Journal of Economic Dynamics & Control 64 (2016) 23–38

Contents lists available at ScienceDirect

Journal of Economic Dynamics & Control

journal homepage: www.elsevier.com/locate/jedc

Multi-period mean–variance portfolio optimization based on Monte-Carlo simulation

F. Cong [a,*], C.W. Oosterlee [a,b]

[a] TU Delft, Delft Institute of Applied Mathematics, Delft, The Netherlands
[b] CWI-Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 16 September 2015
Received in revised form
26 November 2015
Accepted 1 January 2016
Available online 7 January 2016

JEL Classificiation:
C61
C63
G11

Keywords:
Dynamic portfolio management
Mean–variance optimization
Constrained optimization
Simulation method
Least squares regression

ABSTRACT

We propose a simulation-based approach for solving the constrained dynamic mean–variance portfolio management problem. For this dynamic optimization problem, we first consider a sub-optimal strategy, called the multi-stage strategy, which can be utilized in a forward fashion. Then, based on this fast yet sub-optimal strategy, we propose a backward recursive programming approach to improve it. We design the backward recursion algorithm such that the result is guaranteed to converge to a solution, which is at least as good as the one generated by the multi-stage strategy. In our numerical tests, highly satisfactory asset allocations are obtained for dynamic portfolio management problems with realistic constraints on the control variables.

1. Introduction

Since Markowitz's (1952) pioneering work on a single-period investment model, the mean–variance portfolio optimization problem has become a cornerstone of investment management in both academic and industrial fields. An interesting topic, extending Markowitz's work, is to consider the mean–variance target for a continuous or multi-period optimization problem. Along with introducing dynamic control into the optimization process, constraints on the controls can be included.

In some situations, the constrained dynamic mean–variance optimization problem can be solved analytically. For example, Li et al. (2002) solve this portfolio management problem with no-shorting of stock allowed and Bielecki et al. (2005) solve the problem with bankruptcy prohibition. In Fu et al. (2010), the authors investigate the mean–variance problem with a borrowing constraint, where the investor faces a borrowing rate different from the risk-free saving rate. However, all this research is performed in the framework of continuous optimization. In fact, as mentioned in Cui et al. (2014), the continuous constrained optimization problem is usually easier than the discrete one. In general, an elegant analytic mean–variance formulation can be derived in case of a complete market, where re-balancing can be performed continuously and there are no constraints on the controls and no jumps in asset dynamics. If we consider a realistic problem

* Corresponding author.
E-mail address: f.cong@tudelft.nl (F. Cong).

http://dx.doi.org/10.1016/j.jedc.2016.01.001
0165-1889/

which is designed in an incomplete market, it will be difficult to obtain analytic solutions and thus utilizing computational techniques to calculate numerical solutions is our preferred choice.

The nonlinearity of conditional variance is the main obstacle for solving the dynamic mean–variance optimization problem. In Zhou and Li (2000) and Li and Ng (2000), an embedding technique, by which the mean–variance problem is transformed into a stochastic linear-quadratic (LQ) problem, is introduced. For the linear-quadratic problem, an investor does not need to choose a trade-off parameter between mean and variance. Instead, she decides a final optimization target of her investment. In Zhou and Li (2000) and Li and Ng (2000), it is proved that varying the final investment target traces out the same efficient frontier as varying the mean–variance trade-off parameters. To generate the efficient frontier of the mean–variance optimization problem, we can thus solve the LQ problem with different target parameters.

To solve the constrained target-based problem, the Hamilton–Jacobi–Bellman partial differential equation (HJB PDE) is often considered. Accurate results can be generated by solving the HJB PDEs for a one-dimensional scenario. For example, Wang and Forsyth (2010) solve the continuous constrained mean–variance optimization problem with various constraints and the risky asset following geometric Brownian motion. The authors of Dang and Forsyth (2014) solve a similar problem with the risky asset following jump-diffusion dynamics. In both papers, realistic constraints are cast on the control variables. However, it is rather expensive to implement the algorithm, which is based on solving the HJB PDE, for a problem with several risky assets. Reducing the dimensionality of such a problem is a potential solution. However, when constraints are introduced, the assumption for establishing the well-known mutual fund theory is not valid and the ratio between the different risky assets is not constant any more. A general multi-dimensional problem can hardly be transformed into a one-dimensional problem.

To deal with the curse of dimensionality, using Monte-Carlo simulation constitutes a possible solution. A well-known simulation-based dynamic portfolio management algorithm is proposed in Brandt et al. (2005). However, they discuss a problem where the investor has constant relative risk aversion. For such problems, the investor's optimal asset allocation is only influenced by the dynamics of the risky asset, so dynamic programming can be performed after the (forward) simulation of the risky assets. For an investor with other types of risk aversion, her optimal intermediate decisions usually do not only depend on the dynamics of the risky asset but also on the amount of wealth at that time. The simulation approach proposed in Brandt et al. (2005) is therefore not feasible for a general investment problem. Solving the constrained dynamic mean–variance problem based on Monte-Carlo simulation is the focus of our work.

Our methods depend on transforming the mean–variance problem into the LQ problem, which is a target-based problem. In Basak and Chabakauri (2010), the investment strategy for solving the LQ problem is named the *pre-commitment strategy*, which however does not guarantee time consistency. As mentioned in Wang and Forsyth (2011), a time-consistent strategy can be formulated as a pre-commitment strategy plus time consistent constraints on the asset allocations. Thus, the pre-commitment strategy generally yields an efficient frontier which is superior to the one generated by a time consistent strategy. In this paper, we will contribute to pre-commitment strategies and propose two solutions for the dynamic mean–variance problem, one is performed in a forward manner and the other in a backward manner.

In the forward approach, we decompose the dynamic optimization problem into several static optimization problems by specifying intermediate investment targets at all re-balancing time steps. A reasonable approximation for the optimal controls can be determined at each single stage and solving the problems at all stages provides us a sub-optimal strategy called the "multi-stage strategy". We prove that the multi-stage strategy is the optimal strategy when there are no constraints on the asset allocations. Although the multi-stage strategy becomes sub-optimal in case of constrained controls, it is straight-forward to implement the multi-stage strategy for either high dimensional problems or problems with complicated constraints. While experimenting, we observe that the multi-stage strategy can yield highly satisfactory results compared to the reference solutions.

The main challenge to perform backward programming for a constrained optimization problem is that the value function at each time step is non-smooth and thus the optimality cannot be computed efficiently by solving the corresponding first order conditions. To tackle this problem, we utilize the idea of differential dynamic programming (Jacobson and Mayne, 1970), by which a stochastic control problem is solved by a local optimization strategy, and we come up with a backward recursive approach. Our backward recursive programming algorithm is an iterative method. With special design of the algorithm, we can guarantee that the outcome converges to a solution, which is not worse than the solution generated by the multi-stage strategy. In the backward process, conditional expectations are calculated recursively via cross-path least-squares regression. To make this numerical approach stable, we implement the "bundling" and "regress-later" techniques, proposed in Jain and Oosterlee (2015). The idea of "bundling" is highly compatible with the local optimization in differential dynamic programming. The backward recursive programming is initiated with a reasonable guess for the asset allocations, which can be, but is not restricted to, the one generated by the multi-stage strategy. In our tests with the initial allocation generated by the multi-stage strategy, we achieve highly satisfactory results after at most four backward iterations. Like the multi-stage strategy, the backward recursive programming can be performed highly efficiently. In our numerical tests, one iteration of the backward recursive programming only takes a few seconds.

This paper is organized as follows. In Section 2, we introduce the formulation of the dynamic mean–variance problem and the embedding into a stochastic LQ problem. Section 3 describes the multi-stage strategy. The optimality of the multi-stage strategy in the unconstrained case is proved in Section 3.2. In Section 4, the backward dynamic programming method is presented. Section 5 displays several realistic constraints for the portfolio management problems and in Section 6 numerical tests are performed for both one- and two-dimensional problems. We conclude in Section 7.

2. Problem formulation

This section describes the dynamic portfolio optimization problem for a defined contribution pension plan. We assume that the financial market is defined on a probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0 \le t \le T}, \mathbb{P})$ with finite time horizon $[0, T]$. The state space Ω is the set of all realizations of the financial market within the time horizon $[0, T]$. $\mathcal{F}$ is the sigma algebra of events at time T, i.e. $\mathcal{F} = \mathcal{F}_T$. We assume that the filtration $\{\mathcal{F}_t\}_{0 \le t \le T}$ is generated by the price processes of the financial market and augmented with the null sets of $\mathcal{F}$. Probability measure $\mathbb{P}$ is defined on $\mathcal{F}$.

