Alois Geyer and Michael Hanke and Alex Weissensteiner

Scenario Tree Generation and Multi-Asset Financial Optimization Problems

Article (Accepted for Publication)
(Refereed)

Original Citation:
Geyer, Alois and Hanke, Michael and Weissensteiner, Alex (2013) Scenario Tree Generation and Multi-Asset Financial Optimization Problems. *Operations Research Letters*, 41 (5). pp. 494-498. ISSN 0167-6377

Scenario Tree Generation and Multi-Asset Financial Optimization Problems

Alois Geyer
WU (Vienna University of Economics and Business), Austria
Vienna Graduate School of Finance (VGSF)

Michael Hanke
Institute for Financial Services
University of Liechtenstein

Alex Weissensteiner
Dept. of Management Engineering
Technical University of Denmark

June 11, 2013

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, we find that moment matching – accompanied by a check to ensure absence of arbitrage opportunities – replicates this solution precisely. On the other hand, even if the scenario trees generated by scenario reduction are arbitrage-free, the solutions to the approximate optimization problem represented by the reduced tree are biased and highly variable. These results hold for correlated and uncorrelated asset returns, as well as for normal and non-normal returns.

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

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., Høyland and Wallace, 2001; Høyland et al., 2003) and scenario reduction methods (see, e.g., Pflug, 2001; Heitsch and Römisch, 2003).[1] 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. (2010) focus on 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. (2010) 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 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 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 ar-

[1] For an overview, see Heitsch and Römisch (2009, p. 372f.).

bitrage 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.

2 Scenario Tree Generation

Using the notation in Pflug (2001) 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 (2001); Høyland et al. (2003). 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., Topaloglou et al., 2008), although counterexamples where the approach leads to bad solutions are also known from the literature (Hochreiter and Pflug, 2007).

Conceptually more appealing and theoretically well-founded, Pflug (2001) 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 (2001) 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., Heitsch and Römisch, 2003, 2009). Applications using scenario reduction algorithms can be found in, e.g., Bertocchi et al. (2006), Hochreiter and Pflug (2007), and Rasmussen and Clausen (2007).

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 Klaassen, 2002; Geyer et al., 2010). 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 Geyer et al., 2010, 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. (2003) and Klaassen (2002)).

Depending on how aggressively scenario reduction methods are tuned, they may arrive at rather sparse trees. Several authors (see, e.g., Bertocchi et al., 2006; Hochreiter and Pflug, 2007; Rasmussen and Clausen, 2007) generate sparse scenario trees without discussing or taking the no-arbitrage requirement into account. In fact, applying 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 Geyer et al., 2010). This is complicated by the

fact that 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 in the second stage and 36 nodes at stage 3, but there may well be one node at stage 2 with only three successors and another node at stage 2 with nine successors).

Transferring the idea in Klaassen (2002) 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.[2] This 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. (2010, p. 612), where the asset allocation is optimized at two decision stages (t=0 and t=1) in order to maximize 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

[2]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., Pflug, 2001; Heitsch and Römisch, 2003), 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.

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.

For trees based on moment matching, a minimum branching factor of six is required to match the first four moments of the marginal asset return distributions together with the pairwise correlation of zero. Applying a branching factor of six to a two-stage problem yields 36 scenarios (tree structure 1-6-36). To examine the statistical properties of optimization results, we use 1000 trees per method and size of the starting fan (only relevant for scenario reduction). To arrive at these 1000 trees, we repeat the following procedure: Generate trees, check for arbitrage opportunities, and discard those where arbitrage opportunities are present. This sequence is repeated until 1000 arbitrage-free trees are found.

