PhD Thesis

Applications of Biased Randomised Algorithms and Simheuristics to Asset and Liability Management

Armando Miguel Nieto Ranero

SUPERVISORS:
Dr. Ángel Alejandro Juan Pérez.
Dr. Montserrat Guillén i Estany
Dr. Renatas Kizys

Universitat Oberta de Catalunya

Abstract

The Asset and Liability Management has captured the attention of academics and financial researchers over the last decades. On the one hand we should try to maximise our wealth taking advantage of the financial market, and on the other hand, we must cover our payments (liabilities) along the time. The purpose of ALM is to give the investor a series of resources or techniques to select the appropriate assets of the financial market to obey the aforementioned two key factors: comply with our liabilities and maximising our wealth.

This thesis presents a set of techniques that are capable of tackling realistic financial problems without the usual requirement of considerable computational resources. These techniques are based on heuristics and simulation. Specifically, a biased metaheuristic model is developed that has a direct application in the usual immunisation operation of insurance companies. The algorithm makes it possible to efficiently select the smallest number of assets, mainly fixed income, on the balance sheet and that guarantees the company's obligations. This development allows incorporating the credit quality of the issuer of the assets used. Likewise, a portfolio optimisation model with liabilities is developed and it is solved with a genetic algorithm. The portfolio optimisation problem differs from the usual one in that it is multi-period, and incorporates liabilities over time. Additionally, the possibility of external financing is included when the entity does not have sufficient cash. These conditions give rise to a complex problem that is efficiently solved by an evolutionary algorithm. In both cases, the algorithms are improved with the incorporation of Montecarlo simulation. This allows the solutions to be robust when we consider realistic market situations.

The results are very promising. This research shows that simheuristics is an ideal method for this type of problem. On the one hand, it is capable of solving problems

of considerable size avoiding unaffordable computational costs, and also avoiding excessive simplification of the model, which would lead to a false representation of reality. On the other hand, the incorporation of the simulation guarantees us that the results are, within the limitations of the financial model, realistic.

Brief History and Acknowledgements

Con esta tesis se repara al menos parcialmente uno de los errores más graves que he cometido en mi vida. Cuando hace 30 años terminé mi licenciatura en Física Teórica, que simultaneé con la especialidad en Electricidad, Electrónica e Informática en la Facultad de Física de la Universidad de Valencia, y al mismo tiempo que terminaba el grado de Piano en el Conservatorio, me decidí a realizar mi doctorado. Mi padre, que no disponía de ningún contacto en el mundo universitario ni en el mundo científico, trató por todos sus medios facilitarme el que hasta la fecha era mi máximo sueño, doctorarme en Estados Unidos en Física. Por mi parte, hice numerosos intentos pero la verdad es que no hubo demasiada fortuna. Sin embargo, mi padre sí lo consiguió, y lo hizo superando su irreconciliable deseo que ninguno de sus hijos rehiciera su vida lejos de él. Sus esfuerzos derivaron en un Full Professor de la Universidad de Urbana en Illinois, una de las mejores universidades a nivel mundial en Física, y en especial en Física de Estado Sólido. Este catedrático revisó mi expediente y consideró que podía unirme a su departamento y becado para poder realizar la tesis. La verdad sea dicha, yo no era consciente del primerísimo nivel de esta universidad, ni de que es una de las universidades con mayor concentración de físicos galardonados con el máximo premio posible. Mi pasión estaba centrada en Cosmología, Astrofísica y Física Matemática y al ver que el campo de investigación se centraba en Estado Sólido le llamé, se lo agradecí y le dije que no me interesaba.

Al año, traté de realizar el doctorado en el propio departamento de Física Teórica de la Universidad de Valencia. Ahí me asignaron una serie de temas bastante interesantes, la principal investigación versaba sobre la discusión que en aquel momento existía sobre si el neutrino tenía o no masa. Tuve incluso la suerte de contactar en una ocasión con el laureado físico Lincoln Wolfeinstein, eminente físico de partículas y autor de los trabajos sobre los que yo empezaba a investigar. El departamento me

asignó a un ruso que venía huyendo del antiguo telón de acero y para mi desgracia su único interés consistía en que yo le ayudara a buscar piso, concesionarios de autos de segunda mano y cosas tan para-científicas. El resultado final fue que no tenía ningún apoyo y abandoné.

Un amigo me comentó sobre la posibilidad de acercarme al departamento de Ingeniería Nuclear de la Universidad Politécnica de Valencia y concretamente dirigirme a un profesor del que él me dio buenas referencias. Así hice y me admitió como doctorando. En esta ocasión el trabajo trataba de fotoneutrones. Una de las aplicaciones de la investigación tenía que ver con determinados tratamientos en oncología y se demostraba que dependiendo de cómo se bombardea al paciente con el acelerador lineal, los neutrones que se liberan como consecuencia de la radiación suministrada podían ser nocivos y por tanto, el tratamiento convertirse en contraproducente. Llegué a desarrollar un simulador con esta interacción de modo que se podía calcular el impacto programando la geometría adecuada. Para mi desgracia, una vez realizado el trabajo y aprobado el título de la tesis, mi director consideró que debíamos probar empíricamente los resultados para poder presentar la tesis y, por tanto, debía agenciarme con un acelerador lineal. Traté de convencer a los hospitales regionales de la importancia del estudio para que me cedieran sus aceleradores, pero él no mostró ningún apoyo, no conseguí ningún acelerador, y tuve que abandonar nuevamente.

En una de las ocasiones que volvía de la Universidad Politécnica, me crucé con un antiguo amigo de mi padre que se interesó por mi caso. Me comentó acerca de un catedrático también del Politécnico, aunque trabajaba en temas relacionados con la Química y particularmente con la depuración de aguas residuales. Nuevamente tuve suerte y me aceptó. Pero esta vez no me permitió acceder a ningún curso de doctorado, prometiendo la tesis para más adelante. Diseñé un sistema CAD que además simulaba toda la circuitería hidráulica del agua, creaba de forma autónoma los planos de la planta incluyendo las cotas, y apliqué la teoría de Grafos de manera que el simulador podía calcular todo tipo de circuitos sin que hasta la fecha existiera esa potencia de cálculo. La tesis nunca llegó. Me ofrecieron un puesto de trabajo en el departamento de Informática en la aseguradora que en aquel momento dirigía mi padre, aunque quien me ofreció el puesto lo hizo sin que mi padre lo supiera. Una vez se enteró mi padre me sugirió que continuara investigando pero finalmente abandoné esa tesis fantasma. Después supe que de mi trabajo surgieron varias tesis aunque nunca fui citado en ninguna de ellas.

En la aseguradora decidí dedicarme a la profesión abandonando casi por completo la idea de doctorarme. Me licencié en Economía, en Ciencias Actuariales y Financieras, obtuve un Master en Bolsa y Finanzas y me acredité a nivel internacional. Pasados unos 10 años decidí volver a intentarlo, aunque esta vez a través de la Universidad a Distancia. Me matriculé en Estadística e Investigación Operativa, eligiendo las asignaturas que más próximas estuvieran a las finanzas. Al terminar los créditos me puse en contacto con un profesor al que le propuse realizar la tesis en finanzas cuánticas. Es un tema que desde un punto de vista teórico me resultó siempre interesante aunque apenas hay economistas que lo trabajan por requerir otros conocimientos en física que no son habituales para ellos. El profesor no aceptó la idea y tampoco se mostró interesado en dirigir ninguna tesis y yo desistí una vez más al no encontrar a nadie a quien dirigirme.

Me ofrecieron al cabo de unos años doctorarme en un departamento de la Facultad de Economía, departamento que no mencionaré por no citar al pecador. El doctorado propuesto era simplemente pagar las tasas correspondientes y realizar un trabajo menor que recogiera en parte la experiencia laboral que ya disponía. Es decir, la típica tesis que por desgracia vemos más de lo que nos gustaría y que desmerece el esfuerzo de las tesis verdaderas. En esta ocasión fui yo quien se negó a semejante propuesta. Respondí diciendo que si algún día yo escribo una tesis, debe tener valor y debe ser algo que me haya obligado a esforzarme.

Volvieron a pasar otros diez años y tuve la inmensa suerte de conocer a Alberto Ferrando, prestigioso actuario y por aquel entonces Presidente del Col·legi d´Actuaris de Catalunya. En una de las varias y excelentes comidas que hemos compartido me propuso conocer a una reputadísima investigadora de la Universidad de Barcelona, Montserrat Guillén, con la idea de que yo le contara mi, digámoslo así, curiosa historia. Y así fue. En una nueva comida, Montse me brindó el privilegio de aburrirle con mi historia. A pesar de que Montse es una de esas personas que marcan la diferencia en las universidades, después de dedicar dos años de peleas con la burocracia, elevando el caso a todos los niveles vio que las distintas disposiciones legales desde que yo terminara mis distintas licenciaturas me dejaban en terreno de nadie. Todo

apuntaba a tener que volver a renunciar, pero Montse tuvo la feliz idea de dirigirme a la Universidad Oberta de Catalunya y hablar de mi a un fuera de serie, inagotable trabajador y brillante investigador, que hoy es el principal supervisor de esta tesis. Por fin, la historia iba a ser derrotada y el error de juventud que cometí, finalmente reparado.

Siento muchísimo que mi padre ya no pueda felicitarme, con absoluta seguridad sé que se habría sentido enormemente orgulloso de mi, él sabía muy bien lo mucho que yo deseaba esto. Pero se que está observando allá donde moran los justos con enorme satisfacción. No puede ser de otro modo que mi primer agradecimiento lo dedique a su memoria. Y junto a él, a mi madre, a quien debo mi vida, el titán Atlas que sobre sus espaldas ha soportado silenciosamente una familia numerosa, forjado a un hombre brillante como fue mi padre y con su amor moldeó y apaciguó mi carácter terriblemente rebelde.

No hay nadie que conozca mejor esta historia que Noemí, quien me dio mis cinco hijos a los que adoro con locura. Ella vio cómo desde el principio una y otra vez recibía un nuevo revés. Y finalmente estos últimos cinco años me ha dado el espacio que necesitaba para poder trabajar esta tesis. Sería injusto simplemente agradecerle su sacrificio, la verdad obliga a reconocer que ella es en realidad coautora y cooperadora necesaria para que esto haya sido posible.

Mis hijos merecen un agradecimiento especial. Cuántas veces les he tenido que decir *me voy a estudiar*, mientras en otras familias podían disfrutar de su padre. Cuántas veces habrán pensado qué extraño es que su padre tenga que estudiar a estas alturas, sin que tenga ninguna necesidad. Y es que he sacrificado muchas horas de estar con ellos y quiero pedirles perdón por ello aunque confío en que haya sido un ejemplo que puedan seguir y recordar en el resto de sus vidas.

Dos son las personas que han demostrado paciencia infinita al escucharme un día sí y otro también con cada uno de mis avances, mis logros, mis rectificaciones y dificultades, en cada uno de los pasos que he dado para llegar hasta aquí. Mis queridos Nati y Ramón merecen un pedestal y gracias a su apoyo, a los cientos de horas escuchándome y miles de kilómetros recorridos que hemos corrido juntos, he podido desarrollar todo el trabajo. Su aliento y su ánimo constante ha impedido que existiera el más mínimo momento de flaqueza o cansancio.

No hay palabras de agradecimiento suficientes para mis tres supervisores, sin los cuales y sin sus enseñanzas nada de esto habría sido posible. He aprendido mucho más de lo que esperaba y me han enseñado mucho más de lo que se necesitaba. Especialmente he de mencionar a Alberto Ferrando, con quien estaré en deuda siempre por presentarme a mi admirada Montse, y a Ángel que mucho más allá de su papel de director de mi tesis, ha sido un maestro y un amigo.

Y quien me conoce sabe que debo terminar con quienes son la fuente de mi energía: Beethoven, Mozart y Bach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In finance, every decision leads to the same question: how to increase our wealth. To do that, we allocate our assets to a huge variety of possibilities, obtaining expected returns for each one at the same time that we are assuming risks. This is equivalent to an optimisation problem, that is, how do I have to invest my assets to get the maximum return subject to a given threshold for the risk level?, or in a similar way, how do I have to invest my wealth assuming the least possible risk and having, at least, a given return? This point was first studied by Markowitz (1952) where he supposed that the returns of assets obey a probability function. From that point of view, the return of an asset is a random variable and what we consider as a risk has to be a measure of dispersion, generally, the standard deviation or momentum of order two. The basic problem considers a set of assets with known probability functions and with finite variance-covariance matrix. Afterwards, many constraints have been added to the basic Markowitz theory to get a more realistic model. It is worth noting that this model only takes into account characteristics that are imposed by the financial market. Nevertheless, the reality is that we don't have only assets: we also have liabilities, so, before thinking about how to increase our wealth, we have to think about how to comply with our obligations.

In other words, liabilities have to be considered in the basic problem of maximising our wealth. This is studied in literature under the name of Asset and Liability Management (ALM). Many different approaches have been developed so far. Probably, the most common approach is the duration theory by Macaulay (1938). The key of the ALM theory is to take into account that our liabilities are scheduled and we have to select those assets that meet the obligations in time. So, the interest rate arises as a fundamental variable in the asset allocation. Thus, the portfolio optimisation has to be split in two steps: the first one is to determine how to guarantee our liabilities by selecting the minimum number of assets from our portfolio; the second step is to invest the rest of assets according to our risk preferences.

As I have mentioned before, the interest rate is crucial because depending on the term structure of interest rates we will have to choose the best moment and timespan for each investment. But the question is that the value of both liabilities and assets vary when the interest rates change, and as we will not have a perfect match between our obligations and the cash-flows generated by our assets, the variation of the present value will depend directly on these changes in the term structure of interest rates. The financial immunisation has to do with how to jointly avoid these variations in the value of assets and liabilities. The Macaulay Duration Theory establishes a parameter, duration, so that if we select assets with the same duration to which our liabilities have, under a slight change in the interest rate, we won't have differences of value between assets and liabilities. There is another approach, the Optimal Control Theory (Kirk, 2012), in which we establish correlations among assets and liabilities, and the variability of the interest rate, so, imitating the basic concept of the theory of portfolio optimisation by Markowitz, we state an optimisation problem having a new risk constraint involving the interest rate. A third and more realistic approach is the Cash-Flow Matching (Mitra and Schwaiger, 2011). The idea involves the selection of assets so that we will have the amount of money at the time we have to pay our obligations. As the reality is that our positive cash-flows won't match in time with those negatives, we will have to select enough assets to guarantee a positive amount of money so that every time we have a negative cash-flow, that quantity will be enough.

One very important fact is that interest rate is not flat, and it is not static. Therefore, we should consider that the interest rate for each spot time is a random variable. Two important models have been developed: Vasicek (1977) and Cox et al. (1985). If we consider this characteristic of the term structure of the interest rate, we cannot use the duration model. Additionally, the cash-flow matching is just a sort of real simulation. So, we can consider new hypotheses or constraints that the theory of optimal control cannot. For example, if our cash-flow balance in any moment of the time is negative, we have to consider paying an extra interest rate just because we have to borrow money. Even, we can find examples in the legal regulation that limits the number of consecutive negative balances or forbid negative balances at the beginning of the year. These kinds of constraints make the problem of finding the optimum assets very complex, so heuristic and metaheuristic approaches can be strong candidates.

The main contribution of this doctoral thesis will be the application of Simheuristics to solve the selection of assets based on cash-flow matching. The need of using simulation in combination with Heuristics/Metaheuristics is based on the nature of Financial Market that is stochastic. This study will face common financial scenarios, considering realistic financial structures and moving away from simplistic hypotheses. So far, the models have had to simplify the balance structure due to the highly demanding mathematical formulation. But now, we will illustrate how we can deal with heuristic algorithms to solve those kinds of problems. On the other hand, we will be able to improve the characteristics of the model itself, with common constraints and other features of the financial market. This will allow us to get a framework that is applicable to the financial industry and specially to the insurance market.

1.2 Thesis Structure

This thesis is a set of works; some of them have been published in conferences and in journals as it is described below. They are presented in the same temporal sequence they have been both developed and published. The document treats to be selfcontained on the basis of some fundamental knowledge in finance and programming. The specific fields used in this thesis are explained in chapter 2, and they are merely a dense and brief exposition of the starting point of this research. If the reader is an expert in those fields, it might not be necessary to read the aforementioned chapter.

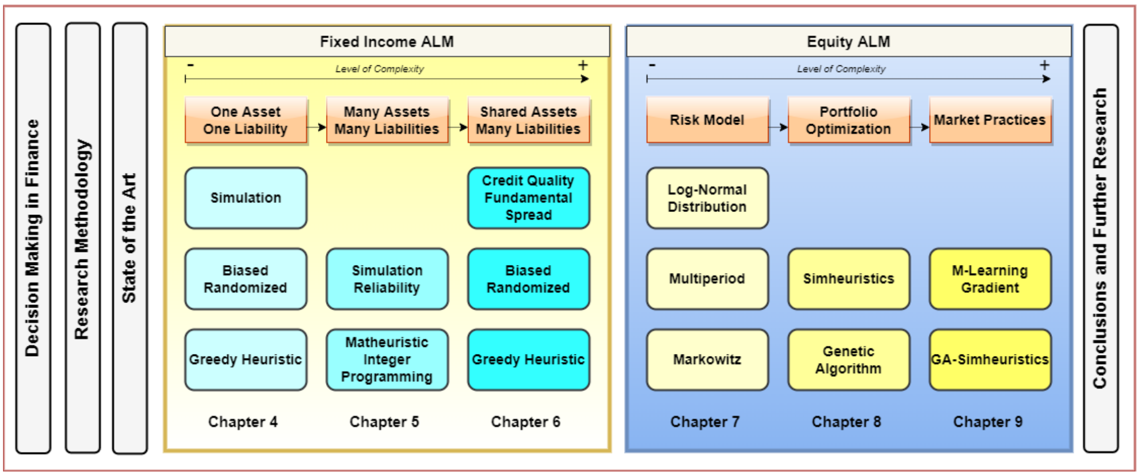


Figure 1.1: General Structure of the Thesis

The document is structured as follows (Figure 1.1):

- In this Chapter 1, **Introduction**, the main research objective is discussed.

- In this Chapter 2, entitled as **Decision Making in Finance**, I introduce a brief summary of the main financial issues that we will work along the research, that includes the theory of asset and liability management and the well-known Markowitz theory for portfolio management.

- In Chapter 3, entitled as **Research Methodology**, I briefly introduce a summary of the main theoretical fields we deal through this thesis. First, I present the family of algorithms classified as Metaheuristics and Simheuristics. As a preliminary note, we can say that Heuristics doesn't try to get the best solution using intractable computational times but to get good and useful solutions. These kinds of algorithms are the keystone of all this research. Indeed, it is not so important to reach the exact solution as to get a good one in the financial market. The nature of finance is purely stochastic, that means that once you have calculated a perfect solution, it has expired just one second after. Sacrificing this ambition gives us the opportunity of introducing more realistic models which their set of solutions can be explored thanks to these kinds of algorithms until we get a good one. I also present one specific case of metaheuristics, the genetic algorithms due to the relevance it has in order to solve a very demanding problem that is treated in the last chapters.

- Chapter 4, entitled as **ALM in Financial Markets: State of the Art**, is a wide survey of what ALM has been along the last decades. As commented in this chapter, the asset and liability management is a field whose interest has been notably increased as a result of the more aggressive environmental conditions that surrounds the financial market. The main conclusion of this chapter points out clearly the need to explore heuristics as a way to tackle the increasing complexity of the financial markets.

  This chapter is an update of Nieto et al. (2022a), which has been approved and is awaiting publication.

- In Chapter 5, entitled as **A Simheuristic Algorithm for Reliable ALM**, I present a simple model for the case of matching liabilities with assets in the scope of an insurance company. The problem is simplified by imposing one asset to one liability constraint to mainly focus on the algorithm. This is a first and experimental approach that, in spite of their simplifications, shows how the metaheuristics can be used to solve this kind of problems. The chapter concludes with a study of reliability of the solutions using Montecarlo simulation.

  With the proper adaptations and changes, this chapter is derived from the following paper Bayliss et al. (2020a).

- In Chapter 6, entitled as **Matheuristic with Simulation for Stochastic ALM**, we explore the best solution of the assignment of assets to cover the liabilities in a stochastic framework. It shows that the deterministic solution is not the best one when we consider stochastic behaviour in both assets and obligations. It also considers the possibility of attributing many assets to many liabilities when the problem is solved.

  This chapter has been based on Bayliss et al. (2020b).

My contributions to this paper have been mainly the conceptualisation of the model together with C. Bayliss, the validation of the results together with A. Juan, and the writing–original draft preparation together with C. Bayliss.

- Chapter 7, entitled as **ALM in Insurance Firms**, is a natural extension of the previous chapter. In this one, we remove the main constraint one to one. The result is a very fast and powerful algorithm that solves the traditional match flow of an insurance company having a list of liabilities along the time and a set of assets in its balance sheet.

  The full article, adapted in this thesis, is published as Nieto et al. (2021).

- In Chapter 8, entitled as **The Multiperiod Risk Model - Markowitz Revisited**, I develop a theorem that proves the expression of the standard deviation of an equity portfolio in a multiperiod context, where the price of each asset obeys a Log-Normal distribution function. This theorem was necessary to formulate adequately a portfolio optimisation problem in a multiperiod context, which is the objective of the next chapters. This theorem proves that a n assets $\times$ p period problem is equivalent to a $n \times p$ assets in one period, where the correlations of the assets have to be with the correlations of each asset with itself in the past and the future.

- Chapter 9, entitled as **A GA-Simheuristic for the Stochastic and Multi-Period ALM**, is the most extended work of this research. This chapter develops a genetic algorithm to solve a portfolio optimisation problem in a multiperiod context and with liabilities. It also includes the option of a loan if punctually the treasury is not enough to fulfil with our liabilities. The algorithm is extended with a simheuristic version. The aim of this extension is to provide the genetic algorithm with a wider vision in terms of simulated risks. The GA can be considered a deterministic solution and it implies that some deviations coming from the stochastic nature of the financial market are not recognised into the algorithm. Thus, introducing inside the genetics a simulated value for the utility function delivers us a more reliable and realistic solution.

  This chapter is taken from Nieto et al. (2022b), with appropriate modifications:

- Chapter 10, entitled as **Simheuristics in Finance: Managerial Insights**, is an application of four common cases from a managerial point of view, using simheuristics. In this chapter I introduce a little piece of machine learning to improve the search of better solutions, and other improvements in the genetic algorithm to avoid the natural noise that the random selection of individuals provokes. The four common cases of the market are studied with diverse conclusions. In some of them, the conclusion is that traditional techniques don't introduce too many deviations, but in others, the study concludes that it is highly recommended to apply these techniques to obtain more robust solutions. A summary of how simheuristics is applied in finance and the main aspects from a managerial perspective of view can be found in our recently published paper Doering et al. (2022).

- Chapter 11, **Final Conclusion and Further Research**, is a short summary with the main conclusions of this work and new possible new lines of research.

- Finally, the Thesis ends by attaching two appendices. The first one, Appendix A, lists all the outcomes of this research work, including all the papers, six in total, and conferences in which I have been the speaker, three in total. In the second one, Appendix B, I show the front page of the aforementioned six papers.

Chapter 2

Decision Making in Finance

2.1 Portfolio Management

The theory of Portfolio Management is by far the most prolific topic in the field of Finance. Although many theories or strategies are developed, the most important is the portfolio theory elaborated by Markowitz (1952). The theory is based on assets whose return is modelled as a Gaussian random variable. Let α be a market with N different assets, and let R_i the return of the asset indexed with the number i, as R_i behaves as a normal random variable, it is completely characterised with only two values, μ_i and σ_i, the expected return or mean and the volatility or standard deviation respectively for a given period of time. We also have to consider that assets are correlated as it is usual between random variables, and we represent the matrix variance-covariance as σ_{ij}. Let's suppose we have a portfolio of assets. If we consider that the value of my portfolio is equal to the monetary unit, and x_i represents the proportion of money we have in the asset i, at the end of the period we have the following value:

$$P = \sum_{i=1}^{i=N} x_i(1 + R_i) \tag{2.1}$$

Thus, the value of the portfolio is also a normal random variable since it is a linear combination of normal variables. The expected return and the volatility of the portfolio are respectively:

$$\mu_P = \sum_{i=1}^{i=N} x_i \mu_i \tag{2.2}$$

$$\sigma_P = \sum_{i,j=1}^{i,j=N} x_i \sigma_{ij} x_j \tag{2.3}$$

We can select any combination of x_i we want. However, we can find combinations with exactly the same portfolio volatility and a different expected return. It means that some possible portfolios are superior than others as we consider that portfolio A is better than B if $\sigma_A = \sigma_B$ and $\mu_A > \mu_B$. This can be synthesised in the very famous optimisation portfolio problem:

$$Max \sum_{i=1}^{i=N} x_i \mu_i \tag{2.4}$$

s.a.

$$\sum_{i,j=1}^{i,j=N} x_i \sigma_{ij} x_j \leq \sigma_P \tag{2.5}$$

$$\sum_{i,j=1}^{i,j=N} x_i = 1 \tag{2.6}$$

$$0 \leq x_i \leq 1 \quad \forall i \tag{2.7}$$

The set of pairs (μ_P, σ_P) that optimises the former expression have the shape of a parabola. That curve is called the Efficient Frontier because it represents the efficient way to invest given a set of assets. If you choose a portfolio that results in an outer point of the efficient frontier, you are assuming an extra risk unnecessarily.

Until now, we haven't considered non risk-free asset. When a non risk-free asset is present things change a little bit. You can regulate much more what amount of risk or volatility you are willing to take. This case was studied by J. Tobin (Tobin, 1958) where he formulated the two-fund separation portfolio. Basically, he proved that when we have a market with N risky assets and one risk-free asset, the risk-return pair of any admissible or feasible portfolio cannot lie above the capital market line (CML) (figure 2.1) in the risk-return space. Moreover, the portfolio represented by any point in the CML maximises the Sharpe Ratio (Sharpe, 1994), defined as follows:

$$S = \frac{R_P - R_f}{\sigma_P} \tag{2.8}$$

The Sharpe Ratio is the slope of the CML and represents how much money I will earn investing one monetary unit in risk.

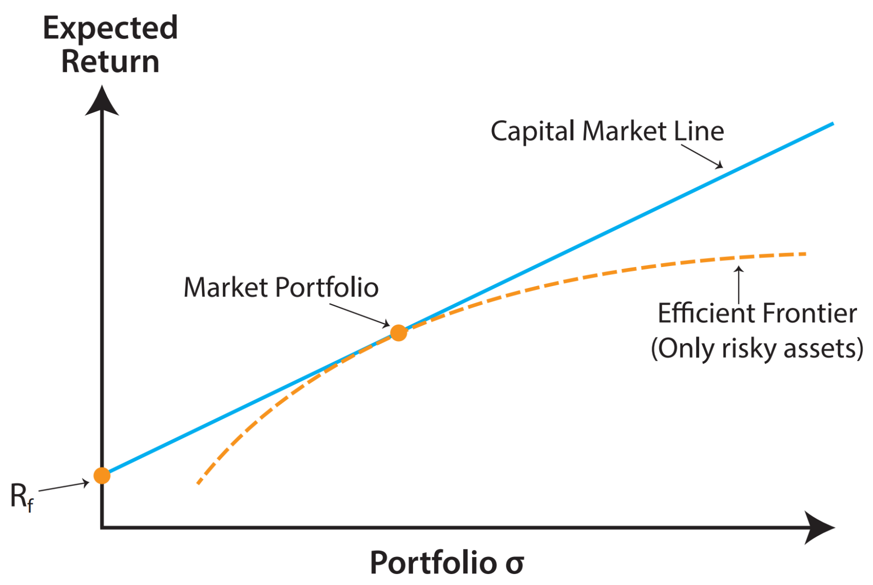


Figure 2.1: Capital Market Line

The lowest point $(0, R_f)$ corresponds to invest all the money in the risk-free asset. The Market Portfolio point (R_M, σ_M) corresponds to the opposite situation, i.e., all the money is invested in risky assets. The upper points need to borrow money, which is related to either a negative or short position in risk-free money, and that money is invested in risky assets. This situation is clearly theoretical since no one lends you money while you invest it in a risky market without an extra payment that compensates your risky investment and your own credit risk. So, the realistic capital market line goes from the risk-free point to the Market Portfolio point.

2.2 Asset and Liability Management

Asset and Liability Management refers to the set of techniques that a company apply to guarantee its debts assigning a number of assets, in a variable interest rate scenario. A financial institution, generally a bank or an insurance company, signs a contract in which firstly it gets money from the client and, in the long term, the institution has to pay back that money with a certain return. In the meantime, interest rates fluctuate so the financial institution assumes a risk because risk can decrease and in that case, the company is forced to have its own capital to keep safe the client's rights. So, the allocation of the assets is a key question to protect the rights of the financial consumer, and also for the health of the financial system. Apart from the interest rate, other aspects can change, as the credit quality of the asset issuers since they can go bankrupt, or the expected return of equity. This question is widely treated in literature and also in legislation. In fact, the legal regulation gives two options to the financial companies: having a matching between the assets and liabilities following precise rules contained in the regulation, or assuming a general provision for its liabilities. The problem of assuming the general provision is that is extremely conservative, and it implies that the company has to accept a reduction of its profit. There are two ways to face the ALM, the first one is in terms of accountability, and the second one is in terms of cash-flow. In the next subsections, I introduce these approaches.

2.2.1 Duration

One very common approach into the ALM theory is the study of financial duration for an asset. This was studied by Macaulay (1938). It is defined in a flat interest rate context and fixed cash-flows. The purpose of this approach is to have stability in our balance sheet under changes of the interest rate. When the interest rate fluctuates, we can have significant changes in the valuation of both our assets and liabilities. Let's suppose we have fixed income, we have a deterministic temporal structure of revenues coming from the coupons of our fixed income securities. In this class of assets, we cannot change the maturity date of each coupon, so the variation in the present value will depend on those dates. On the liabilities side we have the same situation. Clients have to be paid in specific dates. So the calculus of the provision is also affected by the interest rate and its fluctuations. As the amounts of money in assets and liabilities are not equally distributed along the time, it turns out that we have a mismatch in terms of total valuation. It means that the value of our company can be stressed just because the assets and liabilities differ in their present values under changes in the interest rate. The Duration approach guides us to select those assets we have in our balance sheet and assign them to our liabilities so that the total variation is negligible under changes of the interest rate.

The definition of the duration of an asset is:

$$D = \frac{\sum_i t_i PV_i}{\sum_i PV_i} \tag{2.9}$$

where PV_i is the present value of the cash-flow i, and t_i is the time in years or months of the cash-flow i. That expression is the coefficient of the linear term in the Taylor polynomial for the present value formula having a variation in the interest rate. So, the present value, upon small variations of the interest rate, the variation of the present value is as follows:

$$\Delta PV = D\Delta r \tag{2.10}$$

where ΔPV is the variation of the present value of the cash-flow, D is the duration and Δr is the variation in the interest rate. Let's remember that, in this approach, the interest rate is flat. As we can see, the variation of the interest rate is also flat, i.e., we will keep flat the term structure of interest rates all the time. If we choose a set of assets that have the duration equal to the duration of the liabilities, as a first approximation, the net present value of our portfolio won't change, so we get our goal. If the interest rate variation is not so small, we can add the second term of the Taylor's expansion, so that:

$$\Delta PV = D\Delta r + \frac{1}{2}C(\Delta r)^2$$

The C value, named Convexity, is the second term of Taylor for the present value expression. A complete development of this topic can be found in (Fabozzi, 1999). A complete survey of this approach can be found in this research work in chapter 3.

The main weakness of this approach is that it immunises against interest rate changes, but not against mismatching between cash-flows of assets and liabilities. If we have to pay before we collect the money from our assets, we will have to borrow money, so we will have an extra cost of capital that the duration approach doesn't consider. This problem is treated in the next subsection.

2.2.2 Cash-Flow Matching

In general, cash-flow matching is the situation we have when both assets and liabilities match in time. If we had that situation, we could guarantee our liabilities with an accuracy of 100%. The main difficulty in the real world is that the cash-flow coming from our assets does not match perfectly in the dates we have to make, and also, we have different risks among the cash-flows due to their different nature. A plausible matching is to have sufficient positive cash-flows before the maturity dates of our liabilities. Obviously, this circumstance is also idyllic but we could have moments along the time where we would have to make a payment but did not have enough money to perform such payment. In this case, we may be forced to borrow money from a credit institution. All these considerations or circumstances lead us to define the balance S_i in a recursive way:

$$S_{i+1} = \begin{cases} A_{i+1} - L_{i+1} + S_i \frac{D_i}{D_{i+1}} & S_i \geq 0 \\ A_{i+1} - L_{i+1} + S_i \frac{C_i}{C_{i+1}} & S_i < 0 \end{cases} \tag{2.11}$$

Where $C_i > D_i$ and they are both discount factors, A_i is the asset in the time i, and L_i is the liability also in i.

The balance is the amount of money we have in each moment of our life. It is the sum of three components: the addition of positive income (assets), the addition of negative income which are the payments we have to perform, and the previous balance we had coming from the previous period, with its interests. If the previous balance is negative, we need to borrow money, which is equivalent to increase the interest rate and capitalise that negative value with that more elevated rate.

Now, our hedging problem is to select a set of assets from our portfolio having such that $S_n \geq 0$, i.e., such that our final balance cannot be negative. Our optimisation problem has many possibilities. The most direct approach is to maximise the value of the asset portfolio of the company having allocated some of them to the immunisation; it is equivalent to say that we want to select the minimum number of assets for the cash-flow matching. There is another possible optimisation strategy very related to the insurance market. In this market, the mathematical provision concerning to the liabilities have to be calculated with a fixed interest rate. The interest rate you have to use can either be the one imposed by the legal supervisor, or the one you assured to your client, and for that, you must have cash-flow immunisation that guarantees that interest rate. So, you would prefer to immunise those liabilities with those assets with the internal rate of return similar to the guaranteed interest rate to the client. In that case, you would not need to increase the amount of provision in case of having smaller regulatory interest rate. Let's take into account that for a value of 1 billion euros in liabilities, a provision can change around 10 million euros per 0.1% decrease in the interest rate. That could shock the company, even lead it to bankrupt. So, ALM can be absolutely determinant to the survival of the financial institution. I treat this problem of asset allocation in chapters 4 and 5.

Another important consideration in the cash-flow matching is that the nature of these cash-flows can be stochastic. Concerning the liabilities, they depend on the contractual clauses, and they can change in quantity and maturity if certain events are triggered. Those events have to be specified in the contract. On the other hand, our assets have fluctuations in valuation following the evolution of the financial market as a general rule. This stochastic behaviour makes possible that almost any scenery would be possible, so we have to treat with random variables and probability functions. The constraints in our optimisation problem have to deal with tolerable risk values, such as:

$$E[S_n] \geq 0$$
$$\delta_1 \leq P(f(S_n)) \leq \delta_2$$

It means that our final balance has to be necessarily positive at least in a mean term because, in general, we cannot guarantee a particular figure due to its random nature. The second expression indicates that we can desire or need a function of the balance with a probability P within a specific interval. These conditions can be like $P(S_n < 0) < 0.05$ or $P(S_i < 0 \cup S_{i+1} < 0 \cup S_{i+2} < 0) < 0.05$. This problem is a stochastic version of ALM that is also treated in the next chapters and in general throughout this research work.

Another possible cash-flow matching problem is treating it as a Portfolio Optimization Problem with tracking error. In this case, our liability is the portfolio we want to follow, and we have to select our assets to minimise a tracking error index with some constraints (the final balance has to be positive or, in case of a stochastic model, its probability has to be greater than a specific value).

2.3 Legislation

Although this research work is focused on the development of simheuristic algorithms to be applied to the ALM problem, we cannot lose sight that the companies where ALM is highly used, mutual funds, insurers and banks, operate in regulated environments in any country of the world. By far, the site where the regulation is most advanced is Europe. Specifically, in the field of matching and calculation of technical provisions, the EU published the directive 2009/138/EC of the European Parliament and of the Council. This directive, among other regulations, establishes the requirements that an insurance company has to accomplish concerning the technical provisions and the matching adjustment. In particular, we can find in article 77b the rules for the matching adjustment to the relevant risk-free interest rate term structure, and in article 77c the rules for the calculation of the matching adjustment. The first rule we read is that insurers and reinsurers may apply a matching adjustment to the relevant risk-free interest rate term structure to calculate the best estimate of a portfolio of life insurance or reinsurance obligations subject to prior approval by the supervisory authorities. After that, it establishes the conditions of eligibility of the assets to be part of this matching adjustment. These assets can be bonds or other assets with similar cash-flow characteristics and they have to be maintained over the lifetime of the obligations. This rule has been specifically pointed out in chapters 4 and 5 when we mention "we freeze the assets". The article continues declaring that the expected cash-flows of our selected assets have to replicate the cash-flows of the obligations and any mismatch cannot give rise to significant risks. A clever solution for those mismatches is integrating a credit policy, as I have done along all our research. With that solution, the mismatching doesn't exist as we interchange it for another obligation, the loan repayment. Nevertheless, not all the rules are so friendly. There are three points that are, in my opinion, very restrictive:

- the matching adjustment is not permitted if the contracts underlying the obligations give rise to future premiums. The idea underlying this statement is that the future premiums undermine the ability of the insurer to manage its match adjustments.
- the cash-flow generated by the portfolio of assets has to be completely deterministic (except if they depend on inflation in the same way as obligations do). The idea is to minimise all kinds of risks and as a consequence, it restricts the class of assets to bonds and similar securities.
- if the insurer applies the matching adjustment to a portfolio of obligations, it is not permitted to revert back to an approach that does not include a matching adjustment. If a mismatch is produced and it lasts two months, the insurer is obliged to communicate this situation to the supervisor and it will have to cease of applying any matching adjustment for a period of 24 months.

It seems that the regulator is so severe because there is not any advanced method to calculate the intrinsic risk of a matching adjustment, and so, a technique to manage with guaranties the matching adjustment in statistical terms. Precisely, these kinds of clauses incite the importance of researching in the ALM field.

The European Insurance and Occupational Pensions Authority (EIOPA) published a consultation paper to review the impact of the Solvency II Directive (the European insurance regulation) https://www.eiopa.europa.eu/document-library/consultation/consultation-paper-opinion-2020-review-of-solvency-ii_en.

This paper covers many topics including the Matching Adjustment. The results are quite surprising. Only two countries in all Europe are using matching adjustment portfolios, the UK and Spain. (In fact, in the EU, and after the brexit, only Spain is using it as the UK is not a member of the Union anymore). 18 UK firms and 14 in Spain. The consultant Milliman tries to explain why other countries refuse to apply this regulation (https://assets.milliman.com/ektron/SII_2020_EIOPA_Opinion_MA.pdf). The main issue that Milliman points out is that if a firm applies the rules of the matching adjustment, the Solvency Capital Requirement has to be increased. It is a real contradiction with the objectives pursued by Solvency II. The reason is that all the assets we choose for the matching cannot be considered for a parallel requirement of diversification. So, this regulation has a clear disincentive effect in many ways. Therefore, this fact must serve as a stimulus to investigate advanced methods that allow us to resolve the limitations in management that currently exist and that condition the legislator.

Chapter 3

Research Methodology

3.1 Metaheuristics

Metaheuristics are general-purpose algorithms with which Combinatorial Optimisation Problems can be addressed (COP). These COPs are infeasible for exact methods and should generally give a satisfactory approximation to the true optimum within reasonable computing time. The advantages of metaheuristics include their robustness to changes with respect to objective functions, constraints and problem size, their simplicity and transparency as well as their transferability to similar problems (Gilli et al., 2011). Metaheuristics have been classified in numerous ways depending on their characteristics. In the following, the most common classification features are presented.

- **Trajectory versus discontinuous methods:** Metaheuristics can be distinguished based on whether they follow one search trajectory characterised by a closed walk or allowed for larger jumps in the neighbourhood including the temporary acceptance of worse solutions in order to escape local minim (Birattari et al., 2001). Local search algorithms are generally trajectory.

- **Population-based versus single-solution based methods:** In single-solution based metaheuristics, such as tabu search or simulated annealing, a single solution is manipulated at each step of the process, whereas population-based metaheuristics, such as genetic algorithms or ant colony optimisation employs a set of solutions in order to efficiently explore the search space (Birattari et al., 2001). Thus, the former are exploitation or diversification-oriented, while the latter are exploration or intensification-oriented (Talbi, 2009). This fact demonstrates the trade-off between the exploitation of the entire search space and the intensity of the search within specific areas of the search space due to computing time limits.

- **Nature-inspired versus non-nature-inspired methods:** Many metaheuristics are inspired by nature, such as simulated annealing (physics) or ant colony optimisation (swarm intelligence, social sciences) or genetic algorithms (biology) (Talbi, 2009).

- **Memory usage versus memoryless methods:** Some metaheuristics, such as tabu search, explicitly retain a search history by which the future search direction is influenced (Birattari et al., 2001).

- **Iterative versus greedy methods:** This distinction concerns the starting point of the method: Metaheuristics either start with a complete generated solution that is transformed at each iteration (iterative) or with an empty solution that is filled with decision variables at each step until completion (greedy) (Talbi, 2009).

- **Deterministic versus stochastic methods:** This classification is to be accounted for in the performance evaluation of the metaheuristic. A deterministic method will always yield the same solution for a given initial solution, whereas stochastic methods apply rules that are, to some extent, random and thus possibly lead to different solutions (Talbi, 2009).

A complete classification including further classification aspects, such as dynamic or static objective function, and the most common metaheuristic methods is provided in 3.1.

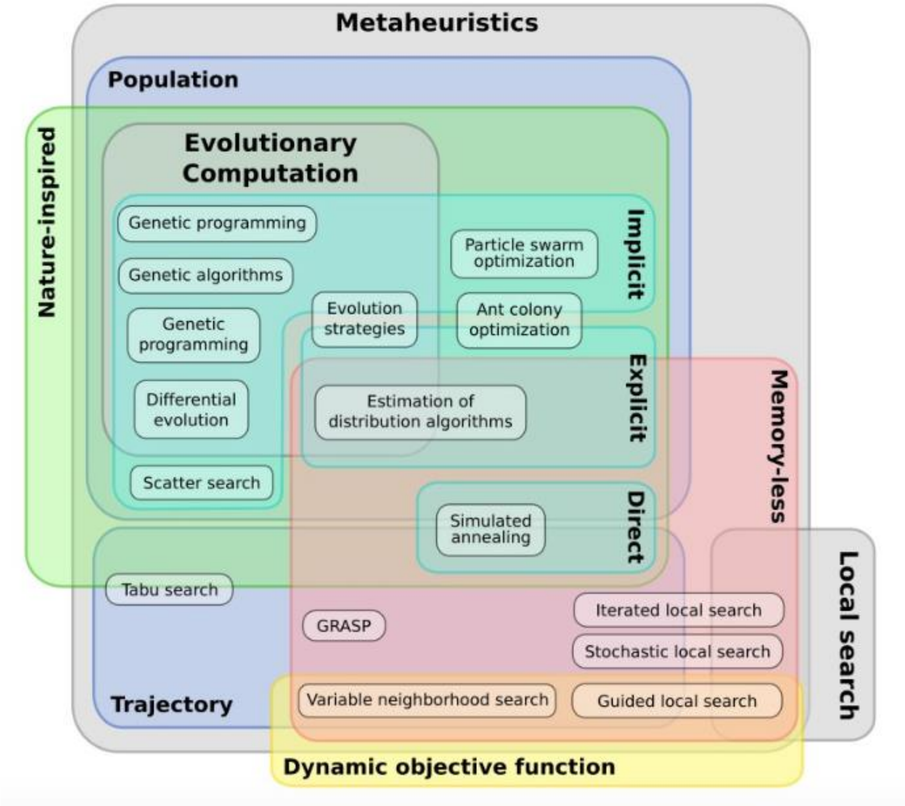


Figure 3.1: Classification of Metaheuristics (Dreo, 2017)

3.2 Genetic Algorithms

3.2.1 Review

The Genetic Algorithms are a kind of algorithms that imitate the behaviour of the biological evolution. This technique was first introduced in 1975 by John H. Holland in his Adaptation in Natural and Artificial Systems (Holland, 1992), where he links a nonlinear mathematical model with the biological evolution. In this work, he settles the application of this relationship to so many diverse fields as economy, psychology, game theory or artificial intelligence, among others. The aim of a GA is to find the best solution in an optimisation problem. To do so, a collection of individuals is established, where each one represents a possible solution. The population is extended creating new individuals using the previous ones, and parents and sons receive a value (fitness function). A reduced number of individuals are selected among all the population, and we repeat the procedure until some convergence criterion is reached. The keystone of this algorithm is the way we create the sons. For that purpose, two functions must be established: crossing and mutation. Each individual has to be codified with a set of parameters (genes) that we call chromosome. Thus, each possible chromosome represents a possible solution of our problem. So, the crossover function gives another chromosome from two previous ones. If nothing more is done, after many applications of the crossover function over the population, we would have the best possible solution made by the combinations of the genes coming from the first generation of individuals. In that case, we do not have any guarantee that we are under a path that drives us to the optimum. In fact, we will have the best individual made from the finite combination of the initial genes. To avoid this situation, another function is needed: the mutation. Mutation gives us the opportunity of creating new individuals in a genetic sense. In other words, although we didn't have initially the necessary genes to get the optimum, we would have the opportunity to get them in the future thanks to the mutation (creation of new genes). We show the scheme of a standard GA in the figure 3.2:

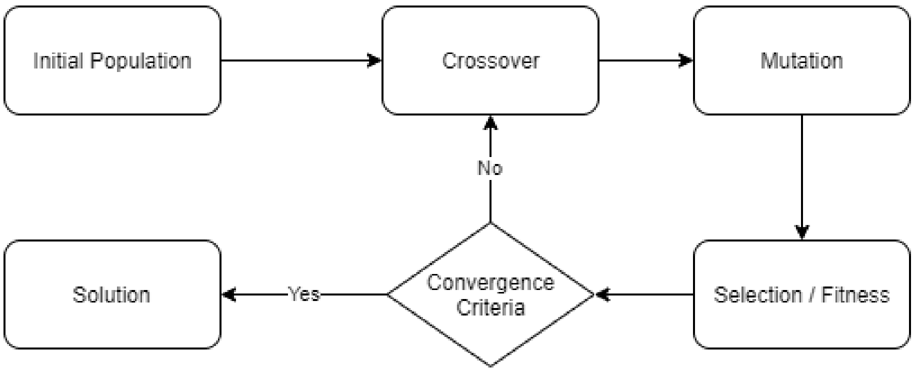


Figure 3.2: Diagram of a Genetic Algorithm

The GA is useful in those problems where we cannot have the derivative of the objective function[1]. When we have some local optimums, the GA does not give us the confidence of getting the best solution but only a good one, i.e., a local optimum. Nevertheless, if the size of the population is sufficiently large, the probability of reaching the best possible solution increases.