For convenience, we consider a portfolio consisting of two assets, one risk-free and one risky. To extend the analysis to a problem with more than one risky asset is feasible. We consider a portfolio, which can be traded at discrete opportunities,[1] $t \in [0, \Delta t, \ldots, T-\Delta t]$, before terminal time T. At each trading time t, an investor decides her trading strategy to maximize the expectation of the terminal wealth and to minimize the investment risk. Formally, the investor's problem is given by

$$v_t(W_t) = \max_{\{x_s\}_{s=t}^{T-\Delta t}} \left\{ \mathbb{E}[W_T | W_t] - \lambda \cdot \mathrm{Var}[W_T | W_t] \right\}, \tag{1}$$

subject to the wealth restriction:

$$W_{s+\Delta t} = W_s \cdot (x_s R_s^e + R_f) + C \cdot \Delta t, \quad s = t, t+\Delta, \ldots, T-\Delta t.$$

Here x_s denotes the asset allocation of the investor's wealth in the risky asset in the period $[s, s+\Delta t)$. It is assumed that the admissible investment strategy x_t is an $\mathcal{F}_t$-measurable Markov control, i.e. $x_t \in \mathcal{F}_t$. R_f is the return of the risk-free asset in one time step, which is assumed to be constant for simplicity, and R_s^e is the excess return of the risky asset during $[s, s+\Delta t)$. We assume that the excess returns $\{R_t^e\}_{t=0}^{T-\Delta t}$ are statistically independent.[2] $C \cdot \Delta t$ stands for a contribution of the investor in the portfolio during $[s, s+\Delta t)$, and a negative C can be interpreted as a constant withdrawal of the investor from the portfolio. The risk aversion attitude of the investor is denoted by λ, which is a trade-off factor between maximizing the profit and minimizing the risk. $v_t(W_t)$ is termed the *value function*, which measures the investor's investment opportunities at time t with wealth W_t.

Remark 2.1. When the mean–variance optimization problem proposed by Eq. (1) is convex, then solving Eq. (1) is equivalent to determination of the Pareto-optimal points, i.e. solving

$$\begin{aligned} &\min_{\{x_s\}_{s=t}^{T-\Delta t}} \quad \left\{ \mathrm{Var}[W_T | W_t] \right\} \\ &\text{s.t.} \quad \mathbb{E}[W_T | W_t] \ge d, \end{aligned}$$

with a suitable choice of d. However, when the problem is not convex, solving Eq. (1) generates Pareto-optimal points, but not all of them.

The difficulty of solving this mean–variance optimization problem is caused by the nonlinearity of conditional variances, namely $\mathrm{Var}[\mathrm{Var}[W_T|\mathcal{F}_t]|\mathcal{F}_s] \ne \mathrm{Var}[W_T|\mathcal{F}_s], s \le t$, which makes the well-known dynamic programming valuation approach not applicable. To tackle this problem, the original mean–variance equations can be transformed into another framework as done in Zhou and Li (2000), Li and Ng (2000), and Wang and Forsyth (2010). The following theorem supports the transformation.

Theorem 2.2. *If* $\{x_s^*\}_{s=t}^{T-\Delta t}$ *is the optimal control for the problem defined in Eq.* (1), *then* $\{x_s^*\}_{s=t}^{T-\Delta t}$ *is also the optimal control for the following problem*:

$$\min_{\{x_s\}_{s=t}^{T-\Delta t}} \left\{ \mathbb{E}\left[\left(W_T - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\} \tag{2}$$

where $\gamma = \frac{1}{\lambda} + 2\mathbb{E}_{x^*}[W_T | W_t]$. *Here the operator* $\mathbb{E}_{x^*}[\cdot]$ *denotes the expectation of the investor's terminal wealth if she invests according to the optimal strategy* $\{x_s^*\}_{s=t}^{T-\Delta t}$.

Proof. See Li and Ng (2000).

Based on this theorem, the original mean-variance problem can be embedded into a tractable auxiliary LQ problem. The investment strategy corresponding to this LQ problem is called the pre-commitment strategy. This technique can also be interpreted as transforming the original mean–variance problem into a target-based optimization problem, which has been discussed in Haberman and Vigna (2002) and in Gerrard et al. (2004). For numerical computation, the pre-commitment optimization problem as shown in Eq. (2) is usually formulated as an HJB PDE and realistic constraints on either controls or state variables can be correspondingly established as boundary conditions. In this manner, Wang and Forsyth (2010) solve the mean-variance problem numerically and derives a solution to the constrained pre-commitment strategy.

[1] We assume that the re-balancing times are equidistantly spread and the total number of re-balancing opportunities before terminal time T is M. The time step Δt between two re-balancing days is $\frac{T}{M}$.

[2] For a risky asset with dynamics following geometric Brownian motion or a Levy process, the independence structure is valid. For some problems where the asset returns are directly defined, for example, as model-free data, the independence assumption is also correct. For a VAR or GARCH model, this assumption is, however, not satisfied.

However, even for the LQ problem, casting constraints in the numerical approach is in general not trivial. Imposing constraints on the controls will substantially change the formulation of the problem and make it nontrivial to solve the problem efficiently. The reasoning is as follows. For the unconstrained problem, the value function at each time step forms a smooth function and thus the optimality can be obtained by solving the first-order conditions associated to this smooth function. Adding constraints will remove the smoothness of the value function. Derivative-based optimization techniques cannot be applied in this situation and the optimality has to be computed by grid-searching on the whole domain of possible controls, for example, as in Wang and Forsyth (2010).

In the following section, we propose a sub-optimal yet highly efficient strategy for the mean–variance portfolio management problem. In this strategy, we avoid dealing with non-smooth value functions even if there are constraints on the controls. It is possible to extend this sub-optimal strategy to high-dimensional problems and to problems with complicated asset dynamics.

3. A forward solution: the multi-stage strategy

When we prescribe constraints on the allocations, neither the original mean-variance strategy associated to solving Eq. (1) nor the pre-commitment strategy associated to solving Eq. (2) is easy to obtain.

Here we treat the mean–variance optimization problem from a different angle. After writing it into the pre-commitment form, or equivalently into the target-based form, the objective of the reformulated optimization problem is to minimize the difference between the final wealth and a predetermined target. Hence the optimization problem at time t reads:

$$J_t^{*pc}(W_t) := \min_{\{x_s\}_{s=t}^{T-\Delta t}} \left\{ \mathbb{E}\left[\left(W_T - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}, \tag{3}$$

or, in a recursive fashion,

$$J_t^{*pc}(W_t) = \min_{x_t} \{ \mathbb{E}[J_{t+\Delta t}^{*pc}(W_{t+\Delta t}) | W_t] \}, \tag{4}$$

with $J_T^{*pc}(W_T) = \left(W_T - \frac{\gamma}{2}\right)^2$.

At the state (t, W_t), i.e. time t and wealth W_t, the value function $J_t(W_t)$ depends on all optimal allocations at subsequent time steps. That is why generally this kind of optimization problem has to be solved in a backward recursive fashion via, for example, solving the HJB PDEs or backward stochastic differential equations (BSDEs) with conditions at the terminal time.

Solving this dynamic programming problem numerically in a backward recursive fashion suffers from two problems. First, solving the optimality, especially for constrained cases, at each time step may be difficult or computationally expensive. Secondly, since we use the value function to transmit information between two recursive steps, the error accumulates as the recursion proceeds.

Reflection on these two issues leads us to define a sub-optimal strategy, which does not involve these two types of errors, which we call the multi-stage strategy.

Notice that at the terminal time, our target is $\frac{\gamma}{2}$. Then, at state (t, W_t), in the multi-stage strategy we choose x_t to be:

$$x_t^{*ms} := \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) + C \cdot \Delta t - \delta_{t+\Delta t} \right)^2 \middle| W_t \right] \right\}, \tag{5}$$

where

$$\delta_t = \frac{\frac{\gamma}{2} - C \cdot \Delta t \cdot \frac{1-(R_f)^{(T-t)/\Delta t}}{1-R_f}}{(R_f)^{(T-t)/\Delta t}}. \tag{6}$$

So, here we do not consider the optimality in the future, but perform a single-stage, or static, optimization with respect to a given target value.

Eq. (6) is straightforward. We set an intermediate target at time t such that once we achieve this target, we can put all the money in the risk-free asset and at the terminal time the wealth reaches the final target. Therefore, this intermediate target is computed by discounting the final target while taking into account the constant contribution.

Considering an intermediate wealth target is not a new idea. In Gerrard et al. (2006), the authors study the target-based optimization problem for a defined-contribution pension plan and make use of intermediate targets mainly for calculating the cumulative losses throughout the management period. In Cui et al. (2012), the authors consider an intermediate threshold. Once the portfolio wealth exceeded this threshold, they proved that withdrawing a suitable amount of money from the portfolio will not influence the performance of the portfolio in the sense of mean–variance optimization.

Since our multi-stage method merely depends on solving a single-stage optimization problem at each time point, the problem can be solved in a forward fashion:

- First, we generate the intermediate target values at each time step.
- Then, starting at the initial state we compute the optimal allocation step by step until the terminal time.

