Centro de Informática
UFPE

Julio Cezar Soares Silva

Data-driven multiobjective algorithms: applications in portfolio optimization

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2024

Julio Cezar Soares Silva

Data-driven multiobjective algorithms: applications in portfolio optimization

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional
Orientador: Adiel Teixeira de Almeida Filho

Recife
2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Silva, Julio Cezar Soares.
Data-driven multiobjective algorithms: applications in portfolio optimization / Julio Cezar Soares Silva. - Recife, 2024.
182f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2024.
Orientação: Adiel Teixeira de Almeida Filho.
Inclui referências e apêndices.

1. Generative adversarial network; 2. Interactive multiobjective optimization; 3. Evolutionary Algorithm; 4. Dominance-based rough set approach; 5. Portfolio Optimization; 6. Index Tracking. I. Almeida Filho, Adiel Teixeira de. II. Título.

UFPE-Biblioteca Central

Tese de Doutorado apresentada por **Julio Cezar Soares Silva** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**Data-driven multiobjective algorithms: applications in portfolio optimization**" **Orientador: Adiel Teixeira de Almeida Filho** e **Aprovado** pela Banca Examinadora formada pelos professores

---

Leandro dos Santos Maciel
Faculdade de Economia, Administração e Contabilidade / USP

Francisco de Assis Tenorio de Carvalho
Centro de Informática / UFPE

Adriano Lorena Inacio de Oliveira
Centro de Informática / UFPE

Sérgio Ricardo de Melo Queiroz
Centro de Informática / UFPE

Tiago Pascoal Filomena
Escola de Administração / UFRGS

Prof. **Orientador**: Adiel Teixeira de Almeida Filho
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 03 de Dezembro de 2024.

Prof. Leopoldo Motta Teixeira
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho à minha família

ACKNOWLEDGEMENTS

Agradeço à minha esposa Izabel, aos meus pais Eleno e Josy e à toda minha família pelo apoio incondicional.

Agradeço ao meu orientador Adiel Teixeira de Almeida Filho pelo suporte, oportunidades, conselhos e momentos de descontração proporcionados.

Agradeço ao Centro de Informática da UFPE que disponibilizou uma ótima infraestrutura e também aos professores pelo conhecimento fornecido ao longo deste período.

Agradeço ao Grupo de Pesquisa em Gerenciamento de Riscos, Estudos de Engenharia Financeira e Otimização (GREEFO) pelas diversas contribuições em trabalhos e momentos de aprendizado, em especial Diogo, Naiara, Fábio e Thiago.

Agradeço aos meus colegas do laboratório de pesquisa pelos momentos de descontração, suporte e discussão sobre diversos temas, em especial Gustavo, Eraylson, Flávio, Leandro, Gunnar, Mucio e Walter.

Agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que financiou este estudo parcialmente.

ABSTRACT

Practical portfolio optimization models have been bringing challenges that computational intelligence tools are helping to solve. A class of portfolio optimization problems that have been attracting computational intelligence applications is index tracking. The index tracking problem aims to build a portfolio that replicates the performance of a market index with a subset of assets. Recent applications of deep learning in index tracking have limited application in real environments since the proposed frameworks are not flexible to include more practical constraints and objectives. A novel application of *Generative Adversarial Network* (GAN) which guarantees model extension flexibility is presented. The efficiency of the GAN was evaluated considering the difficulties imposed by the combinatorial nature of the index tracking problem. We also proposed and evaluated two new metaheuristics for the index tracking model with multiple scenarios. The results showed that solving the model using GAN's market simulations produces more stable portfolios when compared to portfolios optimized with real data. Also, the models trained in a specific rebalancing strategy could perform well in other rebalancing strategies. This work also brings discussions about problems related to the application of GANs in this context. Obtaining the optimal Pareto front in a feasible time can be impractical in multiobjective portfolio optimization with practical constraints. Another unsolved problem is the extraction of preference information to find the most preferable nondominated solution. Thus, it is interesting to consider *Evolutionary Multi-criterion approaches* (EMO) to find good fronts within a time constraint guided by preference information. We propose a way to learn a rough approximation of the investor's preference model to guide the EMO search for the single most preferable portfolio and to perform preference-driven portfolio updates. This model can be obtained using *Interactive Multiobjective Optimization using Dominance-based Rough Sets Approach* (IMO-DRSA), which is able to guide evolutionary algorithms using a rule-based model that is refined in each interaction with the investor. The problem is that there is no evidence on how to reduce the number of representative portfolios to minimize *Decision-Maker* (DM) cognitive effort during the interaction, taking the satisfaction of preferences in future distributions of portfolio components returns into account. The results showed that the proposed simulated IMO-DRSA can study the impact of different variables and approaches to reduce the cognitive effort in the performance of the EMO approach to achieve and maintain good preference satisfaction over time.

Keywords: Generative adversarial network. Interactive multiobjective optimization. Evolutionary algorithm. Dominance-based rough set approach. Portfolio optimization. Index tracking.

RESUMO

Modelos práticos de otimização de portfólio vêm trazendo desafios que as ferramentas de inteligência computacional estão ajudando a resolver. Uma classe de problemas de otimização de portfólio que vem atraindo aplicações de inteligência computacional é index tracking. O problema de index tracking visa construir uma carteira que replica o desempenho de um índice de mercado com um subconjunto de ativos. Aplicações recentes de aprendizado profundo em index trackings têm aplicação limitada em ambientes reais, uma vez que os frameworks propostos não são flexíveis para incluir restrições e objetivos mais práticos. Uma nova aplicação de GAN que garante flexibilidade de extensão do modelo é apresentada. A eficiência da GAN foi avaliada considerando as dificuldades trazidas pela natureza combinatória do problema de index tracking. Duas novas metaheurísticas foram avaliadas para o modelo de index tracking com múltiplos cenários. Os resultados mostraram que resolver o modelo usando as simulações de mercado do GAN produz portfólios mais estáveis quando comparados aos portfólios otimizados com dados reais. Além disso, os modelos treinados em uma estratégia de rebalanceamento específica podem ter um bom desempenho em outras estratégias de rebalanceamento. Este trabalho também traz discussões sobre problemas relacionados à aplicação de GANs neste contexto. A obtenção da frente de Pareto ótima em um tempo viável pode ser impraticável na otimização de portfólio multiobjetivo com restrições práticas. Outro problema não resolvido é a extração de informações de preferência para encontrar a solução não dominada mais preferível. Assim, é interessante considerar abordagens multicritério evolucionárias EMO para encontrar boas frentes dentro de uma restrição de tempo guiada por informações de preferência. Propomos uma maneira de aprender uma aproximação grosseira do modelo de preferência do investidor para orientar a busca de EMO pelo portfólio mais preferencial e realizar atualizações de portfólio orientadas por preferências. Este modelo pode ser obtido por meio da Otimização Multiobjetivo Interativa usando a IMO-DRSA, que é capaz de guiar algoritmos evolutivos usando um modelo baseado em regras que é refinado a cada interação com o investidor. O problema é que não há evidências de como reduzir o número de portfólios representativos para minimizar o esforço cognitivo do DM durante a interação, levando em consideração a satisfação das preferências em distribuições futuras dos retornos dos componentes do portfólio. Os resultados mostraram que o IMO-DRSA simulado proposto pode estudar o impacto de diferentes variáveis e abordagens para reduzir o esforço cognitivo no desempenho da abordagem EMO para alcançar e manter uma boa satisfação de preferência ao longo do tempo.

Palavras-chaves: Rede adversária generativa. Otimização multiobjetivo interativa. Algoritmo evolucionário. Abordagem de Rough sets baseada em dominância. Otimização de portfólio. Index tracking.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AHP *Analytical Hierarchy Process*

CCPO *Cardinality Constrained Portfolio Optimization*

CNN *Convolutional Neural Networks*

CVaR *Conditional Value-at-Risk*

DE *Differential Evolution*

DM *Decision-Maker*

DRSA *Dominance-based Rough Sets Approach*

EIT *Enhanced index tracking*

EMO *Evolutionary Multi-criterion approaches*

FGP *Fuzzy Goal Programming*

GA *Genetic Algorithm*

GAN *Generative Adversarial Network*

GRASP *Greedy Randomized Adaptive Search Procedure*

IMO-DRSA *Interactive Multiobjective Optimization using Dominance-based Rough Sets Approach*

MAD *Mean Absolute Deviation*

MBH *Multi-period buy-and-hold strategy*

MCDM *multiple criteria decision making*

MIA *Multiobjective Immune Algorithm*

MINLP *Mixed-Integer non-linear programming*

MLP *Multilayer Perceptron*

MOEA Multiobjective evolutionary algorithms

MSE *Mean Squared Error*

MVO *Mean-Variance Optimization*

NSGA-II *Nondominated sorting genetic algorithm II*

RDM *Real Data Model*

SAA *Sample Average Approximation*

SBH *Single-period buy-and-hold strategy*

SDM *Synthetic Data Model*

SDM-SBDGA-GAN *Scenario-Based Dominance Genetic Algorithm through Generative Adversarial Network*

SDM-SAAGA-GAN *Sample Average Approximation Genetic Algorithm through Generative Adversarial Network*

TEV *Tracking Error Variance*

VaR *Value-at-Risk*

WGAN *Wasserstein Generative Adversarial Network*

LIST OF SYMBOLS

t	Time index
T	total number days in the out-of-sample period
i	Asset index item[N] Number of assets in the universe
w_i	Proportion of capital allocated to i
K	Cardinality of the portfolio
ℓ_0	Norm that counts the number of nonzero elements
ℓ_1	LASSO penalty term
ℓ_2	Ridge penalty term
ℓ_q	Norm used in non-convex optimization for $0 < q < 1$
R_t^p	Portfolio return at time t
R_t^I	Benchmark return at time t
r_{it}	Return of asset i at time t
μ	Vector of mean returns for N assets
Ω	Covariance of returns for N assets
$\mu^T w$	Mean return of the portfolio
Z_i	Binary variable indicating the inclusion of i in the portfolio
ϵ_i	Minimum capital proportion that can be allocated to i
ψ_i	Maximum capital proportion that can be allocated to i
f	Investment horizon in days
b	Lookback period in days
M_f	market simulation, a $N \times f$ matrix
M_b	historical data, a $N \times b$ matrix
M_T	market test data, a $N \times T$ matrix

v	rebalancing frequency
G	Generator
D	Discriminator
DM_{types}	Set of decision-maker types in a simulated interaction process
v_0	Initial number of iterations of a genetic algorithm
y	Interaction index in a simulated interaction process
y_{max}	Max number of interactions in a simulated interaction process
$Rules^y$	Rules obtained at interaction level y
S^y	Set of solutions obtained at interaction level y
DT^y	Data table obtained at interaction level y

CONTENTS

1 INTRODUCTION

Along with many practical extensions to the portfolio selection problem (KOLM; TUETUENCUE; FABOZZI, 2014; KALAYCI; ERTENLICE; AKBAY, 2019; ALMEIDA-FILHO; SILVA; FERREIRA, 2021), past works verified that Markowitz's 1952 framework (MARKOWITZ, 1952) had some limitations (FAMA, 1970; FABOZZI et al., 2007; KOLM; TUETUENCUE; FABOZZI, 2014):

- **Simplified Assumptions:**
  - **Normal Distribution of Returns:** The model assumes that asset returns follow a normal distribution. However, in practice, returns may exhibit skewness and kurtosis, leading to underestimation of tail risks.
  - **Constant Parameters:** The model assumes that expected returns, volatilities, and correlations between assets are constant over time. This assumption is questionable as financial markets are dynamic and subject to abrupt changes.
- **Difficulty in Parameter Estimation:**
  - **Historical Data:** The model relies on historical data to estimate parameters, but the past may not be a good predictor of the future, especially during periods of high volatility.
  - **Estimation Error:** Small errors in parameter estimates can lead to large differences in model results, making portfolio optimization a complex task.
- **Ignores Non-Financial Factors:**
  - **Investor Preferences:** The model does not account for individual investor preferences, such as risk aversion, investment horizon, and legal and regulatory constraints.