[1]Recall that if we can derive the objective function, we can calculate the gradient, and we are able to apply the gradient-based optimisation without the need to construct a population, giving us a way to obtain an local optimum with extraordinary accuracy and speed.

The first step in this type of algorithms is the creation of an initial population. This step is essential because not any random selection of individuals gives us an evolutionary feasible population. That is, we could have a random set of individuals that, once we cross among them and we proceed to mutations, no son is better than his parents. This can be possible depending on how we define our constraints and thus, our feasible solutions. If our domain is not convex, we can fall into the mistake of selecting individuals that recombined linearly fall outside the domain.

The crossover function is strongly dependent on how we select the genes that define our solution. In general, this selection is not unique. We can find several ways to parameterize the solution but, in order to be used in an evolutionary algorithm, we should think about designing the genes so that each one represents some trait clearly identified. In this way, when we interchange the genes to build a new individual, this one will inherit those traits or features from his parents.

There are many ways to specify the crossover function, but the most common are One-Point Crossover, Two-Point Crossover and Uniform Crossover. Let's imagine that our representative chromosome is a binary vector. If we select a specific coordinate or position into the list of gens, and we swap all the coordinates (gens) between two chromosomes before (or after) this position, we have the case of One-Point Crossover. In figure 3.3 an example is illustrated. The Uniform Crossover consists of swapping or not each genes depending on a probability associated to each one. As we will see in the chapter 8, a variant of One-Point Crossover is used in our GA. We select a random position each time we cross two individuals and we swap the gens before that position.

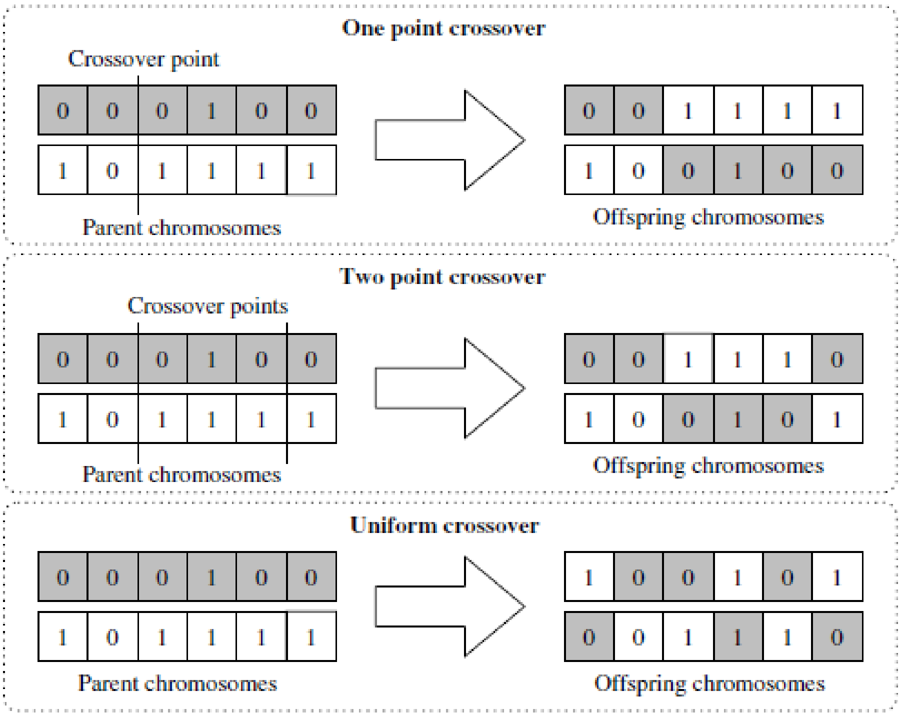


Figure 3.3: Crossover Function in Genetic Algorithm

As we have mentioned before, the crossover function is not sufficient by itself. Let's suppose that our individual can be represented by a vector of natural numbers between 0 to 10. When we select our primitive population of size N, we select random numbers to create all the N chromosomes. Effectively, the first crosses create new individuals, but the number of possible individuals is conditioned by the primitive selection. For instance, if we don't have the number 7 in position 1, we will not be able to create an individual with number 7 in the first position since the possible swaps won't have that possibility. So, an extra mechanism is necessary to extend the traits of our individuals. That process is the mutation. Mutation is absolutely necessary if we want to have variety, and therefore, to have the opportunity of having the traits that belong to the best solution. Only combining both crossover and mutation, we have the chance of having the optimum.

3.2.2 Methodological Contribution

The last chapters of this thesis are dedicated to how we can apply the genetic algorithms to the main topic of this research, the ALM. The problem we solve is completely detailed in chapter 8, but for the purpose of this section we will rewrite it in a more general notation:

$$Maximise\ U(x_{ij}, \pi_j) \tag{3.1}$$

$U(x_{i,j})$ is a nonlinear function of a matrix of unknowns $x_{i,j}$. The index i identifies the money of a transaction of the asset i and j is the time where the transaction is done. The vector π_j has to be with the interest rate in each period of time j and is also a function of x_{ij}. In our problem, we suppose that this interest rate can have only two possible values depending on x_{ij}.

$$\pi_j = \pi_j(x_{ij}) \tag{3.2}$$

This optimisation problem is followed by a considerable list of constraints that can be summarised as:

$$\sum_i C_{ki} x_{ij} \geq c_k \tag{3.3}$$

where the coefficients C_{ki} are also dependent on π_j.

The size of this problem can be enormous since we have to consider $i \times j$ unknowns, and at least, we consider that U is quadratic, so we have an objective function with not less than $i^2 \times j^2$ terms.

This system has two main difficulties, apart from the size of the problem. On the one hand, we have to tackle with many local optimum solutions and near one each other, and on the other hand, the parameter π_j swaps from one value to another so it converts our objective function in non derivable.

The approach based on a genetic algorithm gives us the chance of searching better solutions starting from a trivial one. In this case, if we follow the schema we commented before, we will have a good solution as it is our purpose. But in this kind of problems, in which time is a relevant player, we cannot be satisfied with just a good solution: time imposes on us an order and that solution can be considered bad if it doesn't respect the logic of the time. For instance, let's imagine that we have local continuity in a period p. In this case, we should impose smoothness to the variables x_{ip}. If we visualise this effect in terms of investment, we could have an investment of +3 in period $p-1$, -2 in period p and +2 in period $p+1$ which is not tolerated by any investor since the last two transactions are meaningless.

To correct that, we have applied the concept of civilisation. One civilisation is the population we have as a result of a complete converged cycle of a genetic algorithm. After we get the convergence, what we have to do is to mutate all the individuals of the population and so, we have a new population. Obviously, the new population is worse than the previous one, but we have jumped to another place in our domain. This new population has also to evolve following the GA and we will stop when a new convergence criterion is reached (see fig 3.4). In our experience, this process improves the results not only in the mere value of the objective function but also in the logic of the time. If we measure the dispersion D as the sum of the absolute deviation from a transaction to the next one, i.e.: $D = \sum \mid T_i - T_{i+1} \mid$, we can see that the process of civilisations converges to a minimum dispersion.

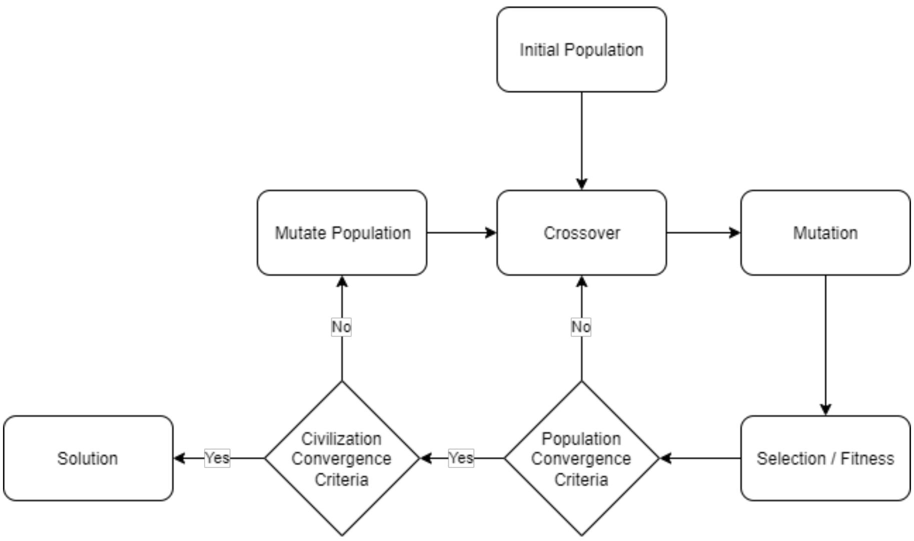


Figure 3.4: Chain of Civilisations in a Genetic Algorithm

Definitely, the process of concatenating civilisations improves notably our result and reduces the dispersion. To the best of our understanding, this is a novel procedure in the practice of Genetic Algorithms, that is very useful when we treat with solutions that represent paths along the time.

3.3 Montecarlo Simulation

3.3.1 Review

In Science, we usually try to describe the behaviour of a system through a mathematical model. In several cases, we need to consider many features because they are relevant for the whole of the system and impact in its evolution. In such situations, it is not uncommon that the solution of the model is unreachable and different alternatives have to be taken. If the complexity of the equations is high, one of those alternatives is Montecarlo simulation. This technique consists of either reproducing all the elements in a vast recreation using computation or creating a relevant number of parallel and identical systems and evolve them independently, to gather statistically the final results, if some of the features follows some known stochastic behaviour.

The system is formed by elements and these elements have attributes whose evolution is described by the specific model. We can differentiate the models, among other classifications, in:

a) deterministic model

b) stochastic model

The first kind of models have deterministic laws that drive the state of the elements along the time. One example could be a planetary system, since the physical laws are precise, deterministic and well known. In such a case we compute the evolution of the elements that conform our system. The second kind of models are ruled by probabilistic functions. So, we cannot proceed as in the deterministic case because each time we run the simulation, we get a different result. So, we have to create many instances of the problem and run the simulation for each one. Ultimately, we must gather all the results to analyse them in statistic terms.

Along this research, we will implement stochastic models because our elements represent stock prices of a financial market, whose nature is purely stochastic.

So, if we have a system S in a specific initial state S_0 described by a set of $\vec{N}_0$ parameters, the simulation will lead to a final state S_1 where those parameters have changed to $\vec{N}_1$ according to the stochastic rules of our model. Having the final state, we can calculate any required value X related to the parameters of the final state S_1. Repeating this simulation, a specific amount of times, we get an estimator of the variable X. Following the Central Limit Theorem, we can state that the mean estimator of X follows a normal distribution with mean $\bar{X}$ and standard deviation $\sigma/\sqrt{n}$, where σ is an unknown standard deviation that comes from our specific stochastic model and n is the number of simulations. If we take a lot of executions, the standard deviation of this estimator tends to zero, so the mean can be determined with accuracy.

3.3.2 Generation of Normal Random Numbers

The simulations we have implemented along this research need the generation of normal random numbers. There is not only one procedure to do this, and in fact, we don't have a library that, by default, gives us that functionality, in the programming language we have used, c#. The computers give us the possibility of getting sequences of pseudo-random numbers in the interval $]0,1[$. So, we have to implement a procedure that gives us normal random numbers having a uniform probability distribution. For that, we have followed Box and Muller (1958). This method states that if U_1 and U_2 are independent random variables that follow a uniform distribution, the magnitudes N_1 and N_2 defined as:

$$\begin{aligned} N_1 &= \sqrt{-2\ln U_1}\cos(2\pi U_2) \\ N_2 &= \sqrt{-2\ln U_1}\sin(2\pi U_2) \end{aligned} \tag{3.4}$$

are independent random variables following standard normal distributions. This method is fast although it has a bias because of the ln. Indeed, when we work with a 64-bit microprocessor, we will be able to generate a number greater or equal than 2^{-64}. At most, we will have $\sqrt{-2\ln(2^{-64})} = 9.42$. So, we won't have numbers greater than 9.49 times the standard deviation. Nevertheless, this is not so dramatic since the cumulative probability we loose is $2 \times (1 - \Phi(9.41)) \approx 5 \times 10^{-21}$.

When we need to generate correlated random numbers, we use the Cholesky decomposition (Higham, 2009). It is a method to rewrite a positive semidefinite matrix Σ in this form: $\Sigma = LL^T$ where L is a lower triangular matrix. Having that, if the vector Z is a vector of independent normal random numbers, the vector $X = LZ$ is a vector of correlated normal numbers with Variance-Covariance Σ.

3.4 Simheuristics

As analysed optimisation problems become more complex, an increasing number of COPs is also characterised by uncertainty. Simheuristics enhance a metaheuristic framework that yields good solutions for deterministic optimisation by including simulation in order to account for stochastic uncertainty that is present in many real-life considerations (Juan et al., 2015a). This uncertainty can be modelled in the objective function or in the constraints. It is to be noted that the presence of extreme volatility in either should generally not be modelled employing optimisation techniques because of the diversity of individual outcomes (Juan et al., 2015a). It is thus assumed that the methodology that yields a high-quality solution for a deterministic formulation of a problem can also yield a high-quality solution when reasonable levels of uncertainty are incorporated through simulation (Juan et al., 2015a). This assumption justifies the following methodological approach 3.5. The deterministic version of the COP is considered by eliminating uncertainty and the metaheuristic is run to determine promising solutions that are consequently sent to the simulation process, during which only several iterations are run at this stage to rank the solution in order to sufficiently explore the search space and identify promising search areas (Juan et al., 2015a). After a stopping criterion is reached intensive simulation can be performed on a reduced set of solutions. Although the proposed methodology is relatively new, potential applications benefit from the extensive research on the application of metaheuristics to deterministic COPs and thus from their simplicity and efficiency, while enhancing their capability to solve more realistic real-life COPs.

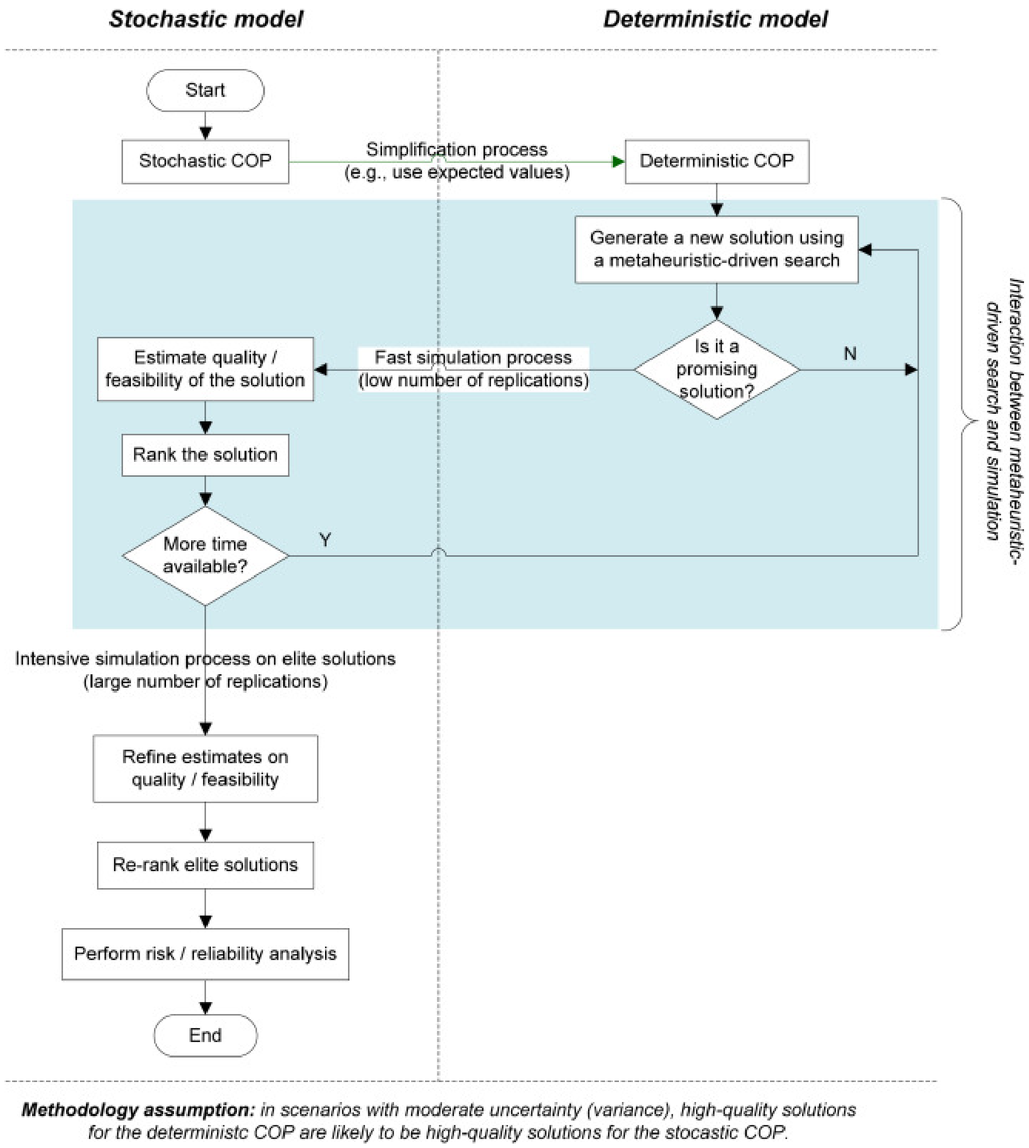


Figure 3.5: General Structure of Simheuristic Approaches (Juan et al., 2015a)

Chapter 4

ALM in Financial Markets: State of the Art

Abstract

Most financial organisations depend on their ability to match the assets and liabilities they hold. This managerial challenge has been traditionally modelled as a series of optimisation problems, which have been mostly solved by using exact methods such as mathematical and stochastic programming. The chapter reviews the main works in this area, with a special focus on three different problems: duration immunisation, multi-stage stochastic programming, and dynamic stochastic control. Hence, the main results obtained so far are analysed, and the open challenges and limitations of the current methods are identified. To deal with these open challenges, we propose the incorporation of new heuristic-based algorithms and simulation-optimisation methods.

4.1 Introduction

All financial companies need to manage the risk associated with their liabilities. This is achieved by properly selecting a convenient set of assets from the market, which are then assigned to cover liabilities, thus reducing the risk of bankruptcy. However, both assets and liabilities are exposed to an innumerable number of external factors, which need to be factored in order to maintain and update the allocation map between assets and liabilities. The asset and liability management (ALM) challenge refers to the set of methods and techniques used to identify those assets that offer an optimal match with a set of given liabilities. ALM can be seen as an optimisation problem: the financial institution has to establish a particular strategy, which gives rise to an objective function subject to a set of constraints. The optimisation problem typically maximises the company's value function, it minimises the price of the selected assets, it maximises the expiration value or terminal wealth or combines several aforementioned objectives.

The management of assets and liabilities is of paramount importance for financial institutions, such as banks, insurance companies, and pension funds. Although all of them are part of the financial system, they differ in terms of the nature of their liabilities. Accordingly, the strategy of selecting the adequate assets to match their liabilities also varies across different financial institutions. Among the different types of institutions, banks take deposits as their main liability. These deposits vary over time. Insurance companies also have time-varying liabilities, which are derived from insurance policies they underwrite. A portfolio of an insurance company tends to be large in order to benefit from the law of large numbers. Pension funds project their liabilities into the future, when the individual is expected to retire. Due to the time consideration, the role of an interest rate becomes relevant in the ALM process. It is also essential to model the stochastic behaviour of the random variables in the optimisation problem, i.e.: liabilities, assets, interest rates, and/or inflation. Due to the stochastic and dynamic nature of assets and liabilities, it is reasonable to assume that the initial asset selection might need to be updated throughout time, as new information becomes available, so the match between assets and liabilities is re-optimised taking into account the new data. Thus, the financial institution's assets are re-balanced in each period by selling and buying asset shares in order to benefit from portfolio returns. These considerations lead us to three main techniques in ALM. Firstly, the duration theory, which aims to define an immunisation strategy so that the value associated with the portfolio of assets matches, at any

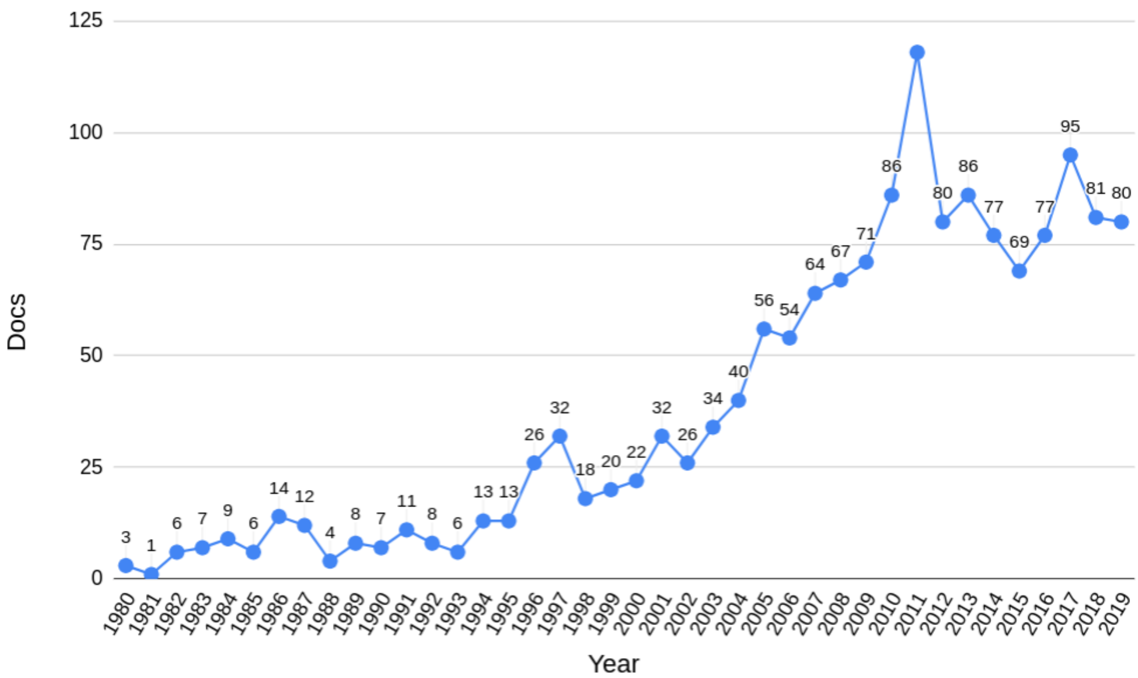


Figure 4.1: Evolution of Scopus-Indexed Documents Related to ALM

time, the value of the liabilities. Hence, a change in the interest rate will not affect the balance. Secondly, one can consider a single-period version of the problem or a multi-period one, in which the optimal asset selection is determined at each stage. Also, the problem can be deterministic – by simplifying some characteristics – or stochastic in nature. In the latter case, we have to provide a stochastic process for each asset, which might include random variables in the objective function and even probabilistic constraints. Thirdly, if time is regarded as continuous, the problem becomes a stochastic control optimisation one, which gives rise to a system of differential equations. As shown in Fig. 4.1, the interest of the scientific community in asset and liability management (ALM) has been increasing during the last decades.

The contribution of this chapter is threefold. Firstly, it reviews the main works in this area, with a special focus on three different problems: duration immunisation, multi-stage stochastic programming, and dynamic stochastic control. Secondly, the main results obtained so far are analysed, and the open challenges and limitations of the current methods are identified. Thirdly, the incorporation of new heuristic-based algorithms and simulation-optimisation methods is proposed in order to deal with these open challenges. The rest of the chapter is structured as follows: Section 4.2 provides a review of existing work on duration immunisation. Section 4.3 analyses applications of stochastic programming to ALM. Section 4.4 completes a review on stochastic control applied to ALM. Section 4.5 discusses the need for considering new simulation-optimisation approaches in dealing with these problems. Finally, Section 4.6 highlights the main conclusions of this work and propose some open research lines.

4.2 Duration Immunisation

Under the assumption of deterministic cash-flows on both sides, assets and liabilities, and constant interest rates, Macaulay Macaulay (1938) sought to devise a strategy for matching values of assets and liabilities. The present value (PV) of a fixed cash-flow (CF), recorded at times $t \in \{0, 1, \ldots, T\}$, and with a constant interest rate i, is commonly defined as:

$$PV = \sum_{t=0}^{T} \frac{CF_t}{(1+i)^t}. \tag{4.1}$$

If the goal is to provide immunisation against variations in the interest rate, we need to compute the derivative of the present value with respect to the interest rate i:

$$\frac{1}{PV}\frac{dPV}{di} = -\frac{D}{1+i}, \tag{4.2}$$

where D is called the Macaulay Duration and is described as:

$$D = \frac{\sum_{t=0}^{T} t \cdot CF_t (1+i)^{-t}}{PV}. \tag{4.3}$$

The immunisation in this approach consists in selecting a set of assets that satisfy two conditions: *(i)* the present value of the assets matches the one of the liabilities; and *(ii)* the time duration of assets also matches the one of liabilities. Under these conditions, it is possible to conceive that 'slight' changes in the interest rate will not have a noticeable effect on the values of assets and liabilities. If more pronounced changes in the interest rate are expected, then it might be necessary to add a third condition, the so-called convexity requirement, which corresponds to the second derivative of the price with regard to a change in the interest rate change in a

Taylor's series. This approach is clearly focused on potential changes in the interest rate, and constitutes the first ALM strategy analysed in the scientific literature. Later, Hicks (1975) introduce the term "corrected duration", to justify variances in the present value when the interest rate changes. These authors measure duration as a percentage, while the previous one measured duration in terms of time.

The first works about immunisation where formalised by Fisher and Weil Fisher and Weil (1971), who defined the conditions under which the value of an investment in a bond portfolio is protected against changes in the level of interest rates. The hypotheses of this work are: i)the portfolio is valued at a fixed horizon date, and ii) the interest rate changes only by a parallel shift in the forward rates. Fong and Vasicek Fong and Vasicek (1984) consider a fixed income portfolio whose duration is equal to the length of a given investment horizon. They prove that, given a change in the term structure of interest rates, there is a lower limit to the value of the portfolio. This lower limit depends on two factors: the interest rate change and the structure of the portfolio. Consequently, they postulate that it is possible to optimise the exposure of the portfolio under interest rate changes. Bierwag et al. Bierwag et al. (1993) study the properties of cash-flow dispersion in duration hedged portfolios. They show that minimising this dispersion is not independent of stochastic processes, and that the optimisation of the immunisation by minimising cash-flow dispersion is only valid under specific convexity conditions. Zenios Zenios (1995) highlights a frequent presence of a mismatch between assets and liabilities in the financial industry, and shows a complex case of portfolios containing mortgage-backed securities under the term structure volatility. Among others, techniques based on duration are explored by this author. Seshadri et al. Seshadri et al. (1999) embed a quadratic optimiser in a simulation model, which is used to generate patterns of dividends and market values, thus computing the duration of capital. This method is used to refine the ALM strategy, and is applied to the Federal Home Loan Bank of New York. Gajek Gajek (2005) introduces the requirement of 'solvency' for a defined benefit pension plan, i.e., under a scenario with a relatively low interest rate, the assets are chosen to be the smallest concave majorant of the accumulated liability cash-flow. Ahlgrim et al. Ahlgrim et al. (2004) study the risk for property-liability insurers of movements in interest rates. Their paper considers that liability cash-flows, affected by future claim payouts, change with interest rate shifts due to the correlation between inflation and the interest rate. This study concludes that the effective duration is lower than the one measured by traditional methods. Benkato et al. Benkato et al. (2011) analyse the case of eight banks in Kuwait, showing that this sample of banks adjusted their portfolio of assets and liabilities by matching their respective Macaulay's duration.

4.3 Multi-Stage Stochastic Programming

The allocation of assets in an ALM context is carried out at specific times. When the manager performs a transaction, she has to cope with transaction costs, asset values that are dependent on the moment and liquidity constraints, among other variables. The main goal is to meet the liabilities, but other objectives can be selected simultaneously, e.g.: maximising the terminal wealth of the company, minimising the risk in terms of volatility, etc. As financial markets run in scenarios under uncertainty, the problem can be regarded as a multi-stage stochastic program (Kouwenberg and Zenios (2008a)). Multi-stage stochastic optimisation problems refer to situations in which decisions need to be made at different periods over a planning horizon and under uncertainty conditions. Typically, at each new period, recourse actions can also be considered to account for the updated information (Pflug and Pichler, 2016). Numerous approaches have been studied in the literature, but all of them share a common structure. On the side of the constraints, two basic sets of equations are defined: the cash-flow accounting and the inventory balance equations at each time point. At this point, the volume of each asset class and its values are recorded, together with information on the number of assets that are purchased and sold. On the objective function side, the common goal is to maximise expected utility subject to terminal wealth. In order to solve the optimisation problem, a scenario tree has to be defined. This represents a lattice of possibilities for each asset, liability, and other elements in the program, including interest rate, inflation, among others. Each node is associated with a probability, and the whole lattice needs to be considered to calculate the expected values. In this regard, Boender et al. (2008) study the role of scenarios in ALM, as a lattice of possibilities for each element in the model, each one with an associated probability. Mulvey et al. (1997) consider a multi-stage stochastic program and assign a probability to each scenario. Any considered scenario ends with a terminal wealth, and contemplates potential purchases or sales of assets in each period. In addition, it is usual to consider a vault cash (security stock of cash). Whenever the liabilities cannot be covered with the existing assets, credit has to be obtained –notice that this should be a last resource, since it will typically be associated with a high interest rate. Following Mulvey et al., we can synthesise the multi-stage stochastic program as follows:

$$Max \sum_{s=1}^{S} \pi_s U(w_\tau^s) \tag{4.4}$$

subject to:

$$\sum_i x_{i,0}^s = w_0, \tag{4.5}$$

$$\sum_i x_{i,\tau}^s = w_\tau, \tag{4.6}$$

$$x_{j,t}^s = (1 + \rho_{j,t-1}^s) x_{j,t-1}^s + p_{j,t}^s - d_{j,t}^s, \tag{4.7}$$

$$x_{0,t}^s = (1 + \rho_{0,t-1}^s) x_{0,t-1}^s + \Sigma_j d_{j,t}^s - \Sigma_j p_{j,t}^s - b_{t-1}^s (1 + \beta_{t-1}^s) + b_t^s, \tag{4.8}$$

where s represents one possible scenario, π_s is the probability of scenario s, w^s is the terminal wealth in scenario s, A is the number of assets, $i \in \{0, 1, \ldots, A\}$, $j \in \{1, 2, \ldots, A\}$, x_j is the amount of money invested in asset i, x_0 is the vault cash, $p_{j,t}^s$ is the purchase of asset j in time t in scenario s, $d_{j,t}^s$ is the amount of asset j sold in time t in scenario s, $\rho_{j,t}^s$ is the yield of asset j in time t in scenario s, $\rho_{0,t}^s$ is the riskless interest rate, b_t^s is the amount of money borrowed in time t in scenario s, and β_t^s is the borrowing rate in time t in scenario s. Finally:

$$x_{0,t}^s \geq l_t, \tag{4.9}$$

where l_t is the liability cash-flow at time t.

Numerous works have been searching for a better and more realistic description of the financial system. Hence, Kusy and Ziemba Kusy and Ziemba (1986) study a model with legal, financial, and bank-related policy considerations. They apply the model to a 5-year period for a Canadian bank. Giokas and Vassiloglou Giokas and Vassiloglou (1991) discuss a multi-objective programming model for the Commercial Bank of Greece, taking into account institutional characteristics, financial, legal, and bank-related policy considerations. Oğuzsoy and Güven Oğuzsoy and Güven (1997) present a multi-period stochastic linear model for ALM in banking, assuming a set of deterministic rates of return on investment and cost of borrowing. They also consider a set of random deposit levels, liquidity, and total reserve requirements. Mulvey et al. Mulvey et al. (2000, 1997) show how the Towers Perrin company plans assets and liabilities to deal with pension-related payments. The model performs an economic projection, spanning a long-term horizon (10 to 40 years), and finding strategies via a dynamic assets and liabilities allocation over a range of different scenarios. Nielsen and Zenios Nielsen and Zenios (1996) study how to apply a multi-period stochastic program to the problem of funding single-premium deferred annuities, for which they consider government bonds, mortgage-backed securities and derivative products. Klaassen Klaassen (1997) shows that, in general, scenarios do not consider the variation over time of some asset prices. Therefore, the solution found by stochastic programming cannot be considered as optimal in a real-world application, where uncertainty has to be considered. The paper remarks the crucial importance of respecting the free of arbitrage hypothesis while defining scenarios. Consiglio et al. Consigli and Dempster (1998) develop a pension fund problem, in which uncertainty affects both assets and liabilities in the form of scenario-dependent payments or borrowing costs. Cariño et al. Carino et al. (1994); Cariño et al. (1998); Cariño and Ziemba (1998) describe the Russell-Yasuda Kasai model. This model, created by the Russell company and the Yasuda Fire and Marine Insurance Co., determines an optimal strategy in a multi-period scenario, and it adds the characteristics of the complex Japanese regulation, such as legal or taxes limitations. In their first publication, Carino et al. (1994) compare the multistage programming model with the classical mean-variance model, resulting in an extra income of 42 basis points. Kouwenberg Kouwenberg (2001) develops a scenario-generation method and applies it to a multi-stage stochastic program for a Dutch pension fund, where the objective function consists of minimising the average of contribution rates, taking into

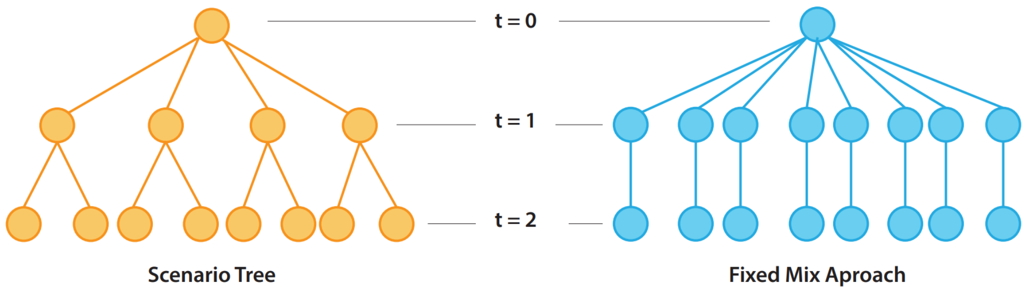


Figure 4.2: Differences between Scenario Trees and Fixed Mix Approaches. Source: Fleten et al. (2002)

account the degree of risk aversion. The scenario-tree model is compared to a fixed mix model as shown in Fig. 4.2.

As sophisticated scenarios are generated in combination with many trading dates, the number of variables in the mathematical programming model tends to explode. Gondzio and Kouwenberg Gondzio and Kouwenberg (2001) deal with the computational complexity of this problem, identifying a bottleneck in memory management. They combine decomposition methods and high-performance computing to cope with large-scale instances of the problem, solving a stochastic problem with near 5 million scenarios, more than 12 million constraints, and 25 million variables to study a pension fund. Gondzio and Grothey Gondzio and Grothey (2006) also solve non-linear programming models using an interior point solver and a massive parallelisation environment. Bogentoft et al. Bogentoft et al. (2001) study the effects of the conditional value at risk (CVaR) as a risk measure, the weighted average of the value at risk (VaR), and the losses exceeding the VaR. They also select similar paths in the scenario creation, simplifying the problem to representative samples. With this technique, they are able to solve problems with a very large number of elements and scenarios. Høyland and Wallace Høyland and Wallace (2001) show that regulation in Norway is not beneficial for the insurance industry, according to the results of a simple stochastic problem that integrate legal issues. Fleten et al. Fleten et al. (2002) compare a fixed mix model with a multi-stage stochastic program (dynamic model). The fixed mix model keeps constant the proportion among the assets, while the dynamic model changes the proportion in each stage. The conclusion is that the dynamic model dominates the fixed mix approach. Dash and Kajiji Dash and Kajiji

(2005) implement a non-linear model based on the Markowitz's mean-variance approach Markowitz (1952) for the optimisation of property-liability insurers. Hibiki Hibiki (2006) compares the results of two different approaches, which model the evolution of assets both using a scenario tree and a hybrid tree (simulation paths). (Dempster et al., 2003) combine dynamic stochastic optimisation with Montecarlo simulation to analyse an ALM problem involving global asset classes and contribution pension plans. Arguably, their approach can also be used to manage financial planning problems related to insurance firms, risk capital allocation, and corporate investment, among others. Additional applications and case studies on ALM can be found in (Zenios and Ziemba, 2007). Also, (Kouwenberg and Zenios, 2008b) review stochastic programming models for ALM. Among other issues, they analyse the performance of these models when applied to pension funds, discussing both their advantages and limitations. Zhang and Zhang Zhang and Zhang (2009) improve Hibiki's model by introducing the CVaR as a risk measure, and market imperfections. A genetic algorithm is used to solve the new model. Consiglio et al. (2006) and Consiglio et al. (2008) study the optimisation problem derived from a liability that includes complex conditions, such as guarantees, surrender options, and bonus provisions. This leads to a non-linear optimisation problem. Papi and Sbaraglia Papi and Sbaraglia (2006) solve a problem with two assets, where one of the assets is risky, and the other risk-free. They use a recourse algorithm. Rosmarin (2008) studies the use of different evolutionary algorithms applied to multi-stage stochastic models. The author generates a set of scenarios using Montecarlo, and solves the problem by applying a multi-objective evolutionary algorithm. Ferstl and Weissensteine Ferstl and Weissensteiner (2011) analyse a multi-stage stochastic program under time-varying investment opportunities, where the asset return follows an auto-regressive process. To minimise the conditional value-at-risk of shareholder value, the authors utilise stochastic linear programming and a decomposition of the benefits in dynamic re-allocation. In general, the models do not place limits on the number of assets, which might be a quite unrealistic assumption in practice. Examples of specialised books dedicated to ALM are (Bauer et al., 2006), (Adam, 2008), (Mitra and Schwaiger, 2011), and (Choudhry, 2011).Nevertheless, Escudero et al. Escudero et al. (2009) propose an approach based on discrete variables, limiting the number of transactions or the number of assets in each time. The model is solved with a recourse algorithm. Berkelaar and Kouwenberg (2010) introduce a singular objective function, which consists of a liability-relative draw-down optimisation approach. Both assets and liabilities are modelled as auto-regressive processes. Corsaro et al. (2010) discuss a valuation system of portfolios of life insurance policies for stochastic ALM models. They use a parallelised Montecarlo method combined with the forward risk-neutral measure to speed-up the simulation process. Gülpinar and Pachamanova (2013) and Gülpınar et al. (2016) treat the problem under the robust optimisation perspective, deriving in a feasible computational tractability. These approaches deal with uncertainty in both assets and interest rates, and are focused on investment products with guarantees, such as guaranteed investment contracts and equity-linked notes. These authors perform a series of computational studies with real market data in order to compare the performance of their approach to that of classical stochastic programming. Several other approaches to the ALM problem have been studied recently. Thus, Zhang and Chen (2016) focus on the mean-variance ALM with constant elasticity of variance. Wei and Wang (2017) focus on random coefficients, while Li et al. (2018b) study models with stochastic volatility. (Fernández et al., 2018) introduce a stochastic ALM model for a life insurance company. They use GPUs to run Montecarlo simulations in parallel. (Dutta et al., 2019) employ big data analytics and stochastic linear programming in ALM under uncertain scenarios. The authors study the relevance of employing a large number of scenarios in solving the stochastic ALM problem. Finally, (Li et al., 2019) use a multi-period mean-variance model to analyse an ALM problem with probability constraints. In their model, investors seek to control for the probability of bankruptcy, while the process is influenced by uncertainty in the cash-flows. In the Iranian regulation framework, Abdollahi (2020) studies a multi-objective ALM programming problem where the constraints are realistic legal conditions of the banking industry. Within a dynamic stochastic control approach, Sun et al. (2019) studies a mean-variance ALM problem where assets and liabilities are both stochastic, and where liabilities transfer part of their risk by means of a reinsurance.Zhou et al. Zhou et al. (2019) construct a program based on the classical mean-variance efficient frontier. Their approach considers quadratic transaction costs. They propose tractability models with and without the risk-less asset, and derive the pre-commitment and time-consistent investment strategies through the application of an embedding scheme and a backward induction approach. Movahed and Fanian (2019) solve a portfolio optimisation problem with realistic constraints using a GA, where the fitness function is obtained through a Montecarlo simulation process. Similarly, Solares et al. (2019) solve a portfolio optimisation problem using a GA, where the fitness function is represented as a confidence interval in order to model uncertainty. The confidence interval is obtained applying Montecarlo simulation to estimate the expected asset return. Orlova (2019) develops an algorithm to solve a discrete dynamic process for cash distribution, in which the goal is to minimise the payment of fines for non-timely financing of expenses. This approach solves the problem of financial resources distribution under uncertainty over time. Kopa and Rusý (2020) formulates a complete stochastic program for ALM credit institutions that grant loans to general customers. In this paper, stochastic multi-stage scenarios are considered and the behaviour of the consumer are modelled. This behaviour impacts on the decisions the credit institution has to take and how it has to allocate its assets.

4.4 Dynamic Stochastic Control

The random behaviour of assets and other market elements, such as interest rate or inflation, are frequently modelled as a geometric Brownian process in a continuous time context. ALM is not an exception. Thus, it is possible to consider a stochastic objective function that also incorporates dynamic equations regulating changes in market elements. Following Chiu and Li Chiu and Li (2006), we consider $n + 1$ assets, where asset 0 is considered to be riskless while the other n assets follow a random walk. The dynamic equation for the price P_0 of a risk-less asset can be written as:

$$dP_0 = P_0 \alpha_0(t) dt, \tag{4.10}$$

where $P_0(0) > 0$ and $\alpha_0(t)$ is the free risk interest rate.

Since the asset is deemed risk-less, no random component is needed in the dynamic equation. By contrast, for the price P_i of a risky asset i, the following equations are used:

$$dP_i = P_i(\alpha_i(t)dt + \sum_{j=1}^{n} \sigma_{ij}(t)dW_j(t)), \tag{4.11}$$

where $i \in \{1, 2, \ldots, n\}$, $P_0(0) > 0$, and $W_1(t), \ldots, W_n(t)$ are independent Wiener processes. Also, $\alpha_i(t)$ represents the interest rate for asset i, while σ_{ij} is the covariance matrix of assets. A typical mean-variance optimisation problem, maximising the terminal wealth, is shown below:

$$Max_{u(t)}E[S(T)] \tag{4.12}$$

subject to

$$Var[S(T)] < \sigma, \tag{4.13}$$

where $u(t) = (u_0, u_1, \ldots, u_n)(t)$ is the amount of money invested in each asset, $S(T) = \sum_{i=0}^{n} P_i(T) - L(T)$ is the terminal wealth, $L(T)$ is the terminal value of the liabilities, and σ is the user-defined threshold for the tolerated risk. Of course, it is also possible to consider other objectives based on a specific utility function.