To avoid problems associated with erroneous own implementations we rely on publicly available packages for moment matching as well as scenario reduction. For moment matching we use the Hoyland et al (2003) algorithm as implemented in the executable file provided by one of the authors.[3] The scenario reduction trees are generated using the tree_con algorithm of the SCENRED 2 package which comes with GAMS. With this algorithm, we can only impose the total number of nodes per stage (e.g., 6 nodes in stage 2 and 36 in stage 3), but not the branching factor for each individual successor node. To put both methods on an equal footing, we begin by imposing a tree structure of 1-6-36. This is well above the minimum tree structure required for the absence of arbitrage (1-3-9). tree_con starts with a fan of Monte-Carlo simulated paths, the size of which can be chosen by the user. We examine the impact of the starting fan size on the accuracy of the optimization and on the fraction of arbitrage-free trees for varying the fan size between 250 and 2000 paths. Since trees which allow for arbitrage are discarded, all results

[3] `http://work.michalkaut.net`

are based on 1000 arbitrage-free trees for both methods.

Table 1 shows that optimization using trees generated via moment matching yields highly accurate results (standard deviations are practically zero). In contrast, although scenario reduction leads to unbiased asset allocations, their standard deviations are high.[4] Even for large starting fan sizes, the coefficient of variation (standard deviation divided by the mean) of optimal asset allocations derived via scenario reduction is around 2. Objective function values show an upward bias which decreases slowly as the starting fan size increases.

Existing scenario reduction algorithms do not allow for controlling the branching factor at each node of the tree. This results in different branching factors at stage 2 for this method, whereas the branching factor for moment matching is six for all nodes at this stage. This may cause part of the increased standard deviations we observe for *objective values* from scenario reduction and might be overcome by improvements in future implementations of this method. The higher standard deviation of *asset allocations*, however, cannot be attributed to this shortcoming of existing implementations of scenario reduction: The branching factor at the root node is six for both methods and we are using log utility, which implies myopic solutions. Hence, the initial asset allocation does not depend on the asset allocation at later stages, and the increased standard deviation of the initial asset allocation we observe for scenario reduction cannot be attributed to unequal branching factors at the final stage.

Larger starting fans increase the chance of arriving at arbitrage-free trees: First, because the number of trees discarded due to an insufficient branching factor for at least one node decreases quickly. For a starting fan size of 1000 or 2000 scenarios, no tree generated via scenario reduction fails this first-stage arbitrage test. Second, the number of trees admitting arbitrage despite meeting the minimum branching factor requirement also decreases, as shown in the second-to-last column. Computing time, however, increases markedly for larger starting fans, whereas it is almost the same for starting fan sizes 250 and 500.

Note that our procedure of repeating the tree generation 1000 times only serves to consider bias and standard deviation of the optimization results. In practice, the procedure is meant to be applied only once. Figure 1 illustrates

[4]Note that we only provide the initial (t=0) asset allocations. Asset allocations at t=1 show even higher variance for scenario reduction.

Table 1: Means and standard deviations (in brackets) of optimization results for the base case using scenario reduction with a tree structure of 1-6-36 (SR, upper part) and moment matching with the same structure (MM, last line), both from 1000 runs per setting. Columns show the size of the starting fan, the objective value at optimum, and the optimal t=0 allocation to assets 1 and 2. Analytical solution: objective value 0.2; asset allocation 1/3 to each asset (note that $x_3 = 1 - x_1 - x_2$). The sixth column shows the share of arbitrage-free trees in the total number of trees generated (computed from the next two columns; trees which admitted arbitrage opportunities were discarded and did not influence any of these results). The next two columns show the reason for discarding trees: either the minimum branching factor was too small, or a positive state price vector could not be found. In total, 1682 trees have been generated to obtain the results in the first row. The final column indicates the average time in seconds required to compute an arbitrage-free tree (incl. optimization, on an Intel i7-2600, 3.4 GHz, 3 GB RAM).

						discarded because of		
Method	Fan size	Obj. value	x_1	x_2	Arb. free	branching factor	state price	time
SR	250	0.25 (0.03)	0.32 (0.70)	0.35 (0.70)	59%	28	654	2.8
SR	500	0.23 (0.02)	0.35 (0.58)	0.33 (0.59)	82%	7	214	2.9
SR	1000	0.22 (0.02)	0.32 (0.64)	0.33 (0.62)	95%	0	57	7
SR	2000	0.21 (0.02)	0.37 (0.61)	0.32 (0.64)	98%	0	22	50
MM		0.20 (4.3E–06)	0.33 (1.0E–04)	0.33 (8.5E–05)	100%	0	0	0.2

