Ann Oper Res
DOI 10.1007/s10479-013-1391-7

Smoothing and parametric rules for stochastic mean-CVaR optimal execution strategy

Somayeh Moazeni · Thomas F. Coleman · Yuying Li

Abstract Computing optimal stochastic portfolio execution strategies under an appropriate risk consideration presents many computational challenges. Using Monte Carlo simulations, we investigate an approach based on smoothing and parametric rules to minimize mean and Conditional Value-at-Risk (CVaR) of the execution cost. The proposed approach reduces computational complexity by smoothing the nondifferentiability arising from the simulation discretization and by employing a parametric representation of a stochastic strategy. We further handle constraints using a smoothed exact penalty function. Using the downside risk as an example, we show that the proposed approach can be generalized to other risk measures. In addition, we computationally illustrate the effect of including risk on the stochastic optimal execution strategy.

Keywords Optimal execution · Computational stochastic programming · Dynamic programming · Penalty functions

1 Introduction

Institutional fund managers typically have large portfolios of hundreds of securities with individual positions constituting significant portions of market daily volumes. To achieve

S. Moazeni (✉)
Department of Operations Research and Financial Engineering, Princeton University, Sherrerd Hall, Charlton Street, Princeton, New Jersey 08544, USA
e-mail: somayeh@princeton.edu

T.F. Coleman
Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada
e-mail: tfcoleman@uwaterloo.ca

Y. Li
David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1, Canada
e-mail: yuying@uwaterloo.ca

Published online: 23 May 2013

investment performance objectives, they often need to rebalance portfolios in a short time horizon. Such trading implementation results in permanent and temporary price impact, potentially yielding unfavorable trading performance. Market impact and importance of the execution strategy to adapt to market information can be seen in the Flash Crash incidence in the US equity market on May 6, 2010 (Kirilenko et al. 2011).

Given a trading target and a trading horizon, the *optimal portfolio execution problem* provides an execution strategy to trade within the trading horizon, typically to minimize a weighted combination of the expected cost and risk in execution, see e.g., Almgren (2008). Since trading takes time and the permanent price impact of a trade can affect the future asset price, the optimal portfolio execution problem is fundamentally a stochastic dynamic programming problem, see e.g., Pérold (1988) and Bertsimas and Lo (1998). In a single asset case, Almgren and Lorenz (2007) provide an optimal adaptive strategy. Stochastic (adaptive) trading strategies can explicitly recognize market price change during the trading horizon. In addition it has been shown in Almgren and Lorenz (2007) that a significant improvement over static strategies can be achieved through stochastic trading strategies.

When no risk is considered, analytical solutions have been found for the stochastic dynamic programming problem which minimizes the expected execution cost under several price models, see e.g., Bertsimas and Lo (1998), Bertsimas et al. (1999) and Moazeni et al. (2013). Under a specific additive market price model with a deterministic market impact model and volatility, Huberman and Stanzl (2005) have obtained a closed-form solution for minimizing the mean and variance of the execution cost.

In addition to the expected execution cost, one is often interested in controlling the risk in execution, e.g., including minimizing variance of the execution cost as an objective. Unfortunately, under general price models, the mean-variance objective formulations for the optimal portfolio execution problem are not amenable to stochastic dynamic programming techniques; the dynamic programming equation may not exist. When this occurs, a time-consistent dynamic solution cannot be determined using a stochastic dynamic programming technique. Even when a dynamic programming equation exists, obtaining a closed-form solution in general may not be possible, particularly when constraints are included.

In Moazeni et al. (2013), a model is proposed which explicitly characterizes uncertain arrivals of other large trades by including jump processes to the market price dynamics. The proposed jump diffusion model includes two compound Poisson processes, with random jump amplitudes capturing uncertain permanent price impact of other large buy and sell trades respectively. Since the execution cost distribution is now asymmetric and may have fat tails, variance is no longer an appropriate risk measure. Alternative to variance, *Value-at-risk* (VaR) is a standard benchmark for a firm-wide measure of risk (Duffie and Pan 1997). For a given time horizon $\bar{t}$ and confidence level β, the value-at-risk of a portfolio is the loss in the portfolio's market value over the time horizon $\bar{t}$ that is exceeded with probability $1-\beta$. However, as a risk measure, VaR has recognized limitations. For example it lacks sub-additivity and convexity, see e.g., Artzner et al. (1997) and Artzner et al. (1999). The CVaR risk measure, also known as the *mean excess loss*, *mean shortfall* or *tail VaR*, is an attractive alternative to VaR. For a given time horizon $\bar{t}$ and confidence level β, CVaR is the conditional expectation of the loss above VaR for the time horizon $\bar{t}$ and the confidence level β. It has been shown that CVaR is a coherent risk measure and has many attractive properties including convexity, see e.g., Artzner et al. (1999). In addition, minimizing CVaR typically leads to a portfolio with a small VaR. The CVaR risk measure is widely used to measure and manage risk in various industries, such as finance, see e.g., Pflug and Romisch (2007) or Follmer and Schied (2011), electricity markets, see e.g., Yau et al. (2011) or Downward et al. (2012), and supply chain management (Goh and Meng 2009). Some of these works

adopt deterministic optimization approaches to deal with the modeled stochastic programming problem, see e.g., Shapiro et al. (2009) for a review on these methods. Using CVaR for the optimal portfolio execution problem seems appropriate as short term asset returns have fat tails and trading impact leads to price jumps (Moazeni et al. 2013).

In Shapiro (2008), dynamic programming equation is applied to dynamically coherent risk measures; however no computational result is provided. In general, when the objective function includes a risk measure such as CVaR, numerical methods are required to compute stochastic dynamic programming solutions. When a portfolio of risky assets are involved, solving a multi-stage optimal portfolio execution problem is computationally challenging, since computational complexity grows exponentially in the number of state variables. Thus computing a stochastic dynamic programming solution is often computationally intractable in practice; this is known as the *curse of dimensionality*. As discussed in Shapiro (2008), while two stage *linear* stochastic programming problems can be solved with a reasonable accuracy, computational complexity in solving multistage stochastic programming problems grows quickly with the increase of the number of stages. Many approximation algorithms in the literature have been considered to obtain approximations to stochastic programming solutions, see e.g., de Farias and Roy (2003) and Powell (2011). Solving a multi-stage stochastic programming problem is even more challenging when there are inequality constraints (Haugh and Lo 2001).

The goal of this paper is to propose a tractable computational approach to obtain an approximate stochastic dynamic programming solution for the optimal portfolio execution problem when mean and some risk measure of the execution cost are minimized. To achieve optimality at each time period k, a new stochastic strategy can be computed by considering optimality conditional on the information set $\mathcal{F}_k$ at time k. In particular, our approach relies on Monte Carlo simulations, where simulation price paths are generated by iid samples for the random variables in the decision time horizon. Compared to the backward iteration in the dynamic programming approach, methods based on forward simulation paths have attractive features. While backward dynamic programming approaches to multi-stage stochastic programming problems suffer the curse of dimensionality when applied to problems with high dimensional state spaces, the use of a forward simulation base approach for multi-stage multi-asset stochastic optimization problem does not incur exponential growth in computational complexity. Simulation based approximation solution approaches have been previously applied successfully in Longstaff and Schwartz (2001) to solve a stochastic dynamic programming for pricing an American option. Coleman et al. (2007) also use a similar method for the total risk minimization with a quadratic objective. In this case, they observe that this approach is capable of achieving relatively good accuracy comparing to the analytic solution. In Coleman et al. (2007), decision variables are approximated using cubic splines.

There are, however, additional computational challenges in solving the multi-stage multi-asset optimal portfolio execution problem based on simulations. Firstly, if a strategy is allowed to be an arbitrarily path dependent, the number of variables in the simulation optimization problem is proportional to the number of scenarios which is very large in general. Furthermore, unlike the single period simulation CVaR optimization problem, the multi-period simulation optimal portfolio execution problem is piecewise nonlinear rather than piecewise linear due to the presence of permanent market impact. This can be problematic since solving a general nonlinear programming problem is more difficult than solving a linear programming problem. Moreover, if constraints, e.g., bound constraints, are imposed, the number of corresponding constraints in the simulation optimization problem also becomes proportional to the number of simulations.

