Universidade Federal do Rio Grande do Sul

Escola de Administração

Programa de Pós-Graduação em Administração

Alan Delgado de Oliveira

Essays on Multistage Stochastic Programming applied to Asset Liability Management

Porto Alegre, May 21, 2018

Delgado de Oliveira, Alan
Essays on Multistage Stochastic Programming applied to Asset Liability Management/ Alan Delgado de Oliveira. – , Porto Alegre, May 21, 2018-
99 p. : il. (algumas color.) ; 30 cm.

Supervisor: Tiago Pascoal Filomena, PhD.

Tese (Doutorado) – Universidade Federal do Rio Grande do Sul
Escola de Administração
Programa de Pós-Graduação em Administração, Porto Alegre, May 21, 2018.

1. Multistage Stochastic Programming. 2. ALM. 2. Scenario Generation. I. Tiago Pascoal Filomena. II. Universidade Federal do Rio Grande do Sul. III. Programa de Pós Graduação em Administração. IV. Essays on Multistage Stochastic Programming applied to Asset Liability Management

Alan Delgado de Oliveira

Essays on Multistage Stochastic Programming applied to Asset Liability Management

Dissertation for Doctorate degree in Business Administration at Business School of Federal University of Rio Grande do Sul.

Tiago Pascoal Filomena, PhD.
Doctorate Dissertation Advisor

Marcelo Carvalho Griebeler, PhD.
Committee Member

Marcelo Brutti Righi, PhD.
Committee Member

Luciano Ferreira, PhD.
Committee Member

Miguel Afonso Selitto, PhD.
Committee Member

Porto Alegre, May 21, 2018

Acknowledgements

Os resultados aqui apresentados refletem a contribuição de todas as pessoas que fizeram parte da minha vida ao longo desses últimos 5 anos. De diferentes formas, elas me apoiaram, acreditaram em mim, me ensinaram, me entenderam, me levaram a refletir, ou seja, colaboraram para que eu me tornasse um profissional melhor, um pesquisador melhor, um professor melhor, uma cristão melhor.

Sendo assim, gostaria de mencionar primeiramente aquele que me resgatou quando eu mesmo não acreditava mais em mim: Jesus Cristo de Nazaré. Com seu espírito santo vivo e verdadeiro no meu interior, me deu vida e paz para que eu pudesse vencer cada desafio que se colocava à minha frente. Além dele, deixo meu reconhecimento à minha querida esposa. Andressa, você lutou incansavelmente ao meu lado, participando de cada momento como se estivesse no meu lugar, e muitas vezes se resignou para que eu pudesse me dedicar a esse grandioso projeto. Querida, essa tese é tanto sua quanto minha porque nela também estão as tuas lagrimas e teu suor. Também menciono meus pais (Nelson e Rosangela), meu sogro e minha sogra (Carlos e Márcia), meu irmão e sua esposa (Antonio e Alessandra), minha irmã e seu marido (Angélica e Márcio), e minha cunhada e seu noivo (Adryanne e Leonardo) que sempre tiveram dispostos a me ajudar e sempre se fizeram presentes. Além deles, registro um agradecimento especial ao Dr. Rogério Belle e a sua família que sempre me inspiraram a perseguir a carreira acadêmica e a quem sigo como modelo de fé e integridade.

Nesse período, também tive a oportunidade de conviver com grandes mestres na academia. Essas pessoas me enriqueceram com a sua postura profissional, ética e exemplo. Eles também se dispuseram a dividir o seu tempo comigo. Jamais vou me esquecer das muitas conversas que tive com meu orientador, Dr. Tiago Filomena. Ele sempre esteve com a porta da sua sala aberta para me receber. Não se limitou a me orientar, mas participou como um verdadeiro amigo de cada momento da minha carreira desde que nos conhecemos. Professor, não tenho palavras para agradecer o que fizeste por mim, desde a entrevista na seleção do mestrado sempre acreditaste no meu potencial. Gostaria de referenciar outros grandes professores com o qual tive o prazer de conviver e dialogar na Universidade: Dr. João Becker, Dr. Denis, Dr. Luciano, Dr. Marcelo Brutti, Dra. Denise e Dr. Marcelo Perlin. Além deles, pude trabalhar com outros professores exemplares como o Professor Dr. Miguel Lejeune que me oportunizou participar do meu primeiro projeto internacional. Também menciono aqueles que farão parte da banca examinadora deste documento: Prof. Dr. Miguel Sellito, Prof. Dr. Marcelo Carvalho Griebler juntamente com os professores Dr. Marcelo Righi Brutti, Dr. Luciano Ferreira desde já deixo o meu obrigado pelos comentários e correções que somente acrescentam valor a esse trabalho.

Gostaria de registar também a minha gratidão a alguns amigos e colegas que muitas vezes me ajudaram: Diego Patrick e a Cibele, Thiago Seleme e a Grace, Tiago Rosa e a Kelly, Renan Prestes e a Dani, Tiago Souza e a Simone, Paulo e Jana, Leonardo Riegel e Pablo. Neste período, também fui agraciado com o suporte financeiro do governo brasileiro através de uma de suas agências de fomento, a CAPES. Eu obtive um recurso mensal por meio da demanda social e a oportunidade de realizar o doutorado sanduíche. Por isso, agradeço ao povo brasileiro e aos seus governantes pelo investimento realizado em mim. Por fim, agradeço à Universidade George Washington por me receber pelo período de 4 meses sob orientação do professor Dr. Miguel Lejeune.

Certamente, me faltaria espaço para mencionar todas as pessoas que de alguma forma participaram dessa conquista, mas a cada um destes também deixo minha gratidão. Levo a todos no meu coração. Grande Abraço.

Abstract

Uncertainty is a key element of reality. Thus, it becomes natural that the search for methods allows us to represent the unknown in mathematical terms. These problems originate a large class of probabilistic programs recognized as stochastic programming models. They are more realistic than deterministic ones, and their aim is to incorporate uncertainty into their definitions. This dissertation approaches the probabilistic problem class of multistage stochastic problems with chance constraints and joint-chance constraints. Initially, we propose a multistage stochastic asset liability management (ALM) model for a Brazilian pension fund industry. Our model is formalized in compliance with the Brazilian laws and policies. Next, given the relevance of the input parameters for these optimization models, we turn our attention to different sampling models, which compose the discretization process of these stochastic models. We check how these different sampling methodologies impact on the final solution and the portfolio allocation, outlining good options for ALM models. Finally, we propose a framework for the scenario-tree generation and optimization of multistage stochastic programming problems. Relying on the Knuth transform, we generate the scenario trees, taking advantage of the left-child, right-sibling representation, which makes the simulation more efficient in terms of time and the number of scenarios. We also formalize an ALM model reformulation based on implicit extensive form for the optimization model. This technique is designed by the definition of a filtration process with bundles, and coded with the support of an algebraic modeling language. The efficiency of this methodology is tested in a multistage stochastic ALM model with joint-chance constraints. Our framework makes it possible to reach the optimal solution for trees with a reasonable number of scenarios.

Keywords: Multistage Stochastic Programming. Asset Liability Management. Scenario Generation. Chance Constraint Programming.

Resumo

A incerteza é um elemento fundamental da realidade. Então, torna-se natural a busca por métodos que nos permitam representar o desconhecido em termos matemáticos. Esses problemas originam uma grande classe de programas probabilísticos reconhecidos como modelos de programação estocástica. Eles são mais realísticos que os modelos determinísticos, e tem por objetivo incorporar a incerteza em suas definições. Essa tese aborda os problemas probabilísticos da classe de problemas de multi-estágio com incerteza e com restrições probabilísticas e com restrições probabilísticas conjuntas. Inicialmente, nós propomos um modelo de administração de ativos e passivos multi-estágio estocástico para a indústria de fundos de pensão brasileira. Nosso modelo é formalizado em conformidade com a leis e políticas brasileiras. A seguir, dada a relevância dos dados de entrada para esses modelos de otimização, tornamos nossa atenção às diferentes técnicas de amostragem. Elas compõem o processo de discretização desses modelos estocásticos. Nós verificamos como as diferentes metodologias de amostragem impactam a solução final e a alocação do portfólio, destacando boas opções para modelos de administração de ativos e passivos. Finalmente, nós propomos um "framework" para a geração de árvores de cenário e otimização de modelos com incerteza multi-estágio. Baseados na tranformação de Knuth, nós geramos a árvore de cenários considerando a representação filho-esqueda, irmão-direita o que torna a simulação mais eficiente em termos de tempo e de número de cenários. Nós também formalizamos uma reformulação do modelo de administração de ativos e passivos baseada na abordagem extensiva implícita para o modelo de otimização. Essa técnica é projetada pela definição de um processo de filtragem com "bundles"; e codifciada com o auxílio de uma linguagem de modelagem algébrica. A eficiência dessa metodologia é testada em um modelo de administração de ativos e passivos com incerteza com restrições probabilísticas conjuntas. Nosso framework torna possível encontrar a solução ótima para árvores com um número razoável de cenários.

Keywords: Programação Estocástica Multi-estágio. Administração de Ativos e Passivos. Geração de Cenários. Programação com restrições probabilísticas.

Contents

Introduction

This dissertation proposes an investigation into the stochastic optimization field applied to financial models. As the uncertainty is part of all daily decisions, it has been incorporated into mathematical models from different fields. It is possible to find this kind of approach in manufacturing, queueing models, realistic inventory, portfolio selection, finance market prices and economic behavior, marketing models, transportation and communication. These models have been recognized as stochastic programming, (BIRGE; LOUVEAUX, 2011). In other words, the common feature of these applications is the nature of decision-making, in that decisions have to be made without all the information available, i.e. under uncertainty perspective. Some examples are the actual costs of a financial transaction may depend on future interest rates, the production order is related to unknown demands, the seeds cultivation depend on the uncertain climatic conditions.

In our case, we take the asset liability management (ALM) problem as the application for all models presented here. The ALM considers a balance sheet comprised by assets and liabilities, which can be used in different contexts, i.e. a pension fund, bank, insurance company, or any other institution as an enterprise risk management tool, (ZENIOS; ZIEMBA, 2006). Therefore, the model description should be the asset allocation through time and based upon distinct economic environments in order to carry out the liabilities payment. These investment decisions should be done efficiently and dynamically in terms of resources management and risk matching on both sides of a balance sheet (ZIEMBA, 2003). This representation must also consider the particular peculiarities of each domain, such as policies, laws, and cash flow requirements. We take account of these principles to propose a multistage stochastic programming model applied to asset liability problems for the Brazilian context, which is presented in Part I.

The first step of stochastic problem description is to find the appropriate probability model, as we denote the uncertainty through random variables that are incorporated into an optimization model. The search for a probability model requires a definition of a probability space, as these random variable realizations are sampled from that probability space. Additionally, we suppose that we know these random variable distributions, and only their concrete realizations are unknown. Different techniques have already been proposed in the literature to make this sampling, e.g. by having matched state-space distribution moments (DUPAČOVÁ et al., 2000; HØYLAND; WALLACE, 2001; HØYLAND; KAUT; WALLACE, 2003), minimizing Wassertein probability metrics (ROMISCH, 2003; HEITSCH; ROMISCH, 2005; HOCHREITER; PFLUG, 2007), Latin hypercube sampling (MCKAY; BECKMAN; CONOVER, 1979), Voronoi cell sampling (LOHNDORF, 2016), and Resampled average approximation (OLIVEIRA et al., 2017), among others. All of these suggest ways to simulate realistic scenarios, which in multistage problems are usually comprised of data structures denoted as scenario trees, whose aim is to represent and correspond statistically and more plausibly with the mapped environment. In Part II, we make a comparison of several scenarios of tree-sampling methods in order to assess their impact on the ALM problem solutions. We mainly use the Monte Carlo Principle, which defends the convergence of the ideal distribution μ_x with any desired accuracy by a sampling distribution. We generate a dependent stochastic sequence $\xi_1, \xi_2, \ldots, \xi_N$ such as the empirical measure

$$\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\xi_i} \text{ converges weakly to } \mu_x. \tag{1}$$

Here δ_u denotes the point mass at the point u. As x_t may depend on ξ_t, the sequence of decisions is a stochastic process as well.

As the input parameters are defined by simulation, i.e. we have comprised a scenario tree with respective realizations, we have to define the optimization model that is a description of the main features of the application to permit a solver to find the optimal decision in regard to the problem's requirements. Every decision is associated with a cost function. Its role, in uncertain environments, is to determine the expected measure of gain (loss) resulting from each feasible decision. For instance, an investment decision about an asset portfolio, usually, is looking to maximize return or to minimize the risk subjected to a set of constraints that are related to that decision. An optimization problem is characterized by the objective function, for instance, $F(x)$, which is not explicitly known. Additionally, it must be rewritten as a deterministic equivalent (or also a denominated extensive form) problem to allow for the computational simulation. These equivalent models are solved through convergent iterations. The process results in an initial solution $\bar{x}$ for $F(x)$, which is obtained by a sequential decisions vector $x_1, x_2, x_3, \ldots$ which is progressively improved until it converges to the optimizer x^* in a finite number of steps.

As it is impossible to define the truly real costs function for any problem, the real costs of some application and its constraints are approximated by simulation and optimization effectively combined. Therefore, given the original problem P, it is denoted by P_N, i.e. the uncertainty inherent in an original environment is approximated by generating random variables and the optimization is based on them. Initially, we use the simulation to get information about (P) and, after that, we optimize (P_N) to find the best decisions. That interaction between simulation and optimization follows the non-recursive methodology, which is also known as "stochastic counterpart" or "sample path optimization," (PFLUG, 2012). We generate a sequence $\xi_1, \xi_2, \ldots, \xi_N$ of random variables, and approximate it as Eq. 1. The solution of (P_N) is used as an approximative solution of the original problem (P). We depict an example of that approach in Figure 1.

In multistage problems, the elements of sequence random variables $\xi_1, \xi_2, \ldots, \xi_N$ are revealed in distinct time periods, describing the dynamic of information process disclosure inherent to uncertain contexts. It has influence over the decision because the value of the decision vector x_t, chosen at stage τ, may depend on the information (data) available up to time τ, but not on the results of future observations. This timing requirement is known as nonanticipativity, and is algebraically formalized by the filtration process, which is one of probability space components. The nonanticipativity constraints are determined as the new information will be unfolded for the discrete points defined by the elements of that space. Therefore, in the discretization process, the elements of probability space and denominated stages take initial decisions. After that, the new information is unveiled. Thus, some corrective actions are determined in the direction of the objective function's benefit, (KING; WALLACE, 2012). This decision process is known as

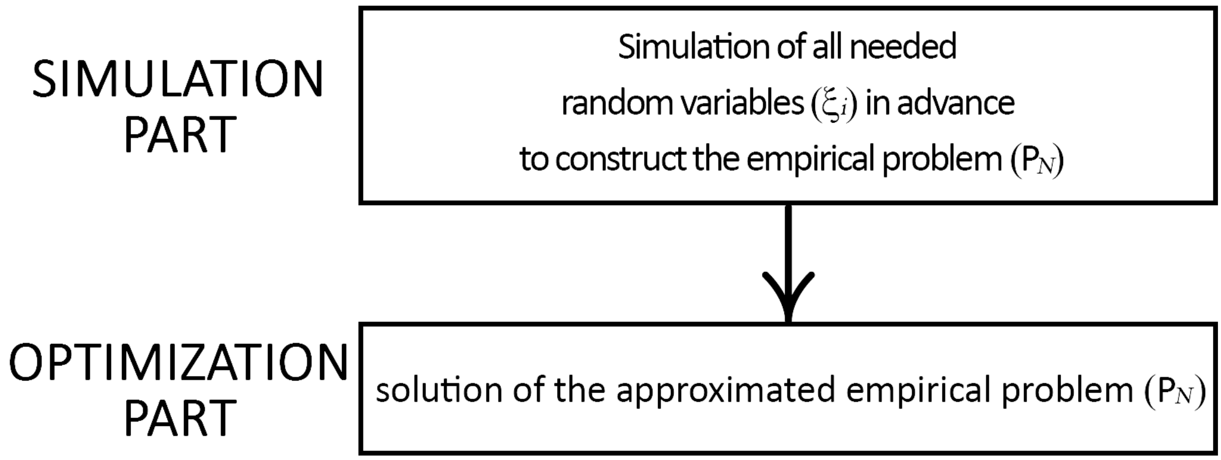


Figure 1 – Non-recursive method diagram from (PFLUG, 2012)

multistage and its formalization can be found in Consigli e Dempster (1998b), Shapiro, Dentcheva e Ruszczynski (2009),

$$\text{decision}(x_1) \rightsquigarrow \text{observation}(\xi_1) \rightsquigarrow \text{decision}(x_2) \rightsquigarrow \\ \ldots \rightsquigarrow \text{observation}(\xi_{\mathcal{T}}) \rightsquigarrow \text{decision}(x_{\mathcal{T}}).$$

Therefore, the information flow and decisions are conditioned by the filtration process, which determines how the interplay of different sampled paths happens according to the decision process above. In Part III, we propose a methodology that employs the space probability filtration and bundles to generate the implicitly deterministic equivalent of a multistage stochastic program. We extend the bundles concept for corresponding nonanticipativity constraints. Our modeling framework includes the stochastic process simulation and the optimization model, and implements a novel implicit deterministic equivalent modeling methodology for both simulation and optimization, also defining an interface between them. In terms of scenario-tree generation, we allocate a multi-way tree represented dynamically by an equivalent binary tree, in such a way that only distinguishable paths are produced and only one database is created, giving origin to a compact event tree. We also formulate a theoretic multistage stochastic model with joint chance constraint, which is introduced in two versions for its corresponding equivalent deterministic. In the implicit extensive form optimization model, the use of bundles guarantees the creation of a unique variable set that is generated without redundancy. We code compact representations of AMLs that determines the interaction between simulation and optimization, as well as handling large-scale multistage stochastic optimization problems very efficiently. This technique is applied in a dynamic multistage stochastic programming ALM with joint chance constraint. Our design brings an effective reduction in terms of computer processing and memory requirement without the need of any relaxation or decomposition mechanism.

In summary, we present the following tree papers that discuss the multistage stochastic problem applied to asset liability management problem:

I. A Multistage Stochastic Programming Asset-Liability Management Model - An Application to the Brazilian Pension Fund Industry

II. Performance comparison between different scenario generation tree approaches applied to asset-liability management

III. Implicit Extensive Form for Multistage Stochastic Programming Models: A New Approach

Part I

A Multistage Stochastic Programming Asset-Liability Management Model - An Application to the Brazilian Pension Fund Industry

Abstract

This paper proposes a multistage stochastic programming approach for the asset-liability management of Brazilian pension funds. We generate asset price scenarios with stochastic differential equations. We use a Geometric Brownian Motion model for stocks and a Cox-Ingersoll-Ross model for fixed income securities. Intertemporal solvency regulatory rules for Brazilian pension funds are considered endogenously in the model and enforced with a combinatorial constraint. A VaR probabilistic constraint is incorporated to obtain a positive funding ratio at each time period with high probability. Our model uses multiple trees to provide a representative characterization of the uncertainty and is not computationally prohibitive. We evaluate the insolvency probability under different initial funding ratios through extensive simulations. The study reveals that the likely decrease of interest rate premiums in the next years will force pension fund managers to significantly change their portfolio strategies. They will have to take more risk in order to deliver the cash-flows required to cover the liabilities and satisfy the regulatory constraints.

keywords:ALM. Brazilian Pension Funds. Stochastic Optimization. Scenario Trees.

Note: this article has been published on Optimization and Engineering v. 18, n. 2, p. 349-368, 2017.

de Oliveira, A. D., Filomena, T. P., Perlin, M. S., Lejeune, M., & de Macedo, G. R. (2017). A multistage stochastic programming asset-liability management model: an application to the Brazilian pension fund industry. Optimization and Engineering, 18(2), 349-368.

1 Introduction

Asset-liability management (ALM) is a classical topic in financial optimization and is increasingly needed for fund managers operating in highly uncertain markets, such as those in developing countries. In simple terms, ALM's central problem is to develop an investment strategy that permits to cover the liabilities over a multi-period horizon (ZIEMBA, 2003). To enable this objective, the left-side (assets) and the right-side (liabilities) of the balance sheet must be matched (ADAM, 2007; MITRA; SCHWAIGER, 2011). ALM models have been used in a variety of environments, ranging from pension funds (JOSA-FOMBELLIDA; RINCÓN-ZAPATERO, 2012; GÜLPINAR; PACHAMANOVA, 2013), insurance companies (FRANGOS; ZENIOS; YAVIN, 2004; CONSIGLIO; SAUNDERS; ZENIOS, 2006; ASIMIT et al., 2014; ASANGA et al., 2014), banks (MUKUDDEM-PETERSEN; PETERSEN, 2008; URYASEV; THEILER; SERRAINO, 2010), corporate and public debt management (DATE; CANEPA; ABDEL-JAWAD, 2011; CONSIGLIO; STAINO, 2012; VALLADÃO; VEIGA, 2014), to personal finance (NIELSEN; POULSEN, 2004; RASMUSSEN; CLAUSEN, 2007; CONSIGLIO; COCCO; ZENIOS, 2007; PEDERSEN; WEISSENSTEINER; POULSEN, 2013). Zenios e Ziemba (2006), Zenios e Ziemba (2007) provide a comprehensive overview of the theoretical and methodological developments in the ALM field and illustrate their application with a few case studies. This study focuses on the modeling of the specific rules and conditions to which the Brazilian pension fund industry is subjected.

Stochastic programming techniques and models have been applied to ALM since the seventies (BRADLEY; CRANE, 1972). Cariño et al. (1994) were probably the first to present a model with commercial application. Subsequently, (BOENDER, 1997) proposed a large-scale model in which heuristics techniques are employed to determine the investment strategy. Ever since, multistage stochastic programming approaches have become a trend see, e.g., Consigli e Dempster (1998b), Kouwenberg (2001). (KOUWENBERG, 2001) focused on the challenge of generating representative scenarios, which is a key aspect in multistage stochastic programming. Stochastic programming ALM models involving non-neutral risk measures and based on the CVaR measure see, e.g.,Rockafellar e Uryasev (2000), Bogentoft, Romeijn e Uryasev (2001), Kilianová e Pflug (2009), Ferstl e Weissensteiner (2011) and on the inclusion of jumps for the asset prices (JOSA-FOMBELLIDA; RINCÓN-ZAPATERO, 2012) have recently been proposed.

However, the legislative side of the ALM problem has seldom been the primary concern of the existing models, especially for emerging markets. Fund managers are bound to comply to laws in their judicial system and, therefore, the practice of ALM should also consider this particular set of restrictions. The objective of this study is to present the regulatory framework faced by the Brazilian pension funds' industry and to develop a multistage stochastic programming model that explicitly accounts for the set of ALM regulatory rules in Brazil. More specifically, we focus on the so-called defined benefit (DB) plan (ZIEMBA, 2003), which is the most commonly used plan in Brazilian public institutions. In DB plans, the benefits received by the members of the plan (i.e., liabilities) are defined in advance which makes the fund liabilities almost deterministic and contrasts with the stochastic nature of assets' returns. Recent changes in the Brazilian capital markets is the main motivation for this application, as presented by Dupačová e Polívka (2009) and Kilianová e Pflug (2009) for other emerging markets. Brazilian pension fund managers have been used to almost exclusively invest in fixed income securities. However, with the capital markets and country developments, the long-term trend of the interest rate is decreasing. In this study, we investigate the possible changes in portfolio allocation due to this new economic environment and to the strict regulation.

Our study proposes a multistage stochastic programming ALM model with chance and combinatorial constraints which is motivated and can be applied by the Brazilian pension fund industry. The chance constraint enforces a Value-at-Risk (VaR) requirement to keep the pension fund solvent across time with a high probability. The combinatorial constraint represents endogenously an intertemporal solvency regulation imposed by the Brazilian pension fund legislation. We construct multiple binary trees and each gives the same importance to catastrophic and normal economic scenarios. Thus, the model tends to be more conservative, an important feature for long-term survivorship in highly volatile environments, such as the Brazilian market.

The scenario generation relies on suggestions given by Kouwenberg (2001) and ans its key ingredients are the use of multiple trees and Stochastic Differential Equations (SDEs) to simulate the asset prices. The fixed income asset prices are simulated with the mean-reverting Cox-Ingersoll-Ross model (CIR), which guarantees the interest rate to be non-negative (COX; INGERSOLL; ROSS, 1985). Stock prices are generated with Geometric Brownian Motion (GBM). Some earlier studies pertaining to portfolio and ALM models have also used SDEs, but they focus on methods in which the portfolio allocation is kept fixed throughout the time (MERTON, 1973; MERTON, 2001; KIM; OMBERG, 1996; MILEVSKY, 1998; WACHTER, 2002). Notably different from the single tree approach typically used in the existing literature see, e.g.,(KOUWENBERG, 2001), we implement an extensive scenario generation method to construct multiple trees. We solve the multistage stochastic programing problem corresponding to each tree and use a variant of the resampling method proposed by Michaud e Michaud (2008) to derive the final investment strategy.

Using empirical data for a specific Brazilian pension fund, we estimate its insolvency probability for different initial funding ratios. The results show that Brazilian pension fund managers shall modify their investment behavior and strategies in the near future: they will be pressured to increase their positions in riskier assets if the long-term downward trend of interest rates gets confirmed. As funds managers become less risk averse, their fund's insolvency probability will increase. However, if pension fund managers decide to keep their current risk profile (in terms of risk allocation and insolvency probability), pension fund's members external contributions would have to be raised in the next years.

The paper is organized as follows. In Chapter 2, we discuss the pension funds industry and the capital markets in Brazil. We motivate and formulate the stochastic programming model in Chapter 3. Chapter 4 describes the scenario generation techniques, while the algorithmic procedure and the data used in the numerical tests are outlined in Chapter 5. The results of the application of the model to the Brazilian market are presented in Chapter 6. Concluding remarks are provided in Chapter 7.

2 Pension Funds and Capital Markets in Brazil

Pension funds play an important role in the Brazilian financial system by promoting financial stability for future retirees. In 2014, according to the Brazilian Association of Closed Supplementary Pension Funds (2014), the total nominal value of Brazilian pension funds assets was approximately 219 billions US dollars. The value was distributed across 317 different pension funds with varying sizes. Most of the pension funds are from private institutions.

In Brazil, every pension fund[1] is under the supervision of the Board of the Pension Funds Management, which is a regulatory authority of the Brazilian National Financial System. The pension fund's auditing activities are managed by the Supplementary Pensions Department that acts as an watchdog for the existing standards. Both are subordinated to the National Monetary Council, the highest level authority in the Brazilian financial system. The private social entities in Brazil are organized in the form of non-profit foundations or civil societies and are accessible only to private employees of a company, group of companies or public employees (federal and state wide). Every investor of a pension fund is called a member.

The main legislation regarding pension's fund asset allocation limits and operation was designed by the National Monetary Council (Brazilian Central Bank, 2012). In Brazil, a pension fund may invest the members' money in the following categories: fixed income, equity, structured investments, investment abroad, properties, and operations with participants. The maximum allocation for each instrument is 100%, 70%, 20%, 10%, 8%, and 15% respectively, and is aimed at controlling the pension's fund financial risk and protecting their members. There is no minimal allocation imposed for any of the instruments.

Solvency across time is another crucial consideration for Brazilian pension funds. According to the Ministry of Social Welfare (2008), the funding ratio, defined as the ratio of current assets to the present value of future liabilities, cannot be smaller than one in more than two consecutive years. The goal of this rule is to protect the members' wealth by ensuring a certain level of liquidity for the pension fund.