Several authors have proposed different approaches relative to this basic model. Thus, for example, Chiu and Li Chiu and Li (2006) assume uncertain liabilities, which follow a Wiener process that is correlated with the assets. Devolder et al. Devolder et al. (2003) solve a defined contribution pension problem where the benefits are paid as annuities. To find an analytical solution, they consider one risky asset and one risk-less asset. The paper shows how the strategy changes immediately before and immediately after the beginning of an annuity, and depending on the utility functions as objective functions. Briys and De Varenne Briys and De Varenne (1994) study a profit-sharing policy in an insurance company. With this policy, the policyholder has the right to receive a guaranteed interest rate and a percentage of the company's revenues. The results are used to evaluate different aspects of regulatory measures that are frequently encountered in life-insurance business, such as rate ceilings, capital ratios, and asset restrictions. Barbarin and Devolder Barbarin and Devolder (2005) develop a model, in which assets are a mix of stocks, bonds, and cash, while liabilities are the result of a guaranteed technical rate to the premium, plus a participation rate in the case of a surplus. The paper integrates a risk-neutral approach with a ruin probability. VaR and CVaR conditions are tested by including an investment guarantee. Koivu et al. Koivu et al. (2005) explore the effects of Finnish regulation within a stochastic model for a pension insurance company. Seven economic factors, pertaining to Finland and the EU, are described as a vector equilibrium correction model. This vector is then used to determine the behaviour of assets and liabilities. Xie et al. Xie et al. (2008) formulate a mean-variance portfolio selection model where assets follow a geometric Brownian motion, while liabilities follow a Brownian motion with a drift. The model also features correlations among assets and liabilities. They derive explicitly the optimal dynamic strategy and the mean-variance efficient frontier by using a general stochastic linear-quadratic control technique. Related to this, Xie Xie (2009) assumes that risk stock prices are governed by a Markov regime-switching geometric Brownian motion. Detemple and Rindisbache Detemple and Rindisbacher (2008) explore a dynamic asset allocation problem with liabilities, where preferences are assumed to be von Neumann Morgenstern Von Neumann et al. (2007), where a running utility function is defined over dividends (withdrawals in excess of net benefit payments), and a terminal utility function defined over liquid wealth in excess of a floor. Chiu and Li Chiu and Li (2009) study how to minimise an upper bound of the ruin probability, which measures the likelihood of the final surplus being less than a given target level. They identify this criterion as the safety-first ALM problem. Not only does the paper study this problem in continuous time, but it also solves the problem in a discrete time context and compares results from the two approaches. The model drives to a mathematical definition regarding the type of investors ('greedy' or not), which is based on the level of disaster. An approach for pension funds can be found in Josa-Fombellida and Ricón-Zapatero Josa-Fombellida and Rincón-Zapatero (2010), who consider a stochastic interest rate, where the investor faces the choice among a risky stock, a bond and cash. Zeng and Li Zeng and Li (2011) analyse a simple but realistic model, which features one risky asset, one risk-free asset, and one liability. The risky asset follows an exponential Levy Process, which allows simulating potential discontinuities in its random walk. The model comprises two objective functions (i.e., it considers two different optimisation problems). The first function is based on a 'benchmark' model: a predefined target value b is considered, and the mean of the quadratic distance between b and the terminal wealth is minimised. The second function is based on the classical mean-variance portfolio selection model. The optimisation problem in Chiu and Wong Chiu and Wong (2012) consists of minimising the variance of terminal wealth in a context of co-integrated assets. Specifically, in this paper, the insurer deals with the payment of uncertain insurance claims, which are assumed to follow a compound Poisson process. In general, there is a lack in the literature regarding the study of time-consistency optimisation of asset allocations in an ALM context – i.e., most studies consider time-dependent investment strategies. This gap is bridged by Wei et al. Wei et al. (2013) considering a Markov regime-switching model. These authors conclude that the time-consistency equilibrium control in this context is state dependent, where that dependency is generated by the uncontrollable liability process. Chiu and Wong Chiu and Wong (2014b) study the problem under a market with correlations among risky assets, where these correlations change randomly over time. In this problem, the objective is to minimise the variance of terminal wealth, given an expected terminal wealth. The liabilities are assumed to follow a compound Poisson process, and the problem becomes a linear-quadratic stochastic optimal control problem with volatility, correlations, and discontinuities – all of them with random behaviour. In a context of low interest rates, the stochastic behaviour becomes relevant. Chiu and Wong Chiu and Wong (2014a) also solve a model with liabilities that follow a compound Poisson process, with a stochastic interest rate distributed according to a Cox-Ingersoll-Ross model Cox et al. (1985). The model consists of maximising the expected constant relative risk averse (CRRA) utility function. Along similar lines, Chang Chang (2015) formulates a model where the interest rate is driven by the Vasicek model Vasicek (1977), and liabilities follow a Brownian motion with drift. Likewise, Liang and Ma Liang and Ma (2015) approach a pension fund with mortality risk and salary risk, with a CRRA utility function. Pan and Xiao Pan and Xiao (2017a) solve a problem with liquidity constraints and stochastic interest rates, which follow a Hull-White process Hull and White (1990). This paper compares the two utility functions that feature CRRA, and constant absolute risk averse, CARA. In another work Pan and Xiao (2017b), these authors also include inflation risk under a mean-variance framework. They also consider a non-common variety of assets, such as a default-free zero coupon bond, an inflation indexed bond, as well as the typical risky assets and risk-free asset. Also, they assume that liability follows a geometric Brownian motion process. Finally, to complete this survey, Li et al. Li et al. (2018a) solve a classical mean-variance model with stochastic volatility, which introduces a novel asset; a derivative whose price depends on the underlying price of the risky stock.

4.5 Need for Metaheuristics & Simheuristics

The growing complexity of the problems being addressed highlights the need for faster approaches such as metaheurisics Glover and Kochenberger (2006). These algorithms will be needed as the models introduce further constraints to account for real-life circumstances. In this regard, Soler-Dominguez et al. Soler-Dominguez et al. (2017) and Doering et al. Doering et al. (2019b) provide quite complete and up-to-date reviews on financial applications of metaheuristics, including risk management and portfolio optimisation problems. In this sense, Kizys et al. Kizys et al. (2019b) have proposed a heuristic approach to solve a *NP-hard* variant of the portfolio optimisation problem. Furthermore, the fact that two or more objectives have to be considered simultaneously to account for the complexity will require multi-objective optimisation methods.

Different simulation-optimisation methods are gaining popularity in the application to stochastic combinatorial optimisation problems in different application areas Juan et al. (2018). Despite the success of simheuristics in solving stochastic optimisation problems in different areas, just a few works have focused on the area of finance. Thus, for example, Panadero et al. Panadero et al. (2018) propose a simheuristic for solving a multi-period project selection problem. Even though financial data is characterised by macro- as well as firm-level uncertainty, to the best of our knowledge, none of the finance-related problems analysed in this work has been addressed with the use of simheuristics so far, which makes this an interesting avenue for future research.

4.6 Conclusions & Future Work

The optimal asset allocation subject to liabilities is a financial problem widely studied in the literature. Broadly speaking, the wealth of strategies can be divided into two categories. The first category is based on immunising the value of the selected assets under changes in the interest rate, which is an accountable approach. The second category is based on cash-flow matching, which is an operational approach. The immunisation approach is based on the concept of duration. The potential of application is very limited because: *(i)* it is a short time approach, and *(ii)* it only works when the interest rate is constant and its shift is small. In addition, the cash-flow matching can bifurcate into continuous time models and discrete time models. The continuous time models show a limitation in the sense that they need to be restrictive analytical models in order to use stochastic differential equations. Hence, it is a method more oriented to knowing the 'good' strategy in qualitative terms, than in obtaining optimal assignment configurations. Finally, the most realistic approach is the multi-stage stochastic program, since it permits to easily model characteristics of the real market. Nevertheless, these models can grow very fast in the number of equations and variables, which eventually make them extremely difficult to solve in short computing times.

Due to the limitations found by exact methods in solving large-scale and stochastic versions of the analysed problems, some research opportunities arise, including: *(i)* the use of heuristic-based algorithms that can provide reasonably good results to complex and large-scale financial problems in short computing times; *(ii)* the introduction of novel simulation-optimisation approaches – other than stochastic programming – that can cope, in a more natural way, with the uncertainty existing in the problems considered in this work; *(iii)* the introduction of the aforementioned methodologies will also allow us to consider richer and more realistic versions of the multi-stage stochastic programming problem; and *(iv)* lastly, we also see a clear opportunity to generalise the use of these optimisation methodologies to support decision making at the level of the individual consumer.

Chapter 5

A Simheuristic Algorithm for Reliable ALM

Abstract

The management of assets and liabilities is of critical importance for insurance companies and banks. Complex decisions need to be made regarding how to assign assets to liabilities in such a way that the overall benefit is maximised over a time horizon. In addition, the risk of not being able to cover the liabilities at any given time must be kept under a certain threshold level. This optimisation challenge is known in the literature as the asset and liability management (ALM) problem. In this work, we propose a biased-randomised (BR) algorithm to solve a deterministic version of the ALM problem. Firstly, we outline a greedy heuristic. Secondly, we transform it into a BR algorithm by employing skewed probability distributions. The BR algorithm is then extended into a simheuristic by incorporating Montecarlo simulation to deal with the stochastic version of the problem.

5.1 Introduction

Financial institutions have to face some critical risk-management processes. Among such processes, asset and liability management (ALM) is of paramount importance due to its potential consequences. ALM consists of a range of techniques necessary to invest adequately, so that the firm's long-term liabilities are met (Ziemba et al., 1998). For an insurance company, a liability constitutes the legal responsibility to repay the insurance contributions that the customer has been making over an agreed length of time, which are increased by the interest rate. This is a typical transaction of pension or life insurance intended to secure retirement income, which gives rise to a three-tier financial problem. First, the insurance company receives the customer's premium. Second, the company invests this premium in the long term, so that the financial benefit envisaged in the insurance policy is secured. Third, in the event of the customer's retirement or death, the insurance company needs to have sufficient funds to meet its liability to the customer. While the aforementioned financial problem unfolds, the insurance company is confronted with a range of risks, which arise either from its role as a financial intermediary or due to complex regulations as well as economic and social policies. If the insurer's obligation to the customer is not honoured, its default becomes a likely scenario. A default can be very costly for the firm, since it can inflict a loss of credibility and reputation. On the one hand, it can face legal action from its creditors. As a result the insurer may be forced to pay hefty fines by the regulatory body. On the other hand, the firm's market share may diminish as its customers may switch to other insurers.

It is thus not surprising that the ALM problem has been widely studied in the literature. As interest rates vary over time, the present value of both assets and liabilities responds to such variation. Consequently, optimal and smart asset management solutions become critical to the insurer, who seeks to ensure that the liabilities can be met at the time when they are required, while at the same time, the value of the firm is maximised. In practical applications, one of most popular solutions to this asset management problem is cash-flow matching (Iyengar and Ma, 2009), whose main objective is to ensure the timely payment of the liabilities. This approach minimises the number of contractual breaches. In some European countries, the legislation does not envisage any specific mechanism to ensure that the firm's obligations are met. Instead, capital is regulated by targeting the value of the reserves that the company needs to build on its balance sheet. In general, regulations impose a specific interest rate to calculate the provisions of the firm's liabilities over the short and medium term. Sufficient provisions are required to achieve the solvency of the firm. Furthermore, if the firm's manager can prove that its assets are adequate to cover its liabilities in the long term, the firm is granted permission to use a higher interest rate in its provisions. This allows its capital value on the balance sheet to be lower.

Heuristic and metaheuristic algorithms have become a new standard when dealing with complex and large-scale portfolio optimisation and risk management problems (Soler-Dominguez et al., 2017; Doering et al., 2019a). Hence, in this chapter we first propose a constructive heuristic to solve the deterministic version of the ALM problem. The greedy behaviour of the heuristic is then relaxed by using a skewed probability distribution, thus transforming it into a biased-randomised (BR) algorithm. This BR algorithm is able to generate many promising solutions to the deterministic version of the ALM problem. Finally, this probabilistic algorithm is extended into a full simheuristic one (Juan et al., 2018) in order to deal with the stochastic version of the ALM problem, in which liability values are modelled as random variables. Thus, our simheuristic algorithm finds out which assets of a firm's portfolio can be efficiently used to reduce the risk of liability default while minimising the monetary cost for the company. The rest of the chapter is structured as follows: Section 5.2 discusses the typical cash-flow behaviour in both assets and liabilities. Section 5.3 provides a formal model for the optimisation problem being analysed. Section 5.4 presents recent work on BR algorithms and simheuristics. Section 5.5 proposes a greedy heuristic as an initial solving method and its extension to a BR algorithm and a full simheuristic. A series of computational experiments are carried out in Section 5.6, while Section 5.7 provides an analysis of the obtained results. Finally, Section 5.8 highlights the most relevant findings of our work and points out future research lines.

colorblack

5.2 Cash-Flows of Liabilities and Assets

Under an insurance policy, the insurer is liable to pay whenever the event described in the contract takes place. This is a 'must' obligation that the insurer has to honour. Otherwise, the company would face a hefty monetary fine, its reputation would be severely damaged, and its administrators could be taken to court. The insurer's liabilities comprise all policies subscribed by its customers. This aggregation results in an irregular and difficult-to-predict cash-flow structure. Indeed, each policy has a different maturity and size, and is bound to a set of conditions. Figure 5.1 illustrates a series of liabilities (L) and assets (A) with different monetary values (vertical axis) and time occurrence (horizontal axis).

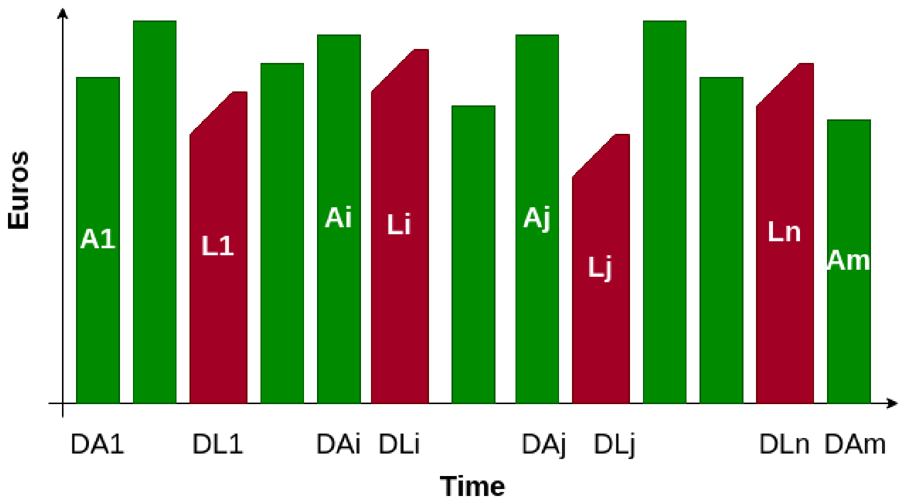


Figure 5.1: An Illustrative Representation of Different Assets and Liabilities over Time

Once the structure of liabilities and assets is known over a time horizon, the manager is tasked to select a set of assets to cover the liabilities in each period. Because of the opportunity cost of these assets, the total value of those selected should be the minimum required, since these assets remain 'frozen' and cannot be used for any other purpose. In other words, once the assets that will cover the firm's liabilities have been selected, they cannot be used in any other transaction. Therefore, this results in an optimisation problem, in which a set of minimum-value assets has to be determined to cover the firm's liabilities. Corporate and government bonds are the predominant asset classes in the insurance market, since returns on a bond market investment can be accurately predicted in advance. The static assumption makes it simpler to predict the value of assets, as opposed to the value of liabilities. It is also worth noting that assets feature a significantly shorter span time than liabilities. For instance, while insurance contracts cover the customer's retirement or full life – which can span over 45 years – typical maturities of bond market instruments do not extend beyond 30 years. This generates a maturity mismatch between assets and liabilities. In addition, while liability cash-flows might arise at any moment in time, the cash-flow structure of assets is more concentrated around some particular time periods.

5.3 Modelling a Stochastic Asset-Liability Management Problem

In this chapter, we first propose a deterministic and relatively simple version of the ALM problem, which is then extended into a stochastic version by modelling liability values as random variables instead of assuming that they are constant values. The deterministic version allow us to use exact methods to generate optimal solutions –at least in some small-sized instances. It is possible then to compare the results of our BR algorithm against these optimal values, which contributes to validate, at least in the deterministic scenario, the quality of the proposed methodology.

Given a set of liabilities L and a set of assets A, the binary variable x_{al} takes the value 1 if asset $a \in A$ is employed to cover liability $l \in L$, being 0 otherwise. Let us denote by t_a the time at which asset $a \in A$ becomes available, and by v_a its value at that time. Similarly, let t_l represent the maturity date of liability $l \in L$, and v_l the associated value to be covered. Our initial goal is to find the asset-to-liability mapping, $(a_{l1}, a_{l2}, \ldots, a_{l|L|})$, that minimises the aggregated net present value (NPV) of the assets employed to cover all of our liabilities, i.e.:

$$min \sum_{a \in A} \sum_{l \in L} \frac{v_a}{(1+d)^{t_a}} x_{al},$$

where d is the discount factor used for calculating the NPV of an asset. Also, we need to make sure that each liability $l \in L$ is covered by exactly one asset $a \in A$, i.e.:

$$\sum_{a \in A} x_{al} = 1, \ \forall l \in L.$$

Likewise, we need to ensure that no asset is assigned to more than one liability:

$$\sum_{l \in L} x_{al} \leq 1, \ \forall a \in A.$$

Also, for each liability $l \in L$, the asset assigned to l needs to be available on or before t_l, i.e.:

$$\sum_{a \in A} x_{al} t_a \leq t_l, \ \forall l \in L.$$

Likewise, it is required that if an asset $a \in A$ is selected to cover a liability $l \in L$, then the monetary value of a (at the time it becomes available) has to be equal or higher than the monetary value of l (at its maturity date):

$$\sum_{a \in A} x_{al} v_a \geq v_l, \ \forall l \in L.$$

Finally, we can add the binary variables to complete the model:

$$x_{al} \in \{0, 1\}, \ \forall a \in A, \forall l \in L.$$

In this chapter, we will also consider a stochastic version in which the value of any liability $l \in L$ is modelled as a positive random variable, V_l. Therefore, Equation (5.3) will be transformed in minimising the expected aggregated NPV, while Equation (5.3) will be substituted by the following probabilistic constraint:

$$Pr\left[\sum_{a \in A} x_{al} v_a \geq V_l\right] \geq p, \ \forall l \in L,$$

where p is a user-defined probability related to the reliability level required of a solution in order to avoid costly defaults. Actually, given a solution of the problem,

$s = (a_{l1}, a_{l2}, \ldots, a_{l|L|})$, its associated reliability level, $R(s)$ can be computed as: $R(s) = \prod_{l \in L} Pr(v_{a_l} \geq V_l)$.

5.4 Recent Work on Biased-Randomised Algorithms and Simheuristics

By combining skewed probability distributions with Montecarlo simulation, biased-randomised techniques can be used to transform a greedy heuristic into a probabilistic algorithm. One of the main advantages of BR algorithms is their ability to generate multiple promising solutions that still follow the logic behind the original heuristic (Juan et al., 2009). BR techniques have been successfully used during the last years to solve different rich and realistic variants of vehicle routing problems (Fikar et al., 2016), permutation flow-shop problems (Ferrer et al., 2016), location routing problems (Quintero-Araujo et al., 2017), facility location problems (Pages-Bernaus et al., 2019), waste collection problems (Gruler et al., 2017a), horizontal cooperation problems (Quintero-Araujo et al., 2019a), and constrained portfolio optimisation problems (Kizys et al., 2019a).

A different concept, also combining simulation with heuristic optimisation, is that of simheuristics (Rabe et al., 2020). Simheuristics can be seen as an extension of metaheuristics, since a simulation module is integrated inside the metaheuristic framework to efficiently deal with *NP-hard* and large-scale stochastic optimisation problems (Ferone et al., 2019). Notice that simheuristics might employ any type of simulation, e.g., discrete-event, agent-based, or Montecarlo. These algorithms have also been used recently in multiple sectors, including: flow-shop scheduling (Hatami et al., 2018), waste collection management (Gruler et al., 2017a), vehicle routing (Gonzalez-Martin et al., 2018; Guimarans et al., 2018), Internet computing (Cabrera et al., 2014), finance (Panadero et al., 2018), e-commerce (Pages-Bernaus et al., 2019), system reliability (Faulin et al., 2008b), and inventory routing (Gruler et al., 2018, 2020). All in all, both BR algorithms and simheuristics demonstrate the great potential that simulation has when combined with heuristics and metaheuristics, either for solving *NP-hard* deterministic optimisation problems as well as to cope with their stochastic counterparts.

5.5 From a Greedy Heuristic to a Simheuristic

Figure 5.2 offers an overview of our solving approach. First, a fast constructive heuristic is designed to solve the deterministic version of the problem. The heuristic completes the following steps: *(i)* it sorts the list of liabilities from the most challenging ones (i.e., those with higher values to cover) to the less challenging ones; *(ii)* it computes the NPV for each asset; and *(iii)* for each liability in the sorted list, it chooses the asset with the minimum NPV among a list including the ones that occur on or before the maturity date of the liability, with a value exceeding that of the liability. Covering the largest liabilities first as efficiently as possible helps to reduce the value of the frozen assets. Algorithm 1 provides the pseudo-code of this greedy heuristic. In the second step, the previous heuristic is transformed into a biased-randomised algorithm by using a Geometric probability distribution to introduce a small random deviation in the order in which the assets are selected from the sorted-by-NPV list (but still respecting the time-precedence constraint). The single parameter of the Geometric distribution $\alpha \in (0, 1]$ defines the probability that the first element of the sorted-by-NPV list is selected, subsequent elements have a diminishing probability of being selected. The BR algorithm is capable of generating multiple solutions per second, all of them following the heuristic criterion (but with greed biased randomisation), with some of them outperforming the solution provided by the greedy heuristic itself. Now, in the final step the most promising solutions generated by the BR algorithm are sent to a Montecarlo simulation process, where a number of executions are run using randomly generated values for the stochastic variables V_l ($\forall l \in L$).

The simulation does not only estimate the expected cost associated with any of the promising solutions generated by the BR algorithm, but it also provides estimates to other key performance indicators, e.g.: the variability of the values that each solution generates (which can be used to compare different solutions in a multiple

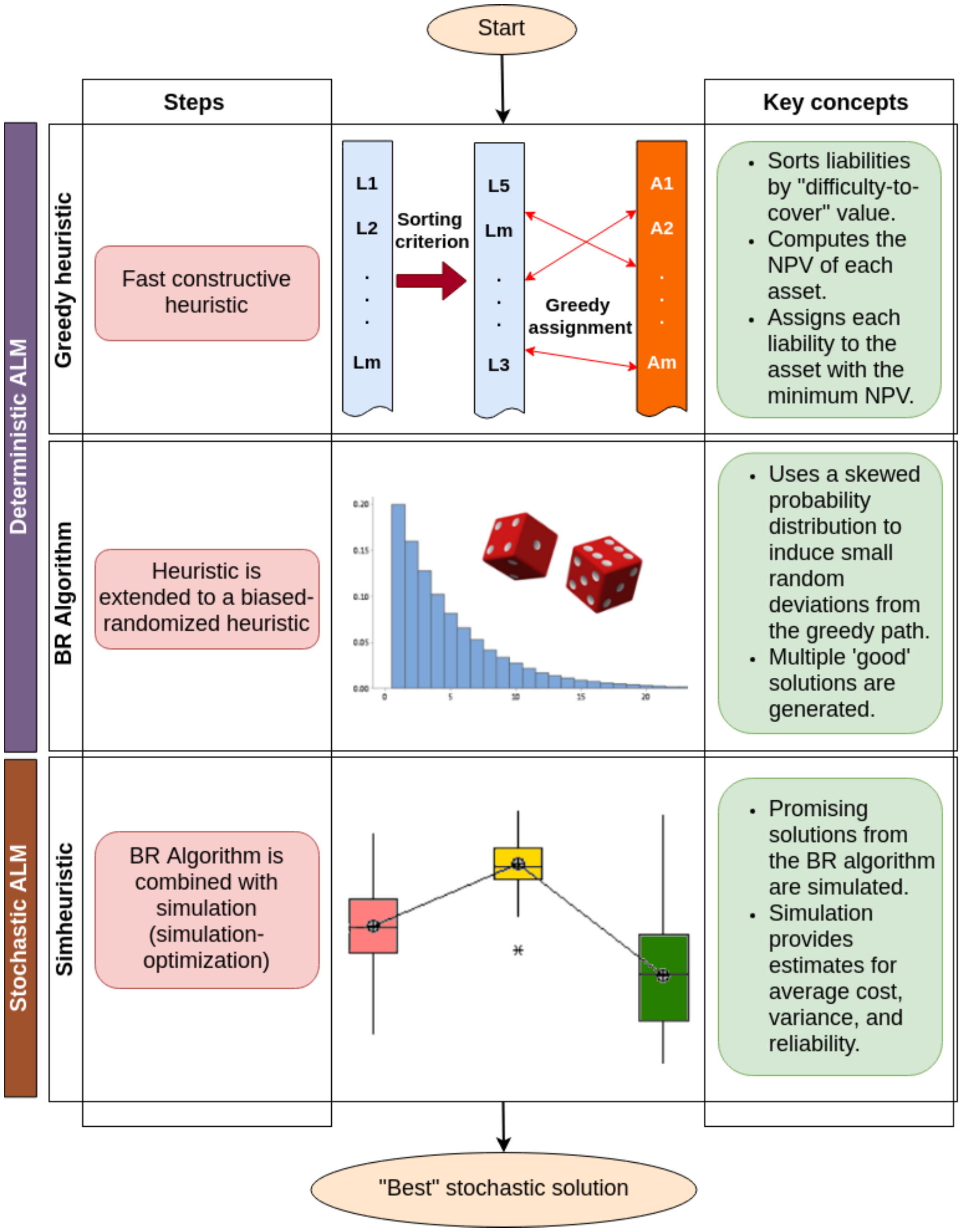


Figure 5.2: Schematic Representation of our Simheuristic ALM Approach

Algorithm 1 Greedy Heuristic.

Sort liabilities list L from higher to lower v_l $(\forall l \in L)$
sol $\leftarrow \emptyset$
for each liability l in L **do**
  Consider $A^* = \{a \in A : t_a \leq t_l \cap v_a \geq v_l\}$
  $a_l \leftarrow \text{argmin}\{NPV(v_a) : a \in A^*\}$
  $A \leftarrow$ delete a_l
  sol $\leftarrow$ add a_l
end for
return sol

boxplot), or the reliability level of each solution –measured as the probability that the corresponding assets-to-liability mapping can be implemented in a stochastic environment without suffering any default (i.e., the probability that all selected assets have successfully covered the assigned liabilities).

5.6 Computational Experiments

The proposed approach has been implemented as a Java application running on a CPU with 3.60 GHz and 16 GB of RAM. Several instances have been generated. Table 5.1 provides some details on the number of assets and liabilities for each instance, as well as the associated discount rate and value modifier –when applicable. Assets and liabilities have been randomly distributed over time using a uniform probability distribution from 0 to 100 and from 50 to 150, respectively. Likewise, values for assets and liabilities have been randomly generated using a uniform probability distribution from 0 to 1 and from 0 to 0.5, respectively. This approach results in feasible instances in which it is possible to cover all of the liabilities. Additionally, asset values in instance 4 have been modified to consider a scenario where their value increases over time, i.e.: given an asset $a \in A$ with a value v_a at time t_a, a new value v'_a is computed $v'_a = v_a(t_a/T)$, with $T = \max\{t_a : a \in A\}$. Likewise, a scenario with decreasing asset values is considered in instance 5 by using $v'_a = v_a(1 - t/T)$. As specified in Table 5.1, similar modifications have been performed on liability values for instances 6 and 7.

#	Instance	# assets	# liabilities	Discount rate	Asset value modifier	Liability value modifier
1	Control_Instance	1000	200	0.05	-	-
2	Large_x3	3000	600	0.05	-	-
3	Large_x5	5000	1000	0.05	-	-
4	Asset_Value_Increases	1000	200	0.05	t/T	-
5	Assets_Value_Decreases	1000	200	0.05	$1-(t/T)$	-
6	Liability_Value_Increases	1000	200	0.05	-	t/T
7	Liability_Value_Decreases	1000	200	0.05	-	$1-(t/T)$
8	Reduced_Discount_Rate	1000	200	0.005	-	-
9	Liabilities_x2	1000	400	0.05	-	-

Table 5.1: Characteristics of the Set of Instances

Some initial experiments have allowed us to set the parameter α associated with the geometric probability distribution that drives the BR algorithm. In this case, a reasonably good performance seems to be reached when $\alpha \in (0.70, 0.95)$. In the stochastic scenario, and in order to search for more reliable solutions, a 'safety-stock' value has been considered. Thus, given a liability $l \in L$, only assets with a value exceeding $E[V_l] + \lambda\sigma_{Vl}$ can be selected to cover l, where σ_{Vl} refers to the standard deviation of V_l and $\lambda \in \{0, 1, 2, 3\}$. Notice that the higher the value of λ, the more reliable the solution will be – i.e., a solution built with a relatively high value of λ will be able to 'absorb' a higher degree of variability in V_l without suffering a default. However, it is also true that increasing λ comes at the cost of using assets with a higher value, which will tend to increase the objective function. A total of 10 seconds per instance has been allowed. Each 'promising' solution s generated by the BR algorithm is sent to the Montecarlo simulation module. In our experiments, a solution was considered to be promising if its deterministic value was equal or better than the one provided by the greedy heuristic. Once in the simulation module, a total of 250 runs are executed per solution. These runs employ random observations from the probability distribution modelling each V_l, $\forall l \in L$. Given a solution $s = (a_{l1}, a_{l2}, \ldots, a_{l|L|})$, for each asset a_l assigned to a liability l, a penalty cost p_l is incurred whenever $V_l > v_{a_l}$ (this cost represents the 'failure-to-cover-a-liability' situation). In that case, $p_l = 2(V_l - v_{a_l})$. Therefore, under the stochastic scenario, the total cost of a solution is computed as the aggregated NPV of its assets plus all the penalty costs incurred. In summary, several statistics can be obtained from the simulation component, among others: *(i)* expected total cost of a solution s; and *(ii)* reliability of s, computed as the percentage of runs in which the mapping has been implemented without any default.

Table 5.2 provides the experimental results. The first column contains the instance number (same as in Table 5.1). The second column contains the value of λ used to compute the 'safety-stock' as described before. For $\lambda = 0$, the adjacent column contains the optimal value for each instance as provided by the popular IBM Cplex solver (this value refers just to the deterministic scenario, and it allows us to validate the results provided by the greedy heuristic). The next three columns refer to the greedy heuristic: *BDS-D* represents the best-deterministic solution evaluated in a deterministic scenario, while *BDS-S* represents the cost of the same asset-to-liability mapping plan being evaluated in a stochastic scenario. The reliability of the mapping is also provided. Likewise, for the BR algorithm, the corresponding columns show the best-stochastic solution under the uncertainty (*BSS-S*), and its associated reliability. Finally, some gaps between pairs of columns are also provided.

			Greedy			BR Algorithm		Gaps			
#	λ	Cplex (1)	BDS-D (2)	BDS-S (3)	Rel. (4)	BSS-S (5)	Rel. (6)	(2) - (1)	(3) - (2)	(5) - (3)	(6) - (4)
1	0	1.25	1.26	2.83	0.00	2.68	0.00	0%	125%	-5.2%	0.00%
1	1	–	1.34	1.59	0.00	1.56	0.00		18%	-1.7%	0.00%
1	2	–	1.43	1.45	0.27	1.45	0.24		1%	-0.4%	-13.24%
1	3	–	1.56	1.56	0.92	1.56	0.95		0%	0.0%	3.93%
2	0	3.73	3.73	10.22	0.00	9.94	0.00	0%	174%	-2.8%	0.00%
2	1	–	4.03	5.19	0.00	5.12	0.00		29%	-1.4%	0.00%
2	2	–	4.33	4.43	0.00	4.42	0.00		2%	-0.2%	0.00%
2	3	–	4.61	4.62	0.69	4.61	0.70		0%	-0.1%	0.58%
3	0	OoM	6.19	17.45	0.00	17.15	0.00	OoM	182%	-1.7%	0.00%
3	1	–	6.70	8.81	0.00	8.73	0.00		32%	-0.9%	0.00%
3	2	–	7.23	7.42	0.00	7.41	0.00		3%	-0.1%	0.00%
3	3	–	7.69	7.70	0.47	7.70	0.47		0%	0.0%	0.00%
4	0	1.22	1.23	2.75	0.00	2.61	0.00	1%	124%	-5.3%	0.00%
4	1	–	1.31	1.56	0.00	1.55	0.00		19%	-0.8%	0.00%
4	2	–	1.42	1.44	0.24	1.44	0.25		1%	-0.2%	3.33%
4	3		Inf.	Inf.		Inf.					
5	0	3.66	3.66	5.87	0.00	5.74	0.00	0%	60%	-2.1%	0.00%
5	1	–	4.27	4.69	0.00	4.66	0.00		10%	-0.5%	0.00%
5	2	–	5.02	5.05	0.12	5.05	0.10		1%	-0.1%	-10.34%
5	3	–	5.85	5.85	0.84	5.85	0.88		0%	0.0%	4.74%
6	0	5.99	7.97	8.49	0.00	8.21	0.00	33%	6%	-3.3%	0.00%
6	1	–	8.28	8.35	0.00	8.09	0.00		1%	-3.1%	0.00%
6	2	–	8.17	8.18	0.42	7.98	0.39		0%	-2.4%	-8.49%
6	3	–	8.76	8.76	0.94	8.53	0.95		0%	-2.6%	0.42%
7	0	10.06	10.18	10.74	0.00	10.65	0.00	1%	6%	-0.8%	0.00%
7	1	–	10.88	10.97	0.00	10.83	0.00		1%	-1.3%	0.00%
7	2	–	11.23	11.24	0.47	11.23	0.45		0%	-0.1%	-4.27%
7	3	–	11.71	11.71	0.95	11.65	0.97		0%	-0.5%	1.68%
8	0	33.99	34.01	36.36	0.00	36.29	0.00	0%	7%	-0.2%	0.00%
8	1	–	37.10	37.50	0.00	37.49	0.00		1%	0.0%	0.00%
8	2	–	39.98	40.01	0.11	40.00	0.12		0%	0.0%	7.14%
8	3	–	42.82	42.82	0.85	42.81	0.90		0%	0.0%	5.16%
9	0	3.58	3.63	6.98	0.00	6.78	0.00	1%	92%	-2.7%	0.00%
9	1	–	3.98	4.56	0.00	4.52	0.00		15%	-0.9%	0.00%
9	2	–	4.24	4.29	0.04	4.28	0.04		1%	-0.2%	0.00%
9	3	–	4.58	4.59	0.83	4.58	0.84		0%	-0.1%	0.48%

Table 5.2: Results Obtained for each Instance and λ Value.

5.7 Analysis of Results

As it can be seen in Table 5.2, the greedy heuristic is providing reasonably good solutions when compared with the optimal ones given by Cplex for the deterministic scenario with $\lambda = 0$. Notice, however, that Cplex is not able to solve all instances since it gets an "out of memory" (OoM) error for instance 3 (which justifies the need of using heuristics even for the deterministic case). Also, notice that the cost of the greedy mapping is quite different in the deterministic scenario (*BDS-D*) and in the stochastic one (*BDS-S*), as can be easily appreciated in Figure 5.3 (instance 8 has been removed from this multi-boxplot figure since its values were outliers that make a clear view difficult in this case). In other words, it is not possible to use the deterministic cost as a good estimate of the stochastic one – therefore, a simulation component is required while solving the stochastic scenario. Regarding reliability values, one can observe that these values rise as λ increases. However, increasing the 'safety stock' will also lead to selecting more expensive assets and, accordingly, to solutions with a typically higher NPV cost. Moreover, in the case of instance 4 no feasible solution is obtained when λ is set to its maximum level (i.e., for $\lambda = 3$ the algorithm cannot find assets with the requested high value). Finally, a relevant result is that the best-stochastic solution provided by the BR algorithm for the stochastic scenario (*BSS-S*) is frequently able to outperform the equivalent *BDS-S*. This justifies the need for using the BR algorithm, which provides different alternative solutions for the simulation component to evaluate.

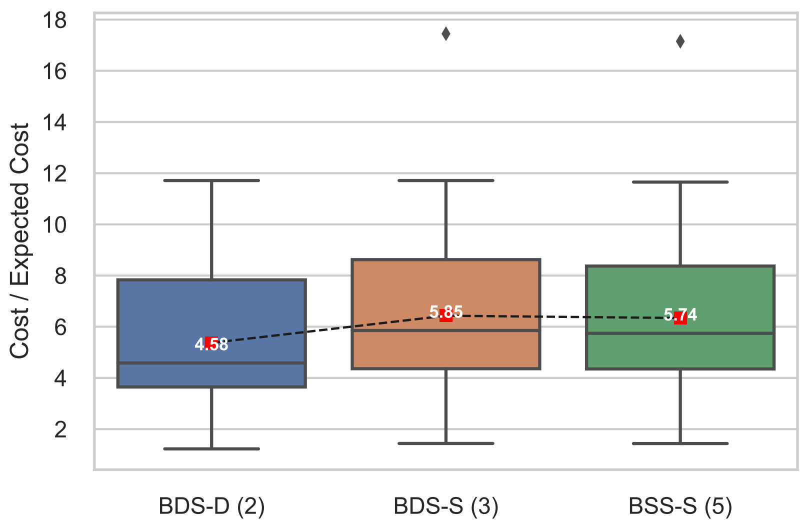


Figure 5.3: A Boxplot Comparison of Different Solutions

5.8 Conclusions

This chapter proposes a simheuristic approach for a stochastic version of the asset and liability management problem. The chapter first addresses the deterministic version of the problem by introducing a greedy heuristic as well as a biased-randomized algorithm. The latter is then extended into a full simheuristic by integrating simulation into the optimisation framework. Our method is flexible and it can be easily extended for new constraints, such as those due to financial regulations or the firm's strategic plans.

The results show that the best deterministic mapping of assets to liabilities is far from being an optimal solution when uncertainty is present. Hence, simulation-optimisation methods become necessary to generate high-quality solutions whenever some components of the asset and liability management problem need to be modelled as random variables instead of deterministic values. Also, according to our computational experiments, the savings generated by the simheuristic are noticeable. Considering that the insurance market is strongly regulated, having an efficient, flexible, and easy-to-implement method to select the proper assets inside a firm's portfolio is extremely important.

As future work, we plan to: *(i)* extend our probabilistic algorithm into a full metaheuristic; *(ii)* include additional characteristics in the model so it fully represents the real-life problem that insurance companies and other financial institutions have to face; and *(iii)* introduce and test the algorithm in real-life bench-mark instances.

Chapter 6

Matheuristic with Simulation for Stochastic ALM

Abstract

Specially in the case of scenarios under uncertainty, the efficient management of risk when matching assets and liabilities is a relevant issue for most insurance companies. This chapter considers such a scenario, where different assets can be aggregated to better match a liability (or the other way around), and the goal is to find the asset-liability assignments that maximises the overall benefit over a time horizon. To solve this stochastic optimisation problem, a simulation-optimisation methodology is proposed. We use integer programming to generate efficient asset-to-liability assignments, and Montecarlo simulation is employed to estimate the risk of failing to pay due liabilities. The simulation results allow us to set a safety margin parameter for the integer program, which encourage the generation of solutions satisfying a minimum reliability threshold. A series of computational experiments contribute to illustrate the proposed methodology and its utility in practical risk management.

6.1 Introduction

Within the enormous variety of insurance types that we can find, long-term life insurance stands out for its complexity in terms of financial management. The cash-flows generated by these insurances extend over several decades and play an important role in the social sphere since they have a close relationship with pensions and retirements and, therefore, with people's vital planning. For this reason, legislation and administrative authorities play a special role in ensuring that insurers faithfully comply with their commitments. The fact that they are extended in the long term, or in the very long term, generates a series of difficulties for their management because the insurer must plan the necessary income with enormous precision to cover its future commitments. Therefore, it is a requirement that the insurer has a range of techniques that allow for matching its assets, as long-term income generators, with its liabilities. Conventionally, we refer to this set of techniques as asset and liability management (ALM) (Ziemba et al., 1998), and it has raised the interest of numerous researchers over the last few years, with a wide variety of approaches being proposed. One of most popular solutions to this asset management problem is cash-flow matching (Iyengar and Ma, 2009), whose main objective is to ensure the timely payment of the liabilities. This approach minimises the number of contractual breaches. Due to the volatility of the financial markets, we always have uncertainty regarding income, and this will be linked to the quality of financial assets. Moreover, the credit quality of assets plays a fundamental role, in particular when we deal with bonds, which are widely used in the insurance industry (Gründl et al., 2016). When the default event occurs, the price of the bond is immediately decreased, in such a way that we have lower income. Since Merton (1974), a lot of models have been developed to forecast the price under a default event.

Likewise, the obligations cannot be considered as exact or totally predictable. Those liabilities or obligations are the customer's premium that the insurance company receives. In practice, we consider average values for obligations and we can establish certain ranges of dispersion that can be estimated based on the insurer's own experience. Once the premium has been paid, the company invests it in the long term, so that the financial benefit envisaged in the insurance policy is secured. Finally, in the event of the customer's retirement or death, the insurance company needs to have sufficient funds to meet its liability to the customer. Consequently, we are facing a highly complex asset allocation problem, since the amount of assets that an insurer can have is large, and the distribution of liabilities over time does not usually follow any regular pattern, both being stochastic in nature.

Heuristic and metaheuristic algorithms have become a new standard when dealing with complex and large-scale portfolio optimisation and risk management problems Soler-Dominguez et al. (2017); Doering et al. (2019a). In this chapter we explore an asset allocation method by means of heuristic techniques, taking into account the random nature of both assets and liabilities. The goal is to find the most efficient (minimum cost) combination of assets that meets certain requirements: they must generate sufficient income to cover the obligations of the insurer with a high probability. In a recent work, Bayliss et al. (2020a) considered a simplified ALM problem, based on the net present value (NPV) concept, in which only one-to-one asset-liability assignment were allowed. Notice that, since we are comparing monetary values of assets that belong to different time periods, it makes sense to consider the NPV associated with each asset in order to make a fairer comparison of assets. Our work goes a step further and allows many-to-many, one-to-many, and many-to-one asset-liability assignments as well. Such an approach increases the efficiency with which liabilities can be covered. This also allows us to address ALM problems regardless of the number of assets and liabilities, as well as their sizes. For addressing large scale instances which could not be solved using exact integer programming techniques, previous approaches were based on the use of greedy heuristics that prioritised larger liabilities over smaller ones. This work, however, proposes an improved approach based on sorting liabilities in ascending due date order, since liabilities with earlier due dates have fewer assets combinations that can be assigned to them. Additionally, assets with earlier maturity dates have higher NPVs, which is what is to be minimised. The main methodological contribution of our approach lies in the introduction of a matheuristic algorithm, which integrates integer programming and Montecarlo simulation. In particular, an integer program is solved recursively to generate feasible and efficient asset-liability assignments for a deterministic scenario (where we assume average values for each random variable in the model). After each iteration, the resulting asset-liability assignment mapping (solution) is assessed under a stochastic scenario by using Montecarlo simulation, which also provides estimates of the mapping reliability. The simulation outcomes are also employed to update a safety margin parameter of the integer program that controls the minimum ratio between the values of the assets and the liabilities of the generated asset-liability assignments. The proposed approach is then tested in a wide variety of problem instances. The combination of simulation and optimisation methods in NPV-related financial problems under uncertainty has been also explored in Panadero et al. (2020).

The rest of the chapter is structured as follows: Section 6.2 introduces a more detailed description of the specific ALM problem considered in this chapter. Section 6.3 proposes a matheuristic algorithm for solving the aforementioned problem. A series of computational experiments are carried out in Section 6.4, while Section 6.5 provides an analysis of the obtained results. Finally, Section 6.6 highlights the most relevant findings of our work and points out future research lines.

6.2 Problem Description and Formulation

When the conditions set out in a contract are met, insurers pay the insured. If they do not have sufficient available funds, they are subject to monetary fines issued by monetary authorities and, most likely, to lost customers. In order to ensure the insurers can meet their liabilities, they perform a process of matching assets to liabilities. Assigning assets to liabilities in an efficient manner is critical to the success of an insurance firm, since assigned (or frozen) assets cannot be used for any other purpose. Assets can only be assigned to liabilities if their maturity date precedes the due date of the liability. The value of the assets assigned to liabilities must equal or exceed the liability values. At the same time, asset maturity values and liability payment values are uncertain, thereby introducing a risk that liabilities cannot be met, even when the expected values imply that they could be met on the average.

An asset-liability assignment is the terminology used in this work to refer to a group of assets used to cover a group of liabilities. A feasible solution to the net present value asset-liability management (NPV-ALM) problem consists of a set of asset-liability assignments such that: *(i)* all liabilities are covered; and *(ii)* no individual assets or liabilities are part of more than one asset-liability assignment. Furthermore, a solution is also required to be robust under uncertain asset and liability values. Specifically, a solution must meet a minimum reliability level, where reliability is defined as the probability that all liabilities can be paid successfully using their assigned assets. Figure 6.1 illustrates a single asset-liability assignment consisting of three assets and two liabilities. Notice that, under the expected values for assets and liabilities (dashed lines), the liabilities can be met. However, due to uncertain asset maturity values and liability payment values, there is a risk that the assets fail to cover the liabilities in the assets-liability assignments. If f_i is the probability that asset-liability assignment i fails to cover its liabilities, then the reliability of a set of asset-liability assignment (I) covering all of our liabilities is computed as $r = \prod_{i \in I} (1 - f_i)$. Following Faulin et al. (2008b), we employ Montecarlo simulation to estimate failure probabilities associated with candidate asset-liability assignments.

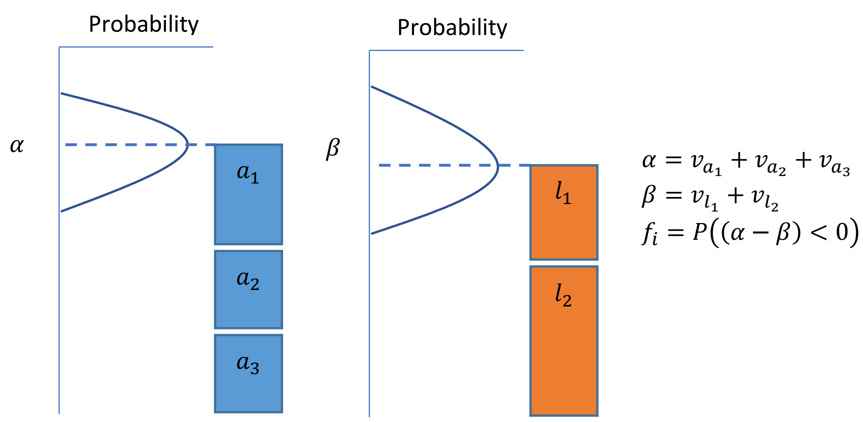


Figure 6.1: An Asset-Liability Assignment with a Failure Probability

In this work, we propose a matheuristic algorithm for solving the NPV-ALM problem. A matheuristic integrates mathematical programming techniques with heuristics in order to develop an algorithm that benefits from exact optimisation as well as from fast and efficient heuristic techniques. For the case of the NPV-ALM problem, an integer program (Section 6.2.2) is used to calculate a set of feasible asset-liability assignment decisions that cover the liabilities. The solution is tested in a simulation to measure its reliability, and the result is employed to tune a safety margin parameter of the integer program. The safety margin parameter controls the minimum ratio between the asset values and the liability values of a generated asset-liability assignment. The process continues until a specified number of iterations have been completed. Section 6.2.1 formulates the NPV-ALM problem.