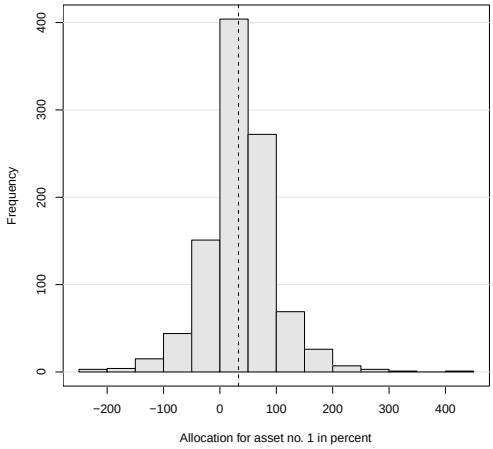

Figure 1: Histogram of the optimal asset allocation for asset no. 1 based on 1000 runs using a 1-6-36 tree structure generated via scenario reduction from a starting fan of 2000 paths.

that due to the large standard deviation the quality of the solution actually observed for a single run becomes rather arbitrary (the dashed line indicates the analytical optimum of 1/3). For the setting in Figure 1 we find that in more than two thirds of all cases, at least one of the asset weights even becomes negative.

Table 1 shows that although tree reduction is not exploited to its fullest, optimal asset allocations have high standard deviations. If we try to exploit the full potential of scenario reduction in terms of substantially reducing the tree size, the minimum tree size for which arbitrage-free trees may (potentially) be found is 1-3-9 (given the minimum required branching factor of three for each node). We repeat the exercise laid out above for this reduced tree size. Table 2 shows that the fraction of arbitrage-free trees declines sharply compared to Table 1. Moreover, the standard deviations of asset allocations increase enormously, leading to coefficients of variation of around 10. Computing times increase markedly compared to the larger 1-6-36 tree structure: While the time for computing a single tree decreases with the smaller tree size, the average time required for constructing an arbitrage-free tree increases due to the large number of trees that have to be discarded.

Table 2: Means and standard deviations (in brackets) of optimization results for the base case using scenario reduction with a tree structure of 1-3-9 from 1000 runs per setting. Columns show the size of the starting fan, the objective value at optimum, and the optimal t=0 allocation to assets 1 and 2. Analytical solution: objective value 0.2; asset allocation 1/3 to each asset (note that $x_3 = 1 - x_1 - x_2$). The sixth column shows the share of arbitrage-free trees in the total number of trees generated (computed from the next two columns; trees which admitted arbitrage opportunities were discarded and did not influence any of these results). The next two columns show the reason for discarding trees: either the minimum branching factor was too small, or a positive state price vector could not be found. In total, 83290 trees have been generated to obtain the results in the first row. The final column indicates the average time in seconds required to compute an arbitrage-free tree (incl. optimization, on an Intel i7-2600, 3.4 GHz, 3 GB RAM).

						discarded because of		
						branching	state	
Method	Fan size	Obj. value	x_1	x_2	Arb. free	factor	price	time
SR	250	0.33 (0.11)	0.32 (4.22)	0.26 (4.33)	1%	64188	18102	29
SR	500	0.32 (0.11)	0.45 (4.33)	0.32 (4.52)	2%	37852	11906	50
SR	1000	0.30 (0.10)	0.31 (3.65)	0.35 (4.01)	2%	29021	10314	157
SR	2000	0.30 (0.11)	0.37 (4.24)	0.56 (4.33)	3%	23589	9274	775

Table 3: Means and standard deviations (in brackets) in percent, computed across 1000 arbitrage-free trees generated by scenario reduction, of means and standard deviations of asset 1 returns. The values of the original return distribution are μ=8% and σ=25%.

tree structure		1-3-9		1-6-36	
starting fan size		500	2000	500	2000
1st stage	mean	7.97 (3.16)	8.05 (2.96)	8.06 (1.58)	8.11 (1.60)
	s.d.	15.3 (5.46)	15.4 (5.25)	18.0 (2.90)	17.9 (2.70)
2nd stage	mean	8.03 (1.84)	8.06 (1.64)	8.00 (0.76)	7.98 (0.68)
	s.d.	15.9 (3.19)	15.8 (2.88)	19.6 (1.21)	18.5 (1.10)

