Operations Research Letters 41 (2013) 494–498

Contents lists available at ScienceDirect

Operations Research Letters

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

Scenario tree generation and multi-asset financial optimization problems

Alois Geyer [a,b], Michael Hanke [c,*], Alex Weissensteiner [d]

[a] WU (Vienna University of Economics and Business), 1190 Vienna, Austria
[b] Vienna Graduate School of Finance, 1190 Vienna, Austria
[c] University of Liechtenstein, Institute for Financial Services, 9490 Vaduz, Liechtenstein
[d] Technical University of Denmark, Department of Management Engineering, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 7 March 2013
Received in revised form
10 June 2013
Accepted 10 June 2013
Available online 19 June 2013

Keywords:
Scenario trees
No-arbitrage
Financial optimization
Moment matching
Scenario reduction

ABSTRACT

We compare two popular scenario tree generation methods in the context of financial optimization: moment matching and scenario reduction. Using a simple problem with a known analytic solution, moment matching – when ensuring absence of arbitrage – replicates this solution precisely. On the other hand, even if the scenario trees generated by scenario reduction are arbitrage-free, the solutions are biased and highly variable. These results hold for correlated and uncorrelated asset returns, as well as for normal and non-normal returns.

1. Introduction

Scenario trees are used in many optimization models as a discrete approximation to a continuous distribution. We consider two approaches which have been applied in the literature: moment matching (see, e.g., [7,6]) and scenario reduction methods (see, e.g., [9,3]). For an overview, see Heitsch and Römisch [4, p. 372f.]. The latter seem appealing for two reasons: first, they explicitly aim at an *optimal* approximation in a sense to be described below. Second, they try to overcome the curse of dimensionality, which frequently arises in multi-stage financial optimization problems: including many time steps and many different assets quickly renders the optimization problem computationally intractable. This makes a method which generates discrete approximations using sparse scenario trees very desirable.

Geyer et al. [2] focus on the applications of stochastic programming in multi-stage financial optimization, and show that scenario reduction may lead to meaningless results if arbitrage opportunities are present in the reduced scenario tree used for optimization. Obviously, arbitrage must be excluded from the scenario tree whenever this is theoretically required (by the subject under investigation). Geyer et al. [2] point out that the principal idea behind scenario reduction implicitly assumes the absence of arbitrage. Our main goal here is to assess the quality of scenario reduction *after* the trees have been checked to be free of arbitrage.

In this paper we focus on the multi-stage and multi-asset financial optimization, and compare moment matching and scenario reduction algorithms. Somewhat surprisingly, numerical comparisons of optimization results based on scenario reduction algorithms to known analytical solutions in this context do not seem to exist in the literature. We confine ourselves to arbitrage-free trees generated by the two approaches and compare the optimal solutions of these approximate problems to the closed-form solution. As a main result we find that moment matching provides highly accurate results, whereas the results from scenario reduction (using the same number of scenarios) may be biased and show very high variance. This applies even if all branching factors in the trees are well above the minimum requirement in order to rule out arbitrage opportunities. Applying a more severe reduction makes the results far worse. Moment matching clearly remains superior for asset returns being correlated or not, as well as for normal, skewed and/or leptokurtic distributions.

The paper is organized as follows: Section 2 briefly sketches the two approaches considered for scenario tree generation, moment matching and scenario reduction. Section 3 discusses ways to ensure the absence of arbitrage in scenario trees. The numerical comparison of the two approaches based on a simple example with a known closed-form solution is presented in Section 4. Section 5 concludes.

* Corresponding author. Tel.: +423 2651155; fax: +423 2651112.
E-mail address: Michael.Hanke@uni.li (M. Hanke).

0167-6377/$ – see front matter
http://dx.doi.org/10.1016/j.orl.2013.06.003

2. Scenario tree generation

Using the notation in [9] and the terminology of multi-asset financial optimization (e.g., portfolio management or asset–liability management), the purpose of scenario generation can be described as follows: given an optimization problem under uncertainty described by a (usually continuous and multivariate) asset return distribution G, we want to generate a scenario tree discretization $\tilde{G}$ such that the objective function $F(x)$ of the original and the objective $\tilde{F}(x)$ of the discretized optimization task are close in some sense. Whereas the original problem is in many cases computationally intractable, the discretized tree representation of the problem is tractable if the tree is sufficiently small (depending on the application, say, on the order of 10^5 or 10^6 scenarios).

One approach to achieve this is to construct the approximated return distribution $\tilde{G}$ to be similar to the original distribution G in the sense that the first few moments of $\tilde{G}$ and G are identical or at least close, and then essentially "hope" that this similarity carries over to the optimization result. This is known as *moment matching* and is described, e.g., in Høyland and Wallace [7], Høyland et al. [6]. Absent further assumptions on the objective function, the approach is rather ad hoc and there is no general theoretical result on the quality of the approximation of the original objective function $F(x)$ by that of the approximated problem, $\tilde{F}(x)$. For reasonably well-behaved objective functions, however, moment matching has been found to work well (see, e.g., [11]), although counterexamples where the approach leads to bad solutions are also known from the literature [5].