In this paper, we propose techniques to overcome these computational challenges for the simulation approach to multi-stage CVaR execution cost minimization. To reduce the number of variables, we first represent execution strategies using a parametric model with unknown parameters. Different parametric forms can be used. In this paper, we assume that an execution strategy depends linearly on the price and the trading accomplished thus far; this parametric form is motivated by the analytic formula for the minimum mean execution strategy derived in Moazeni et al. (2013). To alleviate the piecewise nonlinearity in the objective function arising from the simulation discretization to the CVaR measure, we apply the smoothing technique proposed in Alexander et al. (2006) for a single period CVaR optimization problem. The motivation behind the smoothing is the same as in the single period case: the piecewise nature in the simulation CVaR optimization problem arises from simulation discretization but the CVaR risk measure in the continuous model is in fact continuously differentiable. To handle constraints, we first apply the exact penalty function and then use smoothing to alleviate the piecewise nature of the exact penalty function. Indeed, our proposed smoothing method of the exact penalty function corresponds to applying a new penalty function which is piecewise quadratic but continuously differentiable. The new penalty function can be regarded as a combination of the quadratic and exact penalty functions.

Using the proposed parametric representation and smoothing method, we obtain a static nonlinear optimization problem with a potentially nonlinear objective function. We then use the trust region algorithm in Coleman and Li (1996) to solve this problem. The first and second derivatives of the objective function are computed using automatic differentiation, see e.g., Coleman and Verma (2000). We further note that our proposed computational approach is quite general and it can be applied to alternative risk measures other than CVaR.

The presentation is organized as follows. In Sect. 2, we present the mathematical formulation for the optimal portfolio execution problem. Our smoothing and parametric rules are explained in Sect. 3. In Sect. 3.3, we describe handling constraints using a smoothed exact penalty function. Our computational investigation is provided in Sect. 4. Concluding remarks are given in Sect. 5.

2 The optimal portfolio execution problem

We now present a mathematical formulation for the optimal portfolio execution problem. Without loss of generality, we assume that the decision maker wants to execute a sell order in a given time horizon. Mathematical analysis is similar for a buy execution.

Assume that a decision maker plans to liquidate his holdings in m assets during N periods in the time horizon T. Let $t_0 = 0 < t_1 < \cdots < t_N = T$, where $\tau \stackrel{\text{def}}{=} t_k - t_{k-1} = \frac{T}{N}$ for $k = 1, 2, \ldots, N$. The decision maker's position at time t_k is denoted by the m-vector $x_k = (x_{1k}, x_{2k}, \ldots, x_{mk})^T$. Here x_{ik} is the decision maker's holding in the ith asset at time t_k. We assume that the decision maker's initial position is $x_0 = \bar{S}$ and final position is $x_N = 0$, which guarantees complete liquidation by time T. The amount of trading in the kth period is given by the difference between positions at two consecutive times t_{k-1} and t_k, denoted by an m-vector n_k, where

$$n_k = x_{k-1} - x_k, \quad k = 1, 2, \ldots, N. \tag{1}$$

Negative n_{ik} implies that the ith asset is bought between t_{k-1} and t_k. We refer to a sequence $\{n_k\}_{k=1}^N$ satisfying $\sum_{k=1}^N n_k = \bar{S}$ as an *execution strategy*.

Let the m-vector P_k denote the unit market asset prices at time t_k. The deterministic initial market price, before the trade begins, is denoted by P_0. Assume that the permanent price impact of the decision maker's trade is a deterministic function $g(\cdot)$ of the trading rate:

$$P_k = \Phi(P_{k-1}, \xi_k) - \tau g\left(\frac{n_k}{\tau}\right), \quad k = 1, 2, \ldots, N-1. \tag{2}$$

The random variable $\Phi(P_{k-1}, \xi_k)$ denotes the stochastic model for the market price at time t_k when the decision maker does not trade in $(t_{k-1}, t_k]$, e.g., $\Phi(P_{k-1}, \xi_k) = P_{k-1} + \Sigma\xi_k$ with the covariance matrix $\Sigma\Sigma^T$ and ξ_k a multi-variate standard normal random variable, e.g., see Almgren and Chriss (2000/2001). Random price $\Phi(P_{k-1}, \xi_k)$ at time t_k can also be specified by other models, e.g., it can correspond to a jump diffusion model (Merton 1976) or a stochastic volatility model (Heston 1993). In Moazeni et al. (2013), we use a jump-diffusion model with two compound Poisson processes to represent the uncertain price impact from uncertain arrivals of other large buy and sell trades. Under this model assumption, the execution cost distribution is asymmetric and may have fat tails. Under such a model, using the CVaR risk measure is more appropriate in the optimal portfolio execution problem formulation.

The decision maker's trade n_k induces a temporary price impact on the execution price at period k. Similar to Almgren and Chriss (2000/2001), we assume here that the expected temporary price impact only depends on the trading rate; this temporary impact is represented by the function $h(\cdot)$. Hence, the m-vector unit execution price $\tilde{P}_k$ at time t_k is given by

$$\tilde{P}_k = P_{k-1} - h\left(\frac{n_k}{\tau}\right), \quad k = 1, 2, \ldots, N. \tag{3}$$

The total amount received by the decision maker at the end of the time horizon by executing the strategy $\{n_k\}_{k=1}^N$ is $\sum_{k=1}^N n_k^T \tilde{P}_k$. This random value depends on the specifications of the market price dynamics (2) and the execution price model (3). The difference between this quantity and the value of an ideal benchmark trade is the *execution cost* (Almgren 2008). The benchmark is commonly taken as the portfolio value at the initial price P_0. Hence, the *execution cost* associated with the execution strategy $\{n_k\}_{k=1}^N$ is defined as

$$X = P_0^T \bar{S} - \sum_{k=1}^N n_k^T \tilde{P}_k.$$

The optimal portfolio execution problem yields an execution strategy which minimizes the expected value and a risk measure in the execution cost.

Since trading takes time and the permanent price impact affects the future market price, optimal portfolio execution problem is a multi-stage stochastic programming problem. The solution to this multi-stage stochastic programming problem can potentially yield a solution which adapts to market price and the impact of other large trades.

While the main objective of the decision maker is to minimize the expected execution cost, he may be concerned with the execution risk, i.e., the uncertainty in the total amount that will be received from the trade implementation. When execution risk is considered, the

stochastic programming formulation for the optimal portfolio execution problem is:

$$\begin{aligned} \min_{\substack{n_1,\ldots,n_N \\ n_k:\mathcal{F}_k-\text{measurable},\ k=1,\ldots,N}} \quad & \mathbf{E}\left(P_0^T\bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k\right) + \mu\cdot\Psi\left(P_0^T\bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k\right) \\ \text{subject to} \quad & \sum_{k=1}^{N} n_k = \bar{S}, \end{aligned} \tag{4}$$

where $\Psi(\cdot)$ is a risk measure for the execution cost and $\mu \geq 0$ is a risk aversion parameter. Here $\mathcal{F}_k$ denotes the information set observable at time t_k.

Stochastic dynamic programming has been previously used to minimize the expected execution cost when the market price evolves according to a Brownian motion, and the permanent price impact of the decision maker's trade makes a discrete price change, see e.g., Bertsimas and Lo (1998) and Bertsimas et al. (1999). However, when a risk measure such as variance or CVaR is included in problem (4) with a positive risk aversion parameter, the multi-stage stochastic programming problem becomes significantly more complex. Moreover, when a dynamic programming equation cannot be found, a solution $\{n_k\}_{k=1}^N$ of the stochastic programming problem (4), computed at the initial time t_0, does not necessarily have the time consistency property. More precisely, n_k from problem (4) is not optimal at time t_k, i.e., it may not solve the following problem:

$$\begin{aligned} \min_{\substack{n_k,\ldots,n_N \\ n_j:\mathcal{F}_j-\text{measurable},\ j=k,\ldots,N}} \quad & \mathbf{E}\left(P_0^T\bar{S} - \sum_{i=1}^{N} n_i^T \tilde{P}_i \,\middle|\, \mathcal{F}_k\right) + \mu\cdot\Psi\left(P_0^T\bar{S} - \sum_{i=1}^{N} n_i^T \tilde{P}_i \,\middle|\, \mathcal{F}_k,\right) \\ \text{subject to} \quad & \sum_{k=1}^{N} n_k = \bar{S}. \end{aligned} \tag{5}$$

Here it is assumed that $n_1, \ldots, n_{k-1}$ are given.

Given that problems (4) and (5) yield different solutions at time t_k, $k \geq 2$, the decision maker has two different ways to implement an execution strategy through the multi-stage stochastic programming formulations. The first possibility is to compute the optimal strategy $\{n_k\}_{k=1}^N$ at the initial time based only on problem (4). Then at time t_k, the amount n_k, computed from (4), is implemented even though it may not be optimal from t_k perspective. Alternatively, to ensure conditional optimality at time t_k, the decision maker can ignore the already computed strategy for t_k from problem (4) and adopts the strategy for time t_k by solving a conditional stochastic programming problem (5) to determine trading amount for this period.

No matter which method the decision maker adopts for execution, she needs to solve one of the multi-stage stochastic programming problems (4) or (5). Computing solutions to these problems is a daunting task. In the remaining part of the paper, we focus on developing a tractable computational technique applicable to both problems, and we are not concerned with whether the strategy at time t_k should be computed from problem (4) or problem (5). Since our proposed computational method can be applied to both problems (4) and (5), without loss of generality, we present our proposed approach for problem (4).