First, it is very difficult to predict future returns one or more days before the portfolio's rebalancing day, which impact the consistency and robustness of results produced using this model (FAMA, 1970). Also, the sensitivity of the optimal weight allocation relative to the perturbation of the model inputs brings apprehension concerning the application of the classical MVO in real data (KOLM; TUETUENCUE; FABOZZI, 2014).

In this way, the development of intelligent systems for predicting stock prices (ASADI, 2019; ARAUJO et al., 2019) and more robust estimation techniques, such as shrinkage and Bayesian estimators, and the direct incorporation of uncertainty in the optimization model (FABOZZI et al., 2007), have been proposed to mitigate the construction of counter-intuitive portfolios that produce unnecessary transaction costs over time. Also, other risk

measures and investment strategies were proposed for obtaining consistent returns for more conservative investors in the long run, such as tracking error in passive management (BEASLEY, 2013; BEASLEY; MEADE; CHANG, 2003; GAIVORONSKI; KRYLOV; WIJST, 2005).

Passive fund management emerged from the efficient market hypothesis of Fama (1970), where the best strategy an investor can perform is to follow the market movements, otherwise, his/her portfolio will perform worse than the market. A fund manager can choose between different strategies, which will define how actively or passively he/she will manage the portfolio given the available information. An active strategy consists of trying to outperform the market by picking the winner stocks (ACOSTA-GONZALEZ; ARMAS-HERRERA; FERNANDEZ-RODRIGUEZ, 2015; ROLL, 1992). Passive management is an alternative for conservative investors since it is less risky and usually brings returns close to the benchmark index that is being tracked by the model (RUIZ-TORRUBIANO; SUAREZ, 2009; FABOZZI et al., 2007).

As the size of the stock index grows, full replication implies high transaction costs, therefore harming portfolio returns (FABOZZI et al., 2007; SANT'ANNA et al., 2019; SANT'ANA; CALDEIRA; FILOMENA, 2020). The negative impact on the accumulated returns can be minimized by performing a partial allocation or choosing a subset of the index components. Thus, it is possible to control the portfolio size, by including a cardinality constraint in this type of model (MUTUNGE; HAUGLAND, 2018; WANG; XU; DAI, 2018; SANT'ANNA et al., 2017). Some works involving index tracking research are more concentrated on finding good quality solutions for a given instance of the problem in a practical time.

Although many studies commonly adopt a single performance metric in index tracking optimization, such as tracking error, recent research have been applying computational intelligence techniques for solving portfolio optimization models with multiple objectives (ALMEIDA-FILHO; SILVA; FERREIRA, 2021; FILIPPI; GUASTAROBA; SPERANZA, 2016; SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013; SILVA; FILHO, 2021b; FERREIRA et al., 2018). This is possible by using the mathematical programming framework to develop a multiobjective model and applying a multiobjective optimization algorithm to compute pareto-optimal fronts.

The search for the most preferable solution for a DM in the pareto-optimal frontier can be guided by incorporating preference information in multiobjective evolutionary algorithms (AUGERI; GRECO; NICOLOSI, 2019; KADZIńSKI; TOMCZYK; SłOWIńSKI, 2020). Purshouse et al. (2014) refers to three types of *multiple criteria decision making* (MCDM) and EMO approaches to incorporate DM's preference information: *a priori*, *interactive* and *a posteriori*. In *a priori* methods, the preference of the DM is extracted before the optimization process. In the case of *a posteriori* methods, the preferences are incorporated after the optimization process. In the interactive approach, the algorithm can find the most preferable solution considering the preference information extracted from the DM during the optimization process (Purshouse et al., 2014; AUGERI; GRECO; NICOLOSI, 2019;

KADZIńSKI; TOMCZYK; SłOWIńSKI, 2020).

1.1 MOTIVATION

Although the machine learning and statistical tools developed recently have potential in tracking benchmarks, they have limitations concerning the capacity of extending the portfolio optimization model. Some advantages of the mathematical programming modelling are that it can handle a diversity of real-world objective functions formulations and constraints through single or multiple objective optimization. For example, different types of constraints can be included, such as cardinality, risk exposure, classes of assets, lot size, liquidity, and many others (KALAYCI; ERTENLICE; AKBAY, 2019; LIAGKOURAS; METAXIOTIS, 2018; FERREIRA et al., 2018; SILVA et al., 2021; SILVA; FILHO, 2021b; de Lima Silva; FERREIRA; de Almeida-Filho, 2020; SILVA et al., 2018; ZHAO et al., 2021), which were proposed so that institutional or individual investors could obtain a reduction of costs and adjustment of portfolios to market regulations.

Also, with mathematical programming, it is possible to incorporate uncertainty directly in the model using robust optimization (FERNANDES et al., 2016) or scenario-based optimization (DEB; ZHU; KULKARNI, 2018; MELLO; BAYRAKSAN, 2014), thus making the weights more insensitive to future perturbations of the model parameters, which increases the performance of the portfolio in the out-of-sample period.

Mariani et al. (MARIANI et al., 2019) built and evaluated PAGAN, a GAN for the financial portfolio construction problem, incorporating the uncertainty by using simulations from PAGAN in the mean-variance problem based on the mathematical programming framework. Some of the challenges that aim to use GANs for portfolio optimization is their training process instability and the development of approaches to incorporate simulations in the optimization problem.

The mathematical programming approach offers more flexibility to extend the model and consider multiple criteria in the analysis, which can add more control over future portfolio behaviour (LI; BAO; ZHANG, 2014; LI; BAO, 2014), cost minimization (CHIAM; TAN; MAMUN, 2013; GARCíA; GUIJARRO; MOYA, 2011), and investor's inclination to risk and criteria related to ESG (BRUNI et al., 2015; BILBAO-TEROL; ARENAS-PARRA; CAñAL-FERNáNDEZ, 2012). The main problem is how to find the most preferable portfolio according to the investor's preferences. Some authors propose to rank solutions by performance metrics (LI; BAO, 2014; LI; BAO; ZHANG, 2014), or to prioritize criteria based on the current market state (FILIPPI; GUASTAROBA; SPERANZA, 2016), or models to handle imprecise preference and expectation parameters (BRUNI et al., 2015; BILBAO-TEROL; ARENAS-PARRA; CAñAL-FERNáNDEZ, 2012; WU; TSAI, 2014). At present, it is not possible to obtain a transparent model to understand the portfolio's features most desired by a specific type of investor. Also, there are no approaches considering the decision to update the portfolio based on the learned preferences.

When incorporating the preferences in an interactive manner, the DM progressively understands the problem and adjusts his/her preference information given a computational time budget (LI et al., 2020). Some key challenges of the interactive approach are concerned with limiting the cognitive effort, how frequently a DM should interact (Purshouse et al., 2014; Yu; Jin; Olhofer, 2019; KADZIŃSKI; TOMCZYK; SŁOWIŃSKI, 2020), and learning preferences from a small data set (LI et al., 2020). The complexity involved in the multicriteria portfolio optimization formulations, and computational intractability when considering real-world constraints, make the design and study of interactive multiobjective optimization algorithms applied to this problem a research area with innovative potential (KALAYCI; ERTENLICE; AKBAY, 2019; ALMEIDA-FILHO; SILVA; FERREIRA, 2021).

IMO-DRSA uses rules to represent DM's preferences and guide the evolutionary algorithm search (GRECO; MATARAZZO; SŁOWIŃSKI, 2008). Some of the main advantages of this method in the portfolio selection context are that it does not require the investor to express uncertain parameters, and it uses "*If... Then...*" rules to represent investor preferences, which are transparent and easy to understand (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013). IMO-DRSA only requires that a representative set of non-dominated solutions of the population is presented to the DM in a data table in each interaction.

There are issues that remain unsolved in the application of IMO-DRSA in portfolio optimization. The first is the cognitive effort reduction. Works that solve practical multiobjective portfolio optimization with population-based evolutionary algorithms use hundreds of individuals in the population (ANAGNOSTOPOULOS; MAMANIS, 2010; ANAGNOSTOPOULOS; MAMANIS, 2011; MISHRA; PANDA; MAJHI, 2014; BABAZADEH; ESFAHANIPOUR, 2019; Ferreira et al., 2018), and there is no clue on how to reduce the number of non-dominated portfolios before presenting the data table, such that the DM preferences are satisfied in the out-of-sample investment period. Also it is necessary to develop an investigation on the performance of the evolutionary algorithm in building good portfolios for different investors and maintaining their goodness according to different factors, such as the number of interactions and the data table presentation method.

1.2 RESEARCH QUESTIONS AND HYPOTHESES

RQ1: How can GAN-generated market scenarios improve the robustness of portfolios compared to those constructed using historical data? **H:** Incorporating uncertainty through GAN simulations produces more stable and adaptive portfolios compared to relying solely on historical market data.

RQ2: Are there benefits to using multiobjective metaheuristics in the index tracking model when addressing multiple market scenarios generated by GANs? **H:** Incorporating the tracking performance of a portfolio in each market scenario as an objective improves

the overall mean tracking error across all scenarios, enhancing the robustness of the portfolio.

RQ3: Given that a GAN was trained on a specific rebalancing strategy, is it possible to apply it to other rebalancing strategies without retraining? **H:** The use of GANs for scenario generation, combined with metaheuristics for optimization, improves portfolio performance in out-of-sample periods by reducing sensitivity to model parameter uncertainty, allowing for some degree of flexibility across different rebalancing strategies without the need for retraining.

RQ4: Can the generators learn to produce better market simulations over the training epochs? **H:** The performance of the optimization method is expected to improve as the generators refine their market simulations throughout the training epochs.

RQ5: How does the application of IMO-DRSA influence portfolio optimization in terms of cognitive effort? **H:** The application of IMO-DRSA, paired with rule-based filtering of Pareto frontiers, reduces cognitive effort by presenting a smaller, more relevant set of portfolios that better align with investor preferences during the optimization process.