If we consider no periodic contributions, i.e. $C=0$, we can rewrite the optimal allocation for the pre-commitment problem in Eq. (3) as:

$$x_t^{*pc}(W_t) = \arg\min_{x_t}\left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot \prod_{s=t+\Delta t}^{T-\Delta t} (x_s^{*pc} R_s^e + R_f) - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}, \tag{7}$$

where $\{x_s^{*pc}\}_{s=t+\Delta t}^{T-\Delta t}$ denote the optimal allocations at times $s = t+\Delta t, \ldots, T-\Delta t$.

In this scenario the multi-stage optimization problem can be formulated as:

$$x_t^{*ms}(W_t) = \arg\min_{x_t}\left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot \prod_{s=t+\Delta t}^{T-\Delta t} (R_f) - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}. \tag{8}$$

We see that the objective of the multi-stage optimization is indeed different from that of the original optimization. Instead of considering the true optimization $\{x_s^{*pc}\}_{s=t+\Delta t}^{T-\Delta t}$, we "specify" $\{x_s^{*ms}\}_{s=t+\Delta t}^{T-\Delta t}$ to be zero. That is why we call the multi-stage solution sub-optimal. However, by sacrificing the possibility to pursue the optimality, we also gain some profits which are discussed in the next section.

Remark 3.1. The multi-stage strategy can be treated as a "greedy strategy" for the stochastic optimization problem (Boyd et al., 2013). In each stage, we minimize the distance between our wealth and the target of the current stage.

Remark 3.2. One crucial restriction for constructing the multi-stage strategy is that there should be a risk-free asset in the market. Regarding the wealth-to-income case discussed in Wang and Forsyth (2010), we cannot find a risk-free part in the market and therefore cannot derive the intermediate optimization target for our multi-stage approach.

3.1. Gains and losses

By the multi-stage strategy, we avoid the backward recursive programming by constructing determined intermediate targets. Since there is no error accumulation in the recursion, the optimization problem solved at the intermediate time step is unbiased[3] to what we designed it to be.

On the other hand, casting constraints on this sub-optimal strategy is trivial. Since at every time step we deal with a quadratic optimization problem in the multi-stage strategy, solving the constrained optimization problem can usually be performed efficiently. In the one-dimensional case, for example, solving the constrained quadratic optimization problem is equivalent to first solving the unconstrained problem to generate the optimal control and then truncating this optimal control by the constraints.

The drawback of the multi-stage strategy is also obvious: instead of tackling the original optimization problem, we work on a tailored one. The optimal control for the sub-optimal problem may differ from that for the original problem, so the mean–variance pair corresponding to this sub-optimal strategy may be located below the optimal efficient frontier. However, in the next section we will prove that in some situations the optimal allocation for the multi-stage strategy is exactly the same as that for the original pre-commitment strategy.

3.2. Equivalence in the unconstrained case

In this section, we restrict ourselves to the situation where there is no periodic contribution, i.e. $C=0$. Extending the analysis to the case where $C \neq 0$ is however possible. Under the following condition, we can prove that the multi-stage strategy and the pre-commitment strategy are equivalent for generating the optimal asset allocations.

Condition 3.3. The asset allocations at each time step are unconstrained.

In this case, we can obtain the analytic form of the value function at intermediate time steps for the pre-commitment problem.

Lemma 3.4. *For the pre-commitment problem shown in Eq. (3), the value function $J_t(W_t)$ can be formulated as:*

$$J_t(W_t) = L_t \cdot \left(W_t \cdot (R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2} \right)^2, \tag{9}$$

where $L_t = \prod_{s=t}^{T} l_s$ with l_t defined as follows:

$$l_t = 1 - \frac{\mathbb{E}[R_t^e]^2}{\mathbb{E}[(R_t^e)^2]}, \quad t = 0, \Delta t, \ldots, T-\Delta t,$$

$$l_T = 1.$$

[3] In the traditional backward programming approach, since numerical errors cumulate alongside the recursive procedure, biases in the intermediate optimization problems exist.

Proof. At time step T, the value function is known as:

$$J_T(W_T) = \left(W_T - \frac{\gamma}{2}\right)^2,$$

which satisfies Eq. (9). At time step $T-\Delta t$, the value function reads:

$$\begin{aligned} J_{T-\Delta t}(W_{T-\Delta t}) &= \max_{x_{T-\Delta t}} \{\mathbb{E}[J_T(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f)) | W_{T-\Delta t}]\} \\ &= \max_{x_{T-\Delta t}} \left\{ \mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right)^2 \middle| W_{T-\Delta t} \right] \right\}. \end{aligned} \tag{10}$$

To obtain the analytic form of $J_{T-\Delta t}(W_{T-\Delta t})$, we first need to determine the optimal asset allocation $x^*_{T-\Delta t}$, which satisfies:

$$x^*_{T-\Delta t} = \arg\min_{x_{T-\Delta t}} \left\{ \mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right)^2 \middle| W_{T-\Delta t} \right] \right\}.$$

Solving the first-order conditions of the optimality problem gives us that $x^*_{T-\Delta t}$ is the solution to the following equation:

$$\mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right) \cdot W_{T-\Delta t} R^e_{T-\Delta t} \middle| W_{T-\Delta t} \right] = 0.$$

So, the optimal allocation $x^*_{T-\Delta t}$ can be calculated as:

$$x^*_{T-\Delta t} = \frac{\left(\frac{\gamma}{2} - W_{T-\Delta t} R_f\right) \cdot \mathbb{E}[R^e_{T-\Delta t}]}{W_{T-\Delta t} \cdot \mathbb{E}[(R^e_{T-\Delta t})^2]}. \tag{11}$$

Inserting Eq. (11) into Eq. (10) yields:

$$\begin{aligned} J_{T-\Delta t}(W_{T-\Delta t}) &= \mathbb{E}\left[\left(1 - \frac{\mathbb{E}[R^e_{T-\Delta t}] \cdot R^e_{T-\Delta t}}{\mathbb{E}[(R^e_{T-\Delta t})^2]}\right)^2 \right] \cdot \left(W_{T-\Delta t} R_f - \frac{\gamma}{2}\right)^2 \\ &= \left(1 - \frac{\mathbb{E}[R^e_{T-\Delta t}]^2}{\mathbb{E}[(R^e_{T-\Delta t})^2]}\right) \cdot \left(W_{T-\Delta t} R_f - \frac{\gamma}{2}\right)^2. \end{aligned}$$

It is clear that the value function at time step $T-\Delta t$ has the same form as in Eq. (9). For the remaining time steps, we can formulate the value functions by backward induction.

Assume that at time step $t+\Delta t$ we have:

$$J_{t+\Delta t}(W_{t+\Delta t}) = L_{t+\Delta t} \cdot \left(W_{t+\Delta t} \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2}\right)^2.$$

Then at time step t, the value function can be formulated as:

$$\begin{aligned} J_t(W_t) &= \max_{x_t} \{\mathbb{E}[J_{t+\Delta t}(W_t \cdot (x_t R^e_t + R_f)) | W_t]\} \\ &= \max_{x_t} \left\{ \mathbb{E}\left[L_{t+\Delta t} \cdot \left(W_t \cdot (x_t R^e_t + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2}\right)^2 \middle| W_t \right] \right\} \\ &= \mathbb{E}[L_{t+\Delta t}] \cdot \max_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R^e_t + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2}\right)^2 \middle| W_t \right] \right\}. \end{aligned}$$

Here the last equality is based on the independence of the excess returns. Solving the first-order condition yields:

$$x^*_t = \frac{\left(\frac{\gamma}{2} - W_t \cdot (R_f)^{(T-t)/\Delta t}\right) \cdot \mathbb{E}[R^e_t]}{W_t \cdot (R_f)^{(T-t)/\Delta t - 1} \cdot \mathbb{E}[(R^e_t)^2]}. \tag{12}$$

and the corresponding value function reads:

$$\begin{aligned} J_t(W_t) &= L_{t+\Delta t} \cdot \left(1 - \frac{\mathbb{E}[R^e_t]^2}{\mathbb{E}[(R^e_t)^2]}\right) \cdot \left(W_t \cdot (R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2}\right)^2 \\ &= L_t \cdot \left(W_t \cdot (R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2}\right)^2. \end{aligned}$$

Thus we finalize the proof.□

Remark 3.5. The optimal asset allocation shown in Eq. (12) is exactly the same as the one proposed in Li and Ng (2000) for the one-dimensional case.

With Lemma 3.4, we can prove the equivalence between our multi-stage strategy and the pre-commitment strategy. Formally, the equivalence is shown in the following theorem.

