A Case Study in Multiperiod Portfolio Optimization: A Classic Problem Revisited*

Martin B. Tarlie†

Abstract

Conventional wisdom holds that multiperiod portfolio optimization problems are best, if not only, solved by dynamic programming. But dynamic programming suffers from the curse of dimensionality whereby optimization becomes intractable as time horizon and number of assets increase, thereby limiting its practical applications. In this paper I show for a classic multiperiod investment problem that a feed-forward, open-loop procedure, amenable to solution by conventional methods (e.g. calculus of variations) and not subject to the curse of dimensionality, generates 'here and now' portfolios identical to those generated by the dynamic programming approach. The analytic results in this paper demonstrate that for this classic problem a feed forward approach is not inferior to the more common backward induction approach, suggesting that an 'open-loop with recourse' process is a viable closed-loop approach for some practically useful multiperiod investment problems.

First version: September 18, 2020

This version: October 3, 2020

JEL classification: G11, C61, D81.

Keywords: Multi-period optimization, dynamic programming, multi-period portfolio choice

*I thank Ben Inker, Edmund Bellord, James Montier, and Joshua Livnat for useful comments and suggestions.

†GMO, LLC. 40 Rowes Wharf, Boston, MA 02110. martin.tarlie@gmo.com.

1 Introduction

It is conventional wisdom that the best way to solve multiperiod portfolio optimization problems is by dynamic programming. Part of the intuitive appeal of dynamic programming is that it proceeds by backward induction, implying that the solutions anticipate future actions. In this paper I show, by explicit calculation, that for a classic multiperiod portfolio optimization problem a feed-forward approach generates the identical 'here and now' portfolio as dynamic programming. This result demonstrates that solutions that 'anticipate' future actions do not require backward induction, and that the feed-forward approach is a viable alternative to dynamic programming for some practical problems.

In a seminal paper on lifetime portfolio selection, Merton (1969) applied dynamic programming to the multiperiod portfolio selection problem and paved the way for much subsequent research. The notion that dynamic programming is not only the best, but perhaps the only, approach is succinctly captured in the statement in Eeckhoudt et al. (2011, Section 7.1) that "Solving dynamic decision problems *requires* [emphasis added] an understanding of ... 'backward induction'."

The process of backward induction, whereby a sequence of single period optimizations, each of which is solved for *all* allowable states, is at least partially responsible for the intuitive appeal of the dynamic programming approach. Backward induction derives from the principle of optimality, which essentially states that regardless of the initial policy, remaining decisions must be optimal (Bellman (2003, Chapter 3.3)). Furthermore, the principle of optimality is tied to the notion of time consistency, generally considered an attractive characteristic (see Bjork and Murgoci (2010, Section 1.1)).

While the dynamic programming approach has intuitive appeal, this intuition can come with a significant practical cost. The methodology suffers from the 'curse of dimensionality' whereby optimization becomes exponentially more difficult as time horizon and number of assets increase.[1] This curse of dimensionality is significant because many practical problems involve multiple assets over long horizons – the retirement problem is an obvious example (see e.g. Tarlie (2017)).

A common alternative to the dynamic programming approach is Model Predictive Control (MPC).[2] Herzog et al. (2006) and Calafiore (2008) apply a stochastic version of MPC to portfolio selection. The basic idea is to solve the so-called open-loop problem for the

[1] See the Preface in Bellman (2003) for a general discussion, and the Introduction in Calafiore (2009) for a portfolio optimization discussion.

[2] MPC is typically applied to non-stochastic problems. See, for example, Bertsekas (2017, Section 6.4.3). In the context of portfolio selection, Boyd et al. (2014) define MPC in terms of certainty equivalence. The open-loop feedback control approach in Bertsekas (2017, Section 6.4.4) is a close cousin of MPC but applies to stochastic problems.

optimal control path, extract the 'here and now' control from the full path, and then repeat the process as time evolves and new information becomes available. Calafiore (2008) characterizes this process as 'open-loop with recourse', which if applied in a controlled manner, is closed-loop.[3] As long as next period's optimization can be computed in a timely manner so that the new optimum is available when needed, 'open-loop with recourse' appears to be a viable closed-loop approach.

The advantage of the 'open-loop with recourse' procedure is that it avoids the curse of dimensionality that plagues dynamic programming. The apparent disadvantage is that the process is generally considered to be suboptimal. Herzog et al. (2006), for example, show that the value function for the closed-loop dynamic programming approach is greater than or equal to the value function for the open-loop approach (see also Bertsekas (2017, Sections 6.4.3 and 6.4.4)). But this comparison is unfair in the sense that the information set for evaluating the dynamic programming value function covers the entire investment horizon, whereas the information set for evaluating the open-loop value function only includes the current time and is therefore a subset of the information set used to evaluate the dynamic programming value function. It is not a surprise that such a comparison would favor the dynamic programming approach as more information, properly applied, should lead to better outcomes. However, if for some problems the open-loop methodology generates the same 'here and now' portfolio as dynamic programming, this portfolio is optimal for the simple reason that it is the same portfolio as that generated by dynamic programming.