Notice that problem (4) or (5) may have additional constraints, such as a no-buying requirement while selling. In this case, even when a dynamic programming equation exists, computational methods cannot easily handle constraints since the value function from the

dynamic programming under constraints becomes nondifferentiable, see e.g., Bertsimas and Lo (1998).

Different risk measures can be included in the objective function of problem (4). Typical choices of the risk measure $\Psi(\cdot)$ are variance, VaR, or CVaR. As discussed before, in this paper, we focus on CVaR risk measure since we believe that the short horizon return is far from a normal distribution and it is important to properly capture the tail risk. We note however that our proposed computational approach is applicable to other risk measures including variance and downside risk.

CVaR is frequently defined based on VaR. In the context of the optimal portfolio execution problem, let X denote the execution cost in the given time horizon. For a given confidence level β, VaR is the smallest cost over the time horizon that is exceeded with probability no greater than $1-\beta$, i.e., $\mathrm{VaR}_\beta(X) = \inf\{x \in \mathbb{R} : \Pr(X \le x) \ge \beta\}$, see e.g., Duffie and Pan (1997) and Alexander et al. (2006). Using VaR, CVaR can be defined as

$$\mathrm{CVaR}_\beta(X) = \mathbf{E}\big(X : X \ge \mathrm{VaR}_\beta(X)\big).$$

Without referencing to VaR, a more direct way of defining CVaR is:

$$\mathrm{CVaR}_\beta(X) = \min_\alpha \left(\alpha + \frac{1}{1-\beta}\mathbf{E}\big([X-\alpha]^+\big)\right), \tag{6}$$

where $[z]^+ = \max(z, 0)$, see, e.g., Rockafellar and Uryasev (2000). When the random cost X has a strictly increasing and continuous probability distribution function, these two definitions are equivalent. However the latter definition yields a coherent risk measure even when the distribution is discontinuous. In addition, formulation (6) directly leads to linear or nonlinear programming formulations under simulation discretizations. It then can be solved by available linear programming or nonlinear programming optimization techniques.

The mean-CVaR optimal portfolio execution problem with the risk aversion parameter $\mu \ge 0$ is then given as below

$$\begin{aligned} \min_{\substack{n_1, n_2, \ldots, n_N \\ n_k : \mathcal{F}_k - \text{measurable}}} \quad & \mathbf{E}\left(P_0^T \bar{S} - \sum_{k=1}^N n_k^T \tilde{P}_k\right) + \mu \cdot \mathrm{CVaR}_\beta\left(P_0^T \bar{S} - \sum_{k=1}^N n_k^T \tilde{P}_k\right) \\ \text{subject to} \quad & \sum_{k=1}^N n_k = \bar{S}. \end{aligned} \tag{7}$$

Using CVaR definition (6), formulation (7) is reduced to the following problem:

$$\begin{aligned} \min_{\substack{\alpha \in \mathbb{R}, n_1, n_2, \ldots, n_N \\ n_k : \mathcal{F}_k - \text{measurable}}} \quad & \mathbf{E}\left(P_0^T \bar{S} - \sum_{k=1}^N n_k^T \tilde{P}_k\right) + \mu\alpha + \frac{\mu}{1-\beta}\mathbf{E}\left(\left[P_0^T \bar{S} - \sum_{k=1}^N n_k^T \tilde{P}_k - \alpha\right]^+\right), \\ \text{subject to} \quad & \sum_{k=1}^N n_k = \bar{S}. \end{aligned} \tag{8}$$

Additional $n_k \ge 0$ constraints can also be incorporated in problem (8).

We note that, when the objective of the optimal portfolio execution problem is to minimize only the variance of the execution cost, i.e.,

$$\min_{n_1,\ldots,n_N} \operatorname{Var}\left(P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k\right) \quad \text{subject to } \sum_{k=1}^{N} n_k = \bar{S}, \tag{9}$$

the optimal execution strategy is the strategy of liquidating the entire holding in the first period:

$$n_1 = \bar{S}, \qquad n_k = 0, \quad k \geq 2. \tag{10}$$

This can be easily seen since the variance of the execution cost associated with this strategy equals zero. The CVaR of the execution cost associated with the execution strategy (10) is

$$\mathrm{CVaR}_\beta\left(P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k\right) = \frac{1}{\tau} \bar{S}^T h\left(\frac{\bar{S}}{\tau}\right).$$

We note that the strategy (10) for minimizing the variance is in general not the strategy for minimizing CVaR.

In the next section, we describe our proposed smoothing and parametric approach to obtain an approximate solution of problem (8) efficiently.

3 The proposed smoothing and parametric approach

Since the CVaR risk measure does not have an analytic expression in general, Monte Carlo (MC) simulation is typically applied to discretize the CVaR optimization problem. For the optimal portfolio execution problem, the discretized problem is more complex since the price path changes when the trading amount changes due to permanent price impact. Assume that the market price dynamics in the kth time period is given by $\Phi(P_{k-1}, \xi_k)$ where ξ_k is a random vector. We generate M random paths $\{\xi_1, \ldots, \xi_{N-1}\}$ and these sample values are fixed for simulation CVaR problems even when price paths change with the trading amount $\{n_k\}$. For any given $\{n_k\}_{k=1}^N$, let $\{P_k\}_{k=1}^N$ denote market price path corresponding to $\{\xi_1, \ldots, \xi_{N-1}\}$, we obtain a discretized stochastic optimization problem for problem (8):

$$\begin{aligned} \min_{\substack{n_1,n_2,\ldots,n_N,\alpha \\ n_k:\mathcal{F}_k-\text{measurable}}} \quad & \frac{1}{M}\sum_{j=1}^{M}\left(P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k^{(j)}\right) + \mu\alpha \\ & + \frac{\mu}{M(1-\beta)}\sum_{j=1}^{M}\left[P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k^{(j)} - \alpha\right]^+ \\ \text{subject to} \quad & \sum_{k=1}^{N} n_k = \bar{S}. \end{aligned} \tag{11}$$

The superscript (j) indicates the jth scenario. Note that for each k and j, $\tilde{P}_k^{(j)}$ is a $m \times 1$ vector, where m is the number of assets in the portfolio. The continuously differentiable nonlinear objective function in problem (8) now becomes a piecewise nonlinear objective

function. Each simulation corresponds to one nonlinear function piece; here the nonlinearity arises from the iterative dependence due to the permanent price impact. Using a standard technique of replacing the piecewise function $[\cdot]^+$ with a set of constraints, this piecewise nonlinear minimization problem can be formulated as a nonlinear programming problem with the number of nonlinear constraints proportional to the number of Monte Carlo simulations M. Solving such a large scale nonlinear programming problem is computationally expensive, as the number of scenarios M is typically very large. Therefore, as the first step, we use a smoothing method to avoid dealing with a very large number of constraints; this is described in Sect. 3.1.

3.1 Eliminating non-differentiability

To reduce computational complexity of problem (11), we use a smoothing technique, proposed by Alexander et al. (2006) for a single period CVaR optimization problem. The basic idea is to approximate the piecewise linear function $[z]^+$ with a continuously differentiable piecewise quadratic function $\rho_\epsilon(z)$ with a small resolution parameter ϵ:

$$\rho_\epsilon(z) = \begin{cases} z & \text{if } z > \epsilon \\ \frac{z^2}{4\epsilon} + \frac{1}{2}z + \frac{1}{4}\epsilon & \text{if } -\epsilon \le z \le \epsilon \\ 0 & \text{if } z < -\epsilon \end{cases} \tag{12}$$

Note that $\rho_\epsilon(z) \ge 0$ for every ϵ and z. Using (12), problem (11) is then reduced to the following continuously differentiable nonlinear minimization problem:

$$\begin{aligned} \min_{\substack{n_1, n_2, \ldots, n_N, \alpha \\ n_k : \mathcal{F}_k\text{-measurable}}} \quad & \frac{1}{M} \sum_{j=1}^{M} \left(P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k^{(j)} \right) + \mu\alpha \\ & + \frac{\mu}{M(1-\beta)} \sum_{j=1}^{M} \rho_\epsilon \left(P_0^T \bar{S} - \sum_{k=1}^{N} n_k^T \tilde{P}_k^{(j)} - \alpha \right) \\ \text{subject to} \quad & \sum_{k=1}^{N} n_k = \bar{S}. \end{aligned} \tag{13}$$

In problem (13), the objective function is actually continuously differentiable, since each simulation no longer introduces a nonlinear function piece. Therefore, there is no need to include an additional constraint for each simulation to avoid non-differentiability.