Currently, the fixed income allocation is highly predominant in the case of Brazilian pension funds. It can be explained by the high real interest rate that allows managers to reach, in most cases, the actuarial target without taking much risk. In April 2015, the domestic ***real*** short-term interest rate was approximately 6.5% per year, a very high risk premium that discourages managers to take positions in riskier assets.

However, the context in which Brazilian pension fund managers operate is changing. As in the rest of the world, life expectancy (and thereby liabilities of pension funds,) in Brazil has increased, and the high real interest rate tends to normalize at lower levels as the monetary policy instrument reaches the objective of holding the domestic inflation. One of the objectives

[1] Also called in Portuguese *Entidade Fechada de Previdencia Complementar (EFPC)*.

of this paper is to investigate the consequences of the modified socio-economic context on the asset allocation strategies pursued by Brazilian pension funds.

The proposed multistage stochastic programming ALM model captures the economic changes and regulations in Brazil. The scenario generation method accounts for the modifications of the economic environment. The stochastic programming model incorporates constraints specific to the legislative regulations. The model enforces endogenously the intertemporal funding ratio constraint imposed by the current legislation. This makes the proposed model directly applicable by the Brazilian ALM practitioners. We present the specifics of the proposed model in the next chapter.

3 Stochastic ALM Model

The ALM paradigm resides in the allocation of a certain amount of wealth in a number of financial assets $i = 1, \dots, N$ in order to cover the liabilities l_t in each period $t = 1, \dots, T$. The model takes the form of a stochastic and intertemporal dynamic allocation problem due to the randomness of the asset prices and the time-dependency nature of the investment and rebalancing decisions. The investment policy is defined with three sets of variables: X_{its} is the number of shares of asset i to hold in time period t and scenario s, while B_{its} and V_{its} respectively denote the number of shares of i bought and sold in time t and scenario s. The stochastic variable ξ_{it} is the price of asset i in time t and can take a finite number ($s = 1, \dots, S$) of realizations denoted P_{its}. Table 5 presents the model notations.

Table 1 – Notation Summary

Sets - Indices	
t	time index (stage) $t = 0, 1, \dots, T$
i	index of asset classes $i = 1, \dots, N$
s	index of scenarios $s = 1, \dots, S$
Decision variables	
X_{its}	Number of shares of assets i to hold in time t and scenario s
B_{its}	Number of shares of assets i to buy in time t and scenario s
X_{i0}	Number of shares of assets i hold initially ($t = 0$)
V_{its}	Number of shares of assets i to sell in time t and scenario s
C_{ts}	Binary variable taking value 1 if there is underfunding in time t and scenario s and taking value 0 otherwise
Random variables	
ξ_{it}	Random price of asset i in time t
Deterministic parameters	
Q	Initial wealth
α_t	Reliability level in time t
K	Legally required funding ratio
L_t	Present value of future liability $t = t+1, \dots, T$
l_t	Liability to be paid in period t
F_t	Present value of future external contributions, $t = t+1, \dots, T$
f_t	External contributions in each period t
M	Maximum amount of underfunding allowed
ρ	Discounting factor
π	Maximum weight of an asset in the portfolio
p_{ts}	Probability of scenario s at time t
P_{its}	Price of asset i at time t and scenario s
P_{i0}	Known initial ($t = 0$) price of asset i

The model enforces risk and regulatory restrictions and takes the form of a multi-stage stochastic programming problem with chance constraints formulated as follows:

$$\max \qquad \sum_{s=1}^{S}\sum_{i=1}^{N} p_{Ts}P_{iTs}X_{iTs} \tag{3.1}$$

$$\text{s.}t.: Q = \sum_{i=1}^{N} P_{i0}X_{i0} \tag{3.2}$$

$$X_{its} = X_{i(t-1)s} + B_{its} - V_{its},\ t \in 1,\dots,T, i = 1,\dots,N, s = 1,\dots,S \tag{3.3}$$

$$\mathbb{P}\left(\sum_{i=1}^{N} \xi_{it}X_{it} \geq K(L_t - F_t)\right) \geq \alpha_t,\ t = 1,\dots,T \tag{3.4}$$

$$\sum_{i=1}^{N} P_{its}V_{its} - \sum_{i=1}^{N} P_{its}B_{its} + f_t = l_t,\ t = 1,\dots,T, s = 1,\dots,S \tag{3.5}$$

$$X_{its}P_{its} \leq \pi \sum_{i=1}^{N} X_{its}P_{its},\ t = 1,\dots,T, i = 1,\dots,N, s = 1,\dots,S \tag{3.6}$$

$$K(L_t - F_t) - \sum_{i=1}^{N} P_{its}X_{its} \leq MC_{ts},\ t = 1,\dots,T, s = 1,\dots,S \tag{3.7}$$

$$\sum_{j=0}^{2} C_{(t+j)s} \leq 2,\ t = 1,\dots,T-2, s = 1,\dots,S \tag{3.8}$$

$$X_{its}, B_{its}, V_{its} \geq 0,\ i = 1,\dots,N, t = 1,\dots,T, s = 1,\dots,S \tag{3.9}$$

$$C_{ts} \in \{0,1\},\ t = 1,\dots,T, s = 1,\dots,S\ . \tag{3.10}$$

The model maximizes the expected terminal value of the fund (3.1). The objective function reflects the goals of the fund manager who aims at reaching the largest possible gains in the last period while respecting risk, regulatory and liability constraints. At time $t = 0$, there is no uncertainty affecting the allocation of the initial wealth Q to the asset classes and the price P_{i0} of each asset is known in (3.2). The number B_0 of shares bought is equal to the number X_0 of shares detained at the end of the initial period. The linear equalities (3.3) are the share balance constraints and specify that the number of shares X_{its} of asset i in time t and scenario s is equal to the number of shares $X_{i(t-1)s}$ detained at the previous period augmented by the number of purchased shares B_{it} minus those sold V_{it} at time t. Since the asset prices are known and deterministic, we have $X_{i0s} = X_{i0}, \forall s$. The chance constraints (3.4) enforces the funding ratio requirements. The funding ratio represents the long-term relation between assets and liabilities. The actual funding ratio β_{ts} of a fund in time t and scenario s is computed as:

$$\beta_{ts} = \frac{F_t + \sum\limits_{i=1}^{N} P_{its}X_{its}}{L_t}, \qquad t \in 1,\dots,T, s \in 1,\dots,S\ , \tag{3.11}$$

where $\sum_{i=1}^{N} P_{its}X_{its}$ is the current asset value of the pension fund in scenario s. L_t and F_t are, respectively, the present value of the future liabilities and contributions discounted by ρ:

$$L_t = \sum_{j=t}^{T} \frac{l_j}{(1+\rho)^{j-t}}, \qquad F_t = \sum_{j=t}^{T} \frac{f_j}{(1+\rho)^{j-t}}.$$

The initial funding ratio is denoted by $\beta_0 = (F_0 + \sum\limits_{i=1}^{N} P_{i0}X_{i0})/L_0$. A value of β smaller than 1 signals that the value of the assets might become insufficient to cover the future liabilities and that the fund might run into solvency issues in the near future. The Brazilian legislation on pension funds requires the use of a discounting factor ρ equal to 5%.

Each probabilistic constraint (3.4) enforces a safety level on the funding ratio and does not allow it to fall below a specified threshold K with some large prescribed probability level α_t. The two parameters K and α_t define the risk-aversion of the asset-liability management policy. The risk-aversion level increases monotonically with the value taken by K and α_t. In general, K is between 1 and 1.5, while α_t is defined on $[0.9, 1)$. The constraints (3.4) can be viewed as some sort of VaR constraints that ensure that the value of the fund is at least equal to KL_t in each period t with probability at least equal to α_t. Their purpose is maintain the long-term solvency level of the fund. As discussed in Chapter 2, the Brazilian legislation defines $K = 1$. Haneveld, Streutker e VAN DER VLERK (2010) considers some more risk-averse values for K (i.e., 1.05 and 1.30). Note that the probabilistic constraints (3.4) are individual ones. Alternatively, we could have used joint chance constraints at each period t. Such an approach would not allow for the funding ratio to fall below a specified threshold K with probability level α_t at t and at any of the earlier periods $t' = 1, \ldots, t-1$. This would enforce stricter requirements, may be appropriate in practice, and is much more difficult to solve.

The stochastic equalities (3.5) are the cash balance constraints and model the relationship between the fund's cash inflows and outflows. Inflows include the sales of assets, yields, and external contributions, while outflows are the payments of liabilities and the purchases of assets. External contributions are payments made by members or sponsors to the pension fund. The stochastic constraints (3.6) are motivated by the Brazilian legislation that defines an upper bound π on the proportion of the fund value invested in a particular asset. Each constraint (3.7) indicates if there is underfunding in period t and scenario s. If this is the case, the binary variable C_{ts} is forced to take value one. The parameter M is a large positive number and represents the maximum acceptable underfunding value. The Brazilian regulation stipulates that the funding ratio of the pension fund must not be below the value of 1 for more than two years in a row. This requirement is enforced by the combinatorial constraints (3.8). They make sure that there is no underfunding in three consecutive periods for every scenario. Constraints (3.9) and (3.10) define the non-negativity and integrality restrictions.

An equivalent mixed-integer programming problem can be formulated for the above multi-stage stochastic programming problem:

$$\begin{aligned} \max \quad & (3.1) \\ \text{s.t.:} \ & (3.2) - (3.3); (3.5) - (3.10) \\ & \sum_{s=1}^{S} p_{ts} C_{t,s} \leq 1 - \alpha_t, t = 1, \ldots, T \end{aligned} \tag{3.12}$$

Noticing that $\sum_{s=1}^{S} p_{ts} = 1$ for $t = 1, \ldots, T$, the knapsack constraints (3.12) ensure that the sum of the probabilities of the scenarios with underfunding is below the complement of the enforced reliability level α_t.

4 Scenario Generation

As discussed by Cariño et al. (1994) and Dupačová e Polívka (2009), the benefits of the insights provided by the model depends heavily on the quality and relevance of the scenarios generated to represent the stochasticity of the assets' prices. Simulating prices properly is essential for the model's performance. In this study, the asset prices follow correlated SDEs of form:

$$d\xi_{it} = \mu(\xi_{it}, t)dt + \sigma(\xi_{it}, t)dW_{it}.$$

W_{it} is a Wiener process normally distributed with mean zero and variance Δ for $t < t + \Delta$. If more than one asset is used in the simulation, we should take into account the returns' correlated errors between the different assets (DEMPSTER; GERMANO; MEDOVA, 2003). The correlation coefficients ρ_{ij} between two assets i and j at any time t are defined by:

$$dW_i \ \cdot \ dW_j = \rho_{ij}dt, \qquad \rho_{ii} = 1, \qquad \forall i, j.$$

We consider two assets in the model: stock (ξ_{1t}) and fixed income (ξ_{2t}). We use the widely known geometric Brownian motion model for stock prices (NEFTCI, 1996; DUFFIE, 2001):

$$d\xi_{1t} = \mu\xi_{1t}dt + \sigma\xi_{1t}dW_{1t}.$$

The GBM offers a known closed form solution:

$$\xi_{1t} = \xi_{1(t-1)}e^{\left(\mu - \frac{1}{2}\sigma^2\right)d_t + \sigma\epsilon\sqrt{d_t}}, \tag{4.1}$$

with $\epsilon \sim N(0, 1)$. For the price of the fixed income asset, we use the Cox-Ingersoll-Ross term structure model (COX; INGERSOLL; ROSS, 1985) to avoid the negative values that interest rate simulations can take with Vasicek (1977). The fixed income asset is represented by:

$$d\xi_{2t} = \alpha(\mu - \xi_{2t})dt + \sqrt{\xi_{2t}}\sigma dW_{2t} \ , \tag{4.2}$$

where ξ_{2t} is the interest rate and $\Lambda \equiv (\alpha, \mu, \sigma)$ are model parameters. The drift function $\mu(\xi_{2t}, \Lambda) = \alpha(\mu - \xi_{2t})$ is linear and presents a mean reverting property, i.e interest rate ξ_{2t} moves in the direction of its mean μ at speed α. The diffusion function $\sigma^2(\xi_{2t}, \Lambda) = \xi_{2t}\sigma^2$ is proportional to the interest rate ξ_{2t} and ensures the interest rate to be always positive. It is important to point out that the simulation model is related to the spot price of two assets only, one stock and one fixed income instrument. Since we only have long positions in the assets and we are not using prices from the other instruments in the yield term structure of the Cox-Ingersoll-Ross model, we have not detected any arbitrage opportunities in our simulations. However, it should be noted that in other applications it might be necessary to ensure that the simulations don't allow for arbitrage opportunities, resulting in unrealistic prices of financial assets. Suggestions on how to handle this issue can be found in (HØYLAND; WALLACE, 2001), (KLAASSEN, 2002) and (CONSIGLIO; CAROLLO; ZENIOS, 2016).

5 Data Structure and Algorithm

We construct multiple multistage binary event trees with T time periods (stages) and S scenarios. Every node in a tree has two successors, except for the leaf nodes at time T. Therefore, the total number of nodes in the tree is $2^{T+1} - 1$. Each path from the root to a leaf node represents a scenario, and the nodes represent decision points. The same probability is attributed to each node at the same time (stage). The pension fund manager makes his/her decisions in time t and scenario s based on the currently available information and future expectations of asset prices. This process is completed in time T. Each node is equally likely, and therefore the same probability is given to normal and extreme scenarios, which makes the approach conservative. This is an important and valuable feature for countries like Brazil where the market volatility is quite high when compared to more developed markets.

In the stochastic programming ALM model, a scenario tree must be constructed to depict as accurately as possible the uncertainty structure. It is important to reduce the bias in the generation of scenarios and trees. If one single tree is constructed, this issue could be to some extent overcome by generating a very large tree. This could however lead to the formulation of an enormous stochastic programming problem that could be extremely difficult to solve even approximately. As another alternative to alleviate the issue of scenario bias in a single tree, Kouwenberg (2001) proposes a random sampling adjustment to control the aleatory nature of the problem. In this context of information uncertainty, we design in this paper a method that is based on the generation of scenarios for multiple trees and the solution of an optimization problem of smaller size for each tree, which permits to keep the computational complexity and the solution times under control. This technique has some similarities with the resampled efficient frontier method proposed by Michaud e Michaud (2008)[1] for the construction of portfolio of risky securities. For ALM purposes, Figueiredo (2011) have also recently adopted a multiple scenario tree approach analyzed with the sample-average approximation presented by Linderoth, Shapiro e Wright (2006). We provide below the pseudo-code of our algorithmic procedure.

Algorithm 1: Routine to generate the ALM results.

Step 1: Define the number θ of trees to be used in the ALM simulation.
Step 2: Use (4.1) and (4.2) to generate the asset prices' scenarios for each path and node of every tree.
Step 3: Solve for each tree the ALM Problem (3.1)-(3.3); (3.5)-(3.10); (3.12).
Step 4: Evaluate the results considering the θ trees. At time zero, the portfolio allocation is the average obtained with the θ trees.

In Step 1, we define the number of trees to be solved for each parametrization see, e.g.,Michaud e Michaud (2008). The value of θ must be sufficiently large so that the portfolio allocations are stable and small enough so that the approach does not become computationally prohibitive. Once the number θ of trees is defined, we generate in Step 2 scenarios for each tree

[1] The resampling method was first published by Michaud (1998).

using the method presented in Chapter 4. In Step 3, we solve to optimality the optimization problem (see Chapter 3) corresponding to each tree. In Step 4, we evaluate the results based on the optimal solutions of the θ trees. A key output of the model is the initial asset allocation, which is obtained by taking the average of the optimal initial asset allocations for each of the θ trees.

The resampling solution Michaud e Michaud (2008) presents advantages and disadvantages. On the positive side, it limits extreme weight allocations that can arise with the classical mean-variance portfolio selection model. Additionally, the constructed portfolio is usually less sensitive to estimation error (FLETCHER; HILLIER, 2001). However, Scherer (2002) raises some criticism to the resampling method. First, it is a heuristic without a theoretical economic rationale. Second, when long and short positions are allowed, the resampled efficient frontier is not necessarily an improvement over the classical efficient frontier (Markowitz, 1952). Scherer (2002) also argues that resampling should be compared to Bayesian methods instead to the traditional mean variance. The literature is still controversial on the pros and cons of the results delivered by resampling (MARKOWITZ; USMEN, 2003; ULF; RAIMOND, 2006; BECKER; GURTLER; HIBBELN, 2015). There have been some recent attempts see, e.g.,Frahm (2015) to develop the theoretical foundations of the portfolio resampling approach. While the resampling method is certainly not exempt from criticism, our motivation to use it lies in the possibility it gives to consider many scenarios while preserving computational tractability.

The method described above is not without resemblance to the sample average approximation (SAA) method. As noted by Kim, Pasupathy e Henderson (2014), the method can be applied when the sample and the true problems both enjoy features that are critical (e.g., continuity, differentiability) for optimization solvers. However, in terms of asymptotic efficiency, it was shown that the standard implementation of the SAA method does not perform as well as stochastic approximation. The performance gap is due to the fixed and very large size of the sample set, which makes extremely difficult to solve the resulting SAA formulation. To circumvent this issue, a method called retrospective approximation retrofit was recently proposed, in which the size of the sample set increases iteratively at a controlled rate. We refer the reader to Kim, Pasupathy e Henderson (2014) for a comprehensive discussion of the principles, advantages, limitations, and implementation of the sample average approximation method.

6 Application to the Brazilian Market

We implement the model presented in Chapters 3 to 5 for a Brazilian pension fund. We use 10-period binary scenario trees with two asset classes. In order to calibrate the simulation of the fixed income asset, we collected data of the 1-month Brazilian LTN (similar to a T-Bill in the USA) as a proxy for the short-term interest rate. For the stocks, we collected the returns for the Brazilian Index IBOVESPA (similar to the S&P 500 in the USA). We use maximum likelihood to estimate the parameters in equations (4.1) and (4.2). The training period is from January 2005 to January 2015. In Table 2, we present some descriptive return statistics for the two asset classes.

Table 2 – Annualized return statistics for the two asset classes.

Asset	Mean %	Std. Deviation%
Stocks	10.11	28.6
Fixed Income	9.6	3.34

Table 2 highlights a peculiarity of the Brazilian capital market. Namely, the fixed income asset has an expected return (9.6%) that is very close to the stocks expected return (10.11%), but has a much lower volatility (i.e., 3.34%) than stocks have (i.e., 28.6%). It is thus easy to understand the preference of Brazilian pension fund managers for fixed income assets as their volatility is much lower than that of the stocks. However, as aforementioned, the fixed income premium is likely to decrease in Brazil, which could impact the pension funds asset allocations. The correlation between both assets is slightly negative and equal to -0.036. Therefore, in a node of the tree, we can have the prices of both assets going up, both going down or up and down at the same time.

We design the scenario generation in C++ and Matlab. AMPL and the CPLEX 12.5 solver are used to model and solve the optimization problems. We run CPLEX' standard branch-and-bound algorithm with its default settings on a 64-bit desktop with Intel Core i7-4510U 2GHz CPU with 8GB of RAM. The computational performance is not the central goal of this paper, but it is worth pointing out the complexity of the model. For one single tree, we have $14,329$ variables ($2,047$ integers), $2,047$ nodes and $104,404$ constraints. The computational time for the generation of the scenarios and the solution of the optimization problem varies from 7 to 21 seconds. Without the combinatorial regulatory constraint (3.8), the time decreases to around $2-3$ seconds.

6.1 Number of trees - θ

We want to define the allocation that maximizes the wealth in the last period, while not violating any of the solvency, risk, and regulatory constraints. Our objective in this section is to determine the number of scenario trees needed to reach the stable state of the portfolio, i.e., stable allocations in fixed-income securities and stocks. Figure 2 shows how the average initial portfolio allocations vary with the number of trees.

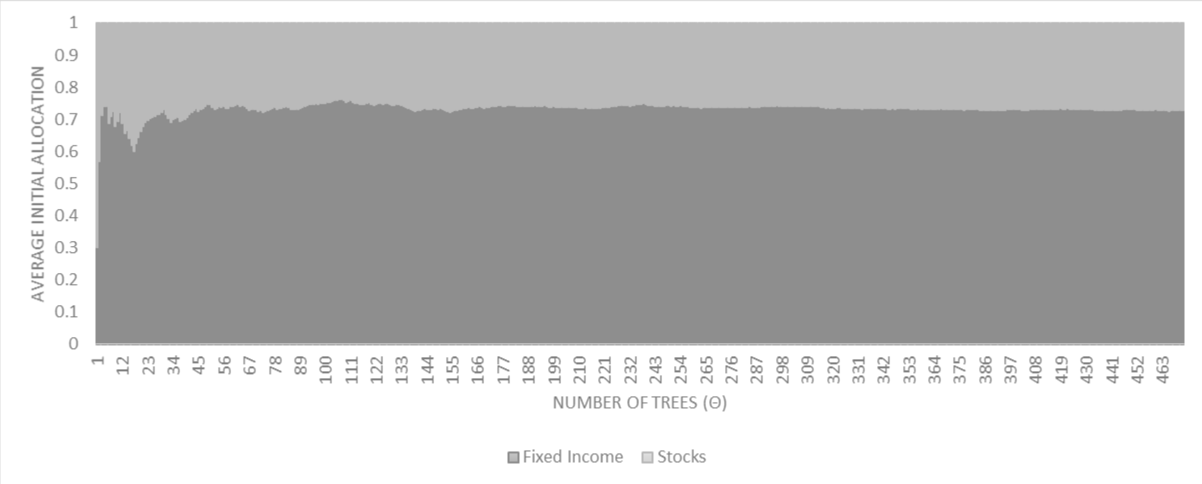


Figure 2 – Stability of average solution with respect to the number of trees - θ

The results displayed in Figure 2 indicate that the initial portfolio allocations become fairly stable with 200 generated trees. The subsequent tests presented in this paper are obtained by using 300 trees in Algorithm 1. The computational time to run the method proposed in this paper with 300 trees takes from 35 to 105 minutes depending on the parametrization. The solution time varies predominantly and not monotonically with the value of the initial funding ratio. If this latter is large (above 1.25), the solution time is minimal and about equal to 35 minutes. The solution process is the longest when the initial funding ratio is slightly smaller than one.

6.2 Insolvency Probability

A key indicator of the financial health of the pension fund is its insolvency probability. A tree is said to be infeasible if there is no feasible solution for the corresponding optimization problem (see Step 3 in Algorithm 1). The probability of insolvency is defined as the following ratio:

$$\mathbb{P}\left(\text{Insolvency} \mid \text{VaR}_{90\%}\right) = \frac{\text{Number of infeasible trees}}{\text{Number of trees}} \tag{6.1}$$

The model is constrained in such a way that the funding ratio must be above 1 with 90% probability in each period t and includes a 90%-VaR constraint at each period t. Furthermore, the intertemporal solvency regulation requires the funding ratio not be lower than 1 ($K = 1$) in more than two consecutive periods in any scenario. If in scenario s the funding ratio is below 1 in stages t and $t + 1$, it must be above 1 in stage $t + 2$ in scenario s. Another regulatory constraint is the 70% maximum amount of wealth to be allocated to stocks (see Chapter 2). The strong regulation and the high market volatility creates an environment in which the probability of a tree to be insolvent (infeasible) is not negligible.

Figure 3 displays how the insolvency probability varies in function of the pension fund's initial funding ratio β_0 for which we consider values ranging from 0.58 to 1.67. Note that the value of K, i.e., the legally required funding ratio, is kept equal 1 in each test and that we construct 300 trees for each considered value of the initial funding ratio. Each point in Figure 3 corresponds to the mean portfolio allocation over 300 trees and the associated insolvency probability.

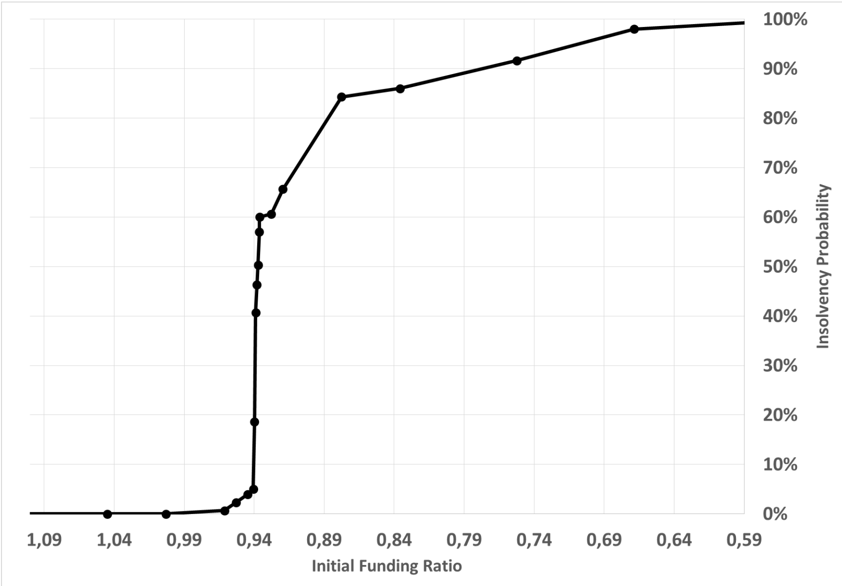


Figure 3 – Insolvency probability chart in terms of initial funding ratios.

Figure 3 highlights that the insolvency probability is stable and low until the pension fund's initial funding ratio reaches 0.95. The insolvency probability increases very fast when the funding ratio goes below 0.95 and makes it virtually impossible to keep the fund solvent when the value of the funding ratio falls below 0.93. For values of the funding ratios below 0.95, the pension fund manager will definitely need additional external contributions. The external contributions can come from the members in two ways: (i) decrease in the future benefits (liabilities) or (ii) increase in the contributions without raising the benefits. Next, we present in Table 3 how the portfolio allocations and insolvency probability vary with the initial funding ratio.

As in the resampled efficiency frontier method (MICHAUD; MICHAUD, 2008), the composition of the "final" fund is obtained by taking the average of the portfolio weights of all the 300 trees. Table 3 shows that the positions of the fund are not monotone with the initial value of the funding ratio. If the initial funding ratio is high (i.e., amount of assets is much higher than the present value of liabilities), the fund tends to allocate 70% and 30% in fixed income and stocks, respectively. The 70%-30% fixed income-stock allocation coincides to the one used in many Brazilian pension funds (Brazilian Association of Closed Supplementary Pension Funds, 2014). As the initial funding ratio becomes slightly lower than 1, the fund tends to concentrate more in fixed income to reduce its risks of not paying the liabilities - the fixed income allocation gets close to 80%. Once the initial funding ratio gets smaller than 0.95, the fixed income yield is not enough to cover the liabilities, a larger part of the portfolio is then dedicated to stocks. The 70% stock allocation is the maximum amount allowed by the Brazilian legislation and such an allocation is associated with a very high insolvency probability. This V-shaped portfolio policy is somewhat similar to the results reported by Berkelaar e Kouwenberg (2003) and Siegmann e Lucas (2005). With lower initial funding ratios (lower wealth levels) the allocation tends to be

Table 3 – Portfolio allocation and insolvency probability under different initial funding ratios.

Fund Ratio	Fixed-Income Bond %	Stocks%	Nº Insolv Scen	Insolv. Prob.%
1.672	70	30	0	0.0
1.463	67.3	32.7	0	0.0
1.254	70.9	29.1	0	0.0
1.045	72.6	27.4	2	0.7
1.003	72.2	27.8	7	2.3
0.961	81	19	12	4
0.953	83	17	15	5
0.945	82.1	17.9	56	18.7
0.941	82.1	17.9	122	40.7
0.940	73.1	26.9	139	46.3
0.939	71.6	28.4	151	50.3
0.938	70.4	29.6	171	57
0.937	63	37	180	60
0.936	64.4	35.6	182	60.7
0.928	30	70	253	84.3
0.919	30	70	258	86.0
0.752	30	70	298	99.3
0.669	NA	NA	300	100
0.585	NA	NA	300	100

concentrated in the risky asset.