From the practical perspective of a portfolio manager, what matters today is the portfolio to own today. And from an optimization perspective, the pertinent question is whether or not today's portfolio – the 'here and now' portfolio – is optimal. The fact that the open-loop approach, based only on the 'here and now' information set, generates a full path of portfolios that are deemed suboptimal when compared to a process that includes the information sets that cover the *entire* horizon, does not necessarily mean that the specific 'here and now' portfolio generated by the open-loop approach is suboptimal. Furthermore, if we follow 'open-loop with recourse', we will never own any of these suboptimal post-'here and now' portfolios. As a practical matter these post-'here and now' portfolios are not relevant because the next portfolio we own will be the 'here and now' portfolio that is updated to reflect the then current 'here and now' information set.

In this paper I study two, related, multiperiod problems for which the dynamic

[3] The closed-loop and open-loop terminology relates to the characteristics of the system diagrams. For closed-loop processes, the output feeds back to the input, illustrated in the system diagram by an arrow emanating from the output and feeding back to the input. For open-loop processes there is no such feedback and all arrows point in one direction. In this paper, I apply the stochastic version of MPC, not the certainty equivalent version.

programming solutions are known analytically. I solve these problems in a feed-forward (open-loop) manner by (i) propagating forward the dynamic constraint that defines state variable dynamics, (ii) computing the expected value of the utility function, conditional on the 'here and now' information set, and (iii) optimizing the resulting deterministic objective function, and show that the resulting 'here and now' solutions are identical to the dynamic programming solutions. Given that the problem setup for the feed-forward and backward induction approaches are exactly equivalent, and that the 'here and now' portfolios are also exactly equivalent, simple logic implies that for these problems the 'open-loop with recourse' approach inherits all of the desirable optimality and time consistency properties of the dynamic programming 'here and now' solutions.

To help visualize the contrast between the feed-forward and backward induction approaches, it is useful to imagine the state variables evolving according to a binomial tree. Each node of the tree at a particular level represents an accessible state at a particular time. Dynamic programming proceeds by starting at the next to last level (e.g. $T-1$), and finds the optimal weight at each node. The process then proceeds by moving backward in time, repeating the optimization for all nodes at each level, until finally ending up at the first node. By contrast, the feed-forward approach proceeds by starting at the first node and propagating forward the dynamic constraint (this is effectively the process that generates the *entire* tree), and then computing the objective function, i.e. expected utility.[4] Because both problems utilize the same tree and the same utility function, they are solving the same problem for the portfolio weights at the first node (the 'here and now'). We will see in the feed-forward approach that the details of the forward propagation of the dynamic constraint, and in particular assumptions about the relationship between the state variable and the control variable, are crucial.

I begin in Section 2 with a simple toy model. This model is interesting for two reasons. First, it illustrates the main points that arise in the practically more interesting portfolio problem, but without the mathematical complexities. Second, this toy model has some historical significance as it appears in Chapter 7 of Dreyfus (1965), the reference and chapter cited in Merton (1969) as the basis for dynamic programming.

In Section 3 I revisit a classic problem in multiperiod portfolio optimization, viz. maximize expected utility of terminal wealth for the case of two assets, one risk-free and the other risky with a mean-reverting expected return. I show by explicit calculation that the 'here and now' portfolio generated using the feed-forward open-loop methodology is

[4]Note the contrast with certainty equivalent MPC, which I do not follow in this paper. In certainty equivalent MPC, the expected value of the state variables is propagated forward, rather than the full stochastic process. This means that the certainty equivalent MPC does not explore all of the nodes of the tree, whereas stochastic MPC does.

exactly equivalent to the corresponding 'here and now' portfolio generated using dynamic programming. In Section 4 I discuss some conceptual differences between the dynamic programming and 'open-loop with recourse' approaches. Section 5 concludes.

2 A toy model

Tracing back the conventional wisdom that dynamic programming is the approach of choice for multiperiod portfolio optimization leads to the companion papers Merton (1969) and Samuelson (1969).[5] Merton sets up the problem using the Bellman principle of optimality, citing Chapter 7 of Dreyfus (1965) in footnote 2. In Section 15 of Chapter 7, Dreyfus studies a simple model and shows in Section 16 that the open-loop solution is different from the dynamic programming solution. Because much of Dreyfus' book contrasts the calculus of variations with dynamic programming, one is left with the impression that it is the dynamic programming approach that drives the difference between the two results.

In this section I show that the feed-forward approach that proceeds by (i) assuming a feedback form for the control variable rather than Dreyfus' implicit assumption that the control variable is independent of the state variable, (ii) integrating forward the dynamic constraint, and (iii) optimizing the resulting objective function using the calculus of variations, generates a 'here and now' solution that is identical to the 'here and now' solution generated by dynamic programming. This means that the difference between the dynamic programming solution and Dreyfus' open-loop solution is not driven by the difference between dynamic programming and open-loop per se, but rather by the implicit assumption about the relationship, or lack thereof, between the control variable and the state variable in the derivation of the open-loop solution.

