N_\tau \setminus \mathrm{Ex}\}$, the set of stopping nodes that not already have been exploited. New nodes with $f(\nu) = M$ and with $y(\nu) = \alpha$ are added to resp. Ex and $\mathcal{S}$.

This construction is repeated until $\tau = 0$. Then $\mathbf{0} \in \mathcal{S}$ and/or $\mathbf{0} \in \mathrm{Ex}$, and in both cases $f(\mathbf{0}) = \mathrm{STVaR}_\alpha(X)$.

5 The algorithm

Let be given $x : N_T \to \mathbb{R}$ as the specification of a path-independent position $X \in \mathcal{X}$ as function of final nodes, $x(\omega_T) = X(\omega)$. The algorithm determines $\mathrm{STVaR}_\alpha(X)$ for a given level $\alpha \in (0, 1]$. The basic variables are the triple (y, g, f) as defined in (3.3), the set $\mathcal{S}$ for collecting all nodes with $y(\nu) = \alpha$, and Ex for the set of all nodes that have functioned as an M-node in a loop; then $\tau = \tau(\mathrm{Ex} \cup \mathcal{S})$. They are initialized according to the weighting function $W \equiv 1$.

- $y(\nu) := 1$ for all $\nu \in N$.

- $g(\nu) := x(\nu)$ for $\nu \in N_T$, and, backward recursively, $g(\nu) := pg(\nu u) + (1 - p)g(\nu d)$

- $f(\nu) := g(\nu)/y(\nu) = g(\nu)$

- $\mathcal{S} := \emptyset$, $\mathrm{Ex} := \emptyset$, $\tau := T$

If $\alpha = 1$, $f(\mathbf{0}) = \mathrm{STVaR}_\alpha(X)$, and the algorithm stops, otherwise repeat the following loop as long as $\tau > 0$, or, equivalently, $y(\mathbf{0}) > \alpha$ and $\mathbf{0} \notin \mathrm{Ex}$.

```
1. $M := \max\{f(\nu) | \nu \in N_\tau \setminus \mathrm{Ex}\}$     (the reduction rate of the loop)
   $N^M := \{\nu \in N_{\leq\tau} \,|\, f(\nu) = M\}$     (the set of (new) $M$-nodes in the loop)

2. For $t = T - 1$ down to 0, for all $\nu \in (N_t \cap N_{<\tau}) \setminus N^M$     (the active nodes at $t$)
   When $f(\nu u) = M > f(\nu d)$,                                   Case (i)
   $yred := (1 - p)y(\nu d)$
   If $yred < \alpha$
       $w := (\alpha - (1 - p)y(\nu d))/(py(\nu u))$
       $y(\nu) := \alpha$,   $g(\nu) := wpg(\nu u) + (1 - p)g(\nu d)$,   $f(\nu) := g(\nu)/\alpha$      (R1)
   else
       $y(\nu) := yred$,   $f(\nu) := f(\nu d)$,   $g(\nu) := y(\nu)f(\nu)$      (R2)
   When $f(\nu d) = M > f(\nu u)$,                                   Case (ii)
   $yred := py(\nu u)$
   If $yred < \alpha$
       $w := (\alpha - py(\nu u))/((1 - p)y(\nu d))$
       $y(\nu) := \alpha$,   $g(\nu) := pg(\nu u) + w(1 - p)g(\nu d)$,   $f(\nu) := g(\nu)/\alpha$      (R3)
   else
       $y(\nu) := yred$,   $f(\nu) := f(\nu u)$,   $g(\nu) := y(\nu)f(\nu)$      (R4)
   When $f(\nu u) \in \mathrm{Ex}$                                   Case (iii)
   $yred := (1 - p)y(\nu d)$
   If $yred < \alpha$
       $\bar{y} := y(\nu) - \alpha$
       $y(\nu) := \alpha$,   $g(\nu) := g(\nu) - \bar{y}M$,   $f(\nu) = g(\nu)/\alpha$      (T1)
   else
       $y(\nu) := yred$,   $f(\nu) := f(\nu d)$,   $g(\nu) := y(\nu)f(\nu)$      (T2)
   When $f(\nu d) \in \mathrm{Ex}$                                   Case (iv)
   $yred := py(\nu u)$
   If $yred < \alpha$,
       $\bar{y} := y(\nu) - \alpha$
       $y(\nu) := \alpha$,   $g(\nu) := g(\nu) - \bar{y}M$,   $f(\nu) = g(\nu)/\alpha$      (T3)
   else
       $y(\nu) := yred$,   $f(\nu) := f(\nu u)$,   $g(\nu) := y(\nu)f(\nu)$      (T4)
   Otherwise                                                         Case (v)
       $y(\nu) := py(\nu u) + (1 - p)y(\nu d)$
```

$$g(\nu) := pg(\nu u) + (1-p)g(\nu d) \tag{T5}$$
$$f(\nu) := g(\nu)/y(\nu)$$

3. Adjust bookkeeping variables

$$\mathrm{Ex} := \mathrm{Ex} \cup N^M$$
$$\mathcal{S} := \mathcal{S} \cup \{\nu \in N_{<\tau} \,|\, y(\nu) = \alpha\}$$
$$\tau := \tau(\mathrm{Ex} \cup \mathcal{S})$$

Here ends the loop. After the last loop, $f(\mathbf{0}) = \mathrm{STVaR}_\alpha(X)$.

6 The weighting function determined by the algorithm

In this section we explain how to interpret the algorithm in terms of admissible weighting functions, and describe the structural properties that are preserved after each loop. Optimality properties are addressed in the next section.

Notation requires some extra attention in carefully discriminating between values of variable at the beginning and the end of a loop. To suppress indices, we simply write W for the weighting function corresponding to the end of the loop under consideration, and use M, $\mathcal{S}$, τ, etc. for the value of the other variables in that stage. By a subscript prev we indicate the values of variables as determined by the *previous* loop, so we write W_{prev}, M_{prev}, $\mathrm{Ex}_{\mathrm{prev}}$, $\mathcal{S}_{\mathrm{prev}}$ etc. For convenience, we write $\overline{\tau}$ for τ_{prev}, and $\overline{y}, \overline{g}, \overline{f}$ for the value of the triple as determined by the previous loop. These are hence the initial values for the loop under consideration. We also need $M_{\mathrm{next}} := \max\{f(\nu)|\nu \in N_\tau \setminus \mathrm{Ex}\}$, the reduction rate in the next loop, not to be confused with $M = \max\{\overline{f}(\nu)|\nu \in N_{\overline{\tau}} \setminus \mathrm{Ex}_{\mathrm{prev}}\}$.