Theorem 3.6. *For a mean–variance portfolio management problem, where Condition 3.3 is satisfied, the optimal control for the multi-stage strategy and for the pre-commitment strategy are identical. That is,* x_t^{*pc} *and* x_t^{*ms}*, as respectively shown in Eqs. (7) and (8), are equal at each time step t.*

Proof. For the pre-commitment problem, the optimal control x_t^{*pc} reads:

$$x_t^{*pc} = \arg\min_{x_t} \{ \mathbb{E}[J_{t+\Delta t}(W_t \cdot (x_t R_t^e + R_f)) | W_t] \}.$$

Using the form of the value function $J_{t+\Delta t}(\cdot)$ as shown in Lemma 3.4, we find:

$$x_t^{*pc} = \arg\min_{x_t} \left\{ \mathbb{E}\left[L_{t+\Delta t} \cdot \left(W_t \cdot (x_t R_t^e + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}.$$

Because of the independence of the excess returns, we can treat $L_{t+\Delta t}$ as a constant factor and thus the minimization problem turns out to be:

$$x_t^{*pc} = \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}, \tag{13}$$

Since the right-hand-sides of Eqs. (8) and (13) have the same form, the proof is finished.□

Notice that we only establish the equivalence between the multi-stage problem and the pre-commitment problem when the excess returns are independent and the allocations are unconstrained. If we equip the excess returns with path-dependent dynamics or cast constraints on the allocations, the equivalence may be lost. However, based on numerical results in Wang and Forsyth (2012), we know that the efficient frontiers for different investment strategies may differ significantly in the unconstrained case but the differences are relatively smaller when constraints are introduced. In our numerical approach we only apply the multi-stage strategy to generate asset allocations and then the mean-variance pair is calculated by combining the simulated trajectories and the corresponding "sub-optimal" allocations. As spotted in Wang and Forsyth (2010), small errors in asset allocations may not influence the accuracy of the mean–variance pair dramatically.

Even though the multi-stage strategy may be not equivalent to the pre-commitment strategy in some situations, it can serve as a sub-optimal solution to the constrained dynamic mean–variance portfolio optimization problem. If some other accurate solutions exist, the corresponding efficient frontiers should be at least above that of the multi-stage strategy. Moreover, for some numerical methods that depend on iteratively updating the asset allocations, a reasonable initial guess can be provided by the multi-stage method, see Section 4.

Skaf and Boyd (2008) proposed a similar technique to the multi-stage strategy for the quadratic convex optimization problem. Our research differs in two aspects. First, instead of dealing with one optimization problem which can yield one point on the efficient frontier, we consider a series of optimization problems which generate results to construct the whole efficient frontier. We find that the multi-stage strategy is particularly satisfactory when the investor is highly risk averse. However, in case the investor is less risk averse, the sub-optimal strategy turns out to be problematic. For this, we propose a backward dynamic programming approach, which is different from the forward strategy.

Remark 3.7. As discussed in Cui et al. (2012) and Dang and Forsyth (2014), a semi-self-financing strategy exists which is better than the pre-commitment strategy. In that strategy, a positive amount of money is withdrawn from the portfolio when the wealth in the portfolio is above a determined value. Similarly, the multi-stage strategy can be adjusted in this respect. However, since the improvement achieved by breaking the self-financing is not significant, we will not deal with the semi-self-financing multi-stage strategy in this paper.

4. Backward recursive programming

In the preceding sections, we considered the "greedy policy" for the dynamic optimization problem, and therefore the constrained dynamic mean–variance problem can be solved in a forward fashion via Monte-Carlo simulation. Except for the unconstrained case, the multi-stage strategy allocation is generally not the optimal solution to the dynamic optimization problem. In order to get the optimal solution, we have to consider a backward dynamic programming solution. In this section, we present an approach to perform backward recursive calculation based on the solution of the multi-stage strategy.

4.1. Benefit from the constraints

In general, constraints complicate dynamic optimization problems. For an unconstrained optimization problem, the value functions are smooth, so the optimality can be obtained by solving the first-order conditions. When constraints are introduced, the smoothness of the value functions is destroyed and derivative-based optimization is thus no longer feasible. However, if we treat the constraints differently, they can also be "helpful".

Consider an optimization problem at the state (t, W_t):

$$J_t(W_t) = \min_{x_t \in A} \{\mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t})|W_t]\}, \tag{14}$$

where A is the admissible set for the asset allocation x_t. $J_t(\cdot)$ is the value function at time t. When $A \neq \mathcal{R}$, this is a constrained problem, which may not be easy to solve. However, if we consider a special case where $A = \{x_t | x_t = K\}$, i.e. x_t is restricted to be a constant, the constrained problem becomes trivial. In fact, since we know that x_t has to be constant, the "optimal" solution in the admissible set is known immediately.

Using the multi-stage strategy, we obtain x_t^{*ms}, which may be a reasonable approximation of x_t^*, which denotes the real optimal allocation. If we construct a truncated admissible control set, $A_\eta = [x_t^{*ms} - \eta, x_t^{*ms} + \eta]$, the solution to the following optimization problem:

$$J_t(W_t) = \min_{x_t \in A_\eta} \{\mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t})|W_t]\},$$

should be the same as that of the problem shown in Eq. (14). Assuming that the optimal allocations for the state (t, W_t) are in an interval $[x_t^{*ms} - \eta, x_t^{*ms} + \eta]$, the investor's optimal wealth $W_{t+\Delta t}$ should be located in the domain[4]:

$$D_{t+\Delta t} := \{W_{t+\Delta t} | W_{t+\Delta t} = W_t \cdot (x_t \cdot R_t^e + R_f) + C \cdot \Delta t, \quad x_t \in A_\eta\}.$$

We further transform the original optimization problem shown in Eq. (14) to be:

$$J_t(W_t) = \min_{x_t \in A} \{\mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t})|W_t, W_{t+\Delta t} \in D_{t+\Delta t}]\}, \tag{15}$$

where an additional condition is introduced into the conditional expectation. Instead of considering the optimization problem on the whole domain of $W_{t+\Delta t}$, we restrict the optimization problem to a finite domain $D_{t+\Delta t}$ and thus establish a local optimization problem. For solving the stochastic optimization problem at state (t, W_t), we focus on the value function $J_{t+\Delta t}(W_{t+\Delta t})$ on a finite interval $D_{t+\Delta t}$.

4.2. Benefit from bundling

By means of the constraints, the original problem in Eq. (14) is simplified to a truncated problem in Eq. (15). To solve this truncated problem, we first need to determine the value function $J_{t+\Delta t}(W_{t+\Delta t})$ on domain $D_{t+\Delta t}$. A common simulation-based approach is that we vary x_t around x_t^{*ms} and perform sub-simulation, which is however involved and costly. One way to avoid sub-simulation is by plain Monte-Carlo simulation combined with bundling. Using the bundling technique, the domain $D_{t+\Delta t}$ can be approximated by:

$$\hat{D}_{t+\Delta t} = \{W_{t+\Delta t} | W_{t+\Delta t} = \hat{W}_t \cdot (x_t^{*ms} \cdot R_t^e + R_f) + C \cdot \Delta t, \quad \hat{W}_t \in B_\delta\},$$

where $B_\delta = [W_t - \delta, W_t + \delta]$. So, instead of varying x_t, we vary W_t by considering the paths whose states are around (t, W_t). For more details about bundling, we refer the readers to Jain and Oosterlee (2015).

4.3. Backward programming algorithm

Now we formally describe the algorithm for the backward programming stage.

- **Step 1: Initiation**:
  Generate an initial guess of optimal asset allocations $\{\tilde{x}_t\}_{t=0}^{T-\Delta t}$ and simulate the paths of optimal wealth values $\{W_t(i)\}_{i=1}^N, t = 0, \ldots, T$. At the terminal time T, we have the determined value function $J_T(W_T)$.
  The following three steps are subsequently performed, recursively, backward in time, at $t = T - \Delta t, \ldots, \Delta t, 0$.
- **Step 2: Solving**
  Bundle paths into B partitions, where each bundle contains a similar number of paths and the paths inside a bundle have similar values at time t. Denote the wealth values associated to the paths in the bundle by $\{W_t^b(i)\}_{i=1}^{N_B}$, where N_B is the number of paths in the bundle. Within each bundle, we perform the following procedure.
  - For paths in the bundle, we have the corresponding wealth values $\{W_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ and the continuation values $\{J_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ at time $t + \Delta t$. So, a function $f_{t+\Delta t}^b(\cdot)$, which satisfies $J_{t+\Delta t}^b = f_{t+\Delta t}^b(W_{t+\Delta t}^b)$ on the local domain, can be determined by *regression*.[5]
  - For all paths in the bundle, since the value function $f_{t+\Delta t}^b(W_{t+\Delta t}^b)$ has been approximated, we solve the optimization problem by calculating the first-order conditions. In this way, we get new asset allocations $\{\hat{x}_t^b(i)\}_{i=1}^{N_B}$.

[4] This domain is much smaller than the domain obtained without restricting x_t. To avoid unnecessary technicalities, we assume that the wealth process is locally bounded.

[5] Here regression refers to the technique of approximating the target function by a truncated basis function expansion, where the expansion coefficients are determined by minimizing the approximation error in the least squares sense.