2.1 The model and Dreyfus' solutions

The problem that Dreyfus studies in Chapter 7, Section 15 is

$$\max_{u(s)} \mathrm{E}_t[\xi^2(T)] \tag{1}$$

where the state variable ξ evolves according to

$$d\xi(s) = u(s)\, b\, ds + u(s)\, \sigma dB(s) \tag{2}$$

[5]Merton (1969) and Samuelson (1969) are companion papers, published at the same time in the same journal. Merton (1969) focuses on a continuous time formulation, and Samuelson (1969) on a discrete time formulation.

over times s from t to T, with initial condition $\xi(t) = x$. The quantity $u(s)$ is the control variable, b and σ are constants, and $dB(s)$ is a standard Wiener increment with mean zero and variance ds. The t subscript on the expectation operator emphasizes that the objective function depends on information only known at time t – the 'here and now.' Applying the dynamic programming methodology to this problem leads to the optimal closed-loop solution for the control variable (see (15.12) in Dreyfus (1965))

$$u = -\frac{b}{\sigma^2}\,x. \tag{3}$$

In Chapter 7, Section 16 Dreyfus contrasts the closed-loop solution (3) with an open-loop solution that he quotes, but does not derive. One way to derive Dreyfus' open-loop solution is to integrate (2) from t to T and compute the expectation in (1) explicitly. This process leads to the following formulation:

$$\max_{u(s)} \Phi_t[u] \tag{4}$$

where

$$\Phi_t[u] = x^2 + b^2 \left(\int_t^T u(s)\,ds \right)^2 + \sigma^2 \int_t^T u^2(s)\,ds + 2bx \int_t^T u(s)\,ds. \tag{5}$$

The t subscript on $\Phi_t[u]$ emphasizes that the objective function is based on information only known at time t and the square brackets emphasize that $\Phi_t[u]$ is a functional of the path $u(s)$ for $s \in [t, T]$. Notice that the objective function depends on $x = \xi(t)$, the initial value of the state variable, but does not depend on $\xi(s)$ for $s > t$ because these variables integrate out when we compute the conditional expectation in (1) after integrating (2) from t to T.

The form of the problem given in (4) and (5) is amenable to solution by the calculus of variations.[6] It is straightforward to solve the first order condition, with the result that

$$u_t(s) = -\frac{b}{\sigma^2 + b^2(T-t)}\,x, \tag{6}$$

which is exactly expression (16.1) in Chapter 7, Section 16 of Dreyfus (1965) – Dreyfus' quoted open-loop result.

Clearly, the dynamic programming solution (3) and Dreyfus' open-loop solution (6) are different. But the problem with the solution in (6) is not that we followed a feed-forward,

[6]The first order condition generates an integral equation. The structure of this equation is similar to the integral equation that arises in Section 3.

open-loop approach, i.e. we solved for the full path $u_t(s)$ by integrating the state equation, computing the expected utility, and then optimizing using the conventional calculus of variations to find the optimal control. Rather, the problem is that when we integrated (2) we assumed the optimal control to be independent of the state variable. The fact that the optimal solution (6) depends on x, the initial value of the state variable, is inconsistent with this assumption and is an indication that herein lies the problem.

2.2 An open-loop solution using feedback form

To test the hypothesis that the problem lies with the assumption that the control variable is independent of the state variable, let us assume a control of *feedback* form, viz.

$$u(s) = \alpha(s)\xi(s), \tag{7}$$

where $\alpha(s)$ is now the control variable of interest. If we insert (7) into (2) and integrate from t to T to generate $\xi(T)$, then we can compute the expectation $\mathrm{E}_t\left[\xi^2(T)\right]$ explicitly. The resulting optimization problem is given by

$$\max_{\alpha(s)} e^{\Phi_t[\alpha]} \tag{8}$$

where

$$\Phi_t[\alpha] = \int_t^T \left(2b\,\alpha(s) + \sigma^2\alpha^2(s)\right) ds, \tag{9}$$

and I have dropped the factor of x^2 multiplying the exponential because it does not depend on $\alpha(s)$. Using the calculus of variations to generate the first order condition, it is easy to see that the first order condition is satisfied by

$$\alpha_t(s) = -\frac{b}{\sigma^2}, \tag{10}$$

where, once again, the t subscript indicates that the optimal path is conditioned on time-t information. Combining this optimal path, which happens to be a constant for all s, with the feedback form (7) implies that for $s = t$, i.e. the 'here and now,' we have

$$u_t(t) = -\frac{b}{\sigma^2}\,x, \tag{11}$$

which is exactly the dynamic programming solution given in (3).

This example shows, by explicit calculation, that the difference between Dreyfus' dynamic programming and open-loop 'here and now' solutions lies in the assumption that the control variable and the state variable are independent, not in the open-loop methodology of propagating the state equation forward, computing the expected value to generate an objective function, and then using the calculus of variations to find the optimal path. If instead of the assumption that the state and control variables are independent, we follow the same open-loop process but assume a feedback form for the optimal control, we then generate an optimal path that coincides, at $s = t$, with the optimal path generated by dynamic programming. The notion that the two approaches only generate solutions that coincide for $s = t$, i.e. the 'here and now', is a crucial conceptual point that I discuss in more detail in Section 4.

3 A classic portfolio problem

In this section I apply the ideas from the previous section to a classic multiperiod optimization problem: maximize expected utility of terminal wealth for a risky asset and a risk free asset where the expected return of the risky asset is mean-reverting. Kim and Omberg (1996) solve this problem analytically using dynamic programming. I show in this section that we can generate the same 'here and now' allocation as Kim and Omberg by following a feed-forward open-loop approach of (i) integrating forward the dynamic constraint, (ii) evaluating the expected utility to generate an explicit objective function, and (iii) optimizing the objective function using a standard method, in this case the calculus of variations. The 'here and now' allocation is then simply the initial value of the optimal path.

