Copula Effect on Scenario Tree

K. Sutiene and H. Pranevicius

Abstract—**Multistage stochastic programs are effective for solving long-term planning problems under uncertainty. Such programs are usually based on scenario generation model about future environment developments. In the present paper, the scenario model is developed for the case when enough data paths can be generated, but due to solvability of stochastic program the scenario tree has to be constructed. The proposed strategy is to generate multistage scenario tree from the set of individual scenarios by bundling scenarios based on cluster analysis. The K-means clustering approach is modified to capture the interstage dependencies. Such generation of scenario tree can be useful in cases when it is difficult to construct the adequate scenario tree from the stochastic differential equations or time-series models, and the sampled paths can be obtained by sampling or resampling techniques. While generating the initial fan of individual scenarios, the copula is employed for modeling the dependence between stochastic variables in a multivariate structure. It allows to model nonlinear dependencies between non-elliptically distributed stochastic variables. While investigating the copula effect on the scenario tree structure, we will try to answer the question: does the copula features are captured in the approximate representation of uncertainty in the form of scenario tree. The proposed scenario tree generation method is implemented on sampled data of discount bond yields. The Gaussian copula and Student's t-copula are employed while generating the set of individual scenarios in the multivariate structure.**

Index Terms—**Copula, K-means clustering, Multistage scenario tree construction, Stochastic programming.**

I. Introduction

The concept of scenarios is usually employed for the modeling of randomness in stochastic programming models [1], [2], in which data evolve over time and decisions have to be made independently upon knowing the actual paths that will occur. Such data are usually subject to uncertainty or some kind of risk. For instance, the random variables are the return values of each asset on an investment in portfolio management problems, and the investment decisions must be implemented before the asset performance can be observed. Each scenario can be viewed as one realization of an underlying multivariate stochastic data process. The modeling of randomness employees the set of available past data with the aim of building submodels for each individual stochastic parameter. These submodels are used to generate a set of scenarios that encapsulate the consistent depictions of pathways to possible futures based on assumptions about economic and technological developments. Thus, the factors driving risky events are approximated by a discrete set of scenarios, or sequence of events. This process is known as scenario generation. Scenarios can be generated using various methods, based on different principles: conditional sampling, sampling from given marginals and correlations, moment matching, path based methods, optimal discretization, as in [3]–[7]. Stochastic programming (optimization) has been applied in the following areas: 1) Manufacturing production capacity planning; 2) Electrical generation capacity planning; 3) Asset liability management; 4) Portfolio selection; 5) Traffic management; 6) Machine scheduling. In these applications decisions must often be taken in the face of the unknown.

A good approximation may involve a very large number of scenarios with probabilities. A better accuracy of uncertainties is described when scenarios are constructed via a simulated data path structure, also known as a scenario fan. But the number of scenarios is limited by the available computing power, together with a complexity of the decision model. To deal with this difficulty, we can reduce the dimension of the initial scenario set by constructing the multistage scenario tree out of it. The decision on the number of stages, on the size of time periods and on the branching scheme is very important for a good representation of the uncertainty in the form of scenario tree, which is input into the multistage stochastic program. The detailed description of both scenario fan and scenario tree will be given in Section III.

In the present paper, we concentrate on the generation of scenario trees when the underlying stochastic parameters have been determined and the data paths of their realizations can be generated. The scenario tree can be constructed out of sampled paths by employing some classifying method, such as clustering analysis. While bundling the scenarios to the clusters, the interstage dependencies have to be captured. An approach similar to our work is introduced in the article [8], but without a detailed clustering algorithm. Due to this, the K-means clustering method is modified to treat properly the interstage dependencies and is implemented while constructing the scenario tree from simulated data paths.

Such generation of scenario tree can be useful in cases when it is difficult to construct the adequate scenario tree from the stochastic differential equations or time-series models, and the sampled paths can be obtained by sampling or simulation

Manuscript received October 03, 2007.

K. Sutiene is with the Business Informatics Department, Kaunas University of Technology, Studentu 56-301, Kaunas LT-51424, Lithuania (corresponding author to provide phone: 370-681-52842; fax: 370-37-451654; e-mail: kristina.sutiene@stud.ktu.lt).

H. Pranevicius is with the Business Informatics Department, Kaunas University of Technology, Studentu 56-301, Kaunas LT-51424, Lithuania (e-mail: hepran@if.ktu.lt).

techniques. The proposed scenario tree construction algorithm allows incorporating a copula-based dependence measure [9], [10] to describe the dependence between stochastic variables in a multivariate structure. Due to assumptions of using the Pearson's correlation coefficient, the usefulness of such correlation is restricted. The main advantage of employing copulas is that they allow to model the nonlinear dependencies between non-elliptically distributed stochastic variables. The copula function has been introduced in finance by Embrechts, McNeil, and Straumann [9]. To our knowledge, the copulas still are not very popular in generating the scenario trees. According to this, we propose to approximate the multivariate stochastic process by a scenario fan with multivariate structure using copulas. Then, the scenario tree is constructed out from the sampled paths using the modified K-means clustering algorithm. Numerical experience is reported for constructing multivariate scenario trees of discount bond yields, employing two separate – Gaussian and Student's t – copulas.

The rest of the paper is organized as follows. The scenario generation model is introduced in Section II. The mathematical model consists of two main components: models for the univariate marginal distributions of uncertain factors and a model of the dependence structure employed in the notion of copulas. Gaussian copula and Student's t-copula are considered. The simulation algorithm of modeling the copula based dependent data is given. Section III discusses how the copula function can be incorporated while generating scenarios. Section IV describes how the simulated data paths can be transformed to the scenario tree using cluster analysis. The K-means clustering algorithm is modified to bundle the time-dependent data. Section V demonstrates the numerical example of scenario tree generation based on discount bond yields data. Finally, some concluding remarks are given.

II. Scenario Generation Computational Procedure

Stochastic programming (optimization) combines model of optimum resource allocation and models of randomness, thereby it creates a decision making framework (see Fig. 1) [11]. Whereas deterministic optimization problems are formulated with known parameters, real world problems almost invariably include parameters which are unknown at the time a decision should be taken. That's why the deterministic approach is expanded.

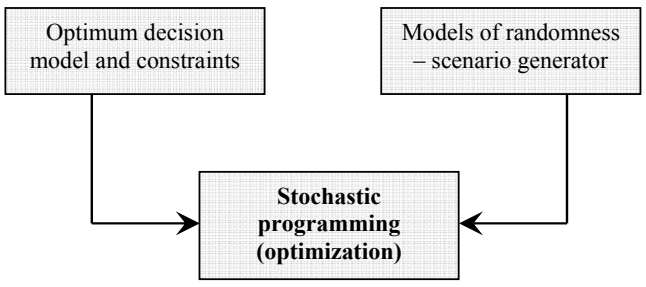


Fig. 1. Stochastic programming paradigm

In stochastic programs the first element is the objective function which together with the constraints describes the core of the problem that has to be solved and varies with each individual application. The second element is the scenario generator, and it is used to describe the uncontrollable (risk) factors affecting the relevant system, such as, inflation rate, interest rate, GPD – factors that are not under control of the decision makers. The uncertain elements are modeled as random variables to which the probability theory can be applied. A concept of scenarios is used to represent of how the future might unfold. Some kind of probabilistic model or simulation can be used to generate a batch of scenarios. The models of randomness with their finite and discrete realizations are called scenario generators. The outcome of such system is uncertain even when the values of all the decision variables are fixed. Scenarios can also be used in descriptive models, where a set of mathematical operations are defined that can predict how a mathematical system will behave, e.g. Markov models.

The main feature of stochastic programming is its multistage formulation. Despite rich involvement of the future, everything is aimed to make a well hedged decision in the present. The attitude is adopted that a decision will be properly made in the present only taking into account, at least some to extent, the opportunities for modification or correction at later times. Decisions at later times can respond to the information that has become available since the initial decision. Thus, during the time the decisions alternate with observations: initial decision $\rightarrow$ observation $\rightarrow$ recourse decision $\rightarrow$ observation $\rightarrow \ldots \rightarrow$ recourse decision. This sequence doesn't go on indefinitely, but the number of stages can be large enough. Decisions that are taken have no effect on the probability structure. Thus, we have a multistage problem. The number of stages is used in modeling the uncertainty; we will formalize this later in terms of the multistage scenario tree.

In the paper [11], the general scenario generation procedure for multistage stochastic programs is given. We append this procedure with additional step (Step 2), paying the important attention to the dependence modeling among risk factors. The following steps (some or all) have to be performed while generating scenarios:

1) Collecting historical data of stochastic parameters, assumptions of a model, estimation/calibration of parameters for a chosen model.
2) Choosing the appropriate model to describe the dependence structure among stochastic parameters.
3) Generation of scenarios according to the chosen model or discretization of the distributions using approximation of statistical properties.

We will consider all these steps in deeper manner.

A. Modeling Paradigm

To solve a stochastic decision making problem, we need knowledge about the probability distribution of all random

variables among the uncontrollable inputs. In paper [12], the author proposes four types of problems, concerning the level of the available information:

1) Full knowledge of underlying probability distribution ;
2) Known parametric family ;
3) Sample information ;
4) Low information level.

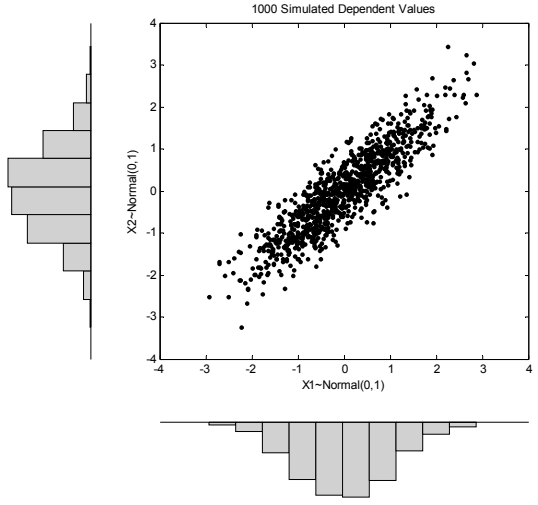
These four groups are not strictly distinguishable. Different information levels can be applied to the distinct parameters of the model. The most popular modeling paradigms are [11]:

- Econometric Models and Time Series (ARMA, GARCH, VAR models) ;
- Geometric Brownian Motion (Diffusion Processes) ;
- Artificial Intelligence (Neural Networks) ;
- Statistical Approaches (Statistical approximation, Forecasting, Moment Fitting) ;
- Sampling.