- Since the wealth values $\{W_t^b(i)\}_{i=1}^{N_B}$ and the allocations $\{\hat{x}_t^b(i)\}_{i=1}^{N_B}$ are known, by *regression* we can also compute the new continuation values $\{\hat{J}_t^b(i)\}_{i=1}^{N_B}$. Here $\hat{J}_t^b(i)$ is the expectation of $J_{t+\Delta t}(W_{t+\Delta t})$ conditional on $W_t^b(i)$ and $\hat{x}_t^b(i)$, that is,

$$\hat{J}_t^b(i) = \mathbb{E}[J_{t+\Delta t}(W_{t+\Delta t})|W_t = W_t^b(i), x_t = \hat{x}_t^b(i)].$$

- **Step 3: Updating**
  For the paths in a bundle, since we have an old guess $\{\tilde{x}_t^b\}_{i=1}^{N_B}$ for the asset allocations, by *regression* we can also calculate the old continuation values $\{\tilde{J}_t^b(i)\}_{i=1}^{N_B}$. For the i-th path, if $\tilde{J}_t^b(i) > \hat{J}_t^b(i)$, we choose $\hat{x}_t^b(i)$ as the updated allocation. Otherwise we retain the initial allocation. We denote the updated allocations by $\{x_t^b(i)\}v_{i=1}^{N_B}$.
- **Step 4: Evolving**
  Once the updated allocations $\{x_t^b(i)\}_{i=1}^{N_B}$ are obtained, again by *regression* we can calculate the "updated" continuation values $\{J_t^b(i)\}_{i=1}^{N_B}$ and proceed with the backward recursion.

In the algorithm, at each time step and inside each bundle, four regression steps are performed. Especially, the last three regression steps are added for calculating value functions. Since the value function is used to evolve information between time steps, an error in calculating them will accumulate due to recursion. In general, we can settle this problem by using a very large number of simulations, which is however expensive. In our numerical approach we always use the "regress-later" technique as applied in Cong and Oosterlee (2015).

When we use the regression, polynomials up to order two are considered as basis functions. For the unconstrained problem, the value function is a quadratic function, so this choice of basis functions is sufficient. In the constrained case, although the value function is non-smooth, it is still piecewise quadratic as stated in Fu et al. (2010). Since the regression is performed with respect to paths in the same bundle, it is a local regression, i.e. local polynomial fitting. In all, our algorithm, which adopts second order local polynomial fitting to approximate piecewise quadratic functions, should yield satisfactory results.

After one iteration of the algorithm, we will obtain an "updated" asset allocation at each time step. The algorithm can be performed iteratively. In the remaining part of this section, we will prove that these iterations will lead to a convergent result.

Remark 4.1. In the engineering field, this type of dynamic programming is called differential dynamic programming. For more details, we refer the reader to Jacobson and Mayne (1970) and Tassa et al. (2008).

4.4. Convergence of the backward recursive programming

For the dynamic backward recursive programming, we have

$$J_t(W_t) = \min_{x_t} \{ \mathbb{E}[J_{t+\Delta t}(W_t \cdot (x_t R_t^e + R_f) + C \cdot \Delta t)|W_t] \}, \quad t = T - \Delta t, \dots, 0,$$

where the expectation is taken over the return R_t^e. This recursion can be written as:

$$J_t = \Psi_t J_{t+\Delta t}, \quad t = T - \Delta t, \dots, 0,$$

where Ψ_t is the Bellman operator, defined as

$$(\Psi_t h)(W_t) = \min_{x_t} \{ \mathbb{E}[h(W_t \cdot (x_t R_t^e + R_f) + C \cdot t)|W_t] \}. \tag{16}$$

According to Bertsekas (1995), the Bellman operator has the monotonicity property, which is described by the following lemma.

Lemma 4.2 (*Monotonicity*). *For any* $f: \mathcal{R} \to \mathcal{R}$ *and* $g: \mathcal{R} \to \mathcal{R}$,

$$f \le g \Rightarrow \Psi_t f \le \Psi_t g,$$

where the inequalities are interpreted pointwise, and Ψ_t *is as in* (16).

We can prove the following proposition:

Proposition 4.3. *The backward recursive updating process, as explained in Section* 4.3, *converges. That is: there exists a function* $J_0^*(\cdot)$ *satisfying*

$$J_0^*(W_0) = \lim_{k \to +\infty} J_0^{(k)}(W_0),$$

where $J_0^{(k)}(W_0)$ *denotes the value function at initial state after k iterations of the algorithm in Section* 4.3.

Proof. The proof is directly based on the design of the backward recursive programming algorithm and the Monotonicity Lemma 4.2. Since at each iteration, we compare the previous and the new allocations and retain the one which generates a smaller value function and since the Bellman operator preserves monotonicity, we get:

$$J_0^{(1)}(W_0) \geq J_0^{(2)}(W_0) \geq \cdots \geq J_0^{(k)}(W_0), \quad k = 1, \ldots, k_{\max}.$$

Since the value function is positive, according to the Monotone Convergence Theorem, we can finalize the proof.□

Whereas the convergence can be proved, the convergence rate is not determined. The convergence rate depends on the smoothness of the value functions and the initial guess of the allocations. For example, for a smooth quadratic value function, any initial guess should give the correct result after one iteration. On the other hand, if we choose zero as the initial guess for all the asset allocations, the solution can never depart from the stationary point, which is generated by the risk-free investment strategy. In our numerical tests, we always achieve satisfactory results using the initial guess of the asset allocations generated by the multi-stage strategy.

On the other hand, the solution of our backward recursive approach is not guaranteed to be the optimal one. We call this algorithm "sub-optimal" because a bias is introduced when we approximate the non-smooth value function by piecewise quadratic polynomials. Although the numerical tests indicate that our algorithm generates highly satisfactory solutions, we expect that in case the value function is highly non-smooth the approximation bias in our algorithm will be large and the accuracy of our algorithm may be unsatisfactory.

5. Constraints on the asset allocations

From the perspective of real-life applications, introducing constraints on the asset allocations is important. For example, when an investor goes bankrupt, she should not be allowed to manage her portfolio any more. Besides, according to regulations for banks, the leverage ratio should be bounded. In this section, we will formulate controls on the asset allocations.

5.1. No bankruptcy constraint

In this paper, the "no bankruptcy" constraint implies that there is zero probability of bankruptcy when the constraint is cast on the asset allocations. To ensure that the allocation at time t does not lead to bankruptcy at time $t+\Delta t$, we need:

$$W_t \cdot (x_t R_t^e + R_f) + C\Delta t \geq 0. \tag{17}$$

Note that if there is a "no bankruptcy" constraint, definitely we have $W_t > 0$. Therefore, to guarantee that Eq. (17) is valid, we need:

$$x_t R_t^e \geq -R_f - \frac{C\Delta t}{W_t}. \tag{18}$$

Since $R_t^e = \exp(r_t^e) - R_f$, where r_e is a random variable, we require

$$-R_f \geq -\frac{R_f}{x_t} - \frac{C\Delta t}{W_t \cdot x_t} \quad \text{and} \quad x_t \geq 0.$$

to ensure Eq. (18) to be valid. Reformulating these equations, we have:

$$0 \leq x_t \leq 1 + \frac{C\Delta t}{W_t \cdot R_f}. \tag{19}$$

The "no bankruptcy" constraint given by Eq. (19) implies $\lim_{W_t \to 0}(x_t \cdot W_t) = 0$, as given in Wang and Forsyth (2010) and Wang and Forsyth (2011) for the continuous portfolio optimization problem. Our version of the "no bankruptcy" constraint for the multi-period case is stronger than the constraint in Wang and Forsyth (2010, 2011). Rather than specifying that wealth should not be invested in the risky asset when the total wealth amount is close to zero, our constraint also indicates that special consideration to the asset allocation should be given even though the wealth amount is far above zero. This is due to the difference between continuous and discrete re-balancing. In the latter case, since we cannot manage the portfolio between two re-balancing opportunities, it is possible that the investor goes bankrupt after an extreme market movement. To avoid this situation, we need to impose more strict constraints on the asset allocations, which is however not necessary for a continuous re-balancing problem.

5.2. No bankruptcy constraint with 1–2α% certainty

When the wealth in a portfolio is large, according to the discussion in the last subsection, the upper bound for "no bankruptcy" constraint (19) will be close to 1, which is quite rigorous. Using this as the upper bound protects an investor from bankruptcy only in the rare case that the risky asset yields zero return. A possible way to relieve this constraint is to take the possibility of bankruptcy into account.

Assume that the α- and the $(1-\alpha)$-quantiles of the excess return R_t^e are respectively $R_t^{e,\alpha}$ and $R_t^{e,1-\alpha}$. Then with certainty 1–2α% and the constraints shown below, we can guarantee that the "no bankruptcy" constraint in Eq. (17) is valid. The bounds for the asset allocations can be computed as:

$$\frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,1-\alpha}} \le x_t \le \frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,\alpha}}. \tag{20}$$

Remark 5.1. The constraints proposed by Eq. (20) correspond to discrete monitoring on re-balancing times. When monitoring is performed continuously, more strict constraints may be obtained.