RQ6: How do the number of interactions influence the satisfaction of the investor? **H:** Increasing the number of iterations and applying effective frontier filtering significantly improve portfolio quality by refining trade-offs between objectives.

RQ7: How do different methods of presenting data tables influence portfolio optimization, considering that different data presentation methods generate distinct constraints? **H:** Various data presentation methods lead to different constraints within the search space, which can either positively or negatively affect the performance of the optimization method.

RQ8: How does investor-specific preferences influence the performance of portfolio optimization methods? **H:** The rules generated by IMO-DRSA during the optimization process can impact the optimization method due to preference constraints of the investor type.

RQ9: How do frontier filters influence portfolio optimization methods? **H:** Well-designed frontier filters strike a balance between maintaining portfolio diversity and satisfying investor preferences, ensuring a stable and effective optimization process throughout investor interactions.

RQ10: How do preference-based update mechanisms impact the adaptability of portfolios over time? **H:** Preference-based updates enable portfolios to dynamically adjust to evolving market conditions while aligning with the specific needs and objectives of the investor.

1.3 OBJECTIVES

1.3.1 General objective

The objective of this work was to propose computational intelligence approaches to deal with recent portfolio optimization formulations that consider practical constraints and investor's objectives.

1.3.2 Specific objectives

- Evaluate if it was possible to build good tracking portfolios through market simulations produced by GANs since they can learn to produce future market trends, given the current market state.
- Study ways to incorporate uncertainty in an index tracking model based on the mathematical programming framework to produce more robust and practical portfolios.
- Develop metaheuristics to solve the multiple scenario index tracking problem.
- Investigate ways to handle GAN models when the cardinality constraint is considered in the portfolio optimization problem.
- Study sources of errors that may produce bad portfolios when using GANs in index tracking.
- Propose a simulated IMO-DRSA to evaluate the robustness of solutions produced from different interaction configurations.
- Propose and evaluate ways to reduce cognitive effort in simulated IMO-DRSA.
- Propose and evaluate ways to present portfolios to investors.
- Evaluate the effect on the investor satisfaction and on the multi-objective optimization algorithm performance, depending on how and with whom the interaction was performed.

1.4 RELEVANCE OF THE RESEARCH PROBLEM

This work expands the analysis performed by Mariani et al. (2019), evaluating it in the index tracking context, considering a simple modification of the GAN proposed by the authors, where we adopt returns instead of prices as the generator's network inputs and outputs. One of the main contributions of this thesis is the comparison of models that use synthetic data generated by GANs against models that use historical data to solve a dynamic index tracking problem using real data from the Brazillian market. In the results,

we observed that this contribution opens doors to the application of index tracking based on mathematical programming models with more practical constraints and other types of tracking objectives since the GAN-based index tracking can provide solutions less sensitive to the deviations of the index in the out-of-sample period.

Another contribution is the development and evaluation of two new metaheuristics, SDM-SAAGA-GAN and SDM-SBDGA-GAN, that incorporate uncertainty in the portfolio optimization problem through simulations performed by the PAGAN model, where it was possible to conclude that the scenario-based dominance metaheuristic performed better than the sample-average metaheuristic. We also consider evaluating the GAN performance for cases for which it was not trained. It was possible to observe that when we reduce the portfolio's rebalancing frequency for which the GAN was trained, it can still maintain a good performance. Thus, it may not be necessary to train new models for the evaluation of some rebalancing strategies.

Also, given the lack of methods to assess the performance of the GAN in this context, other complications emerge due to the combinatorial nature of this problem when cardinality constraints are included. Our approach to handling this type of constraint, which was not considered in the problem that the original PAGAN paper proposed to solve (MARIANI et al., 2019), poses another contribution. The proposed approach simply selects the best PAGAN model, in terms of tracking error, within a subset of training epochs. Thus, we propose to use the best results produced by a PAGAN model in a given training epoch to increase the chances of producing models that generate synthetic data capable of producing robust portfolios that perform better than those that were constructed using historical data. Finally, we present some sources of errors that may emerge and produce bad portfolios when using PAGAN and discuss potential solutions for this problem.

Concerning the interactive optimization area, this study proposes a simulation approach to investigate methods that reduce the number of representative solutions that compose the data table and to deal with distributions of returns over time, as suggested by Salvatore, Matarazzo & Słowiński (2013). The effects of representative solutions in the out-of-sample period were considered because the longer the period that rules are satisfied, the less is the rebalance frequency, which reduces costs associated with portfolio adjustment to the investor's preferences. The simulation approach can also be used to understand how the level of interactivity and investor profile can impact rule satisfaction over time.

Another contribution was a preference learning approach for single and multi-period multiobjective index tracking strategies. A way to exploit the learned rule-based models to perform preference-driven portfolio update strategies was proposed and evaluated. Frontier filters that adapt to the constraints generated from different conditional criteria used in the data table presented for the investor were proposed. New ways to present portfolios to the investors were developed and analyzed considering the DRSA based on stochastic dominance with normality assumptions (GRECO; MATARAZZO; SŁOWIŃSKI, 2008). An

analysis on how the in-sample and out-of-sample performance of an evolutionary algorithm is affected using different frontier filters, data table generation methods, types of investors, and interaction levels was performed.

The innovations and implications of this research for both the field of finance and the development of optimization algorithms are further detailed below.

1.4.1 Contributions to finance

- The thesis presents an innovative application of GANs for the index tracking problem, which aims to replicate the performance of a market index using a subset of assets. GANs are used to simulate different market scenarios, allowing the trained models to offer more robust portfolios, even when compared to real market data. This approach is crucial, as market data is often volatile and unpredictable.

- The application of GANs allows greater flexibility in including practical constraints and objectives that are often encountered in real financial environments. This addresses a significant limitation of traditional deep learning approaches, which often are not adaptable to practical and specific conditions of financial markets.

- An important challenge addressed by the thesis is the reduction of the DM (investor's) cognitive effort during the interaction with the optimization model. A simulation approach was proposed to support the reduction of representative portfolios presented to the decision-maker, maintaining preference satisfaction and improving the decision-making process's efficiency.

- The thesis proposes a way to learn an approximation of the investor's preference model, which can help understand which portfolio attributes are more important to a given investor.

1.4.2 Contributions to optimization algorithms

- The thesis proposes two new metaheuristics to solve index tracking problems in multiple scenarios, addressing the combinatorial complexity and the need for robust solutions. These metaheuristics were evaluated in different rebalancing strategies.

- This work demonstrates different ways for optimization algorithms to utilize simulations generated by GANs or other generative artificial intelligence.

- A significant contribution is the application of the Interactive Multiobjective Optimization using IMO-DRSA. This approach allows the incorporation of investor preferences into the optimization process, guiding the metaheuristic to find the most preferable solution.

- The thesis investigates the impact of different variables and approaches on the satisfaction of investor preferences over time. This analysis is essential to understand how various factors influence the performance of optimization algorithms and investor satisfaction and improving the decision-making process's efficiency.

1.5 ORGANIZATION OF THE THESIS

Figure 1 presents this thesis structure. As shown in Figure 1, Chapters 3, 4, and 5 are interconnected through their shared focus on optimizing financial models using multiobjective algorithms. Chapter 3 employs GANs to generate synthetic market data, which is then used by an algorithm that treats tracking error performance in simulations as distinct objectives. Chapter 4 introduces a simulated multiobjective approach for portfolio selection within mean-variance optimization, incorporating various interaction models and investor preferences. Chapter 5 builds upon this framework by applying it to index tracking, where multiobjective algorithms guide preference-based portfolio rebalancing, considering different interaction configurations.

Chapter 2: : Presents theoretic concepts which the author considered necessary as basic knowledge to start developing this work. This chapter begins with a brief presentation of classical and Wasserstein GANs, and convolutional layers. Then, multiobjective modelling, optimization and the NSGA-II algorithm are described. Finally, fundamental concepts of DRSA and interactive multiobjective optimization with DRSA are presented.

Chapter 3:: Chapter 3 brings a novel application of GANs for index tracking. The efficiency of the GAN was evaluated considering the difficulties imposed by the inclusion of cardinality constraints, which makes the index tracking problem NP-hard. The GANs were evaluated based on the performance of portfolios produced by them and compare to the performance of portfolios produced using real historical data. We also proposed and evaluated two new metaheuristics for the index tracking model with simulated market scenarios and evaluated them using real data from the Brazilian market.

Chapter 4:: In Chapter 4, a simulated IMO-DRSA was proposed to analyze and compare methods that select a small and robust sample of representative solutions to compose data tables. This approach was evaluated using two types of simulated investors in a cardinality-constrained mean-variance optimization problem.

Chapter 5:: An extended analysis of the simulated IMO-DRSA approach is presented in Chapter 5, considering its application in the index tracking context. The analysis involved experiments to understand how the IMO-DRSA could guide the EMO approach to achieve and maintain good preference satisfaction over time by presenting a small number of portfolios to the investor during interactions. This analysis considered different frontier filters and data table presentation methods. The ability of IMO-DRSA to detect important criteria for the investors was investigated and a preference-based rebalancing strategy was proposed.

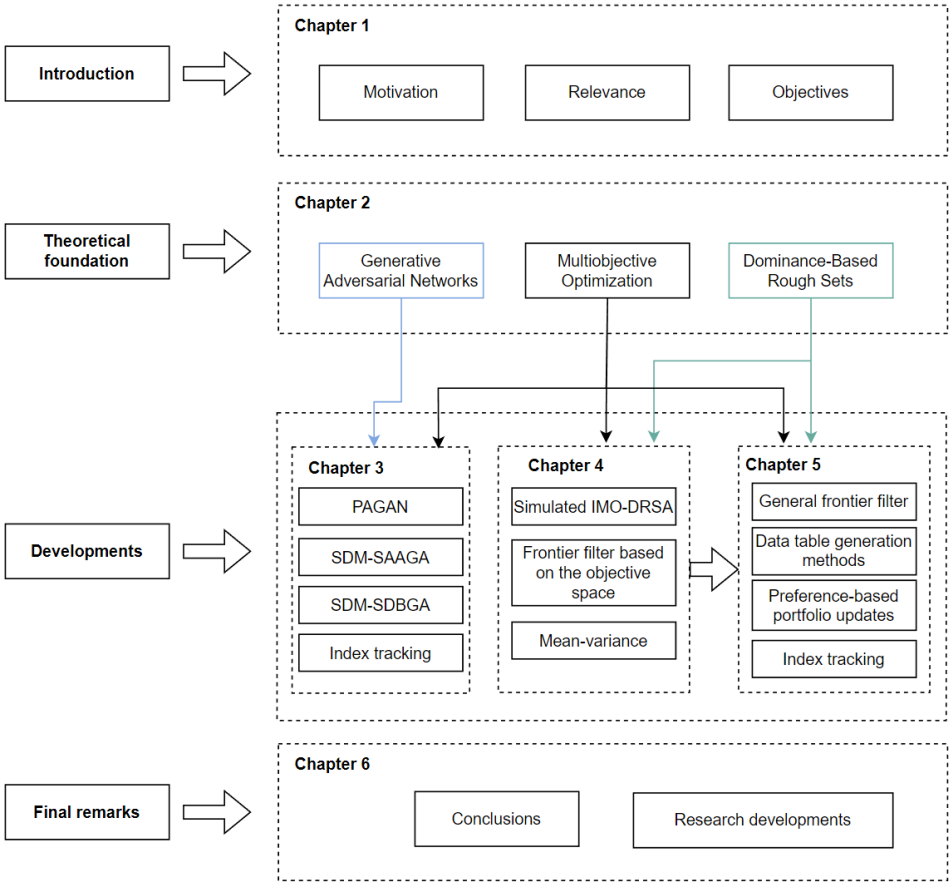


Figure 1 – Thesis structure.