We also tested additional assumptions for the pension fund. We assumed that the fund manager (or the regulator) becomes more risk averse. In such a case, we set $K = 1.1$ (instead of 1) in (3.4). As expected, the insolvency curve plotted in Figure 3 for $K = 1$ shifts to the left for $K = 1.1$. If we remove the intertemporal funding ratio regulatory constraint (3.8), the insolvency probability just slightly decreases.

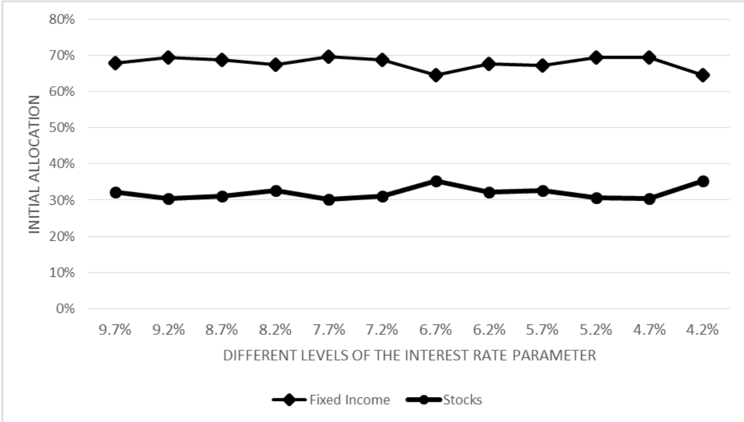
6.3 Allocation with a Decreasing Interest Rate

Despite the current sharp interest rate increase in 2014 and 2015 (more than 400 bps), a decreasing interest rate "looks like" a future tendency in Brazil. Dupačová e Polívka (2009) observed a similar trend in the Czech Republic. We shall now analyze the impact of a long-term decrease in the interest rate on the investment policy of Brazilian pension funds. In our analysis, we consider two scenarios that differ in the value of the initial funding ratio. In Figure 4 panel (a) (resp., panel (b)), we consider a pension fund with initial funding ratio of 1.25 (resp., 0.94). We chose an initial funding ratio of 1.25 that corresponds to a financially healthy pension fund, while 0.94 is the critical level of the initial funding ratio (see Figure 3) when the involvency probability of a fund a pension changes significantly. The resulting portfolio is, also, constructed by taking the average asset positions over the optimal positions of the 300 trees.

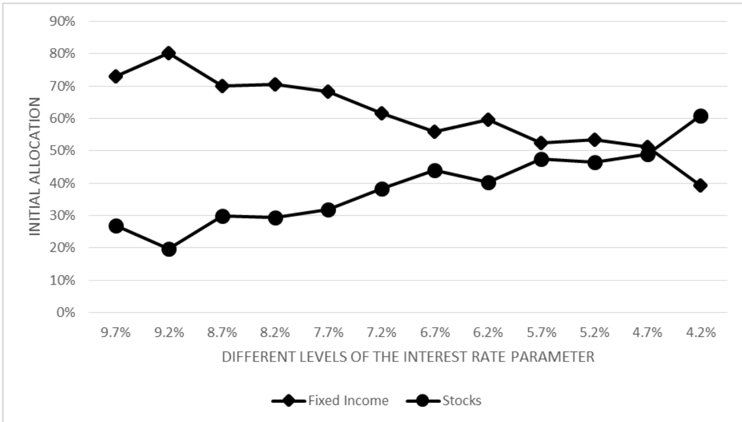
Based on Figure 4 panel (a), we observe that when the initial current asset value is much larger than the present value of future liabilities, the portfolio allocations are stable, i.e., around 70% and 30% allocated to fixed income and stocks, respectively. Despite the lower level of the

Figure 4 – Mean Allocations for different funding ratios and interest rates

(a) Mean Allocation - Initial Funding Ratio 1.25



(b) Mean Allocation - Initial Funding Ratio 0.94



interest rates, these funds do not increase their position in the risky assets due to their high volatility. Under this context, it is not needed to take more risks to be able to pay the liabilities. However, this conservative policy significantly decreases the future wealth of fund members. Now considering in Figure 4 panel (b) funds with low initial funding ratios, we can see that as the interest rate decreases the position in stocks becomes larger in order to possibly generate larger returns allowing for the payments of the liabilities. When the interest rate is close to 5%, the pension fund is required to take a riskier approach, investing more resources in stocks than in fixed income. This is a marked departure with the standard current allocations of pension funds and gives a clear indication the type of allocations most of Brazilian pension fund managers will have to adopt in the near future in case of lower interest rates.

In both cases displayed in Figure 4, pension fund managers will have to change their portfolio allocation if the decrease of interest rate becomes a reality. On one hand, if the fund is financially healthy (initial funding ratio 1.25), this change is necessary to avoid the progressive erosion of the initial wealth and capital. On the other hand, if the resources of the fund are tight (i.e., initial funding ratio 0.94), this adjustment is fundamental to maintain the ability to cover the liability payments. Besides the modification of the investment strategy, it is likely that members will be asked to increase their contributions and/or to accept benefits of lesser value in the future.

6.4 Robustness and Scalability

In this section, we focus on the out-of sample robustness and on the scalability with respect to the number of assets. First, we train the model on two different periods: Training A - from January 2005 to December 2013 and Training B - from January 2005 to December 2014. The application of the model built on the Training A (resp., Training B) data on the out-of-sample 2014 (resp. 2015) data permit to evaluate the out-of-sample performance of the model. As the return data are not at the present time available for the entire 2015 year, the 2015 out-of-sample analysis is based on the January 2015 - November 2015 period. To obtain the initial portfolio allocation, we use the method presented in Chapter 5. We test the model with three different initial funding ratios: 0.95, 1.00 and 1.05. These initial funding ratios become the benchmark for the out-of-sample tests presented in Table 4.

Table 4 – Out-of-sample tests with different funding ratios

		Funding Ratio (FR)		
Initial FR		0.950	1.000	1.050
Out-of-Sample FR	2014	0.958	1.006	1.066
	2015	0.982	1.025	1.085

We have chosen funding ratios that are close to the critical value of 1. We have also avoided ratios below 0.95 because of the V-shaped behavior towards the riskier asset documented in the literature (BERKELAAR; KOUWENBERG, 2003; SIEGMANN; LUCAS, 2005) and discussed in Chapter 6.2. A 15% federal income tax discount was used for the out-of-sample portfolio return. Based on Table 4 and considering the initial level of 1, the out-of-sample funding ratio increases to reach 1.006 in 2014 and 1.025 in 2015. Similar results were obtained with the initial funding ratios of 0.95 and 1.05. The initial funding ratio pension fund' maintenance (or increase) in the out-of-sample tests shows the robustness of the model.

We also note that this study has been conducted by accounting for the main asset classes (i.e., stocks and fixed income) in the ALM context. Those are definitely the main investment options considered by pension fund managers in Brazil (Brazilian Association of Closed Supplementary Pension Funds, 2014). Next, in order to the scalability of the model, we consider a larger number of assets. We design an experiment for two and eight assets with the following features: 500 runs (generating the scenarios and optimizing the tree), each tree with ten periods and funding ratio of 1. The objective here is to verify if the number of assets can be increased without making the computational performance prohibitive. The results showed that each run took on average 17.28 seconds for 2 assets and 22.30 seconds for 8 assets. The standard deviation was 1.59 and 4.2 seconds, respectively, for 2 and 8 assets. Thus, the implementation can easily be scaled up to a larger number of assets.

7 Conclusions

The objective of this study is to develop and apply a multistage stochastic programming ALM model for a Brazilian DB pension fund that takes into account the dynamic of the domestic financial market and its regulatory idiosyncrasies. Besides the multistage aspect, the complexity of the model is exacerbated by (i) the enforcement of a VaR metric modelled with probabilistic chance constraints, and (ii) the intertemporal solvency regulation modelled with combinatorial constraints. Different economic scenarios are simulated with the GBM and CIR models and used to construct multiple multistage scenario trees. The proposed multiple scenario tree approach is inspired from the resampling efficiency frontier method and is aimed at enabling the consideration of a representative set and vast number of contingencies without making the algorithmic procedure computationally prohibitive.

The empirical analysis shows the link between the initial funding ratio on one hand and the insolvency probability and the positions of the fund on the other hand. A simulation assuming a decrease in the interest case provides key insights about the likely changes in the investment strategies of the Brazilian pension funds.

A promising future research avenue is to switch the focus from defined benefit (DB) to defined contribution (DC) with minimum guarantees pension funds. For instance, the Brazilian public sector is currently experiencing this shift from DB to DC with some guarantees. Consiglio, Tummiello e Zenios (2015) not only discuss other countries in which this change is taking place but also present a model integrating option pricing and portfolio optimization to obtain asset allocations considering minimum guarantees.

Part II

In-sample performance comparison of scenario-generation methods applied to asset-liability management

Abstract

In this paper, we provide an empirical discussion of the differences among some scenario tree-generation approaches for stochastic programming. We consider the classical Monte Carlo sampling and Moment matching methods. Moreover, we test the Resampled average approximation, which is an adaptation of Monte Carlo sampling and Monte Carlo with naive allocation strategy as the benchmark. We test the empirical effects of each approach on the stability of the problem objective function and initial portfolio allocation, using a multistage stochastic chance-constrained asset-liability management (ALM) model as the application. The Moment matching and Resampled average approximation are more stable than the other two strategies.

keywords: Scenario Generation. Stochastic Programming. ALM. Sampling Method. Monte Carlo simulation. Moment Matching.

Note: this article has been published on Pesquisa Operacional v. 38, n. 1, p. 53-72, 2018.

Oliveira, A. D. D., Filomena, T. P., & Righi, M. B. (2018). Performance comparison of scenario-generation methods applied to a stochastic optimization asset-liability management model. Pesquisa Operacional, 38(1), 53-72.

1 Introduction

Stochastic programming (SP) is a branch of optimization, in which optimal decisions are made under uncertainty (BIRGE; LOUVEAUX, 2011). In general, real world problems are characterized by uncertain events modeled with random variables. We associate a collection of random events with a probability space containing a bundle of all possible events with their probabilities. From these concepts, we formalize a generic stochastic optimization problem, as denoted by Eq. (1.1)

$$G(x) = \max_{x \in X} \int_F f(x, \xi) dF(\xi), \tag{1.1}$$

Where f is the objective function that is defined in terms of uncertainty, x is the decision variable defined over the feasible set $X \subset \mathbb{R}^I$, and ξ is a random variable defined by the continuous cumulative distribution function $F : \mathbb{R}^I \to [0, 1]$. Furthermore, dF represents the probability measure of the probability space F for the underlying multivariate stochastic process.

As discussed by Pflug (2001), continuous optimization problems become easier to solve if we reduce them to discrete-state multi-period optimization problems. This logical structure may be seen as a tree model with a non-anticipative decision process, in which dependence relies uniquely on this history and on the probabilistic specification (DUPAČOVÁ et al., 2000; ESCUDERO; KAMESAM, 1995). Thus, the decision functions reduce to large decision vectors and, as Lohndorf (2016) points out, it may be calculated numerically by either drawing a sample from F or by approximating F with a discrete distribution $\hat{F}$. Using $\hat{F}$ instead of F, we have the following maximization problem:

$$\max_{x \in X} \sum_{\xi \in \hat{F}} \hat{\phi}(\xi) f(x, \hat{\xi}) \tag{1.2}$$

Where $\hat{\phi}(\xi)$ is the probability of the mass point in $\hat{F}$. If we suppose a uniform distribution for $\hat{\phi}$, the approach in Eq. (1.2) can be seen as sample average approximation (SAA). Stochastic approximation (SA) is another approach to obtain numerical convergence for stochastic optimization problems, which presents a random direction whose origin is usually an objective function's gradient with a step-size for each iteration (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2009).

As the scenario generation methodology becomes a key part of the stochastic optimization process, the main goal of this study is to compare the performance of different scenario sampling methods, in order to highlight which of them is more appropriated for designing a representative discrete-space model for asset-liability management (ALM) problems regarding the in-sample performance. Many efforts have been made in the scenario generation direction, for instance, by having matched state-space distribution moments (DUPAČOVÁ et al., 2000; HØYLAND; WALLACE, 2001; HØYLAND; KAUT; WALLACE, 2003), minimizing Wassertein probability metrics (ROMISCH, 2003; HEITSCH; ROMISCH, 2005; HOCHREITER; PFLUG, 2007), Latin hypercube sampling (MCKAY; BECKMAN; CONOVER, 1979), Voronoi cell sampling (LOHNDORF, 2016), and Resampled average approximation (OLIVEIRA et al., 2017), among others.

This study compares the empirical results of distinct approaches to generating scenarios for ALM: Random sampling and Moment matching. We also test two Monte Carlo sampling variations: Resampled average approximation and Monte Carlo, using a naive allocation strategy as a benchmark. These sampling techniques were chosen among other well-known options, as cited above. We apply sampling methods from distinct perspectives. The Monte Carlo sampling-based algorithms are based on the uniform version of strong law of large numbers. It ensures that the optimal objective value of the SAA problem defined by Eq. (1.2) converges to the true value of the problem, according to the increase of the number of scenarios (GLASSERMAN, 2003). The Moment matching depends mainly on the prior knowledge of the distribution function of the marginals. Several changes in these paradigms can be found in the literature, for instance, Mak, Morton e Wood (1999) and Homem-de-Mello e Bayraksan (2014b) propose variance reduction techniques in Monte Carlo sampling-based approximations to improve the scenario representation. Moreover, Kaut e Wallace (2011) introduce copulas in the definition of moments, but they represent perspectives with essentially distinguished methodologies.

Our intent is not to exhaustively test the sampling methods for ALM, but to outline how, empirically, the method may have an impact and produce different outputs. Based mostly on the resulting values of the objective function, as suggested by Kaut e Wallace (2007), we can conclude that the classical Monte Carlo sampling and Monte Carlo with naive allocation strategy are dominated by the Moment matching and the Resampled average approximation.

The paper is organized as follows. First, we introduce this study's multistage stochastic chance-constrained ALM model in Section 2. Then, in Section 3, the scenario generation methods are described: Monte Carlo sampling (Section 3.1), the Moment matching sampling (Section 3.2), the Resampled average approximation (Section 3.3) and the benchmark Monte Carlo with naive allocation (Section 3.4). Section 4 describes the generation of the sample paths with other explanations on the data and the experiment. A comparison of the results considering the different approaches appears in Section 5. Concluding remarks are in the final section.

2 Stochastic asset-liability management model

ALM is focused on modeling suitable sample paths for the assets and liabilities of a pension fund, bank, insurance company, or any other institution that dynamically manages and matches risks on both sides of a balance sheet, see e.g.,(MULVEY; W.T.ZIEMBA, 1998; ZENIOS; ZIEMBA, 2006; HOCHREITER; PFLUG, 2007; HANEVELD; STREUTKER; VAN DER VLERK, 2010; MITRA; SCHWAIGER, 2011; RIGHETTO; MORABITO; ALEM, 2016). In other words, the objective of ALM is to guarantee that liabilities are paid over a multi-period horizon by efficiently managing investment resources , e.g., (ZIEMBA, 2003; MATOS; PADILHA; BENEGAS, 2014). Our study's model is designed for the pension fund environment. Thus, the fund's assets must be strategically managed so that the total value of all assets is greater than the fund's liabilities with high probability, while respecting all constraints, for instance, the fund's solvency (BOGENTOFT; ROMEIJN; URYASEV, 2001).

The stochastic programming community has discussed the ALM for a while. Early models were proposed by Bradley e Crane (1972), with commercial versions emerging thereafter for banks and insurance companies, respectively, with Cariño et al. (1994), Kusy e Ziemba (1986) and Kosmidou e Zopounidis (2002). Others expanded these proposals, for instance, Boender (1997) presented a large-scale model in which the asset allocation resulted from heuristic techniques. More recently, multistage stochastic programming approaches have become a trend, with examples in Consigli e Dempster (1998b), Kouwenberg (2001), Moraes e Faria (2016). Kouwenberg (2001) and Høyland e Wallace (2001) discussed the challenge of generating representative scenarios; a key aspect in multistage stochastic programming. Stochastic programming ALM models related to non-neutral risk measures and based on the CVaR measure, or including jumps in asset prices, have already been presented in Rockafellar e Uryasev (2000), Bogentoft, Romeijn e Uryasev (2001), Kilianová e Pflug (2009), Ferstl e Weissensteiner (2011), Josa-Fombellida e Rincón-Zapatero (2012).

Our stochastic ALM model is based on Oliveira et al. (2017). Suppose that there is a set of securities denoted by $i = 1, \dots, N$. The ALM manager has to choose from these investment opportunities, allocating the available wealth in order to afford the liabilities denoted by l_t. These decisions occur repeatedly through different time periods, $t = 1, \dots, T$. The utility function maximizes the final expected wealth. Thus, the model takes the form of a stochastic and inter-temporal dynamic allocation problem, due to the randomness of the asset prices and the time-dependent nature of the investment and rebalancing decisions.

The investment strategy is designed with three sets of variables. The here-and-now decisions that are taken before the information is revealed, denoted by X_{its}. This is the position (shares) of asset i to hold in time period t and scenario s. The corrective actions, which are made after information revealing, are described by both B_{its} and V_{its} wait-and-see variables. They are, respectively, the position (shares) of i bought and sold during time t and scenario s.

The input data of the optimization model is determined by the continuous random variable ξ. This stochastic variable may be discretized by ξ_{it} that defines the price of asset i at time t. Additionally, it can take a finite number, $s \in 1, \ldots, S$, of realizations denoted by P_{its}. As the asset returns are simulated by stochastic process, detailed in Section 4.1, the price realization P_{its} is defined as a conditional sampling of the random variable ξ_{it}. Therefore, the realizations for P_{its} depend on P_{it-1s} (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2009).

The scenario tree is comprised of scenarios ranging from the initial to the final period. A scenario k consists of a set of sequential realizations, $\{P_{i1s}, P_{i2s}, \ldots, P_{iTs}\}\ \forall s \in S$, of the random variable ξ_{it}. They must be adapted to the non-anticipative constraints that drive the information unfolding process. For instance, the scenario k is a random vector, generated by a ξ_{it} realization, which is described as $\lambda_k(\xi_{it}) = (P_{i1k}, \ldots, P_{itk}),\ ,\ \forall i \in N$. Table 5 presents a summary of the model's notations.

Table 5 – Notation Summary

Sets - Indices	
t	time index (stage) $t = 0, 1, \ldots, T$
i	index of asset classes $i = 1, \ldots, N$
s	index of scenarios $s = 1, \ldots, S$
Decision variables	
X_{its}	Number of shares of assets i to hold during time t and scenario s
B_{its}	Number of shares of assets i to buy during time t and scenario s
X_{i0}	Number of shares of assets i to hold initially ($t = 0$)
V_{its}	Number of shares of assets i to sell during time t and scenario s
Random variables	
ξ_{it}	Random price of asset i during time t
Deterministic parameters	
Q	Initial wealth
α_t	Reliability level during time t
K	Legally required funding ratio
L_t	Present value of future liability $t = t + 1, \ldots, T$
l_t	Liability to be paid during period t
F_t	Present value of future external contributions, $t = t + 1, \ldots, T$
f_t	External contributions in each period t
M	Maximum amount of underfunding allowed
ρ	Discounting factor
π	Maximum weight of an asset in the portfolio
ϕ_{ts}	Probability of scenario s at time t
P_{its}	Price of asset i at time t and in scenario s
P_{i0}	Known initial ($t = 0$) price of asset i

The model is a multistage stochastic programming model with chance constraints, described below:

$$\max \qquad \sum_{s=1}^{S}\sum_{i=1}^{N}\phi_{Ts}P_{iTs}X_{iTs} \tag{2.1}$$

s.t. :

$$Q=\sum_{i=1}^{N}P_{i0}X_{i0} \tag{2.2}$$

$$X_{its}=X_{i(t-1)s}+B_{its}-V_{its}, \qquad \forall t\in T,\forall i\in N,\forall s\in S \tag{2.3}$$

$$\mathbb{P}\left(\sum_{i=1}^{N}\xi_{it}X_{it}\geq K(L_t-F_t)\right)\geq\alpha_t, \qquad \forall t\in T, \tag{2.4}$$

$$\sum_{i=1}^{N}P_{its}V_{its}-\sum_{i=1}^{N}P_{its}B_{its}+f_t=l_t, \qquad \forall t\in T,\forall s\in S \tag{2.5}$$

$$X_{its}P_{its}\leq\pi\sum_{i=1}^{N}X_{its}P_{its}, \qquad \forall t\in T,\forall i\in N,\forall s\in S \tag{2.6}$$

$$X_{itk}=X_{itk'}\text{ if }\lambda_k(\xi_{it})=\lambda_{k'}(\xi_{it}), \qquad \forall t\in T,\forall i\in N,k,k'\in S \tag{2.7}$$

$$X_{its},B_{its},V_{its}\geq 0, \qquad \forall t\in T,\forall i\in N,\forall s\in S \tag{2.8}$$

The model maximizes the expected terminal value of the fund (2.1). The objective function reflects the fund manager's goal to reach the largest possible gains in the last period, while respecting the risk and liability payment constraints. At time $t=0$, the constraint (2.2) is deterministic, so that the initial wealth Q is distributed among the asset classes. As the price P_{i0} of each asset is known and deterministic, we can rewrite X_{i0s} as $X_{i0},\forall s$. The number B_{i0} of shares bought is equal to the number X_{i0} ($\forall i$) of shares kept at the end of the initial period. The linear equalities (2.3) are the share balance constraints. These specify that the number of shares X_{its} of asset i during time t and scenario s is equal to the number of shares $X_{i(t-1)s}$ maintained in the previous period, augmented by the number of purchased shares B_{its} minus those sold V_{its} at time t.

The chance constraints (2.4) enforce the funding ratio requirements, which represent the long-term relationship between assets and liabilities. The actual funding ratio β_{ts} of a fund during time t and scenario s is computed as:

$$\beta_{ts}=\frac{F_t+\sum\limits_{i=1}^{N}P_{its}X_{its}}{L_t}, \qquad t=1,\ldots,T,s=1,\ldots,S\ , \tag{2.9}$$

where $\sum_{i=1}^{N}P_{its}X_{its}$ is the current asset value of the pension fund in scenario s. L_t and F_t are, respectively, the present value of the future liabilities and contributions discounted by ρ:

$$L_t=\sum_{j=t}^{T}\frac{l_j}{(1+\rho)^{j-t}}, \qquad F_t=\sum_{j=t}^{T}\frac{f_j}{(1+\rho)^{j-t}}.$$

A value of β less than 1 signals that the value of the assets may become insufficient to cover future liabilities, so the fund might run into solvency issues in the near future. The two parameters K

and α_t define the asset-liability management policy's risk-aversion. The constraints (2.4) can be viewed as some sort of VaR constraints, ensuring that the value of the fund is at least equal to $K(L_t - F_t)$ during each period t with a probability at least equal to α_t. These maintain the fund's long-term solvency level.

The model cash flow balance is denoted by the equalities indicated by Eq. (2.5). The financial inputs are represented by the sales of assets and external contributions, while the financial outputs are payments of liabilities and purchases of assets. The constraint (2.6) stipulates that no asset can have a weight greater than an upper bound π. Multistage models have the so-called non-anticipativity condition. This determines that decisions are impacted only by the past, not by the future, i.e. two scenarios with a common history until the tth stage should result in the same decisions until this stage (CARØE; SCHULTZ, 1999). Constraint (2.7) guarantees that scenarios with the same history present identical asset allocation. A maximum admissible underfunding value is defined through the parameter M. Constraint (2.8) defines the non-negativity restriction on the decision variables.

3 Scenario Generation Methods in ALM

In this section, we explain the simulation algorithms for generating scenario trees for the multistage decision problems used in our study: Monte Carlo sampling, Moment matching and Resampled average approximation. They gather the stochastic processes into a multistage scenario tree to model uncertainty. Even though the Monte Carlo with naive allocation strategy is not a scenario generation method, its explanation is still kept in this section.

3.1 ALM with classical Monte Carlo sampling

The traditional method to generate a scenario tree for ALM is through Monte Carlo sampling, with uniformly distributed pseudo-random numbers transformed appropriately into a target distribution (UBEDA; ALLAN, 1994; DEMPSTER; MEDOVA; YONG, 2011). This approach is an efficient way to represent multi-dimensional distributions (ZENIOS; ZIEMBA, 2006). In this study, we generate $W_1, \ldots, W_I$ random vectors from the standard normal distribution. As Homem-de-Mello e Bayraksan (2014a) note, in this case, the vectors $W_1, \ldots, W_I$ are mutually independent; a detail that characterizes this sampling method.

As Monte Carlo is based on the volume of a set distribution for the definition of probability measure, an obvious way to deal with this problem is to increase the number of nodes in the randomly sampled event tree. However, the stochastic program might become computationally intractable due to the tree's exponential growth rate. This hypothesis is supported, not only by the law of large numbers that guarantees the convergence to a correct value as the number of draws increase, but also by the central limit theorem that offers information about the error magnitude after a finite number of simulations. Hence, the convergence and error estimation of the outputs is directly linked with the number of draws. Indeed, one of the features of Monte Carlo is the form of the standard error, which, for a generic function f, can be defined by $\sigma_f/\sqrt{n}$, with n being the number of draws, and σ_f the sample standard deviation. The Monte Carlo sampling is also a good choice for integrals in high dimensions, because its convergence rate holds ($O(n^{-1/2})$) for any dimension. Glasserman (2003), Rubinstein e Kroese (2016) provide more specific features of Monte Carlo sampling.

According to Kouwenberg (2001), despite the intuitiveness of this approach, the mean and covariance matrix may not be correctly specified in most nodes of the tree, given that the states are randomly sampled. Thus, the optimizer might choose an investment strategy from erratic or misspecified parameters.

3.2 ALM with Moment matching sampling

The Moment matching approach aims to mitigate the impact of the inconsistencies in the specification, given that it is not possible to reach a full match with misspecified parameters.

It also allows the decision maker to determine the output features based on the statistical distribution properties considered relevant.

Initially, as argued by Smith (1993), and proposed by Høyland e Wallace (2001), the Moment matching sampling matches the statistical properties to minimize the error between the sampled data of the tree and the first two moments of a theoretical distribution. Therefore, the scenario tree keeps some statistical properties reflecting the same characteristics as the theoretical distribution. Since there are a finite number of moments m on a continuous distribution, in our case two, Dupačová et al. (2000) argue that there always exists a discrete probability with these same moments and its support has at most $m + 2$ points, see Prékopa (1995, chapter 5 for the proof).

Furthermore, the literature presents examples in which higher order moments are matched (HØYLAND; KAUT; WALLACE, 2003). However, it may become very difficult to obtain a solution for non-linear constraints such as skewness and kurtosis. Kouwenberg (2001) and Zenios e Ziemba (2006) also adjust only the first two moments in their analysis. In this study, we construct an event tree that fits the mean and the covariance matrix of the underlying distribution.