We make use of an auxiliary stopping time, in between $\overline{\tau}$ and τ, given by

$$\overline{\tau}' := \tau(N^M \cup \mathrm{Ex}_{\mathrm{prev}} \cup \mathcal{S}_{\mathrm{prev}}). \tag{6.1}$$

This corresponds to reaching $N_{\overline{\tau}}$ or an M-node. The loop acts only *before* $\overline{\tau}'$, i.e., on nodes in $A := N_{<\overline{\tau}'}$, which we call the set of *active* nodes. This reflects the fact that no reductions take place at or after $\overline{\tau}'$, i.e., by definition,

$$W^{\overline{\tau}'} = W_{\mathrm{prev}}^{\overline{\tau}'}. \tag{6.2}$$

Below we show that $W_{\mathrm{prev}} \in \mathcal{W}$ is $\overline{\tau}$-path independent, hence also $\overline{\tau}'$-path independent, so that this is well defined. Correspondingly,

$$(y, g, f) = (\overline{y}, \overline{g}, \overline{f}) \text{ on } N_{\overline{\tau}'}\ . \tag{6.3}$$

This provides the 'initial' values for the loop's backward recursion in time t. Notice that if $\overline{\tau}' = 0$, the loop is ineffective, i.e., $W = W_{\mathrm{prev}}$.

The backward recursive assignments for the active nodes at time t, $\nu \in N_t \cap A$, are translated to weighting schemes as follows. In Case (i), (R1) and (R2) take the form

$$\begin{aligned}
&y(\nu) = wpy(\nu u) + (1-p)y(\nu d)\\
&g(\nu) = wpg(\nu u) + (1-p)g(\nu d) \qquad (6.4)\\
&f(\nu) = g(\nu)/y(\nu)
\end{aligned}$$

where $w = 0$ in (R1) and $0 < w < 1$ such that $y(\nu) = \alpha$ in (R2). We refer to w as the *branch-weight* of the reduction. This corresponds to

$$W^{\nu}(\omega_{[t,T]}) := \begin{cases} wW^{\nu u}(\omega_{[t+1,T]}) & \text{if } \omega_{t+1} = u \\ W^{\nu d}(\omega_{[t+1,T]}) & \text{if } \omega_{t+1} = d. \end{cases} \qquad (6.5)$$

Notice that in Case (i), $W^{\nu u} = W^{\nu u}_{\text{prev}}$, because $\nu \in N^M$. The actual reduction is due to the scaling by w, while possible reductions already determined for $W^{\nu d}$ are just transferred. The interpretation of Case (ii) is similar, with the role of up- and down-branches interchanged. The Cases (i) and (ii) perform the actual reduction of the loop, the corresponding nodes are called pre-M-nodes.

The assignment (T2) in Case (iii) corresponds to (6.5) with $w = 0$. In contrast to the case where (R2) applies, this just recognizes that the branch to νu has already been cut *before* the loop. This means that W_{prev} satisfies the same equation (6.5), also with $w = 0$, but of course with $W^{\nu d}$ replaced by $W^{\nu d}_{\text{prev}}$. If (T2) would bring $y(\nu)$ below α, it is not admissible, and (T1) is used instead, reflecting the assignment

$$W^{\nu}(\omega_{[t,T]}) := \begin{cases} 0 & \text{if } w_{t+1} = u \\ V(\omega_{[t+1,T]}) & \text{if } w_{t+1} = d \end{cases} \qquad (6.6)$$

for some admissible $V \in \mathcal{W}^{\nu d}$ that, like $W^{\nu d}$, corresponds to reduction at rate M, but to a lesser extent, so that $y(\nu) = \alpha$. The existence of such a V is shown below, and V need not be determined explicitly.

A remark on the notation is in order here, because the value of $W^{\nu d}$ is 'overwritten' by (6.6). This can only occur for paths that cross $\mathcal{S}$, hence end *after* τ. As suggested by the notation, however, $W^{\nu d}$ will always be the conditional weighting scheme on $F(\nu d)$, as defined by (3.1), for all paths to νd *that belong to* Ω_τ.

Finally, (T5) in Case (v) corresponds to (6.5) with $w = 1$. This also transfers possible reductions in child nodes, like (T2) and (T4) do, but because no branch is cut, $y(\nu) \geq \alpha$ is guaranteed, so this condition need not be checked.

The weighting scheme W resulting from the loop has the following structure. The proof is in the Appendix.

Lemma 6.1 *W is a τ-path independent admissible weighting scheme in $\mathcal{W}$ satisfying (3.3). If $\mathbf{0} \in \mathrm{Ex}$, $W = W_{\mathrm{prev}}$, otherwise it has the following properties. Firstly, $|W|_\tau = 1_B$ with $B := \{\omega \in \Omega \mid f(\omega_\tau) < M\}$. Consequently, for $\omega \in \Omega$, and $\nu := \omega_\tau$,*

$$W(\omega) = \begin{cases} 0 & \text{if } \nu \in \mathrm{Ex} \\ 1 & \text{if } \nu \in N_T \setminus \mathrm{Ex} \\ W^\nu(\omega_{[\tau,T]}) & \text{if } \nu \in \mathcal{S} \setminus \mathrm{Ex} \end{cases} \tag{6.7}$$

Secondly, $W \leq W_{\mathrm{prev}}$, $W \neq W_{\mathrm{prev}}$, and all reductions in the loop are at rate M, i.e.,

$$\overline{g} - g = M(\overline{y} - y) \geq 0 \text{ on the region } N_{\leq\tau}. \tag{6.8}$$

The stopping time τ can be given the following intuition on the basis of this lemma. It is the first moment that a path reaches minimum probability mass α and/or level at least M, and then it arrives in resp. $\mathcal{S}$ and/or Ex. If such moment does not exists, the path ends at T, outside Ex.

7 Proof of correctness

We have to show that if the algorithm stops, $f(\mathbf{0}) = \mathrm{STVaR}_\alpha(X)$, and furthermore that the number of loops is finite. The claim on correctness will be derived mainly the following optimality property of the weighting scheme W as determined at the end of a loop. We keep the notation of the previous section; in particular, (y, g, f) is the corresponding triple of functions, as defined by (3.3). Further, let

$$\mathcal{R}^\nu := \{(\widetilde{y}, \widetilde{g}, \widetilde{f}) \mid \widetilde{y} = E_\nu[V], \widetilde{g} = E_\nu[VX], \widetilde{f} = \widetilde{g}/\widetilde{y} \text{ for some } V \in \mathcal{W}^\nu\},$$

the set of *reachable* triples in ν, i.e. corresponding to an admissible weighting scheme in $\mathcal{W}^\nu$.

Lemma 7.1 *W is* optimal *at τ, i.e., for all $\nu \in N_\tau$, for all $(\widetilde{y}, \widetilde{g}, \widetilde{f}) \in \mathcal{R}^\nu$, $\widetilde{f} \geq f(\nu)$.*

The proof is in the Appendix. From this lemma, correctness of the algorithm is straightforwardly verified.

Theorem 7.2 *The algorithm terminates within $1/2(T+1)(T+2)$ loops (the number of nodes in the tree) and then $f(\mathbf{0})$ equals $\mathrm{STVaR}_\alpha(X)$.*

Proof The algorithm starts with $\mathrm{Ex} = \emptyset$, and in each loop this set is extended by at least one M-node. If the algorithm would not have terminated before the $1/2(T+1)(T+2)$-th loop, then after that loop $\mathrm{Ex} = N$, hence $\mathbf{0} \in \mathrm{Ex}$, and the algorithm stops. In fact $\mathbf{0} \in \mathrm{Ex}$

already one loop earlier, because it cannot be the case that the root is the only element outside Ex.