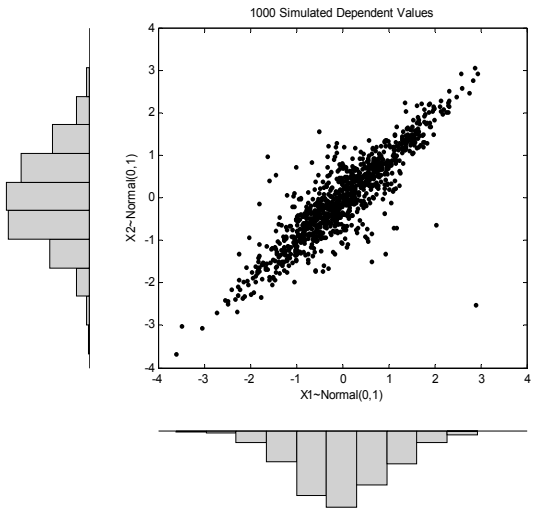
It is very important that the sample we use to represent the stochastic parameters in the form of scenarios would be consistent with empirical data. Therefore, one has to specify the stochastic processes for risk parameters, and estimate the parameters of such models using empirical data.

B. Dependence structure modeling

In this paper we concentrate on generation of scenarios representing the realizations of multivariate stochastic process whose components are correlated. We define such scenarios as intercorrelated scenarios, meaning that they correlate through the components of multivariate structure. Historically measuring and modeling of dependence has centered on correlation. The modeling of dependent variables is performed employing the Pearson's correlation matrix to describe the multivariate structure. Many applications show that relationships among stochastic variables may be very complex, and linear dependence can't reflect these relationships adequately. The reason is that the Pearson's correlation



(a) Dependence between X_1 and X_2 with $\rho = 0.7$



(b) Dependence between X_1 and X_2 with $\rho = 0.7$

Fig. 2. Different dependence structures

coefficient does not capture any non-linear dependencies, and it is usually used assuming the elliptical shape of normal distribution in applications. We include Fig. 2 as motivation for the ideas of this paper. It shows 1000 random variates from two distributions with identical standard Gaussian marginal distributions: case (a) and case (b) depict bivariate structure of X_1 and X_2 with linear correlation coefficient $\rho = 0.7$. However, the dependence structure between X_1 and X_2 is qualitatively quite different. It relates that in case (b) extreme values have a tendency to occur together. This example shows that the dependence between random variables cannot be distinguished on the grounds of correlation alone. Additionally, in real applications it is rare for distributions to follow the strict spherical assumptions with a constant dependence across the distribution implied by correlation.

To overcome the limitations of correlation, the practitioners can draw on copula functions. It is very powerful technique, which allows to represent joint distribution by splitting the marginal behavior, embedded in the marginal distributions, from the dependence, captured by the copula itself. This superiority of using copulas releases the modeling, estimation and simulation of dependent random variables. Let define the copula itself.

A function C is the d -dimensional copula if it fulfills the following properties [13]:

1) The domain of C is $[0,1]^d$;
2) C is grounded and d -increasing ;
3) The margins C_k of C satisfy $C_k(u) = u$, $k = 1,2,\ldots,d$ for all u in $[0,1]$.

Let consider d random variables $Y_1, Y_2, \ldots, Y_d$ with multivariate distribution F and univariate margins $F_1(y_1), F_2(y_2), \ldots, F_d(y_d)$. Sklar's theorem, which is the foundation for copulas, states that any joint distribution can be written in a copula form.

Sklar's Theorem (1959). Given a joint distribution function $F(y_1, y_2, \ldots, y_d)$ for random variables $Y_1, Y_2, \ldots, Y_d$ with marginals $(F_1, F_2, \ldots, F_d)$, F can be written as a function of its marginals:

$$F(y_1, y_2, \ldots, y_d) = C(F_1(y_1), F_2(y_2), \ldots, F_d(y_d)),$$

where copula $C(u_1, u_2, \ldots, u_d)$ is a joint distribution with uniform marginals. Moreover, if each F_i is continuous, C is unique.

The dependence structure can be represented by a proper copula function. Moreover, the following corollary is attained from Sklar's theorem.

Corollary. Let F be an d - dimensional distribution function with continuous margins $F_1(y_1), F_2(y_2), \ldots, F_d(y_d)$, and copula C. Then, for any $u = (u_1, u_2, \ldots, u_d)$ in $[0,1]^d$:

$$C(u_1, u_2, \ldots, u_d) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), \ldots, F_d^{-1}(u_d))$$

where F_i^{-1} is the generalized inverse of F_i.

A good many copulas are available, with differing characteristics that lead to the different relationships among variables generated. Note that copulas differ not so much in the degree of association they provide, but rather in which part of the distributions the association is strongest: the behavior of copulas in the right and left tails can be used to distinguish among joint distributions that produce the same overall correlation.

In this paper, we will consider two copulas: Gaussian copula and Student's t-copula. These copulas do not have a simple closed form, but are implied by well known multivariate distribution functions: multivariate Gaussian and multivariate Student's t distributions respectively. The difference between these two copulas is that the Student's t-dependence structure supports joint extreme movements regardless of the marginal behavior of stochastic variable compared with the Gaussian copula. A complete copula-based joint distribution can be constructed using assessed rank-order correlations and marginal distributions. Examples of rank-order correlations are Spearman's rho and Kendall's tau correlations, which are used to describe the dependence relations of a monotonic nature: it indicates the tendency of two random variables to increase/decrease concomitantly (positive dependence) or contrariwise (negative dependence).

The Gaussian or normal d -copula is given by

$$C_{Cor}^{Ga}(u) = \Phi_{Cor}^{d}(\Phi^{-1}(u_1), \Phi^{-1}(u_2), \ldots, \Phi^{-1}(u_d)),$$

where Φ denotes the standard univariate normal distribution function and Φ_{Cor}^{d} denotes the standard multivariate normal distribution function with matrix Cor of linear correlation coefficients. The main property of such dependence structure is that Gaussian copula does not have neither upper nor lower tail dependence.

Simulation procedure for Gaussian copula is performed as follows:

(a) convert Kendall's tau cor_{ij}^{τ} to the linear correlation coefficient cor_{ij} using formula $cor_{ij} = \sin(\pi\, cor_{ij}^{\tau} / 2)$ (the relationship between the linear correlation coefficient cor_{ij} and Spearman's rho cor_{ij}^{S} is $cor_{ij} = 2\sin(\pi\, cor_{ij}^{S} / 6)$) and construct the lower triangular matrix $A = \lfloor a_{ij} \rfloor$ that holds $Cor = AA'$;

(b) generate independent standard normal variables ε_i , $i = \overline{1,d}$ and form a column vector ε ;

(c) construct a joint probability density function, taking the matrix product $\widetilde{\varepsilon} = A\varepsilon$;

(d) set $\widetilde{u}_i = \Phi(\widetilde{\varepsilon}_i)$;

(e) set $\widetilde{x}_i = \Phi^{-1}(\widetilde{u}_i)$.

At the result, $\widetilde{x}_i$, $i = \overline{1,d}$ are dependent variables based on Gaussian copula. Fig. 3 shows four scatter plots of 1000 random values from a bivariate Gaussian copula for various levels of correlation coefficient to illustrate the range of different dependence structures. Thus, the family of Gaussian copulas is parameterized by the linear correlation matrix.

The Student's t-dependence structure introduces an additional parameter compared with the Gaussian copula, namely the degrees of freedom. Student's t-copula can be written as

$$\widetilde{C}_{Cor,v}^{t}(u) = t_{Cor,v}^{d}(t_v^{-1}(u_1), t_v^{-1}(u_2), \ldots, t_v^{-1}(u_d)),$$

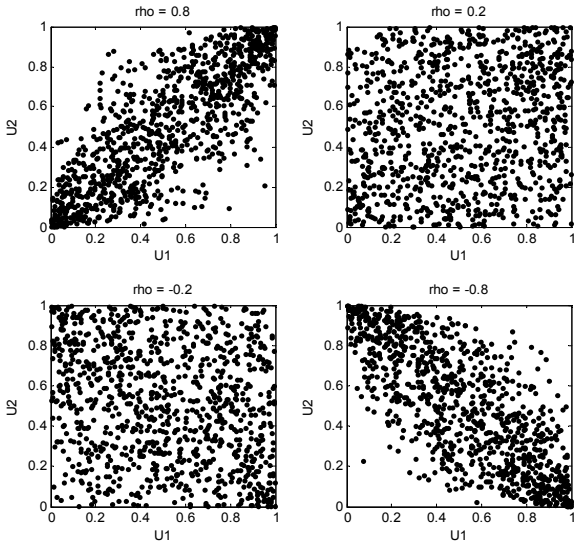


Fig. 3. Gaussian dependence structures with different correlation coefficients

where Cor, v are the parameters of t-copula, $t^d_{Cor,v}$ denotes the joint distribution function of the d -variate Student's t-distribution with v degrees of freedom, t_v^{-1} is the inverse of univariate Student's t-distribution with v degrees of freedom. Student's t-copula has the additional parameter v comparing with Gaussian copula. Increasing the value of v decreases the tendency to discover extreme co-movements.