Chapter 6:: In the last chapter, the conclusions and research developments during the doctorate program were presented.

The main characteristic of this thesis is the use of multiobjective algorithms to optimize different types of models, wether they are stochastic models that consume synthetic data from GANs or multiobjective models tha need algorithms that are guided by investor's preferences.

2 THEORETICAL FOUNDATION

2.1 GENERATIVE ADVERSARIAL NEURAL NETWORKS

A GAN is a type of neural network architecture inspired by Game Theory concepts, proposed by Goodfellow et al. (2014). A zero-sum game is played by two neural networks competing against each other during the training process. The generator network receives noise and outputs fake data with the same characteristics as real data. The discriminator network takes real data and fake data from the generator as input and tries to distinguish which input data is real and which is fake. Thus, along the training process, one network tries to outperform the other by minimizing the performance of the adversary and maximizing its performance. When the training has finished, we expect that the generator is capable of generating new data that is not distinguishable from real data.

The GAN is presented in Figure 2. Formally, the game between the generator G and the discriminator D is the minimax objective:

$$\min_{\theta}\max_{w} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[log(D_w(x))] + \mathop{\mathbb{E}}_{x'\sim\mathbb{P}_\theta}[log(1 - D_w(x'))] \tag{2.1}$$

where θ and w are the parameters for G and D, respectively, $\mathbb{P}_r$ is the real data distribution, $\mathbb{P}_\theta$ is the generative model distribution, $x' = G_\theta(z)$, and the Generator's input noise is given by $z \sim p(z)$. The input noise z can be sampled from a simple noise distribution, i.e. uniform or Gaussian distributions. Originally, the iterative process alternated k optimizing steps for D and one optimizing step for G to avoid overfitting D in the early training steps.

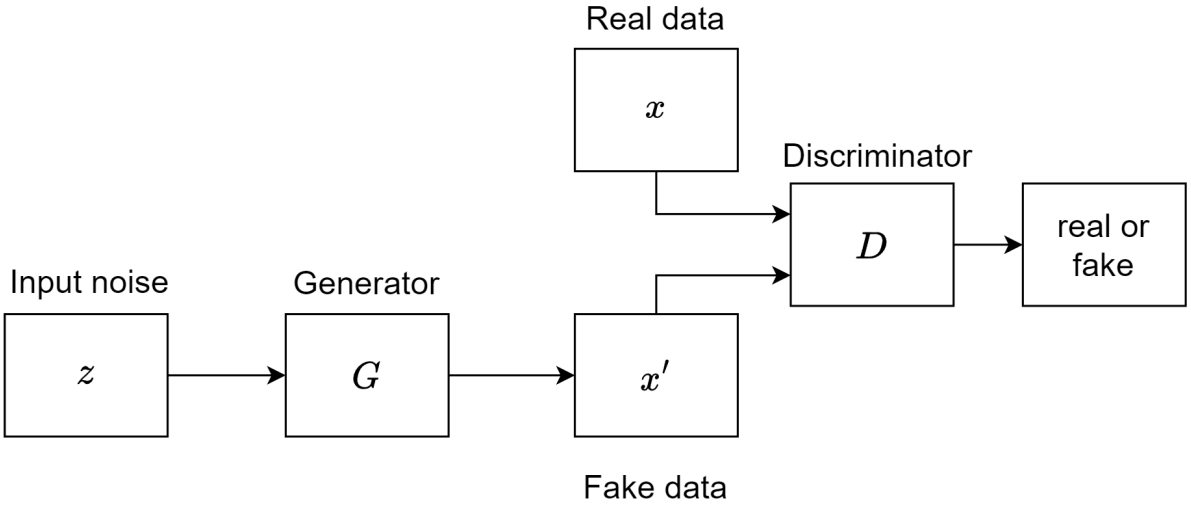


Figure 2 – Generative Adversarial Network

2.1.0.1 WGAN

The Jensen-Shannon divergence measures the distance between two probability distributions. The JS divergence between two probability distributions p and q is defined in equation 2.2.

$$JS(p||q) = KL(p||p_m) + KL(q||p_m) \tag{2.2}$$

where $p_m = \frac{p+q}{2}$ is the "average" distribution, and $KL(p||q) = \int_x p(x) log \frac{p(x)}{q(x)} dx$ is the Kullback-Leibler divergence. The minimization of KL divergence is a classic loss function to generative models, requiring only sample data and no assumptions about $\mathbb{P}_r$, and it is equivalent to maximizing likelihood (ARJOVSKY; CHINTALA; BOTTOU, 2017). It is interesting to note that the results obtained when minimizing $KL(\mathbb{P}_r||\mathbb{P}_\theta)$ are different from those obtained when minimizing $KL(\mathbb{P}_\theta||\mathbb{P}_r)$ since the KL divergence is not symmetrical (ARJOVSKY; BOTTOU, 2017).

The classic min-max loss function of GANs has been shown to minimize the JS divergence when we minimize 2.1 in function of θ when the discriminator is optimal, as shown in equation 2.3 D (ARJOVSKY; BOTTOU, 2017). This loss function is a symmetric middle ground between optimizing $KL(\mathbb{P}_r||\mathbb{P}_\theta)$ and $KL(\mathbb{P}_\theta||\mathbb{P}_r)$.

$$L(D^*, \theta) = 2JS(\mathbb{P}_r||\mathbb{P}_\theta) - 2log2 \tag{2.3}$$

Although this would lead us to train D near to optimality and then optimize θ to better approximate the JS divergence, other problems emerge with this approach, such as vanishing gradients. When the discriminator is trained up to optimality, the JS divergence is locally saturated and the generator's gradients rapidly vanish along the training process (ARJOVSKY; BOTTOU, 2017). Another common failure of GANs, named mode collapse, occurs when the generator can fool the discriminator by learning a small region with very low variety during training, producing always the same outputs. The vanishing gradients and mode collapse problems can be minimized when using Wasserstein distance as the loss function (ARJOVSKY; CHINTALA; BOTTOU, 2017). The tractable computation is performed with the Wasserstein distance under 1-Lipschitz condition, shown in Equation 2.4, where here we have $(\mathbb{P}_r||\mathbb{P}_\theta)$ as the supremum (lowest upper bound) overall 1-Lipschitz functions $f : X \to R$.

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r}[f(x)] - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta}[f(x')] \tag{2.4}$$

where $f : X \to R$ is K-Lipschitz continuous when, given 2 metric spaces (X, d_X) and (Y, d_Y), $d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$, $\forall x_1, x_2 \in X$. Thus, if there is a family of

parameterised functions f_w, where $w \in W$, that are K-Lipschitz continuous, then Equation 2.5 presents the maximization of the Wasserstein distance multiplied by a constant.

$$W(\mathbb{P}_r, \mathbb{P}_\theta) \propto \max_{w \in W} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [f_w(x)] - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta} [f_w(x')] \tag{2.5}$$

Due to this characteristic of the Wasserstein distance now have the optimization of a critic f_w that estimates the Wasserstein distance between $\mathbb{P}_r$ and $\mathbb{P}_\theta$ instead of a discriminator (discriminates real and fake samples). The critic loss is presented in Equation 2.6.

$$L_{critic} = \max_{w \in W} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [f_w(x)] - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta} [f_w(x')] \tag{2.6}$$

The generator loss is shown in Equation 2.7.

$$L_{gen} = \min_{\theta} \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [f_w(x)] - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta} [f_w(x')] = \min_{\theta} - \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta} [f_w(x')] \tag{2.7}$$

Arjovsky, Chintala & Bottou (2017) adopted weight clipping to enforce the 1-Lipschitz constraint. This is done by restricting the weights w to a small range (i.e. [-1e-2, 1e-2]). A drawback of the WGAN algorithm is that the model performance is very sensitive to the clipping parameter. Gulrajani et al. (2017) proposed WGAN-GP, which brings a better way to enforce the 1-Lipschitz constraint. They added a gradient penalty in the loss term that forces the L2 norm of the critic gradients to be unitary, as shown in Equation 2.8.

$$L = \mathop{\mathbb{E}}_{x' \sim \mathbb{P}_\theta} [f_w(x')] - \mathop{\mathbb{E}}_{x \sim \mathbb{P}_r} [f_w(x)] + \lambda \mathop{\mathbb{E}}_{x'' \sim \mathbb{P}_{x''}} [(||\nabla_{x''} f_w(x'')||_2 - 1)^2] \tag{2.8}$$

where $x'' = tx + (1-t)x'$ sampled from a straight line between x (current batch) and x', with t uniformly sampled between 0 and 1 ($t \sim U[0, 1]$).

2.1.1 Convolutional layers

Convolutional Neural Networks (CNN) are widely used to solve Computer Vision tasks, such as image classification (RADFORD; METZ; CHINTALA, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016). The basic building blocks of CNNs are convolutions, pooling (down-sampling) operators, activation functions, and fully-connected layers, which are essentially similar to hidden layers of a *Multilayer Perceptron* (MLP). CNN contain at least one convolution layer, which is composed of a set of filters applied to the entire input vector (GOODFELLOW; BENGIO; COURVILLE, 2016).