In order to fit the first two moments, we use Cholesky decomposition in the same direction as in Høyland, Kaut e Wallace (2003). The first step is to generate random vectors from the standard normal distribution, as described in Section 3.1. Then, the I random vectors are transformed to show a given covariance matrix by multiplying the vectors by a lower triangular matrix L of the covariance matrix Σ,

$$W'_j = LW_j, \Sigma = LL', j = 1, \dots, I, \tag{3.1}$$

where we can obtain L by applying Cholesky decomposition. In other words, as Kouwenberg (2001) states, we specify that the average of the disturbances should be zero, and they should have a covariance matrix equal to Σ. Therefore, we denote this matching in Eqs. (3.2) and (3.3):

$$\frac{1}{S}\sum_{s=1}^{S} W_{js} = 0 \quad \forall j \in 1, ..., I, \tag{3.2}$$

$$\frac{1}{S-1}\sum_{s=1}^{S} W_{js}W_{is} = \Sigma_{ij} \quad \forall j, i \in 1, ..., I. \tag{3.3}$$

This methodology allows for the generation of different sample paths, which matches the first two moments since the disturbance compose the asset price modeling (see Equations 4.1 and 4.2). As in Kouwenberg (2001), Dempster, Medova e Yong (2011) and Lohndorf (2016), it is possible to argue that this sampling approach outperforms other methods, such as Monte Carlo sampling, Wasserstein distance sampling, or even Latin hypercube sampling. This methodology also enables us to capture unlikely scenarios if we consider higher order moments (DUPAČOVÁ et al., 2000; HØYLAND; WALLACE, 2001).

Unlike Monte Carlo sampling, the Moment matching does not necessarily converge when the number of scenarios is increased. It depends mainly on how much statistical properties are being matched and the distribution features, for example, the smoothness (KAUT; WALLACE, 2007). Overspecification and underspecification are also an issue when dealing with the first two moments (HØYLAND; WALLACE, 2001).

The works of Høyland e Wallace (2001) and Høyland, Kaut e Wallace (2003) have been also adapted during the last few years. For instance, Gülpınar, Rustem e Settergren (2004a) discuss adaptations of Moment matching using sequential and overall optimization. Beraldi, Simone e Violi (2010) propose a variant of Moment matching using parallel processing techniques, in which the scenario probabilities also become decision variables. Without being extensive, Mehrotra e Papp (2013) make the case for a more flexible optimization method in which some lower-order moments can be matched exactly, while higher-order moments only approximately.

3.3 ALM with the Resampled average approximation

The Resampled average approximation is a simulation of many scenario trees, with the average of the initial portfolio taken as the solution (OLIVEIRA et al., 2017). When we consider several trees, we account for a wide spectrum of variability inherent to the parameters in the optimization problem. This technique has some similarities to the Resampled efficient frontier method proposed by Michaud e Michaud (2008) to construct portfolios of risky securities. With a mathematical definition similar to that for the Resampled efficient frontier method for mean-variance optimal portfolios, the ALM Resampled average approximation optimality is the expected value in the solution space of the Monte Carlo ALM financial plan, as presented in Section 3.1. Thus, it may be seen as a reshuffle of the classical Monte Carlo . This methodology could also be adapted to the Moment matching, but the volatility of its results has already been regulated through the adjustment of the second moment. In other words, it is a needless further procedure to mitigate the risk, which is supposedly one of the main advantages of the Moment matching when compared to the classical Monte Carlo sampling.

This sampling technique is applied and described in detail by Oliveira et al. (2017). Additionally, this approach can be viewed as a particular case of one of the algorithms discussed by Homem-de-Mello e Bayraksan (2014a), in which the stopping criteria is predefined by the number of runs. The ALM with the Resampled average approximation has four steps. First, we define the number of trees to solve for each parametrization, see e.g. (MICHAUD; MICHAUD, 2008). The value of these outcomes must be sufficiently large to provide stable portfolio allocations, while being small enough to ensure that the approach does not become computationally prohibitive. After defining the number of instances, the second step is to generate the scenarios for each tree according to the ALM model. In Step 3, we solve each corresponding optimization problem to optimality. In Step 4, we evaluate the results based on the optimal solutions of the trees. The average of the initial allocation portfolio of all simulations is then the model solution. It is distinguished from SAA because of the number of scenario trees that are solved. In the SAA, a unique scenario tree is generated and solved. In this case, the expectation is applied on the decisions from the next stage. Unusually, the Resampled average approximation gives origin and solves an arbitrary number of unrelated scenario trees with the same topology, but for distinct scenarios. After that, the expectation is taken from these independent instances.

3.4 ALM with the Monte Carlo with naive allocation strategy

The Monte Carlo with naive allocation strategy is generally a naive policy that is very often reported in the portfolio allocation literature, in contrast to mean-variance portfolio optimization (BENARTZI; THALER, 2001; DEMIGUEL; GARLAPPI; UPPAL, 2009). Although this methodology is heuristic and does not reflect rational behavior, this analysis is justified due to the estimation error inherent in the sampling process. The Monte Carlo with naive allocation assesses whether an optimized portfolio performs better, despite the misspecification data when compared to irrational behavior. DeMiguel, Garlappi e Uppal (2009) assert that, in the sample-based mean-variance strategy, a window with around 3,000 months is necessary for a portfolio with 25 assets to outperform the 1/N benchmark.

In ALM, banks have already used this rule to make their portfolios in the 1960s (COHEN; HAMMER, 1967). Furthermore, pension funds also adopted this policy in the 1980s (HARRISON; SHARPE, 1983), and the USA Pension Benefit Guaranty Corporation had followed this rule. Although the Monte Carlo with naive allocation, or the 1/N portfolio, might provide good returns, they may not be able to meet the legal requirements and cash balance constraints. Additionally, Zenios e Ziemba (2007) show that, as the naive allocation is unable to incorporate new information, stochastic programming can outperform it for ALM. Fleten, Høyland e Wallace (2002) also compared the Fixed-Mix strategy with the multistage stochastic linear programming, verifying the superiority of its model over the Fixed-Mix approach. We consider the $1/N$ naive policy version for our tests as the benchmark for other sampling methods. We define our naive allocation as an equally distributed portfolio, described in Eq. (3.4):

$$X_{i0} = \frac{Q}{N} \cdot \frac{1}{P_{i0}} \quad i = 1, \ldots, N. \tag{3.4}$$

Therefore, we use this popular investment policy with the classical Monte Carlo sampling, defined in Section 3.1, in order to settle a comparative level to other methods simulated. Even with the investment policy defined, we have to guarantee that constraints (2.2)−(2.8) are respected in this strategy. Therefore, in the simulations executed at Section 4, we set the funding ratio to one. Thus, in the beginning of the simulation's time horizon, the pension fund has the all necessary wealth to afford the liabilities until the final time period.

4 Simulation

First we introduce, in Section 4.1, arbitrage-free continuous-time stochastic processes used to simulate prices at a discrete set of dates (GLASSERMAN, 2003). We adopt the Geometric brownian motion model (GBM) and the Cox-Ingersoll-Ross model (CIR), which is a single-factor term structure model. These stochastic processes are adapted to the scenario tree topology. Then, in Section 4.2, the scenario tree structure is described.

4.1 Generating Sample Paths

We generate the scenario trees by asset prices realization sampling. The asset prices follow correlated stochastic differential equations (SDEs). We use the GBM for stock prices (NEFTCI, 1996; DUFFIE, 2001):

$$d\xi_{1t} = \mu\xi_{1t}dt + \sigma\xi_{1t}dW_{1t}. \tag{4.1}$$

Thus, there is a GBM for each stock. For the price of a fixed income asset, we use the Cox-Ingersoll-Ross term structure model (COX; INGERSOLL; ROSS, 1985):

$$d\xi_{2t} = \alpha(\mu - \xi_{2t})dt + \sqrt{\xi_{2t}}\sigma dW_{2t} \ , \tag{4.2}$$

where ξ_{2t} is the interest rate and (α, μ, σ) are model parameters. The drift function $\alpha(\mu - \xi_{2t})$ is linear and has a mean reverting property, i.e. the interest rate ξ_{2t} moves in the direction of its mean μ at speed α. The diffusion function $\xi_{2t}\sigma^2$ is proportional to the interest rate ξ_{2t} and ensures that the interest rate is always positive.

A total of four succeeding nodes for each scenario s at time t are available to describe the conditional distribution of these random variables in a particular node at time $t-1$. We define the disturbance W_{js} as the realization in node s for the jth element of the vector W. The model maximizes the expected wealth of an ALM problem applied to a pension fund, as defined generically in Eq. (1.1) respecting a set of restrictions.

We use only broad stock or bond indexes that cause arbitrage in cases of very poor approximation of the assets' underlying distribution on the stochastic programming tree (KOUWENBERG; ZENIOS, 2006). We use just two stock broad indexes and a floating rate fixed income instrument with long-only positions, without any complex asset. We also present a large number of scenarios for each tree. Thus, in our study settings, the discussion of arbitrage is not critical.

The parameters used in GBM, Eq. (4.1) are estimated through historical time series data. Those used in CIR, Eq. (4.2), are estimated using maximum likelihood. These sampled paths are conditioned to non-anticipative constraints defined in Eq. (2.7), following the multistage stochastic framework integrated with the sampling algorithms. We simulate the classical Monte Carlo sampling, the Moment matching method, the Resampled average approximation and Monte Carlo with naive allocation, as explained in Section 3.

4.2 Scenario Tree Simulation

The simulation occurs prior to the optimization process; this methodology is known as the non-recursive method, see (PFLUG, 2012). In order to simulate the trees and to study the outputs of the sampling methods, we define tree classes of scenario trees that are denoted by small, medium and large according to their number of periods. They have 4, 6 and 8 periods respectively. In each class, we determine a different topology, which produces a distinct number of nodes, scenarios, variables and constraints (Table 6).

Table 6 – Classes and Topologies of scenario trees

Name	Topology	Nodes	Scenarios	Variables	Constraints
Small	1-27-9-9	2,458	2,187	24,843	17,203
Medium	1-81-3-3-3-3	9,802	6,561	101,253	68,613
Large_A	1-16-3-3-3-3-3-3	17,489	11,664	180,707	122,424
Large_B	1-8-6-3-3-3-3-3	17,481	11,664	180,603	122,359

In the large class, we test the behavior of scenario generation and optimization for two different topologies that present the same number of scenarios $(11,664)$. The number of variables and constraints shown in Table 6 is defined by the deterministic linear equivalent problem, denoted through the Eqs. (2.1) – (2.8). We compose each scenario with a discrete sequence of conditional distributions of stocks and fixed income assets, calibrated with historical data from January 2012 to November 2016.

We use data from the Brazilian capital market. Equation (4.1) defines the stock price models, which are calibrated with annualized daily return prices from the Bovespa index (the most liquid stocks in the country) and the Brazilian Small Cap BM&F Bovespa index (stocks with small capitalization). We also have a fixed income asset modeled with Eq. (4.2) and calibrated with data from the 1-month Brazilian LTN (similar to a T-Bill in the USA) as a proxy for the short-term interest rate. In Table 7, we show the parameters used in the model.

Table 7 – CIR and GBM parameters.

Asset	Return Annualized (μ)	Std. Annualized (σ)	Mean Revert. (α)
Fixed Income	0.11297	0.04358	0.14599
Bovespa index	0.13503	0.23486	-
Small Cap index	0.07426	0.17716	-

The covariance matrix Σ takes the variance and covariance among Fixed Income, Bovespa index, and Small Cap index as inputs for Eq. (3.1) in the Moment matching sampling. In Table 8, we present the covariance matrix Σ observed from the data.

Table 8 – Covariance Matrix for the experiments.

Asset	Fixed Income	Bovespa index	Small Cap index
Fixed Income	0.005339086	-0.001021373	-0.000973024
Bovespa index	-0.001021373	0.055221863	0.035719690
Small Cap index	-0.000973024	0.035719690	0.031501802

In the Monte Carlo with naive allocation approach, we take the initial wealth and allocate it uniformly between the financial assets, as defined in Eq. (2.2). This position is fixed until the last period. In each period, we calculate the financial value of the portfolio and discount its liabilities. Monte Carlo with naive portfolio is not rebalanced in any time period. Thus, the final expected value of wealth is the sum of the net financial value of the portfolio multiplied by its probability of occurrence, similar to Eq. (2.1). Even though the Monte Carlo with naive allocation is not a sampling method, we take it into account as an element of comparison for our analysis.

We performed 20 simulations for each sampling method, then estimated the average and standard deviation from both the objective function and the initial allocation for each methodology applied for each class. Unlike the Random sample and Moment matching, where each turn is comprised of only one scenario tree, we generated and optimized 300 unrelated scenario trees to compose each turn of Resample average approximation. This procedure was applied on all topologies shown in Table 6. This analysis consists of 12,280 runs, and evaluated 141,426,000 different scenarios. We designed the scenario generation in C++ and Matlab using AMPL and the CPLEX 12.6.1 solver to model and solve the optimization problems. We run it in a 64-bit desktop with Intel Core i7-4510U 2GHz CPU and with 8GB of RAM. In Table 9, we report each class and method's computational times (mean and standard deviation) in seconds to solve a single tree.

Table 9 – Times elapsed for scenario generation and optimization

Name	Tree Generation Time				Model Optimization Time				Total Time			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.
Small												
Moment matching	0.4	0.06	0.24	0.48	4	0.44	4	5	4.551	0.46	4.24	5.48
Monte Carlo	0.28	0.04	0.24	0.43	4.7	0.65	4	6	4.98	0.65	4.26	6.29
Resampling	0.23	0.02	0.19	0.29	4.36	0.37	3.56	5	4.60	0.38	3.75	5.25
Medium												
Moment matching	1.56	0.12	1.44	1.85	4.65	0.58	4	6	6.21	0.60	5.46	7.7
Monte Carlo	1.92	0.41	1.44	2.62	18.10	11.33	10	59	20.02	11.30	11.48	60.45
Resampling	1.81	0.18	1.59	2.12	16.586	2.26	13.66	21.02	18.39	2.39	15.39	23.14
Large												
Moment matching_A	5.18	1.21	4.5	8.92	14.15	1.66	12	19	19.33	2.63	16.61	27.92
Moment matching_B	6.04	1.26	4.61	8.51	16.6	3.20	12	23	22.64	3.89	16.61	30.38
Monte Carlo_A	5.84	1.39	4.51	8.55	25.1	9.48	11	49	30.94	9.99	15.51	54.14
Monte Carlo_B	6.11	1.03	4.82	8.67	19.70	5.13	14	38	25.81	5.50	20.13	45.56
Resampling_A	5.71	0.64	4.91	6.94	24.74	1.99	20.76	28.14	30.45	2.52	25.67	35.08
Resampling_B	6.35	0.68	5.22	7.66	19.02	3.81	4.10	23.32	25.37	3.91	11.28	30.72

Based on Table 9, we notice that, on average, the computational time to solve a single tree is between 4 and 61 seconds. For the Resampled average approximation, the computational experiment can be taken from 7.5 to 51 hours to be completed: 6,000 trees must be computed for each instance. For the other three methods, the average time has to be multiplied by 20, which is the number of simulations to finish each experiment. The Monte Carlo with naive allocation strategy is not considered on Table 9 because it does not demand much computational effort. Next, we present and describe the results from the simulations.

5 Results

Our results focus on the stochastic programming in-sample stability (KAUT; WALLACE, 2007). The in-sample stability is evaluated by generating different scenarios for each tree and comparing how stable the problem's objective function and solution are. Thus, we analyze the stability of the sampling methods from two different perspectives: objective function and initial portfolio allocation. Dempster, Medova e Yong (2011) argue that the initial portfolio allocation criterion is not often used in the literature, because of the potential for a flat plateau objective function. However, as our scenarios might have sampling error, we conducted the analyses from both perspectives. We focus on the expectation and volatility of both the objective function and initial portfolio allocation. The results of the objective function and initial portfolio allocation are presented in Tables 10 and 11, respectively.

Table 10 – Objective function statistical outputs

Name	Mean	Std. Dev.	Min.	Max.
Small				
Resampling	714,114.47	2,805.67	708,800.98	720,443.70
Moment matching	629,656	7,341.04	619,961	642,192
Naive allocation	814,619.86	30,153.65	762,498.94	864,254.46
Monte Carlo	712,815.40	22,927.45	672,949	756,231
Medium				
Resampling	1,065,780.19	3,850.40	1,056,652.26	1,073,008.54
Moment matching	711,529.65	8,698.65	695,761	727,839
Naive allocation	1,136,827.87	26,529.07	1,072,786.30	1,184,755.13
Monte Carlo	1,074,340.85	20,485.86	1,028,342	1,106,371
Large				
Resampling_A	1,584,145.66	18,498.18	1,554,806.82	1,633,666.16
Resampling_B	1,549,304.86	13,994.94	1,526,511.44	1,572,683.32
Moment matching_A	828,987	21,964.05	795,107	873,140
Moment matching_B	837,172.05	25,982.62	798,804	881,010
Naive allocation_A	1,564,422.42	68,972.70	1,469,548.25	1,736,670.72
Naive allocation_B	1,625,050.78	117,436.33	1,487,900.62	1,902,890.73
Monte Carlo_A	1,586,488.10	137,887.48	1,391,047	1,923,511
Monte Carlo_B	1,531,394.65	128,454.55	1,267,510	1,772,048

In Table 10, we observe that objective function's standard deviation, of the Resampled average approximation, is the smallest for all classes and it is followed by the Moment matching. The Monte Carlo with naive approach and the Monte Carlo sampling present much larger standard deviations.

Furthermore, we notice that the average value of the objective function in the Moment matching is 11.66% and 12.19% lower compared, respectively, to the Monte Carlo sampling and to the Resampled average approximation in the small trees. This difference increases in large trees. In the medium class, it is 33.77% and 33.23% lower. For the large class, it becomes 47.74% and 47.66% for the instance Large_A and 45.33% and 45.96% for Large_B. We believe that this

Table 11 – Initial allocation of decision variables (%)

Name	Fixed Income		Bovespa Index		Small Cap index	
	Mean (%)	Std. Dev.	Mean (%)	Std. Dev.	Mean (%)	Std. Dev.
Small						
Moment matching	44	0.28	56	0.28	0	0
Monte Carlo	77	0.08	20	0.07	3	0.10
Resampling	79.31	2.58	18.23	1.57	2.46	1.96
Medium						
Moment matching	37	0.21	63	0.21	0	0
Monte Carlo	77.76	0.01	22.00	0.01	0.24	0.00
Resampling	77.71	0.53	21.91	0.42	0.36	0.26
Large						
Moment matching_A	51.40	0.32	48.60	0.32	0	0
Moment matching_B	48.57	0.30	51.43	0.30	0	0
Monte Carlo_A	74.50	0.16	20.54	0.08	4.96	0.13
Monte Carlo_B	74.36	0.19	19.49	0.14	6.15	0.12
Resampling_A	75.13	2.30	19.60	1.48	5.27	2.17
Resampling_B	72.8	2.66	18.27	1.33	8.93	1.92

happens due to the adjustment in the tree. When the second moment is adjusted, volatility might be reduced and, thereby, the extreme scenarios (good or bad) are probably taken out of the sampling. Clearly, the Resampled average approximation and the Moment matching dominate the other two methods in terms of the stability of the objective function.

In relation to the initial asset allocation (Table 11), the investment in fixed income is quite similar for the Monte Carlo sampling, the Moment matching and Resampled average approximation. The main difference is that the Moment matching never allocates capital in the Small Cap index. The volatility of the Monte Carlo sampling and the Moment matching are also similar. The Resampled average approximation and Monte Carlo sampling present similar allocations, but very different allocation volatilities. However, given that the Resampled average approximation takes expectations from the Random sampling, the volatility of the allocation is close to zero. This is consistent with the methodology, as the Resampled average approximation mitigates volatility using the expected value from many trees, making the initial allocation smoother. Notice that, for the Monte Carlo with naive approach, by design, the initial allocation is constant among the assets, so we omit the results from Table 11.

In terms of the stability of the objective function, the Moment matching and Resampled average approximation dominate both Monte Carlo sampling and with naive allocation strategy. The stability of the initial allocation is similar for the Monte Carlo sampling and the Moment matching. Furthermore, the initial allocation's low volatility, presented by the Resampled average approximation, is a result of taking an average of averages. Considering not only the mixed results obtained on the initial asset allocation stability but also the argument of Dempster, Medova e Yong (2011), that a flat plateau can make many different allocations, resulting in very close values of objective functions, we can focus on the objective function stability results. Then, clearly, the Moment matching and Resampled average approximation present more stable solutions.

6 Conclusions

In this study, we provided an empirical discussion of the differences among methods to generate scenario trees for stochastic programming in ALM. We examined Monte Carlo sampling, Moment matching, the Resampled average approximation and Monte Carlo with naive allocation strategy as the benchmarks.

Our empirical analysis approached financial scenario trees with three assets. We defined four distinct topologies, which were assigned in three classes (small, medium and large). We ran the simulations so that the model would be robust to represent a realistic environment, but we regarded the technological limitations. The number of assets is a key point for the size of financial scenario trees. It should be arranged so that it is smaller than or equal to the number of scenario tree branches in order to guarantee the absence of arbitrage conditions (GEYER; HANKE; WEISSENSTEINER, 2010). Therefore, the model becomes computationally intractable very quickly when the number of assets rises, since they define a lower bound to number of branches, which can be used to calculate the exponential growth of scenario tree (GÜLPINAR; RUSTEM; SETTERGREN, 2004a).

The simulations are assessed by in-sample point of view. Thus, the best approximation is the one that minimizes the error between the ”true” objective value and the optimal scenario generation method objective value (KAUT; WALLACE, 2007). Considering the obtained results of the objective function, we can conclude that the Moment matching and the Resampled average approximation are more efficient in terms of in-sample stability, when compared to Monte Carlo sampling and Monte Carlo with naive allocation. For the initial portfolio allocation, the stability of the Moment matching and Monte Carlo sampling are similar and dominated by the Resampled average approximation. However, the low volatility of the Resampled average approximation is a result of taking an average of averages. Taking all the results into account, the Moment matching and the Resampled average approximation are more appropriate for ALM scenario generation.

There are other opportunities for subsequent studies. Scenario reduction and parallel implementation are techniques developed to deal with the curse of dimensionality in stochastic programming problems (BERALDI; SIMONE; VIOLI, 2010; DUPAČOVÁ; GRÖWE-KUSKA; RÖMISCH, 2003; HEITSCH; RÖMISCH, 2003). For instance, advances in the definitions of bounds (HENRION; KÜCHLER; RÖMISCH, 2009), different metrics (HEITSCH; RÖMISCH, 2007) and clustering (BERALDI; BRUNI, 2014) have been proposed. The integration of these techniques with different scenario generation methods might present promising results.

Acknowledgements

This work was funded by the following Brazilian Research Agencies: CAPES and FAPERGS.

Part III

Implicit Extensive Form for Multistage Stochastic Programming Models: A New Approach

Abstract

Deterministic equivalent models allow us to rewrite stochastic programing problems from a computational perspective. Nonetheless, these models become computationally intractable quickly with the increase on the number of stages or the probability distribution sampling. In this context, we propose a framework for the scenario tree generation and optimization of multistage stochastic programming problems. Relying on the Knuth transform, we generate the scenario trees, taking advantage of the left-child, right-sibling representation and making simulation more efficient. We also present a reformulation for the optimization model with an implicit extensive form approach, using a filtration process with bundles. We adopt an asset-liability management multistage stochastic model with joint chance constraints as an application to test this framework. The use of these methodologies saved enough computational resources, enabling us to find the optimal solution for instances with more than 160,000 scenarios in a few minutes, without the need of any relaxation or decomposition mechanism.

keywords: Stochastic Programming. Implicit Deterministic Equivalent. Algebraic Language Modeling. Multistage. Scenario Tree. Knuth Natural correspondence.

1 Introduction

Stochastic programming (SP) is a representation of practical problems in which uncertainty plays a major role. These problems can be approximated by equivalent convex problems that should be feasible, solvable, dualizable and stable (WETS, 1966; WETS, 1974). These equivalent representations are not only known as deterministic equivalent but also denominated as extensive form. To reproduce reality properly, the size of these models might become large, turning them out to be computationally expensive (or intractable). Thus, some research streams have risen to deal with this complexity. Among many propositions, we can mention the solution algorithms that exploit the model structure, such as those based on interior point methods (GASSMANN, 1990; LUSTIG; MULVEY; CARPENTER, 1991; BIRGE; HOLMES, 1992; BERGER et al., 1995; BERKELAAR et al., 2005; SUN; LIU, 2006), branch and bound (CARØE; SCHULTZ, 1999; AHMED; TAWARMALANI; SAHINIDIS, 2004), branch and price (LULLI; SEN, 2004; SINGH; PHILPOTT; WOOD, 2009), branch and cut (ZHANG; KÜÇÜKYAVUZ; GOEL, 2014), branch cut and price (NOWAK, 2006) and branch and fix (ALDASORO et al., 2017). There are also the scenario decomposition methods, which are based on Bender's decomposition (SLYKE; WETS, 1969; BIRGE, 1985; GASSMANN, 1990; DANTZIG; INFANGER, 1993; RUSZCZYŃSKI, 1997; CONSIGLI; DEMPSTER, 1998b; EGGING, 2013) or augmented Lagrangians (ROCKAFELLAR; WETS, 1991; MULVEY; RUSZCZYŃSKI, 1995; ROSA; RUSZCZYŃSKI, 1996; TAKRITI; BIRGE, 2000; SCHULTZ, 2003; GOEL; GROSSMANN, 2006; HIGLE; RAYCO; SEN, 2009). Another approach to dealing with SP deterministic equivalent implementation is the use of more compact representations with algebraic modeling languages (AMLs). Some optimization modeling languages have already been proposed: GAMMS (BROOK; KENDRICK; MEERAUS, 1988), AMPL (FOURER; GAY; KERNIGHAN, 1990), MODLER (GREENBERG, 1992), AIMMS (BISSCHOP; ENTRIKEN, 1993), and other examples are found on Kallrath (2013). Their benefits were outlined as a lower overall cost alternative to the matrix generators (FOURER, 1983). Kuip (1993) drew attention for the contributions of the AMLs in terms of its understandable, maintainable and verifiable formulation. These languages can even be integrated with algorithms, enabling them to solve optimization problems faster. The early versions of these AMLs did not include support for more complex data structures. In such a way, they have been improved and extended, which made possible the increase of their capability to describe a larger range of models, for instance, piecewise-linear functions and networks (FOURER; GAY, 1995)[1].

SP was also covered by the AMLs see e.g. (GASSMANN; IRELAND, 1995; GASSMANN; IRELAND, 1996; KALL; MAYER, 1996; MESSINA; MITRA, 1997; BUCHANAN; MCKINNON; SKONDRAS, 2001; FOURER; LOPES, 2006; WALLACE; ZIEMBA, 2005; KARABUK, 2008; VALENTE et al., 2009). In this field, the development of the algebraic languages enabled the formalization of some discrete stochastic processes as an event tree describing the unfolding of the uncertainty over the planning period. This event tree is comprised of a collection of scenarios (also known as paths), which are determined by conditional decisions from the initial stage

[1] See Entriken (2001) and Colombo et al. (2009) for more extensions

(tree root) to the final stage (tree leaf). These scenarios are the realizations and samples of the probability distributions (WALLACE; ZIEMBA, 2005). Hence, the event tree establishes an approximation of the stochastic processes' probability distribution. As the simulation of this scenario tree is done before the optimization process, we denote this process as the overall optimization or non-recursive method (GÜLPINAR; RUSTEM; SETTERGREN, 2004b; PFLUG, 2012).