3.2 Using parametric trading rules

To obtain a stochastic execution strategy which adapts to the market price, one can let n_k freely depend on each price scenario, i.e.,

$$\begin{aligned} \min_{\substack{n_1^{(j)}, n_2^{(j)}, \ldots, n_N^{(j)}, \alpha \\ n_k : \mathcal{F}_k\text{-measurable}}} \quad & \left(P_0^T \bar{S} - \frac{1}{M} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} \right) + \mu\alpha \\ & + \frac{\mu}{M(1-\beta)} \sum_{j=1}^{M} \rho_\epsilon \left(P_0^T \bar{S} - \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} - \alpha \right) \end{aligned} \tag{14}$$

$$\text{subject to} \qquad \sum_{k=1}^{N} n_k^{(j)} = \bar{S}, \quad j = 1, 2, \ldots, M.$$

The number of decision variables in the nonlinear minimization problem (14) is of order $M \cdot N$, where M is the number of scenarios and N is the number of periods. Hence, solving problem (14) directly is computationally expensive as the number of scenarios M is typically large.

In addition we need to ensure that the execution strategy is *non-anticipatory*, n_k is $\mathcal{F}_k$-measurable. More precisely, execution strategy at stage k must only depend on the information available up to time t_k.

To resolve these two issues, we explicitly require that the execution strategy to have a parametric representation as below:

$$n_k = f_k(P_{k-1}, x_{k-1}), \quad k = 1, 2, \ldots, N-1. \tag{15}$$

Here f_k is a deterministic function of P_{k-1} and x_{k-1}, where P_{k-1} represents the market price at t_{k-1} and x_{k-1} quantifies the total number of shares to be sold. This explicitly restricts the strategy to be non-anticipatory.

Applying the decision rule (15) in problem (14), we arrive at:

$$\begin{aligned} \min_{\substack{n_1^{(j)}, n_2^{(j)}, \ldots, n_N^{(j)}, \alpha \\ n_k^{(j)} = f_k(P_{k-1}^{(j)}, x_{k-1}^{(j)})}} \quad & \frac{1}{M} \sum_{j=1}^{M} \left(P_0^T \bar{S} - \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} \right) + \mu\alpha \\ & + \frac{\mu}{M(1-\beta)} \sum_{j=1}^{M} \rho_\epsilon \left(P_0^T \bar{S} - \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} - \alpha \right) \\ \text{subject to} \quad & \sum_{k=1}^{N} n_k^{(j)} = \bar{S}, \quad j = 1, 2, \ldots, M. \end{aligned} \tag{16}$$

Assuming that the parametric function f_k depends on a small number of parameters, the number of unknown variables in the optimization problem (16) is then significantly reduced. The equality constraint can also be eliminated by an explicit variable substitution. Thus problem (16) can be represented as an unconstrained continuously differentiable nonlinear minimization problem with a total of $O(l \times (N-1))$ variables, where l denotes the number of parameters in the definition of f_k.

Now, we describe a specific linear trading rule used in our computational investigation for approximating the optimal execution strategy. This parametric representation is motivated by the explicit formula derived in Moazeni et al. (2013) for minimizing the expected execution cost under a multiplicative jump-diffusion model.

Specifically we assume the following linear parametric model for a stochastic optimal execution strategy:

$$\begin{aligned} n_k &= Y_k P_{k-1} + Z_k x_{k-1} + c_k, \quad k = 1, 2, \ldots, N-1, \\ n_N &= \bar{S} - \sum_{k=1}^{N-1} n_k, \end{aligned} \tag{17}$$

where Y_k and Z_k are $m \times m$ unknown matrix parameters, and c_k is an m unknown parameter vector. The m vector P_{k-1} represents the market price in the previous period and x_{k-1} is the m vector of shares remaining to be sold.

Indeed the optimal execution strategy for minimizing the expected execution cost has exactly this linear parametric representation (Moazeni et al. 2013). Thus the computed optimal execution strategy based on (17) when $\mu = 0$ and no constraint is included, attains minimum execution cost (i.e. no loss of optimality). When a positive risk aversion parameter is used, the parametric model assumption (17) may lead to a suboptimal solution. When $Y_k = 0$ and $Z_k = 0$, the strategy is a static execution strategy. One further may assume that $Y_1 = 0$ and $Z_1 = 0$ to reduce parameter redundancy since the strategy at $k = 1$ is deterministic and n_1 can be determined solely by c_1.

Using representation (17) for n_k, problem (16) is reduced to computing

$$c_1, \quad Y_2, \quad Z_2, \quad c_2, \quad \ldots, \quad Y_{N-1}, \quad Z_{N-1}, \quad c_{N-1} \quad \text{and} \quad \alpha$$

from the following problem:

$$\begin{aligned}
\min \quad & \left(P_0^T \bar{S} - \frac{1}{M} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} \right) + \mu\alpha \\
& + \frac{\mu}{M(1-\beta)} \sum_{j=1}^{M} \rho_\epsilon \left(P_0^T \bar{S} - \sum_{k=1}^{N} \left(n_k^{(j)}\right)^T \tilde{P}_k^{(j)} - \alpha \right) \\
\text{subject to} \quad & \sum_{k=1}^{N} n_k^{(j)} = \bar{S}, \quad j = 1, 2, \ldots, M, \\
& n_1^{(j)} = c_1, \\
& n_k^{(j)} = Y_k P_{k-1}^{(j)} + Z_k x_{k-1}^{(j)} + c_k, \quad k = 2, 3, \ldots, N-1.
\end{aligned} \tag{18}$$

After eliminating the decision variables $n_k^{(j)}$ in problem (18), the number of decision variables in this problem equals $(N-2)(2m^2 + m) + m + 1$ which does not depend on the number of simulations M.

3.3 Handling inequality constraints using penalty functions

In an optimal portfolio execution problem, one may want to impose additional inequality constraints, for example, no buying during a selling order execution. Handling inequality constraints in stochastic dynamic programming is in general challenging, see e.g., Grossman and Vila (1992) and Bertsimas and Lo (1998). This is because stochastic constraints make the value function nondifferentiable while applying dynamic programming equation. Using the simulation approach as in problem (16), the number of constraints becomes proportional to the number of simulations, since there exists a constraint corresponding to each future scenario. Thus computational complexity becomes prohibitive, particularly when the objective function is nonlinear due to permanent price impact.

Penalty functions are well established methods for handling constraints in nonlinear optimization, see e.g., Nocedal and Wright (2000). Quadratic penalty functions, exact penalty functions, and barrier functions are frequently used in practice. While barrier functions typically require a strictly feasible point to start with, the quadratic penalty function and exact

penalty function achieve feasibility in the optimization process. One attractive property of the exact penalty function, in comparison to the quadratic penalty function, is the existence of a finite penalty parameter (under suitable assumptions) using which a minimizer of the penalized optimization problem is a minimizer of the original optimization problem. If a quadratic penalty function is used, the penalized optimization yields a solution of the constrained optimization problem asymptotically as the penalty parameter converges to $+\infty$.

Consequently we prefer to use the exact penalty function. To illustrate this technique, assume that we want to include the following set of L constraints in optimization problem (7):

$$a_\ell(n_1, \ldots, n_N) \le 0, \quad \ell = 1, 2, \ldots, L.$$

Therefore, the simulation problem corresponding to (16) will have the following $M \cdot L$ constraints:

$$a_\ell\big(n_1^{(j)}, \ldots, n_N^{(j)}\big) \le 0, \quad j = 1, 2, \ldots, M,\ \ell = 1, 2, \ldots, L.$$

When the number of simulations M increases, the number of constraints increases accordingly. Consequently the computational cost for solving the corresponding nonlinear optimization problem can quickly become prohibitive. Using the exact penalty function $\max\{0, a_\ell(n_1^{(j)}, \ldots, n_N^{(j)})\}$ for the inequality $a_\ell(n_1^{(j)}, \ldots, n_N^{(j)}) \le 0$ and a large enough penalty parameter $\vartheta > 0$, we arrive at the following penalty optimization problem:

$$\begin{aligned}
\min_{\substack{\alpha\in\mathbb{R}, n_1^{(j)}, n_2^{(j)}, \ldots, n_N^{(j)} \\ n_k^{(j)} = f_k(P_{k-1}^{(j)}, x_{k-1}^{(j)}) \\ k=1,2,\ldots,N-1}} \quad & \left(P_0^T \bar{S} - \frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{N}\big(n_k^{(j)}\big)^T \tilde{P}_k^{(j)}\right) + \mu\alpha \\
& + \frac{\mu}{M(1-\beta)}\sum_{j=1}^{M}\rho_\epsilon\left(P_0^T \bar{S} - \sum_{k=1}^{N}\big(n_k^{(j)}\big)^T \tilde{P}_k^{(j)} - \alpha\right) \\
& + \vartheta \cdot \sum_{\ell=1}^{L}\sum_{j=1}^{M}\max\big\{0, a_\ell\big(n_1^{(j)}, \ldots, n_N^{(j)}\big)\big\} \\
\text{subject to} \quad & \sum_{k=1}^{N} n_k^{(j)} = \bar{S}, \quad j = 1, 2, \ldots, M.
\end{aligned} \tag{19}$$