5.3. Bounded leverage

Eqs. (19) and (20) imply that when an investor's wealth is close to zero, the upper bound for the allocation in the risky asset goes to infinity. The "no bankruptcy" constraint does not forbid an investor from gambling when she is almost bankrupt. To avoid this, we can impose constraints on the leverage ratios, for example, by restricting the proportion of investor's wealth in the risky asset to be within $[x_{\min}, x_{\max}]$.

6. Numerical experiments

In this section, we test our algorithms by solving several multi-period mean–variance portfolio management problems. We start with a simple case with one risky asset and one risk-free asset in the portfolio. We choose geometric Brownian motion as the dynamics of the risky asset, and assume that the log-return of the risky asset has volatility σ and mean $r_f + \xi \cdot \sigma$. Here r_f is the log-return of the risk-free asset and ξ the market price of risk. Since only the return of the risky asset is stochastic, we call this problem a "1D problem". We will also consider a "2D problem" with two risky assets and one risk-free asset in the portfolio. We consider only bounded leverage constraints in the 2D problem, and therefore it can be solved highly efficiently.

In the numerical tests, we choose the sample size to be 50 000 and the number of bundles to be 20 in the backward recursive programming stage. When we employ the "no bankruptcy" constraint (20), we choose parameter α to be sufficiently small, $\alpha = 10^{-8}$, which ensures that the undesired event will not happen. Regarding the backward recursive programming, we use a common random seed to generate Monte-Carlo paths for one run of the algorithm (including one forward and several backward processes). To ensure that the choice of random seed does not introduce a bias, we consider 20 different random seeds in all tests.

Based on reference results, we choose three different sets of parameters, shown in Table 1.

6.1. 1D problem

6.1.1. Multi-stage optimization with constraints

We first test the plain multi-stage strategy using the model parameters from Set I in Table 1. Different types of constraints are prescribed on the asset allocations. The bounded control is chosen to be [0,1.5]. Fig. 1 shows that constraints on the controls have a significant influence on the efficient frontiers obtained by the multi-stage strategy.

Without periodic contribution, i.e. $C=0$, an analytic solution to the "no bankruptcy" case with continuous re-balancing is presented in Bielecki et al. (2005). For a corresponding test, we choose the model parameters from Set II in Table 1. In Fig. 2, we can see that when the number of re-balancing opportunities is sufficiently large, the efficient frontier generated by the multi-stage strategy is close to the analytic solution, especially when the investment risk is not large.

6.1.2. Backward recursive programming

In this section, we show the performance of the backward recursive programming stage. Unless indicated differently, the initial asset allocations are generated by the multi-stage strategy. We first consider a continuous re-balancing scenario shown in Fig. 2, where an analytic solution is available. We observe that when backward recursive programming is

Table 1
Parameter settings used in the test.

Set I (From Wang and Forsyth, 2010):
r_f=0.03, $\xi = 0.33$, $\sigma = 0.15$, C=0.1, T=20 (years), $W_0 = 1$, $M^* = 80$.
Set II (From Bielecki et al., 2005):
r_f=0.06, $\xi = 0.4$, $\sigma = 0.15$, C=0, T=1 (year), $W_0 = 1$.
Set III (From Forsyth and Vetzal, 2014):
r_f=0.03, $\xi = 0.4$, $\sigma = 0.15$, C=0, T=30 (years), $W_0 = 100$, M=30.

* M denotes the number of re-balancing opportunities, which are equidistantly distributed in T years.

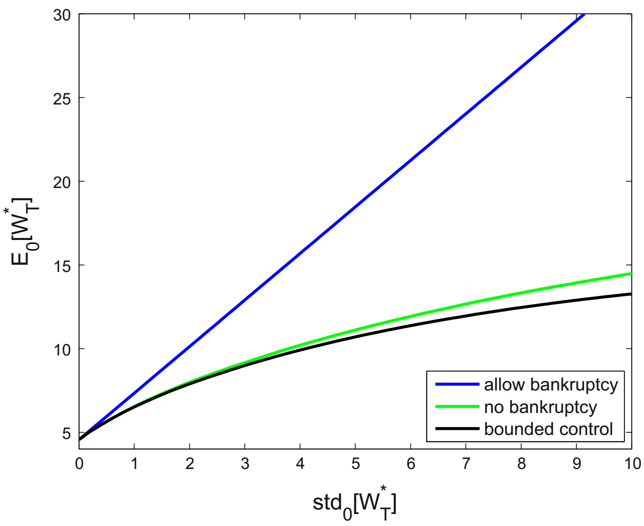

Fig. 1. Different types of constraints on the management strategies. By varying the target $\gamma \in (9.125, 85.125)$, we trace out the efficient frontiers.

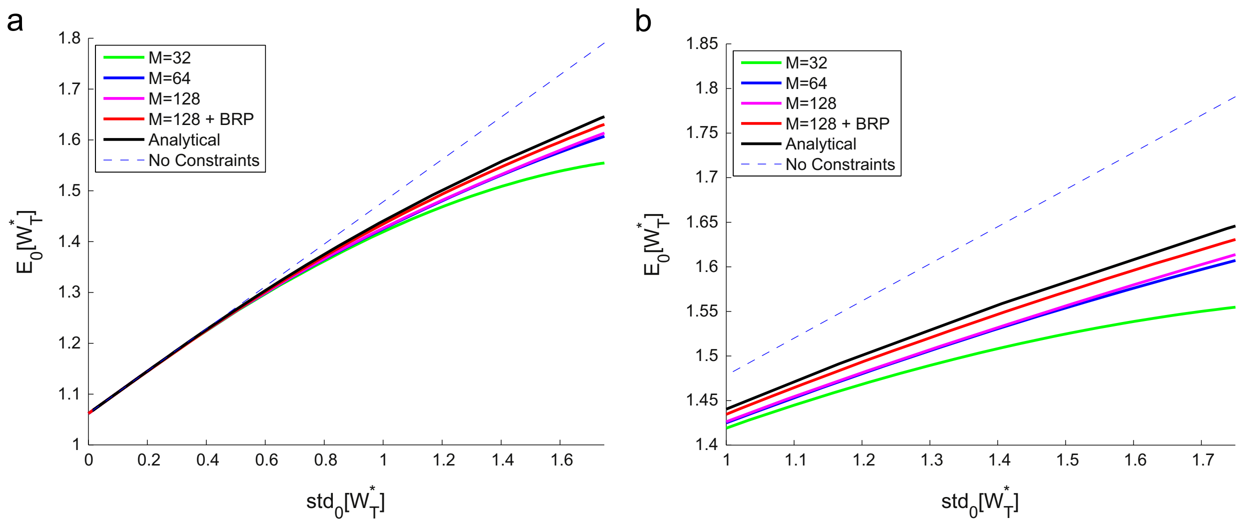

Fig. 2. The "no bankruptcy" case. An analytic solution is available if continuous re-balancing is applied. We check the efficient frontier generated by our algorithm. For backward recursive programming, we use the results obtained after four backward iterations when $M=128$. (a) Efficient frontier with no bankruptcy constraint. (b) Efficient frontier with no bankruptcy constraint (zoom in).

implemented, we obtain a better efficient frontier than the one generated by the multi-stage strategy. Moreover, we find that the efficient frontier resulting from backward recursive programming is close to the analytic solution.

Subsequently, we perform tests for a multi-period portfolio management case presented in Forsyth and Vetzal (2014) with the parameters from Set III in Table 1. We consider bounded constraints [0,1.5] on the asset allocations.

In Fig. 3, we present the efficient frontiers generated by the multi-stage approach and by the first four iterations in the backward recursive programming stage. Again we can see that the backward recursive approach generates better results than the multi-stage approach. Even with one iteration of backward programming, the result is already highly satisfactory.

In Table 2, we consider two cases for which also reference results are available. We find that in general, even with the same terminal target, the mean–variance pair calculated by the multi-stage strategy is different from the reference value. However, after the backward recursive programming, the result is highly satisfactory. In fact, the backward programming even provides results somewhat superior to the reference values.[6] Moreover, the multi-stage strategy and the backward recursive programming stage cost only seconds.

[6] For generating the efficient frontiers, we implement Monte-Carlo simulation while the authors of Forsyth and Vetzal (2014) utilize a PDE approach. As explained in Ma and Forsyth (2016), the simulation based approach usually yields slightly better results.