Using AMLs, we should be able to provide a specification of the optimization model that must be correspondent with the event tree, such as that both structures must have the same features (cardinality, scenario links, number of stages, number of scenarios). This congruency between the simulated scenario tree and algebraic optimization model brings some challenges. The constraints that establish the link between the parent-child pair of each node are particularly difficult to be generated from an AML. This procedure is determined by the nonanticipativity constraints. They prevent a decision that is taken now from using information that will only become available in the future. Hence, scenarios that share a common history up to a point must have identical solutions until their information paths diverge. This is what makes the stochastic programming approach so realistic and its resulting computational problems so challenging to be solved (FOURER; LOPES, 2006). This difficulty comes from the lack of standard description of the event tree or, more precisely, the lack of a tree-structure indexing systems in AMLs. The algorithms based on decomposition, for instance, have been addressing the nonanticipativity constraints through their relaxation. These algorithms break the main problem (also called master problem) in smaller instances to fit them into the computer memory. In this approach, it is necessary to produce redundant data, in other words, independent copies of decision variables corresponding to every ancestor in the optimization model are generated for every child of this node. The same process must be done for the event tree simulation. The data from nonanticipativity scenario portion is replicated, taking them as independent scenarios (WALLACE; ZIEMBA, 2005). Furthermore, nonanticipativity has to be defined by explicit constraints that bind these redundant data. The resulting mathematical program is large, but the algorithms may take advantage of the added structure (GASSMANN; GAYL, 2005). Consequently, they overload the size of any multistage stochastic program and they do not operate at the optimum solution of the problem (GUPTA; GROSSMANN, 2011). There are some language extensions whose syntax are mostly based on the explicit formulation of the nonanticipativity , for example, the SPiNE (VALENTE; MITRA; POOJARI, 2005). Gassmann e Ireland (1995), Gassmann e Ireland (1996) realized that stochastic programming modeling could greatly benefit from the implicit declaration of scenarios. In this approach, the nonanticipativity is determined implicitly, i.e. there is only one dataset, so only one set of decision variable is defined. This approach can be applied to the simulation, sampling only the distinguishable realizations, or it can also be used on the optimization model when the concept of bundles is employed. The implicit extensive form of stochastic program reduces the size of the model because only the essential random and decision variables have to be described, becoming as brief as possible (GASSMANN; GAYL, 2005). Based on this methodology, Valente et al. (2009) created a notation that makes the filtration process the central syntactic construction of multistage stochastic recourse problems. In a similar way, Calfa (2014) used external matrices to inform the nonanticipativity condition implicitly and measured the memory demand decrease. Nevertheless, this last reduction is applied only on the optimization model, using the scenario generation to distinguish the paths.

In summary, this paper addresses the following issues: (i) the simulation and optimization performance for multistage stochastic problems, and (ii) the gap of compatibility between scenario generator and stochastic program from the implicit extensive form perspective. Therefore, we do not intend to make an extension on an AML language like Bisschop e Fourer (1996), Entriken (2001), Valente et al. (2009), Colombo et al. (2009), or create a modeling tool as Fourer e Lopes (2009), or even present a software like Kall e Mayer (1996), Messina e Mitra (1997), Karabuk (2008). Our aim is to gather ideas from Domenica et al. (2009) which argue mainly for the scenario generation (simulation), and from Calfa (2014) which mostly discuss the optimization problem, to provide a modeling framework that includes the stochastic process simulation and the optimization model. Our work implements the modeling implicit extensive form methodology on both simulation and optimization, defining also an interface between them.

The scenario generation preceding the overall optimization strategy is processed through C/C++ algorithm, which allocates a multi-way tree represented by an equivalent binary tree dynamically in such a way that only distinguishable paths are produced and only one database is created, giving origin to a compact event tree. We adopt AMLs to describe our implicit extensive stochastic optimization model. We describe the nonanticipativity condition for the event tree by using only a numeric vector that indicates a constant cardinality for each stage. The method is related with the information from the filtration process to compose the bundles, following some ideas from Valente et al. (2009). We extend the bundles concept for corresponding nonanticipativity constraints implicitly. Our bundles definitions are tuples that reference the relationship between a node and their children, which is slightly distinct from Valente (2011). This approach yields an implicit deterministic equivalent structure, which is generic enough to be applied on other SP models, for instance, the ones with recourse, chance constraints, or even integrated chance constraints.

Our contribution consists of the application of popular concepts, either in the context of stochastic programming, or in computer science, to formalize shorter stochastic programs with not only the implementation of an implicit nonanticipativity constraints based on bundles, but also using a multi-way scenario tree reformulated with the left-child-right-sibling strategy. We express the bundle concept to be more algebraically adequate with the AMLs, rather than having extensions. We propose a novel algorithm for bundles generation. We design a simpler data structure with dynamic allocation for the SP model event tree, in such a way that it becomes less expensive in terms of computational memory. These guidelines settle an innovative framework, allowing us to find the optimal solutions for an ALM multistage stochastic program, with joint chance constraint and with more than 160,000 scenarios without any relaxation or decomposition approach.

The remainder of this paper is organized as follows. In Section 2, we formulate a theoretic multistage stochastic model with joint chance constraint. We also describe two versions for its corresponding equivalent deterministic: the explicit and the implicit formulation of nonanticipativity constraints, respectively. In Section 3, we explain how to employ the filtration of a space probability and bundles to generate the implicitly extensive form of a multistage stochastic program. Furthermore, we detail, in Section 3, the scenario generation of an event tree driven by the filtration and bundles concepts. This framework is applied in an asset-liability management problem (ALM) in Section 4. We discuss the empirical results in Section 5. Final remarks are presented in Section 6. In the Appendix .1 and Appendix .2, we illustrate the use of an AML for encoding the implicit deterministic equivalent version of multistage stochastic program.

2 Multistage Stochastic Model with Joint Chance Constraint

A stochastic multistage optimization problem presents a set $\mathbb{X} \subseteq \mathbb{R}^N$ for some dimension N in which x are decision variables ($x \in \mathbb{X}$). The parameter $\xi \in \mathbb{R}^N$ is a continuous random variable (RV) defined in the probability space $(\Xi, \mathcal{F}, \mathbb{P})$. This RV could be, for instance, asset returns, demand for products or weather conditions discretized throughout multiple time stages. Without loss of generality, we consider stages as time periods and decisions as asset allocations. Thus, portfolio allocation decisions are made through discrete time steps $\tau \in \mathcal{T} := \{2, \ldots, T\} \in \mathbb{Z}$. The filtration $\mathfrak{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_T = \mathcal{F})$ originates a conditioned stochastic process such as $\xi_t \lhd \mathcal{F}_T$. The σ-algebra of the probability space is also indexed by the time, $\sigma(\xi) \subseteq \mathcal{F}$. This stochastic process, also known as filtration process, defines how the new information becomes available to the decision maker at each point in time (VALENTE et al., 2009).

The decisions through the time take into account the gain of information which is influenced by the stochastic process. The allocation in each asset i at time τ also comprises conditional decisions known as a policy, i.e.

$$x_{\tau,i} = \{x_{1,i}, x_{2,i}(\xi_{1,i}), \ldots, x_{\tau,i}(\xi_{1,i}, \ldots, \xi_{\tau-1,i})\}. \tag{2.1}$$

Considering that $i = 1, \ldots, N$, we can also define a vector that comprehends the portfolio allocation.

$$x_\tau = \{x_{\tau,1}, x_{\tau,2}, \ldots, x_{\tau,N}\} \tag{2.2}$$

Notice that this is a dynamic problem, meaning that the decisions of period τ include the information from period 1 to $\tau - 1$. As discussed by Zhang, Küçükyavuz e Goel (2014), it differs from the classical static models in which decisions are taken in the beginning and kept fixed without an update once uncertainty is revealed . Thus, in the dynamic setting, the decisions are adapted to the state of the random variables, considering the remaining future uncertainty of the system (HANEVELD; STREUTKER; VAN DER VLERK, 2010).

A generic stochastic multistage programming problem with joint chance constraint is presented through the optimization problem (2.3) − (2.7).

$$\text{Maximize } cx + \sum_{\tau=2}^{\mathcal{T}} \mathbb{E}_{\xi_\tau}\{Q(x_\tau, \xi_\tau)\} \tag{2.3}$$

s.t.

$$Ax = b \tag{2.4}$$

$$T(\xi_\tau)x_{\tau-1} + W(\xi_\tau)x_\tau(\xi_\tau) \leq h_\tau \qquad \forall\, \tau \in \mathcal{T} \tag{2.5}$$

$$P\{g(x_\tau(\xi_\tau), \xi_\tau) \geq 0, \forall\, \tau \in \mathcal{T}\} \geq \zeta \tag{2.6}$$

$$x \in \mathbb{X} \tag{2.7}$$

The objective function (2.3) presents the first stage terms and the expected value of the Q function. In the financial context, the value function, $Q(x_\tau, \xi_\tau)$, denotes the utility of choosing the portfolio x_τ in stage τ with optimal allocations in all the subsequent periods. We assume $Q(x_\tau, \xi_\tau)$ as linear and continuous function. Deterministic constraints of the first stage are represented by (2.4). The inequality constraints (2.5) link both the previous stage with the forward stages. The respective parameter's matrices of constraints (2.5) are: the technology matrix, T, which determines the impact from the previous stages, and the recourse matrix, W, which models the result from the recourse decisions. T has dimensions of $((\tau - 1) \times N)$; W has dimensions of $(\tau \times N)$; and $h_\tau := \{h_1, \dots, h_\tau\}$ is a deterministic vector of dimension τ. The joint chance constraint (2.6) is a probabilistic constraint which depends on the recourse actions, i.e., $g(x_\tau(\xi_\tau), \xi_\tau) = h_\tau - T(\xi_\tau)x_{\tau-1} + W(\xi_\tau)$. Constraints (2.7) limit the solution for a feasible polyhedra. Next we present, respectively, the explicit and implicit equivalent formulation of the stochastic programming problem.

2.1 Explicit deterministic equivalent formulation

The model formalized through Eq. (2.3) $-$ (2.7) can be reformulated as a large linear program, which is known as a deterministic equivalent or extensive form. Let S be a scenario tree describing how ξ develops randomly over time. A path of price realizations in S is called a scenario. The realized version of ξ is formalized as the vector $\omega_\tau^s := \{\omega_{\tau,1}^s, \dots, \omega_{\tau,N}^s\} \in \mathbb{R}^N$ for all assets available in time period τ and scenario s. The decision variable vector x_τ^s is also described by $x_\tau^s := \{x_{\tau,1}^s, \dots, x_{\tau,N}^s\}$ for all N assets available in the period time τ and scenario s. This extensive form is defined by the Eq. (2.8)$-$(2.13).

$$\text{Maximize } \omega_1 x_1 + \sum_{t=2}^{\mathcal{T}} \sum_{s=1}^{S} p_\tau^s \omega_\tau^s x_\tau^s \tag{2.8}$$

s.t.

$$A_1 x_1 = b_1 \tag{2.9}$$

$$T_{\tau-1}^s x_{\tau-1}^s + W_\tau^s x_\tau^s \le h_\tau^s \qquad \forall\, s \in S, \forall\, \tau \in \mathcal{T} \tag{2.10}$$

$$x_\tau^s = x_\tau^z \text{ if } \lambda(\omega_\tau^s) = \lambda(\omega_\tau^z) \qquad \forall s, z \in S, \forall\, \tau \in \mathcal{T} \tag{2.11}$$

$$\sum_{s=1}^{S} p^s \eta(g(x_\tau^s, \omega_\tau^s)) \le 1 - \zeta \qquad \tau \in \mathcal{T} \tag{2.12}$$

$$x_\tau^s \ge 0 \qquad \forall\, s \in S, \forall\, \tau \in \mathcal{T} \tag{2.13}$$

The objective function (2.8) corresponds to the function $\mathbb{E}_{\xi_\tau}$ which is defined as a risk-neutral expectation function. The first stage terms are added with this expected value of the following periods and stage realizations. The function Q is the inner product vector between the arguments, $Q(x_\tau, \xi_\tau) = (\xi_\tau)^\top x_\tau$, in which the operator $(\cdot)^\top$ is the transpose operator. We denote p_τ^s as the probability associated with the S samples in each time period τ of the random variable ξ_τ. The deterministic first stage decision is defined by constraint (2.9). The technology matrix and the recourse matrix remain with the same function as in constraint (2.5), but they are indexed by the sampled realizations in the constraint (2.10).

The nonanticipative constraints are indicated by the constraint (2.11). They are defined as the subset $\mathcal{D} \subseteq \mathcal{F}$ whose scenarios are indistinguishable, in the same fashion as described by Mulvey e Ruszczyński (1995). Hence, considering the conditional realized vector $\lambda(\omega_\tau^s) := \{\omega_1^s, \ldots, \omega_\tau^s\}$ for the scenario s, if there are two scenarios s, z and their respective conditional realized vectors $\lambda(\omega_\tau^s) = \lambda(\omega_\tau^z)$, then we must have $x_\tau^s = x_\tau^z$. In other words, the decision variable must be the same. In the explicit extensive form, these constraints need to be specified in the model.

The probabilistic constraint is rewritten on constraint (2.12). We use an indicator function $\eta(a) = 0$ if $a \leq 0$ and 1 if at least one component is strictly positive, (BIRGE; LOUVEAUX, 2011). The expected value from η must satisfy a minimal confidence level. This constraint defines a risk measure for the problem. We define non-negativity of decision variables in the constraints (2.13).

2.2 Implicit deterministic equivalent formalization

The implicit representation of a multistage stochastic model is viewed as a compact version of a particular model. The structure given by the scenario tree can be enforced by definition of a reduced number of decision variables, for which the nonanticipativity constraints are implicitly satisfied. The implicit formulation is given by (2.14)−(2.18).

$$\text{Maximize } \omega_1 x_1 + \sum_{t=2}^{\mathcal{T}} \sum_{s=1}^{S} p_\tau^s \omega_\tau^s x_\tau^s \tag{2.14}$$

s.t.

$$A_1 x_1 = b_1 \tag{2.15}$$

$$T_{\tau-1}^s x_{\tau-1}^s + W_\tau^s x_\tau^s \leq h_\tau^s \qquad \forall\, n_{\tau,s} \in desc_{s-1}, \tau \in \mathcal{T} \tag{2.16}$$

$$\sum_{s=1}^{S} p^s \eta(g(x_\tau^s, \omega_\tau^s)) \leq 1 - \zeta \qquad \tau \in \mathcal{T} \tag{2.17}$$

$$x_\tau^s \geq 0 \qquad \forall\, s \in S, \forall\, \tau \in \mathcal{T} \tag{2.18}$$

This deterministic equivalent from the program (2.3)−(2.7) is obtained by refining the relationship definition between the decision variables in $s-1$ and s. Suppose that $n_{\tau s}$ is a discrete point (node) from the event tree in time period τ and scenario s, then, we refer to its set of children or descendants as $desc_s$. Thus, we determine the variable x_τ^s as the decision to be taken at time τ under all scenarios $s \in desc_{s-1}$, (FOURER; LOPES, 2009; VALENTE, 2011). In the implicit reformulation, we do not need to express the nonanticipativity explicitly, as in constraint (2.11).

3 Proposed modeling framework

In this section, we describe the modeling framework proposed in this work, as depicted in Figure 5. The filtration informs how the data will be disclosed in the scenarios of an event tree. Our objective is to design a filtration approach and to show its benefits in terms of problem size reduction. In Section 3.1, we present how to operationalize the filtration into bundles, which is followed in Section 3.2 by an algorithm for bundle generation. This approach allows us to compute multistage stochastic programs without explicit nonantecipative constraints. Furthermore, in Section 3.3, we analyze the size reduction entailed by this method. Finally, in Section 3.4, we discuss the implicit scenario generation process.

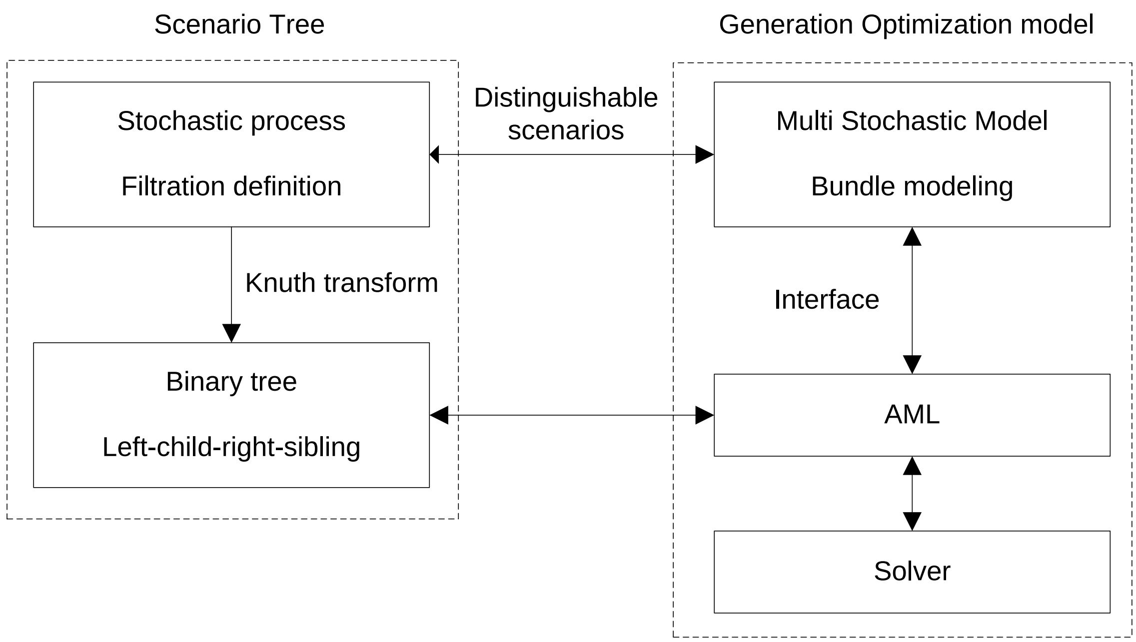


Figure 5 – Framework overview.

3.1 Filtration and Bundle Definition

We adopt the filtration process as the key feature for modeling the multistage stochastic optimization model. Not only considering the space probability $(\Xi, \mathcal{F}, \mathbb{P})$ defined in Section 2 but also denoting Ξ_t the set of all nodes at stage $t \in \{1, \ldots, T\}$, we formalize a correspondence among the elements of set Ξ_T and the sigma algebra $\mathcal{F}_T$ of all Ξ_T subsets. According to Shapiro, Dentcheva e Ruszczynski (2009), the set Ξ_T can be denoted as the union of disjoint sets $desc_{\tau-1}$, $\forall\, n_{\tau,s} \in \Xi_{T-1}$, which represent all nodes contained in the Ξ_{T-1}. The collection of these sigma algebras $\mathcal{F}_1 \subset \cdots \subset \mathcal{F}_T$ is called filtration. The sigma algebra $\mathcal{F}_1$ is mapped by the tree root, i.e. $\mathcal{F}_1 = \{\Xi_T\}$. Figure 6 provides an example of event tree representation used in Valente (2011).

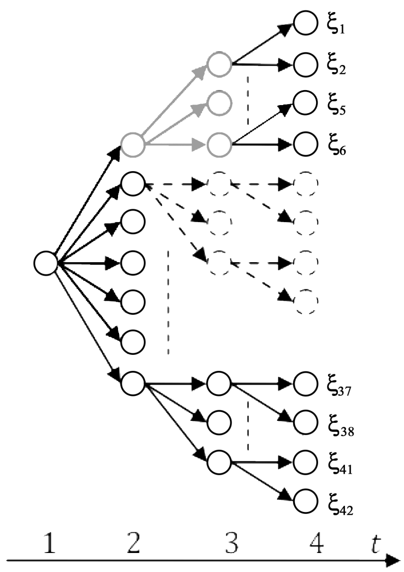


Figure 6 – Example of event tree.

In this example, the initial period is the root and it has only one node. This first element originates seven children. Thus, in the second period, each child becomes the parent of three more nodes. They are connected by arcs that come from the parents to their children. In Figure 6, we construct a scenario tree with 4 stages, 42 scenarios and 71 nodes. A similar framework could be used for any scenario tree model but, depending on the application, with a different number of periods and branches for each period. The following sets comprise the $\sigma-$algebra for the filtration generation process in the tree from Figure 6.

$$
\begin{aligned}
\mathcal{F}_1 &= \sigma\{\Xi_T\},\\
\mathcal{F}_2 &= \sigma\{\{\xi_1,\xi_2,\ldots,\xi_6\},\{\xi_7,\xi_8,\ldots,\xi_{12}\},\ldots,\{\xi_{37},\xi_{38},\ldots,\xi_{42}\}\}\\
\mathcal{F}_3 &= \sigma\{\{\xi_1,\xi_2\},\{\xi_3,\xi_4\},\ldots,\{\xi_{41},\xi_{42}\}\}\\
\mathcal{F}_4 &= \sigma\{\{\xi_1\},\{\xi_2\},\ldots,\{\xi_{42}\}\}
\end{aligned}
$$

With this filtration characterization, we can start discussing the bundles definition. A bundle is associated with the nodes of the tree and consists of a stage and scenario. Rockafellar e Wets (1991), King (1994), Fourer e Lopes (2009) and Valente et al. (2009) use the same information structure to constitute bundles with tuples that inform (t, s). The use of bundles for a description of the implicit extensive form guarantees that the database is generated without redundancy.

Our bundle structure is in the same direction as Valente et al. (2009) and Valente (2011) with some significant differences. Unlike Valente et al. (2009) that employs a new syntax to describe the relationship between the single-stage linear programs, we regard the filtration as a tree representation through bundles containing only elementary scenario identifiers. Contrasting with Valente (2011) in which elements of each set are composed only by the number of scenarios in the last period, we enumerate each component for all time stages. Thus, the elements of bundles are tuples that represent the inter-temporal relationship between the index of a node in $s-1$ and its descendants in s. Defining S_t as the number of nodes in stage t, and n_{ts} as the s-th node of the t-th stage of the event tree, with $s \in \{1,\ldots,S_t\}$ and $t \in \{1,\ldots,T\}$, then, the relationship between node n_{ts} and the scenarios passing through it are identified by $\mathcal{B}_{ts}$. These bundles are formalized in Eq. 3.1:

$$\mathcal{B}_{ts} = \{(a, b) \in [1, \ldots, S_t] \times [1, \ldots, S_{t+1}] : a = s, b = desc_s\}. \tag{3.1}$$

Based on the event tree from Figure 6, our bundle's definition is exemplified through Table 12.

Table 12 – Bundles under our proposition for example event tree

Stage	Bundles
1	$\mathcal{B}_{11} = \{(1,1),(1,2),\ldots,(1,7)\}$
2	$\mathcal{B}_{21} = \{(1,1),(1,2),(1,3)\}$, $\mathcal{B}_{22} = \{(2,4),(2,5),(2,6)\}$, ..., $\mathcal{B}_{27} = \{(7,19),(7,20),(7,21)\}$
3	$\mathcal{B}_{31} = \{(1,1),(1,2)\}$, $\mathcal{B}_{32} = \{(2,3),(2,4)\}$, ..., $\mathcal{B}_{321} = \{(21,41),(21,42)\}$

For instance, the bundle $\mathcal{B}_{21}$ in Table 12 is comprised by the first node of the second time period and three of its descendants. This definition incorporates into bundles the relationship between the event tree components, through different time stages. It becomes very informative for the interpretation of the intertemporal constraints in the optimization problems, since they can be rewritten taking advantage of existing features of algebraic modeling languages. Considering this concept of bundles, constraint (2.16) could be reformulated as constraint (3.2):

$$T^a x^a + W^b x^b \leq h^b \quad \forall\, (a,b) \in \mathcal{B}_{\tau,s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}. \tag{3.2}$$

An implementation example of a binary tree with an algebraic modeling language (AMPL) is detailed in the Appendix sections .1 and .2.

3.2 Algorithm for Bundle Generation

In this section, we present the algorithm designed to generate the bundles. Considering a tree in which the number of branches is constant for each node in a given stage, we can define a vector Ψ_t with $t \in \{1, \ldots, T\}$ indicating how many descendants come from each node in each stage. For example, based on Figure 6, a scenario tree could be defined as $\Psi := [1, 7, 3, 2]$. Thus, we have T (4) elements on this vector in which $\Psi_1 = 1, \Psi_2 = 7, \Psi_3 = 3$ and $\Psi_4 = 2$. Furthermore, we calculate S_t with $t \in \{1, \ldots, T\}$ based on Ψ_t. As the first time period is the root tree, we have S_1 equal to one. The others time steps of vector S_t are defined by the Eq. (3.3)

$$S_t = S_{t-1} \cdot \Psi_t \qquad \forall\, t \in 2, \ldots, T. \tag{3.3}$$

With these definitions, we provide algorithm 2 which implements the bundles generation with Ψ as an input.

The procedure in Algorithm 2 generates a list with all bundles for the model. The algorithm implements the algebraic formulation presented in Eq. (3.1). Throughout the time

Algorithm 2: Procedure for generation of the bundles list

```
Links (Ψ, S, T);
Input  : Ψ: numerical vector that represents the cardinality for each scenario in each time
         period, T: number of time periods, S: numerical vector determinated by the
         number of scenarios for each time period
Output : ListOfTuples: List of tuples for each time period
while t ← 1 to T do
    if t > 1 then
        a ← 1, cen ← 1, b ← 1
        while b ≤ S_t do
            if cen > Ψ_t then
                a ←a +1
                cen ← 1
            end
            ListOfTuples_t ← (a,b)
            b ←b +1
            cen ←cen +1
        end
    end
    t ←t +1
end
return ListOfTuples;
```

periods, we compose pairs with indexes from stage $t-1$ and t. These references are aggregated using the Ψ vector as a reference. The index b is enumerated one-by-one up to the number of scenarios delimited by S_t. An auxiliary variable, *cen*, is also defined to check each time that the cardinality is achieved. Every time that the counter *cen* is greater than the cardinality indicated by Ψ_t, we increment the index a and the counter *cen* is initialized again with 1.

3.3 Model Size Reduction with Bundles

The focus of this section is to show the size reduction of the problem in terms of its variables and constraints with the use of bundles. We compare the reduction obtained by an implicit model versus an explicit one. In other words, we contrast a multiple stage optimization problem with implicit to one with explicit nonanticipativity constraints.

Considering the general optimization problem defined in Section 2, we have the decision variable x with dimension N, e.g. asset classes. Then, the overall number of variables for the implicit equivalent deterministic model could be calculated by Eq. (3.4).

$$\left(\sum_{t=1}^{T} S_t\right) \cdot N \tag{3.4}$$

Contrasting the implicit model, in the extensive approach, scenarios are simulated and optimized independently. However, they must still satisfy the nonanticipativity condition, which means that decisions must be the same over indistinguishable scenarios at a given stage. This approach increases the number of variables when compared to the implicit model. The number of the variables in the extensive representation is defined in Eq. 3.5.

$$T \cdot S_T \cdot N \tag{3.5}$$

Therefore, when we choose the implicit extensive form for the multistage stochastic problem, we reduce the size problem in terms of its variables. The variable reduction is given by Eq. (3.6) which is obtained by subtracting Eq. (3.4) from (3.5).

$$\sum_{t=2}^{T} (S_T - S_t) \tag{3.6}$$

The implicit extensive form also fosters a decrease in the number of constraints, once there is no need for formalizing the nonanticipativity constraints. The economy, in terms of constraints, is equivalent to the number of variables in the extensive model and it is presented in Eq. (3.5).