Simulation procedure for Student's t-copula is performed as follows:

(a) convert Kendall's tau cor_{ij}^{τ} to the linear correlation coefficient cor_{ij} using formula $cor_{ij} = \sin\left(\pi\, cor_{ij}^{\tau}/2\right)$ (the relationship between the linear correlation coefficient cor_{ij} and Spearman's rho cor_{ij}^{S} is $cor_{ij} = 2\sin\left(\pi\, cor_{ij}^{S}/6\right)$) and construct the lower triangular matrix $\mathrm{A} = \lfloor a_{ij} \rfloor$ that holds $Cor = AA'$;

(b) generate independent standard normal variables ε_i , $i = \overline{1,d}$ and form a column vector ε ;

(c) construct a joint probability density function, taking the matrix product $\widetilde{\varepsilon} = A\varepsilon$;

(d) generate a random variate $\gamma \sim \chi^2_v$;

(e) calculate $\widetilde{\widetilde{\varepsilon}} = \sqrt{v}\widetilde{\varepsilon}/\sqrt{\gamma}$;

(f) $\widetilde{\widetilde{u}}_i = t_v\left(\widetilde{\widetilde{\varepsilon}}_i\right)$;

(g) $\widetilde{\widetilde{x}}_i = \Phi^{-1}\left(\widetilde{\widetilde{u}}_i\right)$.

At the result, $\widetilde{\widetilde{x}}_i$ are dependent variables based on Student's t-copula. Fig. 4 shows four scatter plots of 1000 random values from a bivariate Student's t-copula, with degrees of freedom equal 2, for various levels of correlation coefficient to illustrate the range of different dependence structures. These plots demonstrate that t_2-copula differs from Gaussian copula (see Fig. 3), even when their components have the same correlation.

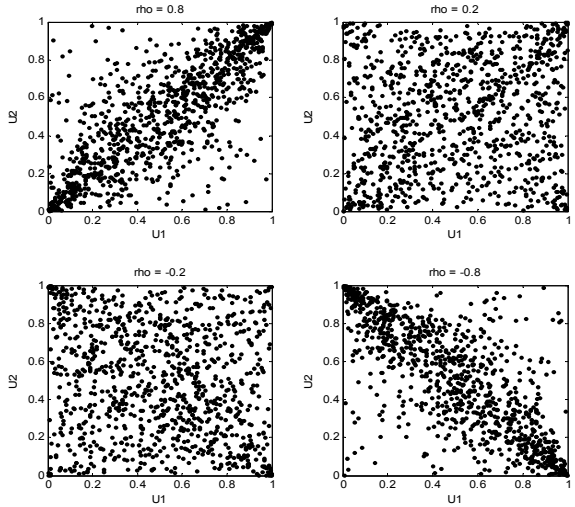


Fig. 4. Student's t dependence structures with degrees of freedom equal 2, but different correlation coefficients

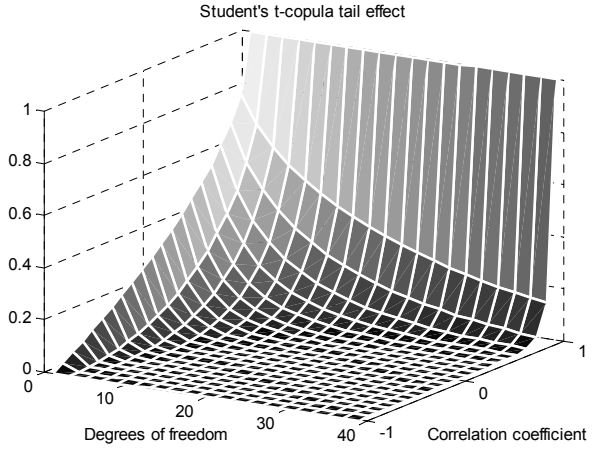


Fig. 5. Tail dependence for t_2-copula

The main difference between the considered copula functions is in measuring the dependence between the occurrences of extreme values. Bivariate tail dependence coefficient measures the strength of dependence in the upper and lower quadrant tail of a bivariate distribution. The upper tail dependence coefficient is as follows [10]:

$$\lambda_U\left(Y_1, Y_2\right) = \lim_{\alpha \to 1} P\left(Y_2 > F_2^{-1}(\alpha) \middle| Y_1 > F_1^{-1}(\alpha)\right).$$

Analogously, the lower tail dependence coefficient is

$$\lambda_L\left(Y_1, Y_2\right) = \lim_{\alpha \to 0} P\left(Y_2 \le F_2^{-1}(\alpha) \middle| Y_1 \le F_1^{-1}(\alpha)\right).$$

The upper (lower) coefficient quantifies the probability to observe a large (small) Y_2 , when Y_1 is large (small). If $\lambda_U, \lambda_L \in (0,1]$, then two random variables Y_1 and Y_2 are said to be asymptotically dependent in tails. And if $\lambda_U = 0, \lambda_L = 0$, then variables are said to be asymptotically independent in tails. Furthermore, given the radial symmetry property of elliptical distributions, the lower and upper tail dependence coefficients coincide. In the work [10], it was shown that tail dependence coefficient is equal to zero, confirming the asymptotic independence in tails of the Gaussian copula. Student's t-copula tail effect from both degrees of freedom and correlation coefficient is depicted in Fig. 5. One can see that the stronger the linear correlation coefficient and the lower the degrees of freedom, the stronger is tail dependence.

Calibrating the copula parameters to the real data is the active research area in the current statistics literature [10, 14, 15, 16]. Most popular approaches used in estimation of copula are Exact Maximum Likelihood method (EML), Inference Functions for Margins method (IFM), Canonical Maximum Likelihood method (CML) and others. In this work, we won't go into the details of parameterizing the copula.

During the discretization process of d-dimensional distribution function F , one can strengthen the dependence in different parts of distribution through the choice of copula.

Indeed, the assumption of normality for the margins can be removed and $(F_1, F_2, \ldots, F_d)$ may be fat-tailed distributions (e.g. Student, Weibull, Pareto), and dependence may be characterized by a Normal or other chosen copula. That is, the dependence structure between stochastic variables can be modeled independently of marginal distributions.

C. Generation of Scenarios

The four main scenario generation approaches are [3]:

1) Sampling. Sampling approaches are Monte Carlo (Random) Sampling, Importance Sampling, Bootstrap Sampling, and Conditional Sampling. For each sampling method, the main principle is to take a sample from a probability distribution function so that for a given value we have an associated probability, which gives the scenario value and its branch probability.
2) Statistical Approaches. The main principal is to determine the value of particular statistical properties of given data. The most popular is statistical moment or property matching approach, whereby we do not assume knowledge of a random variable's probability distribution function. Instead we describe the distribution by its statistical moments or other properties, e.g. mean, variance, percentile.
3) Simulation. It is an approach for scenario generation, where some underlying mathematical process is simulated: random numbers are incorporated into the random component of an equation and the result is recorded. Scenario generation by simulation results the set of simulated data paths with equal probability. To reduce the number of paths sometimes paths are bounded by some method. Stochastic Process Simulation, Error Correction Model, Vector Autoregressive model are most popular simulations used for stochastic programming.
4) Other methods. It can be methods from other fields, such as Artificial Neural Networks, Clustering, or it can be combination of some scenario generation methods.

None method of scenario generation is approved as optimal but the goal should be the adequate representation of uncertainty.

To remember the idea of this paper, we aim to incorporate a copula in generation of scenarios. In the paper [17], the moment matching method was used to generate copula based correlated scenarios. In our paper, we will use the combination of some methods allowing us to employ copula functions for modeling dependent stochastic variables: the simulation and clustering approaches are combined to construct the scenario tree. In the next section the stochastic programming notation is given to make the description of scenario generation more formal.

III. Stochastic Programming Notation

In multistage stochastic programs the underlying multivariate stochastic data process has to be discrete in time. Mathematically, we have a time index $t = \{0, \ldots, T\}$ and a time horizon consisting of T stages. The stochastic process $\xi = \{\xi_t\}_{t=1}^{T}$ is defined on some filtered probability space $(\Omega, S, \mathcal{F}, P)$. The sample space Ω is defined as $\Omega := \Omega_1 \times \Omega_2 \times \ldots \Omega_T$, where $\Omega_t \subset \mathcal{R}^d$. Note that the sample spaces are taken as finite dimensional. In this case, we consider d-dimensional spaces, but in other applications it is possible to vary the dimensionality. For instance, these data may correspond to the observed return of d financial assets at different time moments t. The σ-algebra S is the set of events with assigned probabilities by measure P, and $\{\mathcal{F}_t\}_{t=1}^{T}$ is a filtration on S. The decisions are of two types: the initial decision x_0 taken at initial time moment and the recourse decisions x_t, $t > 0$ taken at T recourse stages. Thus, the decision at stage t is the random variable $x_t : \Omega \to \mathcal{R}^{n_t}$; decisions are set as finite dimensional, but of possibly varying dimensionality. In the stochastic programming model the observations and decisions are given as a sequence $x_0, (\xi_1, x_1), (\xi_2, x_2), \ldots, (\xi_T, x_T)$, where $x = \{x_t\}_{t=0}^{T}$ is a decision process, measurable function of ξ. The constraints on a decision at each stage involve past observations and decisions. It means that decision x_t at t is measurable with respect to $\mathcal{F}_t \subseteq \mathcal{F}$. Thus, following [8] the decision process is said to be nonanticipative. It means that the decision $x_t = x_t(x_{t-1}, \xi_{t-1})$ taken at any $t > 1$ does not directly depend on future realizations of stochastic parameters or on future decisions. At the time when initial decision must be chosen, nothing about the random elements in our process has been pinpointed. But in making a recourse decision we have the revealed current information until this moment and the residual uncertainty till the end of time horizon. More information on multistage stochastic programs can be found in literature [1], [2], [18]. We continue on with stochastic notation of scenario generator.