Regular neural networks receive a single vector as input and process this input through a set of hidden layers. Each hidden layer is constituted by a set of neurons, and each

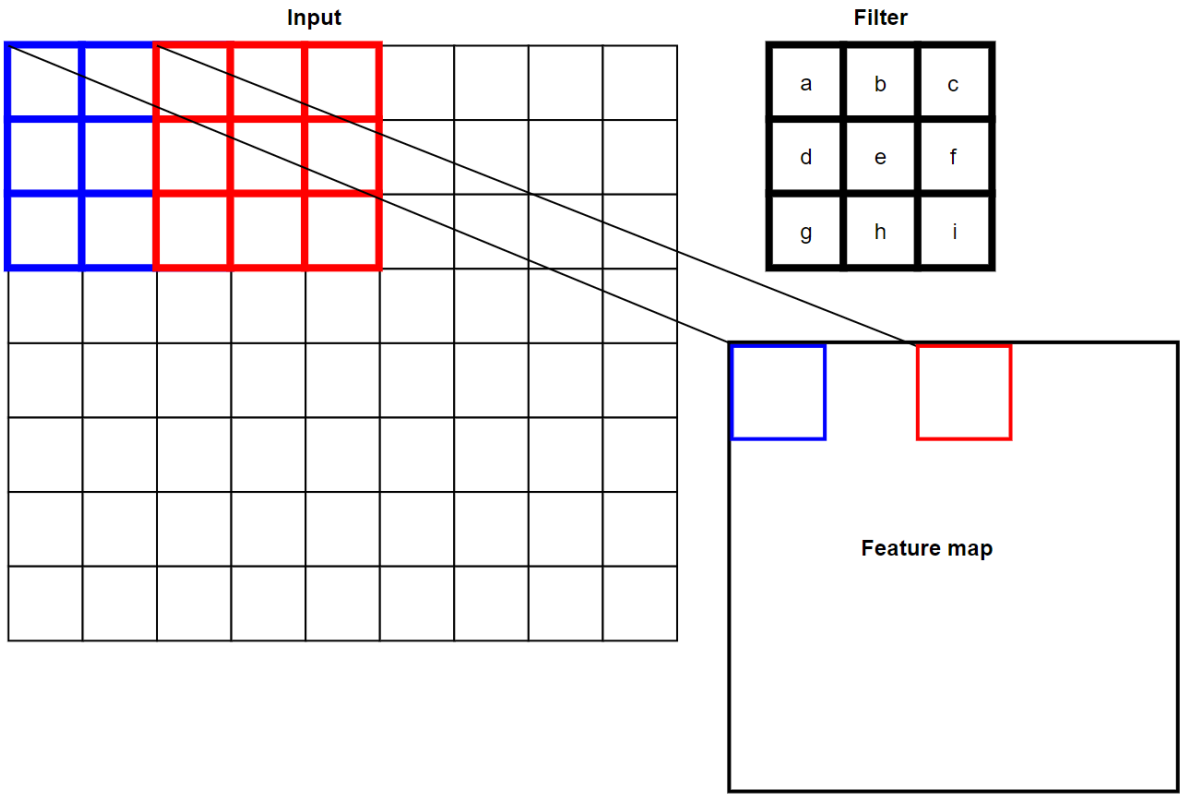


Figure 3 – Example of a filter applied to a 2D input to generate a feature map

neuron is fully connected to all neurons in the previous layer. Thus, the neurons in a single layer function completely independently and do not share any connections (no parameter sharing). The last fully-connected layer is called the "output layer" and in the context of a classification problem, it is associated with the computation of the class scores.

CNNs use filters to process a given input, such as a 2D image. Each filter is a matrix $k \times k$ of weights to be learned during the training process. The convolution operation consists of a dot product, which is an element-wise multiplication, and it is applied between the filter-sized region of the input and the filter, which is then summed, resulting in a single value. A sequence of convolutional operations generates a feature map, as shown in Figure 3, where a 3x3 filter is successively applied to a 2D input to obtain a 2D feature map.

A typical convolutional layer consists of three stages (GOODFELLOW; BENGIO; COURVILLE, 2016). In the first phase, the filter is applied to produce several convolutions to generate several linear activations. Initially, the filter is applied to the upper left region of the input, then the dot product is performed, generating the entry (1,1) of the feature map. Then, the filter is moved to the right according to the **stride** value before another dot product is performed. If the stride value is equal to (1,1), then the filter moves one column (or one pixel in the case of 2D images) to the right. Also, the filter moves one row to the bottom every time the final column is processed. The feature map size is controlled by the stride value. The larger the stride, the smaller the feature map.

When the data is associated with 2D images, each filter produces a linear combination of all pixel values in a neighbourhood defined by the size of the filter. A grayscale image input has $w \times h$ dimensions and when considering a colour image, the input has $w \times h \times c$ dimensions, where c is the number of channels (or depth of the input), in this case, $c = 3$, which are the RGB channels. It is also possible to consider 1D convolutions, which can be applied in time series data, where the channels are the number of time series to be processed. One advantage of convolutions is that we store much fewer parameters than MLP models since the filter is much smaller than the input (i.e. 2D image with thousands of pixels). The learned filter can detect small, meaningful features on an image, such as its edges.

In the second phase, we can apply a nonlinear activation function, such as ReLU, to the feature map. This process is similar to the application of nonlinearities in the outputs of a fully connected layer. In the third phase, we use a **pooling** function to reduce the dimensionality of the feature map to increase the computational efficiency. In the pooling layer, a summary statistic, generally the mean or max, is obtained from some regions (i.e. rectangular regions) of the feature map to reduce the number of the parameters of the network and to create spatial invariance. The spatial invariance is useful if we give more importance to the detection of a feature (such as a face or a tree), independent of where this feature is located (GOODFELLOW; BENGIO; COURVILLE, 2016; ISLAM et al., 2021).

Padding can be used to add extra "borders" to the input. For example, in computer vision, it is possible to use padding to reduce the image shrinking every time a convolution is applied, by adding extra pixels with a value equal to zero (ISLAM et al., 2021). Moreover, it ensures that all pixels are used equally frequently. This same concept can be used to add zero-valued data to other types of inputs to reduce their rapid reduction with the sequential application of convolutions and to ensure the equal use of each part of the input.

2.2 MULTIOBJECTIVE OPTIMIZATION

Multiobjective models are associated with finding the best compromise solution for M conflicting objectives. Considering a solution vector $x = \{x_1, ..., x_N\}$, the general form of the multiobjective optimization problem with J inequalities and K equalities is stated as follows (DEB, 2001):

$$\min_x / \max \quad f_m(x), \text{ m = 1,2,...,M} \tag{2.9}$$

$$\text{subject to} \quad a_j(x) \geq 0,\ i \in 1, ..., J \tag{2.10}$$

$$b_k(x) = 0,\ j \in 1, ..., K \tag{2.11}$$

$$l_i \leq x \leq u_i,\ i \in 1, ..., N \tag{2.12}$$

$$\tag{2.13}$$

where l_i and u_i are the lower and upper boundaries for variable i. A solution x that satisfies all the inequalities, equalities and boundaries is a feasible solution, otherwise, it is an infeasible solution. There are M objective functions $f(x) = \{f_1(x), ..., f_M(x)\}$, where each objective can be maximized or minimized. One of the main differences between single and multiobjective optimization is that the last contains an objective space, in addition to the decision variable space. Thus, each solution x is associated with a point $z = f(x) = \{z_1, ..., z_M\}$ in the objective space.

Considering a bi-objective problem, shown in Figure 4, where we want to minimize both f_1 and f_2, we have two non-overlapping regions in the feasible search space: Pareto-optimal set and non-Pareto-optimal set. Different from single optimization problems, multiobjective problems have multiple optimal solutions, called Pareto-optimal solutions, which constitute the Pareto-optimal front. This front appears when the problem is associated with conflicting objectives. We cannot say that a Pareto-optimal solution $x^{(1)}$ is better than a Pareto-optimal solution $x^{(2)}$ concerning both objectives since choosing between two Pareto-optimal solutions will decrease the performance in one of the objectives and increase the performance in another objective. Thus, we can say these two solutions are non-dominated.

To choose between two non-dominated solutions, it is necessary to obtain more information to make a more biased search. However, in the absence of any information about which objective is preferable, all non-dominated solutions are equally important. Thus, there are two goals in multiobjective optimization: to find a set of solutions as close as possible to the Pareto-optimal front and find a set of solutions as diverse as possible.

Multiobjective optimization algorithms use the concept of dominance to compare two solutions. Assuming that there are M objectives to be minimized, a solution $x^{(1)}$ dominates $x^{(2)}$ if 1 and 2 are true:

1. $x^{(1)}$ is at least as good as $x^{(2)}$ in all objectives: $f_m(x^{(1)}) \leq f_m(x^{(2)})$ for all $m = 1, ..., M$
2. $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective: $f_m(x^{(1)}) < f_m(x^{(2)})$ for at least one $m \in \{1, ..., M\}$

One can generalize this definition to a mix of minimization and maximization objectives by changing '$\leq$' to '$\geq$', and '$<$' to '$>$' when necessary. Considering a solution set P, the non-dominated set of solutions P' contains solutions that are not dominated by any member of the set P. When the set P is the entire search space, the set P' is the Pareto-optimal set (DEB, 2001).

2.2.1 NSGA-II

The problem of classical optimization methods is that they convert the multiobjective problem into a single objective problem, which computes a Pareto-optimal solution in a unique simulation run, and thus requires running multiple simulations to find multiple

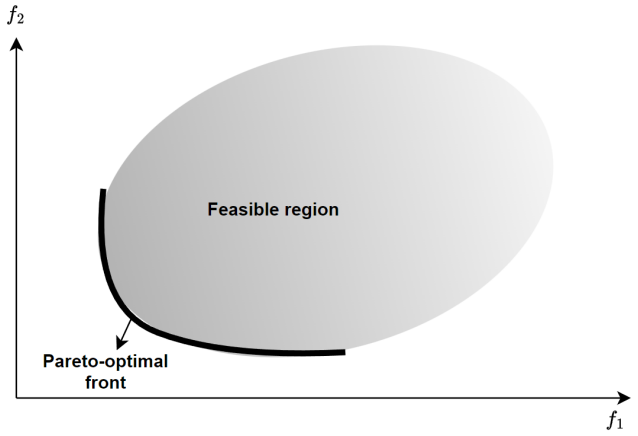


Figure 4 – Illustration of the feasible region and Pareto front of a bi-objective problem

Pareto-optimal solutions. Multiobjective evolutionary algorithms (MOEA) have been developed to find multiple and diverse Pareto-optimal solutions in a single simulation run. Another advantage of MOEAs is that they eliminate the need for weight, target vectors, and other parameters that are needed in classical optimization to transform multiobjective problems into single objective problems (Deb et al., 2002; DEB, 2001).

NSGA-II was proposed by Deb et al. (2002) to reduce the computational complexity when performing non-dominated sorting, ensure diversity using a nonparametric diversity-preservation mechanism, and use elitism to preserve good solutions over the generations to speed up the genetic algorithm's convergence. The NSGA-II procedure is shown in Algorithm 1 and its complexity is given by $O(n_I^2 M)$.

Algorithm 1: NSGA-II

Input: n_I, n_G, p_c, p_m
Output: P
1: $g \leftarrow 1$
2: $R_g \leftarrow getInitialPop(n_I)$
3: **while** $g \leq n_G$ **do**
4: $\quad fit \leftarrow getFitness(R_g)$
5: $\quad F \leftarrow sort(fit)$
6: $\quad P \leftarrow getCurrentPop(F, n_I)$
7: $\quad matingPool \leftarrow selection(P, fit)$
8: $\quad Q \leftarrow crossover(matingPool, R_g, p_c)$
9: $\quad Q \leftarrow mutation(Q, n_I, p_m)$
10: $\quad R_g \leftarrow concat(P, Q)$
11: **end while**

The generation counter g is initialized in step 1. In step 2, a population R_g with size n_I is generated. Steps 4-10 define the evolutionary process associated with NSGA-II. The fitness of the individuals of R_g in each objective is evaluated in step 4. In step 5 the non-dominated sorting procedure is applied to classify the entire population R_g. This procedure first encounters F_1, which is the front that contains individuals non-dominated

by any individual from R_g, and has $rank = 1$. Then, another front F_2 is computed, which contains individuals non-dominated by any individual from $R_g - F_1$, and has $rank = 2$. Thus, individuals from R_g are ranked until all the population is fully described by d subsets, where $|F_1| + |F_2| + ... + |F_D| = |R_g|$. The smaller the rank the better the solution.