3.4 Implicit scenario generation

The random variables are discretized through the scenario generation. These uncertain values, e.g. asset returns, give origin to finite outputs that comprise a dependent sequence of events (stages) denoting the scenario. For each time period, these stages are succeeded by several possible realizations (DUPAČOVÁ et al., 2000; GÜLPINAR; RUSTEM; SETTERGREN, 2004b). Our objective is to produce a compact simulation that generates and saves only the distinguishable scenarios. According to Domenica et al. (2009), since the event tree must be compatible with the optimization model, our approach implements a C/C++ algorithm whereby a complete multi-way tree mimics an event tree. We underpin our algorithm with left-child, right-sibling representation. This data structure comprehends nodes that are linked only by two classes of arcs (sibling or child). The elements that pertain to the same period are classified as siblings (rightmost nodes), while the elements that belong to next period are pointed by child link (leftmost node) (PFALTZ, 1975; FREDMAN et al., 1986; CORMEN, 2009). Through this methodology, it is possible to reformulate any class of rooted trees as binary rooted tree by linked data structure. The algorithm used for this transformation is known as *Knuth transform* or *natural correspondence*, which was proposed by Knuth (1998). Pfaltz (1975) provides the algorithm which describes the Knuth transform for a rooted multi-way rooted tree.

We redesign a multi-way scenario tree into a binary tree using a method based on Knuth transform. Suppose a directed graph, denoted by $G = (P, E)$, is a set P of points (or data items) together with a binary relation, E, (or set of ordered pairs) defined on P. Furthermore, we define the theoretical mathematic simulation of event tree G_m, and its computer representation G_R. As argued by Pfaltz (1975), a representation is faithful if, from G_R alone, a computer procedure can reproduce all of the information that exists in G_m. We characterize as elements of P_{G_m} [1] as the tree nodes that are sequentially numbered, i.e.

$$P_{G_m} := \left\{ 1, 2, \ldots, \sum_{t=1}^{T} S_t \right\}. \tag{3.7}$$

[1] We subscribe the graph whose component pertain.

They indicate the nodes n_{ts} pertaining to the scenario tree. The set E_{G_m} is defined using the collection of $\mathcal{B}_{ts}$, see Eq. (3.1) for bundles definition. The Knuth transformation is seen as a function $f : G_m \rightarrow G_R$. Thus, we rewrite each point $x \in P_{G_m}$, which composes the model G_m, as a component $c_x \in P_{G_R}$. The set E_{G_R} is defined through children and sibling edges. Their formulation is given, respectively, by Eq. (3.8) and Eq. (3.9).

$$\{a : (a, b) \in \mathcal{B}_{t1}\} \qquad \forall\, t \ \in T \tag{3.8}$$

$$\{b : (a, b) \in \mathcal{B}_{ts}, \forall\, s \in S_t\} \qquad \forall\, t \ \in T \tag{3.9}$$

Figure 7 shows an example for the G_m and G_R representation from the tree already illustrated in the Figure 6. G_R is represented as a linked list, which is allocated dynamically in

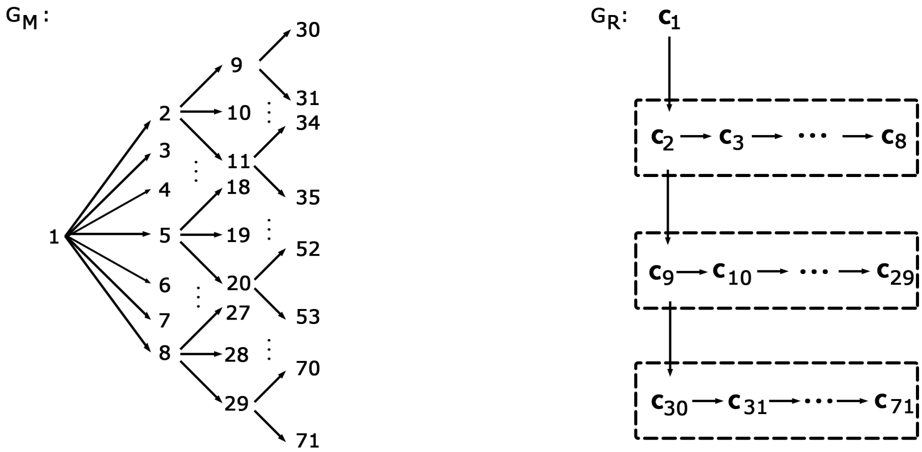


Figure 7 – Our scenario simulation proposition

the principal memory. Thus, only two link fields are necessary to completely represent the three structured relationships between the points (PFALTZ, 1975). Thus, the cardinality of E_{G_R} is $(T-1)+\sum_{t=1}^{T}(S_t-1)$ links. Consequently the memory demand is lower than the usual structure used to implement the multi-way tree, which has an array or list of pointers for all children that use $\sum_{t=2}^{T}(\Psi_t \cdot S_{t-1})$ links. Once that data structure is available in the memory, see Appendix .3, we have defined the insertion and search operations. They are based on Ψ which will determine how many elements compose each scenario, since they are all linked in the data structure. In terms of computational time and complexity, we can derive an exponential asymptotic behavior, since we have different operations of complexity $O(m \cdot n)$ multiplied by the number of tree nodes $(\sum_{t=1}^{T} S_t)$ whose growth is also exponential.

4 Practical Application - ALM

The asset-liability management (ALM) is a classical financial problem. It can be defined by a balance sheet presenting assets, liabilities and the surplus. The aim of the manager is to allocate the wealth throughout the time in such a way to be able to pay the liability until the last time period, meanwhile respecting the features of the problem, for instance, laws and requirements determined by pension funds authority. The ALM problem can be viewed: (i) dynamically because its current portfolio position is a result of previous actions (CONSIGLI; DEMPSTER, 1998a), (ii) stochastically due to the uncertain nature of asset prices and liabilities (ZIEMBA, 2003), and (iii) as a multistage problem as regular discrete time periods are defined in order to review the asset allocation and guarantee the liability payment (DUPAČOVÁ et al., 2000; PFLUG, 2001). ALM models may be applied in many different financial environments, for instance, banks (KUSY; ZIEMBA, 1986; FERSTL; WEISSENSTEINER, 2011; URYASEV; THEILER; SERRAINO, 2010), insurance (CONSIGLIO et al., 2001; CONSIGLIO; SAUNDERS; ZENIOS, 2006), pension funds (CARIÑO et al., 1994; KOUWENBERG, 2001; ZENIOS; ZIEMBA, 2006; ZENIOS; ZIEMBA, 2007; OLIVEIRA et al., 2017) and debt management (CONSIGLIO; STAINO, 2012; VALLADÃO; VEIGA, 2014). Furthermore, the stochastic multistage ALM problem has been addressed before; some examples are in Cariño et al. (1994), Consigli e Dempster (1998b), Carino e Ziemba (1998), Klaassen (1998), Kouwenberg (2001), Hilli et al. (2007), Consiglio, Cocco e Zenios (2007), Geyer e Ziemba (2008), Dupačová e Polívka (2009). They proposed models whose main purpose is to support decisions of long-term investors who want to achieve certain goals and meet future obligations through an investment police.

In this section, we present a practical application of stochastic programming in ALM. We describe a stochastic multistage dynamic ALM problem with joint chance constraint in Section 4.1. The deterministic equivalent of this stochastic problem is in Section 4.2.

4.1 ALM Multistage stochastic program with joint chance constraint

In this section, we use the implicit deterministic equivalent formalization, which is defined by Eq. (2.14)−(2.18), as support for a definition of ALM multistage stochastic model with joint chance constraint. The price for each asset is discretized through sampling, considering different realization (scenarios) throughout the time period. It gives origin to a conditioned stochastic process in the same way of Eq (2.1). Our portfolio is comprised by a vector such as Eq. (2.2). We want to consider different scenario possibility realizations in our model. Thus, we adopt the set S as a scenario tree, describing how ξ develops randomly over time, as defined in Section 2.1. In other words, the decision variables are also described by vector x^s_τ, and the realized version of ξ_τ is also denoted by ω^s_τ for all N assets available in the τ period time and in the s scenario used. We apply the bundles to formalize the nonanticipativity condition, see the definition on Eq. (3.1).

The decision vector x_1 is a first-stage variable (here-and-now), and it is determined by the initial wealth allocation. The vector ω_0 is the actual price of asset, which is known and deterministic before the first decision. Thus, the initial allocation constraint only allocates the initial wealth (W) through the available assets.

$$W = (\omega_0)^\top x_1 \tag{4.1}$$

The balance constraint among time periods is defined by the different corrective actions made by the wait-and-see auxiliary variables $B^s_\tau := \{B^s_{\tau,1}, \ldots, B^s_{\tau,N}\}$ for all N securities available in the time period τ and scenario s, and $V^s_\tau := \{V^s_{\tau,1}, \ldots, V^s_{\tau,N}\}$ for all N assets available in the period time τ and in the scenario s. They indicate, respectively, the number of shares bought or sold for each asset. They also determine the inter-temporal dependence of decisions, in the same way of the Eq. (3.2).

$$x^b - x^a = B^s_\tau - V^b \quad \forall\,(a,b) \in \mathcal{B}_{\tau,s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}. \tag{4.2}$$

In our model, we consider the short-term insolvency level which is defined by the fund's liquidity level. Therefore, the cash flow constraints take account only the respective time period liability.

$$(\omega^b)^\top V^b - (\omega^b)^\top B^b = l_\tau - f_\tau \quad \forall\,(a,b) \in \mathcal{B}_{\tau,s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}. \tag{4.3}$$

There are some policy constraints that are linked with the nature of asset classes, in other words, boundaries depending on their features. It is formalized by setting an upper bound in the allocation of these securities, which is described by some percentage of total portfolio value, denoted in Eq. (4.4) by π.

$$(\omega^b_i)^\top x^b_i \leq \pi\,(\omega^b)^\top x^b \quad i = 1, \ldots, N\ \forall\,(a,b) \in \mathcal{B}_{\tau,s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}. \tag{4.4}$$

A joint chance constraint is also formalized to guarantee an ζ solvency level for the fund through scenarios. The aim is to model the insolvency level using the scenarios which determined by the long-term pension fund solvency. In order to do that, the pension fund investment portfolio has to be greater than the long-term liability (L_τ), which is composed by the present value of discounted liability cash flows minus the present value of some extraordinary participants contributions (F_τ). This requirement is modeled similarly to a joint chance constraint as presented by Andrieu, Henrion e Römisch (2010), Haneveld, Streutker e VAN DER VLERK (2010), Zhang, Küçükyavuz e Goel (2014), Guigues e Henrion (2017). We want to guarantee the pension fund long-term solvency. Therefore, we define a joint chance constraint that sets the pension fund to have a portfolio value greater than the long-term liability discounted by the extraordinary contributions, which are multiplied by a risk-aversion factor K, with an ζ solvency probability.

$$\mathbb{P}\left(\sum_{i=1}^{N} \xi_{\tau,i} x_{\tau,i} \geq K(L_\tau - F_\tau) \quad \tau \in \mathcal{T}\right) \geq \zeta \tag{4.5}$$

We have the non-negative constraints in such a way that the short positions are not allowed.

$$x^b, B^b, V^b \geq 0 \quad \forall\,(a,b) \in \mathcal{B}_{\mathcal{T},s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}. \tag{4.6}$$

The function $\mathbb{E}_{\xi_\tau}$ is defined as a risk-neutral expectation and it is defined in Eq. (4.7).

$$\mathbb{E}_{\xi_\tau} = \sum_{s=1}^{S} p^s((\omega^b)^\top x^b) \quad \forall\,(a,b) \in \mathcal{B}_{\mathcal{T},s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T}, \tag{4.7}$$

in time period τ. Furthermore, the probability of scenario realization has to follow the $\sum_{s=1}^{S} p^s = 1$ and $p^s > 0$ $s \in S$.

4.2 Deterministic equivalent

The deterministic equivalent of Eq. (4.5) is defined through a set of mix-integer constraints. Firstly, we should identify the scenarios that are underfunded. This is represented by the Eq. (4.8), which sets to one the binary variable C_τ^s scenarios in that condition. Then, the model checks, along the time period, how many scenarios have underfunding nodes, Eq. (4.9), that is formalized by variable γ^s. Hence, as the total of insolvent scenario can not be greater than or equal to $(1-\zeta)$ multiplied by the scenario tree cardinality, Eq. (4.10).

$$K(L_\tau - F_\tau) - \sum_{i=1}^{N} \omega_i^b x_i^b \leq M C_\tau^s, \qquad \forall\,(a,b) \in \mathcal{B}_{\mathcal{T},s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T} \tag{4.8}$$

$$\gamma^b \geq C^b, \qquad \forall\,(a,b) \in \mathcal{B}_{\mathcal{T},s}, \forall\, s \in S_\tau, \forall\, \tau \in \mathcal{T} \tag{4.9}$$

$$\sum_{s=1}^{S_\mathcal{T}} \gamma^s \leq (1-\zeta)|S_\mathcal{T}| \tag{4.10}$$

5 Simulations and Empirical Results

In this section, we present the computational results obtained by optimizing the ALM problem described in Section 4 solving Eqs. (4.1)-(4.4) and Eqs.(4.6)-(4.10). Using the implicit extensive form through the filtration-oriented approach with bundles and the implicit scenario generation with natural correspondence (Section 3), we were able to perform the scenario generation and optimization of scenario trees with more than 160,000 scenarios, without ex-ante relaxation or decomposition procedure. We focus the results mainly on the computational times, but we also discuss some outcomes related to the in-sample stability of the objective function. The tests were conducted on an Intel(R) Core(TM) i7-4790 processor (3.6 GHz) with 16 Gbytes of memory RAM on 64bits-Windows 8.1. The scenario tree parameters are generated through the discrete stochastic differential equations. These stochastic generator processes are the Cox–Ingersoll–Ross model (COX; INGERSOLL; ROSS, 1985), CIR, and the Geometric Brownian Motion (DUFFIE, 2001; NEFTCI, 1996), GBM. In our tests, each scenario of the tree is consisted of three assets: a fixed-income asset and two stocks. The calibration is made using historical annualized daily prices realization. We collected data of the 1-month Brazilian LTN (similar to a T-Bill in the USA) as a proxy for the short-term interest rate, the Bovespa index (the most liquid stock index in Brazil) and the Brazilian Small Cap BM&F Bovespa index (index for small capitalization stocks) from January 2012 to November 2016. We present the calculated statistics for the model in Table 13.

Table 13 – Parameters used for the CIR and GBM.

Asset	Return Annualized (μ)	Std. Annualized (σ)	Mean Revert. (α)
Fixed Income	0.11296	0.04358	0.14599
Bovespa index	0.13510	0.23499	-
Small Cap index	0.07443	0.17748	-

Furthermore, the matching of the first two moments is adopted for the scenario generation (HØYLAND; WALLACE, 2001; HØYLAND; KAUT; WALLACE, 2003). We avoid the arbitrage opportunities in our tests constraining the optimization model to have only long positions (not allowing short selling and leveraging), and not using prices from the other instruments in the yield term structure of the Cox–Ingersoll–Ross model. Additionally, the branches cardinality is always equal or bigger than the number of assets.

We adopt a deterministic liability model without external contributions, i.e. f_τ and F_τ are always zero. The other model's parameters are presented in Table 14. In all cases, the initial funding ratio is one. So, in the beginning, the fund is able to afford all the long-term liability.

We build the simulations by defining three classes of trees: small, medium and large. These classes are determined by their number of periods (stages), respectively, 4, 6 and 8. For

Table 14 – model's parameters

Parameter	Value
W	576,000
K	1
π	0.7
M	999,999
ζ	1

each tree category, we executed eight cases that are named by alphabet letters (A, B, ..., H). We describe the scenario tree topologies using their Ψ vector definition, see Section 3.2. The characteristics of each type of tree (topology, number of nodes, number of scenarios, number of variables and number of constraints) are show in the Table 15.

First, it is important to notice how big the trees we are solving are. We list, in Table 16, some previous works as benchmarks to the number of scenarios from event trees. Some of them do not include joint chance constraints, but they are also solved without any auxiliary method or heuristic.

Comparing the number of scenarios on our study to the current literature, we can notice that we are solving much bigger problems. For instance, Small_H, Medium_H and Large_H have, respectively, 162,000, 121,500 and 104,000 scenarios. Thus, they are, respectively, more than 16, 12 and 10 times bigger than the study of Domenica et al. (2009), see Table 16. Furthermore, our problem considers a joint chance constraint. Obviously, this a rough comparison, but it gives an idea of the problem size that our approach can handle.

The rest of this chapter is organized as follows. In Section 5.1, we report the computational time to simulate and optimize the scenario trees described in the Table 15. In Section 5.2, we discuss how these distinct scenario tree arrangements can impact on the problems' in-sample objective function stability.

5.1 Computational Time

The computational time to solve this stochastic optimization problem is the focus of this section. First, as there exists uncertainty in these instances, we optimize twenty samples for each case, resulting in 480 tree generations, 1,071,900 different scenarios taking about 18 hours and 30 minutes to be finished. We exhibit the computational times descriptive statistics (measured in seconds) in the Table 17. We separate the computational time (Table 17) in the tree generation and the model optimization, with the total time being the sum of both. We are able to solve quite large trees in a reasonable amount of time. For instance, trees with a smaller number of scenarios in each class, Small_A, Medium_A and Large_A, have, respectively, 27, 243 and 2,187 scenarios and can be solved, on average, in 2.02, 2.13 and 2.36 seconds. Furthermore, trees with 19,683 scenarios (Small_E, Medium_D e Large_C), which are almost twice the size, in terms of scenarios, compared to the biggest problem in the current literature without the use of specific algorithms (see Table 16), can be computed, on average, in-between 18.32 (Large_C) and 27.01 (Medium_D) seconds.

Additionally, trees with more scenarios in each class, Small_H, Medium_H and Large_H, present 162,000, 121,500 and 104,976 scenarios and are, respectively, solved in 821.79, 455.42 and 417.78 seconds. Considering multiple stages (between 4 and 6) and a large number of scenarios (between 104,976 and 162,000), for problems which online solutions are not required, computational times between 7 and 14 minutes are quite reasonable.

Table 15 – Characteristics of the generated scenario trees.

Name	Topology - Ψ	Nodes	Scenarios	Variables	Constraints
Small					
Small_A	1-3-3-3	40	27	381	263
Small_B	1-9-3-3	118	81	1,137	785
Small_C	1-27-3-3	352	243	3,405	2,351
Small_D	1-81-3-3	1,054	729	10,209	7,049
Small_E	1-81-81-3	26,326	19,683	256,611	177,635
Small_F	1-81-81-9	65,692	59,049	650,271	453,197
Small_G	1-243-18-18	83,350	78,732	828,876	578,828
Small_H	1-2000-9-9	182,001	162,000	1,800,003	1,254,002
Medium					
Medium_A	1-3-3-3-3-3	364	243	3,513	2,423
Medium_B	1-9-3-3-3-3	1,090	729	10,533	7,265
Medium_C	1-81-3-3-3-3	9,802	6,561	94,773	65,369
Medium_D	1-81-9-3-3-3	29,242	19,683	282,855	195,131
Medium_E	1-81-9-9-3-3	86,104	59,049	833,979	575,669
Medium_F	1-81-33-3-3-3	107,002	72,171	1,035,183	714,179
Medium_G	1-24-21-21-3-3	138,121	95,256	1,338,339	923,978
Medium_H	1-1500-3-3-3-3	181,501	121,500	1,755,003	1,210,502
Large					
Large_A	1-3-3-3-3-3-3-3	3,280	2,187	31,701	19,676
Large_B	1-9-3-3-3-3-3-3	9,838	6,561	95,097	65,585
Large_C	1-27-3-3-3-3-3-3	29,512	19,683	285,285	196,751
Large_D	1-21-9-3-3-3-3-3	68,818	45,927	665,283	458,831
Large_E	1-81-3-3-3-3-3-3	88,534	59,049	855,849	590,249
Large_F	1-27-9-3-3-3-3-3	88,480	59,049	855,363	589,925
Large_G	1-18-18-3-3-3-3-3	117,955	78,732	1,140,321	786,458
Large_H	1-72-6-3-3-3-3-3	157,321	104,976	1,520,859	1,048,898

Table 16 – Characteristics of scenario tress on previous literature.

Year	Article	Tree Topology	Number of Scenarios
1996	Kall e Mayer (1996)	-	7000
2007	Domenica et al. (2009)	3 stages	10000
2010	Andrieu, Henrion e Römisch (2010)	3 stages	1000
2010	Haneveld, Streutker e VAN DER VLERK (2010)	4 stages (1-10-10-10)	1000
2014	Ackooij et al. (2014)	24 stages	100
2016	Mello e Pagnoncelli (2016)	4 stages (1-10-10-10)	1000

Table 17 – Computational time in seconds for the scenarios trees.

Name	Tree Generation Time				Model Optimization Time				Total Time			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.
Small												
Small_A	0.02	0.01	0.02	0.00	2	0	2	2	2.02	0.01	2.02	2.03
Small_B	0.02	0.01	0.02	0.04	2.10	0.30	2	3	2.12	0.30	2.02	3.03
Small_C	0.03	0.00	0.03	0.04	2.1	0.30	2	3	2.13	0.31	2.03	3.04
Small_D	0.057	0.01	0.05	0.08	2.30	0.47	2	3	2.35	0.47	2.05	3.08
Small_E	13.91	0.92	12.92	16.14	11.2	1.54	8	15	25.11	2.10	20.92	29.43
Small_F	104.73	4.91	97.53	117.19	24	3.56	20	37	128.73	5.98	118.55	141.19
Small_G	213.25	10.22	194.93	233.52	32.05	3.20	27	40	245.30	11.97	223.93	266.52
Small_H	766.94	44.86	642.47	831.28	54.85	7.30	50	85	821.79	47.05	692.47	884.28
Medium												
Medium_A	0.03	0.01	0.03	0.04	2.10	0.30	2	3	2.13	0.30	2.03	3.04
Medium_B	0.05	0.01	0.05	0.09	2.10	0.30	2	3	2.15	0.30	2.05	3.06
Medium_C	1.51	0.03	1.46	1.61	5.05	0.22	5	6	6.56	0.23	6.46	7.56
Medium_D	13.96	0.83	13.06	15.67	13.05	1.57	11	17	27.01	2.18	24.28	32.42
Medium_E	92.94	0.40	92.03	93.66	4.27	21	37	61	124.846	4.30	113.96	130.66
Medium_F	234.51	16.23	212.20	282.66	78.5	9.44	59	103	313.01	21.35	285.18	364.35
Medium_G	258.26	8.03	249.76	282.10	50.75	11.84	71	35	309.01	12.07	285.76	329.74
Medium_H	393.92	3.82	387.73	400.76	61.50	5.89	53	72	455.42	8.21	441.73	470.08
Large												
Large_A	0.21	0.02	0.2	0.3	2.15	0.36	2	3	2.36	0.37	2.2	3.25
Large_B	1.54	0.03	1.62	1.5	6	0.64	5	8	7.54	0.65	6.52	9.54
Large_C	10.72	0.06	10.66	10.89	7.6	2.68	6	14	18.32	2.69	16.66	24.72
Large_D	54.02	0.64	53.17	55.77	19.25	3.50	18	34	73.27	3.78	71.17	88.8
Large_E	120.97	40.23	88.40	171.41	25.70	0.47	25	26	146.67	40.27	113.66	196.41
Large_F	88.13	0.58	87.24	89.69	26.6	4.82	25	47	114.73	4.66	112.53	134.24
Large_G	156.04	0.77	154.78	157.43	40.7	4.56	39	60	196.74	4.69	194.78	216.47
Large_H	320.28	19.16	300.49	396.04	97.5	93.66	63	479	417.78	94.30	363.49	790.86

We notice that the increase of nodes, scenarios, variables and constraints are exponential. On the one hand, Small_A presents 27 scenarios, 381 variables and 263 constraints. On the other hand, Small_H has 162,000 scenarios, 1,800,003 variables and 1,254,002 constraints. The growth of the number of scenarios for each class of tree is presented in the Figure 8.

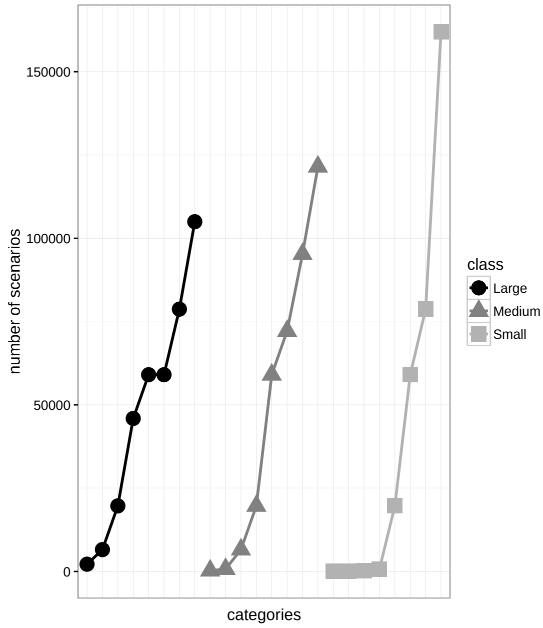


Figure 8 – Exponential growth of scenario number

The same growth rate is observed in the tree generation processing times, as section 3.4. For instance, the generation of 121,500 scenarios for Medium_H case and 104,976 for Large_H case take 393.92 and 320.28 seconds, respectively, while it takes 766.94 seconds to generate 162,000 scenarios for Small_H. The time of optimization process also increases as the number of scenarios gets larger, but the rate of increase is smaller than the one for the scenario generation. The mean of the time elapsed to generate the scenario trees is presented in the Figure 9 and the mean for optimization of the model is shown in the Figure 9.

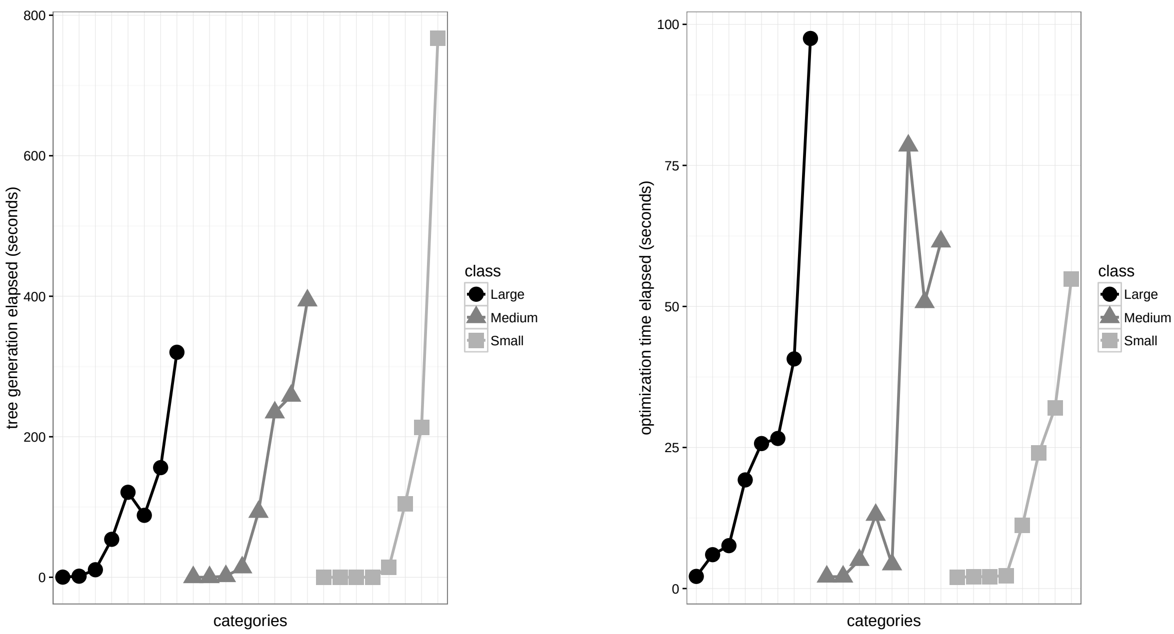
In summary, our approach can very efficiently handle large scale multistage stochastic optimization problems with the compact representations of AMLs and the scenario generation method.