The d-dimensional probability distribution function of $\xi_t = (\xi_t^1, \ldots, \xi_t^d)'$ at point $y = (y_1, \ldots, y_d)'$ is denoted by $f(y)$, the d-dimensional cumulative distribution function is denoted by $F(y)$. The joint distribution F provides a complete information concerning the behavior of ξ. The marginal probability distribution function and cumulative distribution function of each element ξ_t^i at point y_i, $i = 1, \ldots, d$ is denoted by $f_i(y_i)$ and $F_i(y_i)$, respectively. The primary aim of scenario generator is to represent the distribution f in a reasonable way. In stochastic programming the underlying probability distribution f is replaced by a discrete distribution P carried by a finite number of atoms $\xi^s = (\xi_1^s, \ldots, \xi_T^s)$, $\xi_t^s = (\xi_t^{s,1}, \ldots, \xi_t^{s,d})'$, $s = 1, \ldots, S$ with probabilities $p_s = P(\xi^s)$, $p_s \geq 0$ and $\sum_{s=1}^{S} p_s = 1$. The atoms ξ^s, $s = 1, \ldots, S$ of the

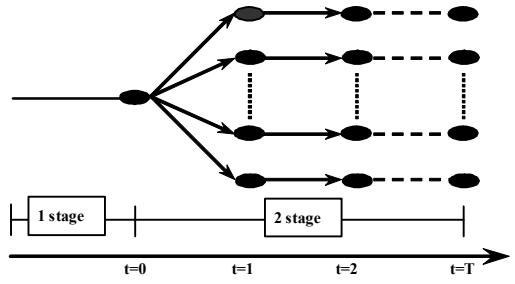


Fig. 6. Scenario fan

distribution P are called as scenarios. Naturally, the historical data in conjunction with an assumed background model are used to generate the scenarios, applying suitable estimation, simulation and sampling procedures.

While approximating the multivariate stochastic distribution F employing copulas, the set of *d*-dimensional intercorrelated scenarios $\xi^s = \left(\xi_1^s, \ldots, \xi_T^s\right)$, $\xi_t^s = \left(\xi_t^{s,1}, \ldots, \xi_t^{s,d}\right)'$, $s = 1, \ldots, S$ is generated. Assuming that all scenarios coincide at $t = 0$, the initial root node is formed, and thus the simulated data paths are called as a scenario fan (see Fig. 6). The structure of simulated data paths can be divided into two stages. The first stage is usually represented by a single root node, and the values of random parameters during the first stage are known with certainty. Moving to the second stage, the structure branches into individual scenarios at time $t = 1$, as shown in the Fig. 6. If such scenario fan is used as input into the multistage stochastic program, the model is of 2-stage problem, as all σ -fields $\mathcal{F}_t$, $t = 1, \ldots, T$ coincide. The 2-stage multiperiod stochastic program has the following properties, as in [8]:

1) Decisions at all time instances $t = 0, 1, \ldots, T$ are made at once and no further information is expected.
2) Except for the first stage no nonanticipativity constraints appear.

Depending on the considered problem, such properties can be regarded as disadvantages. Our aim is to create a multistage scenario tree which can be used for multistage models. Multistage formulation is characterized by its robustness, stability of solutions: similar subscenarios result in similar decisions. The multistage tree reflects the interstage dependency and decreases the number of nodes while comparing to the scenario fan. The structure of multistage tree (see Fig. 7) at $t = 0$ is also described by a sole root node and by branching into a finite number of scenarios as it was in previous case. The stages are connected with possibility to take additional decisions based on newly revealed information. Such information can be obtained periodically (every day, week, month) or based on some events (expiration of investment portfolio). The distinction between stages, which correspond to the decision moments, and time periods is essential, because in practical application it is important that the number of time periods would be greater than the corresponding nodes. The arcs linking nodes represent various realizations of random variables. The number of branches from each node can vary depending on problem specific requirements, and not definitely constant through the tree. One of the strategies is to use an extensive branching at the beginning of time horizon and a relatively poor branching at the last levels of tree. Each path through the tree from its root to one of its leaves corresponds to one scenario, i.e. to a particular sequence of realizations of random coefficients.

The algorithm of transforming the scenario fan to the multistage scenario tree of prescribed structure is described in the next section.

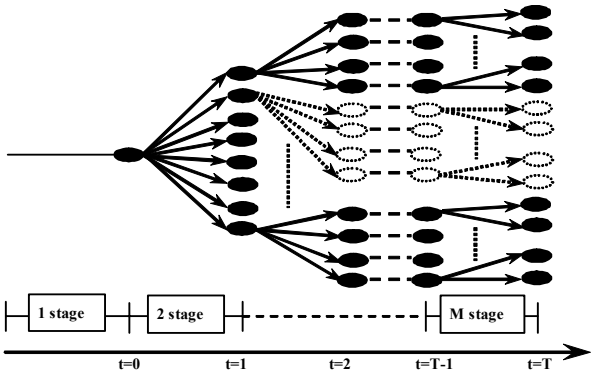
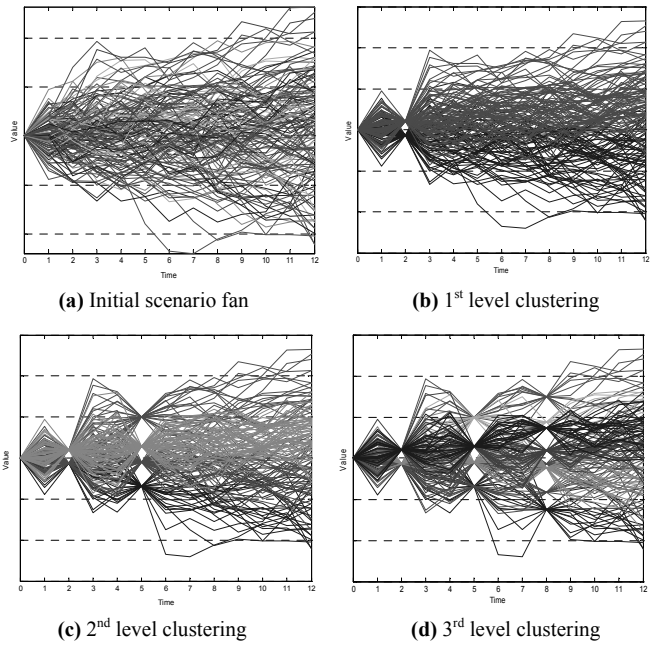


Fig. 7. Multistage scenario tree

IV. K-Means Clustering: Fan to Tree

While constructing the multistage scenario tree from the scenario fan, the fan of individual scenarios is modified by bundling scenarios based on the cluster analysis.

The idea of bundling scenarios to the clusters is depicted in the Fig. 8.



(a) Initial scenario fan **(b)** 1st level clustering

(c) 2nd level clustering **(d)** 3rd level clustering

Fig. 8. Illustration of 4-stage tree construction

It is assumed that a set of individual scenarios for the entire time horizon (12 time moments) is already generated (see Fig. 8a). The scenario fan of 100 scenarios is schematically illustrated. At time $t = 0$ all these scenarios (which are the same) form the root node of the tree. The strategy is to construct the scenario tree with two branches for each decision moment. If two branches are desired from the current scenario tree node, then two clusters have to be formed. Let assume that we have three decision dates, i.e. we are planning to make decisions at 2, 5 and 8 time moments. With this initial setting, we are ready to construct the multistage scenario tree. Thus, at the 2 time moment two clusters are formed by the first iteration of some clustering algorithm. The result is displayed in Fig. 8b. The center of each cluster is computed, which represents the one-level nodes at time $t = 2$. Next, at previous step formed clusters are divided into two subclusters. It results that at time $t = 5$ we have four clusters representing two-level nodes, since the centers are calculated (see Fig. 8c). Such strategy of bundling scenarios to the clusters continues for all defined decision moments. Fig. 8d depicts the third level of some clustering algorithm, since two more subclusters are formed at time $t = 8$. The constructed scenario tree has 4 stages and 8 scenarios.

The computed nodes (cluster centers) are denoted by black points in the generated scenario tree (see Fig. 9). Joining the black points by line, we get the graphical representation of scenario tree. Such strategy of bundling scenarios to the clusters can continue till the end of time horizon is reached.

The discussed technique allows to produce the tree with such characteristics:

1) The projection of random variable nearer the time horizon is less critical than those for the near future, because number of scenarios grows smaller down the tree and the centers that represent the scenario cluster are calculated from a smaller sample size.
2) It allows to model extreme events because at every stage the simulated scenarios in all of the clusters are not discarded, and at the next stage all simulated scenarios in all of the clusters are used to calculate the centre of cluster.

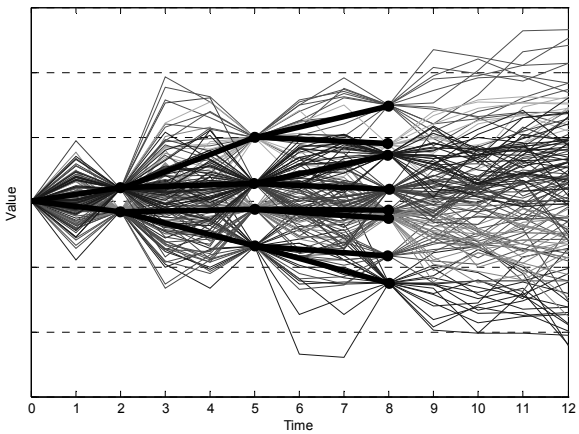


Fig. 9. Graphical representation of 4-stage scenario tree

In the following, we discuss how the approach from cluster analysis is applied to group similar scenarios. The scenario fan usually consists of large number of scenarios, that's why the hierarchical methods can fail. We don't also require the method that in finding the clusters would be optimal by some measures. In the literature, the clustering methods usually are used for stable data; thus we have to make some modifications in order to cluster the time dependent data, such as scenarios. One of the main factors to delineate the structure of scenario tree is the branching scheme. Let assume that K branches are desired from each scenario tree node: the tree is homogeneous. It means that K clusters will need to be formed. Thus, the K-means clustering algorithm [19] is chosen to construct the scenario tree from the set of simulated paths (scenario fan). Clustering consists in partitioning of a data set into subsets, so that the data in each cluster share the common attribute. This similarity is often defined by some distance measure. After a discussion of the kind of requirements we are using, we describe the modified K-means clustering algorithm.