Unfortunately the above penalty optimization problem is piecewise differentiable due to the use of the exact penalty function, with the number of function pieces proportional to the number of simulations. Once again, computational cost for solving the penalty optimization problem can quickly become prohibitive. Instead of resorting to the quadratic penalty, we choose to smooth the exact penalty function, given its similarity to nondifferentiability in the CVaR risk measure. Using smoothing based on the function $\rho_\epsilon(\cdot)$ defined in (12), we approximate the penalty optimization problem (19) by the following smooth unconstrained

minimization problem:

$$\begin{aligned}
\min_{\substack{\alpha\in\mathbb{R},n_1^{(j)},n_2^{(j)},\ldots,n_N^{(j)}\\ n_k^{(j)}=f_k(P_{k-1}^{(j)},x_{k-1}^{(j)})\\ k=1,2,\ldots,N-1}} \quad & \left(P_0^T\bar{S}-\frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{N}\big(n_k^{(j)}\big)^T\tilde{P}_k^{(j)}\right)+\mu\alpha \\
& +\frac{\mu}{M(1-\beta)}\sum_{j=1}^{M}\rho_\epsilon\left(P_0^T\bar{S}-\sum_{k=1}^{N}\big(n_k^{(j)}\big)^T\tilde{P}_k^{(j)}-\alpha\right) \\
& +\vartheta\cdot\sum_{\ell=1}^{L}\sum_{j=1}^{M}\rho_\epsilon\big(a_\ell\big(n_1^{(j)},\ldots,n_N^{(j)}\big)\big) \\
\text{subject to} \quad & \sum_{k=1}^{N}n_k^{(j)}=\bar{S},\quad j=1,2,\ldots,M.
\end{aligned} \tag{20}$$

Here we can regard the smoothed function $\rho_\epsilon(\cdot)$ as a new penalty function; it is a hybrid of the quadratic penalty function and the exact penalty function. Indeed this new penalty function can be regarded as an exact penalty function with a resolution determined by the parameter ϵ. This parameter ϵ can be different from that in the smoothed function for CVaR and it can vary with the constraints. We are currently investigating theoretical properties of this new penalty function.

The objective function of problem (20) is continuously differentiable but quite nonlinear due to smoothing of piecewise functions as well as the existence of the permanent price impact. Optimization methods for minimizing a continuously differentiable objective function typically require derivative calculations to achieve a good computational performance. In our subsequent computational investigation, we use the trust region method in Coleman and Li (1996) with the derivative evaluations using automatic differentiation; for further discussion on automatic differentiation we refer an interested reader to Griewank and Corliss (1991), Coleman and Verma (2000), Nocedal and Wright (2000) and references therein.

4 Computational results

This section presents several computational examples to illustrate feasibility and efficacy of our proposed smoothing and parametric representation approach for approximating optimal stochastic execution strategies. In addition we assess performance of the computed stochastic execution strategy. The objective of our computational investigation is to demonstrate

- Accuracy of the computed execution strategies by comparing them to the strategies from analytic formulae when they exist;
- Capability of the proposed technique to handle inequality constraints;
- Applicability of the technique to alternative risk measures. This also allows us to study the effect of the choice of a risk measure on the optimal execution strategy.

Specifically, we approximate the optimal execution strategy by solving problem (18). We assume that the market price follows a jump diffusion process

$$\Phi(P_{k-1},\xi_k,\mathcal{J}_k)=\mathbf{Diag}(P_{k-1})\big(e_m+\tau^{1/2}\Sigma\xi_k+\mathcal{J}_k\big),$$

where e_m is the m-vector of all ones, Σ is the $m \times l$ volatility matrix, ξ_k is the l-vector of independent standard normals, and $\mathcal{J}_k$ is the m-vector of random jumps which mainly captures permanent price impacts of other concurrent trades. As in Moazeni et al. (2013), $\mathcal{J}_k$ is defined as below:

$$\left(\sum_{\ell=1}^{Y_{t_k}^{(1)} - Y_{t_{k-1}}^{(1)}} \left(\chi_\ell^{(1)}(k) - 1\right) - \sum_{\ell=1}^{X_{t_k}^{(1)} - X_{t_{k-1}}^{(1)}} \left(\pi_\ell^{(1)}(k) - 1\right), \ldots, \right.$$
$$\left. \sum_{\ell=1}^{Y_{t_k}^{(m)} - Y_{t_{k-1}}^{(m)}} \left(\chi_\ell^{(m)}(k) - 1\right) - \sum_{\ell=1}^{X_{t_k}^{(m)} - X_{t_{k-1}}^{(m)}} \left(\pi_\ell^{(m)}(k) - 1\right) \right)^T,$$

where $\{X_t^{(i)}\}$ and $\{Y_t^{(i)}\}$, $i = 1, 2, \ldots, m$, are two independent Poisson processes with constant arrival rates $\lambda_x^{(i)}$ and $\lambda_y^{(i)}$, respectively. We assume that the jump amplitudes are log-normally distributed and identically distributed over period, i.e., $\log \pi_\ell^{(i)}(k)$ and $\log \chi_\ell^{(i)}(k)$ have normal distributions for all i and k, with means μ_x and μ_y, and standard deviations σ_x and σ_y, respectively. We further assume that the arrival rates $\lambda_x^{(i)}$ and $\lambda_y^{(i)}$ of different assets in the portfolio are equal to λ_x and λ_y, respectively.

In our computation, price impacts are assumed to be proportional to the trading rate, as linear price impact functions have been frequently used in the literature, see e.g., Bertsimas and Lo (1998), Bertsimas et al. (1999), Almgren and Chriss (2000/2001), Huberman and Stanzl (2004), Moazeni et al. (2010) and Moazeni et al. (2013). Linear price impact functions are defined by the temporary impact matrix H and the permanent impact matrix G, as below:

$$g(v) = Gv, \qquad h(v) = Hv, \tag{21}$$

where $v = \frac{n}{\tau}$ is the trading rate. Here impact matrices H and G are expected price depressions caused by trading assets at a unit rate.

In summary the execution price model and market price dynamics are as follows:

$$\tilde{P}_k = P_{k-1} - \frac{H}{\tau} n_k, \tag{22}$$

$$P_k = \mathbf{Diag}(P_{k-1})\left(e_m + \tau\alpha + \tau^{1/2}\Sigma\xi_k + \mathcal{J}_k\right) - Gn_k. \tag{23}$$

Unless otherwise stated, our computation generates $M = 12{,}000$ sample paths of random variables $\{(\xi_1, \mathcal{J}_1), \ldots, (\xi_{N-1}, \mathcal{J}_{N-1})\}$. We use automatic differentiation in ADMAT: Automatic Differentiation Toolbox (Coleman and Verma 2000) to compute gradients. The Hessian is then computed using the finite difference method.

The optimal execution strategy in general differs with the choice of the risk measure. For example, it can be shown that the variance of the execution cost, under our assumed model, does not depend on the impact matrices. However, CVaR of the execution cost depends on the impact matrix.

The proposed computational method can be applied to other downside risk measures such as *Semi-standard deviation*, see, e.g., Fabozzi et al. (2007, p. 59):

$$\Psi\left(P_0^T\bar{S}-\sum_{k=1}^{N}n_k^T\tilde{P}_k\right)\stackrel{\text{def}}{=}\mathbf{E}\left(\left[\bar{S}^TP_0-\sum_{k=1}^{N}n_k^T\tilde{P}_k\right]^+\right)$$
$$\approx\frac{1}{M}\sum_{j=1}^{M}\rho_\epsilon\left(\bar{S}^TP_0-\sum_{k=1}^{N}\left(n_k^{(j)}\right)^T\tilde{P}_k^{(j)}\right). \tag{24}$$

To assess accuracy and effect of risk measures, we compare the following execution strategies:

Strategy$_M$: strategy which minimizes the expected execution cost, i.e., $\mu=0$ in problem (18).
Strategy$_C$: strategy which minimizes CVaR$_{95\,\%}$, without considering the expected execution cost.
Strategy$_S$: strategy which minimizes the variance (or standard deviation) of the execution cost.
Strategy$_N$: the naive strategy, $n_k=\frac{\bar{S}}{N}$, $k=1,2,\ldots,N$.
Strategy$_D$: strategy which minimizes the semi-standard deviation risk measure (see (24)).