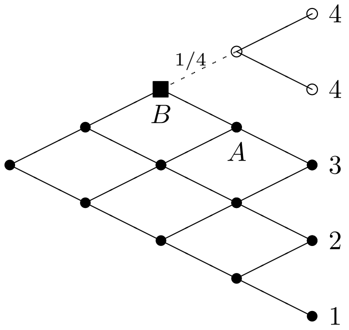
At termination, $\tau = 0$, so either $\mathbf{0} \in \mathcal{S}$ or $\mathbf{0} \in \mathrm{Ex}$. If $\mathbf{0} \in \mathcal{S}$, it follows from the previous lemma that W is optimal at $\tau = 0$, which means that W must solve the STVaR-problem (2.3). Otherwise, $\mathbf{0} \in \mathrm{Ex}$. This can only be the case if $\overline{f}(\nu) = M$ for all $\nu \in N_{\overline{\tau}}$. From (IH1) in the Appendix it then follows that W_{prev} already is optimal in $\mathbf{0}$, and indeed the last loop was ineffective, so that $W = W_{\text{prev}}$ solves the STVaR-problem (2.3). □

8 Example

We illustrate the working of the algorithm by an example. Meanwhile we discuss discuss some aspects of it that may be less obvious.

8.1 First step

We consider a binary tree with $p = 1/2$, $T = 4$, and payoff X as indicated in the picture below, with maximum value $M = 4$. $E[X] = 2\frac{15}{16}$. We apply the first loop of the algorithm for STVaR at level $\alpha = 3/8$. The end result of the first step can be visualized as follows.



The M-nodes are indicated by open circles. The pre-M-nodes are nodes A and B, they have exactly one branch to an M-node. Starting in node A, the last one, we see that the branch to its M-node can be cut completely, by (R2). Formally, we set $W^A(Au) = 0$, and keep $W^A(Ad) = 1$. This leaves node A with probability mass $1/2$, which is not below α, as required. Obviously, the level in node A after this cut is given by $f(A) = 3$.

For node B the branch to the M-node cannot be cut completely, taking into account that node A has not full probability mass anymore: this would yield probability mass $1/4$ in B, which is below α. According to (R1), setting the transition weight equal to $w = 1/4$, as depicted above, the probability mass in B is reduced to α exactly. This brings the value $f(B)$ down to $wp4 + (1-p)^2 3 = 3\frac{1}{3}$. This is actually the STVaR$_\alpha$ value of X on $F(B)$, and therefore B is called an STVaR-node.

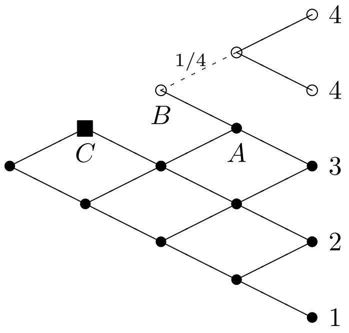
The probabilities and levels in the other nodes follow (T5), corresponding to unit transition weights. This gives $y(\mathbf{0}) = \frac{23}{32}$, $f(\mathbf{0}) = 2\frac{12}{23}$.

At the end of the loop, the stopping time τ equals $\tau(B)$. Intuitively speaking, in the backward recursions of later loops, node B will pass its STVaR_α value to earlier nodes, regardless of any further reductions of nodes in $F(B)$. So, in forward perspective, for paths through B there is no reason to 'look behind' node B. Formally, the outcome space can now be restricted to Ω_τ. Notice that the path $duuu$ still belongs to Ω_τ. It is the only path that arrives at τ in Ex, hence the only one with zero weight in Ω_τ, in line with Lemma 6.1.

We remark that it is not crucial that the reduction in node A is passed through to node B. Alternatively, one could take e.g. weight 1/6 for all paths to M-nodes via B. The choice in the algorithm is best in line with the backward recursion in time.

8.2 Second step

Taking starting point in the end result of the first step, now the maximum reduction rate M has become $3\frac{1}{3}$, and B becomes the new M-node. Reduction now takes place in the only pre-M-node C, through (R2), and it turns out that cutting its branch to B reduces its probability mass exactly to $y(C) = \alpha$. The corresponding level is $f(C) = 2\frac{2}{3}$, copied from the node below B. So the picture now becomes



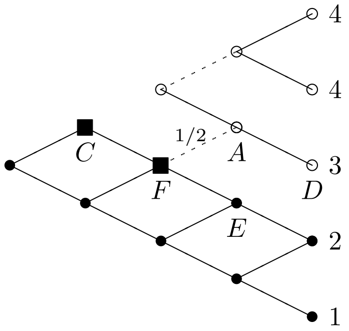
This updates the stopping time τ to $\tau(B \cup C) = \tau(C)$. Notice that B, which was already a stopping node in $\mathcal{S}$, now also belongs to Ex. At the root now $y(\mathbf{0}) = \frac{5}{8}$ and $f(\mathbf{0}) = 2\frac{2}{5}$.

Notice that not all paths from C to $X < M$ have unit weight after reduction at rate M. For example, the path $\omega = uudd$ leads to $X(\omega) = 3$, yet $W(\omega) = 0$. By giving this path some extra weight in W^C the pair $y(C), g(C)$ would increase to $y(C) + \delta, g(C) + 3\delta$. This seems to contradict Lemma 7.1, which states that the increase rate in g is at least $M = 3\frac{1}{3}$, hence cannot be 3. However, this extra weight would make W^C (and hence W) an inadmissible weighting scheme, violating (3.4) in node B. The admissible minimum rate of increase is indeed exactly $M = 3\frac{1}{3}$, corresponding to giving back the cut branch to B some positive weight.

This is the crucial difference with considering TVaR over the remaining period, as discussed in Section 10. It also illustrates that $\mathcal{W}$ is *not* closed under increasing weighting schemes bounded by 0 and 1, if it is allowed to increase the support of $\mathcal{W}$.

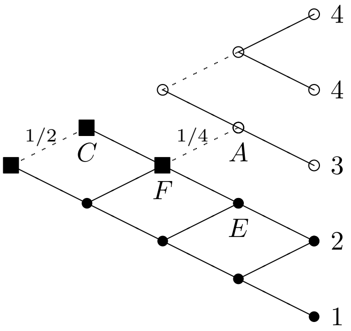
8.3 Remaining steps

In the third loop, the reduction rate is $M = 3$, and there are two M-nodes, D and A. Reduction at this rate takes place in the pre-M-nodes E by (R2) and then in F by (R1). This leads to the following situation after the third loop.