The new parent population is obtained in step 6 by using the rank and the crowding distance information. The crowding distance is used to maintain the diversity among the population and is defined as the average side-length of a cuboid formed by the nearest neighbours of $s \in F_d$, where $d = 1, 2, ...D$. It is calculated as follows (Deb et al., 2002):

1. The number of solutions in a front F_d is defined as $l = |F|$. For each $s \in F_d$, assign $d_s = 0$

2. For each objective function $m = 1, 2, ..., M$, sort the set in the worst order of f_m. Thus, we can obtain a sorted indexes vector for each objective I^m.

3. For $m = 1, 2, ..., M$, assign a large distance to the boundary solutions, which gives $d_{I_1^m} = d_{I_l^m} = \infty$. For all the other solutions $j = 2, ..., l - 1$, assign $d_{I_j^m} = d_{I_j^m} + \frac{f_m^{I_{j+1}^m} - f_m^{I_{j-1}^m}}{f_m^{max} - f_m^{min}}$. The parameters f_m^{max} and f_m^{min} can be set as the population maximum and minimum values of the m-th objective function.

The bigger the value of the crowding distance the better the solution. The parent population is generated in step 6 by adding individuals with the best ranks and then using the crowding distance to prioritize individuals belonging to the same front. The solutions of a front $d = 1, ..., D$ are added to the parent population P, while $n_I > |P| + |F_d|$. When $n_I < |P| + |F_d|$, the solutions of the $d - th$ front are included until the total number of individuals n_I is fulfilled. In this situation, individuals from F_d are prioritized according to their crowding distance. Thus, individuals with the worst crowding distance values will be discarded.

In step 7 the mating pool is built by using the binary tournament selection operator. In Deb et al. (2002), this operator was modified to consider the crowding distance operator, so that individuals with the same front rank could be prioritized. After that, in steps 8 and 9, the offspring population is obtained using crossover and mutation operators with probabilities p_c and p_m respectively. Finally, NSGA-II uses elitism, which requires the concatenation of the parent and offspring populations (step 10), and thus preserves good parents along the evolutionary process.

2.3 DRSA

Ordinal classification or sorting refers to allocating a set of alternatives in pre-defined classes that are ordered in terms of preferences (ROY, 1996). Several sorting methods have been used in the last decades, and new proposals and studies continue to emerge in the

literature (FERNANDEZ; FIGUEIRA; NAVARRO, 2021). Ordered classes can be characterized by rules when adopting the DRSA method (GRECO; MATARAZZO; SLOWINSKI, 2001), which is an evolution of the classical rough set approach proposed by Pawlak (PAWLAK, 1982). Thus, in DRSA, the decision-maker's preferences can be represented by a set of "$If..., then...$" rules R.

Let DT be a data table, which is a tuple $< U, Q, V, f >$, where U is a finite set of objects or decision examples, $Q = C \cup D$, where C is the finite set of the conditional attributes, and D is a decision attribute (here we consider a total of $|D|$ classes or groups), V_q is the value set of attribute q, $V = \cup_{q \in Q} V_q$, and the information function $f(x, q)$ is defined as $f : U \times Q \rightarrow V$, such that $f(x, q) \in V_q$ for each $q \in Q$, $x \in U$ (GRECO; MATARAZZO; SLOWINSKI, 2001; SLOWINSKI; GRECO; MATARAZZO, 2012).

It is possible to identify inconsistencies in classifications in U exploring dominance information. The information granules $D_p^+(x) = \{y \in U \colon yD_px\}$ and $D_p^-(x) = \{y \in U \colon xD_py\}$ are outranking binary relations extracted from the data table, where the first information granule contains all the objects that dominate x with respect to a set $P \subseteq C$ of attributes and the second contains all the objects that are dominated by x with respect to a set $P \subseteq C$ of attributes (GRECO; MATARAZZO; SLOWINSKI, 2001; SLOWINSKI; GRECO; MATARAZZO, 2012).

After computing the information granules, two partitions of U, $Cl_t^{\geq} = \cup_{s \geq t} Cl_s$ and $Cl_t^{\leq} = \cup_{s \leq t} Cl_s$, can be used to detect inconsistencies. Considering that the preference direction grows from class 1 to class $|D|$, $Cl_t^{\geq}$ is an upward union of classes, containing elements that belong to a class t or better, and $Cl_t^{\leq}$ is a downward union of classes, which contains elements belonging to a class t or worse. Inconsistencies occur when $x \in Cl_t^{\geq}$ and $D_p^+(x) \cap Cl_{t-1}^{\leq} \neq \emptyset$ or $x \notin Cl_t^{\geq}$ and $D_p^-(x) \cap Cl_t^{\geq} \neq \emptyset$. Given that there may be ambiguity relative to the inclusion of x in some classes, we need to consider a way of representing this ambiguity. To represent certain knowledge about an object x, lower approximations are used, $P_*(Cl_t^{\geq}) = \{x \in U \colon D_p^+(x) \subseteq Cl_t^{\geq}\}$ and $P_*(Cl_t^{\leq}) = \{x \in U \colon D_p^-(x) \subseteq Cl_t^{\leq}\}$. To represent possible knowledge, upper approximations are used, $P^*(Cl_t^{\geq}) = \{x \in U \colon D_p^-(x) \cap Cl_t^{\geq} \neq \emptyset\}$ and $P^*(Cl_t^{\leq}) = \{x \in U \colon D_p^+(x) \cap Cl_t^{\leq} \neq \emptyset\}$. Boundaries represent the ambiguous knowledge concerning x, $Bn_p(Cl_t^{\geq}) = P^*(Cl_t^{\geq}) - P_*(Cl_t^{\geq})$ and $Bn_p(Cl_t^{\leq}) = P^*(Cl_t^{\leq}) - P_*(Cl_t^{\leq})$(GRECO; MATARAZZO; SLOWINSKI, 2001; SLOWINSKI; GRECO; MATARAZZO, 2012).

It is possible to transform the patterns of the classification of an object into a language that the DM can understand more easily by using a rule induction algorithm, which will contribute to the decision-making process. At the end of the induction process, rules of the type "If ..., Then" will be obtained, where the first part refers to the condition criteria and thus describe decision examples in terms of elements of C, and the second part is the conclusion about the object, which describes decision examples in terms of elements of D (GRECO; MATARAZZO; SLOWINSKI, 2001). The general structure of a decision rule is shown below.

IF $a_{u,1}$ satisfies $h_{r,1}$ and $a_{u,2}$ satisfies $h_{r,2}$ and ... and $a_{u,c}$ satisfies $h_{r,c}$; **THEN** u belongs to d_r.

where $u \in U$, and $h_{r,c}$ is the c-th criteria threshold , where $c \leq |C|$, for rule $r \in R$, which defines the conditional part of the rule, and $d_r \in D$, defines the decision part for rule r. Certain rules will learn thresholds from objects that are associated with certain knowledge $P_*(Cl_t^{\geq})$ and $P_*(Cl_t^{\leq})$. Possible rules will learn thresholds from objects that are associated with possible knowledge $P^*(Cl_t^{\geq})$ and $P^*(Cl_t^{\leq})$. To understand the types of rules in more detail, data reduction, preference discovery, rule induction algorithms, and classification issues, please refer to (GRECO; MATARAZZO; SLOWINSKI, 2001; SLOWINSKI; GRECO; MATARAZZO, 2012; BŁASZCZYŃSKI; GRECO; SŁOWIŃSKI, 2007; BŁASZCZYŃSKI; SŁOWIŃSKI; SZELĄG, 2011; BŁASZCZYŃSKI; GRECO; SŁOWIŃSKI, 2012).

In this study, the software jMAF (BŁASZCZYŃSKI et al., 2013) was used to induce rules. This software induces two types of rules: certain and possible. The set of rules induced by jMAF is minimal and complete. Therefore, the elimination of any rule belonging to the induced set implies that some consistent decision examples from the universe U will not be reallocated to their original classes, or some inconsistent decision examples will not be allocated to clusters of classes that reflect this inconsistency.

2.3.1 IMO-DRSA

Rules can be used as preferences and be exploited in an interactive multiobjective optimization approach. An interactive procedure is composed of two alternating phases: the computation phase and the dialogue phase. In the computation phase, a sample of feasible solutions is calculated and presented to the DM. Then, in the dialogue phase, the DM has two options: to discriminate between good and bad solutions or to choose the single most preferable solution. When a single solution is chosen, the procedure stops. Otherwise, a preference model is extracted from the DM evaluation of the proposed solutions.

The preference model can be exploited to a new sample of feasible solutions in the next computation phase, to better fit the DM's preferences. The decision rules stemming from DRSA has some advantages. The first is that in IMO procedures the preference model appearing between the dialogue stage and the computation phase is implicit, whereas the decision rules can be explicitly shown to the DM for his/her approval. Also, the preference model should be easily understandable, which is also fulfilled by decision rules. The IMO-DRSA procedure, proposed in Greco, Matarazzo & Słowiński (2008), is shown in 5.