4.1 Accuracy of the computational approach

To illustrate accuracy of the proposed computational approach, we compare the computed execution strategy from (18) and its performance with the exact optimal execution strategy for minimizing the expected execution cost only and for minimizing the variance of the execution cost only, since an analytic solution exists for both cases. Strategy$_S$ is obtained by solving problem (18) with the objective function replaced by the variance of the execution cost:

$$\mathrm{Var}\left(P_0^T\bar{S}-\sum_{k=1}^{N}n_k^T\tilde{P}_k\right)\approx\frac{1}{M}\sum_{j=1}^{M}\left(\left(P_0^T\bar{S}-\sum_{k=1}^{N}\left(n_k^{(j)}\right)^T\tilde{P}_k^{(j)}\right)\right.$$
$$\left.-\left(P_0^T\bar{S}-\frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{N}\left(n_k^{(j)}\right)^T\tilde{P}_k^{(j)}\right)\right)^2.$$

Let Strategy$_M^*$ and Strategy$_S^*$ denote the exact strategies from the analytic formulae to minimize mean and variance of the execution cost, respectively.

We consider an execution problem for a portfolio of three assets with the parameter setting described in Table 1.

Table 1 Parameter values used in our simulations

Parameters	Values
Trading horizon	$T=5$ days
Number of periods	$N=5$
Interval length	$\tau=T/N=1$ day
Initial portfolio price	$P_0=50e_3$ \$/share
Initial holdings	$\bar{S}=10^6e_3$ shares
CVaR confidence level	$\beta=0.95$

Table 2 Mean, standard deviation, and $\text{CVaR}_{95\ \%}$ of the execution cost corresponding to each strategy

	Mean	Standard deviation	CVaR
Strategy_S	1.0319544883×10^6	0.0338536247	1.0319549344×10^6
Strategy_S^*	1.0319545000×10^6	0	1.0319545000×10^6
Strategy_M	1.9618159824×10^5	3.2645688653×10^5	8.6482124208×10^5
Strategy_M^*	1.9618206384×10^5	3.2634725096×10^5	8.6451163525×10^5

We assume that the daily asset return covariance matrix is

$$C = \begin{pmatrix} 0.324625 & 0.022983 & 0.420395 \\ 0.022983 & 0.049937 & 0.019247 \\ 0.420395 & 0.019247 & 0.764097 \end{pmatrix} \times 1\ \%. \tag{25}$$

We further assume:

$$H = 0.5 \times 10^{-4} \cdot C, \qquad G = 0.5 \times 10^{-5} \cdot C, \qquad \Sigma = (0.001 \cdot C)^{1/2}.$$

We let arrival rates and jump amplitudes be identical for the three assets:

$$\lambda_x = 0.5, \qquad \mu_x = 10^{-4}, \qquad \sigma_x = 10^{-3}, \qquad \lambda_y = 2, \qquad \mu_y = 10^{-4}, \qquad \sigma_y = 10^{-3}.$$

When only variance of the execution cost is minimized, the exact optimal execution strategy is given in (10) for which the optimal objective value equals zero. Furthermore, when only the expected execution cost is minimized, an analytical formula for the optimal execution strategy obtained from the stochastic dynamic programming is provided in Moazeni et al. (2013). We use these two cases as benchmarks to illustrate the accuracy of the proposed technique.

Table 2 compares the expected execution cost, standard deviation, and CVaR of the computed execution strategies with those of the optimal execution strategies using explicit formulae. Comparing Strategy_M with Strategy_M^*, we observe approximately five significant digits of accuracy in the expected execution cost and three significant digits in standard deviation. The variance of the Strategy_S is about 10^{-3} compared to zero for Strategy_S^*; however the expected execution cost agrees in about 6 significant digits.

To examine the difference in the execution strategy, we quantify the percentage difference between the exact optimal execution strategy and the computed execution strategy using the following measure:

$$\varepsilon(i,k) \stackrel{\text{def}}{=} \left(100/\|\bar{S}\|_\infty\right) \times \max_{1 \le j \le M} \left| n_*^{(k)}(i,j) - \hat{n}^{(k)}(i,j) \right|, \quad i = 1, \ldots, m,\ k = 1, \ldots, N,$$

where, for asset i in simulation j, $n_*^{(k)}(i,j)$ and $\hat{n}^{(k)}(i,j)$ are the analytical solution and the computed solution at period k, respectively. Values of $\varepsilon(i,k)$ are reported in Table 3 for $M = 12,000$ simulations. The results indicate that the computed solutions are relatively close to the exact ones, and the maximum difference between them is at most 1.5 % which most likely comes from computational errors.

For minimizing CVaR, there is no analytic solution. Table 4 presents mean and $\text{CVaR}_{95\ \%}$ of the execution cost corresponding to the computed solution of problem (18) for different choices of μ. Even though we cannot explicitly assess the accuracy in this case, we do

Table 3 Comparisons to benchmark strategies Strategy$^*_M$ and Strategy$^*_S$

Percentage difference $\varepsilon(i,k)$ corresponding to Strategy$_M$					
Asset	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
1	0.02150	1.49049	0.360879	−0.00558	−0.94410
2	−0.21806	−0.79442	1.43315	0.05718	0.25880
3	−0.01826	−0.74518	−0.00558	0.34084	0.69127

Percentage difference $\varepsilon(i,k)$ corresponding to Strategy$_S$					
Asset	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
1	1.91090×10^{-6}	2.82690×10^{-4}	3.34015×10^{-4}	1.66590×10^{-4}	3.40619×10^{-4}
2	2.35354×10^{-6}	3.19347×10^{-4}	1.20114×10^{-4}	2.76442×10^{-4}	2.74782×10^{-4}
3	-2.51376×10^{-6}	2.55169×10^{-4}	2.03557×10^{-4}	1.72027×10^{-4}	2.72080×10^{-4}

Table 4 Mean and CVaR$_{95\ \%}$ of the execution cost in dollar per share

μ	CVaR (95 %)	Expected execution cost
0	0.86482	0.19620
1	0.77781	0.20439
10	0.77485	0.20505
$+\infty$	0.77464	0.20512

observe that, for the computed strategy, the expected execution cost increases while the CVaR$_{95\ \%}$ decreases, when the risk aversion parameter μ increases.

Improvements in the objective function value by the optimization solver over iterations are presented in Fig. 1. These plots demonstrate that for the portfolio example of three assets considered, a relatively small number of iterations (10 to 15) in the optimization solver is enough to obtain a near optimal solution. The computational time for each iteration varies significantly according to the objective function. The investigation of the computational time of the approach and its improvement remains for future work.

4.2 Handling constraints

We now illustrate effectiveness of the smoothed penalty function to handle constraints. We also investigate the effect of the constraint $n_k \geq 0$ on the computed optimal execution strategy and the corresponding objective function value. We consider liquidation of $\bar{S} = 10^6$ shares of a single asset whose initial market price is $P_0 = 50$ dollar per share. Permanent and temporary price impact values are assumed to be $G = 2.5 \times 10^{-7}$ and $H = 2.5 \times 10^{-6}$, respectively, and $\Sigma = 0.009$. Jump parameters are as follows:

$$\lambda_x = 3, \qquad \mu_x = 9.5 \times 10^{-3}, \qquad \sigma_x = 10^{-2}, \qquad \lambda_y = 0.5,$$
$$\mu_y = 6.9 \times 10^{-4}, \qquad \sigma_y = 3.2 \times 10^{-2}.$$

CVaR and mean of the execution costs corresponding to the optimal execution strategies with and without the constraint $n_k \geq 0$ are presented in Table 5.