Given a fan of individual scenarios $\xi^s = (\xi_1^s, \ldots, \xi_T^s)$, $s = 1, \ldots, S$ and the number K of desired clusters $\widetilde{C}^1, \ldots, \widetilde{C}^K$, it is needed to find the cluster centers $\overline{\xi}^k$, $k = 1, \ldots, K$ such that the sum of the 2-norm distance squared between each scenario ξ^s and its nearest cluster center $\overline{\xi}^k$ is minimized:

$$\sum_{k=1}^{K} \sum_{\xi^s \in \widetilde{C}^k} \left\| \xi^s - \overline{\xi}^k \right\|_2^2 \to \min.$$

While clustering the scenarios, the main ideas are:

- In current stage M new subclusters have to be formed from clusters formed in previous stage $(M-1)$. That's why it is called multi-level clustering.
- Centroids should be calculated only at stage indexed time moments, but distance measure should evaluate all scenario.
- Other constraints that are used to realize various requirements for new formed clusters can be added.
- The probabilities of each node should be evaluated.

According to the ideas given above, the modified K-means clustering algorithm is given as follows. At the beginning, the decision moments are set, corresponding to the stage index $t \in (1, \ldots, T)$. Then iterate:

Step 1: *Setting initial centers*. Let $\overline{\xi}^k$, $k = 1, \ldots, K$ be the cluster centers. Some method can be used to choose initial cluster centroid positions, sometimes known as "seeds. It might be chosen to be the first K scenarios, since the scenarios are independently generated; or K scenarios by random.

Step 2: *Cluster assignment*. For each scenario ξ^s, assign ξ^s to the cluster $\widetilde{C}^k$, such that center $\overline{\xi}^k$ is nearest to ξ^s in the

2-norm, which is modified to exploit the whole sequence of simulated data path:

$$d\left(\xi^s, \bar{\xi}^k\right) = \sum_{i=1}^{T} \left\| \xi_i^s - \bar{\xi}_i^k \right\|_2 .$$

If scenarios are all in the same physical units, then the Euclidean distance metric is sufficient to successfully group similar data. It is possible to apply other distance metrics, such as Manhattan distance, Maximum norm, Mahalanobis distance, to group similar scenarios, only some modifications have to be done to employ the whole simulated data sequence.

Step 3: *Cluster update*. Compute $\bar{\xi}^k$ as the mean of all scenarios assigned to the cluster $\tilde{C}^k$:

$$\bar{\xi}^k = \mathbb{E}\left\{\xi^s\right\}_{\xi^s \in \tilde{C}^k} .$$

This formula can be replaced by other estimate, such as median, mode or else.

Step 4: *Repeat.* Go to Step 2 until convergence, i.e. no scenario moves the group.

Step 5: *Calculation of probabilities.* Probability of $\bar{\xi}^k$ is equal the sum of probabilities of the individual scenarios ξ^s, belonging to the relevant cluster $\tilde{C}^k$.

Step 6: *Modification.* Modify $\xi^s = \left(\xi_1^s, \ldots, \xi_T^s\right)$ by replacing ξ_t^s with $\bar{\xi}^k$ if $\xi_t^s \in \tilde{C}^k$.

Step 7: *Repeat*. Go to Step 1 if next stage index exists. The clustering procedure starts over using each of clusters formed in current iteration.

This algorithm produces a separation of scenarios into groups. The given algorithm lets to treat properly the interstage dependencies, exploiting the whole sequence of simulated scenario path. At the end, the scenario tree is constructed, consisting of nodes $\bar{\xi}_k$ with their probabilities and the branching scheme.

V. Computational Experiment

The scenario tree generation approach is applied to construct scenario trees out of sampled scenarios provided by Hibbert, Mowbray and Turnbull (HMT) stochastic asset model [20]. The following are some general properties incorporated in the scenario generator:

1) Mean-reversion. It is assumed that a long-term stationary equilibrium level exists to which the asset return process will tend over time.
2) Autoregression. The time series do fluctuate around a certain equilibrium level and at each step the process reacts from a previous deviation with one time lag. The dependence over time (intertemporal dependence) is considered.
3) Volatility. In this paper, variances will be considered constant over time.
4) Correlation. Noises are correlated. The general model is multivariate mean reverting model of financial returns. HMT model is composed of a number of component parts that are driven by a set of stochastic drivers. Thus, the dependencies among various risk factors (contemporaneous) are modeled. The alternative method – copula based dependence measure – is chosen to model the relationships among stochastic parameters.

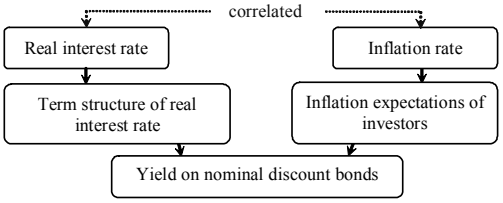


Fig. 10. A cascade structure of HMT model

On the technical level, the Monte-Carlo simulation is selected to generate very large number of plausible scenarios because of its flexibility and intuitive presentation.

We use this model to generate the sample, which consists of a finite number of scenarios, representing realizations of discount (zero-coupon) bond yields. A cascading structure is a characteristic of HMT model: real interest and inflation rates are simulated, which then, depending on the relationship structure assumed, influence the realization of discount bond yields (see Fig. 10).

In HMT model presented here, the economic relationship between inflation, inflation expectations, real interest rates and nominal interest rates is explicitly considered. The model produces the term structure that has closed-form solutions for bond prices so that the entire term structure for any future projection date can be quickly generated: the analytical expressions are available for discount bond prices.

In HMT model, the main stochastic drivers – inflation rate and real interest rate – are described by two factors Ornstein-Uhlenbeck process in continuous time:

$$\begin{cases} dr_1(t) = \alpha_{r1}\left(r_2(t) - r_1(t)\right)dt + \sigma_{r1} dZ_{r1}(t), \\ dr_2(t) = \alpha_{r2}\left(\mu_r - r_2(t)\right)dt + \sigma_{r2} dZ_{r2}(t), \end{cases}$$

where the short-term rate (denoted by r_1) reverts to a long-term rate (denoted by r_2) that is itself stochastic. The long-term rate reverts to an average mean reversion level μ_r. The autoregressive parameters α_{r1}, α_{r2} are mean reversion speeds. The second term on the right-hand side represents the uncertainty in the process: the standard deviations σ_{r1} and σ_{r2} denote the volatility, $dZ_{r1}(t)$ – the shock to the real

short-term rate process which is distributed normally, $dZ_{r2}(t)$ – the shock to the real long-term rate process which is distributed normally. This model allows the possibility of negative real rates. The pricing equation, which is used in generating the price P_r of discount bond at time t that pays one unit in real terms (protected from inflation) at time T, is given by

$$P_r(t,T) = \exp\left[A(T-t) - B_1(T-t)r_1(t) - B_2(T-t)r_2(t)\right], \quad (1)$$

where $A(s)$, $B_1(s)$, $B_2(s)$ are functions of the parameters for real interest rate movements. Their expressions can be found in paper [20]. Of course, once we have obtained prices for real discount bonds, it is then possible to calculate the continuously compounded yield at time t for maturity T

$$R_r(t,T) = -\log(P_r(t,T))/(T-t). \quad (2)$$

The implementation exercise was solved in such way: using short time intervals, the discrete process approximates the continuous process. In financial applications the long-term modeling is required and the use of intervals, such as monthly, is enough appropriate. The model presented here includes a standard Brownian motion, which in the discrete form is represented by ε_{1t}, ε_{2t}. So we get:

$$\begin{cases} \Delta r_{1_t} = \alpha_{\kappa 1}(r_{2_t} - r_{1_t})\Delta t + \sigma_{r1}\varepsilon_{1t}, \\ \Delta r_{2_t} = \alpha_{\kappa 2}(\mu_r - r_{2_t})\Delta t + \sigma_{r2}\varepsilon_{2t}. \end{cases}$$

We rearrange the last equation to show that this process is an autoregressive process:

$$\begin{cases} r_{1t+1} - r_{1_t} = \alpha_{\kappa 1}(r_{2_t} - r_{1_t})\Delta t + \sigma_{r1}\varepsilon_{1t}, \\ r_{2t+1} - r_{2_t} = \alpha_{\kappa 2}(\mu_r - r_{2_t})\Delta t + \sigma_{r2}\varepsilon_{2t}; \end{cases}$$

$$\begin{cases} r_{1t+1} = r_{1_t} + (\alpha_{\kappa 1} r_{2_t} - \alpha_{\kappa 1} r_{1_t})\Delta t + \sigma_{r1}\varepsilon_{1t}, \\ r_{2t+1} = r_{2_t} + (\alpha_{\kappa 2}\mu_r - \alpha_{\kappa 2} r_{2_t})\Delta t + \sigma_{r2}\varepsilon_{2t}; \end{cases}$$

and

$$\begin{cases} r_{1t+1} = \alpha_{\kappa 1} r_{2t}\Delta t + (1 - \alpha_{\kappa 1}\Delta t) r_{1t} + \sigma_{r1}\varepsilon_{1t}, \\ r_{2t+1} = \alpha_{\kappa 2}\mu_r\Delta t + (1 - \alpha_{\kappa 2}\Delta t) r_{2_t} + \sigma_{r2}\varepsilon_{2t}. \end{cases} \quad (3)$$

Equation (3) shows that the short rate $r_{1_{t+1}}$ is a weighted average between the current level r_{1_t} and the long rate r_{2_t}. The long rate $r_{2_{t+1}}$ is itself a weighted average of the long-term mean μ_r and its current value r_{2_t}. Equation (3) can be used in order to estimate the parameters of the model. In this paper, we don't deal with estimation procedure, and the parameters will be used with reference to HMT work [20].