Analyzing the statistical properties of the simulated returns in the trees provides some interesting insights. With moment matching, means and standard deviations of asset returns on the trees correspond exactly to the target values (μ=8% and σ=25% for all assets). Table 3 compares means and standard deviations for asset returns generated by scenario reduction, both for 1-3-9 and 1-6-36 tree structures and selected starting fan sizes. With respect to the mean of the asset return distribution, trees generated by scenario reduction are unbiased. However, standard deviations of return distributions in the trees are biased downwards, with the bias increasing for smaller tree sizes: Instead of the pre-specified σ=25%, the 1-3-9 trees yield standard deviations of around 15-16%, while the 1-6-36 trees lead to standard deviations of around 18-20%. This fact explains the higher expected utility in the objective function (above the theoretical maximum of 0.2 in our example). The standard deviations of parameters decrease markedly for larger trees and slightly for increasing starting fan size. While exact matching of return properties is not the goal of scenario reduction methods, the high variation in both parameters may explain the high variation in the optimal asset allocation using trees generated via scenario reduction.

4.2 Extensions: Effects of Correlation and Higher Moments

The base case example assumes uncorrelated asset returns, which may be viewed as an unrealistic simplification. Moreover, empirical asset return dis-

Table 4: Correlations for asset returns in the extended numerical example

asset	1	2	3
1	1	–0.3	–0.2
2	–0.3	1	0.6
3	–0.2	0.6	1

tributions are often found to be skewed and leptokurtic. Therefore, in this section we first investigate the effect of correlations on the precision of asset weights computed from scenario trees generated by moment matching and scenario reduction. In a second step, we also consider skewed and leptokurtic asset returns.

Maintaining the assumption of normally distributed asset returns with μ=8% and σ=25%, we now assume the correlation matrix for assets 1–3 shown in Table 4, which is taken from Høyland and Wallace (2001, Table 3). In this case the closed-form solution for the optimal asset allocation is $\mathbf{x} = (0.46, 0.33, 0.21)$. In order to match non-zero correlations, we increase the tree structure to 1-7-49 for moment matching. We use the same tree size for scenario reduction to put both methods on an equal footing again. Similar in structure to Table 1, Table 5 provides the results for 1000 optimization runs based on scenarios generated by scenario reduction (for different fan sizes, first four lines) as well as scenarios generated by moment matching (bottom line) when using the correlations in Table 4.

Comparing Table 5 to Table 1 shows that moment matching still yields unbiased results. Standard deviations of asset weights are higher than in the uncorrelated base case, but the precision is still very high (coefficients of variation are on the order of 3–4%). In contrast, for scenario reduction, standard deviations of asset weights increase enormously, leading to coefficients of variation ranging up to 17 for x_2 and a fan size of 500, as compared to roughly 2 in the base case. Hence, for correlated asset returns, optimization results based on trees generated by scenario reduction are far worse than the results for uncorrelated asset returns. For scenario reduction, the standard deviations for the asset allocations x_2 are around five times as large as for x_1. We do not find such differences for the standard deviations of asset allocations for moment matching, and take this as another indicator in its favor.

Finally, we consider skewed and leptokurtic asset return distributions.

Table 5: Means and standard deviations (in brackets) of optimization results for the extended case (including the correlations from Table 4), using scenario reduction with a tree structure of 1-7-49 (SR, upper part) and moment matching with the same structure (MM, last line), both from 1000 runs per setting. Columns show the size of the starting fan, the objective value at optimum, and the optimal t=0 allocation to assets 1 and 2. Analytical solution: objective value 0.2; asset allocation $x_1 = 0.46$, $x_2 = 0.33$, $x_3 = 0.21$. The sixth column shows the share of arbitrage-free trees in the total number of trees generated (computed from the next two columns; trees which admitted arbitrage opportunities were discarded and did not influence any of these results). The next two columns show the reason for discarding trees: either the minimum branching factor was too small, or a positive state price vector could not be found. In total, 1417 trees have been generated to obtain the results in the first row. The final column indicates the average time in seconds required to compute an arbitrage-free tree (incl. optimization, on an Intel i7-2600, 3.4 GHz, 3 GB RAM).