6.2.1 A Model for the Net Present Value Asset and Liability Management Problem

Summary of the Notation

		Sets
A	:	Set of all assets
L	:	Set of all liabilities
		Stochastic variables
$\tilde{v}_a$	:	The uncertain value of asset a at maturity
$\tilde{v}_l$	:	The uncertain value of liability l on its due date
		Decision variables
y_{ga}	:	Binary variable indicating whether asset a is selected as part of asset-liability assignment g
z_{gl}	:	Binary variable indicating whether liability l is selected as part of asset-liability assignment g
w_a	:	Binary variable indicating whether asset a is selected as part of a generated asset-liability assignment
x_l	:	Binary variable indicating whether liability l is selected as part of a generated asset-liability assignment
		Input parameters
v_a	:	The expected maturity value of asset a
v_l	:	The expected value of liability l on its due date
t_a	:	The maturity maturity date of asset a
t_l	:	The due date of liability l
d	:	Discount factor used to calculate the net present value of an asset
r_{min}	:	Minimum reliability level
m	:	Safety parameter decrease factor
h	:	Safety parameter increase factor
		Other parameters
f_g	:	Failure probability of asset-liability assignment g
N_g	:	Asset-liability assignment g
npv_g	:	Net present value associated with Asset-liability assignment g

The objective (6.1) is to minimise the NPV of the assets committed to covering liabilities. In this context, y_{ga} is a binary decision variable indicating whether asset a is an element of asset-liability assignment g. Similarly, z_{gl} is a binary decision variable indicating whether liability l is an element of asset-liability assignment g. Each asset $a \in A$ can only be part of at most one asset-liability assignment, as specified by Constraint (6.2). Each liability $l \in L$ can only be part of one selected asset-liability assignment, as specified by Constraint (6.3). As a result of Constraints (6.2) and (6.3), the maximum number of asset-liability assignments is $|G| = min\left(|A|, |L|\right)$. A feasible asset-liability assignment requires that each of the selected assets matures before all of the selected liabilities in the asset-liability assignment. Constraint (6.4) introduces a continuous variable ϕ_g representing the latest maturity date of an asset in asset-liability assignment g. Constraint (6.5) introduces a continuous variable σ_g representing the earliest due date of a liability in asset-liability assignment g. Here, H is a large number which ensures the feasibility of Constraint (6.5) in asset-liability assignments that the liability l is not part of. Then, Constraint (6.6) enforces the time constraints for each asset-liability assignment. Constraint 6.7 requires that the sum of the asset values exceeds the value of the covered liabilities by a factor S in each asset-liability assignment g, thus ensuring that our liabilities are covered. Also, S is a multiplicative safety margin parameter for ensuring that the asset values are able to cover the liabilities under uncertain asset returns and liability values. Constraints (6.8) and (6.9) define the binary decision variables.

$$min \sum_{g \in G} \sum_{a \in A} y_{ga} \left(\frac{v_a}{(1+d)^{t_a}} \right). \tag{6.1}$$

$$\sum_{g \in G} y_{ga} \leq 1, \;\; \forall a \in A. \tag{6.2}$$

$$\sum_{g \in G} z_{gl} = 1, \;\; \forall l \in L. \tag{6.3}$$

$$\phi_g \geq y_{ga} t_a, \;\; \forall a \in A, \;\; \forall g \in G. \tag{6.4}$$

$$\sigma_g \leq z_{gl} t_l + H\left(1 - z_{gl}\right), \;\; \forall l \in L, \;\; \forall g \in G. \tag{6.5}$$

$$\phi_g \leq \sigma_g, \;\; \forall g \in G. \tag{6.6}$$

$$\sum_{a \in V} y_{ga} v_a \geq S \sum_{l \in U} z_{gl} v_l, \;\; \forall g \in G. \tag{6.7}$$

$$y_{ga} \in \{0, 1\}, \;\; \forall a \in A, \;\; \forall g \in G. \tag{6.8}$$

$$z_{gl} \in \{0, 1\}, \ \forall m \in L, \ \forall g \in G. \tag{6.9}$$

6.2.2 An Integer Programming Model for Generating Feasible Asset-Liability Assignments

Since solution time and memory requirements become an issue when solving the mixed integer program specified in Section 6.2.1 for realistic sized problem instances, our heuristic solution approach is based upon solving an integer program repeatedly to generate a sequence of efficient asset-liability assignments that cover all of the liabilities. This iterative approach is an alternative to generating all of the required asset-liabilities assignments in one go. This approach also vastly reduces the size and complexity of the mathematical programs that need to be solved. This integer program is denoted as $IP\,(U, V, k, S)$. Here, U is the set of remaining uncovered liabilities, and V is the set of available assets currently unassigned to any liabilities. Initially, $U = L$ and $V = A$. Every time a new asset-liability assignment is generated using the integer program, the selected assets are removed from V and the selected liabilities are removed from U. The integer program is solved repeatedly until the set U is empty. The input k is a randomly selected uncovered liability that must be covered by the next asset-liability assignment generated. This provides a mechanism for randomising the sets of asset-liability assignments generated. The i^{th} asset-liability assignment generated is denoted as N_i. It contains the set of selected assets and liabilities. The efficiency of an asset-liability assignment is measured by the value of the liabilities covered minus the value of the assets used, which encourages asset-liability assignments to cover as many liabilities as possible with the fewest assets possible. The net present value of the assigned assets is then subtracted, which captures our overall objective. Higher values of this efficiency measure correspond to more efficient asset-liability assignments. This efficiency objective function is expressed by Objective (6.10). In this expression, x_l is a binary variable indicating which liabilities, $l \in U$, are part of the generated asset-liability assignment, and w_a is a binary variable indicating which assets, $a \in V$, are part of the generated asset-liability assignment.

$$max \sum_{l \in U} x_l v_l - \sum_{a \in V} w_a v_a \left(1 + \frac{1}{(1+d)^{t_a}}\right). \tag{6.10}$$

A feasible asset-liability assignment requires that each of the selected assets matures before the selected liabilities. Constraint (6.11) expresses this, where t_m is the asset maturity date or liability due date of an asset or liability $m \in V \cup U$. Also, H is a large number which is used to ensure that Constraint (6.11) remains feasible in cases where liabilities are not selected. Optionally, Constraint (6.11) can be replaced by a constraint using the same form used in Constraints (6.4)-(6.6).

$$w_a t_a \leq x_l t_l + H\left(1 - x_l\right), \ \forall a \in V, \ \forall l \in U. \tag{6.11}$$

Constraint (6.12) requires that the sum of the asset values exceeds the value of the covered liabilities by a factor S, where S is a multiplicative safety margin parameter for ensuring that the asset values are able to cover the liabilities under uncertain asset returns and liability values.

$$\sum_{a \in V} w_a v_a \geq S \sum_{l \in U} x_l v_l. \tag{6.12}$$

Constraint (6.13) states that the randomly selected uncovered liability, k, must be included in the next asset-liability assignment generated.

$$x_k = 1. \tag{6.13}$$

Constraints (6.14) and (6.15) define the binary decision variables.

$$x_l \in \{0,1\}, \ \forall l \in U. \tag{6.14}$$

$$w_a \in \{0,1\}, \ \forall a \in V. \tag{6.15}$$

6.3 Our Matheuristic Approach

This section describes our matheuristic algorithm, which combines integer programming and Montecarlo simulation for solving the NPV-ALM problem. This solving approach consists of two main phases: *(i)* generation of 'promising' solutions; and *(ii)* simulation and parameter tuning of the aforementioned solutions. The solution generation phase uses integer programming (specified in Section 6.2.2) to generate a set of asset-liability assignments that cover the liabilities. This process is iterative, i.e., each iteration generates one new asset-liability assignment from the remaining unused assets and uncovered liabilities. In order to increase the diversity of these solutions, a random factor is introduced: we randomly select one of the remaining liabilities and add a constraint which forces this liability to be part of the next asset-liability assignment. The simulation phase is used to measure the reliability of the generated solution. Montecarlo simulation is used estimate the *failure probability* associated with each asset-liability assignment. This is the probability that the sum of the maturity values of the assets, in an asset-liability assignment, is less than the corresponding sum of the liabilities. If the solution is sufficiently reliable, a best solution check is performed to see if the solution has the lowest associated NPV of any reliable solution found. The reliability result is also used to update the safety margin parameter of the integer program. The procedure followed is given in Algorithm 2.

6.4 Computational Experiments

The proposed heuristic has been implemented as a Python application running on a CPU with 3.60 GHz and 16 GB of RAM. Instances from Bayliss et al. (2020a) have been used to test the new approach, plus two new instances that could not be solved with the methodology presented in the former chapter. Table 6.1 provides the details on the number of assets and liabilities for each instance, discount rate, and value modifier (if any was employed). Assets and liabilities have been distributed over time using a random uniform probability distribution from 0 to 100 and from 50

Algorithm 2 *AssetLiabilityAssignmentGeneration* $(A, L, r_{min}, \beta, m, h, runs)$

Data: A set of available assets, L set of liabilities, $maxIterations$, r_{min} the minimum reliability level, β geometric distribution parameter, m safety margin decrease factor, h safety margin increase factor, $runs$ the number of Montecarlo simulation runs used to estimate asset-liability assignment failure probabilities

```
iteration = 1 // the number of asset-liability assignments generated so far.
bestSolution ← ∅
bestNPV = ∞
//Initialise the safety margin parameter S = 1
S = 1
while iteration ≤ maxIterations do
    //Reset the set of unassigned assets V and uncovered liabilities U
    V ← A
    U ← L
    newSolutionNPV ← 0
    N ← ∅
    i ← 1
    while U ≠ ∅ do
        //Select an uncovered liability k from an ascending due date sorted list according to a geometric
        distribution with parameter β.
        //Solve integer program to obtain the get the next asset asset-liability assignment N_i.
        (N_i, npv_i) ← IP(U, V, k, S)
        //Estimate the failure probability f_i of the new asset-liability assignment using Montecarlo sampling of
        asset return and liability values.
        f_i ← simulation(N_i, runs)
        newSolutionNPV ← newSolutionNPV + npv_i
        U ← U \ N_i
        V ← V \ N_i
        i ← i + 1
    end
    //Calculate the reliability r of the new solution
    r = ∏_{j=1}^{i-1} (1 − f_i)
    //Update the safety margin parameter using the reliability level of the new solution
    if r ≥ r_min then
        //Decrease the safety margin parameter (slowly)
        S ← mS
        //Check for a new best solution
        if newSolutionNPV < bestNPV then
            bestNPV ← newSolutionNPV
            bestSolution ← N
        end
    else
        //Increase the safety margin parameter (relatively quickly)
        S ← hS
    end
    iteration ← iteration + 1
end
return bestSolution
```

to 150, respectively. Similarly, values for assets and liabilities have been randomly generated using a uniform probability distribution from 0 to 1 and from 0 to 0.5, respectively. Asset values from instances 4 and 5 have been modified to simulate scenarios where its value varies over time, i.e.: given an asset $a \in A$ with a value v_a at time t_a, a new value v'_a is computed $v'_a = v_a f(t_a, T)$, with $T = \max\{t_a : a \in A\}$ and f the asset value modifier function. Likewise, instances 6 and 7 consider scenarios with liability values varying over time: given a liability $l \in L$ with a value v_l at time t_l, a new value v'_l is computed $v'_l = v_l g(t_l, T)$, with $T = \max\{t_l : l \in L\}$ and g the liability value modifier function. Instance 10 simulates a scenario with small assets and large liabilities, which encourages the use of multiple assets to cover a liability, while instance 11 considers a scenario with a few large assets and several small liabilities, to force the use of a single asset to cover multiple liabilities.

#	Instance	# assets	# liabilities	Discount rate	Asset value modifier	Liability value modifier
1	Control_Instance	1000	200	0.05	-	-
2	Large_x3	3000	600	0.05	-	-
3	Large_x5	5000	1000	0.05	-	-
4	Asset_Value_Increases	1000	200	0.05	t/T	-
5	Asset_Value_Decreases	1000	200	0.05	$1-(t/T)$	-
6	Liability_Value_Increases	1000	200	0.05	-	t/T
7	Liability_Value_Decreases	1000	200	0.05	-	$1-(t/T)$
8	Reduced_Discount_Rate	1000	200	0.005	-	-
9	Liabilities_x2	1000	400	0.05	-	-
10	Small_Asset_Large_Liability	1000	200	0.05	0.5	10
11	Large_Asset_Small_Liability	50	1000	0.05	10	0.2

Table 6.1: Characteristics of the Set of Instances

Some initial experiments have been performed using instance 1 to set the parameter α associated with the geometric probability distribution that drives the liability selection and the relative mixed integer programming optimality gap, MIPGap, which is used to terminate the integer programming algorithm. Experiments to determine α have been carried out in a deterministic scenario, while experiments to determine MIPGap have been performed with stochastic variables. In this case, a better performance is attained with $\alpha = 0.75$ and MIPGap $= 0.4$. Figure 6.2 and Figure 6.3 present the results of the numerical tests.

Each instance in Table 6.1 has been solved using the integer programming algorithm presented in Algorithm 2, with a limit of 100 iterations. A time-limit of 300 seconds has also been imposed to terminate the algorithm after a solution has been generated

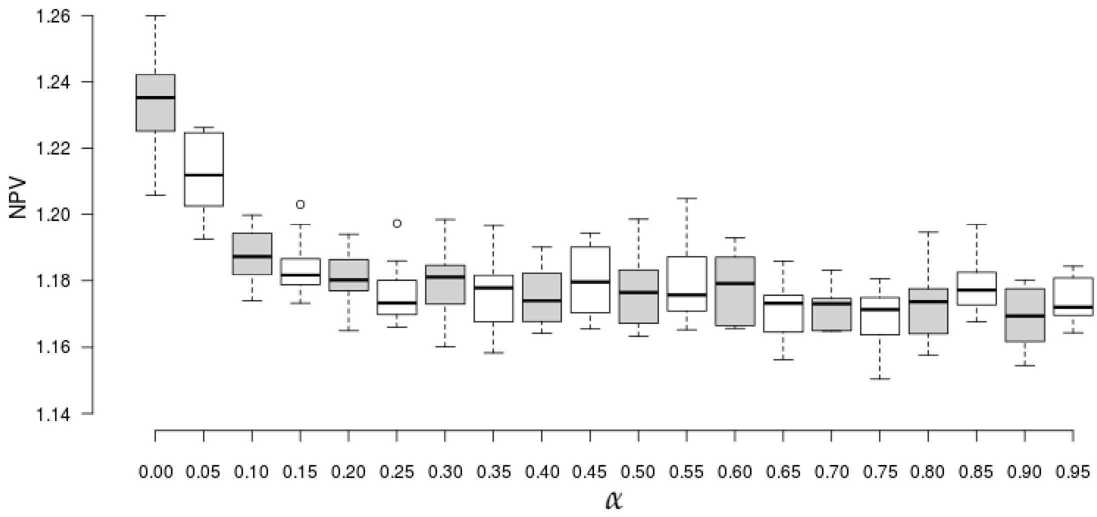


Figure 6.2: Boxplot Comparison of Instance 1 - Results with Different *alpha* Values

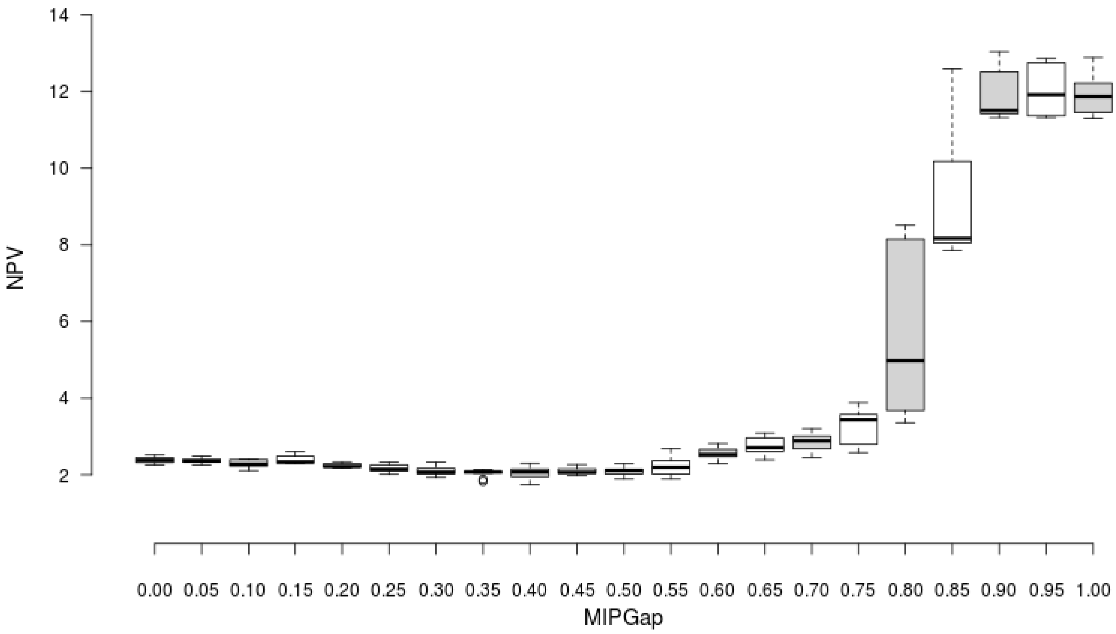


Figure 6.3: Boxplot Comparison of Instance 1 - Results with Different MIPGap Values

if the aforementioned time-limit has been reached. The minimum reliability r_{min} to consider a solution as feasible in the stochastic scenario is 0.95. The values of the parameters to increase and decrease the safety margin parameter S used are $m = 0.99$ and $h = 1.1$. In the stochastic scenario, both asset and liability values have been considered uncertain, with a standard deviation of 5% of its expected maturity value. 500 iterations are executed for each asset-liability assignment generated in the Montecarlo simulation.

Table 6.2 provides the experimental results, compared with the results obtained in Bayliss et al. (2020a). The first column contains the instance number (same as in Table 6.1). The second column (Cplex) contains the optimal value for each instance in a one-to-one asset-to-liability mapping. The third column (BR) contains the results of a previous biased-randomised algorithm, with its associated reliability values in the next column. Then, column 5 contains the best values obtained in the deterministic scenario with our matheuristic algorithm. Similarly, the best solutions obtained with a reliability higher than 0.95 and its reliability are presented in the next two columns. Finally, some gaps between pairs of columns are also provided.

	Bayliss et al. (2020a)			Our Matheuristic			Gaps			
#	Cplex (1)	BR (2)	r (3)	Det. (4)	Stoch. (5)	r (6)	(4) - (1)	(5) - (4)	(5) - (2)	(6) - (3)
1	1.25	1.56	0.95	1.17	1.81	0.95	-6.56%	54.72%	15.84%	0.12%
2	3.73	4.61	0.70	3.51	5.92	1.00	-5.89%	68.65%	28.43%	42.57%
3	OoM	7.7	0.47	5.96	9.43	0.99	-	58.25%	22.44%	111.07%
4	1.22	1.44	0.25	1.18	2.30	0.99	-2.95%	94.06%	59.56%	295.22%
5	3.66	5.85	0.88	1.99	2.97	0.96	-45.73%	49.75%	-49.15%	9.61%
6	5.99	8.53	0.95	3.13	3.72	0.98	-47.69%	18.75%	-56.38%	2.76%
7	10.06	11.65	0.97	9.97	12.28	0.96	-0.88%	23.20%	5.45%	-1.15%
8	33.99	42.81	0.90	34.10	42.64	0.95	0.34%	25.03%	-0.39%	5.69%
9	3.58	4.58	0.84	2.49	5.04	1.00	-30.41%	102.21%	9.99%	18.81%
10	-	-	-	5.25	10.96	0.97	-	108.77%	-	-
11	-	-	-	7.70	11.53	0.96	-	49.79%	-	-

Table 6.2: Results for each Instance

6.5 Analysis of Results

As it can be seen in Table 6.2, the stand-alone matheuristic is providing reasonably good solutions when compared with the optimal ones given by Cplex for the deterministic scenario. Actually, Cplex is not able to solve all instances since it gets an "out of memory" (OoM) error for instance 3 (which justifies the need of using matheuristics even for the deterministic case). Also, notice that the cost of the assets-to-liabilities mapping is quite different in the deterministic scenario (*Det.*) and in the stochastic one (*Stoch.*). In other words, the deterministic scenario represents an 'ideal' (but not realistic) situation that provides a lower-bound to the real NPV cost under uncertainty conditions. Probably, the most interesting comparison in this table is between columns *BR* and *Stoch.* As one can see, the proposed matheuristic-simulation algorithm is usually able to outperform the previous simulation-optimisation approach proposed in Bayliss et al. (2020a). This is mainly due to the fact that the methodology proposed in this chapter does not require to assume a one-to-one mapping between assets and liabilities, thus allowing for an increasing number of mapping combinations. The main benefit of using the matheuristic-simulation algorithm is that it treats reliability as a hard constraint, an issue which is very important in the context of meeting liabilities. However, since the matheuristic is a more complex algorithm than *BR*, the 300 second time limit meant that there was not enough time for it to find solutions that met the 95% reliability constraint exactly, allowing it to achieve a low NPV. Notice that the gap between the NPVs of *BR* and the matheuristic are largest when the matheuristic return very reliable solution, while *BR* returns solution with low reliability. Figure 6.4 highlights the large average reliability gain attained from using the matheuristic, at the expense of a slightly higher NPV on average.

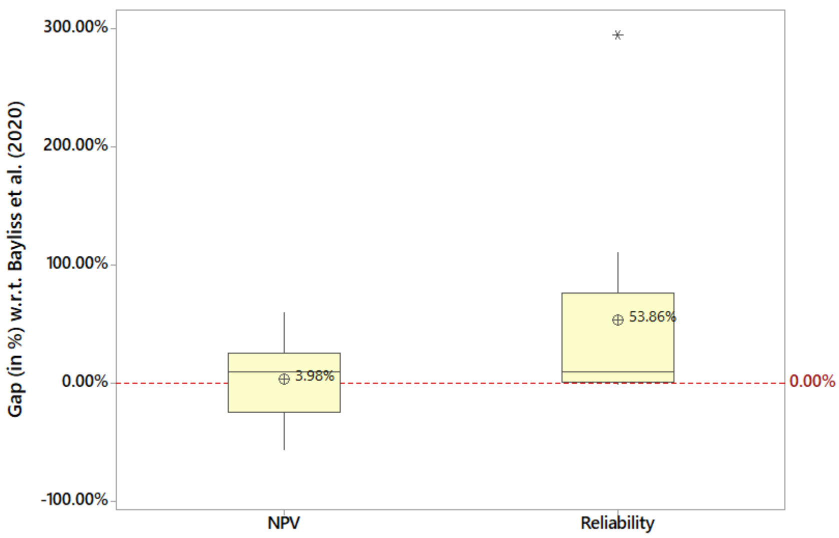


Figure 6.4: Boxplot Comparison of NPV and Reliability Results w.r.t. a Previous Work

6.6 Conclusions

This chapter proposes a hybrid matheuristic simulation approach to solve the stochastic version of the asset and liability management problem, where the goal is to minimise the net present value of the assets that are employed to cover the liabilities, while satisfying a reliability constraint. First, a matheuristic is designed by combining integer programming with a heuristic. The heuristic prioritises the selection of liabilities with an earlier maturity date, and it also makes use of a random procedure to increase the diversity of solutions generated. Then, the most promising solutions generated in the previous stage are simulated in a stochastic scenario. For this, a Montecarlo simulation is run multiple times in order to obtain estimates of the NPV-cost and the associated reliability of each solution. One of the main novelties of this chapter is that approach integrates Montecarlo simulation with a matheuristic to provide and algorithm which can guarantee reliable solutions for the asset and liability management problem. It also considers the possibility of aggregating different assets, or different liabilities, before completing the assignment mapping, i.e.: several assets can be aggregated to cover each liability, and multiple liabilities can be covered by a single asset. To the best of our knowledge, it is the first time that this many-to-many assignment procedure is considered in the literature on asset and liability management.

The results show that the best deterministic mapping of assets to liabilities is far from being an optimal solution when uncertainty is present. Hence, simulation-optimisation methods become necessary to generate high-quality solutions whenever some components of the asset and liability management problem need to be modelled as random variables instead of deterministic values. In addition, the numerical experiments show how, by allowing many-to-many assignments between assets and liabilities, our combined matheuristic-simulation algorithm is able to outperform other simulation-optimisation approaches. As future work, we plan to: *(i)* include additional characteristics in the model so it fully represents the real-life problem that insurance companies and other financial institutions have to face; and *(ii)* introduce and test the algorithm in real-life benchmark instances.

Chapter 7

ALM in Insurance Firms

Abstract

The management of assets and liabilities is of critical importance for insurance companies and banks. Complex decisions need to be made regarding how to assign assets to liabilities such in a way that the overall benefit is maximised over a multi-period horizon. At the same time, the risk of not being able to cover the liabilities at any given period must be kept under a certain threshold level. This optimisation problem is known in the literature as the asset and liability management (ALM) problem. In this work, we propose a biased-randomised algorithm to solve a real-life instance of the ALM problem. Firstly, we outline a greedy heuristic. Secondly, we transform it into a probabilistic algorithm by employing Montecarlo simulation and biased-randomisation techniques. According to our computational tests, the probabilistic algorithm is able to generate, in short computing times, solutions that outperform by far the ones currently practised in the sector.

7.1 Introduction

Financial institutions have to face some critical risk-management processes Cornett and Saunders (2003). Among such processes, asset and liability management (ALM) is of paramount importance due to its potential consequences. ALM consists of a range of techniques necessary to invest adequately, so that the firm's long-term liabilities are met Ziemba et al. (1998). For an insurance company, a liability constitutes the legal responsibility to repay the insurance contributions that the customer has been making over an agreed length of time, which are increased by the interest rate. This is a typical transaction of pension or life insurance intended to secure retirement income, which gives rise to a three-tier financial problem. First, the insurance company receives the customer's premium. Second, the company invests this premium in the long term, so that the financial benefit envisaged in the insurance policy is secured. Third, in the event of the customer's retirement or death, the insurance company needs to have sufficient funds to meet its liability to the customer. While the aforementioned financial problem unfolds, the insurance company is confronted with a range of risks, which arise either from its role as a financial intermediary or due to adverse regulatory as well economic and social policies. If the insurer's obligation to the customer is not honoured, its default becomes a likely scenario. A default can be very costly for the firm, since it can inflict a loss of credibility and reputation. On the one hand, it can face a legal action from its creditors. As a result the insurer may be forced to pay hefty fines by the regulatory body. On the other hand, the firm's market share may diminish as its customers may switch to other insurers.

It is thus not surprising that the ALM problem has been widely studied in the literature. As interest rates vary over time, the present value of both assets and liabilities responds to such variation. Consequently, optimal and smart asset management solutions become critical to the insurer, who seeks to ensure that the liabilities can be met at the time when they are required, while at the same time, the value of the firm is maximised. In practical applications, one of most popular solutions to this asset management problem is the so-called cash-flow matching Iyengar and Ma (2009), whose main objective is to ensure the timely payment of the liabilities. In some European countries, the legislation does not envisage any specific mechanism to ensure that the firm's obligations are met. Instead, capital is regulated by targeting the value of the reserves that the company needs to build on its balance sheet. In general, regulations impose a specific interest rate to calculate the provisions of the firm's liabilities over the short and medium term. Sufficient provisions are required to achieve the solvency of the firm. Furthermore, if the firm's manager can prove that its assets are adequate to cover its liabilities in the long term, the firm is granted permission to use a higher interest rate in its provisions. This allows its capital value on the balance sheet to be lower.

Heuristic and metaheuristic algorithms have become a new standard when dealing with complex and large-scale portfolio optimisation and risk management problems Doering et al. (2019a). Hence, in this paper we propose a heuristic-based algorithm to find out which assets of a firm's portfolio can be efficiently used to reduce the risk of default liability while minimising the monetary cost for the company. Our approach combines Montecarlo simulation (MCS) with a greedy heuristic. This combination results in a biased-randomised probabilistic algorithm. Biased-randomised algorithms make use of random sampling from a skewed probability distribution (e.g., a geometric one) in order to 'inject' some non-uniform (oriented) randomness into a greedy heuristic. That way, the latter is transformed into a more efficient probabilistic algorithm without losing the logic behind the heuristic Grasas et al. (2017). The rest of the chapter is structured as follows. Section 7.2 reviews biased-randomised algorithms using MCS. Section 7.3 discusses the typical cash-flow behaviour in both assets and liabilities. Section 7.5 outlines the optimisation problem. Then, Section 7.6 proposes a greedy heuristic as an initial solving method, while Section 7.7 extends the aforementioned heuristic into a probabilistic algorithm. A series of computational experiments, based on real-life data, are carried out in Section 7.8. Finally, Section 7.9 concludes.

7.2 Recent Work on Biased-Randomised Algorithms

Different examples on the use of Montecarlo simulation methods to guide the search of heuristic-based algorithms can be found in the literature Faulin and Juan (2008); Faulin et al. (2008a); Juan et al. (2009). One particular case is that of biased-randomisation (BR) techniques. As described in detail by Grasas et al. (2017), BR techniques make use of Montecarlo simulation and skewed probability distributions in order to transform a greedy heuristic into a probabilistic algorithm without losing the logic behind the heuristic. This transformation is achieved after sorting each constructive movement by a given criterion and then assigning diminishing probabilities of being selected as the movement becomes less promising. In practice, the use of randomised greediness here allows for a fuller exploration of the solution space, but with the advantage that the effective logic behind the greedy heuristic is retained (Figure 7.1).

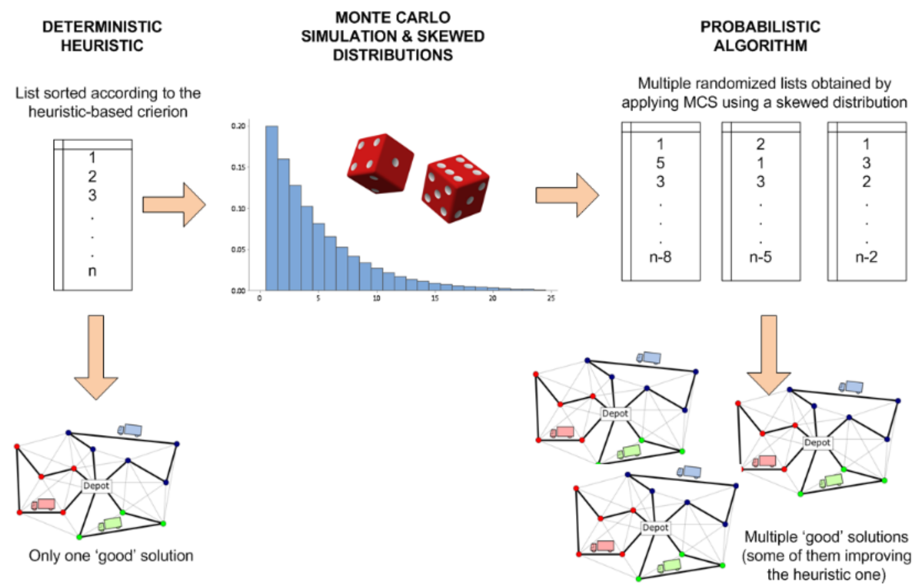


Figure 7.1: Schematic Representation of the Biased-Randomisation Process

BR techniques have been successfully used during the last years to solve different rich and realistic variants of vehicle routing problems Dominguez et al. (2016); Calvet et al. (2016), permutation flow-shop problems Martin et al. (2016); Gonzalez-Neira et al. (2017), location routing problems Quintero-Araujo et al. (2017), facility location problems De Armas et al. (2017), waste collection problems Gruler et al. (2017a), horizontal cooperation problems Quintero-Araujo et al. (2019a), and constrained portfolio optimisation problems Kizys et al. (2019a).

7.3 Cash-Flows of Liabilities and Assets

Under an insurance policy, the insurer is liable to pay whenever the event described in the contract takes place. This is a 'must' obligation that the insurer has to honour. Otherwise, the company would face a hefty monetary fine, its reputation would be severely damaged, and its administrators could be taken to court. The insurer's liabilities comprise all policies subscribed by its customers. This aggregation results in an irregular and difficult-to-predict cash-flow structure. Indeed, each policy has a different maturity and size, and is bound to a set of conditions. Being based on real-life data, Figure 7.2 shows a typical example of how liabilities are distributed over a period of 30 years. Figure 7.2 unveils a long term liability schedule, which sheds light on frequent cash-flows arising from transactions in each time period. To complicate things further, these liabilities are not static, since a common policy can end in different ways: *(i)* when a customer decides to cancel it; *(ii)* when the policy reaches its maturity date; or *(iii)* when the customer dies.

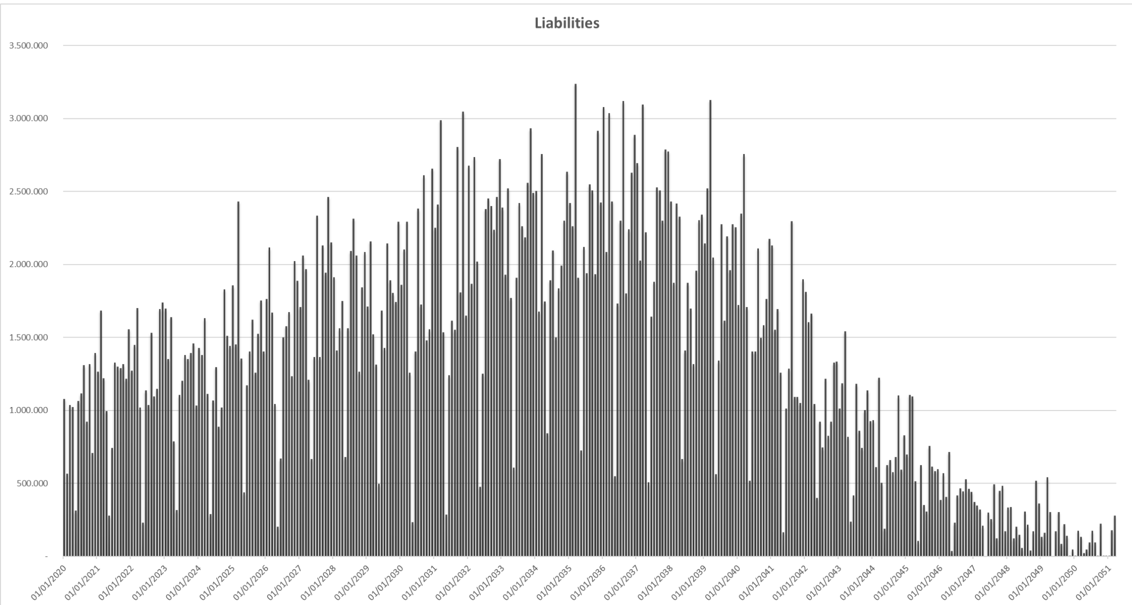


Figure 7.2: Liability Cash-flow Profile of a Hypothetical Portfolio

On the flipside of the insurer's balance sheet, the manager is tasked to select a set of assets to cover the liabilities in each period. Because of the opportunity cost of these assets, the total value of these selected values should be just the necessary one, since these assets remain 'frozen' and cannot be used for any other purpose. In other words, once the assets that will cover the firm's liabilities have been selected, they cannot be used in any other transaction. Therefore, this results in an optimisation problem, in which a set of minimum-value assets has to be determined to cover the firm's liabilities. If liabilities are assumed to be static (deterministic), assets can be optimally selected in advance. Corporate and government bonds are the predominant asset classes in the insurance market, since returns on a bond market investment can be accurately predicted in advance. The static assumption makes it simpler to predict the value of assets, as opposed to the value of liabilities. It is also worth noting that assets feature a significantly shorter span time than liabilities. For instance, while insurance contracts cover the customer's retirement or full life – which can span over 45 years – typical maturities of bond market instruments do not extend beyond 30 years. This generates a maturity mismatch between assets and liabilities. In addition, while liability cash-flows might arise at any moment in time, the cash-flow structure of assets is more concentrated around some particular time periods. Figure 7.3 shows a typical asset portfolio associated with an insurance company. If we compare this structure with the previous one for liabilities, we can observe remarkable differences that suggest a non-trivial matching problem.

7.4 Credit Quality and Fundamental Spread

Along this chapter we are studying an efficient method to assign assets included in the balance sheet to match the liabilities that the company has to tackle in the future. The asset allocation on which the insurance industry is based is mainly fixed income, such corporate bonds, treasury, T notes, sovereign debt, loans, etc., as set forth by the legislation (directive 2009/138/EC of the European Parliament and of the Council). These kinds of assets are characterised by means of two relevant features, the first one is that the income is determined a priori and cannot be changed by the asset issuer or any third parties. This is key to understand why this is the preferred type of selected asset for cash-flow matching or immunisation in long term

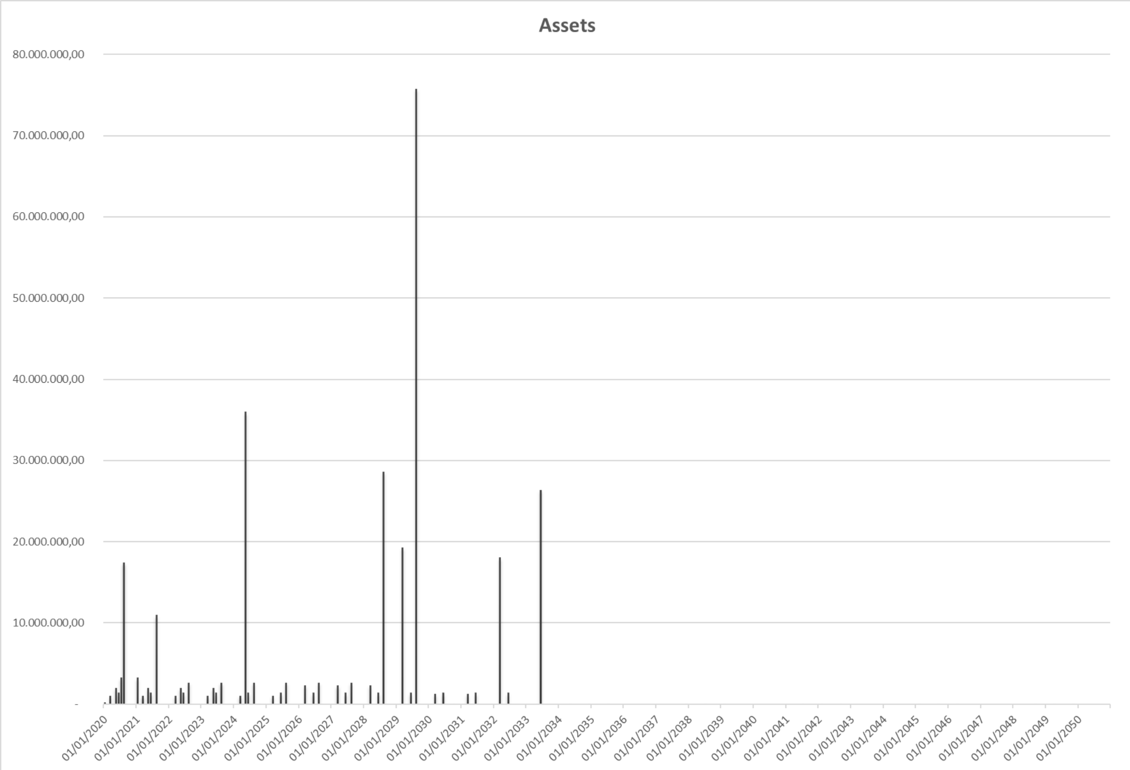


Figure 7.3: Asset Cash-flow Profile in a Typical Portfolio

liabilities portfolio; the second one is the credit risk of the issuer, in other words, to what extend we can assure or rely on the fact of receiving the income in the future. There is a huge amount of study in the literature about modelling the credit risk of fixed income issuers. In (Adamko et al., 2014) we have a short history of credit risk models. The first step historically speaking is done by the rating agencies as a response of the lack of models, highlighting Moody, Poor and Fitch as the pioneers of their field. Moreover, Fitch in 1924 introduces his rating letter-type, AAA, AA ... until D, having AAA for the most reliable issuer and D for the worst one, becoming the method for qualification that is followed for any rating agency nowadays. The first mathematical models are made by (Beaver, 1968) and (Altman, 1968). These are based on econometric models using bankruptcy history in a specific industry and taking a few relevant parameters that describe the company, like revenues, size, etc. The model calculates a Z score, so that below a threshold we suspect a near bankruptcy of the company. (Merton, 1974) studies the first structural model for credit risk. In this paper, the value of the company has a stochastic walk in the same manner as the Black-Scholes model, such that if the value is less than the liabilities, a default event is achieved. Thanks to this model, we can link the probability of default (PD) and the recovery rate (RR) of the bond. The Recovery Rate is the value of the bond in case of default. If the PD increases, the model predicts a decrease in RR, as it can be expected. In contrast with this approach, based on the risk neutral probability, we can find another way to calculate the price using the actuarial default probability (Hull et al., 2005). Following this approach, we identify the bond price as the equivalent policy price where the bond has a mortality; nevertheless, prices observed in the market are always cheaper due to the extra yield the investor claims to compensate the risk.

The risk in one period is not only the study of the probability of default but also the change of the rating of the asset. In (Crouhy et al., 2000) we find a comparative analysis of current credit risk models where they use a transition matrix that gives us the probability between one rating state in one period to another possible state in the next period (table 7.1). Another relevant question is that the probability is dependent on the issuer industry. In (Qi and Zhao, 2011) we find a comparison of modelling methods of the Loss Given Default (LGD), which is the part of the value we lose in case of default. In that paper different LGD are shown among industries.

Rating	AAA	AA	A	BBB	BB	B	C	Def
AAA	91.00	8.00	0.70	0.13	0.10	0.04	0.02	0.01
AA	0.60	90.90	7.60	0.60	0.10	0.10	0.06	0.04
A	0.10	2.30	91.20	5.00	0.70	0.20	0.10	0.40
BBB	0.00	0.30	5.50	88.20	4.70	1.00	0.10	0.20
BB	0.00	0.10	0.60	7.00	82.70	7.60	0.90	1.10
B	0.00	0.10	0.20	0.40	6.00	85.00	3.40	4.90
CCC	0.20	0.00	0.30	1.00	2.20	9.60	67.30	19.40

Table 7.1: Transition Probabilities of Credit Risk. Source: (Crouhy et al., 2000)

The algorithm presented below calculates the present value just to select what asset is best to be matched with the portfolio of liabilities. In that present value we use in our approach a constant risk-free interest rate without considering the credit quality of the issuer. The easiest way to bring in the default probability is through the concept of Fundamental Spread, which is recognised in Article 77c(2)(a)(i) of Directive 2009/138/EC, and in the Delegated Regulation (EU) 2015/35, subsection 4 Matching Adjustment, art. 54. This spread is the number we have to increase to the free risk interest rate so that we consider the cost of default and downgrade.

The aforementioned article describes the conditions the company has to fulfil for the calculation to be valid. In particular, it states that the calculation has to be transparent and prudent, it has to be considered that 30% of the market value will be recovered in case of default and the calculation has to be based on data relating to the last 30 years.

Once we have the fundamental spread of each asset, we merely could use it in our present value calculation prior to the selection of each asset. Indeed, in equation 7.4 we define our optimisation goal that is related to the Present Value of each asset. The present value of asset α is defined as:

$$PV_\alpha = \sum_i \frac{CF_i^\alpha}{(1+\pi)^i} \tag{7.1}$$

Where CF_i^α refers to all the cash-flows of the asset and π is the risk-free interest rate. Introducing the fundamental spread, the present value is now:

$$PV_\alpha = \sum_i \frac{CF_i^\alpha}{(1+\pi+\delta_\alpha)^i} \tag{7.2}$$

Where δ_α is the fundamental spread of asset α. Let's keep in mind that we will have to use a different fundamental spread for each kind of asset, even we could use a fundamental spread having the same issuer since the probability of default not only depends on the issuer but also on the characteristics of the bond (specifically the duration of it). And the effect of inserting this spread is lowering the value of the asset, so our solution will require more amount of money as is easily understandable.

From a computational point of view, the fundamental spread does not bring any complexity since the algorithms are exactly the same, but it extends the former analysis to a more complete financial approach. So, we can say that the algorithm we develop along this chapter, although it doesn't incorporate the fundamental spread, is in this sense, complete.