5.2 In-sample stability

We take the opportunity of presenting several range of tree topologies in our tests, in order to explore some results related to the in-sample stability of the objective function. Considering each topology and the same instances generated for the tests of Section 5.1, Table 18 shows the mean, standard deviation, minimum and maximum values for the problem's objective function. Based on Table 18, the number of scenarios is inversely proportional to objective function standard deviation, which follows the stochastic average approximation theory (KLEYWEGT; SHAPIRO; MELLO, 2002). The standard deviation of Small_A (27 scenarios) is 22,438.91 while

Figure 9 – Application performance tests

(a) tree generation time elapsed (seconds) (b) Optimization time elapsed (seconds)



962.25 for Small_H (162,000 scenarios). This result repeats again for Medium_A and Medium_H and Large_A and Large_H.

Furthermore, we generate some trees with the same number of scenarios but with distinct topologies. It is the case of Small_E (1-81-81-3), Medium_D (1-81-9-3-3-3) and Large_C (1-27-3-3-3-3-3-3), all with 19, 683 scenarios, but with a different number of variables and constraints. It seems that the topology impacts directly on the in-sample stability. Based on Table 18, trees with the bigger number of scenarios, in the early stages of the tree, presented smaller standard deviation, indicating that they are closer to the 'true value'. This phenomenon repeats for the cases Small_F, Medium_E, Large_E and Large_F, all with 59, 049 scenarios. This is even clearer when comparing just Large_E and Large_F, in which the number of stages are the same.

Table 18 – In-sample stability - statistic data description

Name	Mean	Std. Dev.	Min.	Max.
Small				
Small_A	640,803.2	22,436.91	611,101	681,156
Small_B	642,605.7	12,691.43	624,357	670,358
Small_C	637,719.3	6,988.39	627,927	651,698
Small_D	637,426.6	6,830.81	628,080	650,761
Small_E	629,107.7	3,678.078	622,844	634,190
Small_F	627,890.1	5,240.45	620,890	638,982
Small_G	625,652.2	2,886.70	619,948	629,944
Small_H	627,856.0	962.25	626,456	629,643
Medium				
Medium_A	707,157.3	22,987.64	676,520	758,256
Medium_B	715,601.2	22,040.23	679,350	768,897
Medium_C	710,105.8	6,489.04	697,252	723,451
Medium_D	703,796.3	7,565.77	691,884	717,514
Medium_E	697,698.4	5,962.75	688,136	710,605
Medium_F	701,587.9	7,530.58	690,836	717,896
Medium_G	695,364.95	10,668.65	683,593	713,895
Medium_H	707,818.8	1,494.31	705,265	711,262
Large				
Large_A	836,973.5	39,365.70	771,480	907,561
Large_B	837,408.3	25,976.09	802,403	897,048
Large_C	841,157.3	17,553.62	817,831	879,715
Large_D	825,551.0	16,468.40	803,660	862,746
Large_E	828,417.5	9,336.38	811,772	842,299
Large_F	827,965.8	13,923.86	801,798	852,605
Large_G	825,294.7	19,665.84	797,876	869,533
Large_H	819,502.9	10,102.82	806,444	836,548

6 Conclusion

The size of multistage stochastic programs can grow very quickly. Consequently, a large scenario tree becomes hard to compute (GUPTA; GROSSMANN, 2011; GUIGUES; HENRION, 2017). We propose a framework which enables the solving of larger instances of multistage stochastic programs, without the support of any relaxation or decomposition on the initial program. The scenario generation is carried out by a compact memory representation based on left-child, right-sibling description, relying on Knuth transform encoded in C/C++. The optimization model is driven by the directives of an implicit extensive form, which allows a new formalization of nonanticipativity constraints that has direct correspondence with algebraic modeling languages. Therefore, with the use of these guidelines, we formalize a deterministic equivalent version of stochastic multistage programming models, which demands reasonable time to be completely solved for large scale problems. This methodology was tested in an ALM multistage stochastic program with joint chance constraint. It allowed us to simulate and optimize multistage stochastic model instances with more than 160,000 scenarios, about 200,000 elements, 2 million variables and 1 million constraints to optimality in a few minutes.

Our study presents some limitations which are also opportunities for future research. We represent the randomness in this framework with two stochastic processes (GBM and CIR), which could be replaced by other processes, like autoregressive conditional heteroscedasticity (ARCH) or generalized autoregressive conditional heteroscedasticity (GARCH). The probability distribution of events in the scenario tree could also be endogenous and determined by *a priori* hypotheses. The flexibility in the bundle definition also permits different branch strategies to be formulated. Furthermore, we notice that there are some issues that still need attention in the interface between the simulation and optimization. We compact the scenario tree in the memory by using a data structure based on the Knuth transform. In this data structure, only the first element from the linked list can have a child, but it would be possible to determine a structure so that each point could give origin to their own children , increasing the performance in the insertion and search operations. Our mechanism could also, probably, even be used to speed up some other methods, such as algorithms based on branch and bound or branch and cut.

.1 Appendix 1

Modeling bundles in AMPL: first, we have to define the set which will contain the tuple elements.

```
set scenario1:=1;
set scenario2:=1..2;
set scenario3:=1..4;
```

After that, we define the connections among the bundles.

```
set links1 within {scenario1,scenario2};
set links2 within {scenario2,scenario3};
```

Therefore, with the sets links1 and links2, we are able to explicitly define which links exists. It is made in the data file, as illustrated below.

```
set links1:=
(1,1)(1,2)
;

set links2:=
(1,1)(1,2)
(2,3)(2,4)
```

As we can define the links freely, the tree model has flexibility in the definition of its topology. It allows the formalization of different degrees in each level of tree. In the moment of constraints' formalization, it is possible use these tuples as parameters for generic definition. We also can iterate among them, such as in the following example.

```
set scenario1:=1;
set scenario2:=1..2;
set scenario3:=1..4;

set ativos:=1..3;

set links1 within {scenario1,scenario2};
set links2 within {scenario2,scenario3};

var qtd0{ativos} >= 0;

var qtd1{ativos, scenario1} >= 0;
var comp1{ativos, scenario1} >= 0;
var vend1{ativos, scenario1} >= 0;

var qtd2{ativos, scenario2} >= 0;
var comp2{ativos, scenario2} >= 0;
var vend2{ativos, scenario2} >= 0;

subject to
##Intertemporal dependence of variables.

balanco1{s in scenario1, a in ativos}:
qtd1[a,s] = qtd0[a] + comp1[a,s] - vend1[a,s];

balanco2{(i,j) in links1, a in ativos}:
qtd2[a,j] = qtd1[a,i] + comp2[a,j] - vend2[a,j];
```

.2 Appendix 2

The prices from the simulation are written in a file as AMPL standard. Therefore, before the beginning of the optimization process, the prices of assets in every time period have already been defined. An example of data file is presented below.

```
data;

set links1:=
(1,1)(1,2)
;

set links2:=
(1,1)(1,2)
(2,3)(2,4)

##Initial prices: these prices are deterministic.
param price0:=
1 10.00
2 10.00
3 10.00
;

param price1:=
1 1 11.23993
2 1 8.33407
3 1 13.38071
1 2 11.45946
2 2 12.39967
3 2 11.20237
;

param price2:=
1 1 12.79296
2 1 8.32131
3 1 11.51957
1 2 12.72361
2 2 10.36140
3 2 11.26605
1 3 13.10480
2 3 8.03804
3 3 13.18248
1 4 12.96897
2 4 16.80583
3 4 12.24611
;
```

.3 Appendix 3

The scenario tree is mapped in the computer memory through an approach based on Knuth transform (KNUTH, 1998). Thus, every multi-way tree can be redefined more compactly as a left-child-right-sibling binary tree. The reformulation process is detailed in the Section 3.4. Below, we present the algorithm that allocates this data structure in the principal memory.

Algorithm 3: Request and allocate the multi-way scenario tree data structure described as left-child, right-sibling representation in the memory space

```
Build Tree (S, T);
Input  : T: number of time periods, S: numerical vector determined by the number of
         scenarios for each time period
Output : root: tree data structure
if root is empty then
    root ← allocateMemory
    root.price ← List of Prices
end
point ← root
for t ← 1 to T do
    point.nextNivel ← allocateMemory
    point 2 ← point.nextNivel
    point 2.price ← List of Prices
    for s ← S_{t−1} to S_t do
        point 2.sibling.price ← List of Prices
        point 2 ← point 2.sibling
    end
    point ← point.nextNivel
end
return root;
```

Final Remarks

In this dissertation, we present three articles which approach multistage stochastic programming using the asset liability management (ALM) problem as the practical application. In the first article, we define a multistage stochastic ALM model with chance constraints for a Brazilian pension fund industry. In the second article, we compare different ALM scenario generation sampling models by measuring the impact of several scenario tree sampling methodologies in the final solution and portfolio allocation from an in-sample perspective. In the last article, we propose a framework for scenario tree generation and the optimization of multistage stochastic programming problems. We propose the formalization and implementation of nonantecipativity implicit version of simulation and optimization model allowing to generate and optimize a large number of scenarios.

In the first chapter, our study proposes a multistage stochastic programming ALM model with chance and combinatorial constraints which are motivated and can be applied by the Brazilian pension fund industry. The chance constraint enforces a Value-at-Risk (VaR) requirement to keep the pension fund solvent across time with a high probability. The combinatorial constraint represents an intertemporal solvency regulation imposed by the Brazilian pension fund legislation. We construct multiple binary trees, with each giving the same importance to catastrophic and normal economic scenarios. The results show that Brazilian pension fund managers should modify their investment behavior and strategies in the near future, as they will be pressured to increase their positions in riskier assets if the long-term downward trend of interest rates gets confirmed. As funds managers become less risk-averse, their fund's insolvency probability will increase. However, if pension fund managers decide to keep their current risk profile (in terms of risk allocation and insolvency probability), pension fund members' external contributions would have to be raised in the next few years. As we deal with a strategic allocation, there is an opportunity for implementation of tactical asset-liability management regarding the same contextual constraints.

In the second chapter, we tackle scenario generation methodology. The main goal of this study is to compare the performance of different scenario sampling methods in order to highlight which of them is more appropriate for designing a representative discrete-space model for asset-liability management (ALM) problems regarding the in-sample performance. This study compares the empirical results of distinct approaches to generating scenarios for ALM: Random sampling and Moment matching. We also test two Monte Carlo sampling variations: Resampled average approximation and Monte Carlo, using a naive allocation strategy as a benchmark. Our intent is to outline how, empirically, the method may have an impact and produce different outputs. Based mostly on the resulting values of the objective function, we can conclude that the classical Monte Carlo sampling and Monte Carlo with naive allocation strategy are dominated by the Moment matching and the Resampled average approximation. In spite of considering several sampling methods, there is room for an extension of similar comparisons with another sampling method, such as minimizing Wasserstein probability metrics

(ROMISCH, 2003; HEITSCH; ROMISCH, 2005; HOCHREITER; PFLUG, 2007) or Voronoi cell sampling (LOHNDORF, 2016). Other opportunities for subsequent works would be scenario reduction and parallel implementation, which are techniques developed to deal with the curse of dimensionality in stochastic programming problems (BERALDI; SIMONE; VIOLI, 2010; DUPAČOVÁ; GRÖWE-KUSKA; RÖMISCH, 2003; HEITSCH; RÖMISCH, 2003).

The last chapter addresses the simulation and optimization performance for multistage stochastic problems, as well as the gap of compatibility between scenario generators and stochastic programs from the implicit extensive form perspective. Our work implements the modeling implicit extensive form methodology on both simulation and optimization, defining an interface between them. Moreover, our technique is generic enough to be applied to other SP models, for instance, the ones with recourse, chance constraints, or even integrated chance constraints. The scenario generation allocates a multi-way tree represented by a reformulated tree with the left-child-right-sibling strategy. We also employ the information from the filtration process to compose the bundles, which also correspond with nonanticipativity constraints implicitly in the optimization process. The bundle concept is more algebraically adequate, with the AMLs avoiding support of the extensions to be done. These guidelines allow us to settle an innovative framework, which formalizes shorter stochastic programs. This methodology was able to provide the optimal solutions for an ALM multistage stochastic program, with a joint chance constraint and with more than 160,000 scenarios without any relaxation or decomposition approach. Our study presents some limitations which are also opportunities for future research. We represent the randomness in this framework with two stochastic processes (GBM and CIR), which could be replaced by other processes, like autoregressive conditional heteroscedasticity (ARCH) or generalized autoregressive conditional heteroscedasticity (GARCH). Besides that, the probability distribution of events in the scenario tree could also be endogenous and determined by *a priori* hypotheses. We also could use the flexibility in the bundle definition to formulate different branch strategies. Furthermore, we notice that there are some issues that still need attention in the interface between the simulation and optimization. We compact the scenario tree in the memory by using a data structure based on the Knuth transform. In this data structure, only the first element from the linked list can have a child, but it would be possible to determine a structure so that each point could give origin to their own children, increasing the performance of the insertion and search operations.

Finally, we add the Section Annex in order to highlight some other contributions given by us. They consist of two published papers, which form this dissertation, and one more paper, which is under review. Furthermore, the other two papers were accepted in conferences. Currently, we are involved in two working papers. In terms of academic experience, we could be part of an exchange visiting program in a top university, presenting our papers at two conferences.

Bibliography

ACKOOIJ, W.; HENRION, R.; MÖLLER, A.; ZORGATI, R. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, Springer US, v. 15, n. 2, p. 509–531, 2014. Cited on page 75.

ADAM, A. *Handbook of asset and liability management: From models to optimal return strategies*. 1. ed. London: John Wiley and Son, 2007. Cited on page 14.

AHMED, S.; TAWARMALANI, M.; SAHINIDIS, N. V. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, Springer, v. 100, n. 2, p. 355–377, 2004. Cited on page 55.

ALDASORO, U.; ESCUDERO, L.; MERINO, M.; PÉREZ, G. A parallel branch-and-fix coordination based matheuristic algorithm for solving large sized multistage stochastic mixed 0–1 problems. *European Journal of Operational Research*, Elsevier, v. 258, n. 2, p. 590–606, 2017. Cited on page 55.

ANDRIEU, L.; HENRION, R.; RÖMISCH, W. A model for dynamic chance constraints in hydro power reservoir management. *European journal of operational research*, Elsevier, v. 207, n. 2, p. 579–589, 2010. Cited 2 times on pages 69 and 75.

ASANGA, S.; ASIMIT, A.; BADESCU, A.; HABERMAN, S. Portfolio optimization under solvency constraints: A dynamical approach. *North American Actuarial Journal*, v. 18, n. 3, p. 394–416, 2014. Cited on page 14.

ASIMIT, A.; BADESCU, A.; SIU, T. K.; ZINCHENKO, Y. Capital requirements and optimal investment with solvency probability constraints. *IMA Journal of Management Mathematics*, p. 1–31, 2014. Cited on page 14.

BECKER, F.; GURTLER, M.; HIBBELN, M. Markowitz versus Michaud: portfolio optimization strategies reconsidered. *The European Journal of Finance*, v. 21, n. 4, p. 269–291, 2015. Cited on page 25.

BENARTZI, S.; THALER, R. Naive diversification strategies in defined contribution saving plans. *The American Economic Review*, American Economic Association, v. 91, n. 1, p. 79–98, 2001. Cited on page 44.

BERALDI, P.; BRUNI, M. E. A clustering approach for scenario tree reduction: an application to a stochastic programming portfolio optimization problem. *TOP*, Springer, v. 22, n. 3, p. 934–949, 2014. Cited on page 51.

BERALDI, P.; SIMONE, F. de; VIOLI, A. Generating scenario trees: a parallel integrated simulation–optimization approach. *Journal of Computational and Applied Mathematics*, Elsevier, v. 233, n. 9, p. 2322–2331, 2010. Cited 3 times on pages 43, 51, and 85.

BERGER, A.; MULVEY, J.; ROTHBERG, E.; VANDERBEI, R. Solving multistage stochastic programs using tree dissection. *Statistics and Operations Research Research Report, Princeton University, Princeton, NJ*, 1995. Cited on page 55.

BERKELAAR, A.; GROMICHO, J.; KOUWENBERG, R.; ZHANG, S. A primal-dual decomposition algorithm for multistage stochastic convex programming. *Mathematical Programming*, Springer, v. 104, n. 1, p. 153–177, 2005. Cited on page 55.

BERKELAAR, A.; KOUWENBERG, R. Retirement saving with contribution payments and labor income as a benchmark for investments. *Journal of Economic Dynamics and Control*, v. 27, p. 1069 – 1097, 2003. Cited 2 times on pages 28 and 31.

BIRGE, J. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, INFORMS, v. 33, n. 5, p. 989–1007, 1985. Cited on page 55.

BIRGE, J.; HOLMES, D. Efficient solution of two-stage stochastic linear programs using interior point methods. *Computational Optimization and Applications*, Springer, v. 1, n. 3, p. 245–276, 1992. Cited on page 55.

BIRGE, J. R.; LOUVEAUX, F. *Introduction to Stochastic Programming*. 2. ed. New York: Springer Science+Business Media, 2011. Cited 3 times on pages 8, 35, and 61.

BISSCHOP, J.; ENTRIKEN, R. *AIMMS: the modeling system*. [S.l.]: Paragon Decision Technology, 1993. Cited on page 55.

BISSCHOP, J.; FOURER, R. New constructs for the description of combinatorial optimization problems in algebraic modeling languages. *Computational Optimization and Applications*, Springer, v. 6, n. 1, p. 83–116, 1996. Cited on page 57.

BOENDER, G. C. E. A hybrid simulation/optimisation scenario model for asset/liability management. *European Journal of Operational Research*, v. 99, n. 1, p. 126 – 135, 1997. ISSN 0377-2217. Cited 2 times on pages 14 and 37.

BOGENTOFT, E.; ROMEIJN, H. E.; URYASEV, S. Asset/liability management for pension funds using cvar constraints. *The Journal of Risk Finance*, v. 3, p. 57–71, 2001. Cited 2 times on pages 14 and 37.

BRADLEY, S. P.; CRANE, D. B. A dynamic model for bond portfolio management. *Management Science*, v. 19, p. 139–151, 1972. Cited 2 times on pages 14 and 37.

Brazilian Association of Closed Supplementary Pension Funds. *Consolidade Estatístico*. 2014. Accessed in 2015-01-05. Disponível em: <http://www.abrapp.org.br/Consolidados/Consolidado%20Estat%C3%ADstico_12_2014.pdf>. Cited 3 times on pages 17, 28, and 31.

Brazilian Central Bank. *Resolution number 3792*. 2012. Accessed in 2015-01-05. Disponível em: <http://www.bcb.gov.br/pre/normativos/res/2009/pdf/res_3792_v1_O.pdf>. Cited on page 17.

BROOK, A.; KENDRICK, D.; MEERAUS, A. GAMS, a user's guide. *ACM Signum Newsletter*, ACM, v. 23, n. 3-4, p. 10–11, 1988. Cited on page 55.

BUCHANAN, C.; MCKINNON, K.; SKONDRAS, G. The recursive definition of stochastic linear programming problems within an algebraic modeling language. *Annals of Operations Research*, Springer, v. 104, n. 1, p. 15–32, 2001. Cited on page 55.

CALFA, B. *A memory-Efficient implementation of multi-period two-and multi-stage stochastic programming Models*. [S.l.]: Carnegie Mellon University. Technical Report, 2014. Cited 2 times on pages 56 and 57.

CARIÑO, D. R.; KENT, T.; MEYERS, D. H.; STACY, C.; SYLVANUS, M.; TURNER, A. L.; WATANABE, K.; ZIEMBA, W. T. The Russel-Yasuda Kasai model: An asset/liability model for japanese insurance company using multistage stochastic programming. *Interfaces*, v. 24, n. 1, p. 29–49, 1994. Cited 4 times on pages 14, 22, 37, and 68.

CARINO, D. R.; ZIEMBA, W. T. Formulation of the russell-yasuda kasai financial planning model. *Operations Research*, INFORMS, v. 46, n. 4, p. 433–449, 1998. Cited on page 68.

CARØE, C. C.; SCHULTZ, R. Dual decomposition in stochastic integer programming. *Operations Research Letters*, Elsevier, v. 24, n. 1, p. 37–45, 1999. Cited 2 times on pages 40 and 55.

COHEN, K.; HAMMER, F. Linear programming and optimal bank asset management decisions. *The Journal of Finance*, [American Finance Association, Wiley], v. 22, n. 2, p. 147–165, 1967. Cited on page 44.

COLOMBO, M.; GROTHEY, A.; HOGG, J.; WOODSEND, K.; GONDZIO, J. A structure-conveying modelling language for mathematical and stochastic programming. *Mathematical Programming Computation*, Springer, v. 1, n. 4, p. 223–247, 2009. Cited 2 times on pages 55 and 57.

CONSIGLI, G.; DEMPSTER, M. The calm stochastic programming model for dynamic asset-liability management. *Worldwide asset and liability modeling*, Cambridge University Press Cambridge, v. 10, p. 464, 1998. Cited on page 68.

CONSIGLI, G.; DEMPSTER, M. A. H. Dynamic stochastic programming for asset - liability management. *Annals of Operations Research*, v. 81, p. 131–161, 1998. Cited 5 times on pages 10, 14, 37, 55, and 68.

CONSIGLIO, A.; CAROLLO, A.; ZENIOS, S. A. A parsimonious model for generating arbitrage-free scenario trees. *Quantitative Finance*, Taylor & Francis, v. 16, n. 2, p. 201–212, 2016. Cited on page 23.

CONSIGLIO, A.; COCCO, F.; ZENIOS, S. et al. Asset and liability modeling for participating policies with guarantees. *Journal of Risk and Insurance*, 2001. Cited on page 68.

CONSIGLIO, A.; COCCO, F.; ZENIOS, S. A. Scenario optimization asset and liability modelling for individual investors. *Annals of Operations Research*, v. 152, n. 1, p. 167 – 191, 2007. Cited 2 times on pages 14 and 68.

CONSIGLIO, A.; SAUNDERS, D.; ZENIOS, S. A. Asset and liability management for insurance products with minimum guarantees: The UK case. *Journal of Banking and Finance*, v. 30, n. 2, p. 645 – 667, 2006. ISSN 0378-4266. Cited 2 times on pages 14 and 68.

CONSIGLIO, A.; STAINO, A. A stochastic programming model for the optimal issuance of government bonds. *Annals of Operations Research*, v. 193, n. 1, p. 159–172, 2012. ISSN 0254-5330. Cited 2 times on pages 14 and 68.

CONSIGLIO, A.; TUMMIELLO, M.; ZENIOS, S. Designing guarantee options in defined contribution pension plans. *Insurance: Mathematics and Economics*, forthcoming, 2015. Cited on page 32.

CORMEN, T. *Introduction to Algorithms*. [S.l.]: MIT press, 2009. Cited on page 66.

COX, J. C.; INGERSOLL, J. E.; ROSS, S. A. A theory of the term structure of interest rates. *Econometrica*, v. 53, n. 2, p. 385–408, 1985. Cited 4 times on pages 15, 22, 45, and 71.

DANTZIG, G.; INFANGER, G. Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, Springer, v. 45, n. 1, p. 59–76, 1993. Cited on page 55.

DATE, P.; CANEPA, A.; ABDEL-JAWAD, M. A mixed integer linear programming model for optimal sovereign debt issuance. *European Journal of Operational Research*, v. 214, n. 3, p. 749 – 758, 2011. ISSN 0377-2217. Cited on page 14.

DEMIGUEL, V.; GARLAPPI, L.; UPPAL, R. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies*, Soc Financial Studies, v. 22, n. 5, p. 1915–1953, 2009. Cited on page 44.

DEMPSTER, M.; MEDOVA, E.; YONG, Y. Comparison of sampling methods for dynamic stochastic programming. In: *Stochastic Optimization Methods in Finance and Energy*. New York: Springer Science+Business Media, 2011. cap. 16, p. 389–425. Cited 4 times on pages 41, 42, 48, and 50.

DEMPSTER, M. A. H.; GERMANO, M.; MEDOVA, E. A. Global asset liability management. *British Actuarial Journal*, v. 9, p. 137–216, 2003. Cited on page 22.

DOMENICA, N.; LUCAS, C.; MITRA, G.; VALENTE, P. Scenario generation for stochastic programming and simulation: a modelling perspective. *IMA Journal of Management Mathematics*, IMA, v. 20, n. 1, p. 1–38, 2009. Cited 4 times on pages 57, 66, 72, and 75.

DUFFIE, D. *Dynamic asset pricing theory*. Princeton: Princeton University Press, 2001. Cited 3 times on pages 22, 45, and 71.

DUPAČOVÁ, J.; CONSIGLI, G.; ; WALLACE, S. Scenarios for multistage stochastic programs. *Annals of Operations Research*, v. 100, n. 1, p. 25–53, 2000. Cited 5 times on pages 8, 35, 42, 66, and 68.

DUPAČOVÁ, J.; GRÖWE-KUSKA, N.; RÖMISCH, W. Scenario reduction in stochastic programming. *Mathematical programming*, Springer, v. 95, n. 3, p. 493–511, 2003. Cited 2 times on pages 51 and 85.

DUPAČOVÁ, J.; POLÍVKA, J. Asset-liability management for Czech pension funds using stochastic programming. *Annals Operational Research*, v. 165, p. 5–28, 2009. Cited 4 times on pages 15, 22, 29, and 68.

EGGING, R. Benders decomposition for multi-stage stochastic mixed complementarity problems–applied to a global natural gas market model. *European Journal of Operational Research*, Elsevier, v. 226, n. 2, p. 341–353, 2013. Cited on page 55.

ENTRIKEN, R. Language constructs for modeling stochastic linear programs. *Annals of Operations Research*, Springer, v. 104, n. 1, p. 49–66, 2001. Cited 2 times on pages 55 and 57.

ESCUDERO, L. F.; KAMESAM, P. V. On solving stochastic production planning problems via scenario modelling. *TOP*, Springer, v. 3, n. 1, p. 69–95, 1995. Cited on page 35.

FERSTL, R.; WEISSENSTEINER, A. Asset-liability management under time-varying investment opportunities. *Journal of Banking and Finance*, v. 35, n. 1, p. 47–62, 2011. Cited 3 times on pages 14, 37, and 68.

FIGUEIREDO, D. *Investiment Decision Making for Defined Benefit Pension Funds: an Multistage Stochastic Linear Programing Approach (in Portuguese)*. Dissertação (Mestrado) — School of Engineering - São Paulo State University, Brazil, 2011. Cited on page 24.

FLETCHER, J.; HILLIER, J. An examination of resampled portfolio efficiency. *Financial Analysts Journal*, v. 57, n. 5, p. 66–74, 2001. Cited on page 25.

FLETEN, S.-E.; HØYLAND, K.; WALLACE, S. W. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, v. 140, n. 1, p. 37–49, 2002. Cited on page 44.

FOURER, R. Modeling languages versus matrix generators for linear programming. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 9, n. 2, p. 143–183, 1983. Cited on page 55.

FOURER, R.; GAY, D. Expressing special structures in an algebraic modeling language for mathematical programming. *ORSA Journal on computing*, INFORMS, v. 7, n. 2, p. 166–190, 1995. Cited on page 55.

FOURER, R.; GAY, D. M.; KERNIGHAN, B. W. A modeling language for mathematical programming. *Management Science*, INFORMS, v. 36, n. 5, p. 519–554, 1990. Cited on page 55.

FOURER, R.; LOPES, L. A management system for decompositions in stochastic programming. *Annals of Operations Research*, Springer, v. 142, n. 1, p. 99–118, 2006. Cited 2 times on pages 55 and 56.