The value in F is now $2\frac{1}{3}$. The reachable pre-final stopping nodes for τ are now C and F. Notice that the reduction in F is not affecting the level of the STVaR-node C, and hence the newly constructed W^F does not apply to paths via C. This illustrates the difference between ordinary and τ-path independency. At the root, $y(\mathbf{0}) = \frac{15}{32}$, still beyond α, and $f(\mathbf{0}) = 2\frac{1}{5}$.

Finally, node C becomes M-node with level $M = 2\frac{2}{3}$. Maximum reduction at this rate corresponds to weight $w = 1/2$ for the first up-branch, and level $f(\mathbf{0}) = 2\frac{1}{12}$ for the root, which is the outcome of $\mathrm{STVaR}_\alpha(X)$. The end situation is depicted below.



We remark that (T1,3) never have been applied; they typically become relevant for smaller α and larger regions with constant payoff. An example for which (T1) is relevant is provided by taking $\alpha = 3/16$ and payoff $x(\nu)$ equal to 2,2,2,1,0 instead of 4, 4, 3, 2, 1.

8.4 Final state

The final state in the example is typical: the root has one branch to a node in $\mathcal{S}$ (u in this case), with level equal to the last reduction rate, i.e., $f(u) = M$. The other node, d in this case, has level $f(d) < M$, and probability $y(d) > \alpha$. Optimality of the final weighting scheme is reflected by the fact that in node d, probability mass can only be increased at increase rate beyond M, and decreased at rate below M. Intuitively, instead of taking the STVaR value in d as well, the node is filled with extra probability mass until the increase rate begins to exceed the level $f(u)$.

It is clear that in the final situation, in obvious notation, $\mathrm{Ex} = \{\nu \in N \mid \mathrm{STVaR}_\alpha(X|\nu) \geq M\}$. So at termination $\tau(\mathrm{Ex})$ must be the stopping time of reaching STVaR_α level M for the first time. There are hence three types of paths in $\Omega_{\tau(\mathrm{Ex})}$: those ending in T without reaching Ex, hence in $X < M$, having weight 1, those reaching $\mathcal{S}$ before stopping in Ex, having level below M, and mass α, and those having in the last step a cut link to Ex.

So, abstract from the iterations, the end result of the algorithm can be summarized as follows. It determines the region Ex of nodes where the STVaR_α level exceeds a certain level M and maximally reduces the last branch weight of paths arriving at Ex, giving priority to branches to Ex-nodes with higher level, while respecting (3.4). The level M (the final reduction rate) is determined as the highest level for which then the root gets probability mass α, or level equal to the reduction rate.

9 A comparison with cutting-plane methods

Cutting-plane methods have been successfully applied for solving portfolio optimization under TVaR constraints, as mentioned in the introduction. A natural question is, to what extent they could be applied for STVaR. Intuitively, the (primal) cutting-plane method first ignores most restrictions in a given LP-problem, then checks the solution for violated restrictions, and adds some of these to the problem, until no violated restrictions are found. In this way it detects the binding constraints out of a typically huge set of restrictions, thus reducing computation time by a factor hundred or more for applications at realistic scale, as described in the references cited in the Introduction.

We first sketch how the method could be applied to the example in the previous section. Here it is convenient to use the path-specific notation $E[W|\omega_{[0,t]}]$. An obvious choice for the initial problem is standard TVaR, i.e., (2.2) with all restrictions for $t > 0$ removed. In terms of weighting schemes $W = Z/|Z|$, this results in $W(\omega) = 1$ in case $X(\omega) \leq 2$, $W(\omega) = 1/6$ for all six paths with $X(\omega) = 3$, and $W(\omega) = 0$ otherwise. This yields $\mathrm{TVaR}(X) = 2$.

STVaR requires that $J(t, \omega) := W(\omega) - \alpha^{-1}E[W|\omega_{[0,t]}] \leq 0$ for each path $\omega \in \Omega$, and all

t. Violations occur for path *uddd* in C ($J(1, uddd) = 1/2$), for path *uudd* in B ($J(2, uudd) = 1/18$), and for both paths *uddd* and *dudd* in F ($J(2, uddd) = J(2, dudd) = 1/9$).

A choice has to be made which violated restriction(s) should be added. We choose for the restriction with the highest J-value, i.e., the one that is violated in C. This is the most severe violation, scaled in units of weights, and may be expected to have the largest effect on the solution.

So we add the restriction $\alpha W(uddd) - E[W|u] \leq 0$, and solve by LP. This yields weighting function W with weight 4/9 for the three paths through C with $X(\omega) = 3$, 2/3 for *uddd*, 1 for all other paths ending with $X(\omega) \leq 2$, and zero otherwise. The (only) violation is now for *dudd* in F, where $J(2, dudd) = 1/3$.

Adding this restriction as well in a third step, yields a solution with with weight 1/3 for the three paths through C with $X(\omega) = 3$, 1/2 for *uddd*, 1/4 for *duud* and *dudu*, and again 1 for all other paths ending with $X(\omega) \leq 2$.

Then $E[W] = \alpha$, and the outcome of $\alpha^{-1}E[WX]$ is already at the correct level $2\frac{1}{12}$, yet there still one violation to be removed, for path *uudd* at B. After adding this restriction, the same solution as from the STVaR algorithm is obtained, with the only (ineffective) difference of having weights 1/4 for paths *duud* and *dudu*, where the original solution has weights resp 1/2 and 0.

This illustrates that the cutting plane method can be quite efficient in avoiding non-effective restrictions. At the start, a number of 32 restrictions is ignored (16 at $t = 1$, 16 at $t = 2$, the ones at $t = 3$ are apparently ineffective for $\alpha = 3/8$), while only 4 were added to arrive at a feasible solution.

Path dependency, however, severely limits the number of time steps T for straightforward applications of the cutting-plane method. In larger scale applications, it is not only the number of restrictions that hurts, but already the number of variables involved. The outcome space contains 2^T elements, which effectively blocks the mere representation of the STVaR in a cutting plane routine for, say, $N = 50$, let alone the optimization involved. In contrast, the STVaR algorithm can cope with this size without any problem.

The crux of the STVaR algorithm is that it exactly knows where the path dependencies may arise, in particular at the STVaR-nodes, and that this dependency only requires to store the level in such nodes, not the weighting function on the corresponding subtree.

Clearly, more advanced applications of the cutting-plane method may also be much more effective in reducing the number of variables by avoiding irrelevant path dependencies. The development of such methods, which could also involve dual cutting-plane methods, goes beyond the scope of this paper.

10 Time consistency aspects

The *raison d'être* of STVaR is that it avoids time inconsistency problems in defining Value-at-Risk in a multi-period setting. We briefly illustrate this aspect in the context of the algorithm. First we will summarize some basic notions, see RS07 and Roorda and Schumacher (2010) for a more extensive description. This involves the extension of a risk measure ϕ_0, such as STVaR_α, to a sequence of refined risk measures $\{\phi_t\}_{t=0,\dots,T}$, also called updates of ϕ_0, that take t as initial time.