In simulation, after the discretization procedure we get the discrete samples, taken from these stochastic equations and representing plausible scenarios for uncertain variables over the planning period. Simulation procedure of two factor Ornstein-Uhlenbeck process is given below:

(a) generate $\varepsilon_i \sim N(0,1)$, multiply it by $\sqrt{\Delta t}$;

(b) simulate long-term real interest rate using second equation of (3) formula ;

(c) simulate short-term interest rate using first equation of (3) formula ;

(d) determine maturity time T for discount bonds ;

(e) use analytical expressions (1) for the real spot rate and real forward rate at any term T ;

(f) calculate discount bond return at time t for maturity T using (2) formula.

Exactly the same model structure is used to model the behavior of the short-term inflation rate (denoted by q_1) as it was used for real short-term interest rate r_1, but with parameters for inflation process. A term structure for inflation expectations $P_q(t,T)$ can be inferred from the current instantaneous inflation rate q_1 and q_2, using pricing equation (1). Combining a term structure of real interest rate and inflation expectations it is possible to derive a term structure for nominal interest rates: $P_{nom}(t,T) = P_r(t,T)P_q(t,T)$.

As it was discussed, in scenario generator the financial variables have to be projected in such way as to reflect the appropriate interdependencies between them. It is reasonable to consider the case when interest rates and inflation rates move together: short-term real interest rate correlates with short-term inflation rate, and long-term real interest rate correlates with long-term inflation rate. Interdependencies among these variables are identified through Wiener processes $dZ_i(t)$, $i = \overline{1,4}$. The cumulative distribution function of Wiener process is

$$F_{Z_t}(y;t) = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{y} \exp\left(-\frac{s^2}{2t}\right) = \Phi\left(\frac{y}{\sqrt{t}}\right),$$

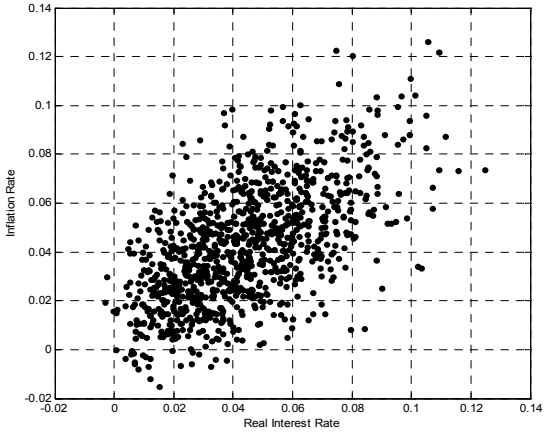
or $dZ_i(t) \sim N(0, dt)$, $i = \overline{1,4}$. Thus, to model the dependence between stochastic drivers, we construct the joint distribution F by linking these marginal distributions through the copula function. We get

$$F_t(y_1, y_2, y_3, y_4) = C\left(\Phi\left(\frac{y_1}{\sqrt{t}}\right), \Phi\left(\frac{y_2}{\sqrt{t}}\right), \Phi\left(\frac{y_3}{\sqrt{t}}\right)\Phi\left(\frac{y_4}{\sqrt{t}}\right)\right),$$

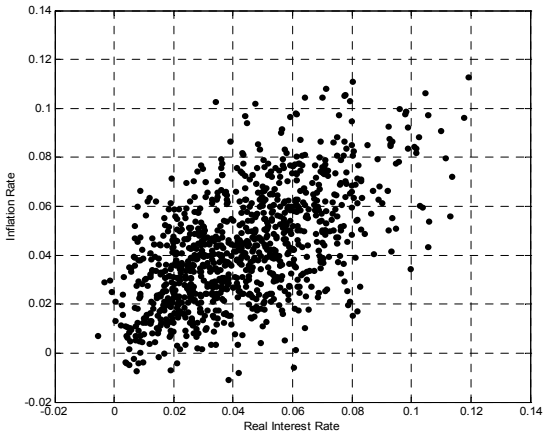
where C is the 4-dimensional copula function. In this work, two copula functions are considered: Gaussian copula and Student's t_2 copula. Their simulation algorithms are given in Section B.

At this moment, let assume that a matrix $Cor^{\tau} = \left[cor_{ij}^{\tau}\right]$, $-1 \le cor_{ij}^{\tau} \le 1$, $i, j = \overline{1,4}$ of Kendall's tau correlations has already been assessed, which denotes rank-order correlations between two random variables. In HMT model, Kendall's tau correlation coefficient is set equal to 0.25 between short-term real interest rate and short-term inflation rate, and it is set equal to 0.25 between long-term real interest rate and long-term inflation rate.

HMT model is used to simulate 1, 3, 5, 7, 10 year coupon bond yields over a horizon of 20 years with time increments of one month. The initial parameters are set with the reference to the Hibbert's et al. work. The conditions about the environment are assumed as follows: inflation level is 2.5%, long-term inflation level is 2.83%, current 3-month T-bill norm is 5% and current 10-year T-bond yield 5.58%. The lower bounds on the levels of inflation and of real interest rates are placed to ensure that negative rates don't appear. At the output of this scenario generator the data consisted of a finite number of scenarios ($S = 1000$), representing the realizations of discount bond yields. In Fig. 11 the dependence structures between simulated values of real interest rate and inflation rate are displayed in the time moment $t = 20$ years.



(a) Gaussian dependence structure



(b) Student's t_2 dependence structure

Fig. 11. Dependence structures between simulated values

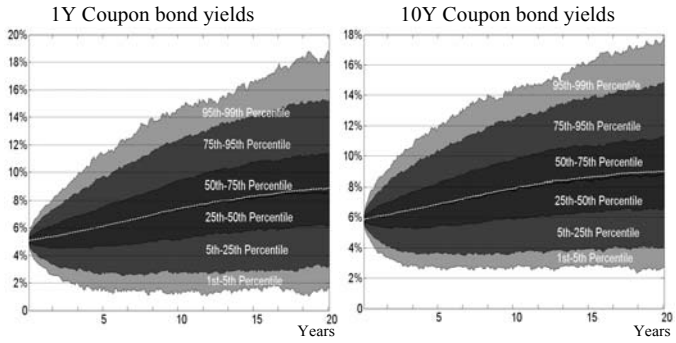
Table I. Dimension of scenario fan

	Nodes	Time periods	Scenarios
Scenario fan of discount bond yields	240000	240	1000

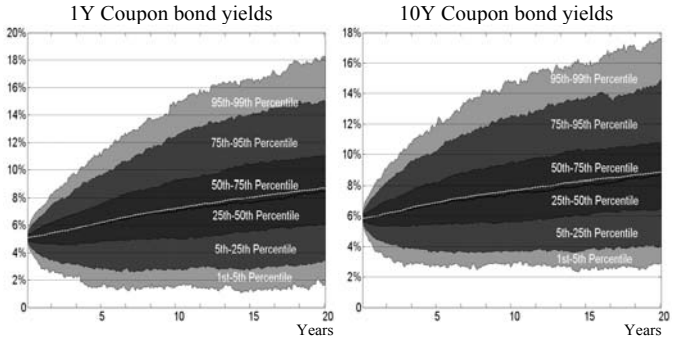
In Fig. 11a) the dependence structure is determined by the correlation matrix *Cor* for Gaussian copula, in Fig. 11b) the dependence is conditioned by the correlation matrix *Cor* and by degrees of freedom $\upsilon = 2$ for Student's t-copula. The lower tail dependence is limited by the lower bounds on the levels of inflation and real interest rates. The upper tail dependence differs with respect to the employed copula.

Each scenario fan of 1, 3, 5, 7, 10 year coupon bond yields generated by scenario generator is described by its dimension. The dimension of the particular scenario fan is given in Table I.

The scenario fan is illustrated using the "funnel of doubt" plot (see Fig. 12), resulting from uncertainty in the future values. In the following analysis, 1-year and 10-year discount bond returns are considered. The "funnel of doubts" graph displays the 1st, 5th, 25th, 50th, 75th, 95th, 99th percentile values and the mean sample value (light dashed line). The spread around its median expands as the time increases, carrying a certain risk of uncertainty that increases with time, but tends to stabilize at the end of time horizon, which is the effect of mean reversion value. The assumption of avoiding negative values of nominal interest rate determines that the expected value of discount bond yields drifts up over time. VaR (Value-at-Risk) type conclusion is that in $100(1-p)\%$ of the cases the yield is higher or equal to VaR_p value (vertical axis), where $0 < p < 1$ is a percentile value. The spread of 10-year discount bond



(a) Gaussian dependence structure



(b) Student's t_2 dependence structure

Fig. 12. Scenario fan of 1Y and 10Y Coupon bond yields

yields is less than the spread of 1-year discount bond yields, because of the effect of mean reversion. Some of statistical characteristics, mean value, dispersion, 1st and 99th percentiles of 1-year and 10-year discount bond returns are calculated for the evaluation of generated scenarios (see Table II – Table III).

The simulated scenario fan is aimed to transform to the scenario trees with different number of stages and with different branching factor, employing the clustering algorithm discussed in Section IV. The number of stages depends on the number of decision moments. The branching scheme of the scenario tree influences the number of clusters. For instance, we choose the number of scenarios equal to $K = 2$ and $K = 3$ which is generated per scenario tree node. Two types of scenario trees are generated for the analysis: 3-stage scenario tree with decisions at $t = 10, 20$ and 5-stage scenario tree with decisions at $t = 5, 10, 15, 20$. Table IV shows the dimension of scenario trees for the cases $K = 2$ and $K = 3$. It shows that the dimension of scenario fan is notably reduced while transforming the scenario fan to the scenario tree.

In the following analysis, we aim to investigate how dependence structure affects the values of target variables and the structure of scenario tree. The mean value, dispersion, 1st and 99th percentiles of 1-year and 10-year discount bond returns are calculated for the evaluation of generated scenario trees (see Table V – Table VIII).