An important conceptual point that I discuss in more detail in Section 4 is that the full optimal path generated by this open-loop approach has no direct counterpart in the dynamic programming approach. In dynamic programming, backwards induction requires that we solve each member of the sequence of single period problems for all possible values of the state variables. On the other hand, the allocations defined by the open-loop optimal path are conditional on the 'here and now' information set, so it is only the 'here and now' allocations that are directly comparable to the dynamic programming output.

3.1 The model

Let r be the constant annualized return on the risk free asset, and let the price of the risky asset follow

$$\frac{dP(s)}{P(s)} = \mu(s)\,ds + \sigma dB(s), \tag{12}$$

where σ^2 is the variance rate and $dB(s)$ is a standard Wiener increment with mean zero and variance ds. The annualized expected return $\mu(s)$ is given by

$$\mu(s) = r + \sigma X(s) \tag{13}$$

where the premium for risk follows the Ornstein-Uhlenbeck process[7]

$$dX(s) = \eta\left(\bar{X} - X(s)\right)\,ds + \sigma_X dB_X, \tag{14}$$

with $\bar{X}$ the average premium, η the mean reversion speed, σ_X^2 the variance rate, and dB_X another standard Wiener increment with mean zero and variance ds. The contemporaneous correlation between dB and dB_X is ρ_X.

Kim and Omberg (1996) formulate the wealth dynamics in terms of the monetary amount y invested in the risky asset so that wealth evolves as

$$dW(s) = (W(s) - y(s))\,r\,ds + y(s)\frac{dP(s)}{P(s)}. \tag{15}$$

The utility function in Kim and Omberg (1996) is a member of the well-known two parameter HARA family, i.e. power law but with an additional wealth level. Specifically,

$$U = \frac{(W - W_*)^{1-\gamma}}{1-\gamma} \tag{16}$$

where γ is the risk aversion parameter (Kim and Omberg specify the risk tolerance, which is $1/\gamma$), and W_* is the wealth level. If we define $Z(s) = W(s) - W_* e^{r(s-T)}$, then using (15) we can write

$$dZ(s) = (Z(s) - y(s))\,r\,ds + y(s)\frac{dP(s)}{P(s)}. \tag{17}$$

Motivated by the feedback form of the previous section, let us write the control variable

[7] An Ornstein-Uhlenbeck process is the continuous time equivalent of a discrete time AR(1) process.

in feedback form, i.e. $y = \alpha Z$.[8] Dividing both sides of (17) by $Z(s)$, we can use (12)-(14) to write the evolution equation as

$$\frac{dZ(s)}{Z(s)} = r\,ds + \alpha(s)\,\sigma X(s)\,ds + \alpha(s)\,\sigma dB(s). \tag{18}$$

Following the prescription in the previous section, if we integrate (18) forward from t to T and explicitly compute the (conditional) expected utility $\mathrm{E}_t[U(Z(T))]$, then we can formulate the problem as

$$\max_{\alpha(s)} e^{\Phi_t[\alpha]} \tag{19}$$

where $\Phi_t[\alpha]$ is a functional of $\alpha(s)$, detailed explicitly in (A.2), and I have dropped the factor $Z(t)^{1-\gamma}/(1-\gamma)$ multiplying the exponential because it does not depend on $\alpha(s)$.

3.2 The optimal portfolio path

As in the previous section, we can find the optimal portfolio by solving the first order condition for $\Phi_t[\alpha]$. As detailed in Appendix A.1, we can write the first order condition for the optimal control as

$$\alpha_t(s) = f(s) + \lambda \int_t^s K(s,s')\alpha_t(s')\,ds', \tag{20}$$

where λ is a constant that depends on the dynamic properties of the risk premium and on the investor's risk aversion γ. The kernel $K(s,s')$ is anti-symmetric in the arguments s and s' and depends on the mean reversion speed η. Equation (20) for the optimal path $\alpha_t(s)$ is a linear Fredholm integral equation of the second kind, a ubiquitous, well-studied problem (see Tricomi (1985)).

I show in Appendix A that we can reduce the integral equation (20) to a second order ordinary differential equation with constant coefficients, which is easy to solve in closed form. The solution is a path $\alpha_t(s)$ defined for all times $s \in [t,T]$, conditional on the information set at time t. As a practical matter, however, the most important part of the optimal path corresponds to $s = t$, the 'here and now'; in Section 4 I discuss the interpretation of $\alpha_t(s)$

[8] Note that feedback form for $W_* = 0$ is nothing more than defining the control variables to be portfolio weights ($\alpha(s)$) rather than portfolio monetary values ($y(s)$).

for $s > t$. The solution for $s = t$ is given by (see (A.10))

$$\alpha_t(t) = \frac{X(t)}{\gamma\sigma} + \left\{ \frac{\frac{X(t)}{\gamma\sigma}\left(1 - e^{-2\theta\tau}\right) + \frac{\bar{X}}{\gamma\sigma}\frac{\eta}{\theta}\left(1 - e^{-\theta\tau}\right)^2}{2\theta - [(\gamma^{-1} - 1)\rho_X\sigma_X - \eta + \theta]\left(1 - e^{-2\theta\tau}\right)} \right\} \left(\gamma^{-1} - 1\right)\rho_X\sigma_X, \qquad (21)$$

where $\theta = \sqrt{\eta^2 - 2\eta\lambda}$ and $\tau = T - t$. In this expression we see the familiar decomposition of the solution into the myopic term – the first term – and the hedging demand term – the second term, which is characterized by the mean-reverting properties of the risk premium and the investor's risk aversion.