Conceptually more appealing and theoretically well-founded, Pflug [9] suggests an approach to define an optimal discretization $\tilde{G}$ in terms of the difference between $F(x)$ and $\tilde{F}(x)$. He shows that the goal of minimizing $\sup_x |F(x) - \tilde{F}(x)|$ (i.e., minimizing the worst-case difference between original and approximated objective function) is equivalent to the minimization of the Wasserstein distance between G and $\tilde{G}$. As the title of Pflug [9] implies, he recommends this approach in particular for *financial* optimization problems. So-called scenario reduction algorithms based on this or conceptually related ideas have been implemented in software modules, e.g., as part of GAMS (see, e.g., [3,4]). Applications using scenario reduction algorithms can be found in, e.g., Bertocchi et al. [1], Hochreiter and Pflug [5], and Rasmussen and Clausen [10].

3. Arbitrage and scenario trees

For optimization models using tradable assets there is an additional requirement from financial theory: arbitrage opportunities must be ruled out in order to arrive at meaningful results (see [8,2]). Therefore, unless scenario trees are guaranteed to be arbitrage-free by construction, they have to be checked for arbitrage opportunities. Only trees which pass this test can be used in the subsequent optimization. A necessary condition for the absence of arbitrage is that the branching factor (i.e., the number of arcs emanating from a node) at each node of the tree must be greater than or equal to the number of non-redundant assets in the optimization problem (see [2], for a more formal exposition). Intuitively, more assets than states provide excess degrees of freedom, which can be exploited to form arbitrage portfolios.

For moment matching, it is straightforward to implement this necessary condition by imposing the minimum branching factor for each node of the tree. However, since the condition is not sufficient, trees constructed in this way may still admit arbitrage opportunities. A very simple approach to ensure the absence of arbitrage is to construct a tree, check it for arbitrage, and discard it if arbitrage opportunities are detected. This procedure is then repeated until an arbitrage-free tree is found (essentially, this combines the ideas of Høyland et al. [6] and Klaassen [8]).

Depending on how aggressively scenario reduction methods are tuned, they may arrive at rather sparse trees. Several authors (see, e.g., [1,5,10]) generate sparse scenario trees without discussing or taking the no-arbitrage requirement into account. In fact, applying the existing implementations of scenario reduction techniques entails a high risk of arriving at scenario trees which admit arbitrage opportunities: as soon as the branching factor for at least one node in the tree is smaller than the number of assets, arbitrage opportunities *must* arise (see [2]). This is complicated by the fact that the existing implementations of scenario reduction algorithms do not allow the user to control the branching factor for each node in the tree, but only the overall tree structure (e.g., six nodes at $t = 1$ and 36 nodes at $t = 2$, but there may well be one node at $t = 1$ with only three successors and another node at $t = 1$ with nine successors).

Transferring the idea in Klaassen [8] to scenario reduction algorithms, one approach to arrive at arbitrage-free trees using scenario reduction is to impose lower bounds on the tree size, which would in principle admit the resulting trees to be free of arbitrage. The generated trees can then be checked for arbitrage opportunities, and this procedure is repeated until a tree passes the no-arbitrage test. This emphasizes once again that "extremely sparse" trees are not compatible with the no-arbitrage condition of many financial optimization models, which requires that the branching factor be greater than or equal to the number of non-redundant assets. For trees with arbitrage opportunities the supremum of the distance between the objective function $F(x)$ of the original and the objective $\tilde{F}(x)$ of the discretized optimization problem, which is required in the derivation of scenario reduction algorithms (see, e.g., [9,3]), does not exist. Moreover, a scenario reduction algorithm which accounts for a minimum branching factor requires a constrained minimization of the Wasserstein distance. Deriving such a constrained solution in closed form does not seem to be a trivial task, and we do not aim to pursue this issue in the present paper. This restriction on the branching factor does not apply to other areas where severe reduction of tree sizes using scenario reduction methods may still be valuable.

4. Numerical example

It is quite common to test newly devised numerical methods using problems with known analytical solutions. Surprisingly, we could not find any results on the accuracy of scenario reduction methods when applied to such prototype problems in the context of financial optimization. We fill this gap by building on and extending a numerical example from Geyer et al. [2, p. 612], where the asset allocation is optimized at two decision stages ($t = 0$ and $t = 1$) in order to maximize the expected log utility of terminal ($T = 2$) wealth for normally distributed and uncorrelated asset returns. Testing both scenario generation methods in this simple, well-known framework allows for a comparison of the numerical results with the correct analytical solution. In a second step, we investigate the stability of both methods when asset returns are correlated and non-normal (i.e., skewed and leptokurtic).

4.1. Base case: uncorrelated, normally distributed asset returns

The investment universe in the base case consists of three normally distributed, uncorrelated risky assets with expected returns of 8% and standard deviations of 25% in each stage. Since the assets have identical properties, the analytical solution to this problem is to allocate 1/3 of available wealth to each asset in each stage. For a starting wealth of 1, the optimal value of the objective function is 0.2.