						discarded because of		
Method	Fan size	Obj. value	x_1	x_2	Arb. free	branching factor	state price	time
SR	250	0.30 (0.09)	0.61 (1.12)	0.66 (5.52)	71%	15	402	6.8
SR	500	0.29 (0.08)	0.59 (0.97)	0.44 (7.54)	76%	4	318	7.2
SR	1000	0.27 (0.10)	0.64 (1.77)	0.92 (9.58)	83%	3	209	17
SR	2000	0.26 (0.08)	0.64 (1.26)	0.69 (7.74)	84%	1	195	51
MM		0.20 (8.1E–4)	0.46 (1.7E–02)	0.33 (1.6E–02)	100%	0	0	0.2

We use skewness of (0.49,–0.75,–0.74) from Høyland and Wallace (2001, Table 2), and we deliberately choose a kurtosis of 4 for all assets. These input parameters require a branching factor of 8 for moment matching, so we use a tree structure of 1-8-64 for both methods. While no analytical solution is available for this case, coefficients of variation for asset allocations from moment matching improve (compared to the base case) to 10^{-3}, while those for scenario reduction remain orders of magnitude higher. These assumptions regarding non-normality are rather moderate. Without presenting any further details we conjecture that more extreme (and more realistic) assumptions about skewness and kurtosis would not make comparisons more favorable to scenario reduction.

5 Conclusions

We have applied publicly available implementations of moment matching and scenario reduction methods to a small multi-stage, multi-asset financial optimization problem. Our goal was to compare the two methods numerically on a problem with a known (closed-form) solution to assess their suitability for this class of problems. Our results show that, even after ensuring the absence of arbitrage and for comparatively large trees, optimization results based on scenario reduction fluctuate widely. In contrast, for comparable tree sizes, moment matching exactly replicates the analytically known asset allocation. Moment matching yields very precise asset allocations for correlated and uncorrelated as well as for normal and non-normal assets returns, whereas those based on scenario reduction are biased and strongly fluctuating.

References

Bertocchi, M., Dupačová, J., Moriggia, V., 2006. Horizon and stages in applications of stochastic programming in finance. Annals of Operations Research 142, 63–78.

Geyer, A., Hanke, M., Weissensteiner, A., 2010. No-arbitrage conditions, scenario trees, and multi-asset financial optimization. European Journal of Operational Research 206 (3), 609–613.

Heitsch, H., Römisch, W., 2003. Scenario reduction algorithms in stochastic programming. Computational Optimization and Applications 24, 187–206.

Heitsch, H., Römisch, W., 2009. Scenario tree modeling for multistage stochastic programs. Mathematical Programming, Ser. A 118, 371–406.

Hochreiter, R., Pflug, G. C., 2007. Financial scenario generation for stochastic multi-stage decision processes as facility location problems. Annals of Operations Research 152 (1), 257–272.

Høyland, K., Kaut, M., Wallace, S. W., 2003. A heuristic for moment-matching scenario generation. Computational Optimization and Applications 24, 169–185.

Høyland, K., Wallace, S. W., 2001. Generating scenario trees for multistage decision problems. Management Science 47 (2), 295–307.

Klaassen, P., 2002. Comment on 'Generating scenario trees for multistage decision problems'. Management Science 48, 1512–1516.

Pflug, G. C., 2001. Optimal scenario tree generation for multiperiod financial planning. Mathematical Programming 89 (2), 251 – 271.

Rasmussen, K. M., Clausen, J., 2007. Mortgage loan portfolio optimization using multi-stage stochastic programming. Journal of Economic Dynamics and Control 31, 742–766.

Topaloglou, N., Vladimirou, H., Zenios, S. A., 2008. A dynamic stochastic programming model for international portfolio management. European Journal of Operational Research 185, 1501–1524.