Figure 2 depicts the optimal execution strategy for minimizing mean and CVaR of the execution cost with the risk aversion parameter $\mu = 100$ in the presence of the non-negativity

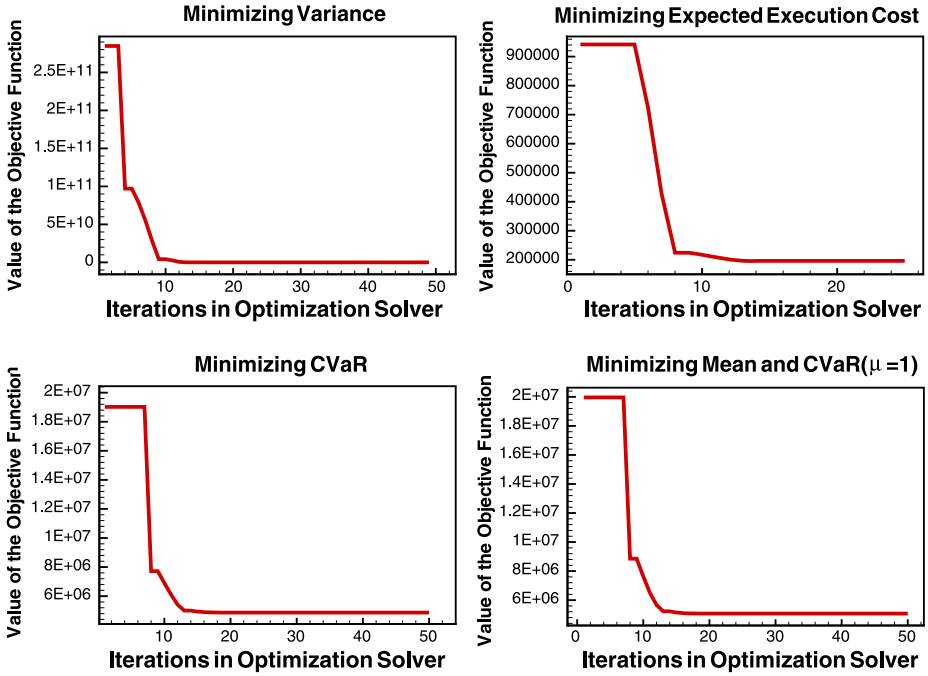


Fig. 1 Progress of the optimization solver over iterations

Table 5 Effect of constraint $n_k \geq 0$: cost and risk values in dollars per share

Execution strategies	$\text{CVaR}_{95\,\%}$	Expected execution cost
n_k unconstrained ($\mu = 100$)	3.13030	1.42585
$n_k \geq 0$ ($\mu = 100$)	5.38058	2.50723
n_k unconstrained ($\mu = 0$)	3.28077	1.41696
$n_k \geq 0$ ($\mu = 0$)	5.38061	2.50704

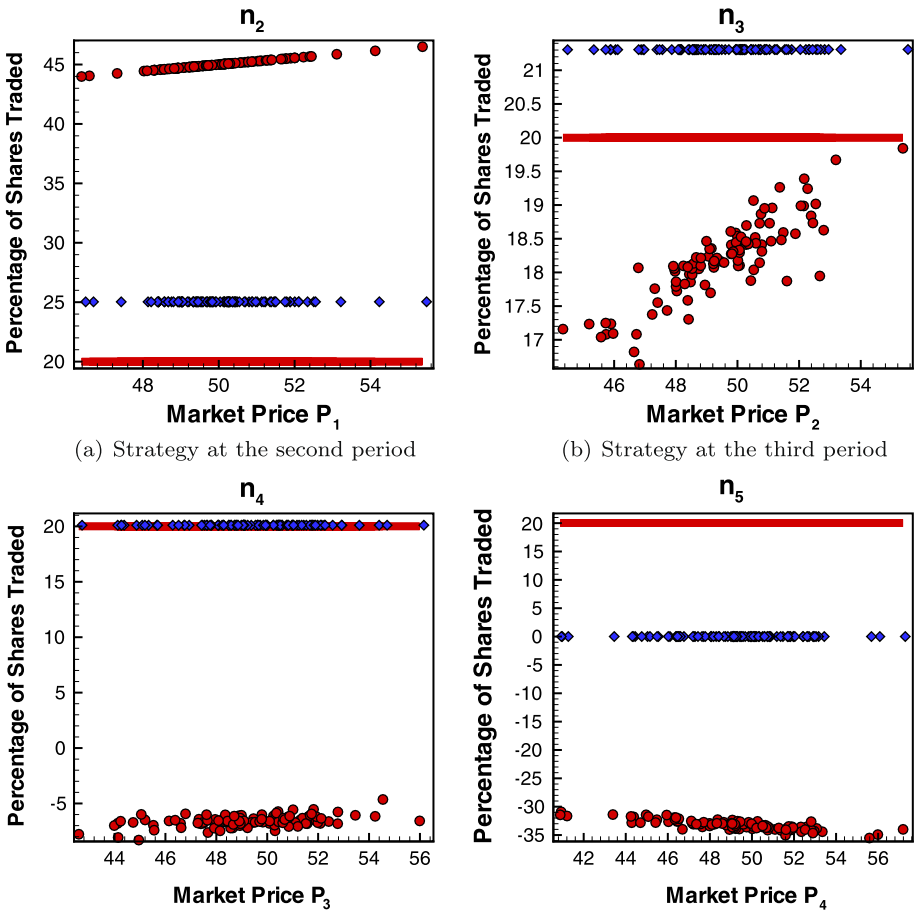
constraints $n_k \geq 0$. These plots show that the computed optimal execution strategy using the penalty parameter $\vartheta = 10^4$ indeed satisfies $n_k \geq 0$. In particular, while the execution strategy when n_k is not bound constrained suggests to sell more in the first period and buy in the last periods ($k = 4, 5$); the execution strategy computed under $n_k \geq 0$ is more conservative and the strategy does not seem to vary with the asset price significantly.

4.3 Applicability to other risk measures

Here we illustrate application of the proposed approach for the semi-standard deviation risk measure when trading a single asset. The setting is as in Sect. 4.2.

Figure 3 demonstrates that Strategy_D is very similar to Strategy_S^*. Furthermore, Strategy_M is more aggressive comparing to Strategy_C, i.e., it suggests to trade more in the first periods and buy over the last periods.

It is worth mentioning that the results provided in this section depend on our assumed linear parametric representation in (17). If we choose other representations, the configuration of the computed optimal execution strategies might differ. We leave investigating properties



(a) Strategy at the second period

(b) Strategy at the third period

(c) Strategy at the fourth period

(d) Strategy at the fifth period

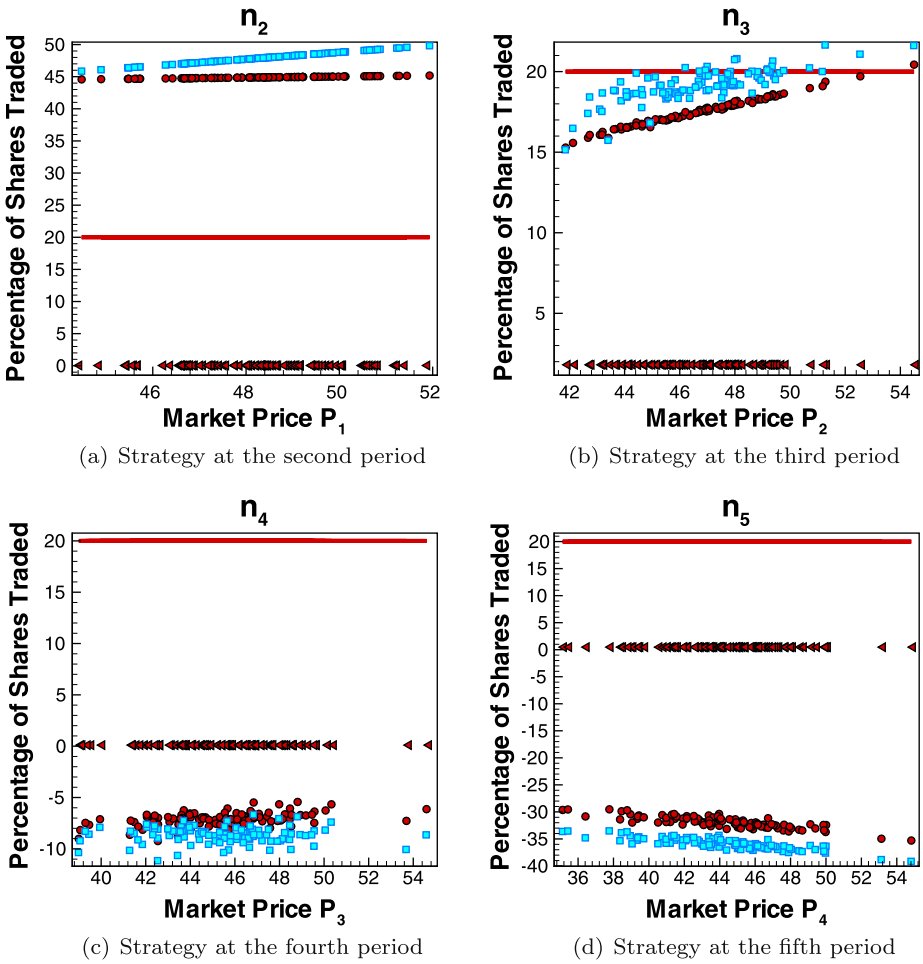
$$\lambda_x = 3, \quad \mu_x = 9.5 \times 10^{-3}, \quad \sigma_x = 10^{-2}, \quad \lambda_y = 0.5, \quad \mu_y = 6.9 \times 10^{-4}, \quad \sigma_y = 3.2 \times 10^{-2}$$
$$\bar{S} = 10^6, \quad P_0 = 50, \quad H = 2.5 \times 10^{-6}, \quad G = 2.5 \times 10^{-7}, \quad \Sigma = 0.009.$$

Fig. 2 The 100 realizations of the computed optimal execution strategies as functions of the market price for a single asset trading with and without non-negativity constraints. The *circles* show the execution strategy when buying is allowed, and the *diamonds* are the strategy when buying is prohibited. The *line* in each graph indicates Strategy$_N$. Strategies have been computed using the penalty parameter $\vartheta = 10^4$ and the risk aversion parameter $\mu = 100$. In the first period when buying is allowed, $n_1^* = 76.64867$ % of the initial holding and when buying is prohibited, $n_1^* = 75.28936$ % of the initial holding

of the solutions under different parametric representations for the execution strategy for future research.