A sequence of updates $\{\phi_t\}_{t=0,\dots,T}$ is called strongly time consistent if the 'risk-equivalence' principle

$$\phi_0(\phi_t(X)) = \phi_0(X) \tag{10.1}$$

holds, and sequentially consistent if the weaker requirement

$$\phi_t(X) \geq 0 \Rightarrow \phi_u(X) \not< 0 \text{ and } \phi_t(X) \leq 0 \Rightarrow \phi_u(X) \not> 0 \text{ for } t < u \tag{10.2}$$

is satisfied. Intuitively, this means that $\phi_t(X)$ is in the range of $\phi_u(X)$, or, for binomial trees, that $\phi_\nu(X) \in \{\lambda\phi_{\nu u}(X) + (1-\lambda)\phi_{\nu d}(X) \mid \lambda \in [0,1]\}$.

A fundamental observation is that initial risk measures ϕ_0 admit only one specific update at t that has a chance of being sequentially and/or strongly time consistent, so (weak) time consistency can also be seen as *a property of the initial risk measure itself.* For a coherent risk measures ϕ_0, which is representable as the worst expected value operator over a set of probability measures, this update ϕ_t amounts to conditioning expected values on the information at t, cf. RS07. We refer to Roorda and Schumacher (2010) for a much more general result on updating convex and even non-convex risk measures, for an even weaker type of time consistency.

Applying this to $\phi_0 = \mathrm{STVaR}_\alpha[\cdot]$, as defined in (2.3), we have

$$\phi_t(X) = \inf_{W \in \mathcal{W}} E_t[WX]/E_t[W] \tag{10.3}$$

which is nothing else than STVaR_α over subtrees with roots in N_t.

It may be illuminating to compare this to taking the initial measure equal to TVaR over the entire horizon, cf. the remark after (3.4). This is not sequentially consistent, as is shown by the following small example. Other examples, involving a path-dependent payoff, can be found in RS07 and in Artzner et al. (2007).

Example 10.1 Consider a binary tree with two steps, $\Omega = \{uu, ud, du, dd\}$, and $X(uu) = 0$, $X(ud) = X(du) = 1$, and $X(dd) = -1$. Assume probability 3/4 for up, 1/4 for down. Consider TVaR_α with $\alpha = 1/2$. Then in u and d, the outcome is 0, while for the entire period $[0, T]$ the algorithm yields outcome -1/16.

STVaR is in fact the most conservative risk measure dominated by TVaR that is sequentially consistent. So, combining the restriction that risk should not be seen as higher than TVaR_α over the entire horizon, with the natural requirement that risk levels should never increase or decrease for sure, automatically leads to the STVaR concept. This also holds for each subtree separately.

The violations for TVaR, as determined in the previous section, show that TVaR allows a degree of conservatism in future states, that will be considered as too excessive when that state actually materializes. It is exactly this type of inconsistency that STVaR avoids. We remark that in other examples this difference can be much more pronounced.

On the other hand, STVaR is *not* strongly time consistent, and we argued in RS07 why this can be a desirable property, in particular in the context of capital requirements (as opposed to *pricing* measures). Examples of strongly time consistent risk measures dominated by TVaR over the entire period correspond to taking TVaR_{α_t} over each period $[t, t+1]$, e.g. $\alpha_t = \alpha^{1/T}$. These can be computed backward recursively according to the Dynamic Programming principles. This prescribes the level of TVaR in each period *a priori*, as if the time profile of most adverse events is time homogeneous or predetermined, regardless the position. Moreover, for large T there is in fact no reasonable approximation of TVaR that is strongly time consistent, and for continuous time the whole concept fails, as already indicated in the introduction. In contrast, STVaR lets the induced level of conservatism depend on the position X, without making any *ad hoc* choice on the timing of risk. It is the position X that determines whether nodes contribute at most conservative level (the ones in $\mathcal{S}$ and Ex), or less conservatively, just as it turns out to be most adverse.

This clearly indicates that STVaR is fundamentally different from strongly time consistent versions of TVaR per period, on the one hand, and TVaR over the remaining horizon, on the other. This discrepancy is further underlined by the fact that STVaR is not comonotonically additive, and hence can also not be represented as a mixture of TVaR with different confidence levels (Kusuoka 2001; Föllmer and Schied 2004, Thm 4.87). We conclude this section by a counter example for comonotonic additivity.

Example 10.2 In the same setting as the previous example, consider now the comonotone family of positions $\{X_\mu\}_{\mu\in(0,1)}$ with $X_\mu(uu) = 1$, $X_\mu(ud) = X_\mu(du) = 0$, and $X_\mu(dd) = \mu$. Take $\alpha = 3/4$ and $p = 1/2$. The STVaR algorithm starts with creating an STVaR node in the node u (in obvious notation) with corresponding level $f(u) = 1/3$. If $\mu < 1/3$, the algorithm proceeds with reducing the branch weight to u, and results in $\mathrm{STVaR}(X_\mu) = \frac{1}{9} + \frac{1}{3}\mu$. If $\mu \geq 3$, the second loop reduces the branch weight to dd, which creates first an STVaR-node in d of level $\frac{1}{3}\mu$, and then, in the same loop, determines the STVaR value of X_μ in the root as $\frac{1}{6} + \frac{1}{6}\mu$. Now it is easily verified that for a pair of positions $X_\mu, X_{\mu'}$ with $0 < \mu < 1/3 < \mu' < 1$,

$\mathrm{STVaR}(X_\mu) + \mathrm{STVaR}(X_{\mu'}) < \mathrm{STVaR}(X_\mu + X_{\mu'}) = 2\,\mathrm{STVaR}(X_{(\mu+\mu')/2})$. For instance, for $\mu = 1/6$ and $\mu' = 1/2$, the left-hand side equals $\frac{1}{6} + \frac{1}{4}$, while the right-hand side is $\frac{4}{9}$.

11 Conclusions

We showed how STVaR can be computed by a sequence of backward recursions, and used the algorithm to illustrate and motivate the difference with other versions of multiperiod TVaR. This has been done in a very simple setting, for path-independent positions on a binomial tree, so that attention can be focussed on conceptual aspects. The algorithm illustrates how exactly the weakly time consistent dynamics of risk processes can deviate from the certainty equivalence principle for value processes.

It is straightforward to generalize the results to multinomial trees, and mild forms of path dependency in the position. Time steps can be refined in order to approximate continuous time STVaR. The interpretation of τ as the first time that conditional STVaR hits a level M suggests a link with optimal stopping problems in the spirit of American option pricing. This may serve as a blueprint for computing weakly time consistent risk measures in continuous time, and to develop stochastic calculus for this type of risk processes.

The notion of path independency with respect to a stopping time was crucial in suppressing the number of parameters involved in its representation as an LP-problem. In this respect it is somewhat complementary to (primary) cutting-plane methods, which have been successfully applied in coping with a huge number of *restrictions* in the LP problem corresponding to portfolio optimization under TVaR-constraints. It would be interesting to combine the strong points of both methods for optimizing under STVaR-constraints.