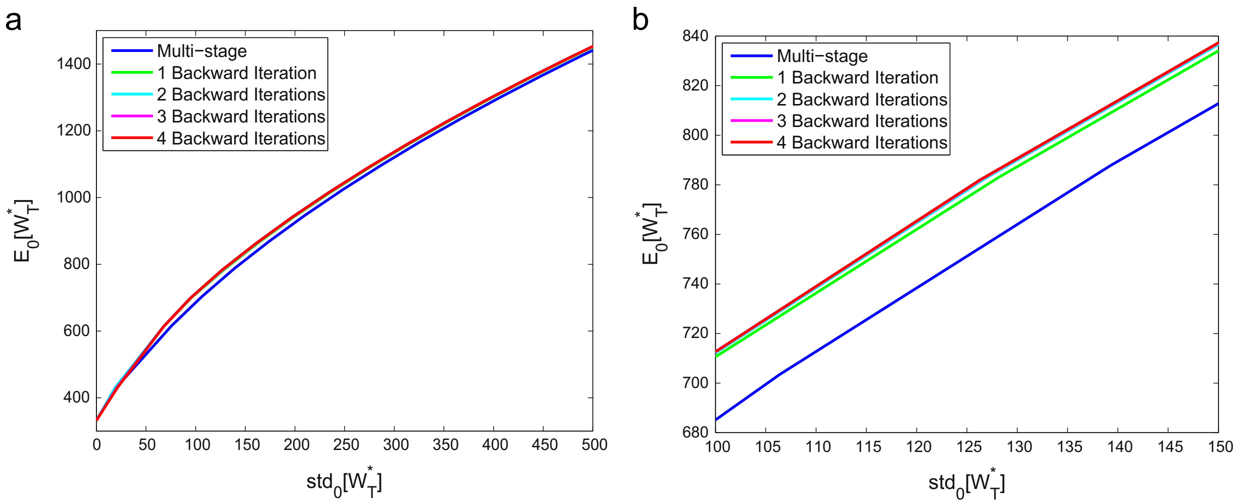

Fig. 3. Backward recursive calculation using the allocations generated by the multi-stage strategy as guesses for the optimal asset allocations. (a) Efficient frontier with bounded control. (b) Efficient frontier with bounded control (zoom in).

Table 2
Comparison of the results generated by the multi-stage strategy and the backward recursive programming to the reference values in Forsyth and Vetzal (2014).

Target	$\gamma = 1751.94$			$\gamma = 5856.15$		
	$E_0^*(W_T)$ (s.e.)	$\text{Std}_0^*(W_T)$ (s.e.)	CPU time (in seconds)	$E_0^*(W_T)$ (s.e.)	$\text{Std}_0^*(W_T)$ (s.e.)	CPU time (in seconds)
Reference	816.62	142.85		2008.55	969.33	
Multi-stage	823.84 (0.71)	154.37 (1.28)	4.71	2031.65 (4.86)	987.55 (2.54)	4.17
One backward iteration	818.83 (0.70)	143.33 (1.30)	9.09	2018.47 (4.73)	969.29 (2.58)	8.23
Four backward iterations	817.74 (0.70)	141.40 (1.28)	22.25	2014.90 (4.73)	964.80 (2.62)	20.34

The backward recursive programming technique is robust regarding different choices of initial allocations. As shown in Fig. 4, even though we fix the initial asset allocations to be constant,[7] the backward recursive programming stage still gives us a satisfactory result after some iterations.

6.2. 2D problem with box constraints

To tackle dynamic portfolio management problems with either the forward or the backward strategy proposed, we essentially need to deal with a constrained convex optimization problem. Some fast numerical solvers exist for this kind of problems in high dimensional scenarios. In this section, we will consider a simple 2D case where box constraints are cast on the asset allocations. In this case, bounded controls are prescribed for the allocations of both risky assets. Solving this constrained 2D convex optimization problem is therefore equivalent to solving five simple optimization problems and choosing the best results from them. The reason is that for a quadratic optimization problem with box constraints the optimal solution lies either at the boundary or in the interior of the admissible set. These five optimization problems include one unconstrained 2D problem[8] and four 1D problems with bounded constraints.

We use the parameters displayed in Set III from Table 1 and for the other risky asset we use the same market price of risk but a higher volatility, $\sigma_b = 0.4$. The correlation ρ between these two risky assets is fixed at $\rho = 0.4$, unless mentioned otherwise. For both assets, we prescribe bounded constraints [0,0.75] on their asset allocations.

First, we test the influence of adding another risky asset to the portfolio. Here we simply implement the multi-stage strategy for generating the mean–variance efficient frontier. As shown in Fig. 5, increasing diversification in the portfolio has a significant impact on the solution of the dynamic optimization problem. When the correlation between the two risky assets is closer to -1, a higher efficient frontier can be obtained. This is intuitive, because a large part of the volatility can be hedged in the case of two negatively correlated risky assets. When their correlation gets larger, the efficient frontier gets

[7] In general this will lead to quite a rough initial approximation of the mean–variance pair.
[8] First, we solve the unconstrained 2D problem. Then we penalize the optimal solution when the constraints are not satisfied.

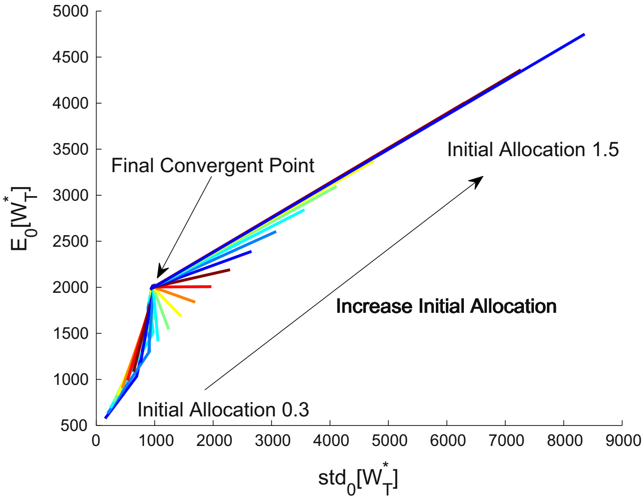

Fig. 4. A constant initial guess for the asset allocation. The constant varies from 0.3 to 1.5. Generally, this choice will lead to an inaccurate estimate of the mean–variance pair. However, after several (in our tests, at most four) iterations of backward programming, we achieve highly satisfactory results.

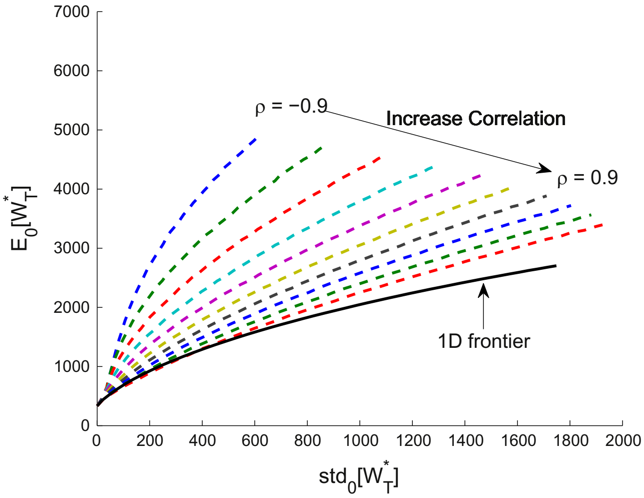

Fig. 5. Efficient frontiers obtained by investing in two risky assets with different correlations. In each scenario, the two risky assets share the same market price of risk, but one with high volatility and the other with low volatility. Bounded control [0,0.75] is cast on both assets. For the 1D test case, we employ the risky asset with low volatility in the portfolio and bounded control [0,1.5].

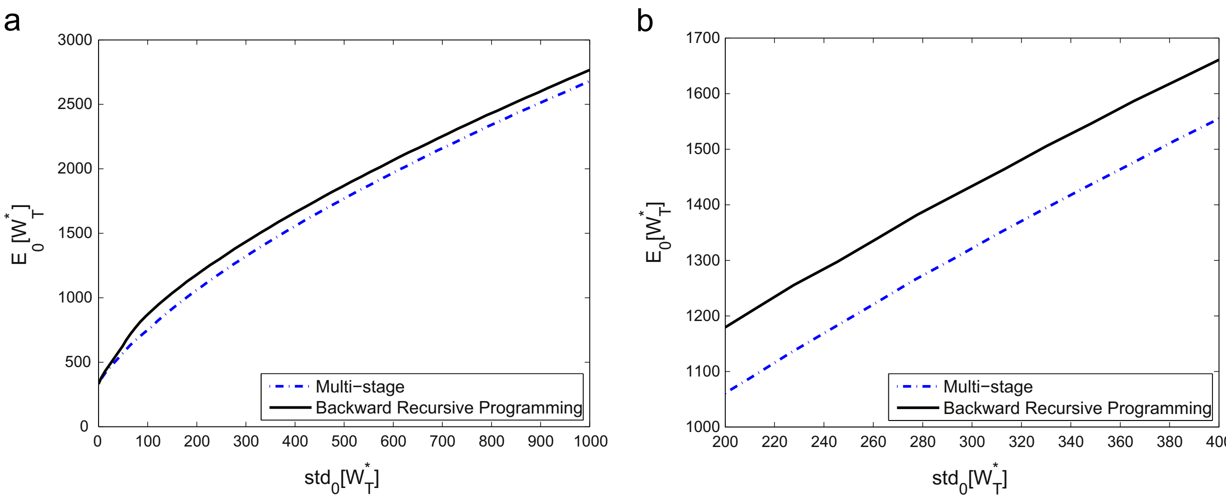

Fig. 6. We compare forward and backward strategies for the 2D dynamic portfolio management problem with box constraints. When considering the backward recursive programming, we use the results obtained after four iterations. (a) 2D efficient frontier. (b) 2D efficient frontier (zoom in).

worse. However, in most cases two risky assets in the portfolio yield better results than having one risky asset in the portfolio. For example, when we choose the correlation to be 0.4 and the final target to be 5856.15, as used in Section 6.1.2, we obtain $[\mathbb{E}_0[W_T^*], \text{Std}_0[W_T^*]] = [2501.41, 893.87]$ which is significantly better than $[\mathbb{E}_0[W_T^*], \text{Std}_0[W_T^*]] = [2031.65, 987.55]$ as acquired in the 1D case.