5 Concluding remarks

Solving multi-stage stochastic programming problem is a daunting task, particularly when there are constraints. Under both temporary and permanent price impact, the objective func-



(a) Strategy at the second period

(b) Strategy at the third period

(c) Strategy at the fourth period

(d) Strategy at the fifth period

$\lambda_x = 3, \quad \mu_x = 9.5 \times 10^{-3}, \quad \sigma_x = 10^{-2}, \quad \lambda_y = 0.5, \quad \mu_y = 6.9 \times 10^{-4}, \quad \sigma_y = 3.2 \times 10^{-2},$
$\bar{S} = 10^6, \quad P_0 = 50, \quad H = 2.5 \times 10^{-6}, \quad G = 2.5 \times 10^{-7}, \quad \Sigma = 0.009.$

Fig. 3 The 100 realizations of the optimal execution strategies Strategy$_M$ (*squares*), Strategy$_C$ (*circles*), and Strategy$_D$ (*triangles*) as functions of market price for a single asset trading when no constraint is imposed. The *line* in each graph indicates the naive strategy Strategy$_N$. In the first period, Strategy$_N$ suggests to sell $n_1^* = 20.00$ %, Strategy$_M$ suggests to sell $n_1^* = 77.44418$ %, and Strategy$_C$ suggests to sell $n_1^* = 76.63369$ %, and Strategy$_D$ suggests to sell $n_1^* = 97.57436$ % of the initial holding

tion of the optimal portfolio execution problem can be quite nonlinear when a risk measure for the execution cost is included.

In this paper, we propose a tractable computational approach to compute an optimal portfolio execution strategy. The approach relies on Monte Carlo simulations, a smoothing technique, and parametric rules for the optimal strategy. The smoothing technique alleviates the nondifferentiability arising from the CVaR risk measure for each simulation. The parametric rule allows a strategy to be stochastic and reduces the number of optimization variables. In particular, a linear parametric representation permits the exact representation of the ex-

ecution strategy for minimizing the expected cost. The approach then yields a stochastic execution strategy which depends on the price and holdings at trading time. The computational complexity of the resulting method does not depend on the number of simulations.

While we focus on CVaR risk measure, the proposed computational method is applicable to different risk measures, e.g., downside risk as well as variance. In addition, a smoothed exact penalty function is applied to handle stochastic constraints.

Since the CVaR risk measure has become a widely used risk measure in many industries beyond finance, for example in energy market or supply chain management, it will be useful to investigate the effectiveness of the proposed computational stochastic programming method for other applications or embedded in alternative risk management methodologies, e.g., Wu and Olson (2010). Performance of the approach, however, relies on an appropriate choice of the parametric rule, or policy function approximation as explained in Powell (2011). Furthermore, applying some tools such as structure-exploiting automatic differentiation modes and parallel computing can improve the computational efficiency of the methodology.

References

Alexander, S., Coleman, T. F., & Li, Y. (2006). Minimizing VaR and CVaR for a portfolio of derivatives. *Journal of Banking & Finance*, *30*(2), 583–605.

Almgren, R., & Chriss, N. (2000/2001). Optimal execution of portfolio transactions. *The Journal of Risk*, *3*(2), 5–39.

Almgren, R., & Lorenz, J. (2007). Adaptive arrival price. In B. R. Bruce (Ed.), *Algorithmic trading III: precision, control, execution* (pp. 59–66). Institutional Investor Journals.

Almgren, R. F. (2008). Execution costs. In *Encyclopedia of quantitative finance* (pp. 1–5).

Artzner, P., Delbaen, F., Eber, J., & Heath, D. (1997). Thinking coherently. *Risk*, *10*, 68–71.

Artzner, P., Delbaen, F., Eber, J., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, *9*, 203–228.

Bertsimas, D., & Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*, *1*(1), 1–50.

Bertsimas, D., Lo, A. W., & Hummel, P. (1999). Optimal control of execution costs for portfolios. *Computing in Science & Engineering*, *1*(6), 40–53.

Coleman, T. F., & Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, *6*(2), 418–445.

Coleman, T. F., & Verma, A. (2000). ADMIT-1: automatic differentiation and MATLAB interface toolbox. In *ACM transactions on mathematical software* (pp. 150–175).

Coleman, T. F., Li, Y., & Patron, C. (2007). Total risk minimization. In J. R. Birge & V. Linetsky (Eds.), *Handbook of financial engineering* (pp. 593–635). Amsterdam: Elsevier.

de Farias, D. P., & Roy, B. V. (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, *51*(6), 850–865.

Downward, A., Young, D., & Zakeri, G. (2012, submitted). Electricity contracting and policy choices under risk-aversion. *Operations Research*, 1–43.

Duffie, D., & Pan, J. (1997). An overview of value at risk. *The Journal of Derivatives*, *4*(3), 7–49.

Fabozzi, F. J., Kolm, P. N., Pachamanova, D., & Focardi, S. M. (2007). *Robust portfolio optimization and management*. New York: Wiley.

Follmer, H., & Schied, A. (2011). *Stochastic finance: an introduction in discrete time* (3rd ed.). Berlin: de Gruyter.

Goh, M., & Meng, F. (2009). A stochastic model for supply chain risk management using conditional value at risk. In T. Wu & J. V. Blackhurst (Eds.), *Managing supply chain risk and vulnerability: tools and methods for supply chain decision makers* (pp. 141–157). Berlin: Springer.

Griewank, A., & Corliss, G. (1991) *SIAM proceedings series:* In *EDS, automatic differentiation of algorithm: theory, implementation and applications*.

Grossman, S. J., & Vila, J. (1992). Optimal dynamic trading with leverage constraints. *Journal of Financial and Quantitative Analysis*, *27*(2), 151–168.

Haugh, M. B., & Lo, A. W. (2001). Computational challenges of financial engineering. *Computing in Science & Engineering*, *3*, 54–59.

Heston, S. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, *6*(2), 327–343.

Huberman, G., & Stanzl, W. (2004). Price manipulation and quasi-arbitrage. *Econometrica*, *72*(4), 1247–1275.

Huberman, G., & Stanzl, W. (2005). Optimal liquidity trading. *Review of Finance*, *9*(2), 165–200.

Kirilenko, A., Samadi, M., Kyle, A. S., & Tuzun, T. (2011, working paper). The flash crash: The impact of high frequency trading on an electronic market. 1–63.

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, *14*(1), 113–147.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, *3*(1–2), 125–144.

Moazeni, S., Coleman, T. F., & Li, Y. (2010). Optimal portfolio execution strategies and sensitivity to price impact parameters. *SIAM Journal on Optimization*, *20*(3), 1620–1654.

Moazeni, S., Coleman, T. F., & Li, Y. (2013). Optimal execution under jump models for uncertain price impacts. *Journal of Computational Finance*, *16*(4), 1–44.

Moazeni, S., Coleman, T. F., & Li, Y. (2013). Regularized robust optimization: The optimal portfolio execution case. *Computational Optimization and Applications*, *55*(2), 341–377.

Nocedal, J., & Wright, S. J. (2000). *Numerical optimization*. Berlin: Springer.

Pérold, A. (1988). The implementation shortfall: paper versus reality. *The Journal of Portfolio Management*, *14*, 4–9.

Pflug, G. C., & Romisch, W. (2007). *Modeling, measuring and managing risk*. Singapore: World Scientific.

Powell, W. B. (2011). *Approximate dynamic programming: solving the curses of dimensionality* (2nd ed.). New York: Wiley.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, *2*(3), 21–41.

Shapiro, A. (2008). Stochastic programming approach to optimization under uncertainty. *Mathematical Programming*, *112*(1), 183–220.

Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2009). *Lectures on stochastic programming: modeling and theory (MPS-SIAM series on optimization)*. Philadelphia: SIAM.

Wu, D. D., & Olson, D. (2010). Enterprise risk management: a DFA VaR approach in vendor selection. *International Journal of Production Research*, *48*(16), 4919–4932.

Yau, S., Kwon, R. H., Rogers, J. S., & Wu, D. (2011). Financial and operational decisions in the electricity sector: contract portfolio optimization with the conditional value-at-risk criterion. *International Journal of Production Economics*, *134*(1), 67–77.