Table II. Characteristics of scenario fan

Gaussian dependence		Decision moments, in Years			
		t=5	t=10	t=15	t=20
Scenario fan of 1Y coupon bond return, %	Mean	6.13	7.26	8.17	8.76
	Dispersion	0.06	0.10	0.14	0.16
	1st Percentile	0.78	0.87	0.95	0.91
	99th Percentile	11.84	15.65	18.35	19.25
Scenario fan of 10Y coupon bond return, %	Mean	6.82	7.72	8.50	8.91
	Dispersion	0.05	0.09	0.11	0.12
	1st Percentile	2.49	2.59	2.38	2.71
	99th Percentile	12.69	15.18	17.43	18.31

Table III. Characteristics of scenario fan

Student's t_2 dependence		Decision moments, in Years			
		t=5	t=10	t=15	t=20
Scenario fan of 1Y coupon bond return, %	Mean	6.14	7.15	8.01	8.65
	Dispersion	0.05	0.10	0.13	0.16
	1st Percentile	1.08	1.19	1.25	1.23
	99th Percentile	11.27	14.64	17.50	19.10
Scenario fan of 10Y coupon bond return, %	Mean	6.84	7.70	8.32	8.76
	Dispersion	0.05	0.08	0.10	0.13
	1st Percentile	2.42	2.51	2.63	2.34
	99th Percentile	12.11	14.91	16.66	17.81

Table IV. Dimension of scenario trees

	K=2		K=3	
	Nodes	Scenarios	Nodes	Scenarios
3-stage tree	7	4	13	9
5-stage tree	31	16	121	81

Table V. Characteristics of scenario trees when $K = 2$

Gaussian dependence			Decision moments, in Years			
			t=5	t=10	t=15	t=20
1Y coupon bond return, %	3-stage tree	Mean	–	7.27	–	8.77
		Dispersion	–	0.05	–	0.09
		1st Percentile	–	5.55	–	4.85
		99th Percentile	–	10.05	–	14.72
	5-stage tree	Mean	6.16	7.27	8.18	8.77
		Dispersion	0.01	0.06	0.11	0.12
		1st Percentile	5.39	4.15	3.93	3.95
		99th Percentile	7.39	12.08	15.29	17.81
10Y coupon bond return, %	3-stage tree	Mean	–	7.74	–	8.93
		Dispersion	–	0.04	–	0.07
		1st Percentile	–	6.09	–	5.57
		99th Percentile	–	10.25	–	13.63
	5-stage tree	Mean	6.84	7.74	8.49	8.93
		Dispersion	0.01	0.05	0.08	0.09
		1st Percentile	5.99	5.20	4.93	4.66
		99th Percentile	8.13	12.37	14.56	17.28

Table VI. Characteristics of scenario trees when $K = 3$

Gaussian dependence			Decision moments, in Years			
			t=5	t=10	t=15	t=20
1Y coupon bond return, %	3-stage tree	Mean	–	7.27	–	8.77
		Dispersion	–	0.06	–	0.11
		1st Percentile	–	4.78	–	4.46
		99th Percentile	–	11.54	–	16.33
	5-stage tree	Mean	6.16	7.27	8.18	8.77
		Dispersion	0.01	0.08	0.13	0.14
		1st Percentile	5.04	3.28	1.97	2.52
		99th Percentile	8.19	13.74	18.70	18.97
10Y coupon bond return, %	3-stage tree	Mean	–	7.74	–	8.93
		Dispersion	–	0.05	–	0.08
		1st Percentile	–	5.46	–	5.40
		99th Percentile	–	11.95	–	15.72
	5-stage tree	Mean	6.84	7.74	8.49	8.93
		Dispersion	0.01	0.06	0.09	0.10
		1st Percentile	5.74	4.36	3.35	3.55
		99th Percentile	9.23	14.02	16.61	18.09

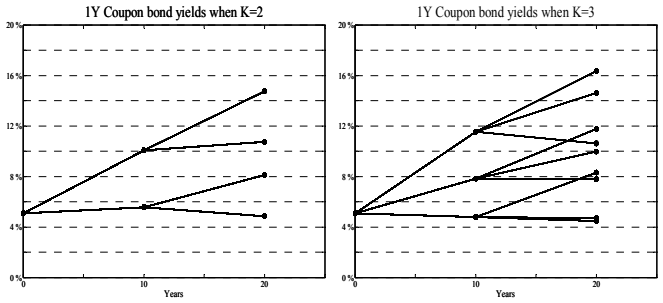
Table VII. Characteristics of scenario trees when $K = 2$

Student's t_2-dependence			Decision moments, in Years			
			t=5	t=10	t=15	t=20
1Y coupon bond return, %	3-stage tree	Mean	–	7.16	–	8.66
		Dispersion	–	0.04	–	0.09
		1st Percentile	–	5.49	–	5.16
		99th Percentile	–	9.60	–	15.33
	5-stage tree	Mean	6.15	7.16	8.02	8.66
		Dispersion	0.01	0.06	0.10	0.12
		1st Percentile	5.44	3.89	3.34	3.23
		99th Percentile	7.19	12.11	14.75	17.34
10Y coupon bond return, %	3-stage tree	Mean	–	7.72	–	8.76
		Dispersion	–	0.04	–	0.07
		1st Percentile	–	6.17	–	6.03
		99th Percentile	–	10.02	–	14.10
	5-stage tree	Mean	6.87	7.72	8.32	8.76
		Dispersion	0.01	0.05	0.08	0.09
		1st Percentile	6.08	4.76	4.25	4.65
		99th Percentile	8.04	12.35	15.71	15.00

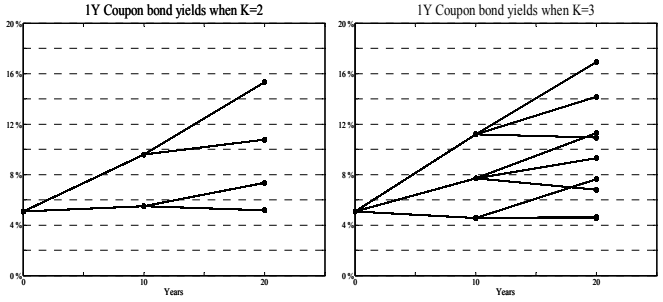
Table VIII. Characteristics of scenario trees when $K = 3$

Student's t_2-dependence			Decision moments, in Years			
			t=5	t=10	t=15	t=20
1Y coupon bond return, %	3-stage tree	Mean	–	7.16	–	8.66
		Dispersion	–	0.06	–	0.11
		1st Percentile	–	4.56	–	4.53
		99th Percentile	–	11.17	–	16.93
	5-stage tree	Mean	6.15	7.16	8.02	8.66
		Dispersion	0.01	0.07	0.11	0.14
		1st Percentile	5.03	3.17	2.29	2.49
		99th Percentile	7.81	13.66	15.66	17.90
10Y coupon bond return, %	3-stage tree	Mean	–	7.72	–	8.76
		Dispersion	–	0.05	–	0.09
		1st Percentile	–	5.39	–	5.06
		99th Percentile	–	11.40	–	15.46
	5-stage tree	Mean	6.87	7.72	8.32	8.76
		Dispersion	0.01	0.06	0.09	0.11
		1st Percentile	5.65	4.16	3.41	2.98
		99th Percentile	8.63	13.62	15.18	16.12

The remarks on obtained results are as follows. It turns out that for a larger branching factor K, the data of discount bond returns become more diverse, the interval between the 1st and the 99th percentiles becomes wider, but the mean value remains the same. The same effect is observed when more decision dates are defined. It holds for both Gaussian copula and Student's t_2-copula correlated data. The initial fan of individual scenarios and the constructed scenario trees show that the data obtained under Student's t_2-dependence have smaller mean value, representing the smaller bond returns than data obtained under Gaussian dependence. These inferences can be approved from the graphical representation of constructed scenario trees (see Fig. 13 – Fig. 16).

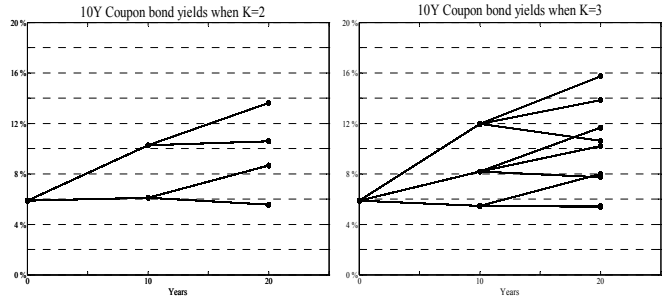


(a) Gaussian dependence structure

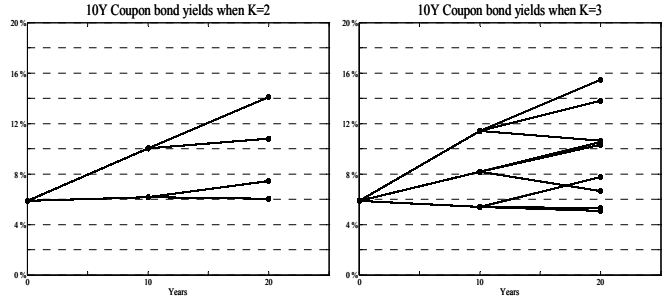


(b) Student's t_2 dependence structure

Fig. 13. 3-stage scenario trees of 1Y Coupon bond yields with decisions at t={10,20} years