In Fig. 6, we compare the multi-stage strategy and the backward recursive programming approach. The outcome is similar to that observed in the 1D tests. When we implement the backward recursive programming stage, a significant improvement is obtained. For example, when the standard deviation is around 200, an almost 10% higher expected return can be obtained if we consider the backward recursive programming approach rather than the multi-stage approach.

Remark 6.1. In the 2D case, we also observe that satisfactory results can be obtained after several iterations of backward recursive programming even if we start with an inaccurate initial guess of the asset allocations.

7. Conclusion

In this paper, we propose simulation-based approaches for solving the dynamic mean–variance portfolio management problem. To deal with the nonlinearity of the conditional variance, we use the embedding technique introduced in Li and Ng (2000) to transform the mean–variance optimization problem into a linear-quadratic problem, which has a determined final optimization target. To tackle this target-based dynamic optimization problem, also known as "the pre-commitment problem", we propose two approaches, one in a forward fashion and the other in a backward fashion.

The forward approach, called the "multi-stage strategy", is based on determining an intermediate investment target at each re-balancing time. The intermediate target is chosen as the amount of wealth, which, if obtained, an investor can invest with a risk-free strategy and still reach the final investment target. Although it is generally believed that backward programming is essential for solving a dynamic optimization problem, we prove that the multi-stage strategy yields optimal controls when no constraints are involved. In the case that there are constraints on the controls, the multi-stage strategy can only yield a sub-optimal solution. However, since it is a forward and thus highly efficient approach, it is always feasible even when the dimensionality of the problem increases.

Although the forward approach is fast and easy-to-implement, in general it is not optimal for a dynamic optimization problem. Therefore, we propose another simulation-based approach which involves backward recursive programming. The main idea of the backward recursive approach is that we consider local quadratic optimization instead of global optimization. By tailoring the numerical algorithm, the backward recursive programming is guaranteed to yield convergent results after several iterations. In the numerical tests, it is shown that, although the backward approach is also sub-optimal, it always generates better efficient frontiers than the multi-stage strategy.

In the backward approach, we need to calculate conditional expectations associated to each simulated path recursively by least-squares regression. To make this regression-based numerical approach stable, we implement "bundling" and the "regress-later" techniques, as introduced in Jain and Oosterlee (2015). We find that our backward recursive approach is very robust. Even if it is initiated by an inaccurate guess for the allocation, highly satisfactory results can be obtained after several iterations.

As both the forward and backward approaches are based on simulation, it is feasible to implement them for high-dimensional problems. Moreover, since at each single stage or recursive step we consider a constrained quadratic optimization problem, it is possible to solve them efficiently with suitable quadratic optimization algorithms. Combining the algorithms with these quadratic optimization algorithms for a general high-dimensional case will be our future work. Another research direction is to implement our backward recursive programming approach to generate a time-consistent strategy.

References

Basak, Suleyman, Chabakauri, Georgy, 2010. Dynamic mean–variance asset allocation. Rev. Financ. Stud. 23 (8), 2970–3016.

Bertsekas, Dimitri P., 1995. Dynamic Programming and Optimal Control, Vol. 1. Athena Scientific, Belmont, MA.

Brandt, Michael W., Goyal, Amit, Santa-Clara, Pedro, Stroud, Jonathan R., 2005. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. Rev. Financ. Stud. 18 (3), 831–873.

Bielecki, Tomasz R., Jin, Hanqing, Pliska, Stanley R., Zhou, Xunyu, 2005. Continuous-time mean–variance portfolio selection with bankruptcy prohibition. Math. Finance 15 (2), 213–244.

Boyd, Stephen, Mueller, Mark, O'Donoghue, Brendan, Wang, Yang, 2013. Performance bounds and suboptimal policies for multi-period investment. Found. Trends Optim. 1 (1), 1–69.

Cui, Xiangyu, Gao, Jianjun, Li, Xun, Li, Duan, 2014. Optimal multi-period mean–variance policy under no-shorting constraint. Eur. J. Oper. Res. 234 (2), 459–468.

Cui, Xiangyu, Li, Duan, Wang, Shouyang, Zhu, Shushang, 2012. Better than dynamic mean–variance: time inconsistency and free cash flow stream. Math. Finance 22 (2), 346–378.

Cong, Fei, Oosterlee, Cornelis W., 2015. Accurate and Robust Numerical Methods for the Dynamic Portfolio Management Problem. Available at SSRN 2563049.

Dang, Duy-Minh, Forsyth, Peter A., 2014. Better than Pre-Commitment mean–Variance Portfolio Allocation Strategies: A Semi-Self-Financing Hamilton–Jacobi–Bellman Equation Approach. Available at SSRN 2368558.

Dang, Duy-Minh, Forsyth, Peter A., 2014. Continuous time mean–variance optimal portfolio allocation under jump diffusion: an numerical impulse control approach. Numer. Methods Partial Differ. Equ. 30 (2), 664–698.
Fu, Chenpeng, Lari-Lavassani, Ali, Li, Xun, 2010. Dynamic mean–variance portfolio selection with borrowing constraint. Eur. J. Oper. Res. 200 (1), 312–319.
Forsyth, Peter A., Vetzal, Kenneth R., 2014. Long Term Asset Allocation for the Patient Investor.
Gerrard, Russell, Haberman, Steven, Vigna, Elena, 2004. Optimal investment choices post-retirement in a defined contribution pension scheme. Insur.: Math. Econ. 35 (2), 321–342.
Gerrard, Russell, Haberman, Steven, Vigna, Elena, 2006. The management of decumulation risks in a defined contribution pension plan. N. Am. Actuar. J. 10 (1), 84–110.
Haberman, Steven, Vigna, Elena, 2002. Optimal investment strategies and risk measures in defined contribution pension schemes. Insur.: Math. Econ. 31 (1), 35–69.
Jacobson, David H., Mayne, David Q., 1970. Differential Dynamic Programming. American Elsevier Pub. Co, New York, NY.
Jain, Shashi, Oosterlee, Cornelis W., 2015. The stochastic grid bundling method: efficient pricing of Bermudan options and their Greeks. Appl. Math. Comput. 269 (1), 412–431.
Li, Duan, Ng, Wan-Lung, 2000. Optimal dynamic portfolio selection: multiperiod mean–variance formulation. Math. Finance 10 (3), 387–406.
Li, Xun, Zhou, Xunyu, Lim, Andrew E.B., 2002. Dynamic mean–variance portfolio selection with no-shorting constraints. SIAM J. Control Optim. 40 (5), 1540–1555.
Markowitz, Harry M., 1952. Portfolio selection. J. Finance 7 (1), 77–91.
Ma, Kai, Forsyth, Peter A., 2016. Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous time mean variance asset allocation under stochastic volatility. J. Comput. Finance (Forthcoming).
Skaf, Joëlle, Boyd, Stephen, 2008. Multi-Period Portfolio Optimization With Constraints and Transaction Costs. Technical Report, Citeseer.
Tassa, Yuval, Erez, Tom, Smart, William D., 2008. Receding horizon differential dynamic programming. In: Advances in Neural Information Processing Systems, pp. 1465–1472.
Wang, Jian, Forsyth, Peter A., 2010. Numerical solution of the Hamilton–Jacobi–Bellman formulation for continuous time mean variance asset allocation. J. Econ. Dyn. Control 34 (2), 207–230.
Wang, Jian, Forsyth, Peter A., 2011. Continuous time mean variance asset allocation: a time-consistent strategy. Eur. J. Oper. Res. 209 (2), 184–201.
Wang, .Jian, Forsyth, Peter A., 2012. Comparison of mean variance like strategies for optimal asset allocation problems. Int. J. Theor. Appl. Finance 15 (02).
Zhou, Xunyu, Li, Duan, 2000. Continuous-time mean–variance portfolio selection: a stochastic LQ framework. Appl. Math. Optim. 42 (1), 19–33.