7.5 The Financial Balance Problem

The financial balance problem consists of choosing a portfolio of assets and blocking them just to match the liabilities. Thus, the insurer needs to manage the income arising from assets, invest it in the short term, and use the investment to pay the liabilities claimed by its customers. If its assets do not generate sufficient cash-flows, then the insurance company needs to borrow, which inflicts a penalty cost. Accordingly, it is possible to formulate the financial balance as follows, where $t = 1, 2, \ldots, T$ represents the time period:

$$S_0 = A_0 - L_0 \quad \text{and} \quad S_t = \begin{cases} S_{t-1} r_{t-1,i} + A_t - L_t & \text{if } S_{t-1} \geq 0 \\ S_{t-1}(r_{t-1,t} + \delta) + A_t - L_t & \text{if } S_{t-1} < 0 \end{cases} \tag{7.3}$$

In Equation (7.3) S_0 (S_t) is the capital in period 0 (t), A_0 (A_t) denotes the value of assets in period 0 (t), L_0 (L_t) denotes the value of liabilities in period 0 (t), and $r_{t-1,t}$ is the interest rate used to capitalise the resources from period t to $t-1$. Finally, δ represent the bid-ask spread on the interest rate. it is worth noting that Equation (7.3) is recursive, and the balance sheet sign determines if the bid or the ask interest rate is used to capitalise the resources until the next term. If the balance is negative, the company will need to borrow. As a result, it will need to pay the ask interest rate on the credit line, which will require more capital. Notably, the selected assets have an effect on the balance sign, creating a binary tree of 2^T nodes. Moreover, if the balance falls negative its size is restricted by the credit limit.

Based on the aforementioned, we are now in a position to outline an optimisation program that solves for the optimal choice of the assets and the associate weights to match the liabilities. On an individual basis, let A_t^j be the portfolio of the firm's assets, where super-index j refers to a particular asset, and the sub-index t refers to the cash-flow of asset A^j in period t. Let L_t be the cash-flow associated with liabilities in period t.

The goal is to select a portion α^j of each asset A^j with the following goal:

$$\min \sum \alpha^j PV(A^j) \tag{7.4}$$

where PV is the present value, which is computed using the term structure of interest rates. Moreover, the selection of assets is subject to the following constraints, where τ refers to the maximum credit line of the firm:

$$S_0 = A_0 - L_0 \tag{7.5}$$

$$S_n \geq 0 \tag{7.6}$$

$$\forall t \geq 1 \quad S_t >= -\tau \tag{7.7}$$

7.6 A Greedy Heuristic

In this section we propose a greedy heuristic that finds a selection of our assets, α^j. In the next section, this heuristic is extended into a biased-randomised algorithm, which allows to improve the solutions provided by the greedy heuristic. The heuristic constructs a feasible solution, one step at a time, by always choosing the 'best-next-move' in the short run (i.e., without taking into account the possible long-run implications of this selection). For that, we consider that the liability cash-flow can be estimated by aggregating individual cash-flows in each period of time. Then, we are interested in solving a simplified matching problem, which considers just the cash-flow associated with one of these liabilities; the specific liability is randomly selected. Once the chosen liability has been matched by a set of assets, a new liability cash-flow is randomly chosen and new assets (from the remaining ones) are drawn to cover it. This process is re-iterated until all the liabilities have been covered by asset cash-flows (Algorithm 3).

Notice that the first step in Algorithm 3 is to decide an order for the list of liability cash-flows. A natural order is the one given by the maturity date, so that the next-in-time cash-flow that will have to be payed is introduced first, the second one is

Algorithm 3 Greedy heuristic

```
Order liabilities by maturity date.
for each liability k do
   Insert k
   repeat
      for each asset j do
         Calc asset fraction needed to match k
         Select best asset so far, j*
      end for
   until Liability k is matched
end for
```

next, and so on. The selection of the best asset is quite simple, since we have to match only one liability cash-flow at a time. Hence, we only have to iterate over the remaining assets to get the minimum fraction needed to match our new liability. Only assets with a value larger than the current liability value are considered.

7.7 A Biased-Randomised Algorithm

By examining Algorithm 3, one can notice the following: once the order of the liabilities to be matched has been fixed, the solution (set of assets chosen to cover the liabilities) is unique. This suggests than one way to generate different solutions is by introducing a biased-randomised process when sorting the liabilities. To this end, we make use of a skewed probability distribution (the geometric one in our case) to re-order the liabilities list, hence using Montecarlo simulation to generate a differently ordered lists in each run of the algorithm. The geometric distribution only requires a parameter, $p \in (0, 1)$. As p converges to 1, the list tends to be sorted following the greedy criterion employed by the initial heuristic (i.e., by maturity date). On the contrary, as p converges to 0, the list tends to follow a uniformly random order. The values in between are the interesting ones, since they represent a compromise between a greedy and a uniform random order. Figure 7.4 shows the Java code employed to generate the biased-randomisation effect.

```
indexMax = L;
Random rnd = new Random();
for (int i = 0; i < L; i++)
{
    index = ((int)(Math.Log(rnd.NextDouble()) / Math.Log(1 - GeomPar))) % indexMax;
    LiabilityList[i] = RevertedLiability[indexMax - 1 - index];
    RevertedLiability.RemoveAt(indexMax - 1 - index);
    indexMax--;
}
```

Figure 7.4: Code for the Biased-Randomised Selection of Liabilities

7.8 Computational Experiments

In order to test our method, we have considered data from a real-life insurance firm. This firm holds 21 assets, which are predominantly government bonds and interest rate swaps. We have considered a discount rate of 1.09%, an interest rate to capitalise resources of 0.5%, and a time span of 33 years. For the geometric distribution, a parameter $p = 0.8$ has been selected after a quick trial-and-error process. Also, we have used 100 iterations, a maximum credit of 1 million euros, and assumed that the credit line carries a 5% interest rate. The origin of the liabilities are pensions, and their present value is 442 million euros.

The assets selected by the actuarial team add up to 490 million euros. Running our algorithm for a few seconds, we found a solution with an associated value of 450 million euros, which represents an 8% savings with respect the solution provided by the actuarial team. As shown in Figure 7.5, the solution structure is not trivial, so it is not surprising that it could not be found without the help of an algorithm as the one proposed here.

Testing different values for the parameter p does not seem to provide significantly better results. The fastest result is found if the original liability order is based on the present value of each liability cash-flow. This makes sense, as we first match the largest liability values with the best possible asset. Using this initial order criteria, only 100 iterations are necessary to get a high-quality solution.

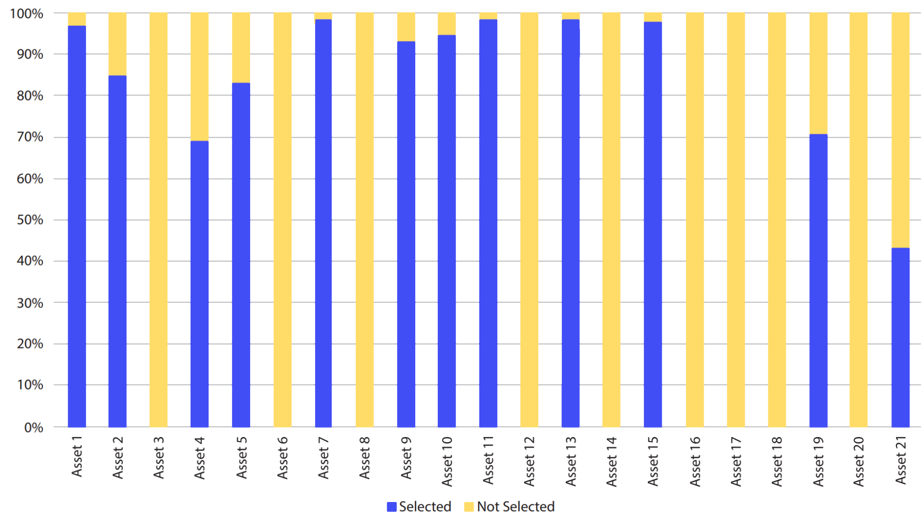


Figure 7.5: Solution Showing the Selection of Assets and Percentages

7.9 Conclusions

This chapter proposes a solving approach for the asset and liability management problem. Our algorithm makes use of Montecarlo simulation to transform a greedy heuristic into a probabilistic algorithm. The resulting biased-randomised algorithms is a fast and easy-to-implement method for selecting the minimum amount of assets to cover a portfolio of liabilities. Our method is flexible and it can be easily extended to new constraints, either if they provide from a specific regulation or from the firm's strategy. Our approach can be used in a real-life situation by iteratively applying it to a set of liabilities. According to our computational experiments, the savings it generates can be considerable. Considering that the insurance market is strongly regulated, having an efficient, flexible, and easy-to-implement method to select the proper assets inside a firm's portfolio is extraordinarily important.

As future work, we plan to: *(i)* extend our probabilistic algorithm into a full meta-heuristic one; and *(ii)* test the algorithm in more benchmark data sets –some of them using real-life data.

Chapter 8

The Multiperiod Risk Model - Markowitz Revisited

8.1 Introduction

One of the most successful theories in the field of finance was written by Markowitz in the last 50's. In essence, all the architecture of that theory is based on the hypothesis that any asset in the market behaves as a random variable with a well known mean and variance. Although we tend to think that the hypothesis is considering the referred behaviour as Gaussian random variable, the fact is that we don't need to follow any specific probability distribution provided that the central second moment of the distribution $f(x)$ is finite:

$$< x^2 >= \int_\Omega x^2 f(x)dx < +\infty \tag{8.1}$$

The Portfolio Optimisation Problem derived by Markowitz only needs the previous condition since the key of his idea is the diversification. In other words, let X and Y two random variables with finite variances σ_X and σ_Y and with correlation matrix σ_{XY}; then the random variable $aX + bY$ has a mean $a < X > +b < Y >$ but a variance $a^2\sigma_X^2 + b^2\sigma_Y^2 + 2ab\sigma_{XY} \leq (a\sigma_X + b\sigma_Y)^2$. The trick is in the last equation: if the mean is just additive but the variance is reduced, we can build better portfolios mixing independent assets.

Up to now, the relevant feature of the random variable that models our assets is the finite variance. In the derivative market we need an extra and strong hypothesis, which is to consider that the assets must follow the normal distribution. There are two reasons of why we suspect the inner behaviour has to be normal. It is the only distribution function that meets these two properties: a) a finite variance, and b) an stable one (it means that the sum of two random variables results in a random variable of the same nature) (Mantegna and Stanley, 1999). The real market seems to violate at least one of these two hypotheses. We think that we need to keep the stability, so the alternative is to consider that the distribution probability has no finite variance. Empirical evidence is found since real distributions are characterised by fatter tails than the Gaussian distribution. Nevertheless, we will continue assuming the Gaussian distribution along our research assuming that the differences of the fatter tails are assumable for our interests.

The Markowitz approach is also based on two more characteristics. The first one is that it drives a strategy of only one period, and the second one is that it only manages assets. In other words, if we have to match specific treasury conditions in the future, the Markowitz approach has to be extended. With treasure conditions we mean liabilities, i.e., those obligations of the company that are known a priori. The common Markowitz formulation uses proportions to refer the amount of money we invest in each asset. But when we include liabilities, we have to speak about raw money instead of proportions as we don't know a priori the value of the proportions but we have the price (cash) of the liability in a further moment of the time. To sum up, we have to transform the standard Markowitz model in an equivalent one with these characteristics:

- It has to be multiperiod.
- It has to contain liabilities in any moment of time.
- It has to refer to price, not to proportions.

8.2 The Model Definition

From now on, bold variables like $\mathbf{A}$ will refer to random variable, meanwhile A will mean average of the respective random variable. Otherwise, any variable is deterministic.

Following our pretensions of extending the standard optimisation portfolio problem, we define the Terminal Wealth as the amount of money we have at the expiry date of the transactions along the settled time. We have to determine how many monetary units we have to purchase or sell in each period of time and subtracting the liabilities associated to the referred moment. As the assets are stochastic, the Terminal Wealth is also a random variable. So, we need to know both mean W_T and variance $\sigma^2_{W_T}$ of the Terminal Wealth $\mathbf{W_T}$ to extend naturally the Markowitz model.

$$\text{Max } W_T \tag{8.2}$$

s.t.

$$\sigma_{W_T} \leq \sigma_{max} \tag{8.3}$$

To derive the expressions for W_T and $\sigma^2_{W_T}$ we will suppose that the risk-free interest rate r is constant along all time, the asset returns $\boldsymbol{\mu}_\alpha$ are normal distributions and so, stock prices are Log-Normal distributions. All the asset returns are correlated and we have the Variance-Covariance matrix $\sigma_{\alpha\beta}$, assuming that this matrix will remain constant along the time. We recall that we have T periods and we will use Greek letters to index the assets and Roman letters to refer to each period of time.

At the beginning of each period i, a known number n_i^α of stock purchases/sells transactions are done with the asset α. All the transactions conclude at the period T and we will sell all the remaining assets at the end of the expiry date. The amount of money in that moment will be the Terminal Wealth.

The price of each asset at the end of each period is $\mathbf{P}_i^\alpha$ and it is related with the price at the beginning of the period as follows:

$$\mathbf{P}_i^\alpha = \mathbf{P}_{i-1}^\alpha e^{\boldsymbol{\mu}_\alpha} \tag{8.4}$$

So, recursively up to the beginning, we can formulate the price of any asset in terms of the deterministic initial price P_0 and the sum of all the returns from the beginning up to the period i:

$$\mathbf{P}_i^\alpha = P_0^\alpha e^{\boldsymbol{\mu}_\alpha^1 + \boldsymbol{\mu}_\alpha^2 + \ldots \boldsymbol{\mu}_\alpha^i} \tag{8.5}$$

Since the asset return $\boldsymbol{\mu}_\alpha$ is considered as a normal random variable, the sum of i normal variables is also a normal one. Let's take into account that if the probability distribution function $\mathbf{Y} = exp(\mathbf{X})$ is the Log-Normal, and so, $\mu_Y = exp(\mu_X + \frac{1}{2}\sigma_X^2)$, and $\sigma_Y^2 = exp(\sigma_X^2) - 1$, we can conclude that for $\mathbf{P}_i^\alpha$ we can express its mean and variance as follows:

$$P_i = P_0 e^{i(\mu + \frac{1}{2}\sigma^2)} \tag{8.6}$$

$$\sigma_i^2 = P_i^2 \left(e^{i\sigma_\alpha^2} - 1 \right) \tag{8.7}$$

We can define the effective return μ_α^{eff} for the asset α as:

$$\mu_\alpha^{eff} = \mu_\alpha + \frac{1}{2}\sigma_\alpha^2 \tag{8.8}$$

We also define the transaction value T_i^α of asset α at the period i, valuated at the end of the transaction, as:

$$T_i^\alpha = P_\alpha \; e^{i\mu_\alpha^{eff} - ir + Tr} n_i^\alpha \tag{8.9}$$

And we define the temporal Variance-Covariance Matrix $\widetilde{\sigma}$ as:

$$\widetilde{\sigma}_{\alpha\beta}^{ij} = e^{(T - \max(i,j))\sigma_{\alpha\beta}} - 1 \tag{8.10}$$

Having our portfolio of assets along the time and all of them are in general correlated, W_T is the Terminal Wealth, r is the risk-free interest rate, T is the number of periods, and L_i is the obligation we have to fulfil in period i, the Terminal Wealth and the Terminal Wealth Variance follows the next expressions:

$$W_T = \sum_{\alpha} \sum_{i=0}^{T} \left(T_i^{\alpha} - L_i e^{(T-i)r} \right) \tag{8.11}$$

$$\sigma_{w_T}^2 = \sum_{\alpha,\beta} \sum_{i,j=0}^{T} T_i^{\alpha} \tilde{\sigma}_{\alpha\beta}^{ij} T_j^{\beta} \tag{8.12}$$

The Terminal Wealth is the sum of all transactions valuated at the maturity date, using the effective return to capitalise each asset, minus all the liabilities capitalised according to the risk-free interest rate.

In the next sections we discuss and demonstrate the former expression for the Terminal Wealth Variance.

8.3 Theorem: the Terminal Wealth Variance

Let Ω the set of assets we have in our market and $\alpha \in \Omega$ and $\beta \in \Omega$ be two assets with returns following a normal distribution with mean μ^{α} and μ^{β} respectively, and $\sigma_{\alpha\beta}$ as the variance-covariance matrix. We will also suppose that the returns are not autocorrelated and homoscedastic along the time. In other words, the returns stay unaltered along the time and they don't have memory of the value of the previous periods.

Let $P_i^{\alpha} \in \mathbb{R}$ be the known price of $\alpha \in \Omega$ in the period i. This known price corresponds to the amount of money we will transact with that asset in that period. In our general optimisation problem, the set of these prices are our decision variables. It implies that we will have to purchase or sell an unknown amount of stock n_i^{α} because we cannot assure what price the asset will have in that moment since it follows a Log-Normal random walk.

Let $\mathbf{P}_{i,T}^{\alpha} = P_i^{\alpha} \prod_{k=i}^{k<T} e^{\boldsymbol{\mu}_k^{\alpha}}$ be the terminal random price corresponding to the random walk of the normal returns. As said before, the return used to build this random variable has to be not autocorrelated.

Thus, for two terminal prices $\mathbf{P}_{i,T}^{\alpha}$ and $\mathbf{P}_{j,T}^{\beta}$ the covariance is:

$$\sigma_{\mathbf{P}_i^{\alpha}\mathbf{P}_j^{\beta}} = P_{i,T}^{\alpha} P_{j,T}^{\beta} \left(e^{(T-\max(i,j))\sigma_{\alpha\beta}} - 1 \right) \tag{8.13}$$

where $P_{i,T}^{\alpha} = ...$ is the mean value of the random variable $\mathbf{P}_{i,T}^{\alpha}$.

Demonstration

We can write the covariance $\sigma_{\alpha\beta}$ with the well-known expression:

$$\sigma_{\mathbf{P}_{i,T}^{\alpha}\ \mathbf{P}_{j,T}^{\beta}} = E\left(\mathbf{P}_{i,T}^{\alpha}\ \mathbf{P}_{j,T}^{\beta}\right) - E\left(\mathbf{P}_{i,T}^{\alpha}\right) E\left(\mathbf{P}_{j,T}^{\beta}\right) \tag{8.14}$$

First, we calculate the first member of the above equation. Each asset follows a Log-Normal distribution, so $\mathbf{P}_i = \mathbf{P}_{i-1} e^{\boldsymbol{\mu}}$ for each asset, where $\boldsymbol{\mu}$ is the return random variable. Therefore, applying the definition of the terminal price we have that:

$$E\left(\mathbf{P}_{i,T}^{\alpha}\mathbf{P}_{j,T}^{\beta}\right) = E\left(P_i^{\alpha} P_j^{\beta} e^{\sum\limits_{k=i}^{k<T} \boldsymbol{\mu}_k^{\alpha} + \sum\limits_{l=j}^{l<T} \boldsymbol{\mu}_l^{\beta}} \right) = P_i^{\alpha} P_j^{\beta} E(e^{\boldsymbol{\mu}_{i,j}^{*}}) \tag{8.15}$$

Where the random variable $\boldsymbol{\mu}_{i,j}^{*} = \sum\limits_{k=i}^{k<T} \boldsymbol{\mu}_k^{\alpha} + \sum\limits_{l=j}^{l<T} \boldsymbol{\mu}_l^{\beta}$ is a normal variable because it is the sum of normal variables. Thus, we can calculate its mean:

$$E\left(\boldsymbol{\mu}_{i,j}^{*}\right) = (T-i)\mu^{\alpha} + (T-j)\mu^{\beta} \tag{8.16}$$

In order to calculate the variance of $\boldsymbol{\mu}_{i,j}^{*}$ we have to take into account that the return $\boldsymbol{\mu}$ is a not autocorrelated random variable, therefore, $\sigma_{\mu_i^{\alpha},\mu_j^{\beta}} = \sigma_{\alpha\beta}$ with $i = j$

$\forall \alpha \neq \beta \in \Omega$, otherwise 0. So, we can write that result in a compact way:

$$\sigma_{\mu_i^\alpha,\mu_j^\beta} = \sigma_{\alpha\beta}(1-\delta_{\alpha\beta})\delta_{i,j} \tag{8.17}$$

Where δ_{ab} is the Kronecker delta that is equal to 1 if $a = b$, otherwise is 0.

With that result, we can calculate the variance of $\boldsymbol{\mu}^*_{i,j}$:

$$Var\left(\boldsymbol{\mu}^*_{i,j}\right) = \sum_{k=i}^{k<T}\sigma_\alpha^2 + \sum_{l=j}^{l<T}\sigma_\beta^2 + 2\sum_{k=i}^{k<T}\sum_{l=j}^{j<T}\sigma_{\alpha\beta}(1-\delta_{\alpha\beta}\delta_{i,j}) \tag{8.18}$$

Now, we can solve the sums. To solve easily the double sum, we have to consider only the upper index of both assets because lower indexes don't find a pair in the other asset. So, the only terms that are different from zero are those with $index \geq max(i,j)$. At the same way as the other two terms, we have $T - max(i,j)$ terms other than zero. So:

$$Var\left(\boldsymbol{\mu}^*_{i,j}\right) = (T-i)\sigma_\alpha^2 + 2(T-\max(i,j))\,\sigma_{\alpha\beta} + (T-j)\sigma_\beta^2 \tag{8.19}$$

Knowing that the expected value of a Log-Normal random variable e^μ is $e^{\mu+\frac{1}{2}\sigma^2}$, and using the last result to 8.15 we have that:

$$\begin{aligned} E\left(\mathbf{P}^\alpha_{i,T}\mathbf{P}^\beta_{j,T}\right) &= \\ &= P_i^\alpha P_j^\beta e^{(T-i)\mu^\alpha+(T-j)\mu^\beta+\frac{1}{2}\left((T-i)\sigma_\alpha^2+2(T-\max(i,j))\sigma_{\alpha\beta}+(T-j)\sigma_\beta^2\right)} \\ &= E\left(\mathbf{P}^\alpha_{i,T}\right)E\left(\mathbf{P}^\beta_{j,T}\right)e^{(T-\max(i,j))\sigma_{\alpha\beta}} \end{aligned} \tag{8.20}$$

because $E(\mathbf{P}^\alpha_{i,T}) = P_iE(Exp(\sum_{k=i}^{k<T}\boldsymbol{\mu}_k)) = P_ie^{(T-i)(\mu+\frac{1}{2}\sigma^2)}$.

And finally, we have the final result by subtracting $E\left(\mathbf{P}^\alpha_{i,T}\right)E\left(\mathbf{P}^\beta_{j,T}\right)$:

$$\sigma_{\mathbf{P}_i^\alpha\mathbf{P}_j^\beta} = P^\alpha_{i,T}P^\beta_{j,T}\left(e^{(T-\max(i,j))\sigma_{\alpha\beta}} - 1\right) \tag{8.21}$$

with $P^{\alpha}_{i,T} = E(\mathbf{P}^{\alpha}_{i,T}) = P^{\alpha}_{i} e^{(T-i)(\mu+\frac{1}{2}\sigma^2)}$

Corollary

Eventually, the variance of the terminal wealth W_T generated for a set of asset transactions along T periods of time is:

$$\sigma^2_{w_T} = \sum_{i,j=1,\alpha,\beta}^{i,j=T} P^{\alpha}_{i,T} P^{\beta}_{j,T} \left(e^{(T-\max(i,j))\sigma_{\alpha\beta}} - 1\right) \tag{8.22}$$

where

$$P^{\alpha}_{i,T} = P^{\alpha}_{i} e^{(T-i)(\mu_\alpha+\frac{1}{2}\sigma^2_\alpha)} \tag{8.23}$$

is the expected price of asset α at the period T whose price in the period i is P^{α}_{i}.

8.4 Conclusions

Analysing the former expression we get these conclusions:

- The expression is formally a classical variance-covariance formula if we interpret each transaction as an independent asset itself.
- Although the returns are not autocorrelated, the prices do. It is easy to understand this because prices have memory!
- Each individual asset (i.e., each transaction) has a correlation with each other, even if it belongs to the same kind of stock. In that case, the factor $(T - \max(i,j))\sigma_{\alpha\alpha}$ is the variance of the asset α times the number of periods that both transactions (regarding to the same asset) have concurred along the time.
- The factor $e^{(T-\max(i,j))\sigma_{\alpha\beta}} - 1$ comes from considering the price as a Log-Normal distribution. If we consider a few periods and small variance-covariance numbers, we recuperate the standard matrix: $e^{(T-\max(i,j))\sigma_{\alpha\beta}} - 1 \approx (T - \max(i,j))\sigma_{\alpha\beta}$.

- The values of wealth and volatility are higher than if we consider prices normally distributed. It is necessary to consider Log-Normal behaviour to get better predictions of the market. Moreover, considering prices normally distributed would break the necessary homogeneity of the market as prices would be the product of normal distributions, and that is not possible. For instance, if $P_2 = P_0 exp(\mu_1) exp(\mu_2)$ and we consider each exponential behaving as a normal distribution, the product is not a normal, but a Bessel function. So, we wouldn't guarantee the necessary stability property for the probability distribution function for prices.

To sum up, in this chapter we have developed a natural extension of the Markowitz portfolio optimisation problem. We have to consider a multiperiod scenario since in a ALM context the optimisation problems generally run in the mid and the long term. Obviously, this problem is harder to solve computationally. Even, if we add some constraints, the size of the problem (recall that is an equivalent problem to $N \times T$ assets), joint to the complexity of the extra constraints suggests that metaheuristics and simheuristics have to go on stage.

Chapter 9

A GA-Simheuristic for the Stochastic and Multi-Period ALM

Abstract

The efficient management of assets to cover a firm's liabilities over a multi-period horizon is a relevant challenge for many banks and insurance companies, and one which can generate significant benefits when an optimal / near-optimal investment plan is found. Even in its deterministic version, this problem is complex in nature, since managers have to make difficult decisions about their portfolio of assets at each period. With the goal of maximising the expected terminal wealth in a scenario under uncertainty, this chapter proposes a novel simheuristic approach that integrates Montecarlo simulation at different stages of a Genetic Algorithm. Our approach is capable of generating effective solutions to the considered problem in relatively short computational times. In addition, our simheuristic is enriched with several 'smoothing' techniques that enhance the attractiveness for managers of the generated solutions, so they can be effectively employed in real-life applications. A series of computational experiments, including the use of advanced evolutionary strategies, contribute to illustrate these concepts and to justify the advantages of including simulation in financial optimisation problems under uncertainty.

9.1 Introduction

Dealing with liabilities (financial obligations) in volatile markets with a limited yield or return is one of the main challenges that insurance firms have to face. The insurer is forced to pay the amounts agreed in the policy at a specific maturity date. In order to do that, a set of firm's assets have to be 'frozen' in advance to cover future payments. Asset-liability management (ALM) refers to the study of techniques employed in selecting the appropriate assets to face the firm's liabilities over time. The financial market is constituted by a huge number of companies, whose aim is to transform an initial wealth into large returns during a giving time horizon. According to the consumer preference theory (Mankiw et al., 2007), an investor would select those assets that provide the highest returns, while taking into account her budget constraint. As markets are plenty of uncertainty, the volatility of the assets also has to be considered. This transforms the ALM into a stochastic and multi-period portfolio optimisation problem. Markowitz (1952) considered the assets as random variables, so he formulated the classical mean-variance model, in which different amounts of assets have to be selected in order to maximise a portfolio's return, while considering a specific volatility. Alternatively, one might want to minimise the risk subject to achieving a user-defined level of return (Kizys et al., 2019b). In any case, these approaches are only valid if our wealth is not associated with a set of liabilities. Whenever it is, we need to consider a different strategy, since the obtained returns are employed to cover liabilities. In general, the purpose of ALM is to support the assets selection process –i.e., by selecting those that maximise returns while maintaining enough financial resources to satisfy the liabilities. Among the different ALM approaches in the scientific literature, the following ones are the most popular ones: *(i)* duration theory, which is based on the work of Macaulay (1938), who assumes that the interest rate is almost constant and also that assets and liabilities have the same present value; *(ii)* cash-flow matching, where we select assets in a way that allows us to match them with our debts; and *(iii)* stochastic control theory, a quite theoretical approach that studies the evolution of assets and liabilities in a continuous and stochastic scenario.

This work focuses on the cash-flow matching strategy, which extends the Markowitz's theory to an ALM scenario. Hence, we will consider a realistic mean-variance problem in a multi-period context under uncertainty, where our decision variables are the amount of assets we have to buy or sell in each period of time, considering an initial wealth and a given set of liabilities along time. In other words, given an initial wealth and a set of financial duties that need to be covered in the future, the goal is to find the ALM plan that maximises our expected wealth at the end of the time horizon, taking into account different uncertainty sources. In order to solve the aforementioned stochastic optimisation problem, we propose a novel simheuristic approach (Juan et al., 2015b) that integrates Montecarlo simulation (Carsey and Harden, 2013) at different stages of a Genetic Algorithm or GA (Kramer, 2017). Depending on the specific application, simheuristics can use Montecarlo simulation, discrete-event simulation, agent-based simulation, or a combination of different simulation approaches, thus leading to hybrid simulation methods (Brailsford et al., 2019). Our approach is capable of generating effective solutions in relatively short computational times. In addition, our methodology is enriched with several 'smoothing' techniques that enhance the attractiveness for managers of the solutions. While simulation is a tool frequently employed in financial studies (Huang and Willemain, 2006; Khabibullin et al., 2020), and the same can be said for GAs (Soler-Dominguez et al., 2017; Doering et al., 2019b), the combination of both has been rarely explored in the ALM literature. Simheuristics, however, have been successfully employed to solve stochastic optimisation problems in transportation (Gonzalez-Martin et al., 2018), logistics (Quintero-Araujo et al., 2019b), and telecommunication systems (Alvarez Fernandez et al., 2021). Still, this is the first time they have been used to solve the stochastic and multi-period ALM problem. Also, while most of the simheuristic approaches make use of trajectory-based metaheuristics (Grasas et al., 2016), the one presented in this article makes use of a population-based metaheuristic. Since they combine simulation with optimisation methods, simheuristics can be considered as a particular case of hybrid models, which are particularly useful to promote transdisciplinary research actions (Tolk et al., 2021).

The remainder of the chapter is structured as follows: Section 9.2 introduces the mathematical model that provides a formal description of the considered ALM problem. Section 9.3 explains the simheuristic approach proposed to solve the ALM problem. Section 9.5 includes a series of computational experiments. These experiments, together with the 'white-box' explanation of the hybrid methodology employed, allow managers to build trust in our approach (Harper et al., 2021). Finally, Section 9.6 highlights the main conclusions of this work and points out some open research lines.

9.2 A Formal Description of the Multi-Period ALM Problem

As already mentioned, the typical goal of the ALM problem consists in maximising terminal wealth subject to a initial wealth w_0. The investor has to decide, in each period of time, what amount of money is invested, sold, or held for each class of asset in order to satisfy the scheduled liabilities on time. There is also the possibility of borrowing money for those circumstances in which we do not have enough cash to pay our liabilities (negative cash-flows). When considering the stochastic behaviour of assets, a set of possible scenarios S appear, each weighted by a probability. These probabilities are used to calculate the value of the manager's utility function (e.g., the firm's final wealth) at the expiry time τ of the planning horizon. We define the following parameters and variables:

	Parameters
S	set of possible scenarios
s	a particular scenario. Each scenario is characterised by a particular value of each asset
A	set of assets, not including the risk-free asset indexed 0
U	the Utility function of the investor
$p(s)$	the probability of the scenario s
τ	expiry time of the operation
β	the borrowing rate
ρ^s_{it}	the yield of asset i in time t at scenario s
x_{it}, x^s_{it}	the amount of money we have in asset i at time t (in scenario s when scenarios are considered)
w_τ, w^s_τ	the terminal wealth (in scenario s when scenarios are considered)
	Variables
p^s_{it}	the purchase of asset i in time t at scenario s
d^s_{it}	the amount of sold asset i in time t at scenario s
b^s_t	the amount of borrowed money in time t at scenario s

Table 9.1: Parameter and Variable Definitions

Following Mulvey et al. (1997), and using the previously defined variables, we have the standard deterministic and multi-period ALM model. Objective (9.1) is to maximise the (subjective) utility $U(w^s_\tau)$ of the terminal wealth (w^s_τ) attained over a set S of financial scenarios, where τ is the number of time periods taken as the planning horizon in each scenario:

$$Maximise \sum_{s \in S} p(s) U(w_\tau^s) \tag{9.1}$$

Constraint (9.2) defines the initial wealth (w_0^s) available for investment in each scenario:

$$\sum_{i \in A} x_{i0}^s = w_0^s, \ \forall s \in S. \tag{9.2}$$

Constraint (9.3) defines the terminal wealth (w_τ^s), in each scenario s, as sum of the values of all investments at the end of the planning horizon:

$$\sum_{i \in A} x_{i\tau}^s = w_\tau^s, \ \forall s \in S. \tag{9.3}$$

Constraint (9.4) updates the value of an investment after one period of time, accounting for the interest rate in the previous time period, $\rho_{j(t-1)}$, any addition investment in the asset in the current time period (p_{jt}^s), or amount of the asset sold in the current time period, d_{jt}^s:

$$x_{jt}^s = (1 + \rho_{j(t-1)}^s) x_{j(t-1)}^s + p_{jt}^s - d_{jt}^s, \ \forall j \in A, \ \forall t \in \{1..\tau\}, \ \forall s \in S. \tag{9.4}$$

Constraint (9.5) updates the amount of liquid assets/cash after one time period, accounting for the cash interest rate, $\rho_{0(t-1)}^s$, the amounts of each investment sold/cashed in, $\sum_{j \in A} d_{jt}^s$, the amounts of each investment purchased, $\sum_{j \in A} p_{jt}^s$, any loans taken and interest paid on the loan, minus the cash liabilities (l_t) due in that time period:

$$x_{0t}^s = (1+\rho_{0(t-1)}^s) x_{0(t-1)}^s + \sum_{j \in A} d_{jt}^s - \sum_{j \in A} p_{jt}^s - b_{t-1}^s (1+\beta_{t-1}^s) + b_t^s - l_t, \ \forall t \in \{1..\tau\}, \ \forall s \in S. \tag{9.5}$$

We have to consider the possibility of having to borrow money due to: *(i)* at a certain time, there is not enough cash to cover the liabilities; or *(ii)* it can be inconvenient to exchange some assets for cash. In the previous model, l_t is the liability cash-flow at time t. One more equation is needed in order to ensure that liquid assets are never negative:

$$x_{0t}^{s} \geq 0,\ \forall t \in \{1..\tau\},\ \forall s \in S. \tag{9.6}$$

In our approach, we will consider the stochastic behaviour of assets by assuming that the return will follow a Gaussian distribution. Therefore, the price obeys to a Log-Normal distribution. We will also work with continuous interest rates. Having included the stochastic nature of assets in the model, scenarios are not needed anymore. Still, it is necessary to reformulate the former equations, so they include the risk aversion of the investor. The volatility of the final terminal wealth is considered as the measure of risk. Thus, the objective (9.7) will be to maximise the total wealth accrued in a time period τ:

$$Maximise\ w_{\tau} \tag{9.7}$$

Constraints (9.8) and (9.9) state that all the initial wealth is in our current account, and that, initially, no money is invested in any of the available investments:

$$x_{00} = w_0. \tag{9.8}$$

$$x_{j0} = 0,\ \forall j \in A. \tag{9.9}$$

Constraints (9.10) and (9.11) state that, at the last period, we have to sell every asset and accumulate all the wealth in the current account –since we have to refund the money to the investor:

$$x_{0\tau} = w_{\tau}. \tag{9.10}$$

$$x_{j\tau} = 0,\ \forall j \in A. \tag{9.11}$$

Constraint (9.12) describes the value evolution of asset j along time. As described above, it accounts for return rate of the investment, any additional investments made, or the amount of that asset sold in the given time period:

$$x_{jt} = x_{j(t-1)} e^{\mu_j + \frac{1}{2}\sigma_j^2} + p_{jt} - d_{jt},\ \forall j \in A,\ \forall t \in \{1..\tau\}. \tag{9.12}$$

Constraint (9.13) describes the cash-flow evolution of our account. In this equation, we consider not only the free interest rate and the purchases and sales of assets, but also transaction costs, cash loaned, loan interest payments, and any liabilities that must be covered:

$$x_{0t} = x_{0(t-1)}e^{\mu_0} + \sum_{j\in A} d_{jt} - \sum_{j\in A} p_{jt} - b_{t-1}e^{\beta_{t-1}} + b_t - l_t - \sum_{j\in A} d_{jt}c_j^d - \sum_{j\in A} p_{jt}c_j^p,\ \forall t \in \{1..\tau\}\,. \tag{9.13}$$

Constraint (9.14) states that we cannot have negative money, since financial authorities impose fines otherwise:

$$x_{0t} \geq 0,\ \forall t \in \{1..\tau\}\,. \tag{9.14}$$

Constraint (9.15) states that purchasing and selling an asset at the same time are mutually exclusive events (either you buy or you sell):

$$d_{jt}p_{jt} = 0,\ \forall j \in A,\ \forall t \in \{1..\tau\}\,. \tag{9.15}$$

Constraint (9.16) introduces the variable u_{jt} to represent the transaction amount in investment j at time t, which is positive in cases when we divest in asset j and negative when we invest in asset j. This variable captures buying and selling events in a single variable, and is used in the following equation:

$$u_{jt} = p_{jt} - d_{jt},\ \forall j \in A,\ \forall t \in \{1..\tau\}\,. \tag{9.16}$$

Equation (9.17) is the risk condition. This condition is the mathematical extension of the Markowitz's model considering that the asset behaves as a Log-Normal random variable. Although the expression seems quite complex, we can recognise two terms like $u_{it_1}e^{(\tau-t_1)(\mu_i+\frac{1}{2}\sigma_i^2)}$ which are the expected values of each transaction at the maturity date, and $e^{(\tau-\max(i,j))\sigma_{\alpha\beta}-1}$ which is the correlation between two transactions in different moments of the time, valued at the maturity date:

$$\sigma^2_{w_\tau} = \sum_{i\in A}\sum_{j\in A}\sum_{t_1\in\{1..\tau\}}\sum_{t_2\in\{1..\tau\}} u_{it_1} e^{(\tau-t_1)(\mu_i+\frac{1}{2}\sigma_i^2)} u_{jt_2} e^{(\tau-t_2)(\mu_j+\frac{1}{2}\sigma_j^2)} \left(e^{(\tau-\max(i,j))\sigma_{\alpha\beta}}-1\right)$$
$$\sigma^2_{w_\tau} \leq \sigma^2_{MAX}. \tag{9.17}$$

Constraint (9.18) describes a realistic limitation of the loan, i.e.: using external aid to finance cash-flow defaults is usually channelled by means of a mortgage or a credit policy –as with any financial tool, there is always a limitation of funds that can be used:

$$b_t \leq M \tag{9.18}$$

Following the Lagrange's theory of multipliers (Bertsekas, 2014), we can substitute the Equations (9.7) and (9.17) by the following one:

$$Maximize\ U = w_\tau - \frac{1}{2}\lambda\sigma^2_{w_\tau}. \tag{9.19}$$

Where λ is the Arrow-Pratt aversion index (Arrow, 1974; Pratt, 1978). Equation (9.13) deserves some attention. It is an obvious market condition that $\beta > \mu_0$. In other words, the interest of the loan is higher than the return of the free risk asset. In that case, there will not be any situation where we take the money of a loan to invest it in a current account. Thus, we can treat both $x_{0,t}$ and b_t as a unique account, such that if the balance is positive we will have a return of μ_0, while if it is negative we will have a return of β. Notice that, in that case, Equation (9.14) will be $x_{0t} \geq -M$, $\forall t \in \{1..\tau\}$.

9.3 Our GA-Simheuristic Approach

The proposed approach to solve the stochastic and multi-period ALM problem is based on a novel simheuristic Chica et al. (2020), which integrates Montecarlo simulation at different stages of a GA (Kramer, 2017). By doing so, the simulation component is not only used to provide estimates of the expected terminal wealth under uncertainty conditions, but it is also employed to guide the GA while the latter is searching for an efficient investment plan. As discussed in Chica et al. (2020), simheuristics combine heuristic-based optimisation with simulation. They have been recently applied in solving stochastic optimisation problems , mainly in the areas of logistics (Pages-Bernaus et al., 2019; Gruler et al., 2018), transportation (Guimarans et al., 2018; Onggo et al., 2019), smart cities (Gruler et al., 2017b), production (Hatami et al., 2018), and finance (Panadero et al., 2020). They have also been used to extend existing metaheuristic frameworks, such as the Iterated Local Search (Grasas et al., 2016) or the GRASP (Ferone et al., 2019). In this paper, it is employed to extend a GA framework, so that a stochastic financial optimisation problem can be efficiently solved. Roughly speaking, our GA considers a number of feasible solutions (individuals) that evolve at each iteration (generation). Hence, at each generation, new offspring solutions emerge as a result of combining pairs of individuals (parents). Also, at each generation, only the best solutions are kept for the next one, while poor-performance individuals are discarded (notice that this process mimics a natural evolutionary strategy). For each offspring, some alterations (mutations) can be applied in order to generate a new variety of feasible solutions. This procedure continues until a given number of generations has been explored or until the algorithm reaches a certain equilibrium (e.g., new offsprings are not able to improve upon the parents anymore).

In a realistic model, transaction costs might be not negligible. Hence, our GA has been designed to account for these costs. This is achieved by adding the transaction costs to the utility function, which is the reference value employed to select the survivors in each generation. Extending this concept, we can consider that any additional regulation associated with transactions can be treated also as a transaction cost, e.g.: if a soft regulation is not fully satisfied, a credit must be required to pay a penalty fee. Each individual (one possible solution) in the GA population contains a list of periods, where the levels of each asset are registered. The GA has three key factors: the initial population, the crossover between two parents, and the mutation of an individual. To build a primitive but feasible individual, we loop along all periods, considering if the wealth at the start of each period is positive or negative. To select the initial relative weight of each asset in each period, a simple rule is utilised: if the wealth before transactions is negative, we will only use our credit policy, thus forcing the algorithm to sell all the non-cash assets; if it is positive, we assign a random number for each asset, taking into account that the loan will not be used, i.e., the loan is assigned to asset 0, which is the cash. The wealth in each period is calculated recursively, so the wealth of one period is transferred to the next one by capitalising the position of each asset with its own factor. Notice that we avoid using the loan when the initial wealth in a period is positive. Otherwise, the GA would have to find solutions that compensate the extra cost of using the loan, which would make its task increasingly difficult due to the high interest rate of these long-term loans. Figure 9.1 describes the mechanism we use to cross two individuals. Each box represents the genetic code of each person, as described before. To create an offspring, the following actions are carried out: *(i)* we select a random period, taking the information of relative weights of the first individual until that period; and *(ii)* we complete the information using the remaining periods of the other individual. Once a new offspring has been generated, we mutate all the relative weights adding a positive, negative, or null noise. After that, we re-normalise the relative weights so they add up to one.

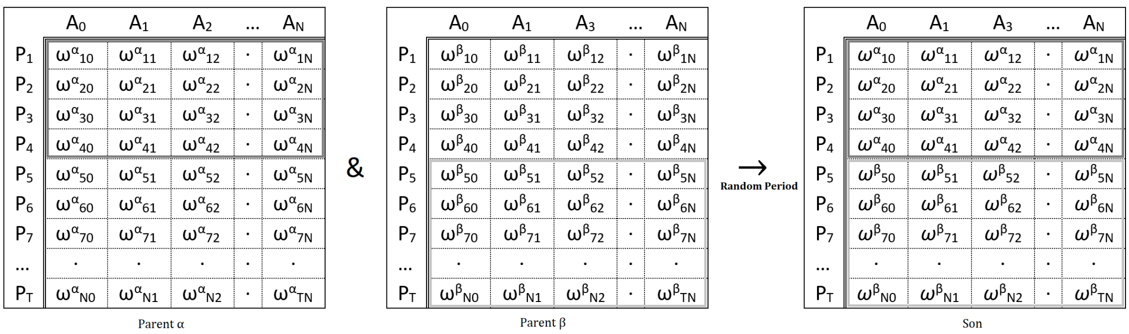


Figure 9.1: Crossover Operator

The proposed GA works independently of how the utility (fitness) function is evaluated. Actually, we propose two ways to compute this utility function. On the one hand, it can be quickly assessed by employing an analytical expression. However, this strategy does not allow us to fully consider the scenario under uncertainty. On the other hand, we can evaluate it via simulation, which allows us to fully consider the stochastic behaviour of assets. The utility function is composed of two elements, W_T (terminal wealth) and $sigma^2$ (the risk). The first one has been computed via simulation, and it represents the terminal wealth that we aim to maximise. The second element is still computed using an analytical expression.

Figure 9.2 presents a flow-chart of the simheuristic algorithm. The "Population Growth" box represents the procedure where all parents are mixed to increase the population, following the crossover function described before. Once the population has been created, we select a set of individuals with high values of the utility function. According to Equation (9.19), the fitness (utility) function has two components. On the one hand, the positive terminal wealth. On the other, the risk component (terminal variance), which is negative. In this work, we have run two versions of the GA. The first one uses the exact mathematical model introduced in the previous section. It means that both the terminal wealth and the risk level are analytically evaluated. In the second one (the simheuristic approach), we compute the expected terminal wealth using simulation, while the risk is analytically evaluated. Thus, in order to estimate the expected terminal wealth, we have simulated the time-path evolution of each individual until the end (maturity date), recreating a tree of future values in each asset and following the assumed Log-Normal behaviour. Both GA versions are compared in the next section.

In addition, we enrich the GA-simheuristic with a debugging procedure. For every asset that changes it sign (from buying to selling or vice versa), this procedure reviews all pairs of transactions associated with it, and tests if both transactions can be regrouped in the first one, while leaving the second one to zero. Such a situation is common in assets that are not relevant: the GA selects a transaction for a given asset in period t, and chooses the opposite transaction for that asset in period $t+1$ without the objective function being affected by any of these transactions. This contradictory decisions are difficult to understand for a manager and, therefore, the algorithm tries to remove them to 'refine' the solution.