FOURER, R.; LOPES, L. Stampl: A filtration-oriented modeling tool for multistage stochastic recourse problems. *INFORMS Journal on Computing*, INFORMS, v. 21, n. 2, p. 242–256, 2009. Cited 3 times on pages 57, 61, and 63.

FRAHM, G. A theoretical foundation of portfolio resampling. *Theory and Decision*, v. 79, n. 1, p. 107–132, 2015. Cited on page 25.

FRANGOS, C.; ZENIOS, S.; YAVIN, Y. Computation of feasible portfolio control strategies for an insurance company using a discrete time asset/liability model. *Mathematical and Computer Modelling*, v. 40, n. 3–4, p. 423 – 446, 2004. Cited on page 14.

FREDMAN, M.; SEDGEWICK, R.; SLEATOR, D.; TARJAN, R. The pairing heap: A new form of self-adjusting heap. *Algorithmica*, Springer, v. 1, n. 1, p. 111–129, 1986. Cited on page 66.

GASSMANN, H. Mslip: a computer code for the multistage stochastic linear programming problem. *Mathematical Programming*, Springer, v. 47, n. 1, p. 407–423, 1990. Cited on page 55.

GASSMANN, H.; GAYL, D. An integrated modeling environment for stochastic. *Applications of Stochastic Programming*, SIAM, v. 5, p. 159, 2005. Cited on page 56.

GASSMANN, H.; IRELAND, A. Scenario formulation in an algebraic modelling language. *Annals of Operations Research*, Springer, v. 59, n. 1, p. 45–75, 1995. Cited 2 times on pages 55 and 56.

GASSMANN, H. I.; IRELAND, A. On the formulation of stochastic linear programs using algebraic modelling languages. *Annals of Operations Research*, Springer, v. 64, n. 1, p. 83–112, 1996. Cited 2 times on pages 55 and 56.

GEYER, A.; HANKE, M.; WEISSENSTEINER, A. No-arbitrage conditions, scenario trees, and multi-asset financial optimization. *European Journal of Operational Research*, Elsevier, v. 206, n. 3, p. 609–613, 2010. Cited on page 51.

GEYER, A.; ZIEMBA, W. T. The innovest austrian pension fund financial planning model innoalm. *Operations Research*, v. 56, n. 4, p. 797–810, 2008. Cited on page 68.

GLASSERMAN, P. *Monte Carlo Methods in Financial Engineering*. 1. ed. New York: Springer-Verlag New York, 2003. Cited 3 times on pages 36, 41, and 45.

GOEL, V.; GROSSMANN, I. A class of stochastic programs with decision dependent uncertainty. *Mathematical Programming*, Springer, v. 108, n. 2, p. 355–394, 2006. Cited on page 55.

GREENBERG, H. Modler: Modeling by object-driven linear elemental relations. *Annals of Operations Research*, Springer, v. 38, n. 1, p. 239–280, 1992. Cited on page 55.

GUIGUES, V.; HENRION, R. Joint dynamic probabilistic constraints with projected linear decision rules. *Optimization Methods and Software*, Taylor & Francis, v. 32, n. 5, p. 1006–1032, 2017. Cited 2 times on pages 69 and 80.

GÜLPINAR, N.; PACHAMANOVA, D. A robust optimization approach to asset-liability management under time-varying investment opportunities. *Journal of Banking and Finance*, v. 37, n. 6, p. 2031 – 2041, 2013. ISSN 0378-4266. Cited on page 14.

GÜLPINAR, N.; RUSTEM, B.; SETTERGREN, R. Simulation and optimization approaches to scenario tree generation. *Journal of economic dynamics and control*, Elsevier, v. 28, n. 7, p. 1291–1315, 2004. Cited 2 times on pages 43 and 51.

GÜLPINAR, N.; RUSTEM, B.; SETTERGREN, R. Simulation and optimization approaches to scenario tree generation. *Journal of Economic Dynamics and Control*, Elsevier, v. 28, n. 7, p. 1291–1315, 2004. Cited 2 times on pages 56 and 66.

GUPTA, V.; GROSSMANN, I. Solution strategies for multistage stochastic programming with endogenous uncertainties. *Computers & Chemical Engineering*, Elsevier, v. 35, n. 11, p. 2235–2247, 2011. Cited 2 times on pages 56 and 80.

HANEVELD, W. K. K.; STREUTKER, M. H.; VAN DER VLERK, M. H. An ALM model for pension funds using integrated chance constraints. *Annals of Operations Research*, v. 177, n. 1, p. 47–62, 2010. ISSN 0254-5330. Cited 5 times on pages 21, 37, 59, 69, and 75.

HARRISON, J. M.; SHARPE, W. F. Optimal funding and asset allocation rules for defined-benefit pension plans. In: *Financial Aspects of the United States Pension System*. [S.l.]: University of Chicago Press, 1983. p. 91–106. Cited on page 44.

HEITSCH, H.; RÖMISCH, W. Scenario reduction algorithms in stochastic programming. *Computational optimization and applications*, Springer, v. 24, n. 2-3, p. 187–206, 2003. Cited 2 times on pages 51 and 85.

HEITSCH, H.; ROMISCH, W. Generation of multivariate scenario trees to model stochasticity in power management. In: *IEEE St. Petersburg Power Tech*. [S.l.: s.n.], 2005. Cited 3 times on pages 8, 35, and 85.

HEITSCH, H.; RÖMISCH, W. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, Elsevier, v. 35, n. 6, p. 731–738, 2007. Cited on page 51.

HENRION, R.; KÜCHLER, C.; RÖMISCH, W. Scenario reduction in stochastic programming with respect to discrepancy distances. *Computational Optimization and Applications*, Springer, v. 43, n. 1, p. 67–93, 2009. Cited on page 51.

HIGLE, J.; RAYCO, B.; SEN, S. Stochastic scenario decomposition for multistage stochastic programs. *IMA Journal of Management Mathematics*, Oxford University Press, v. 21, n. 1, p. 39–66, 2009. Cited on page 55.

HILLI, P.; KOIVU, M.; PENNANEN, T.; RANNE, A. A stochastic programming model for asset liability management of a finnish pension company. *Annals of Operations Research*, Kluwer Academic Publishers-Plenum Publishers, v. 152, n. 1, p. 115–139, 2007. ISSN 0254-5330. Cited on page 68.

HOCHREITER, R.; PFLUG, G. Financial scenario generation for stochastic multi-stage decision processes as facility location problems. *Annals of Operations Research*, v. 152, n. 1, p. 257–272, 2007. Cited 4 times on pages 8, 35, 37, and 85.

Homem-de-Mello, T.; BAYRAKSAN, G. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, v. 19, n. 1, p. 56–85, 2014. Cited 2 times on pages 41 and 43.

Homem-de-Mello, T.; BAYRAKSAN, G. Stochastic constraints and variance reduction techiques. In: FU, M. (Ed.). *Handbook of Simulation Optimization*. [S.l.]: International Series in Operations Research and Management Science - Springer, 2014. v. 216, p. 245–276. Cited on page 36.

HØYLAND, K.; KAUT, M.; WALLACE, S. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, v. 24, n. 2, p. 169–185, 2003. Cited 5 times on pages 8, 35, 42, 43, and 71.

HØYLAND, K.; WALLACE, S. Generating scenario trees for multistage decision problems. *Management Science*, INFORMS, v. 47, n. 2, p. 295–307, 2001. Cited 7 times on pages 8, 23, 35, 37, 42, 43, and 71.

JOSA-FOMBELLIDA, R.; RINCÓN-ZAPATERO, J. P. Stochastic pension funding when the benefit and the risky asset follow jump diffusion processes. *European Journal of Operational Research*, v. 220, n. 2, p. 404 – 413, 2012. ISSN 0377-2217. Cited 2 times on pages 14 and 37.

KALL, P.; MAYER, J. SLP-IOR: an interactive model management system for stochastic linear programs. *Mathematical Programming*, Springer, v. 75, n. 2, p. 221–240, 1996. Cited 3 times on pages 55, 57, and 75.

KALLRATH, J. *Modeling Languages in Mathematical Optimization*. [S.l.]: Springer Science & Business Media, 2013. Cited on page 55.

KARABUK, S. Extending algebraic modelling languages to support algorithm development for solving stochastic programming models. *IMA Journal of Management Mathematics*, IMA, v. 19, n. 4, p. 325–345, 2008. Cited 2 times on pages 55 and 57.

KAUT, M.; WALLACE, S. W. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, v. 3, n. 2, p. 257–271, 2007. Cited 4 times on pages 36, 42, 48, and 51.

KAUT, M.; WALLACE, S. W. Shape-based scenario generation using copulas. *Computational Management Science*, Springer, v. 8, n. 1, p. 181–199, 2011. Cited on page 36.

KILIANOVÁ, S.; PFLUG, G. C. Optimal pension fund management under multi-period risk minimization. *Annals of Operations Research*, v. 166, n. 1, p. 261–270, 2009. ISSN 0254-5330. Cited 3 times on pages 14, 15, and 37.

KIM, S.; PASUPATHY, R.; HENDERSON, S. A guide to sample average approximation. In: FU, M. (Ed.). *Handbook of Simulation Optimization*. [S.l.]: International Series in Operations Research and Management Science - Springer, 2014. v. 216, p. 207 – 243. Cited on page 25.

KIM, T. S.; OMBERG, E. Dynamic nonmyopic portfolio behavior. *The Review of Financial Studies*, v. 9, p. 141–161, 1996. Cited on page 15.

KING, A. J. *SP/OSL: Version 1.0: Stochastic Programming Interface Library: User's Guide*. [S.l.]: IBM Thomas J. Watson Research Division, 1994. Cited on page 63.

KING, A. J.; WALLACE, S. W. *Modeling with stochastic programming*. [S.l.]: Springer Science & Business Media, 2012. Cited on page 9.

KLAASSEN, P. Financial asset-pricing theory and stochastic programming models for asset/liability management: A synthesis. *Management Science*, INFORMS, v. 44, n. 1, p. 31–48, 1998. Cited on page 68.

KLAASSEN, P. Comment on “generating scenario trees for multistage decision problems”. *Management Science*, INFORMS, v. 48, n. 11, p. 1512–1516, 2002. Cited on page 23.

KLEYWEGT, A.; SHAPIRO, A.; MELLO, T. H. de. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, SIAM, v. 12, n. 2, p. 479–502, 2002. Cited on page 77.

KNUTH, D. *The Art of Computer Programming: Sorting and Searching*. [S.l.]: Pearson Education, 1998. Cited 2 times on pages 66 and 82.

KOSMIDOU, K.; ZOPOUNIDIS, C. An optimization scenario methodology for bank asset liability management. *Operational Research*, Springer, v. 2, n. 2, p. 279–287, 2002. Cited on page 37.

KOUWENBERG, R. Scenario generation and stochastic programming models for asset liability management. *European Journal Operational Research*, v. 134, n. 1, p. 279–292, 2001. Cited 7 times on pages 14, 15, 24, 37, 41, 42, and 68.

KOUWENBERG, R.; ZENIOS, S. A. Stochastic programming models for asset liability management. In: *Handbook of asset and liability management*. [S.l.]: Elsevier, 2006. v. 1, cap. 6, p. 253–303. Cited on page 45.

KUIP, C. Algebraic languages for mathematical programming. *European Journal of Operational Research*, Elsevier, v. 67, n. 1, p. 25–51, 1993. Cited on page 55.

KUSY, M. I.; ZIEMBA, W. T. A bank asset and liability management model. *Operations Research*, v. 34, p. 356–376, 1986. Cited 2 times on pages 37 and 68.

LINDEROTH, J.; SHAPIRO, A.; WRIGHT, S. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, v. 142, n. 1, p. 215–241, 2006. ISSN 0254-5330. Cited on page 24.

LOHNDORF, N. An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research*, forthcoming, 2016. Cited 4 times on pages 8, 35, 42, and 85.

LULLI, G.; SEN, S. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Science*, INFORMS, v. 50, n. 6, p. 786–796, 2004. Cited on page 55.

LUSTIG, I.; MULVEY, J.; CARPENTER, T. Formulating two-stage stochastic programs for interior point methods. *Operations Research*, INFORMS, v. 39, n. 5, p. 757–770, 1991. Cited on page 55.

MAK, W.; MORTON, D. P.; WOOD, R. K. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, Elsevier, v. 24, n. 1, p. 47–56, 1999. Cited on page 36.

Markowitz, H. Portfolio selection. *Journal of Finance*, v. 7, p. 77–91, 1952. Cited on page 25.

MARKOWITZ, H.; USMEN, N. Resampled frontiers versus diffuse bayes: An experiment. *Journal of Investment Management*, v. 1, n. 4, p. 9–25, 2003. Cited on page 25.

MATOS, P.; PADILHA, G.; BENEGAS, M. On the management efficiency of brazilian stock mutual funds. *Operational Research*, Springer, p. 1–35, 2014. Cited on page 37.

MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, [Taylor and Francis, Ltd., American Statistical Association, American Society for Quality], v. 21, n. 2, p. 239–245, 1979. Cited 2 times on pages 8 and 35.

MEHROTRA, S.; PAPP, D. Generating moment matching scenarios using optimization techniques. *SIAM Journal on Optimization*, SIAM, v. 23, n. 2, p. 963–999, 2013. Cited on page 43.

MELLO, T. H. de; PAGNONCELLI, B. Risk aversion in multistage stochastic programming: a modeling and algorithmic perspective. *European Journal of Operational Research*, Elsevier, v. 249, n. 1, p. 188–199, 2016. Cited on page 75.

MERTON, R. C. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, v. 3, p. 373–413, 1973. Cited on page 15.

MERTON, R. C. *Continuous-time finance*. [S.l.]: Blackwell Publishers Ltda, 2001. Cited on page 15.

MESSINA, E.; MITRA, G. Modelling and analysis of multistage stochastic programming problems: a software environment. *European Journal of Operational Research*, Elsevier, v. 101, n. 2, p. 343–359, 1997. Cited 2 times on pages 55 and 57.

MICHAUD, R. *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. 1. ed. [S.l.]: Harvard Business School Press, 1998. Cited on page 24.

MICHAUD, R.; MICHAUD, R. *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. 2. ed. [S.l.]: Oxford University Press, 2008. Cited 5 times on pages 15, 24, 25, 28, and 43.

MILEVSKY, M. A. Optimal asset allocation towards the end of the life cycle: to annuitize or not to annuitize? *The Journal of Risk and Insurance*, v. 65, n. 3, p. 401–426, 1998. Cited on page 15.

Ministry of Social Welfare. *CGPC Resolution number 26*. 2008. Accessed in 2015-01-05. Disponível em: <http://www.previdencia.gov.br/arquivos/office/3_081029-134807-632.pdf>. Cited on page 17.

MITRA, G.; SCHWAIGER, K. (Ed.). *Asset and Liability Management Handbook*. 1. ed. [S.l.]: Palgrave Macmillan, 2011. Cited 2 times on pages 14 and 37.

MORAES, L. A. M.; FARIA, L. F. T. A stochastic programming approach to liquified natural gas planning. *Pesquisa Operacional*, SciELO Brasil, v. 36, n. 1, p. 151–165, 2016. Cited on page 37.

MUKUDDEM-PETERSEN, J.; PETERSEN, M. A. Optimizing asset and capital adequacy management in banking. *Journal of Optimization Theory and Applications*, v. 137, n. 1, p. 205–230, 2008. ISSN 0022-3239. Cited on page 14.

MULVEY, J.; RUSZCZYŃSKI, A. A new scenario decomposition method for large-scale stochastic optimization. *Operations research*, INFORMS, v. 43, n. 3, p. 477–490, 1995. Cited 2 times on pages 55 and 61.

MULVEY, J.; W.T.ZIEMBA. Asset and liability management systems for long-term investors: Discussion of the issues. In: *Worldwide Asset and Liability Modeling*. [S.l.]: Cambridge University Press, 1998. p. 3–40. Cited on page 37.

NEFTCI, S. N. *An introduction to the mathematics of financial derivatives*. 1. ed. United States: Academic Press, 1996. Cited 3 times on pages 22, 45, and 71.

NIELSEN, S. S.; POULSEN, R. A two-factor, stochastic programming model of danish mortgage–backed securities. *Journal of Economic Dynamics and Control*, v. 28, p. 1267–1289, 2004. Cited on page 14.

NOWAK, I. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. [S.l.]: Springer Science & Business Media, 2006. Cited on page 55.

OLIVEIRA, A. de; FILOMENA, T.; PERLIN, M.; LEJEUNE, M.; MACEDO, G. de. A multistage stochastic programming asset-liability management model: an application to the brazilian pension fund industry. *Optimization and Engineering*, Springer, v. 18, n. 2, p. 349–368, 2017. Cited 5 times on pages 8, 35, 37, 43, and 68.

PEDERSEN, A. M. B.; WEISSENSTEINER, A.; POULSEN, R. Financial planning for young households. *Annals of Operations Research*, v. 205, n. 1, p. 55–76, 2013. ISSN 0254-5330. Cited on page 14.

PFALTZ, J. Representing graphs by knuth trees. *Journal of the ACM (JACM)*, ACM, v. 22, n. 3, p. 361–366, 1975. Cited 2 times on pages 66 and 67.

PFLUG, G. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, v. 89, n. 2, p. 251–271, 2001. Cited 2 times on pages 35 and 68.

PFLUG, G. C. *Optimization of Stochastic models: The interface between simulation and optimization*. [S.l.]: Springer Science & Business Media, 2012. Cited 4 times on pages 9, 10, 46, and 56.

PRéKOPA, A. Moment problems. In: *Stochastic Programming*. Budapeste: Springer Netherlands, 1995. cap. 5, p. 125–178. Cited on page 42.

RASMUSSEN, K. M.; CLAUSEN, J. Mortgage loan portfolio optimization using multi-stage stochastic programming. *Journal of Economic Dynamics and Control*, v. 31, p. 742–766, 2007. Cited on page 14.

RIGHETTO, G. M.; MORABITO, R.; ALEM, D. A robust optimization approach for cash flow management in stationery companies. *Computers & Industrial Engineering*, Elsevier, v. 99, p. 137–152, 2016. Cited on page 37.

ROCKAFELLAR, R.; WETS, R. J.-B. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, INFORMS, v. 16, n. 1, p. 119–147, 1991. Cited 2 times on pages 55 and 63.

ROCKAFELLAR, R. T.; URYASEV, S. Optimization of conditional value-at-risk. *Journal of Risk*, v. 2, p. 21–41, 2000. Cited 2 times on pages 14 and 37.

ROMISCH, W. Stability of stochastic programming problems. In: RUSZCZYNSKI, A.; SHAPIRO, A. (Ed.). *Stochastic Programming, Volume 10 of Handbooks in Operations Research and Management Science*. Amsterdam: Elsevier, 2003. cap. 8, p. 483–554. Cited 3 times on pages 8, 35, and 85.

ROSA, C.; RUSZCZYŃSKI, A. On augmented lagrangian decomposition methods for multistage stochastic programs. *Annals of Operations Research*, Springer, v. 64, n. 1, p. 289–309, 1996. Cited on page 55.

RUBINSTEIN, R. Y.; KROESE, D. P. *Simulation and the Monte Carlo method*. [S.l.]: John Wiley & Sons, 2016. Cited on page 41.

RUSZCZYŃSKI, A. Decomposition methods in stochastic programming. *Mathematical Programming*, Springer, v. 79, n. 1, p. 333–353, 1997. Cited on page 55.

SCHERER, B. Portfolio resampling: Review and critique. *Financial Analysts Journal*, v. 58, n. 6, p. 98–109, 2002. Cited on page 25.

SCHULTZ, R. Stochastic programming with integer variables. *Mathematical Programming*, Springer, v. 97, n. 1, p. 285–309, 2003. Cited on page 55.

SHAPIRO, A.; DENTCHEVA, D.; RUSZCZYNSKI, A. *Lectures on Stochastic Programming*. 1. ed. [S.l.]: Society for Industrial and Applied Mathematics, 2009. Cited 4 times on pages 10, 35, 38, and 62.

SIEGMANN, A.; LUCAS, A. Discrete-time financial planning models under loss-averse preferences. *Operations Research*, v. 53, n. 3, p. 403 – 414, 2005. Cited 2 times on pages 28 and 31.

SINGH, K.; PHILPOTT, A.; WOOD, R. Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems. *Operations Research*, INFORMS, v. 57, n. 5, p. 1271–1286, 2009. Cited on page 55.

SLYKE, R.; WETS, R. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, SIAM, v. 17, n. 4, p. 638–663, 1969. Cited on page 55.

SMITH, J. E. Moment methods for decision analysis. *Management science*, INFORMS, v. 39, n. 3, p. 340–358, 1993. Cited on page 42.

SUN, J.; LIU, X. Scenario formulation of stochastic linear programs and the homogeneous self-dual interior-point method. *INFORMS Journal on Computing*, INFORMS, v. 18, n. 4, p. 444–454, 2006. Cited on page 55.

TAKRITI, S.; BIRGE, J. Lagrangian solution techniques and bounds for loosely coupled mixed-integer stochastic programs. *Operations Research*, INFORMS, v. 48, n. 1, p. 91–98, 2000. Cited on page 55.

UBEDA, J. R.; ALLAN, R. N. Stochastic simulation and monte carlo methods applied to the assessment of hydro-thermal generating system operation. *TOP*, Springer, v. 2, n. 1, p. 1–23, 1994. Cited on page 41.

ULF, H.; RAIMOND, M. Portfolio choice and estimation risk. a comparison of bayesian to heuristic approaches. *Astin Bulletin*, v. 36, n. 1, p. 135–160, 2006. Cited on page 25.

URYASEV, S.; THEILER, U. A.; SERRAINO, G. Risk-return optimization with different risk-aggregation strategies. *The Journal of Risk Finance*, v. 11, n. 2, p. 129–146, 2010. Cited 2 times on pages 14 and 68.

VALENTE, C. *Design and architecture of a stochastic programming modelling system*. Tese (Doutorado) — Brunel University, School of Information Systems, Computing and Mathematics, 2011. Cited 4 times on pages 57, 61, 62, and 63.

VALENTE, C.; MITRA, G.; SADKI, M.; FOURER, R. Extending algebraic modelling languages for stochastic programming. *INFORMS Journal on Computing*, INFORMS, v. 21, n. 1, p. 107–122, 2009. Cited 5 times on pages 55, 56, 57, 59, and 63.

VALENTE, P.; MITRA, G.; POOJARI, C. A stochastic programming integrated environment (spine). *Applications of Stochastic Programming. USA: Society for Industrial Mathematics*, p. 115–136, 2005. Cited on page 56.

VALLADÃO, D. M.; VEIGA, A. A multistage linear stochastic programming model for optimal corporate debt management. *European Journal of Operational Research*, v. 237, n. 1, p. 303 – 311, 2014. ISSN 0377-2217. Cited 2 times on pages 14 and 68.

VASICEK, O. An equilibrium characterization of the term structure. *Journal of Financial Economics*, v. 5, p. 177–188, 1977. Cited on page 22.

WACHTER, J. Portfolio and consumption decisions under mean-reverting returns: an exact solution. *Journal of Financial and Quantitative Analysis*, v. 37, p. 63–91, 2002. Cited on page 15.

WALLACE, S. W.; ZIEMBA, W. T. *Applications of stochastic programming*. [S.l.]: SIAM, 2005. Cited 2 times on pages 55 and 56.

WETS, R.-B. Programming under uncertainty: the equivalent convex program. *SIAM Journal on Applied Mathematics*, SIAM, v. 14, n. 1, p. 89–105, 1966. Cited on page 55.

WETS, R.-B. Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM review*, SIAM, v. 16, n. 3, p. 309–339, 1974. Cited on page 55.

ZENIOS, S. A.; ZIEMBA, W. T. *Handbook of Asset and Liability Management - Volume 1: Theory and Methodology*. 1. ed. United Kingdom: Elsevier, 2006. Cited 6 times on pages 8, 14, 37, 41, 42, and 68.

ZENIOS, S. A.; ZIEMBA, W. T. *Handbook of Asset and Liability Management - Volume 2: Applications and Case Studies*. 1. ed. United Kingdom: Elsevier, 2007. Cited 3 times on pages 14, 44, and 68.

ZHANG, M.; KÜÇÜKYAVUZ, S.; GOEL, S. A branch-and-cut method for dynamic decision making under joint chance constraints. *Management Science*, INFORMS, v. 60, n. 5, p. 1317–1333, 2014. Cited 3 times on pages 55, 59, and 69.

ZIEMBA, W. T. *The Stochastic Programming Approach to Asset, Liability and Wealth Management*. Vancouver, Canada and Londres, England: AIMR Publisher, 2003. Cited 4 times on pages 8, 14, 37, and 68.

ANNEX A – Achievements - Articles

- Published papers
  - de Oliveira, A. D., Filomena, T. P., Perlin, M. S., Lejeune, M., & de Macedo, G. R. (2017). A multistage stochastic programming asset-liability management model: an application to the Brazilian pension fund industry. Optimization and Engineering, 18(2), 349-368.
  - Oliveira, A. D. D., Filomena, T. P., & Righi, M. B. (2018). Performance comparison of scenario-generation methods applied to a stochastic optimization asset-liability management model. Pesquisa Operacional, 38(1), 53-72.
- Papers under review
  - A.D. de Oliveira, T.P. Filomena and L. Ferreira "Implicit Extensive Form for Multistage Stochastic Programming Models: A New Approach," under review at Annals of Operations Research
- Working papers
  - L. R. Sant'Anna, A. D. de Oliveira, T. P. Filomena and J. F. Caldeira " Convex Reformulation for Cointegration to Solve the Index Tracking Problem,"
  - M. Lejeune, F. Margot and A.D. de Oliveira "A Chance-Constrained Model with Decision-Dependent Uncertainty for MEDEVAC Operations,"
- Conference papers
  - DE OLIVEIRA, ALAN DELGADO; FILOMENA, TIAGO PASCOAL ; DE MACEDO, GUILHERME RIBEIRO . Administração de Ativos e Passivos - Uma abordagem de otimização estocástica. In: Simpósio Brasileiro de Pesquisa Operacional - SBPO 2015, 2015, Porto de Galinhas. ANAIS DO XLVII SBPO, 2015. v. XLVII.
  - OLIVEIRA, A. D.; FILOMENA, T. P. . Stochastic scenario generation: An empirical approach. In: 1º ETC - Encontro de Teoria da Computação, 2016, Porto Alegre. Anais do XXXVI congresso da sociedade brasileira de computação., 2016. v. 2016. p. 883-886.

ANNEX B – Achievements - Experiences

- Research Experience
  - Researcher / Visiting Doctoral Student, Decision Science Department
    - Institution: The George Washington University
    - Period: Aug 2017 - Dec 2017
    - Supervisor: Dr. M. Lejeune.
    - Project: Office of Naval Research (ONR) Grant N00014-17-1-2420, entitled "MINLP Methods for Chance-Constrained Problems with Endogenous and Exogenous Uncertainty".
    - Tasks: Design and coding of reformulations and algorithms for integer nonlinear optimization problems.
- Conference/Seminars Presentation
  - Lecturer
    - Conference: Simpósio Brasileiro de Pesquisa Operacional - SBPO
    - Date: Aug 25, 2015 - Aug 28, 2015
    - Title: Administração de Ativos e Passivos - Uma abordagem de otimização estocástica.
    - Conference: INFORMS Annual Meeting (scheduled)
    - Date: Nov 4, 2018 - Nov 7, 2018
    - Title: Mixed-integer Nonlinear Programming Method For Chance-constrained Models With Endogenous And Exogenous Uncertainty.
  - Poster
    - Conference: 1º ETC - Encontro de Teoria da Computação
    - Date: July 04, 2016 - July 05, 2016
    - Subject: Stochastic scenario generation: An empirical approach.