Finally, we would like to emphasize that it is crucial for the acceptance of a risk measure in the industry that it allows for an absolutely transparent interpretation. In this respect VaR over a single period still sets the standard, despite its shortcomings that have been widely addressed in the literature. Now VaR can be reconstructed from TVaR by the rule

$$\mathrm{VaR}_\alpha := \lim_{\delta \searrow 0} \frac{(\alpha+\delta)\,\mathrm{TVaR}_{\alpha+\delta}}{\alpha+\delta},$$

and, inspired by this rule, one could develop a 'sequential' version of Value-at-Risk at confidence level $1-\alpha$. Such a notion can be helpful in further developing sequentially consistent risk measures that can compete with VaR in terms of transparency.

12 Appendix

Proof of Lemma 6.1

The case $\mathbf{0} \in \mathrm{Ex}$ is obvious. We will prove the two claims for the case $\mathbf{0} \notin \mathrm{Ex}$. Notice that in the first claim, indeed (6.7) follows from $|W|_\tau = 1_B$, using (3.2).

As induction hypothesis (IH) for W_{prev}, assume that W_{prev} is a $\overline{\tau}$-path independent admissible weighting scheme in $\mathcal{W}$, satisfying (3.3) with $(\overline{y}, \overline{g}, \overline{f})$, and that $|W_{\mathrm{prev}}|_{\overline{\tau}} = 1_{\overline{B}}$ with $\overline{B} := \{\omega \in \Omega \,|\, f(\omega_{\overline{\tau}}) < M_{\mathrm{prev}}$. Then (6.7) holds for W_{prev} with Ex, $\mathcal{S}$ and ν replaced by resp. $\mathrm{Ex}_{\mathrm{prev}}$, $\mathcal{S}_{\mathrm{prev}}$ and $\overline{\nu} := \omega_{\overline{\tau}}$. (IH) is easily verified for the initial weighting scheme with all weights equal to 1 (taking M_{prev} any value larger than the maximum of X).

We first analyze W_{prev} in detail. From (3.2) it follows that $E[W_{\mathrm{prev}}] = E[1_{\overline{B}} W^{\overline{\tau}}_{\mathrm{prev}}] = E[1_{\overline{B}} E_{\overline{\tau}}[W^{\overline{\tau}}_{\mathrm{prev}}]] = E[1_B \overline{y}^{\overline{\tau}}]$. Translated in terms of function triples, it must hold that

$$\overline{y}(\nu) = E_\nu[1_{\overline{B}} \overline{y}^{\overline{\tau}}] \text{ and } \overline{g}(\nu) = E_\nu[1_{\overline{B}} \overline{g}^{\overline{\tau}}] \text{ for } \nu \in N_{<\overline{\tau}}. \tag{12.1}$$

Because $1_{\overline{B}} \overline{f}^{\overline{\tau}} \leq M$, also $\overline{f} \leq M$ on $N_{<\overline{\tau}}$. For those $\nu \in N_{<\overline{\tau}}$ with both $\overline{f}(\nu u) \leq M$ and $\overline{f}(\nu d) \leq M$, the triple hence follows the straightforward backward recursion,

$$\begin{aligned} \overline{y}(\nu) &= p\overline{y}(\nu u) + (1-p)\overline{y}(\nu d) \\ \overline{g}(\nu) &= p\overline{g}(\nu u) + (1-p)\overline{g}(\nu d) \\ \overline{f}(\nu) &= \overline{g}(\nu)/\overline{y}(\nu) = \lambda \overline{f}(\nu u) + (1-\lambda)\overline{f}(\nu d) \end{aligned} \tag{12.2}$$

with $\lambda = \frac{p\overline{y}(\nu u)}{\overline{y}(\nu)} \in [0,1]$. If $\overline{f}(\nu u) > M$, then $\overline{f}(\nu d) \leq M$ (otherwise $\nu \notin N_{<\overline{\tau}}$) and

$$\overline{y}(\nu) = (1-p)\overline{y}(\nu d), \quad \overline{g}(\nu) = (1-p)\overline{g}(\nu d), \quad \overline{f}(\nu) = \overline{f}(\nu d) \tag{12.3}$$

and similarly, if $\overline{f}(\nu d) > M$, then $\overline{f}(\nu u) \leq M$, and

$$\overline{y}(\nu) = p\overline{y}(\nu u), \quad \overline{g}(\nu) = p\overline{g}(\nu u), \quad \overline{f}(\nu) = \overline{f}(\nu u). \tag{12.4}$$

According to (6.3), the loop keeps these values for all ν that are outside the active region $A = N_{<\overline{\tau}'} = N_{<\overline{\tau}} \setminus N^M$, so the relevant 'initial' values for (y, f, g) are given by

$$(y(\nu), g(\nu), f(\nu)) = (\overline{y}(\nu), \overline{g}(\nu), \overline{f}(\nu)) \text{ for } \nu \in N_{\overline{\tau}'}$$

By assumption, $\mathbf{0} \notin \mathrm{Ex}$, so $\overline{f}(\mathbf{0}) < M$, hence $\overline{\tau}' \neq 0$, and the set of active nodes A is non-empty. Regarding to the classification, notice that all active nodes $\nu \in \mathcal{A}$ must have at least one child node at level strictly below M. Now if the other one is at level M, (R1-4) applies, if its level is higher, (T1-4), and if it is also below M, (T5) is applied. So the classification used in the loop is indeed exhausting A.

The first claim, that $|W|_\tau = 1_B$, is proved at the end, because in the loop recursions it is not yet clear which nodes eventually belong to N_τ. For all other claims, we use the following (inner) induction hypothesis (ih) for the backward recursion in time, which is trivially true for $t = T$.

For all $\nu \in N_{\leq\overline{\mathcal{T}}'} \cap N_{t'}$, for all $t' \geq t$,

(ih1) $W^\nu \in \mathcal{W}^\nu$ satisfying (3.3)

(ih2) $W^\nu \leq W^\nu_{\text{prev}}$, and for every $\omega \in \Omega$ for which the inequality is strict, $f(\omega_{\overline{\mathcal{T}}'}) = M$

(ih3) for all $\widetilde{y} \in [\underline{y}(\nu), \overline{y}(\nu)]$, there is a $V' \in \mathcal{W}^\nu$ with $E_\nu[V] = \widetilde{y}$, $W^\nu \leq V' \leq W^\nu_{\text{prev}}$, also satisfying the strictness property in (ih2)

A straightforward consequence of (ih2) is

$$\overline{g}(\nu) - \underline{g}(\nu) = M(\overline{y}(\nu) - \underline{y}(\nu)) \geq 0. \tag{12.5}$$

For the intuition, this equality will also be proved directly. V' in (ih3) satisfies a similar equation, taking for $\underline{y}$ and $\underline{g}$ resp. $\widetilde{y}$ and $\widetilde{g} := E_\nu[V'X]$.

In the sequel we only consider the three cases with $\underline{f}(\nu d) < M$; the other Cases (ii) and (iv) are symmetry to resp. Case (i) and (iii).