Apart from using parallelisation techniques to speed up computational times, we also create a chain of 'civilisations', i.e.: islands of populations that only interact from time to time. Thus, after completing a civilisation, several procedures are run.

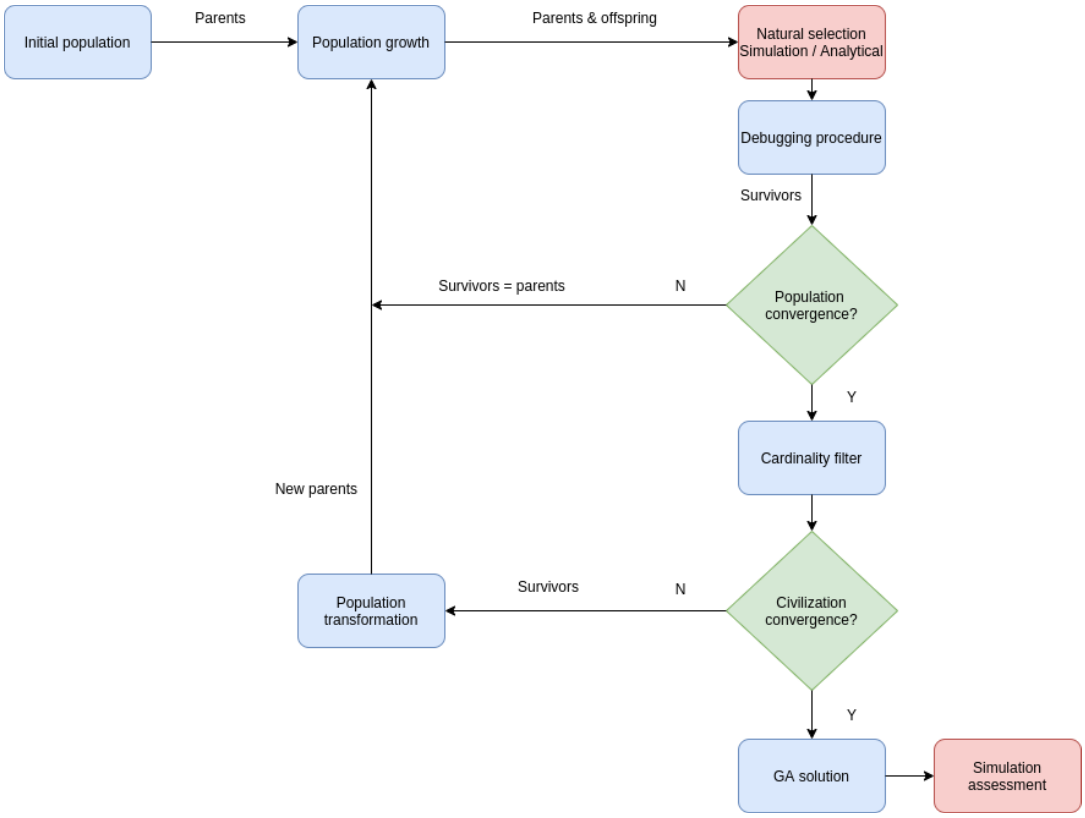


Figure 9.2: Flow-Chart of the Proposed GA-Simheuristic

First, assets are classified in terms of importance. Then, we compute the sum of the transactions along time, and verify if there are assets with a gap of 0.5% or lower with respect to the most important asset. These non-relevant assets are deleted from the final solution in order to simplify it and make it easier-to-understand for managers. In order to get a new civilisation, we take the population of the previous civilisation and smooth the transactions with a moving average of three points. Apart from that, we randomly perturbate their relative weights, so that the algorithm starts from a different point. Although this initial point might be worse than the current one, starting from a different point reduces the chances of getting trapped into a local minimum.

9.4 Classes, Structures and Pseudo-Code

The pseudo-code *4 - Classes and Data Structures* provides the programming classes and structures (structs) of the proposed GA. Each individual (solution) in the GA population contains a list of periods where the levels of each asset are registered. For each period and asset, we start with an initial wealth, *WealthBeforeTransact*. The *RelativeWeight* is the fraction of the total wealth in this period that is assigned to a specific asset before any transaction (asset buying or selling) takes place. Hence, once we buy or sell, we will have a *WealthAfterTransact*, which is defined by the level of each asset after the transaction. The difference between *WealthAfterTransact* and *WealthBeforeTransact* is the transaction cost. Positive numbers indicate buying assets. Each individual also has the essential descriptive values, i.e.: *TerminalValue*, *TerminalVariance*, and the utility value which is the value used as the fitness of our GA. The collection of individuals is gathered in the class *Civilisation*.

Algorithm 4 Classes and Data Structures

```
public class Civilisation
    public Person[] persons
    ...
end
class Person
    double utility ← 0
    double TerminalWealth ← 0
    double TerminalVariance ← 0
    Period[] periods ← 0
end
class Period
    ...
    Asset[] assets
    double TotalWealthBeforeTransact
    ...
end
struct Asset
    int TypeStock
    double WealthBeforeTransact
    double WealthAfterTransact
    double Transaction
    double RelativeWeight; // random number between 0 and 1
    ...
end
```

The GA has three key factors: the initial population, the crossover between two parents, and the mutation of an individual. Figure 9.3 describes the creation of the first individuals. To build a primitive but feasible individual, we loop along all periods, considering if the wealth at the start of each period is either positive or negative. To select the initial relative weight of each asset in each period, a simple rule is utilised: if the wealth before transactions is negative, we will only use our credit policy, thus forcing the algorithm to sell all the assets; if it is positive, we assign a random number for each asset taking into account that the loan will not be used, i.e., the loan is assigned to asset 0 (cash or risk-free asset). The wealth in each period is calculated recursively, so the wealth of one period is transferred to the next one by capitalising the position of each asset with its own factor. Notice that we avoid using the loan when the initial wealth in a period is positive. Otherwise, the GA would have to find solutions that compensate the extra cost of using the loan, which would make its task increasingly difficult due to the high interest rate of these long-term loans.

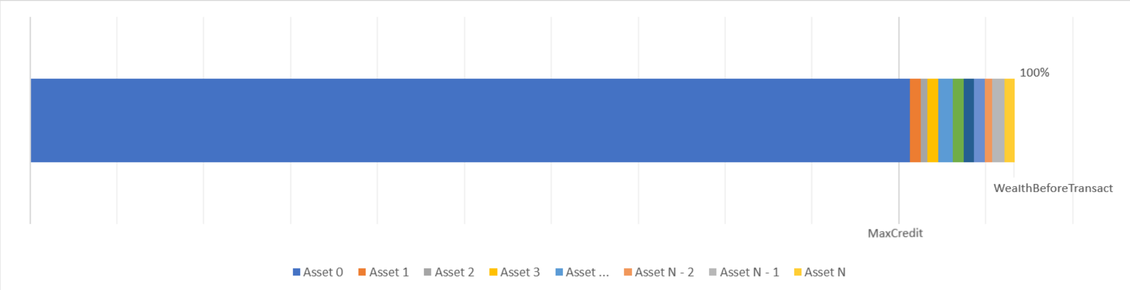


Figure 9.3: Creating a Primitive Solution of a Genetic Algorithm

The algorithm that generates the evolution of the civilisation is described in pseudo-code 5 - Start Method. In each generation, a new collection of individuals is created following the pseudo-code 6 - Person::SonOf Method. After 10 generations without improvements, the mutation rate is reduced. Thus, we account for the fact that current mutations over the new individuals are situating them farther away than their parents with respect the optimal value.

In the process, we insert a procedure called *CleanJumps*, which reviews all pairs of transactions associated with every asset that changes its sign and tests if both transactions can be regrouped in the first one, thus leaving the second one to zero. This phenomenon is common in assets that are not relevant. In that case, the GA selects a transaction of one asset in one period and, if the asset is not relevant for the final result, in the next period the GA will choose the opposite movement, to compensate the previous movement. The result is that we have two strange movements of opposite sign that we should delete, since they are difficult to understand for managers. The *CleanJumps* procedure gives us a faster algorithm with easier-to-understand solutions. For each generation, we create a deterministic number of sons, *NumSons*, which is based on two parents (survivors). This part of the code is implemented by parallelising the computation, so that we have a huge improvement in terms of speed. We loop this process until we complete a pre-established number of generations.

Algorithm 5 Start Method

```
Function Evolution():
    for int j = 0; j < NumGenerations; j++ do
        Parallel.For(0, NumSons, i => persons[NumSurvivors + i].SonOf(i, Persons)
        Sort(Persons)
        if j is integer multiple of SomeNumber then
            CleanJumps(j);

        if persons[0].utility > ant then
            ant ← persons[0].utility
            count = 0

        else
            count ← count + 1

        end
        if count > 10 then
            MutationRate ← MutationRate × 0.9
            count ← 0

        end
    end
End
```

Pseudo-code 6 - Person::SonOf Method describes the way we create a new individual from two parents. We take two random individuals (parents) and copy the first x periods from parent 1 and complement them with the last $n - x$ periods of parent 2 (where n refers to the number of total periods in the considered horizon). This hybridisation of periods will generate a new individual. The periods of this new individual are mutated according to pseudo-code 7. This mutation is absolutely necessary, since it is the way to obtain a variety of solutions, thereby diversifying the search process. Otherwise, we would only have the best solution among the elements of the initial population. In our case, we have implemented a mutation over all the assets in each period. This mutation has only three possibilities for each asset in the given period: increasing its value, decreasing its value, or keeping the current value. The mutation rate allows us to speed up or slow down the probability of applying these mutations to each asset in each period.

Finally, we create a chain of civilisations. This process is described in pseudo-code 8 - Civilisation Chain Process Method. After completing a civilisation, two procedures are run. The first one computes the classification of the assets in terms of importance. Then, we compute the sum of the transactions along the time and

Algorithm 6 Person::SonOf Method

```
Function Person::SonOf(numson, Persons):
    crosspoint ← RandomNatural(0, NumPeriods)
    u1 ← numson/NumSonsPerSurvivor
    u2 ← RandomNatural(0, NumSurvivors)
    for p in Periods do
        if p < crosspoint then
            periods[p] ← Persons[u1].periods[p]
        else
            periods[p] ← Persons[u2].periods[p]
        end
        periods[p].Mutate()
    end
End
```

Algorithm 7 Period::Mutate Method

```
Function Period::Mutate():
    for asset in Assets do
        k ← RandomNatural(0, 2) − 1
        asset.RelativeWeight ← asset.RelativeWeight ∗ (1 + k ∗ MutationRate)
    end
End
```

verify whether there are assets with a gap of 0.5% or lower with respect to the most important asset. These non-relevant assets are deleted from the final solution in order to simplify it and make it easier to understand for managers. This process is called *ComputeAssetCardinality* in pseudo-code 8. In order to get a new civilisation, we take the population of the previous civilisation and smooth the transactions with a moving average of three points. Apart from that, we perturb their *RelativeWeights* with an initial *MutationRate*. That transformation allows the algorithm to start from a different point. Although this initial point might be worse than the current one, the fact of starting from a different point prevents us from getting trapped in local minimum and increases the chances of improving the results of the previous civilisation.

9.5 Computational Experiments

The proposed heuristic has been implemented as a C# application running on a CPU with Intel(R) Core(TM) i7-8700 CPU@3.20GHz and 16 GB of RAM. As a preliminary study, and in order to test the efficiency of our simheuristic approach,

Algorithm 8 Civilisation Chain Process Method

```
Function Start():
    for i == 1 to NumCivilizations do
        if i == 1 then
            civilisation[i].CreateInitialPopulation
        else
            civilisation[i].CreatePopulationFromCivilisation[i − 1]
        end
        civilisation[i].Evolution
        civilization[i].ComputeAssetCardinality
    end
End
```

three small instances have been created and solved using both an exact method and the simheuristic algorithm. These instances contain 5 periods and 2 assets, and we use a high value for the credit loan to show how the two approaches differ. In the case of the exact method, a Montecarlo simulation has been applied on the obtained solutions in order to get an accurate estimate of their values in a real-life scenario under uncertainty. The results are shown in Table 9.2. Notice that, when simulated in a real-life scenario, the solutions generated using the exact method offer a worse performance than the ones generated by our simheuristic approach, both in terminal wealth and volatility. This justifies the need for employing our simheuristic approach in scenarios under uncertainty.

The remaining of this experimental section is divided into two parts. The first one (case A) explores the solution of a multi-period case without extra cash-flows, i.e.: only cash-flows associated with the initial investment are considered. In the second one (case B), a stochastic cash-flow is considered. In both cases, realistic market data with 20 assets is employed. This data comes from different industries, and covers different periods and risk aversion levels. Figure 9.4 shows the wealth-volatility efficient frontiers for both cases. For the respective 10, 20, and 40 period cases. These curves represent the maximum terminal wealth per final volatility. They have been computed with the same risk aversion parameters and with 100 parents, 5 sons per parent, 1500 generations, and a total of 3 civilisations. The last point assumes absolute risk aversion (the associated volatility is zero), which has been computed after setting all decision variables to zero.

The noise that these curves show is due to the fact that the GA does not guarantee optimal solutions. Nevertheless, it is possible to recognise the shape of the efficient

Instance		Term. Wealth	Term. Volatility	Sim. Wealth	Sim. Volatility	Error TW	Error TV
1	Exact	27,058 €	19,215 €	- 632 €	52,447 €	-102%	173%
	SimHeur	13,441 €	19,515 €	12,185 €	41,293 €	-9%	112%
2	Exact	54,165 €	31,738 €	29,448 €	58,484 €	-46%	84%
	SimHeur	49,851 €	31,342 €	39,934 €	40,231 €	-20%	28%
3	Exact	102,661 €	49,003 €	95,388 €	55,578 €	-7%	13%
	SimHeur	86,707 €	50,307 €	82,814 €	51,081 €	-4%	2%

Table 9.2: Results of the Exact and Simheuristic Methods with Instances with 2 Assets and 5 Periods

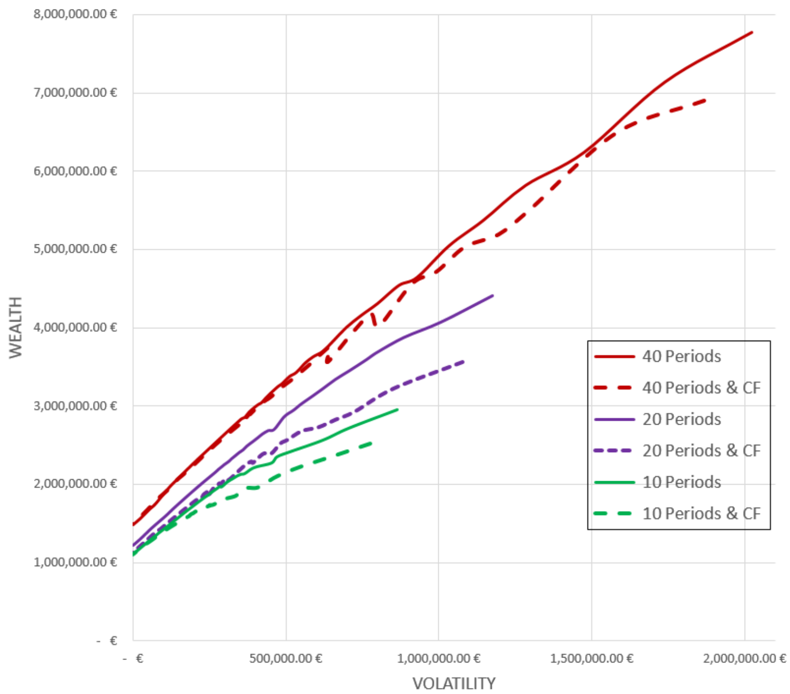


Figure 9.4: Efficient Frontiers

frontiers, which are quite similar to the ones obtained in the Markowitz-Tobin's model (Tobin, 1958). Notice that the first part of each line is a straight line, while the second one is a curve and corresponds to the speculative range. The difference between these two parts vanishes as the period size is increased, which is due to the fact that the volatility effect decreases as the number of periods is increased. We can also see that the curves corresponding to case B (with cash-flows) have a lower slope. This is due to the fact that cash-flows distribute the investment along time, thus shortening the investment period.

Table 9.3 displays the experimental results following the exact utility function. Each test has been repeated 5 times, and the result is shown in the output columns as the average value. The error is computed as the difference between the largest and the smallest of the obtained values divided by the average value. The instances are in groups of four, where differences in each group are due to the corresponding GA parameters. In each group, we combine 50 or 100 survivors with 5 or 10 sons per parent, which provides four different variants. The market data we have used is listed in Table 9.4.

By inspecting the utility error, one can notice that results are quite accurate. However, the greater the number of periods we consider, the lower the accuracy is. Notice that this accuracy value is not related to the gap with respect to the optimal solution (which is unknown), but to the stability of the solution provided by the algorithm. If we compare the instances where the product $survivors \times sons$ is constant (tests with the same number of individuals to be evaluated per generation, i.e., 50×10 and 100×5 in our tests), one can observe the following: *(i)* if the aversion risk is high ($Av = 4$), it is better to use more survivors; and *(ii)* if the aversion risk is low ($Av = 40$), we have similar results using either more survivors or more sons. In general, when more survivors are considered there are more chances to explore new paths in the search for the optimal solution (and, hence, a higher chance of obtaining better results). However, if the number of survivors is reduced while the number of sons is increased, there is a higher level of exploration per survivor. As a result, it is possible to converge to the best solution for that survivor faster.

As can be observed, only a reduced set of assets (out of the 20 assets considered)

Type	Periods	Aversion	Survivors	Sons	Time (s)	Wealth (€)	Volatility (€)	Utility (€)	% Error	Best (€)
A	10	4	50	5	13	2,358,199	491,263	1,925,640	2.2%	1,951,784
A	10	4	50	10	21	2,351,879	478,897	1,940,784	1.1%	1,949,999
A	10	4	100	5	27	2,358,095	477,748	1,948,979	1.3%	1,954,704
A	10	4	100	10	38	2,346,317	470,859	1,949,076	0.9%	1,955,041
A	10	40	50	5	9	1,388,465	88,011	1,249,681	0.4%	1,252,112
A	10	40	50	10	15	1,389,901	88,599	1,249,255	0.3%	1,251,003
A	10	40	100	5	17	1,389,874	88,413	1,249,819	0.5%	1,252,283
A	10	40	100	10	26	1,391,240	88,356	1,251,365	0.4%	1,253,478
A	20	4	50	5	77	3,532,947	754,085	2,610,918	4.1%	2,653,723
A	20	4	50	10	117	3,580,576	759,290	2,645,795	2.8%	2,663,542
A	20	4	100	5	131	3,564,083	765,356	2,614,264	2.9%	2,664,681
A	20	4	100	10	228	3,579,912	765,220	2,630,482	2.8%	2,665,033
A	20	40	50	5	56	1,614,149	109,271	1,420,576	0.2%	1,422,285
A	20	40	50	10	85	1,614,978	109,549	1,420,416	0.3%	1,421,770
A	20	40	100	5	90	1,616,858	109,754	1,421,571	0.5%	1,425,124
A	20	40	100	10	156	1,616,704	109,780	1,421,323	0.3%	1,423,525
A	40	4	50	5	352	5,331,440	1,140,497	3,604,676	4.5%	3,665,794
A	40	4	50	10	605	5,360,064	1,145,383	3,618,386	3.7%	3,675,175
A	40	4	100	5	526	5,397,406	1,142,927	3,663,502	0.6%	3,673,083
A	40	4	100	10	989	5,364,962	1,138,455	3,644,638	4.1%	3,685,045
A	40	40	50	5	170	2,084,857	148,813	1,790,883	1.4%	1,798,195
A	40	40	50	10	335	2,086,945	149,268	1,791,190	0.9%	1,795,477
A	40	40	100	5	349	2,095,890	150,346	1,795,866	0.5%	1,799,240
A	40	40	100	10	556	2,089,541	149,595	1,792,504	0.4%	1,795,594
B	10	4	50	5	13	2,212,216	481,497	1,837,064	2.5%	1,858,076
B	10	4	50	10	21	2,222,288	479,405	1,850,600	3.4%	1,881,531
B	10	4	100	5	21	2,223,724	474,013	1,860,603	2.0%	1,879,867
B	10	4	100	10	42	2,231,684	478,752	1,860,555	2.8%	1,884,339
B	10	40	50	5	9	1,463,555	82,396	1,353,751	1.6%	1,359,171
B	10	40	50	10	12	1,471,013	84,182	1,356,477	1.0%	1,361,002
B	10	40	100	5	16	1,476,345	85,115	1,359,291	0.6%	1,362,414
B	10	40	100	10	27	1,473,315	84,069	1,359,098	0.7%	1,363,337
B	20	4	50	5	75	2,742,241	601,106	2,104,725	2.7%	2,126,379
B	20	4	50	10	126	2,750,436	609,303	2,095,710	4.3%	2,131,542
B	20	4	100	5	132	2,765,805	601,979	2,126,745	1.0%	2,135,910
B	20	4	100	10	259	2,718,116	592,996	2,097,882	6.0%	2,148,412
B	20	40	50	5	52	1,434,341	92,149	1,284,596	1.2%	1,291,395
B	20	40	50	10	86	1,438,900	92,808	1,286,996	1.1%	1,292,405
B	20	40	100	5	71	1,432,231	91,557	1,284,334	1.1%	1,292,075
B	20	40	100	10	202	1,443,114	93,543	1,288,764	0.9%	1,291,972
B	40	4	50	5	339	5,000,434	1,086,280	3,431,944	7.1%	3,538,602
B	40	4	50	10	612	5,140,309	1,104,041	3,521,593	1.6%	3,546,089
B	40	4	100	5	727	4,971,247	1,079,097	3,424,646	3.9%	3,479,514
B	40	4	100	10	1,086	5,040,306	1,092,249	3,455,198	11.4%	3,570,234
B	40	40	50	5	233	2,063,311	145,415	1,782,475	1.1%	1,793,671
B	40	40	50	10	359	2,057,467	144,773	1,779,113	0.6%	1,784,228
B	40	40	100	5	320	2,046,794	142,831	1,775,824	1.2%	1,788,071
B	40	40	100	10	637	2,069,224	146,260	1,785,138	0.7%	1,790,863

Table 9.3: Summary of Results

Asset	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Return	1.0%	15.2%	15.7%	7.1%	7.2%	8.0%	6.4%	1.5%	4.4%	5.1%	3.7%
Std Dev	0.0%	14.6%	18.3%	15.9%	17.8%	14.4%	18.0%	1.1%	6.2%	16.3%	31.4%
Asset		**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Return		2.0%	1.2%	2.5%	3.5%	3.9%	0.3%	7.1%	0.2%	0.8%	1.5%
Std Dev		8.4%	5.2%	22.0%	8.4%	8.3%	7.8%	9.1%	0.8%	1.7%	6.0%

Table 9.4: Market Data

play a relevant role in each solution. We can also observe a minor contribution of other assets, and the rest of them play no role. Moreover, it is possible to recognise regular and well-known shapes, like the exponential one or different polynomials. Notice that it is common to find solutions with a predominant asset along the first periods. In a real-life case, this is not a convenient solution: although we use a plausible risk model, each asset has other unknown or unpredictable risks. Hence, it is usual that the investor imposes additional constraints, such as limiting the amount of money per asset or per class of asset.

Another interesting conclusion is related to the connection of consecutive civilisations. We have considered slight modifications that introduce a relatively small change in the population of the last civilisation. On the one hand, we analyse the weight of each asset in the whole solution. Hence, if the asset is not relevant in comparison with the most used one, it will not be considered anymore. We also review transactions located between two adjacent transactions that are either higher or lower simultaneously (i.e., we recognise "V" or "Λ" patterns). Under some general conditions, we substitute the transaction at the centre for the average of the three consecutive ones. Finally, we add some random variation over all the transactions of each asset. All of these changes might generate a slight degradation of the current solution, but they also allow us to obtain new individuals that are used to explore different investment paths.

Figure 9.5 displays the evolution of all civilisations associated with one instance. Notice that the first civilisation shows a huge improvement. This is because it starts with an initial random population and the survivors are getting better after each generation until a stationary stage is reached. At this point, a new civilisation begins, improving over the previous one, although the marginal improvement decreases. This process continues until the algorithm's termination criterion is reached. Moreover, Figure 9.6 displays the evolution of the dispersion, where dispersion D is the sum of the absolute deviation from a transaction to the next one, i.e.: $D = \sum \mid T_i - T_{i+1} \mid$. Notice that there is a noticeable improvement in terms of solution stability, which is mainly produced by the innovation of the linked civilisations.

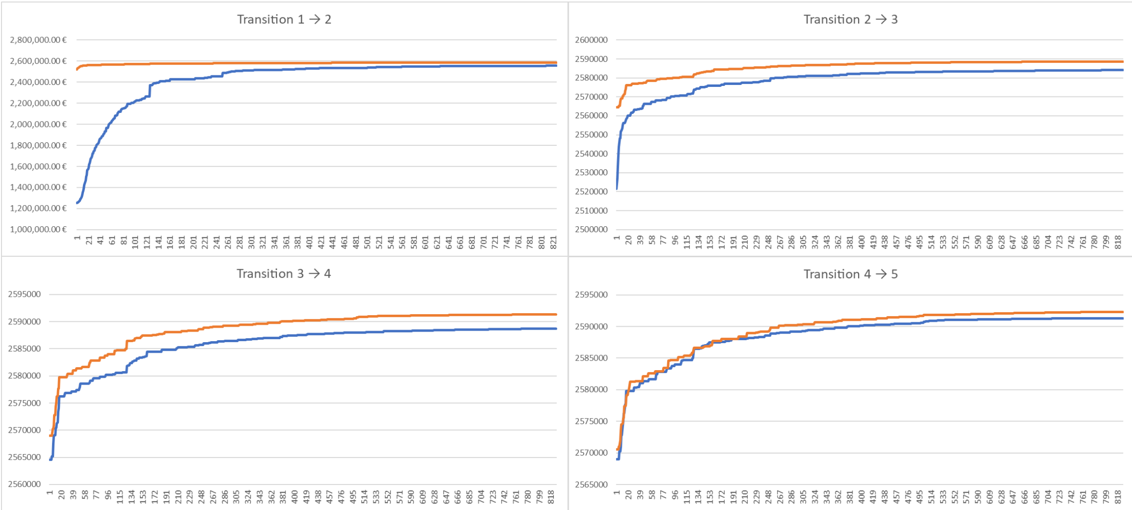


Figure 9.5: Utility Evolution - Instance with 20 periods, A - $P100 \times S10$ - Aversion = 4

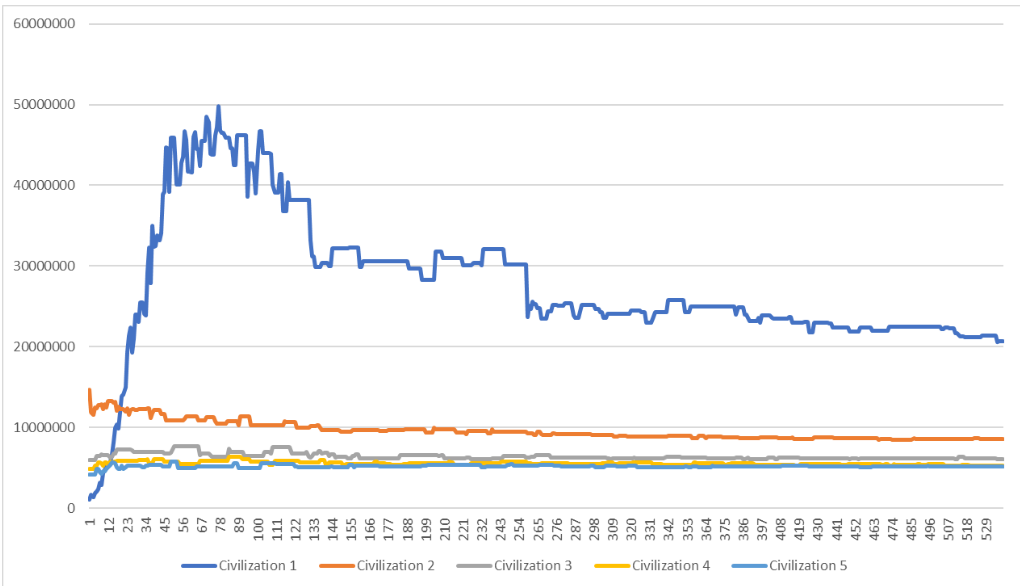


Figure 9.6: Dispersion Evolution - Instance with 20 periods, A - $P100 \times S10$ - Aversion = 4

These instances have also been computed using the full simheuristic approach, i.e., by replacing the analytical expression of the terminal wealth with a Montecarlo simulation. Indeed, the use of simulation also allows us to include rich and real-life characteristics that are difficult (or even impossible) to consider by just using analytical expressions. Therefore, the use of simulation leads to more realistic solutions, although the computational time increases. We present the results provided by our simheuristic approach in terms of terminal wealth and volatility.

As a final step in our approach, we also make use of Montecarlo simulation to assess the solutions provided by the GA when they are implemented in a realistic scenario under uncertainty. This is done both for the GA-alone solution as well as for the solution provided by our GA-simheuristic. The best results per instance have been chosen (12 in total), and in order to test the robustness of the provided solutions we have simulated the random walk of each asset using a Log-Normal probability distribution. The number of simulation runs per test is 1000. During the simulation, we define the *deviation* concept as any situation in which the predicted plan cannot be accomplished. That situation occurs whenever the plan includes a future sale of an asset and the simulated balance of that asset is less than the planned one. In those cases, we count a deviation event. The magnitude of this deviation is also considered in our computations. The simulation results are provided in table 9.5 and table 9.5. The first five columns reproduce the best results for each instance –outcomes of the GA-alone approach in the first table, and those of the GA-simheuristic in the second one. The next three columns refer to the terminal values obtained by the Montecarlo simulation final assessment. The last columns give us the average amount of money that could not be transacted (deviation amount) and the ratio between the shortfall and the utility given by the GA.

The following conclusions can be derived: both the GA-alone and the GA-simheuristic values are quite similar, which confirms the good quality of the predicted values provided by the GA. Moreover, we see that the deviation amounts are similar to the predicted volatility values. According to our simulation results, the average deviation value is similar to the volatility, with a variation of $\pm 25\%$. This gives us a new interpretation of the risk: the risk value is not only the possible variation of

Type	Periods	Wealth	Volatility	Utility	SimWealth	SimVol	SimUtility	Deviation	Dev/Utlty
A-4	10	2,354,261 €	472,039 €	1,955,041 €	2,366,183 €	509,492 €	1,901,098 €	317,905 €	17%
A-40	10	1,394,636 €	88,762 €	1,253,478 €	1,398,951 €	110,193 €	1,181,398 €	89,529 €	8%
A-4	20	3,607,262 €	762,367 €	2,665,033 €	3,634,443 €	844,866 €	2,477,255 €	561,442 €	23%
A-40	20	1,622,672 €	110,388 €	1,425,124 €	1,626,673 €	140,393 €	1,307,139 €	115,076 €	9%
A-4	40	5,423,180 €	1,144,341 €	3,685,058 €	5,438,825 €	1,315,474 €	3,141,967 €	878,654 €	28%
A-40	40	2,101,708 €	150,958 €	1,799,240 €	2,106,720 €	193,932 €	1,607,526 €	157,373 €	10%
B-4	10	2,241,337 €	470,057 €	1,884,339 €	2,119,337 €	487,271 €	1,735,713 €	263,139 €	15%
B-40	10	1,478,082 €	84,272 €	1,363,337 €	1,362,446 €	96,974 €	1,210,505 €	71,904 €	6%
B-4	20	2,760,055 €	588,966 €	2,148,412 €	2,713,467 €	693,312 €	1,865,898 €	411,823 €	22%
B-40	20	1,446,561 €	93,502 €	1,292,405 €	1,442,396 €	115,830 €	1,205,827 €	95,052 €	8%
B-4	40	5,209,902 €	1,111,176 €	3,570,234 €	4,938,387 €	1,376,608 €	2,421,806 €	819,998 €	34%
B-40	40	2,083,029 €	147,612 €	1,793,671 €	2,082,589 €	196,695 €	1,568,811 €	153,518 €	10%

Table 9.5: Results of the Montecarlo Simulation: GA-alone Utility Function

Type	Periods	Wealth	Volatility	Utility	SimWealth	SimVol	SimUtility	Deviation	Dev/Utility
A-4	10	2,050,870 €	338,391 €	1,845,709 €	2,075,337 €	372,073 €	1,827,302 €	249,307 €	14%
A-40	10	1,288,876 €	58,681 €	1,227,180 €	1,293,987 €	70,919 €	1,203,876 €	57,284 €	5%
A-4	20	2,610,240 €	469,905 €	2,252,269 €	2,727,568 €	529,791 €	2,272,540 €	381,048 €	17%
A-40	20	1,390,444 €	58,301 €	1,335,339 €	1,401,191 €	66,631 €	1,329,217 €	54,712 €	4%
A-4	40	3,596,210 €	721,195 €	2,905,851 €	3,725,761 €	934,240 €	2,567,288 €	620,066 €	24%
A-40	40	1,829,065 €	109,349 €	1,670,356 €	1,854,895 €	124,263 €	1,649,943 €	88,034 €	5%
B-4	10	2,035,842 €	360,382 €	1,826,001 €	1,926,590 €	380,201 €	1,693,034 €	222,362 €	13%
B-40	10	1,392,940 €	57,507 €	1,339,507 €	1,278,916 €	66,361 €	1,207,763 €	49,580 €	4%
B-4	20	2,117,869 €	407,265 €	1,825,406 €	2,157,399 €	453,658 €	1,794,510 €	309,732 €	17%
B-40	20	1,283,880 €	53,980 €	1,232,502 €	1,295,817 €	62,874 €	1,226,112 €	52,811 €	4%
B-4	40	3,760,930 €	730,028 €	3,053,197 €	3,882,929 €	942,914 €	2,702,241 €	602,770 €	22%
B-40	40	1,622,043 €	61,839 €	1,571,260 €	1,649,946 €	73,129 €	1,578,927 €	56,961 €	4%

Table 9.6: Results of the Montecarlo Simulation: GA-simheuristic Utility Function

the terminal wealth, but also the deviation in the transaction amounts compared to those in the investment plan generated by the GA. However, the most relevant result comes when we compare the ratio Deviation/Utility: in all of the cases, the GA-simheuristic utility function presents a lower ratio than the equivalent case with the GA-alone utility function. Hence, as expected, integrating the simulation component at different stages of the GA optimisation module leads to a more robust solution. The reason, of course, is because the solution evolution is driven by a more realistic terminal wealth. All in all, it is possible to conclude that: *(i)* the transaction plan given by the GA-alone is fairly reasonable; and *(ii)* this plan can be further enhanced by inserting simulation at each stage of the GA, as proposed in our GA-simheuristic approach.

9.6 Conclusions and Further Work

This chapter has analysed the stochastic and multi-period asset-liability problem. This is a complex optimisation problem that many financial institutions have to face in order to guarantee that their liabilities are covered by a number of dedicated assets, which vary from period to period. The goal is to define the portfolio of assets to be kept at each period so that the terminal wealth is maximised while satisfying all liabilities in due time. In order to solve the problem, we have proposed a GA-simheuristic approach that integrates Montecarlo simulation at different stages of a genetic algorithm. As discussed in the experimental section, our GA-simheuristic constitutes an effective methodology for solving complex financial problems that require the matching of assets and exogenous cash-flows (liabilities). The proposed GA-simheuristic provides accurate results in terms of utility value, and it can be easily extended to consider other risk definitions or models with richer characteristics.

The experimental section shows that the GA-simheuristic is able to obtain good-quality solutions in short computing times. However, the methodology required some refinements in order to generate 'smooth' solutions that can be accepted by most managers. Our results also show that injecting a simulation component along the evolution of the population increases the robustness of the results, thus reducing the number and volume of defaults along the planned horizon.

Several open research lines are listed next: *(i)* to explore the use of a multivariate regression model as a proxy (surrogate) model for some of the intermediate simulations, which will speed up the computations; *(ii)* to test constraints according to the standards of the market –e.g., dividing the assets into classes and limiting the investment per asset class; and *(iii)* to introduce fixed incomes with default conditions.

Chapter 10

Simheuristics in Finance: Managerial Insights

10.1 Introduction

In the last chapter I presented a combination of two powerful techniques. On the one hand, I developed a genetic algorithm to solve a complex asset and liability optimisation problem. The big number of constraints and also the common size of a problem like this suggests us the implementation of a solution based on heuristics since an exact solution seems hard to achieve. This approach gives in a reasonable time a plausible solution although it has a clear weakness which comes from the mathematical formulation. Our model is an extension of the traditional Markowitz – Tobin problem and it is well formulated in mathematical terms as I proved in Chapter 7. Nevertheless, the risk formula we developed has a crucial hypothesis, the asset returns behave as normal probability distributions. When we impose constraints like $x_i \geq 0$ we are violating the mentioned hypothesis. Indeed, the one period two-fund theorem proves that the capital market line is just that, a straight line, and that is possible because you can purchase or sell any amount of assets, i.e., any amount positive or negative. In other words, you can have negative assets in you balance sheet. In some markets it is possible, like in the derivatives market. In those markets it is not relevant to physically own stocks because when the maturity date is achieved, you clear the transaction by differences among the prices in the contract and the quoted prices in the market. In the field of portfolio management things work differently, you purchase real stocks and you keep them in your balance as real assets. This affects directly to our model because it is not possible to have negative positions, so the random walk of one asset is necessarily truncated. And so, it is not a normal distribution anymore. As we did before, simulation is then necessary, and we had to develop a simheuristic model to solve our multiperiod optimisation problem.

However, the solution is not free of difficulties. Inserting Montecarlo as a technique to evaluate the terminal wealth increases extraordinarily the needs of computational resources and the execution time, and not only that, it adds a great amount of noise in our solutions in spite of having reasonable final values. In figure 10.1 we present the time functions of the main assets that are part of the solution of the instance A with aversion risk = 4 and 40 periods without simulation, and in figure 10.2 we show the equivalent sketch inserting simulation. As we see, the first figure tends to be regular and the second one lacks absolutely of regularity or continuity. Even if the results of simheuristics are better (in terminal wealth and in stability terms), it is unlikely that any chief financial officer would admit that solution as investing strategy plan in the mid or long term. Along this chapter we will explore a different way to include simulation into our genetic algorithm just to solve that inconvenience, we will compare this new approach with the previous one, and at the same time, we will check our results for a set of four simple instances that imitate real management problems in the financial industry.

10.2 Simheuristic Model

As we have just mentioned, we will continue solving the multiperiod optimization problem, fully described in the previous chapter of this thesis. We have seen that this deterministic model can be well solved using a Genetic Algorithm (GA). Nevertheless, this model is limited in many aspects. First, the solution that our GA gives us

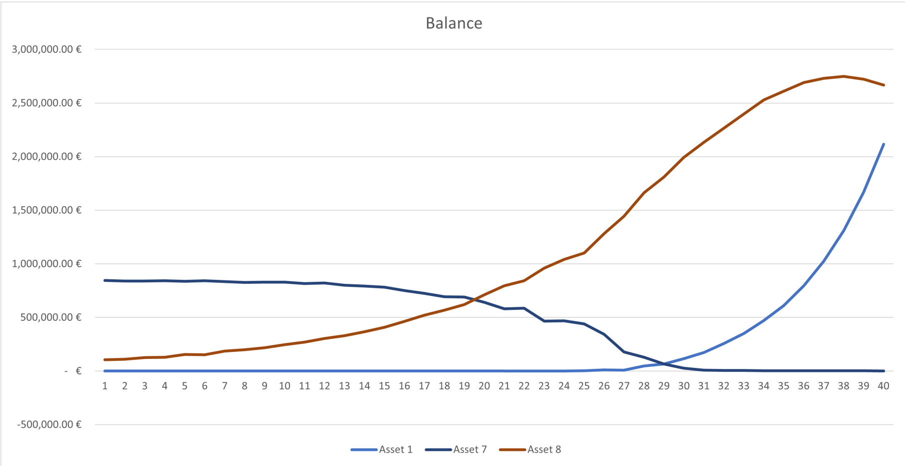


Figure 10.1: Shape of Main Assets as a part of a GA Solution

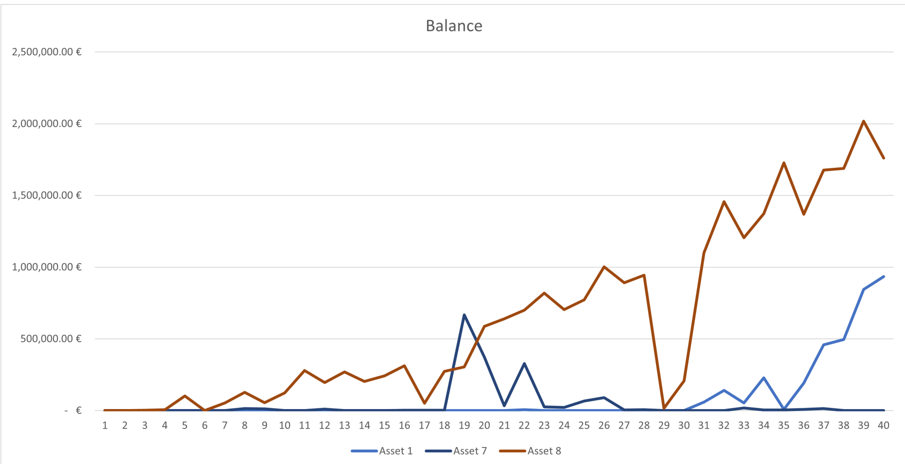


Figure 10.2: Shape of Main Assets as a part of a GA & MC Solution

is based on average values. But it is only an approximation because the random walk of a price can decrease so that the equation $x_{\alpha,t} \geq 0, \ \forall \alpha \in A^*, \ \forall t \in \{1 \ldots T-1\}$ is not met. That means that our problem is a set of deterministic values, but if we run a further simulation, the price of an asset can decrease enough to violate that constraint. Recall that we don't admit negative balances in our assets. Moreover, the liability cash-flow is considered deterministic but a realistic model should consider it as stochastic. In fact, the possible cash-flows can present complex behaviours that should be considered in a hypothetical realistic model. Finally, the λ parameter is useful to solve the equations but it is a theoretical construction. An investor wants to determine the maximum risk capital (or another reference parameter), he doesn't know anything about the Arrow-Pratt aversion index, so a method to pair risk capital (volatility) and λ is needed.

Therefore, we will consider an extension of the deterministic problem adding random behaviour in all our prices using Montecarlo simulation, and we will provide an efficient way to match the aversion risk with the required volatility using Machine Learning.

The basis of our model is to maximise the terminal wealth selecting a specific risk capital that is suitable with our risk profile. At the same time, we have to meet the expected cash-flow. Using the Lagrangian formulation, we find the concept of Arrow-Pratt aversion index that gives us a parameter that represents our risk profile or in other words, the capital risk that we are willing to tolerate. Maximising the formulation with this parameter, we get the values of the decision variables that represent the sales and purchases of our assets along the transaction. Our genetic algorithm consists in generating a specific number of random solutions, called individuals, and selecting a fraction of them with the criteria of having the best utility values. After that, as it is common in a GA, we cross pairs of survivors getting a new population, and simultaneously we mutate the decision variables that are in each individual. We iterate this procedure until we reach a stationary state. The crossing method consists in selecting a random period and interchange the decision variables before and after that selected period for the two selected individuals. This GA converges in a relatively short time with a reasonable good solution. To improve the solution, we establish a chain of civilisations. It means that the converged population we get is shocked randomly generating a new one population but with an interesting feature: this population is near the last population but with new seeds (new individuals) and it gives us the opportunity of repeating the same procedure we have just commented but with better results. This interlinking of populations ends when no improvement is achieved or a convergence criterion is reached.

The best solution of the genetic algorithm is only a solution in a complete deterministic scheme. Unfortunately, this approach is not so realistic. A real market has a complete random behaviour. It is true that our model considers a random walk of equities, and more specifically, a Log-Normal random walk for prices as a consequence of considering normal distribution for the returns of the assets. Nevertheless, we know that many other hypotheses can be feasible in equities, like volatility of the volatility, inflation, etc., which in turn it admits a great variety of models. Moreover, our problem is dual: not only we have the intrinsic random behaviour of assets, but also we have our cash-flow. In general, the cash-flow represents, on the one hand, the liabilities of our firm, and on the other, the revenues we will get along the time and both of them can be also random variables (in fact they are always stochastic). In short, our model is insufficient to respond to the question of which financial plan I have to expect along the horizon to fulfil the obligations, maximising my investment, and assuming a certain level of risk as it doesn't consider those other risks. If we consider the equivalence of the capital risk we are willing to assume with the Arrow-Pratt aversion risk, we sub-estimate this risk. Indeed, we also have to consider in our model the extra volatility that comes from the random behaviour of the cash-flows and the random expectation of the assets that is not considered in the classical Markowitz model.

Implicit Aversion Risk Index

The easiest way to calculate the utility function taking into account all our considerations is using Montecarlo techniques. From this point, we have two alternatives. The first one is to calculate our deterministic problem with some initial Arrow-Pratt aversion risk index, and after that, to re-estimate that solution using Montecarlo. The simulation will give us a bigger amount of capital risk, so we will be forced to adjust the aversion risk index and we will have to repeat the loop again until the system converges. This derives in a final aversion risk index that we can call it the implicit aversion risk index. It means that this new index can be interpreted as the one we have to use in the deterministic general model to get our capital in a stochastic context. This is not a new idea. In fact, we can interpret this implicit aversion risk index as the equivalent Lagrangian multiplier of the implicit volatility. This implicit volatility would be similar to the one we have to use in the derivatives market (Rogers et al., 1994). In it, we have to recalculate the volatility that we have to use in our Black-Scholes model to reproduce realistic prices. Here, we have to calculate (via simulation) the new volatility to use it in our GA and get the final solution. This schema is represented in figure 10.3.