In the first step, an initial set of feasible solutions is generated by optimizing the problem. Then, a representative set of non-dominated solutions is presented to the DM. If the DM is satisfied with a unique solution, then this is the compromise solution and the

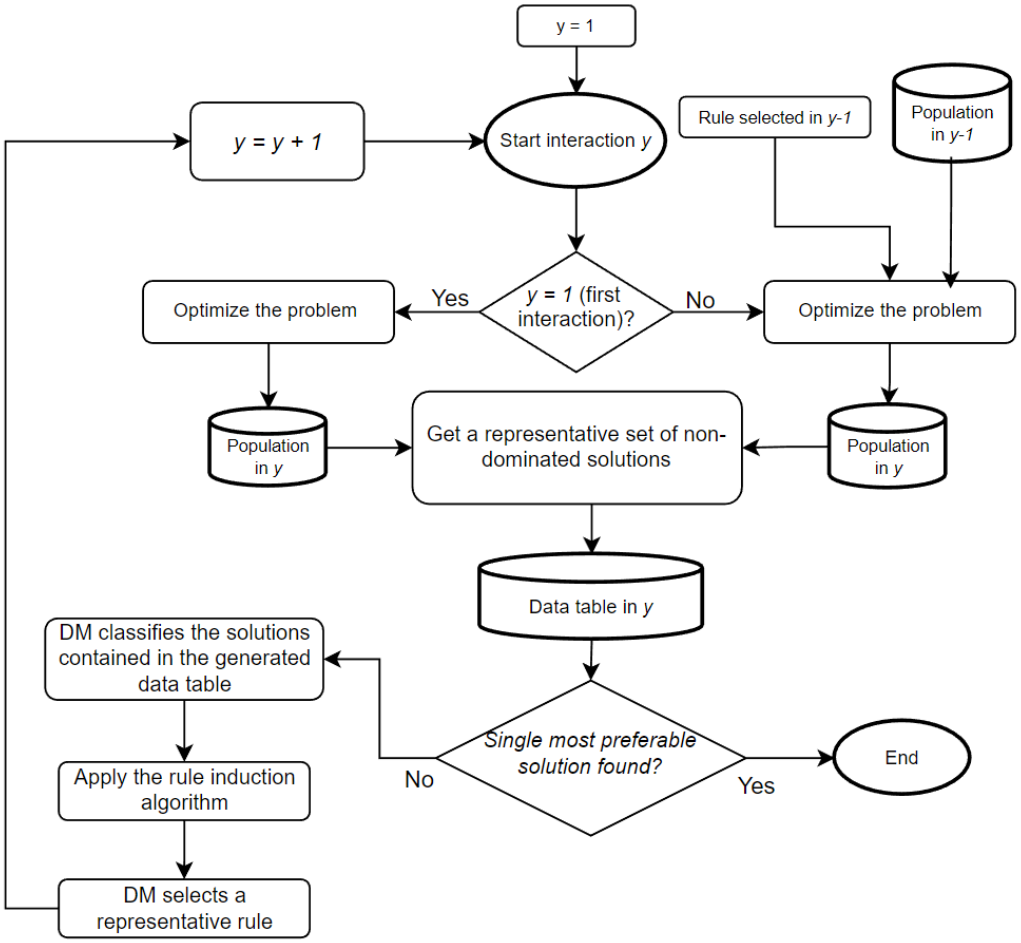


Figure 5 – IMO-DRSA procedure

procedure stops. Otherwise, the DM is asked to indicate a subset of 'good' solutions in the sample. A data table DT is constructed with the information provided by the DM concerning 'good' solutions, and the remaining solutions are classified as 'other'. Next, DRSA is applied to DT and a set of rules R is induced. Then, R is presented to the DM and he/she is asked to select the rule he/she judges to be the most important.

We start the procedure again. Now, we can use the preference information to find more interesting solutions for the DM. The rule selected by the DM is used as a constraint to the problem to guide the optimization algorithm to find a region of feasible solutions that are more interesting for the DM. Then we start the dialogue phase again until the single most preferable solution is found.

2.4 PORTFOLIO OPTIMIZATION

Traditionally, the problem of portfolio selection, which possess a central role in financial management, involves the computation of proportions of capital that must be allocated in a set of available assets with the objective of maximizing return and minimizing risk of an investment portfolio. For this problem, the rule of expected return-variance or E-V rule

provides efficient and, often, diversified portfolios (MARKOWITZ, 1952). Given that p_{it} is the price of asset i a time t, where $t \in \{1, ..., T\}$, the return of asset i after d periods is defined as $r_{it} = \frac{p_{it} - p_{i(t-d)}}{p_{i(t-d)}}$.

The log-return formula can substitute the simple return formula in the context of the Markowitz framework because of the desirable statistical properties of log-returns. Unlike simple returns, log-returns are time-additive, which simplifies the calculation of multi-period returns. Additionally, log-returns are more robust when dealing with highly volatile assets because they do not permit negative prices, ensuring that the logarithm is always defined.

The log-return for an asset i after d periods, denoted as $r_{it}^{\log}$, is defined as:

$$r_{it}^{\log} = \log\left(\frac{p_{it}}{p_{i(t-d)}}\right)$$

This formula is preferred because it approximates the continuously compounded rate of return, which aligns better with the assumptions of normality in asset returns, a common assumption in financial models like the E-V rule.

Portfolio selection problems that give optimal utility based on expected return and variance, as proposed by Markowitz (1952), are classified as *Mean-Variance Optimization* (MVO), a case of quadratic optimization problem (KOLM; TUETUENCUE; FABOZZI, 2014). The risk aversion formulation of the classical MVO is presented below.

$$\text{minimize} \quad w^T \Omega w - \lambda w^T \mu \tag{2.14}$$

$$\text{subject to} \quad \sum_{i=1}^{N} w_i = 1 \tag{2.15}$$

$$w_i \geq 0 \tag{2.16}$$

Equation (2.14) is the objective function and it expresses preferences of the DM relative to risk and return. λ is the risk aversion factor, which reflects the investor's objectives, ranging from 0 (risk-averse investor) to 10 (highly tilted toward higher risk) (FABOZZI et al., 2007), w_i represents stock's i proportion in the portfolio, Ω is the covariance matrix of the returns of assets composing the index, μ_i is the expected return of stock i and N is the number of assets in the universe. (2.15) is the budget constraint. In this model, non-dominated solutions are generated and a set of portfolios, named efficient frontier, is formed. If one chooses a solution contained in the set of non-dominated portfolios, one cannot change to another solution contained in this set without deteriorating one of the objectives (BEASLEY, 2013; RESENDE; RIBEIRO, 2016) (return or risk). Then, if one wants to improve return, it must deteriorate risk and vice-versa. A generic plot of this frontier is illustrated in Figure 6, where the horizontal and vertical axis represents the standard deviation and the expected return of the portfolio, respectively.

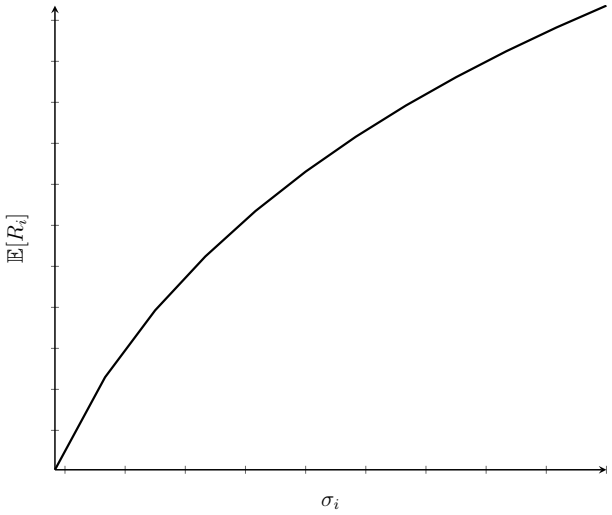


Figure 6 – The efficient frontier

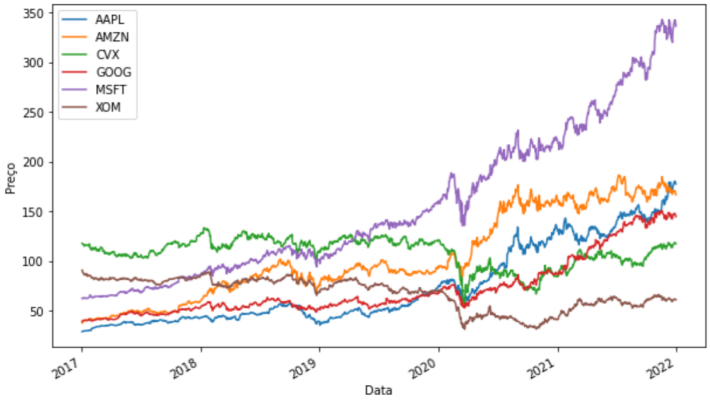


Figure 7 – Price data for 6 stocks

Investors' attitude towards risk and return is used as an indicator for the choice of a portfolio belonging to this frontier. If the investor is more tilted to high return, then it may select an asset that produces the highest expected return possible, otherwise, it may diversify his/her portfolio to mitigate risk or, if there is full risk aversion, choose an asset that contains the minimum standard deviation possible (FABOZZI et al., 2007).

Consider an example of price data for six stocks presented in Figure 7. The associated return data, shown in Figure 9, is used as input for the Markowitz optimization model. After optimizing the markwotiz model, it is possible to obtain a frontier of non-dominated portfolios, as shown in Figure.

In this example, 500 random portfolios were generated to demonstrate the effect of diversification in a universe of 6 assets. The random portfolios are diversified and therefore have less risk compared to Apple and Chevron. It is evident that there are portfolios a rational decision-maker would never choose, such as Chevron's, because there are portfolios that offer higher returns at a lower level of risk.

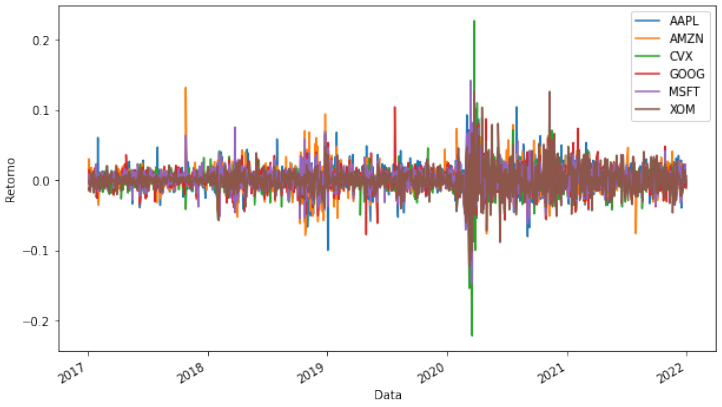


Figure 8 – Return data for 6 stocks

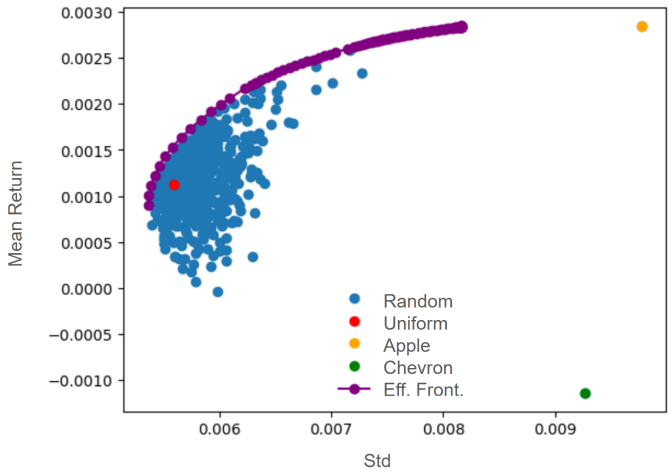


Figure 9 – Efficient Frontier generated from the data of the 6 stocks

2.4.1 Practical Portfolio Optimization

It is possible to modify or extend this classical framework, incorporating, for example, additional criteria or constraints, such that it becomes more realistic, reflecting the context and objectives of the financial agent, and also to obtain a more diversified portfolio. Extensions inserted in the model can include lot sizing, transaction costs, portfolio cardinality, various types of constraints reflecting specific characteristics of the investment, financial agent or country involved; alternative risk measures or modeling and quantification of the impact of wrong estimates of risk and return (FABOZZI et al., 2007; KOLM; TUETUENCUE; FABOZZI, 2014; BEASLEY, 2013).