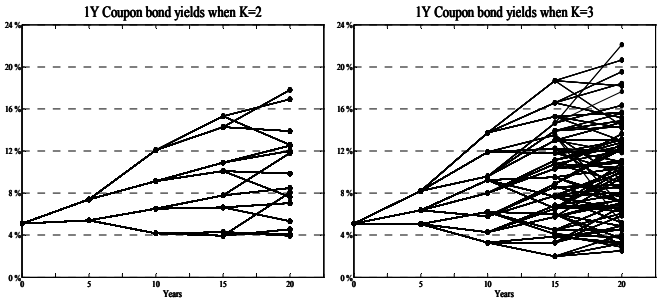


(a) Gaussian dependence structure

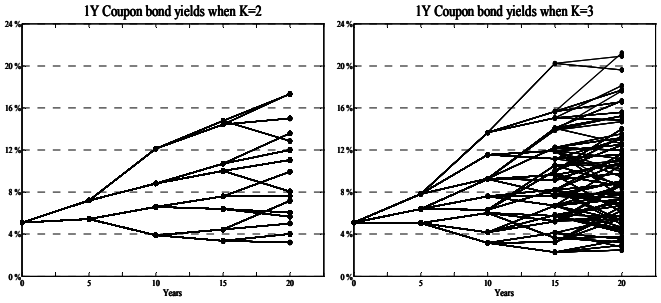


(b) Student's t_2 dependence structure

Fig. 14. 3-stage scenario trees of 10Y Coupon bond yields with decisions at t={10,20} years

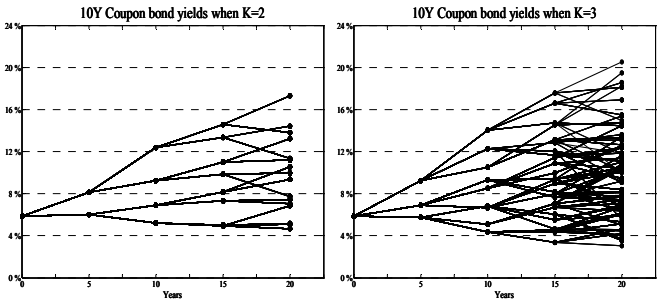


(a) Gaussian dependence structure

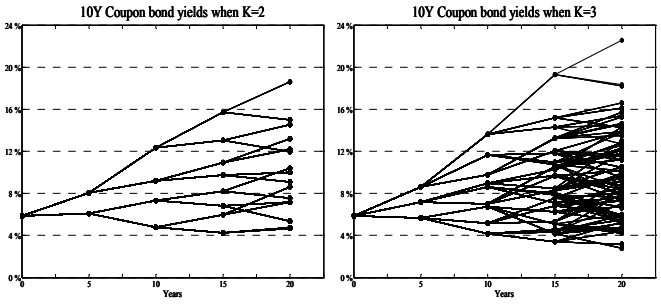


(b) Student's t_2 dependence structure

Fig. 15. 5-stage scenario trees of 1Y Coupon bond yields with decisions at t={5,10,15,20} years



(a) Gaussian dependence structure

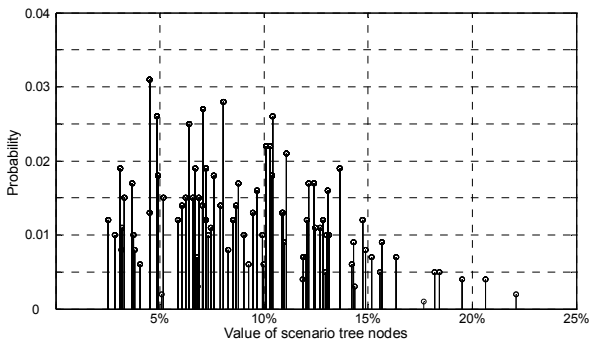


(b) Student's t_2 dependence structure

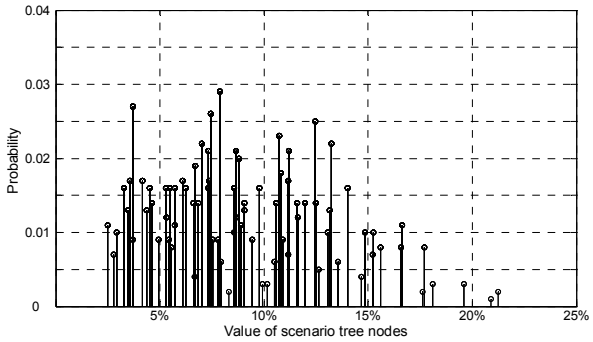
Fig. 16. 5-stage scenario trees of 10Y Coupon bond yields with decisions at t={5,10,15,20} years

Scenario tree with a higher branching factor lets to model more extreme scenarios. Using of Student's t_2-copula as dependence measure between real interest rate and inflation rate has effect to obtain lesser values of discount bond yields comparing with the case when the Gaussian copula is used.

In the scenario fan, each scenario has the same probability. If 1000 scenarios are generated, then probability of any scenario



(a) Gaussian dependence structure



(b) Student's t_2 dependence structure

Fig. 17. Values of 1Y Coupon bond yields with their probability

is equal to 10^{-3}. But this is not the case for scenario tree, where probability of each scenario is conditioned by cluster size. For instance, let take the scenario tree with $K = 3$ branching factor and decisions at $t = \{5,10,15,20\}$ years. Fig. 17 depicts the distribution of 1Y Coupon bond yields under different dependence structures at time $t = 20$. It shows the relationship between value of random variable (node of scenario tree) and its probability.

VI. Concluding Remarks

In the present paper we described the procedure based on both simulation and clustering to generate the scenario trees out of data paths. The computational experiment showed that the size of generated scenario trees is much smaller than the dimension of scenario fan, and nevertheless, they are good approximations with respect to the Euclidean distance used to measure the time-dependent data paths. Answering to our question, does the copula features are captured in the approximate representation of uncertainty in the form of scenario tree, we conclude that different dependence structures with the same correlation coefficient between stochastic variables affect the structure of multistage scenario tree. The accomplished analysis of scenario trees shows that scenario trees generated from dependent data based on Student's t_2-copula are more extreme than generated from dependent data employing Gaussian copula. The effect of using Student's t_2-copula as dependence measure between real interest rate and

inflation rate is to decrease value of discount bond yields. It results from the feature that using Gaussian copula the extreme events are independent, so we don't get really extreme scenarios.

Future research on this topic includes the improving clustering approach by adding some constraints on cluster's size, on cluster's character. Then, the evaluation of quality/stability of scenario generation method for a given stochastic program can be considered.

REFERENCES

[1] L. Y. Yu, X. D. Ji, and S. Y. Wang, "Stochastic programming models in financial optimization: survey," *AMO – Advanced Modeling and Optimization*, vol. 5(1), 2003, pp. 1–26.

[2] J. Dupačová, J. Hurt, and J. Štěpán, *Applied Optimization 75: Stochastic Modeling in Economics and Finance*. Dordrecht, Holland: Kluwer Academic Publishers, 2002, ch. 3.

[3] S. Mitra, "Scenario generation for stochastic programming," White Paper, Optirisk Systems, UK, 2006.

[4] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming," *Mathematical Programming*, vol. 95(3), 2003, pp. 493–511.

[5] K. Høyland and S. W. Wallace, "Generating scenario trees for multistage decision problems," *Management Science*, vol. 47(2), 2001, pp. 295–307.

[6] H. Heitsch and W. Römisch, „Generation of multivariate scenario trees to model stochasticity in power management," in *IEEE St. Petersburg PowerTech Proceedings*, Russia, 2005.

[7] G. Pflug, "Scenario tree generation for multiperiod financial optimization by optimal discretization," *Mathematical Programming*, vol. 89, 2001, pp. 251–271.

[8] J. Dupačová, G. Consigli, and S. W. Wallace, "Scenarios for multistage stochastic programs," *Annals of Operations Research*, vol. 100, 2000, pp. 25–53.

[9] P. Embrechts, A. McNeil, and D. Straumann, „Correlation and dependency in risk management: properties and pitfalls," in *Risk management: value at risk and beyond*, M. A. H. Dempster, Ed. UK: Cambridge University Press, 2002, pp. 176–224.

[10] K. Aas, "Modelling the dependence structure of financial assets: a survey of four copulas," Research Report, SAMBA/22/04, Norwegian Computing Center, Norway, 2004.

[11] N. D. Domenica, G. Birbilis, G. Mitra, and P. Valente, "Stochastic programming and scenario generation within a simulation framework: an information systems perspective," Technical Report, CARISMA, UK, 2003.

[12] J. Dupačová, "Stochastic programming: approximation via scenarios," *Aportaciones Mathematicas, Ser. Communicationes*, vol. 24, 1998, pp. 77–94.

[13] P. Embrechts, F. Lindskog, and A. McNeil, "Modelling dependence with copulas and applications to risk management," in Rachev, S. (Ed.) *Handbook of Heavy Tailed Distributions in Finance*, Elsevier, 2001, ch. 8, pp. 329-384.

[14] W. Hu, "Calibration of multivariate generalized hyperbolic distributions using the EM algorithm, with applications in risk management, portfolio optimization and portfolio credit risk," PhD Thesis, Florida State University, 2005.

[15] S. Daul, E. DeGiorgi, F. Lindskog, and A.J. McNeil, "The grouped t-copula with an application to credit risk", *RISK*, 16, 2003, pp. 73–76.

[16] O. Roch and A. Alegrea, "Testing the bivariate distribution of daily equity returns using copulas. An application to the Spanish stock market," *Computational Statistics & Data Analysis,* 51(2), 2006, pp. 1312–1329.

[17] M. Kaut and S. W. Wallace, "Shape-based scenario generation using copulas," Stochastic Programming E-Print Series, vol. 19, 2006. Available: http://edoc.hu-berlin.de/docviews/abstract.php?lang=ger&id=27659

[18] Stochastic Programming Community Home Page sponsored by the Committee on Stochastic Programming, Online Resource. Available: http://www.stoprog.org/

[19] L. Kaufmann and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Canada: Wiley-Interscience, 1990, ch. 3.

[20] J. Hibbert, P. Mowbray, and C. Turnbull, "A stochastic asset model & calibration for long-term financial planning purposes," Technical Report, Barrie&Hibbert Limited, UK, 2001.