I show in Appendix A.3 that the solution given in (21) coincides exactly with the 'here and now' dynamic programming solution in Kim and Omberg (1996).[9] Because this solution matches the dynamic programming one, we see that in this example the open-loop procedure generates an optimal 'here and now' allocation. As long as we continuously update the 'here and now' portfolio by following the process of 'open-loop with recourse', this procedure will generate a sequence of portfolios identical to that generated by dynamic programming. By simple logic, this means that for this problem the 'open-loop with recourse' process inherits all of the optimality and time consistency characteristics of dynamic programming.

4 Discussion

The key takeaway from the two, related problems presented in Sections 2 and 3 is that for these problems a properly formulated feed-forward open-loop problem generates the same 'here and now' solution as dynamic programming, a process that proceeds by backward induction. Because the 'here and now solutions' are identical, it follows that the feed-forward 'here and now' solutions inherit the desirable optimality and time consistency attributes of the dynamic programming 'here and now' solutions. But what about the rest of the open-loop optimal path, i.e. what is the interpretation of the optimal allocations for the future beyond the 'here and now'?

Recall that the open-loop process I followed proceeds in three steps: (i) integrate forward the dynamic constraint using initial conditions at time t, fully incorporating the stochastic nature of the dynamic constraint, and paying particular attention to the relationship between the control and state variables, (ii) evaluate the expected utility, conditioned on time t information, to generate an explicit, deterministic objective function, and (iii) optimize the deterministic objective function using conventional methods. This last step generates an

[9]Kim and Omberg present four solutions. I show in Appendix A.3 that for $s = t$, all four solutions are nested in (21).

optimal allocation $\alpha_t(s)$ for all times s between t and T, where the t subscript on α reminds us that the optimal allocations are only based on information known at time t – the 'here and now.' This means that the allocations for times s beyond the 'here and now' are only optimal conditional on information known at time t.[10]

The future allocations $\alpha_t(s)$, for $s > t$, have no direct counterpart in the dynamic programming framework. To see this, recall that dynamic programming operates by backward induction. Conceptually, the process proceeds by starting at the end and solving a sequence of single period optimization problems. But because the state of the system is not known at the future times, the optimization must be done for all admissible state variables. This means that the dynamic program generates a mapping between the admissible state variables (i.e. those that satisfy the constraints) and the optimal control variables (e.g. portfolio weights). For the Kim-Omberg problem, the domain of the mapping is the entire horizon from t to T, as well as all allowable wealth (W) and risk premium (X) values over this horizon, so we can write the optimal dynamic programming allocation as $\alpha(s, W, X)$. Solving the the Hamilton-Jacobi-Bellman equation amounts to solving for this mapping.[11] Generating this mapping generally requires a great deal of effort, especially as the state space gets larger – this is the origin of the 'curse of dimensionality.'

By contrast, the domain of the optimal path generated by the open-loop methodology is, again, the entire horizon from t to T, but only the wealth and state variables at time t, not the wealth and state variables over the entire horizon. In this sense, the open-loop process generates a specific curve $\alpha_t(s)$ through time, where the t subscript indicates dependence on the time-t information set, in this case $W(t)$ and $X(t)$. Even though the optimal path does not depend on the state variables for $s > t$, it *is* sensitive to the stochastic dynamics of the state variables over the entire horizon because the objective function is the result of integrating forward the stochastic dynamic equation and then computing the conditional expected utility.

It is only when backwards induction reaches time t – the 'here and now' – that the information set becomes known and we can evaluate the dynamic program mapping explicitly to generate the allocation at time t. It is at this point that the results of the dynamic programming approach and the open-loop approach coincide. If we consistently update the optimal 'here and now' allocation as time and the associated information set evolves, i.e. we follow the 'open-loop with recourse' approach, the optimal allocations will always match those of the dynamic programming approach. Because the updated information set can

[10] These are only the portfolios one would own if isolated (e.g. on a deserted island) with no way of receiving new information and modifying the portfolio to reflect this new information.

[11] See Dreyfus (1965, Chapter 7, Section 12) for further elaboration of this point.

include the output from the previous period, the 'open-loop with recourse' methodology is a viable closed-loop approach.

An implication of the result that 'open-loop with recourse' generates the same 'here and now' portfolio as dynamic programming is that time consistency does not require backward induction. Because the 'here and now' open-loop solutions to the two problems in this paper are identical to the 'here and now' dynamic programming solutions, they inherit dynamic programming's time consistency. But the open-loop process is feed-forward in the sense that the dynamic constraint is integrated forward, starting at the 'here and now' and ending at the terminal point. Yet this feed-forward process generates a time-consistent 'here and now' portfolio, illustrating by example that time consistency does not require backward induction.

5 Conclusion

While it is conventional wisdom that the best, and perhaps only, way to solve multiperiod portfolio optimization problems is by dynamic programming, the limitations, arising in part from the curse of dimensionality, cry out for alternatives. As a possible alternative, the open-loop approach suffers from the perception that it generates sub-optimal solutions relative to dynamic programming. However, the results in Section 2 and Section 3 show that for these problems, an open-loop procedure using a feedback form for the optimal control generates optimal 'here and now' controls that exactly match well-known results obtained using the dynamic programming approach. The conceptual implication is that for portfolio problems amenable to 'open-loop with recourse', the 'here and now' portfolios that are updated to reflect the flow of information are *not* suboptimal relative to dynamic programming for the simple reason that the 'here and now' portfolios are the same in both cases.