First consider Case (i), where $\underline{f}(\nu u) = \overline{f}(\nu u) = M > \overline{f}(\nu d) \geq \underline{f}(\nu d)$; recall that $\nu u \in N_{\overline{\mathcal{T}}'}$, implying that $W^{\nu u} = W^{\nu u}_{\text{prev}}$. In this case, W^ν is given by (6.5) with branch-weight $0 < w < 1$ if (R1) applies, and $w = 0$ if (R2) does. By construction, $\overline{y}(\nu) \geq \alpha$, and hence $W^\nu \in \mathcal{W}^\nu$, cf. Lemma 3.2. From (6.5), it follows that

$$W^\nu_{\text{prev}} - W^\nu = \begin{cases} (1-w)W^{\nu u}_{\text{prev}} & \text{on } \{\omega' \in F(\nu) \,|\, \omega_t = \nu u\} \\ W^{\nu d}_{\text{prev}} - W^{\nu d} & \text{on } \{\omega' \in F(\nu) \,|\, \omega_t = \nu d\}, \end{cases} \tag{12.6}$$

and (ih2) for ν follows from $\nu u \in N_{\overline{\mathcal{T}}}$, $f(\nu u) = M$, and (ih2) for νd. To verify (12.5) directly, note that (12.6) yields

$$\overline{y}(\nu) - \underline{y}(\nu) = (1-w)p\underline{y}(\nu u) + (1-p)(\overline{y}(\nu d) - \underline{y}(\nu d)) \tag{12.7}$$

$$\overline{g}(\nu) - \underline{g}(\nu) = (1-w)p\underline{g}(\nu u) + (1-p)(\overline{g}(\nu d) - \underline{g}(\nu d)), \tag{12.8}$$

and use (12.5) for νu and substitute $\underline{g}(\nu u) = \overline{g}(\nu u) = M\overline{y}(\nu u) = M\underline{y}(\nu u)$.

The verification of (ih3) for ν is straightforward. To reduce the left-hand side in (12.7) by a factor $\gamma \in [0,1]$, choose V' as in (6.5) with w replaced by $\gamma w + 1 - \gamma$, and with $W^{\nu d}$ replaced by $V'' \in \mathcal{W}^{\nu d}$ satisfying the conditions in (ih3) for νd, taking for $\widetilde{y}$ the value $\gamma\underline{y}(\nu d) + (1-\gamma)\overline{y}(\nu d)$.

If (T1) applies, it is easily verified that (12.7) and (12.8) hold with $1 - w = 0$, reflecting that the up-branch is already cut *before* the loop. All claims (ih1-3) follow in a similar way. For (T2), the only extra complication is the existence of V in (6.6), but this follows immediately from (ih3) for νd.

So we proved (ih1-3), from which all claims of the lemma follow, except the claim that $|W|_\tau = 1_B$, or, equivalently, (12.1). To derive this, observe that by construction, for all $\nu \in N_{<\tau}$, $y(\nu) > \alpha$ and $f(\nu) < M$. Because $y(\nu) \neq \alpha$, (R1) and (T1) (and of course, by symmetry, (R3) and (T3)) never have been applied on the region $N_{<\tau}$. This implies that (6.5) (or its symmetric counterpart) applied in all nodes with branch-weight $w = 1$, except precisely for those nodes ν with $f(\nu) \leq M$, where $w = 0$ due to (R2,4) if $f(\nu) = M$, and due to (T2,4) if $f(\nu) > M$. It follows that $|W|_\tau = 1_B$.

□

Proof of Lemma 7.1

We will prove the slightly stronger claims that for any $(\widetilde{y}, \widetilde{g}, \widetilde{f}) \in \mathcal{R}^\nu$,

$$\text{for } \nu \in N_{<\tau} : \widetilde{g} \geq g(\nu) + M[\widetilde{y} - y(\nu)]^+ - M_{\text{next}}[\widetilde{y} - y(\nu)]^- \tag{12.9}$$

$$\text{for } \nu \in \text{Ex} : \widetilde{g} \geq g(\nu) + K_\nu[\widetilde{y} - y(\nu)]^+ - f(\nu)[\widetilde{y} - y(\nu)]^- \tag{12.10}$$

$$\text{for } \nu \in \mathcal{S} : \widetilde{g} \geq g(\nu) + M_\nu(\widetilde{y} - y(\nu)) \text{ and } \widetilde{y} \geq y(\nu) \tag{12.11}$$

with $M_\nu > f(\nu)$ the reduction rate in the loop that made ν belong to $\mathcal{S}$, $K_\nu > f(\nu)$ the reduction rate in the loop before ν became an M-node (for M, M_{next} see Section 6).

To see that this indeed implies optimality at τ, substitute $f(\nu)$ for K_ν in (12.10) and for M_ν in (12.11), which is justified because $f(\nu) < K_\nu, M_\nu$ (the strictness of the inequality is not needed here). For both inequalities this yields $\widetilde{g} \geq g(\nu) + f(\nu)(\widetilde{y} - y(\nu)) = f(\nu)\widetilde{y}$, so $\widetilde{f} = \widetilde{g}/\widetilde{y} \geq f(\nu)$ for all $\nu \subseteq \text{Ex} \cup \mathcal{S}$. Optimality at N_T is trivial, and hence optimality at τ follows.

We prove the three claims by backward recursion in time, using as induction hypothesis that they hold at the end of the previous loop, i.e., on $\Omega_{\overline{\tau}}$ we have

$$\text{for } \nu \in N_{<\overline{\tau}} : \widetilde{g} \geq \overline{g}(\nu) + M_{\text{prev}}[\widetilde{y} - \overline{y}(\nu)]^+ - M[\widetilde{y} - \overline{y}(\nu)]^- \tag{IH1}$$

$$\text{for } \nu \in \text{Ex}_{\text{prev}} : \widetilde{g} \geq \overline{g}(\nu) + K_\nu[\widetilde{y} - \overline{y}(\nu)]^+ - \overline{f}(\nu)[\widetilde{y} - \overline{y}(\nu)]^- \tag{IH2}$$

$$\text{for } \nu \in \mathcal{S}_{\text{prev}} : \widetilde{g} \geq \overline{g}(\nu) + M_\nu(\widetilde{y} - \overline{y}(\nu)]) \text{ and } \widetilde{y} \geq \overline{y}(\nu) \tag{IH3}$$

We first derive (12.10). For $\nu \in \text{Ex} \cap \text{Ex}_{\text{prev}}$, $f(\nu) = \overline{f}(\nu)$, and (12.10) is identical to the induction hypotheses (IH2). For $\nu \in \text{Ex} \setminus \text{Ex}_{\text{prev}} = N^M$, (IH1) must hold, with, by definition of K_ν, $K_\nu = M_{\text{prev}}$, and this also implies (12.10).