The second alternative consists in substituting the exact risk calculus for Montecarlo, inside the genetic algorithm. This approach has been fully developed in chapter 8. It has the advantage that if we choose an accurate Montecarlo simulation (i.e. we use a lot of iterations), we don't need extra concepts and we reuse the GA including our non-standard financial model. But on the other hand, we will have a very heavy system and we will depend on our available computational resources. Indeed, Montecarlo can be fast to have a reasonable result, but it can be very slow if we need very accurate results, and the genetic algorithm is fast if equations are, computationally speaking, fast. In order to get a valuation of the risk formula, the simulation would need million iterations, so it is slow to apply it in our GA frame. Nevertheless, our aim is to apply Montecarlo because as we proved, simheuristics gives us a better solution compared to the pure GA. In our approach we will develop the first alternative, as the second one is in terms of computational times almost unfeasible because in order to the GA we need a precise calculus of our utility function and MC will only give us an approximation so that the natural selection of our population will be erroneous.

Our schema starts with an intermediate λ. Each time we have selected a λ we solve the deterministic problem with the GA. It gives a winner (individual with best

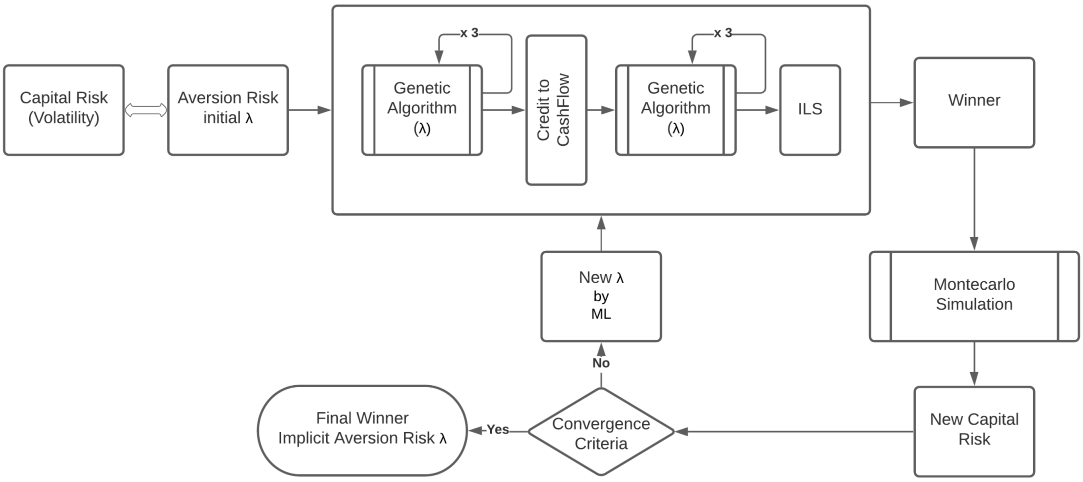


Figure 10.3: Montecarlo and Genetic Algorithm

utility value) and we calculate the realistic volatility of that solution using MC. So, it gives us a pair (λ, σ) that it is used to feed a linear regression. In fact, let's consider our objective function:

$$U(W_T) = W_T - \frac{1}{2}\lambda\sigma^2_{W_T}. \tag{10.1}$$

Generally, when we are near the solution, a linear approximation can be made, so that we can consider a linear relation between the terminal wealth and the terminal volatility:

$$W_T = a + b\sigma \tag{10.2}$$

Substituting the former expression to our utility function, we have that:

$$U(W_T) = a + b\sigma - \frac{1}{2}\lambda\sigma^2_{W_T} \tag{10.3}$$

and setting the derivative equal to zero we have:

$$\frac{dU}{dW_T} = b - \lambda\sigma = 0 \tag{10.4}$$

That means that $\lambda \times \sigma = constant$. So, taking into account that λ is an exogenous variable and the genetic algorithm introduces necessarily a random error, we can write the former expression as follows:

$$\sigma = \frac{\beta_1}{\lambda} + \epsilon \tag{10.5}$$

This means that σ is the endogenous variable of a linear regression where the exogenous variable is the inverse of λ and the independent constant of the regression β_0 is set to zero. Based on the previous simple expression, our algorithm that gets the implicit aversion risk index is described as follows:

Algorithm 9 Calculation of Implicit Aversion Risk

Function `CalcImplicitAversionRisk()`
- $\lambda \leftarrow \lambda_0$
- **repeat**
  - $\sigma = GA(\lambda)$
  - $Regression(\beta_1) \leftarrow (\lambda, \sigma)$
  - $\lambda \leftarrow \beta_1/\sigma$
- **until** $\sigma \approx \sigma_0$ *and* $\Delta\lambda \rightarrow 0$;

End

Ascent Gradient

Concerning our Genetic Algorithm, we divide it into three parts. The first part is the execution of a three-civilisation solution (we will see that we won't need more than three). After that, we select the winner of this stage and we check the use of the credit policy. In other words, we inspect the cash-flow generated by borrowing the money and afterwards, the paid off. Having this, we substitute the credit policy by an extra cash-flow, and we execute again the GA with a three-civilisation configuration. The difference between the first execution and the second one is that we don't have to borrow money in the second one, and therefore, we don't have different free risk interest rates.

This is a critical point of our improvement. Having different interest rates creates a lot of possible branches and that situation contributes to one of the major difficulties of having feasible individuals. We have to keep in mind that changing just a value of one decision variable can imply a change in the interest rate in some period. In other words, let's suppose that a solution (individual) is represented as a vector $\vec{x}$, where each coordinate x_i^α represents a transaction (decision variable) in time i of asset α. The Terminal Wealth can be expressed as $W_T = W_T(\vec{r}, \vec{x})$, where $\vec{r}$ is the interest free rate vector, where each coordinate refers to the interest rate of each period. The vector $\vec{r}$ is a function of the specific solution because each individual determines if we need to use the credit loan or not. So, we have $\vec{r} = \vec{r}(\vec{x})$. We have to take into account that this rate function has accountable jump discontinuities so we cannot guarantee the existence of the derivatives. So, we can make this statement:

$$\exists \vec{x}, \alpha, i : \frac{\partial W_T}{\partial x_i^\alpha} \notin \mathbb{R} \tag{10.6}$$

This is the reason of why we cannot improve our GA with the standard ascent gradient technique, at least directly. To avoid this, we transform the credit policy into a new deterministic cash-flow. If the solution gives us the need of using the loan, we consider an income, and after, we will consider a liability that includes the extra interest rate we would have to pay. So, with this trick, we have a Terminal Wealth that is free of jump discontinuities since $\vec{r} = r(1, 1, 1, \ldots, 1, 1)$, so ascent gradient can be applied.

Therefore, using only three civilisations as maximum, you get a reasonable good solution. With that solution you get an idea of the need of the credit loan. So, you can transform the credit loan that your solution requires into a deterministic asset (income coming from the loan) and a liability (payment of the loan plus interest), and process a common ascent gradient approach that is referred as ILS in our chart (fig. 10.3). In the computational experiments, we will show in what manner our GA solutions are highly improved.

10.3 Description of the Instances

We will test our simheuristic model with four idealised but usual cases we can find in the market, a well capitalised firm or solvent company, an indebted company, an annuity and a simple pension plan. Each one represents typical cases we can find in an ALM context, not only in the financial market but also in the scope of personal finance. These four models don't pretend to be accurate models of real life, but they want to illustrate how our approach is useful as a good tool to guide a long-term financial strategy.

Case A - A Solvent Company

To model this situation in an easy way, we consider that the solvent company gets regular income along the time. The company has also regular expenses but those are always after the income. In other words, this company never needs to borrow money. In Figure 10.4 we can see the cash-flow of the model. This case can represent a company with reverse production life-cycle, like an insurance firm. In this case, we have to worry about investing but reserving part of the capital to pay our obligations in the short time.

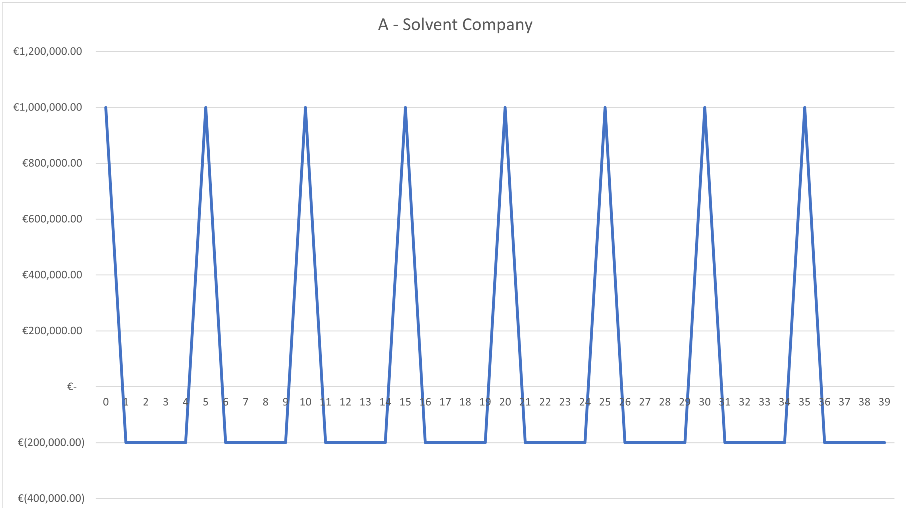


Figure 10.4: Well Capitalised Company Cash-Flow

Case B - An Indebted Company

The indebted company is similar to the previous case, but with the cash-flow in a reverse order. So, this company needs to borrow money all the time because the obligations appear before the income. We can see the cash-flow of this instance in Figure 10.5. As we see, first we have to satisfy our liabilities lending money, so it is quite difficult to manage this situation, and it is very responsive to the risk profile of the company.

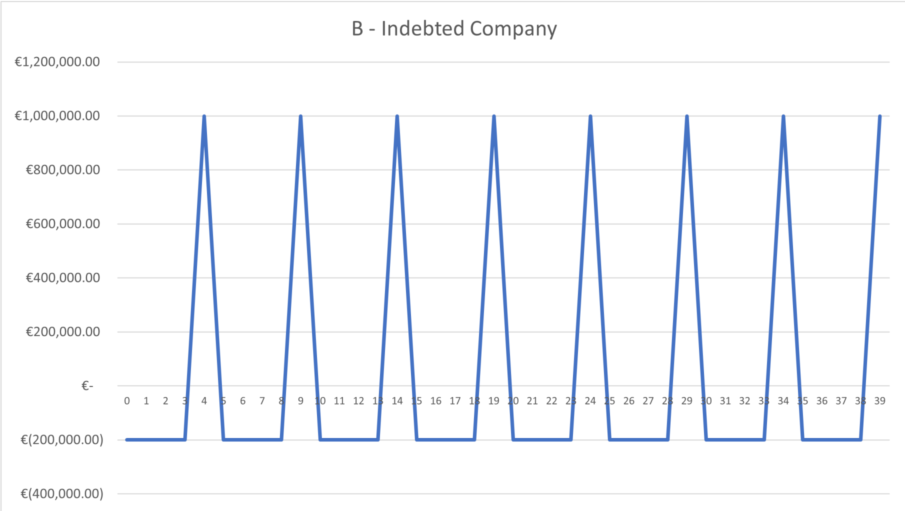


Figure 10.5: Indebted Company Cash-Flow

Case C - An Annuity

This instance is a simplified model of an annuity. First, we have a high income in our balance sheet, and we have to pay an annuity each period until the annuitant passes by. This model is in fact an average of that situation. If we only consider an individual, he receives the income until he dies. But we have considered a decrease of a 5% in each period emulating the death of part of the pensioners (figure 10.6). In other words, this model corresponds to an annuity in statistical or actuarial sense, without considering rigorous mortality details.

Case D - A Pension Plan

This last instance is a fair model of the cash-flows in a pension plan (figure 10.7). During a period of time, we receive the contributions from the stakeholders. That amount of money is considered decreasing because at the same time we get the income we also pay to new pensioners. This schema is repeated along the span of time of our operation, so the cash-flows become negative, which means that we have more pensioners than contributors.

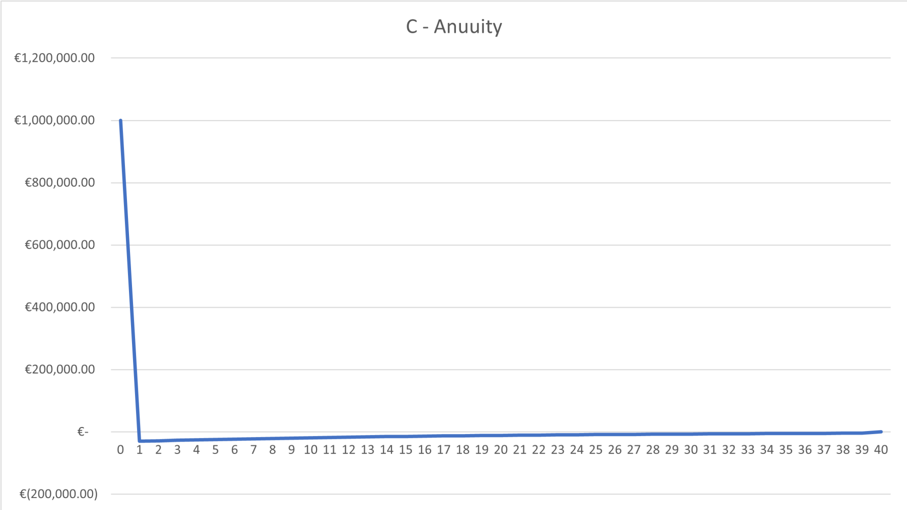


Figure 10.6: Annuity Cash-Flow

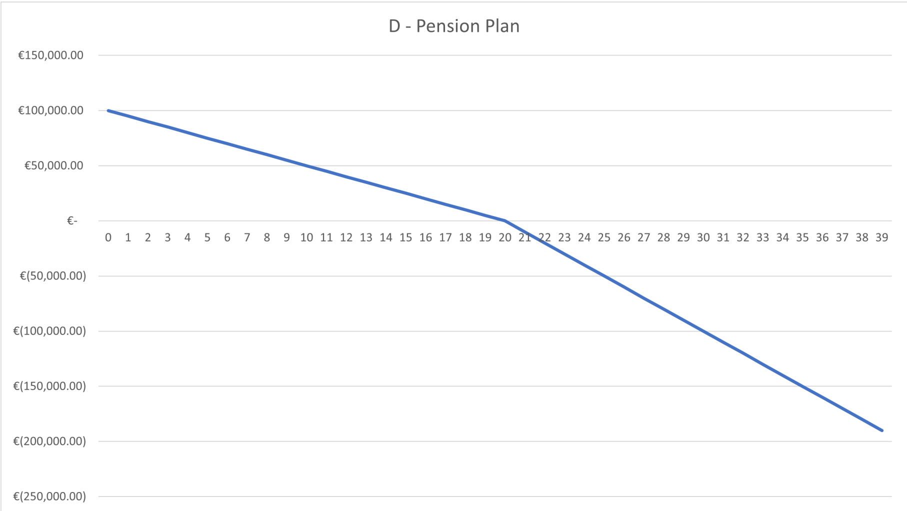


Figure 10.7: Pension Plan Cash-Flow

10.4 Computational Experiments

The proposed heuristic has been implemented as a C# application running on a CPU with Intel(R) Core(TM) i7-8700 CPU@3.20GHz and 16 GB of RAM.

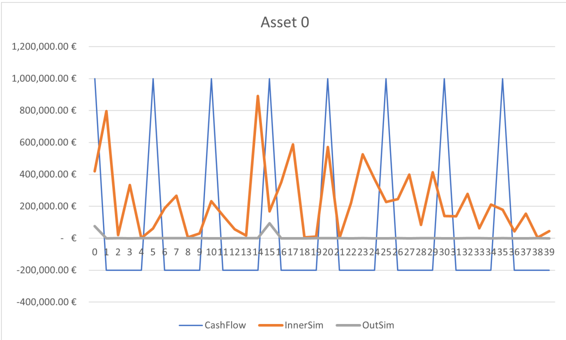
Asset	0	1	2	3	4	5
Return	1.00%	1.5%	3.5%	5.0%	8.0%	10.0%
Standard Deviation	0.00%	1.0%	3.5%	7.5%	14.0%	20.0%

Table 10.1: Portfolio of Assets Used for Application Solutions

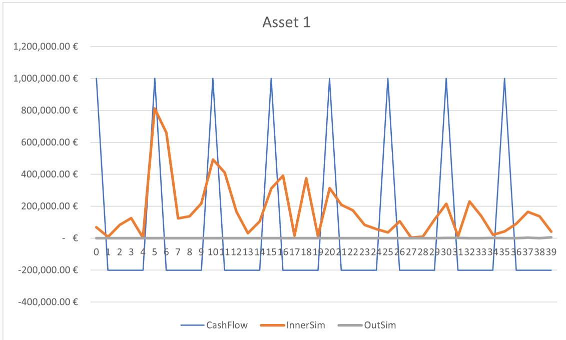
We have considered five assets plus the free risk asset and 40 periods (table 10.1). We didn't consider correlations because methodologically it is not strictly necessary.

First, we have run the instance A (solvent company) with deterministic liability with the two possible simheuristic models, according to the parameters we further define for all the instances. The inner simheuristic approach has the implementation as we have described in the previous chapter, i.e., having a pure genetic algorithm with the objective function evaluated by simulation. The outer simheuristic model (implicit aversion risk approach) is a pure genetic algorithm with the objective function evaluated mathematically (exact method) but corrected dynamically according to the evaluation given by simulation over the final solution.

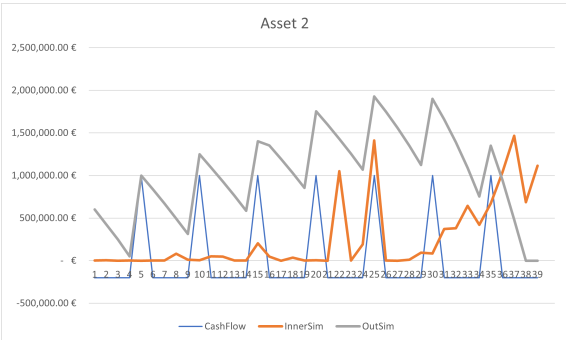
We show the balance sheet of the assets (fig 10.8). In each chart we draw the cash-flow as a reference and the different solutions of the inner and outer simulation. As we can see, the first three assets have different shapes, while the three other assets are quite similar. The higher weight has that asset in the solution, the greater the similarity . We observe clearly that the inner simulation generates a lot of noise, and the outer simulation is well defined. The reason for this great difference is that in the first case we have to simulate each individual in an evolutionary process, which implies that when we have to organise the population from the best to the worst, we will commit mistakes since the evaluation is not exact. On the other hand, we have got an exact solution (recall that eventually we use ascent gradient) and we run the simulation only for the best individual. Running the simulation for only one individual can be done with extreme accuracy without a relevant computational sacrifice, but if we do it into the evolutionary process, we have to simulate hundreds of individuals, in hundreds of generations, so we have to reduce the accuracy of Montecarlo just not to die in the attempt (in this experiment we have used 100,000 iterations per evaluation). In fact, the outer simulation approach gets the solution in 8 minutes, while the inner simulation process needs 8 hours! The other aspect, and not less important, is the shape. An irregular shape means in some manner a chaotic investment strategy, so no CFO will follow that solution.



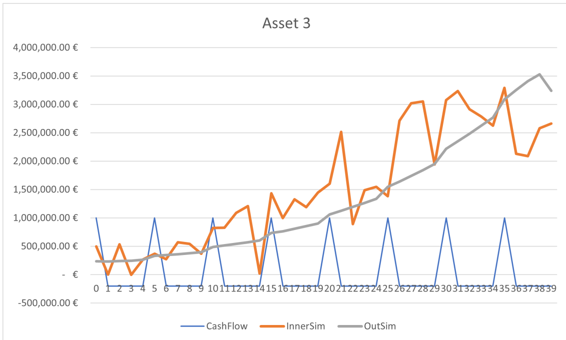
(a) Asset 0



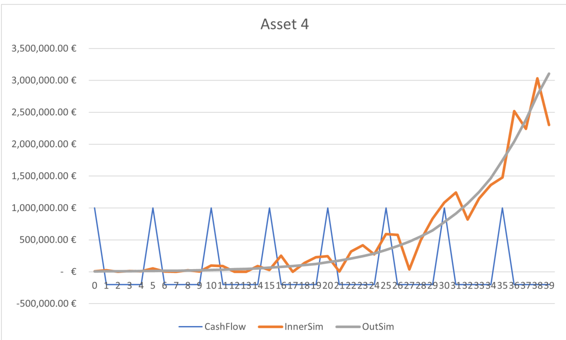
(b) Asset 1



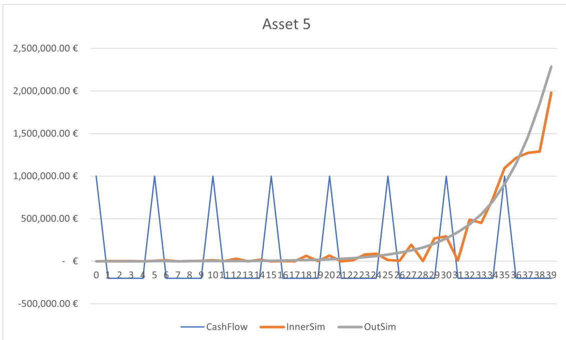
(c) Asset 2



(d) Asset 3



(e) Asset 4



(f) Asset 5

Figure 10.8: Comparison between Simheuristic Models

Following the implicit aversion risk approach, we have considered two scenarios for each instance, one with no volatility in our cash-flow and the other with volatility. In this case, we have considered that our cash-flows follow a normal distribution with a standard deviation of 10% with regard to the cash-flow value. The number of iterations used for Montecarlo simulation is 1,000,000, and we have run it in parallelisation.

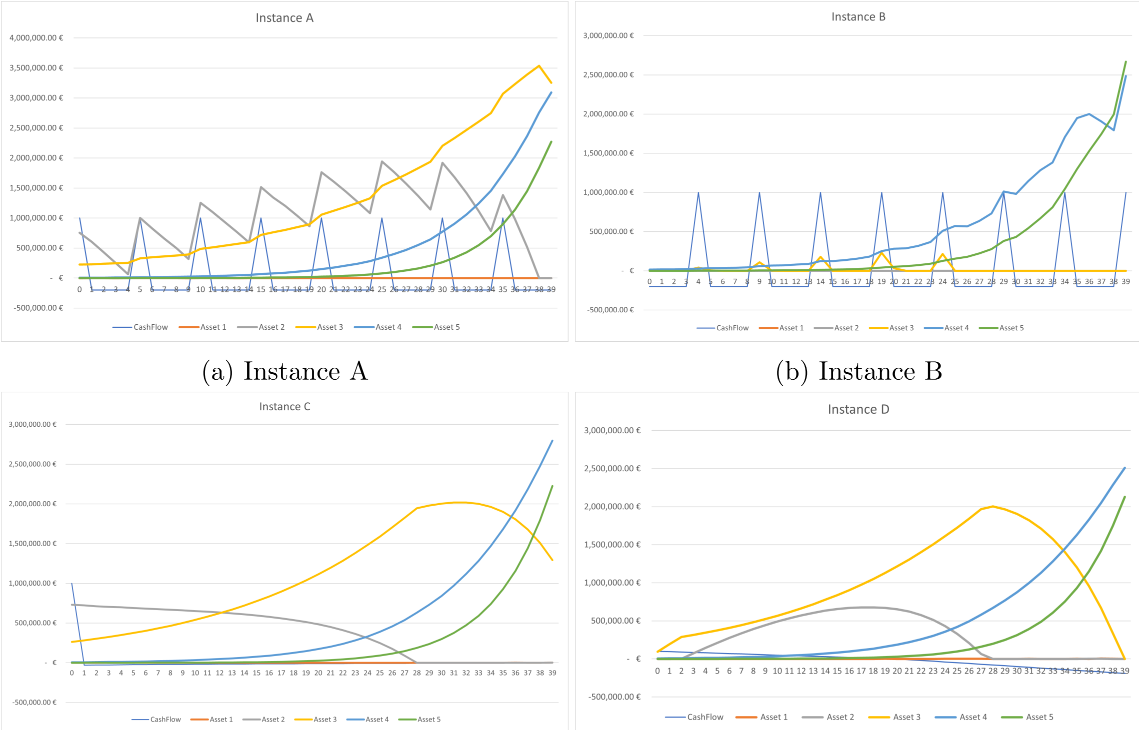
Our objective will be to solve the optimisation problem with a terminal volatility of 2,500,000 for all the instances.

The results for terminal values with and without simulation are in table 10.2, and we show in figure 10.9 the amount of money per each asset, corresponding to the variable $x_{\alpha,t}$ for the case of cash-flows with volatility. The columns TerminalWealth and TerminalVolatility correspond to the values of the implicit aversion risk index. The next two columns correspond to the values given by the simulation. The next columns are the variations between the deterministic solution and its simulation. And the last column shows the computation time and the deviation from the volatility goal. We also present in figure 10.10 the terminal wealth probability distribution for all the cases in figure 10.10.

Genetic Algorithm 3+3							
Instance	Lambda	Terminal Wealth	Terminal Volatility	SimTWealth	SimTVolatility	Exec Time (s)	**Result**
A	7.58	9,359,595.91	2,461,603.75	9,349,148.79 (-0.11%)	2,468,407.42 (0.28%)	473	**-1.26%**
A CF w Vol	8.30	9,047,159.26	2,288,150.80	8,596,490.19 (-4.98%)	2,457,390.88 (7.40%)	1091	**-1.70%**
B	4.08	4,663,215.92	2,500,365.07	4,651,220.44 (-0.26%)	2,494,109.51 (-0.25%)	156	**-0.24%**
B CF w Vol	4.70	4,355,452.79	2,204,710.89	4,084,951.78 (-6.21%)	2,461,496.20 (11.65%)	569	**-1.54%**
C	1.02	6,932,313.93	2,453,980.40	6,881,364.18 (-0.73%)	2,459,717.07 (0.23%)	645	**-1.61%**
C CF w Vol	1.02	6,933,648.95	2,454,880.03	6,866,180.75 (-0.97%)	2,466,952.79 (0.49%)	599	**-1.32%**
D	0.97	5,134,263.67	2,444,418.51	5,097,455.45 (-0.72%)	2,453,065.79 (0.35%)	326	**-1.88%**
D CF w Vol	0.97	5,143,031.01	2,452,109.07	5,042,428.95 (-1.96%)	2,467,517.74 (0.63%)	688	**-1.30%**

Table 10.2: Results

In all the instances with no volatility in cash-flows, we can see that the simulation is very close to the deterministic calculation. When we consider volatility in our cash-flows, things are slightly different. The first two instances present a considerable jump with deviations of at least 5% in the terminal wealth and naturally higher in the terminal volatility. Two factors have to be considered to explain such deviations. On the one hand, perturbations affect to both income and expenditures, that means that sometimes we have higher values in our expenditures and at the same time, less income. That means that our predetermined assets wont be enough in several moments of time, so that we will be reduced our expectation of the terminal wealth. On the other hand, the natural volatility given by our assets has to be increased with the contribution of the volatility coming from our cash-flows. As we see, instance B



(a) Instance A

(b) Instance B

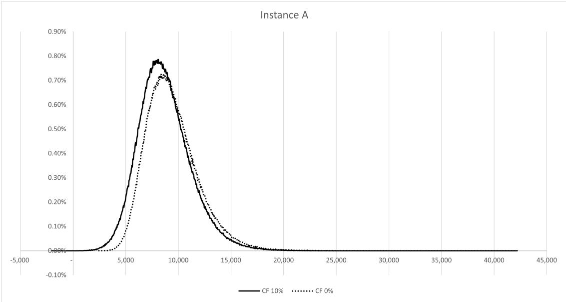
(c) Instance C

(d) Instance D

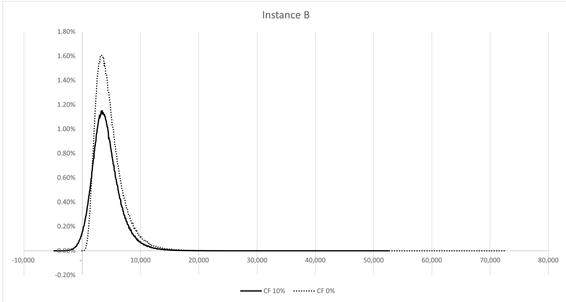
Figure 10.9: Balance Sheets of Instances

has greater effects than instance A because the margin of that operation is smaller. In fact, it needs an intensive use of the credit facility. Moreover, when we check the probability distribution, we observe that when we consider the volatility in our cash-flows, in instance A, as it is a very well capitalised company, it keeps a negligible probability of ruin. This is quite different when we have a highly indebted company, as we see that the probability of ruin starts to be visible. It is remarkable that our model is a very simple one and models that consider higher and more complex volatility models, most likely our probability of ruin won't be negligible.

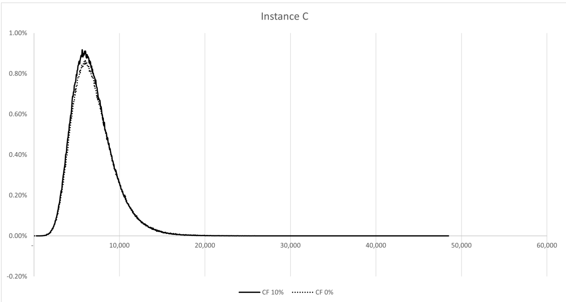
In instance C we see a very low impact. The terminal wealth expectation is the same in both cases, with volatility and without volatility in our cash-flows. It is because the first cash-flow has no volatility, it is deterministic because it occurs now, therefore there is not uncertainty. On the other hand, the values of the annuities are small compared to the initial investment, and assuming they are not correlated, we almost can neglect this impact. This fact is well corroborated in the probability distribution (fig. 10.10c), as we see a very small difference from one curve to the other one.



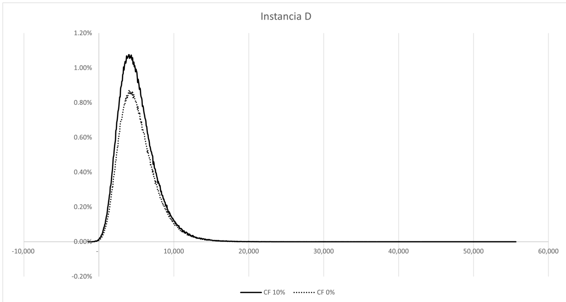
(a) Instance A



(b) Instance B



(c) Instance C



(d) Instance D

Figure 10.10: Terminal Wealth Probability Distribution

In instance D we also see the effects we have just commented, but not so intensely. Indeed, we observe that the impact on volatility is almost negligible. With respect to the terminal wealth, expectations are reduced if we have assets whose balance is near zero, as commented before. Indeed, the simulation can make the prices go below zero, and we don't admit negative assets, so we move them to cash, and recall that our cash-flow crosses the zero in one specific moment of time, so it is in that moment when we have cut down our expectations in statistics terms. But in terms of volatility, as the values of our operation are smaller than in the first two instances, the contribution is not as significant as it was in instances A and B.

Finally, in order to compare the computational time of a not correlated problem with a correlated one, we have to take into account that the computational time is directly proportional to the size of the correlation matrix. In our approach, we considered no correlation between one asset and other different one, but there is correlation between one asset in one specific period and the same asset in other period. For the case of correlation among assets, we have a symmetric correlation matrix with $np(np+1)/2$ different elements, having n the number of assets and p the number of periods. In the case of a not correlated model, we have $p(p+1)/2$

elements for each different asset in its itself correlation sub-matrix, multiplied by n different assets. So, the final ratio between correlated and not correlated market is $\frac{np+1}{p+1}$. In our case, we should expect 5 more times if we run a market with correlated assets.

10.5 Conclusions and Further Work

This chapter has analysed the multiperiod mean-variance optimisation problem applied to ALM. I have used a genetic algorithm to get the decision values (transactions) and I have also considered that the cash-flow can have stochastic behaviour, such that it adds an extra volatility component to our problem. This extra component and furthermore, the need to determine an specific capital risk drives us to introduce the concept of implicit aversion risk index. With this new concept and combining a relatively fast genetic algorithm, Montecarlo Techniques and a little piece of machine learning, we can get good solutions that can be used as the investment strategy that a company need to accomplish with all its liabilities and maximising the terminal wealth.

In particular, we have tested the model with four simple cases, and we have checked that depending on the stochastic behaviour we consider for our market and for our future cash-flows, a corrected aversion risk index is crucial to keep the solvency requirements of the company (capital risk) fulfilling all our liabilities. We have compared two models of management for a company, both with the same nominal cash-flows but in a different order. The conclusion is clear, the company that depends highly on external credit reduces considerably its terminal wealth and even compromises its future increasing its probability of ruin. And we have also checked two very common financial operations like pension funds or annuities, concluding that they are well consistent under stable uncertainty of the future liabilities, so long as the mean values accomplish.

The model can easily be adapted to much more complex market modellings, so we understand that this tool is a very effective one to design the long-term financial plan of a complex cash-flow structure.

In future work we plan to implement techniques that get us a more robust solution. Our solution is optimum for a static market, nevertheless, the financial market changes at every moment. So, it would require a dynamic revision of our solution and sacrifice part of our future wealth for the benefit of a robust solution. Moreover, although our model is completely general, part of our solutions present structural regularity, as we have seen before, so new techniques based on pattern recognition (neural networks, machine learning) could help to improve not only computational times but also the accuracy of the solutions.

Chapter 11

Final Conclusions and Further Research

11.1 Conclusions

Along this thesis we have studied a great variety of aspects that concern the management of liabilities and assets. This field of study is not new, it starts in the second quarter of the last century when Macaulay established the first studies about how to synchronise the variations in price of both assets and liabilities under slight changes in the free interest rate. From then, many branches have been treated or researched to answer the same question: what to do with assets to meet our liabilities.

In chapter 4 we reviewed the most relevant results related to ALM up to now. As I have mentioned, the duration approach by Macaulay gives some light when we worry about changes in the value of the company due to the mismatching between the cash-flows provided by the assets and the liabilities. Although it is an important issue, we have to recognise that in some manner it is a mere nominal problem: the company is the same and nothing has changed but the stock price. So, the scientific literature opens a new branch, multi-stage stochastic programming, or from an easier point of view, the cash-flow matching approach. This field of study is a clear recognition that our real objective in ALM is not to control the stock price but to guarantee the liabilities through several techniques. In parallel, a distinct discipline based on differential stochastic equations treats the same question from a complex and theoretical frame. Needless to say that both approaches have its limits. On the one hand, the cash-flow matching is a predator in computational resources if we want to solve realistic problems, and on the other hand, the complexity of the stochastic equations doesn't allow us to face problems of great size. One of the conclusions of this thesis is that while conceptually the cash-flow matching is the best choice to solve ALM, we need to tackle the problem with a new perspective, which involves using heuristics.

The lack of efficient algorithms and computational techniques derives necessarily in legal restrictions as we highlighted in chapter 2 in the part dedicated to Regulation. These mentioned restrictions constitute a loss in efficiency and profitability and this in essence becomes a risk in a company where it has to manage a huge amount of investment and obligations along the mid and long term. Our first result in this thesis is the development of a simple greedy heuristic algorithm to prove that this methodology is feasible to solve the cash-flow matching. Our results prove that, effectively, a heuristic approach is optimal as the numerical solutions are very near the optimum and the computational resources are very low-demanding. Dwelling on this subject, we recognise that, since the financial market is not deterministic, we need to test our heuristic solutions from a stochastic point of view. This is explored in chapter 5 and chapter 6, and the conclusion is that the greedy heuristic algorithm converted into a simheuristic one is even more convenient as it keeps the properties of a heuristic approach and it also covers the nuances of a random environment. Following these results, we close this first phase of the thesis developing the ALM in Insurance Firms, which is in chapter 7. The scheme is basically the same as the previous results. But we remove some limitations we had imposed in our first greedy algorithm giving us a complete and general procedure to solve the cash-flow matching in an insurance firm. These results are quite powerful because we have got a very fast algorithm that dives among the thousand of possible combinations we can choose between our fixed income assets and our obligations shared along the time. This approach can reduce the asset capital a firm needs to freeze to cover the liabilities more than 10%, which could mean hundred million euros.

Nevertheless, the ALM problem is also extensive to equity instruments beyond the fixed income. To explore a heuristic approach using equities, we needed to review the well-known portfolio theory by Markowitz. This has been necessary because we need to develop a mechanism to manage equities in the long term at the same time we guarantee our debts. In the Markowitz theory we treat the investment management in only one period, and we don't have to worry about any specific liability, so it leads to a standard portfolio optimisation problem. But when we have to guarantee that we have to pay a certain amount of money in some specific dates, we have to take this into our financial strategy. In chapter 8 we have developed this model as another contribution of this thesis. After that, we use that result to build the new portfolio optimisation problem and we design a genetic algorithm to get a good solution. In this genetic algorithm we also contribute with a methodological improvement that gives us better solutions in time dependent functions. But the main contribution in this part of the thesis is the implementation of simulation driving the evolution of the algorithm. We have proved that if we consider the stochastic nature of the problem into the objective function, the evolution prevents defaults we would have in a deterministic solution. This leads again to simheuristics as a very convenient method to solve large and complex financial problems. Finally, we close our research with chapter 10, that is an improvement of the simheuristic method of the previous chapter. This gives amazing accurate results, and we use this to illustrate how it could be applied in the financial industry through four common examples.

To sum up, along this thesis we have contributed with new and powerful techniques that could help the financial market to defeat the limitations imposed by governments such as the eligibility of the type of assets for ALM (for instance, equities are not eligible), or the prohibition of considering future premiums, among other limitations. Moreover, these techniques would improve the efficiency of the investments and reduce risks over time without the need of huge computational resources. All these contributions point in the same direction: Simheuristics.

11.2 Further Research

Following this thesis, new lines of research are opened up, all of them aimed at providing more powerful means than those currently existing in the financial market. I outline some of them.

(i) The multi-period portfolio management model with liabilities discussed in chapter 9 and chapter 10, although sufficient for a large part of the market casuistry, allows for improvements in terms of the policies that any portfolio manager imposes. In particular, we should explore the possibility of grouping assets by category (commodities, currencies, hedge funds, private equity, small/mid cap, energy, technology, etc.) and impose restrictions on the weight that a portfolio should have in each of the defined categories. This type of restrictions can be approached in different ways, both by means of evolutionary algorithms and new matheuristic algorithms.

(ii) Another of the questions we raise as future lines of work consists of being able to define simheuristics that control portfolios where assets of a different stochastic nature are mixed, such as portfolios where fixed income and equities coexist simultaneously.

(iii) Without having to move away from the structure that the models presented in this thesis, it is possible to study the modification of the normal and log-normal behaviour and to establish stochastic perturbations to the aforementioned distributions. In the field of dynamic models we can find models such as Hull and White (2001), or Vasicek (1977) that mathematically deal with these or similar questions. However, and as cited in this thesis, these works are not clearly applicable to more complex market situations such as the ones we have developed. A first step in this direction can be found in the recent work Kizys et al. (2022), where new stochastic elements are incorporated in a one-period Portfolio Optimisation Problem (POP). Therefore, this constitutes an interesting line of research and due to their statistical weight and complexity, the role of simulation combined with heuristics can be decisive.

Appendix A

Outcomes Derived from this Thesis

In this appendix the list of works and results that have been developed during the thesis are published.

a) JCR indexed Paper

- Nieto, A.; Serra, M.; Juan, A.A.; Bayliss, C. (2022): A GA-Simheuristic for the Stochastic and Multi-Period Portfolio Optimisation Problem with Liabilities. *Journal of Simulation* (indexed in ISI SCI, 2020 IF = 2.205, Q3; 2020 SJR = 0.294, Q2). ISSN: 1747-7778. `https://doi.org/10.1080/17477778.2022.2041990`

b) Scopus indexed Papers

- Doering, J.; Nieto, A.; Juan, A.A.; Perez-Bernabeu, E. (2022): Biased-Randomized Algorithms and Simheuristics in Finance & Insurance. *Boletín de Estadística e Investigación Operativa* (Indexed in Scopus, 2020 SJR = 0.108, Q4). ISSN: 1889-3805.
- A. Nieto, A. A. Juan, C. Bayliss, and R. Kizys. Asset and Liability Management in insurance firms: A biased-randomised approach combining heuristics with Monte-Carlo simulation. *Operational Research Society*, 1:405–414, March 2021. doi: 10.36819/SW20.036. URL `https://eprints.soton.ac.uk/449945/`

- Bayliss, C.; Serra, M.; Nieto, A.; Juan, A. (2020): Combining a Matheuristic with Simulation for Risk Management of Stochastic Assets and Liabilities. *Risks* 2020, 8(4), 131 (indexed in ESCI and Scopus, 2019 SJR = 0.230, Q3). ISSN: 2227-9091. https://doi.org/10.3390/risks8040131
- Bayliss, C.; Nieto, A.; Juan, A.; Serra, M.; Gandouz, M. (2020): A Simheuristic Algorithm for Robust Asset and Liability Management. In *Proceedings – Winter Simulation Conference*, pp. 2093-2104 (Indexed in ISI WOS and Scopus, 2019 SJR = 0.410). ISSN: 0891-7736. https://informs-sim.org/wsc20papers/220.pdf
- A. Nieto, A. Juan, and R. Kizys. Asset and Liability Risk Management in Financial Markets. In *Mindful Topics on Risk Analysis and Design of Experiments*. Springer, 2022a

c) Conferences

- 26 March 2021. Technical Session 11: Hybrid Simulation. Simulation Workshop 2021, *The Operational Research Society*. In online format due to the pandemic. https://www.theorsociety.com/media/5842/sw21-programme-v2.pdf.
- 18 December 2020. Simulation Optimization under Uncertainty. *Winter Simulation Congress*. In online format due to the pandemic. https://ssl.linklings.net/conferences/wsc/wsc2020_program/views/at_a_glance.html.
- 25 June 2019. Heuristic and Evolutionary Algorithms for Continuous and Black-Box Optimization: theory and practice. *30th European Conference on Operational Research (EURO)*, Dublin.

Appendix B

Front Page of Publications

Chapter 1
Asset and Liability Risk Management in Financial Markets

Armando Nieto, Angel A. Juan, and Renatas Kizys

Abstract Most financial organisations depend on their ability to match the assets and liabilities they hold. This managerial challenge has been traditionally modelled as a series of optimisation problems, which have been mostly solved by using exact methods such as mathematical and stochastic programming. The chapter reviews the main works in this area, with a special focus on three different problems: duration immunisation, multi-stage stochastic programming, and dynamic stochastic control. Hence, the main results obtained so far are analysed, and the open challenges and limitations of the current methods are identified. To deal with these open challenges, we propose the incorporation of new heuristic-based algorithms and simulation-optimisation methods.

1.1 Introduction

All financial companies need to manage the risk associated with their liabilities. This is achieved by properly selecting a convenient set of assets from the market, which are then assigned to cover liabilities, thus reducing the risk of bankruptcy. However, both assets and liabilities are exposed to an innumerable amount of external factors, which need to be factored in order to

Armando Nieto
IN3 – Computer Science Dept., Universitat Oberta de Catalunya and Divina Pastora Seguros, Spain, e-mail: anietoran@uoc.edu

Angel A. Juan (corresponding author)
IN3 – Computer Science Dept., Universitat Oberta de Catalunya and Euncet Business School, Spain, e-mail: ajuanp@uoc.edu

Renatas Kizys
Southampton Business School, University of Southampton, United Kingdom, e-mail: r.kizys@soton.ac.uk

Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A SIMHEURISTIC ALGORITHM FOR RELIABLE ASSET AND LIABILITY MANAGEMENT UNDER UNCERTAINTY SCENARIOS

Christopher Bayliss
Marti Serra
Mariem Gandouz
Angel A. Juan

IN3 – Computer Science Dept.
Universitat Oberta de Catalunya
Av. Carl Friedrich Gauss 5
Castelldefels, 08860, SPAIN

Armando Nieto

Universitat Oberta de Catalunya &
Divina Pastora Seguros
Calle Xativa 23
Valencia, 46002, SPAIN

ABSTRACT

The management of assets and liabilities is of critical importance for insurance companies and banks. Complex decisions need to be made regarding how to assign assets to liabilities in such a way that the overall benefit is maximised over a time horizon. In addition, the risk of not being able to cover the liabilities at any given time must be kept under a certain threshold level. This optimisation challenge is known in the literature as the asset and liability management (ALM) problem. In this work, we propose a biased-randomized (BR) algorithm to solve a deterministic version of the ALM problem. Firstly, we outline a greedy heuristic. Secondly, we transform it into a BR algorithm by employing skewed probability distributions. The BR algorithm is then extended into a simheuristic by incorporating Monte-Carlo simulation to deal with the stochastic version of the problem.