The advantage of the open-loop approach employed here is that the procedure – propagate the stochastic dynamic forward and compute the expected utility to generate a deterministic objective function – generates an objective function that we can solve using conventional techniques (e.g. calculus of variations for problems amenable to analytic calculations, or numerical methods more generally). Because this approach does not suffer from the curse of dimensionality, it is a viable method for some practically useful problems that involve multiple assets over long horizons.

An example of such a problem, which generalizes the terminal wealth problem studied in Section 3, is given in Tarlie (2017), but with two key differences. First, there is a wealth target at the investment horizon (e.g. an investor's target nest egg). Second, there are asymmetric preferences, i.e. different levels of risk aversion above and below the target, reflecting the intuition that the investor likely cares more about falling short of the target than exceeding it. When applied to life cycle investing, this problem involves multiple assets over long horizons, making a backward induction solution impractical. However, since we can compute the expected utility in closed form in a manner analogous to the process used Section 3, we can optimize the resulting objective function using conventional numerical methods.

While I offer no general proof in this paper that feed-forward ‘open-loop with recourse’ always generates the same ‘here and now’ solution as dynamic programming – after all we see in the analysis of the Dreyfus problem in Section 2 that the details of forward propagation matter – there are many problems that are not amenable to dynamic programming.[12] The success of the ‘open-loop with recourse’ method in replicating the dynamic programming result in Kim and Omberg (1996) suggests that the present approach offers a viable alternative, even in the absence of proof that the solution is optimal. An interesting line of future research is to better understand the scope and limitations of this methodology. Hopefully the present results will motivate this research.

[12] See Bjork and Murgoci (2010).

A Calculation details for the classic portfolio problem

In this Appendix I sketch out the details of how to setup and solve the classic multiperiod optimization problem in Section 3. A central concept is that by integrating the state equation forward and computing the conditional expectation of the utility function, we generate a first order condition in the form of an integral equation. This is in contrast to the dynamic programming approach in which backward induction generates a partial differential equation. We will see below that we can reduce the integral equation to a second order ordinary differential equation with constant coefficients, an equation which is easy to solve in closed form.

A.1 Basic equations and results

The expression for Φ in (19) follows by direct calculation. The first step is to integrate the state equation for $Z(s) = W(s) - W_* e^{r(s-T)}$ given in (18) over s from t to T. The second step is to compute the expected utility over the distribution of $Z(T)$, conditioned on time t information. The result is

$$\mathrm{E}_t[U(Z(T))] = \mathrm{E}_t\left[\frac{Z(T)^{1-\gamma}}{1-\gamma}\right] = \left\{\frac{Z(t)^{1-\gamma}}{1-\gamma}\right\} e^{\Phi_t[\alpha]} \tag{A.1}$$

where

$$\begin{aligned}\Phi_t[\alpha] &= (1-\gamma)\, r\,(T-t) + (1-\gamma)\int_t^T ds\, \left(\sigma\bar{X}(s)\alpha(s) - \frac{\sigma^2}{2}\alpha^2(s)\right) \\ &+ \frac{\sigma^2(1-\gamma)^2}{2}\left\{\int_t^T ds\,\alpha^2(s) + \left(\frac{\sigma_X^2}{\eta} + 2\sigma_X\rho_X\right)\int_t^T ds\, e^{-\eta(s-t)}\alpha(s)\int_t^s ds'\, e^{\eta(s'-t)}\alpha(s')\right. \\ &\left. - \frac{\sigma_X^2}{\eta}\int_t^T ds\, e^{-\eta(s-t)}\alpha(s)\int_t^s ds'\, e^{-\eta(s'-t)}\alpha(s')\right\}.\end{aligned} \tag{A.2}$$

Maximizing expected utility is equivalent to maximizing $\Phi_t[\alpha]$ over $\alpha(s)$ for $s \in [t, T]$. Because $\Phi_t[\alpha]$ is deterministic we can use the calculus of variations to write the first order condition for Φ as

$$\alpha_t(s) = f(s) + \lambda\int_t^s K(s, s')\alpha(s')\, ds'. \tag{A.3}$$

This equation is known as a Fredholm integral equation of the second kind.[13] The subscript t on $\alpha_t(s)$ emphasizes that the optimal path depends on the information set at time t. In

[13] See Tricomi (1985), Chapter 1.

(A.3), the kernel K is given by

$$K(s,s') = e^{-\eta(s-t)}e^{\eta(s'-t)} - e^{\eta(s-t)}e^{-\eta(s'-t)}, \tag{A.4}$$

and the coefficient λ is given by

$$\lambda = (\gamma^{-1}-1)\,\frac{\eta}{2}\,k_* \tag{A.5}$$

with $k_* = (\sigma_X^2 + 2\eta\rho_X\sigma_X)/\eta^2$, the derived parameter given in equation (9) in Kim and Omberg (1996). The inhomogeneous term in (A.3) is given by

$$f(s) = \frac{\bar{X}(s)}{\gamma\sigma} + Cg(s) \tag{A.6}$$

where $\bar{X}(s) = \bar{X} + \left(X(t)-\bar{X}\right)e^{-\eta(s-t)}$,

$$C = \int_t^T e^{-\eta(s-t)}\alpha_t(s)\,ds \tag{A.7}$$

is the exponentially weighted sum of $\alpha_t(s)$ over the entire horizon, and

$$g(s) = (\gamma^{-1}-1)\left(\frac{\sigma_X^2}{\eta}\sinh(\eta(s-t)) + \rho_X\sigma_X e^{\eta(s-t)}\right). \tag{A.8}$$