Next we consider (12.11). For $\nu \in \mathcal{S} \cap \mathcal{S}_{\text{prev}}$, this just coincides with the induction hypothesis (IH3). For $\nu \in \mathcal{S} \setminus \mathcal{S}_{\text{prev}}$, we have to derive (12.11) with $M_\nu = M$ from (IH1). Substituting (12.5) in (IH1), yields

$$\widetilde{g} \geq g(\nu) + M(\overline{y}(\nu) - y(\nu)) + M_{\text{prev}}[\widetilde{y} - \overline{y}(\nu)]^+ - M(\widetilde{y} - \overline{y}(\nu)]^-$$

and (12.11) follows from $M_{\text{prev}} > M$ and $y(\nu) = \alpha$.

So we proved (12.10) and (12.11), and hence the claims are verified for all nodes in N_τ. It remains to derive (12.9) for $\nu \in N_{<\tau}$. The crucial point is to show that no further reduction at rate M is possible in these nodes. Besides the global induction hypothesis (IH1), we also need the 'inner' induction hypothesis

$$\text{(12.9) holds for all } \nu' \in A \cap N_{t'}, \text{ for all } t' \geq t. \tag{ih}$$

Consider $\nu \in A \cap N_{t-1}$. To streamline the exposition, we use the following notation for splitting probability mass in ν according to the incoming branch. Let $V \in \mathcal{W}^\nu$ correspond to $(\widetilde{y}, \widetilde{g}, \widetilde{f}) \in \mathcal{R}^\nu$, and decompose $V = 1_u V + 1_d V$, where 1_d and 1_u are the indicator functions for paths through resp. νd and νu. We define

$$\begin{array}{lll} \delta_u & := E_\nu[1_u(V - W^\nu)] & \text{'the change in probability mass via the up-branch'} \\ \delta_d & := E_\nu[1_d(V - W^\nu)] & \text{'the change in probability mass via the down-branch'} \\ \delta & := \delta_d + \delta_u = \widetilde{y} - y(\nu) & \text{'the (total) change in probability mass. '} \end{array}$$

Rewriting (12.9), we have to prove that

$$\widetilde{g} \geq g(\nu) + M\delta^+ - M_{\text{next}}\delta^- \tag{12.12}$$

We consider the cases where (R2), (T2) or (T5) apply. Notice that (R1,3) and (T1,3) never apply to $\nu \notin \mathcal{S}$, and that (R4) and (T4) follow from symmetry.

If (R2) applies, then $\nu u \in N^M$, and $f(\nu u) = M$. From (12.10) for νu, with $K_{\nu u} = M_{\text{prev}} > M$, and (ih) for νd, it follows that

$$\widetilde{g} \geq g(\nu) + M\delta_u + M\delta_d^+ - M_{\text{next}}\delta_d^-. \tag{12.13}$$

Now (12.12) follows from the fact that $\delta_u \geq 0$, because $1_u W^\nu = 0$ as an effect of cutting the up-branch.

If (T2) applies, then $\nu u \in \text{Ex}_{\text{prev}}$, and $f(\nu u) > M$. Then also $1_u W^\nu = 0$, in fact already $1_u W^\nu_{\text{prev}} = 0$. From (12.10) for νu, with $K_{\nu u} > M_{\text{prev}} > M$, and (ih) for νd, (12.13) also holds true in this case, again with $\delta_u \geq 0$, and (12.12) follows.

Finally, if (T5) applies, (12.12) is immediate from (ih) for both child nodes νu and νd.

□

References

Artzner, Ph., F. Delbaen, J.-M. Eber, and D. Heath, 1999. Coherent measures of risk. *Mathematical Finance*, **9**, 203–228.

Artzner, Ph., F. Delbaen, J.-M. Eber, D. Heath, and H. Ku, 2007. Coherent multiperiod risk adjusted values and Bellman's principle. *Annals of Operations Research*, **152-1**, pp. 5-22.

Bank of International Settlements, 2006. Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework", *www.bis.org.*

Delbaen, F., 2006. The structure of m-stable sets and in particular of the set of risk neutral measures. In: Yor, M., Emery, M. (eds), In Memoriam Paul-André MeyerSeminaire de Probabilits XXXIX, pp. 215258. Springer, Berlin.

Dentcheva, D. and A. Ruszczyński, 2003. Optimization with stochastic dominance constraints. *SIAM Journal of Optimization*, **14-2**, pp. 548–566

Duffie, D. and J. Pan, 1997. An overview of Value at Risk. Journal of Derivatives 4, pp. 749.

Fábián, C.I., G. Mitra, and D. Roman, 2009. Processing second-order stochastic dominance models using cutting-plane representations. *Mathematical Programming*, Ser A. DOI 10.1007/s10107-009-0326-1.

Fishburn, P.C., 1964. *Decision and Value Theory.* John Wiley & Sons, New York.

Föllmer, H., and A. Schied, 2004. *Stochastic Finance. An introduction in discrete time* (2nd ed.). De Gruyter Studies in Mathematics, Vol. 27, de Gruyter, Berlin / New York.

Jorion, P., 1997. Value at risk: the new benchmark for controlling market risk. New York: McGraw-Hill.

Klein Haneveld, W.K., and M.H. van der Vlerk, 2006. Integrated chance constraints: reduced forms and an algorithm. *Computational Management Science*, **3**, pp. 245–269.

Künzi-Bay, A. and J. Mayer (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, **3**, 3-27.

Kupper, M. and W. Schachermayer, 2009. Representation results for law invariant time consistent functions, *Preprint.*

Kusuoka, S., 2001. On law invariant coherent risk measures. *Advances in Mathematical Economics*, **3**, 83–95.

Luedtke, J., 2008. New formulations for optimization under stochastic dominance constraints. *SIAM Journal on Optimization*, **19-3**, pp. 1433–1450.

McNeil, A.J., R. Frey and P. Embrechts, 2005. Quantitative Risk Management: Concepts, Techniques, Tools. Princeton University Press.

Morgan J.P., Inc., 1996. RiskMetrics$^{\mathrm{TM}}$: technical document (4th edn.). New York: Morgan.

Pflug, G.C., and W. Römisch, 2007. *Modeling, Measuring, and Managing Risk.* World Scientific.

Rockafellar, R. T. and S. Uryasev, 2002. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, **26**, 1443–1471.

Roorda, B., and J. M. Schumacher, 2007. Time consistency conditions for acceptability measures, with an application to tail value at risk, *Insurance: Mathematics and Economics*, **40**, pp. 209–230.

Roorda, B., and J. M. Schumacher, 2010. When can a risk measure be updated consistently? Submitted. An earlier version, with a different title, was available as Netspar Discussion Paper DP01/ 2009 006.

Rudolf, G. and A. Ruszczyński, 2008. Optimization problems with second order stochastic dominance constraints: duality, compact formulations, and cut generation methods. *SIAM Journal on Optimization*, **19**, pp. 1326-1343.

Szegö, G.P., 2002. No more VaR (this is not a typo), Editorial in *Journal of Banking and Finance*, **26**, 1247–1251

Tapiero, C.S., 2005. Value at risk and inventory control. *European Journal of Operational Research*, **163-3**, 769–775.