1 INTRODUCTION

Financial institutions have to face some critical risk-management processes. Among such processes, asset and liability management (ALM) is of paramount importance due to its potential consequences. ALM consists of a range of techniques necessary to invest adequately, so that the firm's long-term liabilities are met (Ziemba et al. 1998). For an insurance company, a liability constitutes the legal responsibility to repay the insurance contributions that the customer has been making over an agreed length of time, which are increased by the interest rate. This is a typical transaction of pension or life insurance intended to secure retirement income, which gives rise to a three-tier financial problem. First, the insurance company receives the customer's premium. Second, the company invests this premium in the long term, so that the financial benefit envisaged in the insurance policy is secured. Third, in the event of the customer's retirement or death, the insurance company needs to have sufficient funds to meet its liability to the customer. While the aforementioned financial problem unfolds, the insurance company is confronted with a range of risks, which arise either from its role as a financial intermediary or due to complex regulations as well as economic and social policies. If the insurer's obligation to the customer is not honoured, its default becomes a likely scenario. A default can be very costly for the firm, since it can inflict a loss of credibility and reputation. On the one hand, it can face legal action from its creditors. As a result the insurer may be forced to pay

risks

MDPI

Article

Combining a Matheuristic with Simulation for Risk Management of Stochastic Assets and Liabilities

Christopher Bayliss [1,2], Marti Serra [1], Armando Nieto [1,3] and Angel A. Juan [1,*]

[1] IN3—Computer Science Department, Universitat Oberta de Catalunya, 08018 Barcelona, Spain; christopher.bayliss@liverpool.ac.uk (C.B.); rodesdecarro@uoc.edu (M.S.); anietoran@uoc.edu (A.N.)
[2] Management School, University of Liverpool, Liverpool L69 7ZH, UK
[3] Divina Pastora Seguros, Calle Xativa 23, 46002 Valencia, Spain
* Correspondence: ajuanp@uoc.edu

Received: 31 October 2020; Accepted: 1 December 2020; Published: 4 December 2020

Abstract: Specially in the case of scenarios under uncertainty, the efficient management of risk when matching assets and liabilities is a relevant issue for most insurance companies. This paper considers such a scenario, where different assets can be aggregated to better match a liability (or the other way around), and the goal is to find the asset-liability assignments that maximises the overall benefit over a time horizon. To solve this stochastic optimisation problem, a simulation-optimisation methodology is proposed. We use integer programming to generate efficient asset-to-liability assignments, and Monte-Carlo simulation is employed to estimate the risk of failing to pay due liabilities. The simulation results allow us to set a safety margin parameter for the integer program, which encourage the generation of solutions satisfying a minimum reliability threshold. A series of computational experiments contribute to illustrate the proposed methodology and its utility in practical risk management.

Keywords: assets and liabilities management; risk management; uncertainty; matheuristics; simulation

1. Introduction

Within the enormous variety of insurance types that we can find, long-term life insurance stands out for its complexity in terms of financial management. The cash flows generated by these insurances extend over several decades and play an important role in the social sphere since they have a close relationship with pensions and retirements and, therefore, with people's vital planning. For this reason, legislation and administrative authorities play a special role in ensuring that insurers faithfully comply with their commitments. The fact that they are extended in the long term, or in the very long term, generates a series of difficulties for their management because the insurer must plan the necessary income with enormous precision to cover its future commitments. Therefore, it is a requirement that the insurer has a range of techniques that allow for matching its assets, as long-term income generators, with its liabilities. Conventionally, we refer to this set of techniques as asset and liability management (ALM) (Ziemba et al. 1998), and it has raised the interest of numerous researchers over the last few years, with a wide variety of approaches being proposed. One of most popular solutions to this asset management problem is cash-flow matching (Iyengar and Ma 2009), whose main objective is to ensure the timely payment of the liabilities. This approach minimises the number of contractual breaches. Due to the volatility of the financial markets, we always have uncertainty regarding income, and this will be linked to the quality of financial assets. Moreover, the credit quality of assets plays a fundamental role, in particular when we deal with bonds, which are widely used in the insurance industry (Gründl et al. 2016). When the default event occurs, the price of the bond is immediately

Proceedings of the Operational Research Society Simulation Workshop 2021 (SW21)
M. Fakhimi, D. Robertson, and T. Boness, eds.
DOI: https://doi.org/10.36819/SW21.044

ASSET AND LIABILITY MANAGEMENT IN INSURANCE FIRMS: A BIASED-RANDOMISED APPROACH COMBINING HEURISTICS WITH MONTE-CARLO SIMULATION

Mr. Armando Nieto

Divina Pastora Seguros
Calle Xativa 23
46004 Valencia, SPAIN
armando.nieto@divinapastora.com

Dr. Angel A. Juan
Dr. Christopher Bayliss

Universitat Oberta de Catalunya – IN3
Av. Carl Friedrich Gauss 5
08860 Castelldefels, SPAIN
{ajuanp, cbayliss}@uoc.edu

Dr. Renatas Kizys

University of Southampton
University Road
Southampton SO17 1BJ, UK
r.kizys@soton.ac.uk

ABSTRACT

The management of assets and liabilities is of critical importance for insurance companies and banks. Complex decisions need to be made regarding how to assign assets to liabilities such in a way that the overall benefit is maximised over a multi-period horizon. At the same time, the risk of not being able to cover the liabilities at any given period must be kept under a certain threshold level. This optimisation problem is known in the literature as the asset and liability management (ALM) problem. In this work, we propose a biased-randomised algorithm to solve a real-life instance of the ALM problem. Firstly, we outline a greedy heuristic. Secondly, we transform it into a probabilistic algorithm by employing Monte-Carlo simulation and biased-randomisation techniques. According to our computational tests, the probabilistic algorithm is able to generate, in short computing times, solutions that outperform by far the ones currently practised in the sector.

Keywords: Heuristics, Asset and Liability Management, Biased Randomised Algorithm, Monte Carlo

1 INTRODUCTION

Financial institutions have to face some critical risk-management processes (Cornett and Saunders 2003). Among such processes, asset and liability management (ALM) is of paramount importance due to its potential consequences. ALM consists of a range of techniques necessary to invest adequately, so that the firm's long-term liabilities are met (Ziemba et al. 1998). For an insurance company, a liability constitutes the legal responsibility to repay the insurance contributions that the customer has been making over an agreed length of time, which are increased by the interest rate. This is a typical transaction of pension or life insurance intended to secure retirement income, which gives rise to a three-tier financial problem. First, the insurance company receives the customer's premium. Second, the company invests this premium in the long term, so that the financial benefit envisaged in the insurance policy is secured. Third, in the event of the customer's retirement or death, the insurance company needs to have sufficient funds to meet

JOURNAL OF SIMULATION
https://doi.org/10.1080/17477778.2022.2041990

ORIGINAL PAPER

A GA-simheuristic for the stochastic and multi-period portfolio optimisation problem with liabilities

Armando Nieto[a], Marti Serra[a], Angel A. Juan[b] and Christopher Bayliss[c]

[a]IN3 – Computer Science Department, Universitat Oberta de Catalunya, Barcelona, Spain; [b]Department of Applied Statistics & Operations Research and Quality, Universitat Politècnica de València, Alcoy, Spain; [c]Management School, University of Liverpool, Liverpool, UK

ABSTRACT
The efficient management of assets to cover a firm's liabilities over a multi-period horizon is a relevant challenge for many financial companies. Even in its deterministic version, this problem is complex since managers have to make difficult decisions about their asset portfolio each period. With the goal of maximising the expected terminal wealth in a scenario under uncertainty, this paper proposes a novel simheuristic approach that integrates Monte Carlo simulation at different stages of a Genetic Algorithm. Our approach is capable of generating effective solutions to the considered problem in relatively short computational times. Moreover, our simheuristic is enriched with several "smoothing" techniques that enhance the attractiveness for managers of the generated solutions, so they can be effectively employed in real-life applications. A series of computational experiments, including the use of advanced evolutionary strategies, illustrate these concepts and justify the advantages of including simulation in financial optimisation problems under uncertainty.

ARTICLE HISTORY
Received 30 June 2021
Accepted 6 February 2022

KEYWORDS
Simulation; simheuristics; asset and liability management; genetic algorithms

1. Introduction

Dealing with liabilities (financial obligations) in volatile markets with a limited yield or return is one of the main challenges that insurance firms have to face. The insurer is forced to pay the amounts agreed in the policy at a specific maturity date. In order to do that, a set of firm's assets have to be "frozen" in advance to cover future payments. Asset-liability management (ALM) refers to the study of techniques employed in selecting the appropriate assets to face the firm's liabilities over time. The financial market is constituted by a huge number of companies, whose aim is to transform an initial wealth into large returns during a giving time horizon. According to the consumer preference theory (Mankiw et al., 2020), an investor would select those assets that provide the highest returns, while taking into account her budget constraint. As markets are plenty of uncertainty, the volatility of the assets also has to be considered. This transforms the ALM into a stochastic and multi-period portfolio optimisation problem. Markowitz (1952) considered the assets as random variables, so he formulated the classical mean-variance model, in which different amounts of assets have to be selected in order to maximise a portfolio's return, while considering a specific volatility. Alternatively, one might want to minimise the risk subject to achieving a user-defined level of return (Kizys et al., 2019). In any case, these approaches are only valid if our wealth is not associated with a set of liabilities. Whenever it is, we need to consider a different strategy, since the obtained returns are employed to cover liabilities. In general, the purpose of ALM is to support the assets selection process – i.e., by selecting those that maximise returns while maintaining enough financial resources to satisfy the liabilities. Among the different ALM approaches in the scientific literature, the following ones are the most popular ones: *(i)* duration theory, which is based on the work of Macaulay (1938), who assumes that the interest rate is almost constant and also that assets and liabilities have the same present value; *(ii)* cash-flow matching, where we select assets in a way that allows us to match them with our debts; and *(iii)* stochastic control theory, a quite theoretical approach that studies the evolution of assets and liabilities in a continuous and stochastic scenario.

This work focuses on the cash flow matching strategy, which extends the Markowitz's theory to an ALM scenario. Hence, we will consider a realistic mean-variance problem in a multi-period context under uncertainty, where our decision variables are the amount of assets we have to buy or sell in each period of time, considering an initial wealth and a given set of liabilities along time. In other words, given an initial wealth and a set of financial duties that need to be covered in the future, the goal is to find the ALM plan that maximises our expected wealth at the end of the time horizon, taking into account different uncertainty sources.

CONTACT Armando Nieto armandonietoranero@gmail.com IN3 – Computer Science Department, Universitat Oberta de Catalunya, 08018 Barcelona, Spain

Biased-Randomized Algorithms and Simheuristics in Finance & Insurance

Jana Doering, Armando Nieto
ICSO SGR Consolidated Research Group
Universitat Oberta de Catalunya
{jdoering, anietoran}@uoc.edu

Angel A. Juan*, Elena Perez-Bernabeu
Department of Applied Statistics and Operations Research
Universitat Politècnica de València
{ajuanp, elenapb}@eio.upv.es

Abstract

Managerial decisions in the area of finance and insurance can often be modeled as combinatorial optimization problems. It is also frequent that these optimization problems fall into the category of NP-hard ones, which justifies the need for using metaheuristic algorithms when tackling large-sized instances. In addition, decision-making in real-life financial & insurance activities is usually performed in scenarios under uncertainty. Hence, stochastic versions of the aforementioned NP-hard problems have to be considered, and simulation-optimization methods are required in order to obtain high-quality solutions. This paper analyzes how biased-randomized techniques (which transform greedy heuristics into probabilistic algorithms) and simheuristics (hybridization of simulation with metaheuristics) can be employed to efficiently cope with a variety of challenging optimization problems, even those under uncertainty scenarios.

Keywords: Optimization, Finance, Insurance, Metaheuristics, Biased-Randomized Algorithms, Simheuristics
AMS Subject classifications: 90-10, 90B50, 90B99, 68W20, 68T20

1 Introduction

Numerous managerial challenges in the areas of finance and insurance (F&I) can be modeled as combinatorial optimization problems. Traditionally, exact methods have been employed in determining optimal solutions to these problems. This is the case, for instance, of the classical Markowitz model [1], which minimizes the risk associated with a portfolio of assets while establishing a minimum threshold for its return value. Exact methods, however, present certain limitations when solving large-sized portfolio optimization problems with richer

*Corresponding Author.

Bibliography

H. Abdollahi. Multi-Objective Programming for Asset-Liability Management: The Case of Iranian Banking Industry. *International Journal of Industrial Engineering and Production Research*, 31:75–85, 03 2020. doi: 10.22068/ijiepr.31.1.75.

A. Adam. *Handbook of Asset and Liability Management: From models to optimal return strategies*. John Wiley & Sons, 2008.

P. Adamko, T. Kliestik, and M. Birtus. History of credit risk models. In *2nd international conference on economics and social science (ICESS 2014), Information Engineering Research Institute, Advances in Education Research*, volume 61, pages 148–153, 2014.

K. C. Ahlgrim, S. P. D'Arcy, and R. W. Gorvett. The Effective Duration and Convexity of Liabilities for Property-Liability Insurers Under Stochastic Interest Rates. *The Geneva Papers on Risk and Insurance Theory*, 29(1):75–108, 2004. ISSN 09264957, 15736954. URL `http://www.jstor.org/stable/41953443`.

E. I. Altman. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*, 23(4):589–609, 1968. ISSN 00221082, 15406261. URL `http://www.jstor.org/stable/2978933`.

S. Alvarez Fernandez, D. Ferone, A. Juan, and D. Tarchi. A simheuristic algorithm for video streaming flows optimisation with Qos threshold modelled as a stochastic single-allocation p-hub median problem. *Journal of Simulation*, 1:1–14, 2021.

K. J. Arrow. *The Limits of Organization*. WW Norton & Company, 1974.

J. Barbarin and P. Devolder. Risk measure and fair valuation of an investment guarantee in life insurance. *Insurance: Mathematics and Economics*, 37(2 SPEC. ISS.):297–323, 2005.

R. Bauer, R. Hoevenaars, and T. Steenkamp. *Asset Liability Management.* Oxford University Press, United States, July 2006. ISBN 0199272468. doi: 10.1093/oxfordhb/9780199272464.003.0021.

C. Bayliss, M. Serra, M. Gandouz, A. A. Juan, and A. Nieto. A Simheuristic Algorithm for Reliable Asset and Liability Management Under Uncertainty Scenarios. *2020 Winter Simulation Conference (WSC)*, 1:2093–2104, 12 2020a. doi: 10.1109/WSC48552.2020.9384044.

C. Bayliss, M. Serra, A. Nieto, and A. A. Juan. Combining a Matheuristic with Simulation for Risk Management of Stochastic Assets and Liabilities. *Risks*, 8(4), 2020b. ISSN 2227-9091. doi: 10.3390/risks8040131. URL `https://www.mdpi.com/2227-9091/8/4/131`.

W. H. Beaver. Alternative Accounting Measures as Predictors of Failure. *The Accounting Review*, 43(1):113–122, 1968. ISSN 00014826. URL `http://www.jstor.org/stable/244122`.

O. Benkato, M. Haddad, N. Al-Loughani, and M. Baur. Coping with the risk of interest rate fluctuations: the case of Kuwait banks. *Journal of Business & Economics Research (JBER)*, 1, 02 2011.

A. Berkelaar and R. Kouwenberg. A liability-relative drawdown approach to pension Asset Liability Management. *Journal of Asset Management*, 11(2-3):194–217, 06 2010. doi: 10.1057/jam.2010.13.

D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods.* Academic press, 2014.

G. O. Bierwag, I. Fooladi, and G. S. Roberts. Designing an immunized portfolio: Is m-squared the key? *Journal of Banking & Finance*, 17(6):1147–1170, 1993.

M. Birattari, L. Paquete, T. Strutzle, and K. Varrentrapp. Classification of Metaheuristics and Design of Experiments for the Analysis of Components. *Tech. Rep. AIDA-01-05*, 2001.

G. Boender, C. Dert, F. Heemskerk, and H. Hoek. A scenario approach of ALM. *Handbook of Asset and Liability Management - Set*, 2:829–860, 12 2008. doi: 10.1016/B978-044453248-0.50024-1.

E. Bogentoft, H. E. Romeijn, and S. Uryasev. Asset / Liability Management for pension funds using CVaR constraints. *Journal of Risk Finance*, 3(1):57–71, 2001.

G. E. P. Box and M. E. Muller. A Note on the Generation of Random Normal Deviates. *Annals of Mathematical Statistics*, 29:610–611, 1958.

S. C. Brailsford, T. Eldabi, M. Kunc, N. Mustafee, and A. F. Osorio. Hybrid simulation modelling in operational research: A state-of-the-art review. *European Journal of Operational Research*, 278(3):721–737, 2019.

E. Briys and F. De Varenne. Life insurance in a contingent claim framework: Pricing and regulatory implications. *The GENEVA Papers on Risk and Insurance Theory*, 19(1):53–72, 1994.

G. Cabrera, A. A. Juan, D. Lázaro, J. M. Marquès, and I. Proskurnia. A Simulation-optimization Approach to Deploy Internet Services in Large-scale Systems with User-provided Resources. *Simulation*, 90(6):644–659, 2014.

L. Calvet, A. Ferrer, M. I. Gomes, A. A. Juan, and D. Masip. Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. *Computers & Industrial Engineering*, 94:93–104, 2016.

D. R. Carino, T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba. The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces*, 24(1):29–49, 1994.

D. R. Cariño and W. T. Ziemba. Formulation of the Russell-Yasuda Kasai financial planning model. *Operations Research*, 46(4):433–449, 1998.

D. R. Cariño, D. H. Myers, and W. T. Ziemba. Concepts, technical issues, and uses of the Russell-Yasuda Kasai financial planning model. *Operations Research*, 46 (4):450–462, 1998.

T. M. Carsey and J. J. Harden. *Monte Carlo simulation and resampling methods for social science*. Sage Publications, 2013.

H. Chang. Dynamic Mean–Variance portfolio selection with liability and stochastic interest rate. *Economic Modelling*, 51:172–182, 12 2015. doi: 10.1016/j.econmod.2015.07.017.

M. Chica, A. A. Juan, C. Bayliss, O. Cordón, and W. D. Kelton. Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation. *SORT-Statistics and Operations Research Transactions*, 1:311–334, 2020.

M. C. Chiu and D. Li. Asset and Liability Management under a continuous-time Mean-Variance optimization framework. *Insurance: Mathematics and Economics*, 39(3):330–355, 2006.

M. C. Chiu and D. Li. Asset-Liability Management under the safety-first principle. *Journal of Optimization Theory and Applications*, 143(3):455–478, 2009.

M. C. Chiu and H. Y. Wong. Mean-Variance Asset-Liability Management: cointegrated assets and insurance liability. *European Journal of Operational Research*, 223(3):785–793, 2012.

M. C. Chiu and H. Y. Wong. Optimal investment for insurers with the extended cir interest rate model. *Abstract and Applied Analysis*, 2014:1–12, 06 2014a. doi: 10.1155/2014/129474.

M. C. Chiu and H. Y. Wong. Mean-Variance Asset-Liability Management with asset correlation risk and insurance liabilities. *Insurance: Mathematics and Economics*, 59:300–310, 2014b.

M. Choudhry. *Bank Asset and Liability Management: strategy, trading, analysis*. John Wiley & Sons, 2011.

G. Consigli and M. A. H. Dempster. Dynamic stochastic programming for Asset-Liability Management. *Annals of Operations Research*, 81:131–161, 1998.

A. Consiglio, D. Saunders, and S. A. Zenios. Asset and Liability Management for insurance products with minimum guarantees: The UK case. *Journal of Banking and Finance*, 30(2):645–667, 2006.

A. Consiglio, F. Cocco, and S. A. Zenios. Asset and Liability Modelling for participating policies with guarantees. *European Journal of Operational Research*, 186 (1):380–404, 2008.

M. M. Cornett and A. Saunders. *Financial institutions management: A risk management approach*. McGraw-Hill/Irwin, 2003.

S. Corsaro, P. De Angelis, Z. Marino, F. Perla, and P. Zanetti. On parallel Asset-Liability Management in life insurance: a forward risk-neutral approach. *Parallel Computing*, 36(7):390–402, 2010. ISSN 0167-8191. doi: https://doi.org/10.1016/j.parco.2009.10.002. URL `https://www.sciencedirect.com/science/article/pii/S0167819109001203`. Parallel and Distributed Computing in Finance.

J. C. Cox, J. E. Ingersoll Jr, and S. A. Ross. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407, 1985.

M. Crouhy, D. Galai, and R. Mark. A Comparative Analysis of Current Credit Risk Models. *Journal of Banking & Finance*, 24:59–117, 01 2000. doi: 10.1016/S0378-4266(99)00053-9.

G. Dash and N. Kajiji. A nonlinear goal programming model for efficient Asset-Liability Management of property-liability insurers. *INFOR: Information Systems and Operational Research*, 43(2):135–156, 2005.

J. De Armas, A. A. Juan, J. M. Marquès, and J. P. Pedroso. Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176, 2017.

M. A. Dempster, M. Germano, E. A. Medova, and M. Villaverde. Global Asset Liability Management. *British Actuarial Journal*, 9(1):137–195, 2003. ISSN 13573217, 20440456. URL `http://www.jstor.org/stable/41141589`.

J. Detemple and M. Rindisbacher. Dynamic Asset Liability Management with tolerance for limited shortfalls. *Insurance: Mathematics and Economics*, 43(3):281–294, 2008.

P. Devolder, M. Bosch Princep, and I. Dominguez Fabian. Stochastic optimal control of annuity contracts. *Insurance: Mathematics and Economics*, 33(2):227–238, 2003.

J. Doering, R. Kizys, A. A. Juan, A. Fito, and O. Polat. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives*, 6:100121, 2019a. doi: https://doi.org/10.1016/j.orp.2019.100121.

J. Doering, R. Kizys, A. A. Juan, À. Fitó, and O. Polat. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives*, 6:100121, 2019b. ISSN 2214-7160. doi: https://doi.org/10.1016/j.orp.2019.100121. URL `https://www.sciencedirect.com/science/article/pii/S2214716019300399`.

J. Doering, A. Nieto, A. A. Juan, and E. Perez-Bernabeu. Biased-Randomized Algorithms and Simheuristics in Finance and Insurance. *Boletín de Estadística e Investigación Operativa*, 2022. ISSN 1889-3805.

O. Dominguez, D. Guimarans, A. A. Juan, and I. de la Nuez. A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, 255(2):442–462, 2016.

J. Dreo. Classification of metaheuristics, October 2017. URL `http://metah.nojhan.net/`.

G. Dutta, H. V. Rao, S. Basu, and M. K. Tiwari. Asset Liability Management model with decision support system for life insurance companies: Computational results. *Computers & Industrial Engineering*, 128:985–998, 2019.

L. F. Escudero, A. Garín, M. Merino, and G. Pérez. On multistage stochastic integer programming for incorporating logical constraints in Asset and Liability Management under uncertainty. *Computational Management Science*, 6(3):307–327, 2009.

F. Fabozzi. *Duration, Convexity, and Other Bond Risk Measures*. Frank J. Fabozzi

Series. Wiley, 1999. ISBN 9781883249632. URL `https://books.google.es/books?id=7i6ob9SB5jgC`.

J. Faulin and A. A. Juan. The ALGACEA-1 method for the capacitated vehicle routing problem. *International Transactions in Operational Research*, 15(5):599–621, 2008.

J. Faulin, M. Gilibert, A. A. Juan, X. Vilajosana, and R. Ruiz. SR-1: A simulation-based algorithm for the capacitated vehicle routing problem. In *Proceedings of the 2008 Winter Simulation Conference*, pages 2708–2716. IEEE, 2008a.

J. Faulin, A. A. Juan, C. Serrat, and V. Bargueno. Predicting Availability Functions in Time-dependent Complex Systems with SAEDES Simulation Algorithms. *Reliability Engineering & System Safety*, 93(11):1761–1771, 2008b.

J. L. Fernández, A. M. Ferreiro-Ferreiro, J. A. García-Rodríguez, and C. Vázquez. GPU parallel implementation for Asset-Liability Management in insurance companies. *Journal of Computational Science*, 24:232–254, 2018.

D. Ferone, A. Gruler, P. Festa, and A. A. Juan. Enhancing and Extending the Classical GRASP Framework with Biased Randomisation and Simulation. *Journal of the Operational Research Society*, 70(8):1362–1375, 2019.

A. Ferrer, D. Guimarans, H. Ramalhinho, and A. A. Juan. A BRILS Metaheuristic for Non-smooth Flow-shop Problems with Failure-risk Costs. *Expert Systems with Applications*, 44:177–186, 2016.

R. Ferstl and A. Weissensteiner. Asset-Liability Management under time-varying investment opportunities. *Journal of Banking & Finance*, 35:182–192, 01 2011. doi: 10.1016/j.jbankfin.2010.07.028.

C. Fikar, A. A. Juan, E. Martinez, and P. Hirsch. A Discrete-event Driven Metaheuristic for Dynamic Home Service Routing with Synchronised Trip Sharing. *European Journal of Industrial Engineering*, 10(3):323–340, 2016.

L. Fisher and R. L. Weil. Coping with the Risk of Interest-Rate Fluctuations: Returns to Bondholders from Naïve and Optimal Strategies. *The Journal of*

Business, 44(4):408–431, 1971. ISSN 00219398, 15375374. URL http://www.jstor.org/stable/2352056.

S. . Fleten, K. Høyland, and S. W. Wallace. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, 140 (1):37–49, 2002.

H. G. Fong and O. A. Vasicek. A risk minimizing strategy for portfolio immunization. *The Journal of Finance*, 39(5):1541–1546, 1984. ISSN 00221082, 15406261. URL http://www.jstor.org/stable/2327743.

L. Gajek. Axiom of solvency and portfolio immunization under random interest rates. *Insurance: Mathematics and Economics*, 36(3):317–328, 2005. ISSN 0167-6687. doi: https://doi.org/10.1016/j.insmatheco.2005.02.009. URL https://www.sciencedirect.com/science/article/pii/S016766870500034X.

M. Gilli, D. Maringer, and E. Schumann. *Numerical methods and optimization in finance*. Academic Press, 2011.

D. Giokas and M. Vassiloglou. A goal programming model for bank Assets and Liabilities Management. *European Journal of Operational Research*, 50 (1):48–60, 1991. ISSN 0377-2217. doi: https://doi.org/10.1016/0377-2217(91)90038-W. URL https://www.sciencedirect.com/science/article/pii/037722179190038W.

F. W. Glover and G. A. Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

J. Gondzio and A. Grothey. Solving nonlinear financial planning problems with 10^9 decision variables on massively parallel architectures. *WIT Transactions on Modelling and Simulation*, 43, 2006.

J. Gondzio and R. Kouwenberg. High-performance computing for Asset-Liability Management. *Operations Research*, 49(6):879–891, 12 2001. doi: 10.1287/opre.49.6.879.10015.

S. Gonzalez-Martin, A. A. Juan, D. Riera, M. G. Elizondo, and J. J. Ramos. A Simheuristic Algorithm for Solving the Arc Routing Problem with Stochastic Demands. *Journal of Simulation*, 12(1):53–66, 2018.

E. M. Gonzalez-Neira, D. Ferone, S. Hatami, and A. A. Juan. A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79:23–36, 2017.

A. Grasas, A. A. Juan, and H. R. Lourenço. SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77, 2016.

A. Grasas, A. A. Juan, J. Faulin, J. de Armas, and H. Ramalhinho. Biased randomization of heuristics using skewed probability distributions: A survey and some applications. *Computers & Industrial Engineering*, 110:216–228, 2017.

A. Gruler, C. Fikar, A. A. Juan, P. Hirsch, and C. Contreras-Bolton. Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization. *Journal of Simulation*, 11(1):11–19, 2017a.

A. Gruler, C. L. Quintero-Araujo, L. Calvet, and A. A. Juan. Waste collection under uncertainty: a simheuristic based on variable neighbourhood search. *European Journal of Industrial Engineering*, 11(2):228–255, 2017b.

A. Gruler, J. Panadero, J. de Armas, J. A. Moreno, and A. A. Juan. Combining Variable Neighborhood Search with Simulation for the Inventory Routing Problem with Stochastic Demands and Stock-outs. *Computers & Industrial Engineering*, 123:278–288, 2018.

A. Gruler, J. Panadero, J. de Armas, J. A. Moreno, and A. A. Juan. A Variable Neighborhood Search Simheuristic for the Multiperiod Inventory Routing Problem with Stochastic Demands. *International Transactions in Operational Research*, 27 (1):314–335, 2020.

H. Gründl, M. I. Dong, and J. Gal. The evolution of insurer portfolio investment strategies for long-term investing. *OECD Journal: Financial Market Trends*, 2016 (2):1–55, 2016.

D. Guimarans, O. Dominguez, J. Panadero, and A. A. Juan. A Simheuristic Approach for the Two-dimensional Vehicle Routing Problem with Stochastic Travel Times. *Simulation Modelling Practice and Theory*, 89:1–14, 2018.

N. Gülpinar and D. Pachamanova. A robust optimization approach to Asset-Liability Management under time-varying investment opportunities. *Journal of Banking and Finance*, 37(6):2031–2041, 2013.

N. Gülpınar, D. Pachamanova, and E. Çanakoğlu. A robust Asset–Liability Management framework for investment products with guarantees. *OR Spectrum*, 38 (4):1007–1041, 2016.

A. Harper, N. Mustafee, and M. Yearworth. Facets of trust in simulation studies. *European Journal of Operational Research*, 289(1):197–213, 2021. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2020.06.043. URL `https://www.sciencedirect.com/science/article/pii/S0377221720306019`.

S. Hatami, L. Calvet, V. Fernández-Viagas, J. M. Framiñán, and A. A. Juan. A Simheuristic Algorithm to Set Up Starting Times in the Stochastic Parallel Flowshop Problem. *Simulation Modelling Practice and Theory*, 86:55–71, 2018.

N. Hibiki. Multi-period stochastic optimization models for dynamic asset allocation. *Journal of Banking and Finance*, 30(2):365–390, 2006. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2005.04.027. URL `https://www.sciencedirect.com/science/article/pii/S0378426605000695`.

J. R. Hicks. *Value and Capital: An Inquiry into some Fundamental Principles of Economic Theory*. Oxford University Press, 1975.

N. J. Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, 2009.

J. H. Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, 1992.

H. Huang and T. Willemain. Input modelling for financial simulations using the bootstrap. *Journal of Simulation*, 1(1):39–52, 2006.

J. Hull and A. White. Pricing Interest-Rate-Derivative Securities. *Review of Financial Studies*, 3(4):573–592, 02 1990. doi: 10.1093/rfs/3.4.573.

J. Hull and A. White. The general hull–white model and super calibration. *Financial Analysts Journal*, 57(6):34–43, 2001. URL `https://ssrn.com/abstract=1295228`.

J. C. Hull, M. Predescu, and A. White. Bond prices, default probabilities and risk premiums. *Default Probabilities and Risk Premiums (March 9, 2005)*, 2005.

K. Høyland and S. W. Wallace. Analyzing legal regulations in the Norwegian life insurance business using a multistage Asset-Liability Management model. *European Journal of Operational Research*, 134(2):293–308, 2001.

G. Iyengar and A. K. C. Ma. Cash flow matching: a risk management approach. *North American Actuarial Journal*, 13(3):370–378, 2009. doi: 10.1080/10920277.2009.10597562. URL `https://doi.org/10.1080/10920277.2009.10597562`.

R. Josa-Fombellida and J. P. Rincón-Zapatero. Optimal asset allocation for aggregated defined benefit pension funds with stochastic interest rates. *European Journal of Operational Research*, 201(1):211–221, 2010.

A. A. Juan, J. Faulin, R. Ruiz, B. Barrios, M. Gilibert, and X. Vilajosana. Using oriented random search to provide a set of alternative solutions to the capacitated vehicle routing problem. In J. W. Chinneck, B. Kristjansson, and M. J. Saltzman, editors, *Operations Research and Cyber-Infrastructure*, volume 1, pages 331–345, Boston, MA, 2009. Springer US.

A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62 – 72, 2015a. ISSN 2214-7160. doi: https://doi.org/10.1016/j.orp.2015.03.001. URL `http://www.sciencedirect.com/science/article/pii/S221471601500007X`.

A. A. Juan, J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62–72, 2015b.

A. A. Juan, W. D. Kelton, C. S. Currie, and J. Faulin. Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas. *Proceedings of the 2018 Winter Simulation Conference*, 1:3048–3059, 2018.

R. Khabibullin, A. Ponomarenko, and S. Seleznev. Forecasting the implications of foreign exchange reserve a ccumulation with a microsimulation model. *Journal of Simulation*, pages 1–14, 2020.

D. E. Kirk. *Optimal control theory: an introduction.* Courier Corporation, 2012.

R. Kizys, A. A. Juan, B. Sawik, and L. Calvet. A biased-randomized iterated local search algorithm for rich portfolio optimization. *Applied Sciences - Basel*, 9(17), 2019a. doi: 10.3390/app9173509.

R. Kizys, A. A. Juan, B. Sawik, and L. Calvet. A biased-randomized iterated local search algorithm for rich portfolio optimization. *Applied Sciences*, 9(17):3509, 2019b.

R. Kizys, J. Doering, A. A. Juan, O. Polat, L. Calvet, and J. Panadero. A simheuristic algorithm for the portfolio optimization problem with random returns and noisy covariances. *Computers & Operations Research*, 139:105631, 2022.

P. Klaassen. Discretized reality and spurious profits in stochastic programming models for Asset / Liability Management. *European Journal of Operational Research*, 101(2):374–392, 1997.

M. Koivu, T. Pennanen, and A. Ranne. Modeling assets and liabilities of a finnish pension insurance company: A VEqC approach. *Scandinavian Actuarial Journal*, 2005(1):46–76, 2005.

M. Kopa and T. Rusý. A decision-dependent randomness stochastic program for Asset–Liability Management model with a pricing decision. *Annals of Operations Research*, 04 2020. doi: 10.1007/s10479-020-03583-y.

R. Kouwenberg. Scenario generation and stochastic programming models for Asset Liability Management. *European Journal of Operational Research*, 134(2):279–292, 2001.

R. Kouwenberg and S. A. Zenios. Stochastic programming models for Asset Liability Management. In *Handbook of Asset and Liability Management*, pages 253–303. Elsevier, 2008a.

R. Kouwenberg and S. A. Zenios. Stochastic programming models for Asset Liability Management. In *Handbook of Asset and Liability Management*, pages 253–303. Elsevier, 2008b.

O. Kramer. Genetic algorithms. In *Genetic Algorithm Essentials*, pages 11–19. Springer, 2017.

M. I. Kusy and W. T. Ziemba. A Bank Asset and Liability Management model. *Operations Research*, 34(3):356–376, 1986. URL `https://doi.org/10.1287/opre.34.3.356`.

D. Li, Y. Shen, and Y. Zeng. Dynamic derivative-based investment strategy for Mean–Variance Asset–Liability Management with stochastic volatility. *Insurance: Mathematics and Economics*, 78:72–86, 2018a.

D. Li, Y. Shen, and Y. Zeng. Dynamic derivative-based investment strategy for Mean–Variance Asset–Liability Management with stochastic volatility. *Insurance: Mathematics and Economics*, 78:72–86, 2018b.

X. Li, X. Wu, and H. Yao. Multi-period Asset-Liability Management with cash flows and probability constraints: A mean-field formulation approach. *Journal of the Operational Research Society*, 1:1–18, 2019.

Z. Liang and M. Ma. Optimal dynamic asset allocation of pension fund in mortality and salary risks framework. *Insurance: Mathematics and Economics*, 64:151–161, 2015.

F. R. Macaulay. *Some Theoretical Problems Suggested by the Movements of Interest Rates, Bond Yields and Stock Prices in the United States since 1856*. National

Bureau of Economic Research, Inc, 1938. URL http://EconPapers.repec.org/RePEc:nbr:nberbk:maca38-1.

N. G. Mankiw, R. D. Kneebone, K. J. McKenzie, and N. Rowe. *Principles of Macroeconomics*. Toronto: Nelson Education, 2007.

R. N. Mantegna and H. E. Stanley. *Introduction to econophysics: correlations and complexity in finance*. Cambridge University Press, 1999.

H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082, 15406261. URL http://www.jstor.org/stable/2975974.

S. Martin, D. Ouelhadj, P. Beullens, E. Ozcan, A. A. Juan, and E. K. Burke. A multi-agent based cooperative approach to scheduling and routing. *European Journal of Operational Research*, 254(1):169–178, 2016.

R. C. Merton. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance*, 29(2):449–470, 1974.

G. Mitra and K. Schwaiger. *Asset and Liability Management Handbook*. Springer, 2011.

S. B. Movahed and S. Fanian. *Optimum Portfolio Construction considering Value at Risk using Genetic Algorithms and Monte Carlo Simulation*. Simon Fraser University, 2019.

J. M. Mulvey, D. P. Rosenbaum, and B. Shetty. Strategic financial risk management and operations research. *European Journal of Operational Research*, 97(1):1–16, 1997.

J. M. Mulvey, G. Gould, and C. Morgan. An Asset and Liability Management system for towers Perrin-Tillinghast. *Interfaces*, 30(1):96–114, 2000.

S. S. Nielsen and S. A. Zenios. A stochastic programming model for funding single premium deferred annuities. *Mathematical Programming, Series B*, 75(2):177–200, 1996.

A. Nieto, A. A. Juan, C. Bayliss, and R. Kizys. Asset and Liability Management in insurance firms: A biased-randomised approach combining heuristics with Monte-Carlo simulation. *Operational Research Society*, 1:405–414, March 2021. doi: 10.36819/SW20.036. URL `https://eprints.soton.ac.uk/449945/`.

A. Nieto, A. Juan, and R. Kizys. Asset and Liability Risk Management in Financial Markets. In *Mindful Topics on Risk Analysis and Design of Experiments*. Springer, 2022a.

A. Nieto, M. Serra, A. A. Juan, and C. Bayliss. A GA-simheuristic for the stochastic and multi-period portfolio optimisation problem with liabilities. *Journal of Simulation*, 2022b. doi: 10.1080/17477778.2022.2041990. URL `https://doi.org/10.1080/17477778.2022.2041990`.

B. S. Onggo, J. Panadero, C. G. Corlu, and A. A. Juan. Agri-food supply chains with stochastic demands: A multi-period inventory routing problem with perishable products. *Simulation Modelling Practice and Theory*, 97:101970, 2019.

E. V. Orlova. Model for operational optimal control of financial recourses distribution in a company. *Computer Research and Modeling*, 11(2):343–358, 2019. URL `https://doi.org/10.20537/2076-7633-2019-11-2-343-358`.

C. B. Oğuzsoy and S. Güven. Bank Asset and Liability Management under uncertainty. *European Journal of Operational Research*, 102(3):575–600, 1997.

A. Pages-Bernaus, H. Ramalhinho, A. A. Juan, and L. Calvet. Designing e-commerce supply chains: A stochastic facility-location approach. *International Transactions in Operational Research*, 26(2):507–528, 2019.

J. Pan and Q. Xiao. Optimal Asset–Liability Management with liquidity constraints and stochastic interest rates in the expected utility framework. *Journal of Computational and Applied Mathematics*, 317:371–387, 2017a.

J. Pan and Q. Xiao. Optimal Mean–Variance Asset-Liability Management with stochastic interest rates and inflation risks. *Mathematical Methods of Operations Research*, 85(3):491–519, 2017b.

J. Panadero, J. Doering, R. Kizys, A. A. Juan, and A. Fito. A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics*, 1:1–23, 2018.

J. Panadero, J. Doering, R. Kizys, A. A. Juan, and A. Fito. A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics*, 26(3):353–375, 2020.

M. Papi and S. Sbaraglia. Optimal Asset-Liability Management with constraints: a dynamic programming approach. *Applied Mathematics and Computation*, 173 (1):306–349, 2006.

G. C. Pflug and A. Pichler. *Multistage stochastic optimization*, volume 1104. Springer, 02 2016. ISBN 978-3-319-08842-6. doi: 10.1007/978-3-319-08843-3.

J. W. Pratt. Risk aversion in the small and in the large. In *Uncertainty in economics*, pages 59–79. Elsevier, 1978.

M. Qi and X. Zhao. Comparison of modeling methods for loss given default. *Journal of Banking & Finance*, 35(11):2842–2855, 2011.

C. L. Quintero-Araujo, J. P. Caballero-Villalobos, A. A. Juan, and J. R. Montoya-Torres. A biased-randomized metaheuristic for the capacitated location routing problem. *International Transactions in Operational Research*, 24(5):1079–1098, 2017. doi: 10.1111/itor.12322. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12322`.

C. L. Quintero-Araujo, A. Gruler, A. A. Juan, and J. Faulin. Using horizontal cooperation concepts in integrated routing and facility-location decisions. *International Transactions in Operational Research*, 26(2):551–576, 2019a. doi: 10.1111/itor.12479. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12479`.

C. L. Quintero-Araujo, D. Guimarans, and A. A. Juan. A simheuristic algorithm for the capacitated location routing problem with stochastic demands. *Journal of Simulation*, pages 1–18, 2019b.

M. Rabe, M. Deininger, and A. Juan. Speeding Up Computational Times in Simheuristics Combining Genetic Algorithms with Discrete-Event Simulation. *Simulation Modelling Practice and Theory*, page 102089, 2020.

L. C. Rogers, S. E. Satchell, and Y. Yoon. Estimating the volatility of stock prices: a comparison of methods that use high and low prices. *Applied Financial Economics*, 4(3):241–247, 1994.

J. Rosmarin. *An evolutionary approach to multistage portfolio optimization*. Imperial College London (University of London), 2008. URL http://hdl.handle.net/10044/1/7280.

S. Seshadri, A. Khanna, F. Harche, and R. Wyle. Method for strategic Asset-Liability Management with an application to the Federal Home Loan Bank of New York. *Operations Research*, 47(3):345–360, 1999.

W. F. Sharpe. The Sharpe Ratio. *The Journal of Portfolio Management*, 21(1): 49–58, 1994. ISSN 0095-4918. doi: 10.3905/jpm.1994.409501. URL https://jpm.pm-research.com/content/21/1/49.

E. Solares, C. A. Coello Coello, E. Fernandez, and J. Navarro. Handling uncertainty through confidence intervals in portfolio optimization. *Swarm and Evolutionary Computation*, 44:774–787, 2019. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2018.08.010. URL https://www.sciencedirect.com/science/article/pii/S2210650218300488.

A. Soler-Dominguez, A. A. Juan, and R. Kizys. A survey on financial applications of metaheuristics. *ACM Computing Surveys (CSUR)*, 50(1):15, 2017.

Z. Sun, X. Zhang, and K. Yuen. Mean-Variance Asset-Liability Management with affine diffusion factor process and a reinsurance option. *Scandinavian Actuarial Journal*, 2020:1–27, 08 2019. doi: 10.1080/03461238.2019.1658619.

E.-G. Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

J. Tobin. Liquidity preference as behavior towards risk. *The Review of Economic Studies*, 25(2):65–86, 1958.

A. Tolk, A. Harper, and N. Mustafee. Hybrid models as transdisciplinary research enablers. *European Journal of Operational Research*, 291(3):1075–1090, 2021.

O. Vasicek. An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5(2):177–188, 1977.

J. Von Neumann, O. Morgenstern, and H. W. Kuhn. *Theory of games and economic behavior (commemorative edition)*. Princeton University Press, 2007.

J. Wei and T. Wang. Time-consistent Mean–Variance Asset–Liability Management with random coefficients. *Insurance: Mathematics and Economics*, 77:84–96, 2017.

J. Wei, K. C. Wong, S. C. P. Yam, and S. P. Yung. Markowitz's Mean-Variance Asset-Liability Management with regime switching: A time-consistent approach. *Insurance: Mathematics and Economics*, 53(1):281–291, 2013.

S. Xie. Continuous-time Mean-Variance portfolio selection with liability and regime switching. *Insurance: Mathematics and Economics*, 45(1):148–155, 2009.

S. Xie, Z. Li, and S. Wang. Continuous-time portfolio selection with liability: Mean-Variance model and stochastic LQ approach. *Insurance: Mathematics and Economics*, 42(3):943–953, 2008.

Y. Zeng and Z. Li. Asset-Liability Management under benchmark and Mean-Variance criteria in a jump diffusion market. *Journal of Systems Science and Complexity*, 24(2):317–327, 2011.

S. A. Zenios. Asset / Liability Management under uncertainty for fixed-income securities. *Annals of Operations Research*, 59(1):77–97, 1995.

S. A. Zenios and W. T. Ziemba. *Handbook of Asset and Liability Management: Applications and case studies*, volume 2. Elsevier, 2007.

M. Zhang and P. Chen. Mean–Variance Asset–Liability Management under constant elasticity of variance process. *Insurance: Mathematics and Economics*, 70:11–18, 2016.

X. . Zhang and K. . Zhang. Using genetic algorithm to solve a new multi-period stochastic optimization model. *Journal of Computational and Applied Mathematics*, 231(1):114–123, 2009.

Z. Zhou, X. Zeng, H. Xiao, T. Ren, and W. Liu. Multiperiod portfolio optimization for Asset-Liability Management with quadratic transaction costs. *Journal of Industrial and Management Optimization*, 15(3):1493–1515, 2019.

W. T. Ziemba, J. M. Mulvey, H. Moffatt, et al. *Worldwide Asset and Liability Modeling*, volume 10. Cambridge University Press, 1998.

Applications of Biased Randomised Algorithms and Simheuristics to Asset and Liability Management

AUTHOR:
Armando Miguel Nieto Ranero

SUPERVISORS:

DR. ÁNGEL ALEJANDRO JUAN PÉREZ. *Full Professor of Statistics and Operations Research at the Universitat Politècnica de València and Pristini Full Professor at the Universitat Oberta de Catalunya.*

DR. MONTSERRAT GUILLÉN I ESTANY. *Chair Professor of Quantitative Methods for Economics and Business at the University of Barcelona.*

DR. RENATAS KIZYS. *Associate Professor in Finance, Department of Banking and Finance, Southampton Business School, University of Southampton, UK.*

A thesis submitted for the degree of Doctor of Philosophy. 2022