As a practical matter, the primary quantity of interest is the allocation to the risky asset at time t, i.e. $\alpha_t(t)$. This is the 'here and now' allocation. From (A.3) we have that $\alpha_t(t) = f(t)$ which, using (A.6) and (A.8), we can write as

$$\alpha_t(t) = \frac{X(t)}{\gamma\sigma} + C\left(\gamma^{-1}-1\right)\rho_X\sigma_X. \tag{A.9}$$

The first term on the right hand side is the standard myopic allocation, and the second term is the so-called hedging term, which depends on the constant C.

The following section, Appendix A.2, contains the details of how to calculate C, with the result that we can write the optimal here and now' allocation to the risky asset as

$$\alpha_t(t) = \frac{X(t)}{\gamma\sigma} + \left\{\frac{\frac{X(t)}{\gamma\sigma}\left(1-e^{-2\theta\tau}\right) + \frac{\bar{X}}{\gamma\sigma}\,\frac{\eta}{\theta}\,\left(1-e^{-\theta\tau}\right)^2}{2\theta - \left[(\gamma^{-1}-1)\rho_X\sigma_X - \eta + \theta\right]\left(1-e^{-2\theta\tau}\right)}\right\}\left(\gamma^{-1}-1\right)\rho_X\sigma_X, \tag{A.10}$$

where $\theta = \sqrt{\eta^2 - 2\eta\lambda}$ (see (A.12)). In Appendix A.3 I show that this expression nests all four of Kim and Omberg's dynamic programming solutions.

A.2 Calculating C

As is evident from (A.7), to calculate C we need to know the optimal path over the entire horizon. To solve (A.3) for $\alpha(s)$, we can exploit the properties of the kernel K, noting that the second partial derivative of $K(s, s')$ with respect to s is proportional to K. Differentiating both sides of (A.3) twice with respect to s leads to

$$\frac{d^2\alpha(s)}{ds^2} - \theta^2\alpha(s) + \theta^2 b = 0, \tag{A.11}$$

where

$$\theta^2 = \eta^2 - 2\eta\lambda, \tag{A.12}$$

and

$$\theta^2 b = \eta^2 \frac{\bar{X}}{\gamma\sigma}. \tag{A.13}$$

We can write the formal solution to this linear, inhomogeneous second order differential equation as

$$\alpha_t(s) = b + Ae^{\theta(s-t)} + Be^{-\theta(s-t)}, \tag{A.14}$$

where the constants (i.e. independent of s) A and B are given by

$$A = \frac{f(t) - b + \frac{1}{\theta}\frac{df(t)}{ds}}{2} \tag{A.15}$$

$$B = \frac{f(t) - b - \frac{1}{\theta}\frac{df(t)}{ds}}{2}. \tag{A.16}$$

But we are not quite done yet, because these expressions for A and B are only implicit as they contain dependence on C, which in turn depends on A and B (see (A.6) and (A.7)).

To proceed further, insert the expression for $\alpha_t(s)$ from (A.14) into (A.7) for C and perform the integration, resulting in

$$C = b\frac{1 - e^{-\eta\tau}}{\eta} + A\frac{1 - e^{-(\eta-\theta)\tau}}{\eta - \theta} + B\frac{1 - e^{-(\eta+\theta)\tau}}{\eta + \theta}, \tag{A.17}$$

where $\tau = T - t$. If we further observe, after some tedious but straightforward algebra, that

$$C = \frac{b}{\eta} + \frac{A}{\eta - \theta} + \frac{B}{\eta + \theta}, \tag{A.18}$$

then we can write (A.17) as

$$0 = \frac{b}{\eta} + A\,\frac{e^{\theta\tau}}{\eta - \theta} + B\,\frac{e^{-\theta\tau}}{\eta + \theta}. \tag{A.19}$$

From this expression we can finally isolate C, which again after some tedious but straightforward algebra, is given by

$$C = \frac{\frac{X(t)}{\gamma\sigma}\left(1 - e^{-2\theta\tau}\right) + \frac{\bar{X}}{\gamma\sigma}\,\frac{\eta}{\theta}\,\left(1 - e^{-\theta\tau}\right)^2}{2\theta - \left[(\gamma^{-1} - 1)\rho_X\sigma_X - \eta + \theta\right]\left(1 - e^{-2\theta\tau}\right)}. \tag{A.20}$$

A.3 Reconciling the Kim-Omberg results

In this section I detail the steps required to show that the optimal allocation at time t obtained using dynamic programming in Kim and Omberg (1996) is identical to the result given in (A.10).

The notation in Kim and Omberg (1996) is slightly different than the notation in this paper. The following translations relate the notation in the Kim-Omberg formulas (left of the arrow) to those in this paper (right of the arrow):

$$\gamma \to \gamma^{-1} \tag{A.21}$$

$$\lambda_X \to \eta \tag{A.22}$$

$$\rho_{mX} \to \rho_X \tag{A.23}$$

$$\eta = \sqrt{q} = \sqrt{4\lambda_X^2[1 - (\gamma - 1)k_*]} \to 2\theta \tag{A.24}$$

$$b = 2\left[(\gamma - 1)\rho_{mX}\sigma_X - \lambda_X\right] \to 2\left(\gamma^{-1} - 1\right)\rho_X\sigma_X - 2\eta \tag{A.25}$$

The optimal allocation given in (19) of Kim and Omberg (1996) is

$$\frac{y_*}{Z} = \frac{X(t)}{\gamma\sigma} + \left(C_{KO}(\tau)X + B_{KO}(\tau)\right)\frac{\rho_X\sigma_X}{\gamma\sigma}, \tag{A.26}$$

where $Z = W - W_* e^{-r\tau}$ and $\tau = T - t$.

The Appendix in Kim and Omberg (1996) contains four solutions. For the time t allocation, all of these solutions are nested within $\alpha_t(t)$ given by (A.10). For Kim and Omberg's normal solution, the coefficients B_{KO} and C_{KO} as detailed in their Appendix and using their notation are given by

$$C_{KO}(\tau) = \frac{2(\gamma - 1)\left(1 - e^{-\eta\tau}\right)}{2\eta - (b + \eta)\left(1 - e^{-\eta\tau}\right)} \tag{A.27}$$

$$B_{KO}(\tau) = \frac{4(\gamma - 1)\lambda_X \bar{X}\left(1 - e^{-\eta\tau/2}\right)^2}{\eta\left\{2\eta - (b + \eta)\left(1 - e^{-\eta\tau}\right)\right\}}. \tag{A.28}$$

Translating these formulas using the above relations into the notation of this paper, we have

$$C_{KO}(\tau) \to \frac{\left(\gamma^{-1} - 1\right)\left(1 - e^{-2\theta\tau}\right)}{2\theta - \left[\left(\gamma^{-1} - 1\right)\rho_X \sigma_X - \eta + \theta\right]\left(1 - e^{-2\theta\tau}\right)} \tag{A.29}$$

$$B_{KO}(\tau) \to \frac{(\gamma^{-1} - 1)\eta\bar{X}\left(1 - e^{-\theta\tau}\right)^2}{\theta\left\{2\theta - \left[\left(\gamma^{-1} - 1\right)\rho_X \sigma_X - \eta + \theta\right]\left(1 - e^{-2\theta\tau}\right)\right\}} \tag{A.30}$$

It is straightforward to see that if we insert these expressions for C_{KO} and B_{KO} into (A.26), we recover exactly the 'here and now' allocation given in (A.10).

It is also straightforward to check consistency with the other three Kim and Omberg solutions. The hyperbolic and polynomial solutions can be checked by taking the limit that θ goes to zero in (A.20). The tangent solution corresponds to the condition that $\theta^2 < 0$, which simply means that θ is a purely complex number rather than a real number. Nevertheless, C, and hence the solution, is still real. Finally, nirvana solutions exist if there exists a finite, real-valued critical horizon τ_c defined by the condition that the denominator of C is zero, i.e.

$$\tau_c = \frac{1}{2\theta} \ln \frac{\left(\gamma^{-1} - 1\right)\rho_X \sigma_X - \eta + \theta}{\left(\gamma^{-1} - 1\right)\rho_X \sigma_X - \eta - \theta}. \tag{A.31}$$

Nirvana solutions exist for $\tau \in [0, \tau_c]$ if the right hand side of (A.31) is real and finite.

References

Bellman, R., 2003. Dynamic programming. Dover Publications.

Bertsekas, D. P., 2017. Dynamic programming and optimal control, Vol. I. Athena Scientific.

Bjork, T., Murgoci, A., 2010. *A general theory of Markovian time inconsistent stochastic control problems*. Available at SSRN 1694759 .

Boyd, S. P., Mueller, M. T., O'Donoghue, B., Wang, Y., et al., 2014. Performance bounds and suboptimal policies for multi-period investment. Now Publishers.

Calafiore, G. C., 2008. *Multi-period portfolio optimization with linear control policies*. Automatica 44, 2463–2473.

Calafiore, G. C., 2009. *An affine control method for optimal dynamic asset allocation with transaction costs*. SIAM Journal on Control and Optimization 48, 2254–2274.

Dreyfus, S. E., 1965. Dynamic programming and the calculus of variations. Academic Press.

Eeckhoudt, L., Gollier, C., Schlesinger, H., 2011. Economic and financial decisions under risk. Princeton University Press.

Herzog, F., Keel, S., Dondi, G., Schumann, L. M., Geering, H. P., 2006. *Model predictive control for portfolio selection*. In: *American Control Conference, 2006*, IEEE, pp. 8–pp.

Kim, T. S., Omberg, E., 1996. *Dynamic nonmyopic portfolio behavior*. The Review of Financial Studies 9, 141–161.

Merton, R. C., 1969. *Lifetime portfolio selection under uncertainty: The continuous-time case*. The Review of Economics and Statistics pp. 247–257.

Samuelson, P. A., 1969. *Lifetime portfolio selection by dynamic stochastic programming*. The Review of Economics and Statistics pp. 239–246.

Tarlie, M., 2017. *Investment horizon and portfolio selection*. Available at SSRN 2854336 .

Tricomi, F., 1985. Integral equations. Dover Publications.