2.4.1.1 Alternative risk measures

The advances of research on portfolio management include the development of new risk measures. Artzner et al. (1999) defined properties that a measure of risk ρ must satisfy to be considered coherent:

1. Monotonicity: Consider two asset returns X and Y, which are random variables. If

$X \geq Y$, then $\rho(X) \leq \rho(Y)$. Fabozzi et al. (2007) shows another way of seeing this property. If $X \geq 0$, then $\rho(X) \leq 0$. In other words, if there are only positive returns, then the risk should be non-positive.

2. Subadditivity: Merging assets in a portfolio do not create extra risk. $\rho(X + Y) \leq \rho(X) + \rho(Y)$

3. Positive homogeneity: For any positive real number λ, $\rho(\lambda X) = \lambda\rho(X)$. Portfolio size influences risk. In other words, large portfolio positions implies that their required liquidation time will also be large.

4. Translational invariance: For any real number α, $\rho(X + \alpha r) = \rho(X) - \alpha$. The main function of a risk measure is to rank risks. Therefore, inclusion of cash or any risk free asset does not contribute to portfolio risk,

Fabozzi et al. (2007) categorizes risk measures in two classes: Dispersion and Downside. Dispersion measures are measures of uncertainty that equally penalize overperformance and underperformance relative to the mean. Some types of these measures are listed below:

- **Variance of return**: Incorporated in the works of Markowitz (1952) and one of the most known dispersion measures.

$$VAR(R^p) = E\left[\left(\sum_{i=1}^{N} w_i R_i - E\left[\sum_{i=1}^{N} w_i R_i\right]\right)^2\right] \tag{2.17}$$

- ***Mean Absolute Deviation* (MAD)**: Introduced by KONNO & YAMAZAKI (1991). Uses absolute deviations in place of squared deviations. It is a dispersion measure based on the absolute deviations from the mean. The resulting optimization problem is linear, which is more solver-friendly than the MVO problem.

$$MAD(R_p) = E\left[\left|\sum_{i=1}^{N} w_i R_i - \sum_{i=1}^{N} w_i \mu_i\right|\right] \tag{2.18}$$

The resulting optimization problem is:

$$\text{minimize} \quad \frac{\sum_{t=1}^{T} y_t}{T} \tag{2.19}$$

$$\text{subject to} \quad y_t + \sum_{i=1}^{N} a_{it} w_i \geq 0 \tag{2.20}$$

$$y_t - \sum_{i=1}^{N} a_{it} w_i \geq 0 \tag{2.21}$$

$$\sum_{i=1}^{N} r_i w_i \geq \rho \tag{2.22}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{2.23}$$

$$w_i \geq 0 \tag{2.24}$$

Where r_{it} is the realization of the random variable R_i in period $t \in (1, ..., T)$, $r_i = E[R_i]$ and $a_{it} = r_{it} - r_i$.

A model that uses a downside risk measure aims to maximize the probability of satisfying a certain return threshold.

- **Semi-Variance**: Variance equally penalizes overperformance and underperformance. Markowitz (1959) presented this measure as an increment to its original variance measure of risk.
- ***Value-at-Risk* (VaR)**: Measures the predicted maximum loss at a specified probability threshold $1 - \alpha$ (eg. $\alpha = 5\%$).
- ***Conditional Value-at-Risk* (CVaR)**: Rockafellar & Uryasev (2000) proposed this measure in order to repair VaR deficiencies. This measure is a coherent risk measure, as it satisfies all the properties established by Artzner et al. (1999). Also, its optimization model is linear and can be solved very efficiently by optimization software.

2.4.1.2 Practical constraints

Depending on the institution or investor context, some constraints can be incorporated to the model in order to reflect practical issues. The optimization output may contain a few large positions and many small positions. This is undesirable due to extra transaction costs (FABOZZI et al., 2007). In this case, threshold/holding/floor-ceiling constraints can be included in the model:

$$\epsilon Z_i \leq w_i \leq \psi_i Z_i \text{ for each } i \in 1, ..., N \tag{2.25}$$

Where Z_i are binary variables. $Z_i = 1$ if asset i is included in the portfolio and $Z_i = 0$ if i is not included. Equation (2.25) ensures that if an asset i belongs to the portfolio, then its proportion w_i must lie between ϵ_i and ψ_i, otherwise, if i is not contained in the portfolio, $w_i = 0$. Investors might want to restrict their portfolio size, for example to build an index tracking portfolio and keep transactions cost low (FABOZZI et al., 2007; SANT'ANNA et al., 2017).

$$\sum_{i=1}^{N} Z_i = K \tag{2.26}$$

Equation (2.26) restrict the number of assets contained in the portfolio to K. Those constraints generate discontinuous efficient frontiers (CHANG et al., 2000). This is because some non-dominated portfolios of the original continuous efficient frontier would not be considered by any rational investor, as Fabozzi et al. (2007) exemplifies, there can be portfolios with less risk and greater returns. Thus, from incorporating these new constraints to MVO, a *Cardinality Constrained Portfolio Optimization* (CCPO) model is assembled and shown below.

$$\text{minimize} \quad w^T \Omega w - \lambda \mu^T w \tag{2.27}$$

$$\text{subject to} \quad \epsilon Z_i \leq w_i \leq \sigma_i Z_i \text{ for each } i \in 1, ..., N \tag{2.28}$$

$$\sum_{i=1}^{N} Z_i = K \tag{2.29}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{2.30}$$

$$Z_i \in \{0, 1\} \tag{2.31}$$

Other practical constraints can be incorporated into this model, i.e. class/sector, lot size, and transaction costs constraints.

2.4.1.3 Active and Passive strategies

Fund management and portfolio can be classified in two broad approaches: (BEASLEY; MEADE; CHANG, 2003; FABOZZI et al., 2007; JORION, 2003; ROLL, 1992):

- **Active management**: Markets are not fully efficient and management teams can work to make the portfolio achieve better performance than the market, using available information and their experience. Incurs high fixed costs, paid to the managers, and high transaction costs, because of frequent trading.

- **Passive management**: It is assumed that markets are efficient, so that market prices fully reflect risk and return. Incurs in lower fixed costs and lower transaction costs, but, if the market falls, the return falls.

Those two strategies are pure, but mixed strategies are possible too. This is the case when a portion is invested passively and the remainder is invested actively. This alternative can be illustrated with a model that aims to minimize *Tracking Error Variance* (TEV) conditional to a certain excess return target (ROLL, 1992):

$$\text{minimize} \qquad x^T \Omega x \tag{2.32}$$

$$\text{subject to} \qquad \mu^T x = G \tag{2.33}$$

$$\sum_{i=1}^{N} x_i = 0 \tag{2.34}$$

$$\tag{2.35}$$

Where $\mathbf{x} = \mathbf{w}_p - \mathbf{w}_b$ is a vector representing the difference between the managed portfolio and the benchmark proportions. G is the manager's expected performance relative to the benchmark.

2.5 INDEX TRACKING PROBLEMS

A passively managed fund is known as *index fund/tracker fund*. A manager that adopts this strategy could buy all the stocks of a given stock index and reproduce it perfectly (full replication), but this strategy has some disadvantages (BEASLEY; MEADE; CHANG, 2003; CANAKGOZ; BEASLEY, 2009; SANT'ANNA et al., 2017):

- The composition of the index is revised periodically. Therefore, the holdings of all stocks will change periodically to reflect the new composition's weights of the index.
- Transaction costs associated with the index's stocks cannot be limited since it is necessary to trade all stocks to reduce tracking error periodically.

The index tracking problem is concerned with index replication, but limiting transaction costs by using fewer stocks. Decisions concerning the maintenance of the tracking portfolio are enclosed in a decision support system. Important components of this system are (GAIVORONSKI; KRYLOV; WIJST, 2005):

- Benchmark or Index to be tracked
- Risk measure relative to deviations from the index (tracking error)

- Rebalancing strategies to reflect price changes in the market in portfolio weights
- Specify trade-off: maximum portfolio size and maximum tracking error allowed
- Decision rules regarding the cash flow generated by the portfolio (i.e. dividends)
- Decision rules regarding changes in the benchmark composition (i.e. merges)

There is a variety of country/world indexes to be tracked and these are provided by firms such as S&P (2024) and B3 (2024), that compute indexes' theoretical weights using their own methodology. Some risk measures relative to the benchmark, which are going to be identified as TE, are presented below (GAIVORONSKI; KRYLOV; WIJST, 2005; RUDOLF; WOLTER; ZIMMERMANN, 1999):

- **MAD relative to an index**: Absolute deviations between the benchmark and portfolio returns are minimized. Implies in a linear model.

$$TE = \frac{1}{T}\sum_{t=1}^{T}\left|R_t^I - \sum_{i=1}^{N} r_{it}w_i\right| \tag{2.36}$$

  Where R_t^I is the benchmark return in period $t \in \{1, ..., T\}$.

- ***Mean Squared Error*** **(MSE)**: Quadratic deviations between the benchmark and portfolio returns are minimized.

$$TE = E\left[\left(R_t^I - \sum_{i=1}^{N} r_{it}w_i\right)^2\right] \tag{2.37}$$

- **TEV**: Its formulation is presented in Section 2.4.1.3. The associated formulation is a quadratic optimization problem and it requires the benchmark weights.

$$TE = (w_p - w_b)^T \Omega (w_p - w_b) \tag{2.38}$$

  w_p is a vector that represents the portfolio weights to be optimized and $\mathbf{w}_b$ is a vector representing the benchmark proportions

- **VaR relative to an index**: Similar to VaR measure. It is the largest value w by which the portfolio return can miss the index target in $1 - \alpha$ fraction of cases.

$$TE = VaRI_\alpha = \inf_{\phi}\{\phi | P(\mu^T w \geq R_t^I - \phi) \geq 1 - \alpha\} \tag{2.39}$$

- **CVaR relative to an index**: It shows the mean deviation relative to the benchmark in the worst α cases.

$$TE = E\Big(R_t^I - \mu^T w \Big| \mu^T w < R_t^I - VaRI_\alpha\Big) \tag{2.40}$$

Even though constraint (2.26) is obligatory for this kind of problem, these risk measures can be combined with other practical constraints, such as those presented in Section 2.4.1.2, to reflect the context in which the tracking portfolio of an investor/institution is applied.

A systematic literature review on index tracking, spanning the last 30 years, was conducted and published during the development of this thesis (SILVA; FILHO, 2023). This review highlighted the critical role of the index tracking approach in portfolio optimization and the growing adoption of advanced techniques in recent years.

The objective of this systematic review was to develop an investigation concerning the current solution approaches for the index tracking problem using a set of research questions. The methodology consisted of searching for articles and conference papers written in the English language related to this class of portfolio selection problems. The research questions are presented in Table 1

RQ	Description
#1	Are index tracking solution methods more relevant to journals focusing on operations research and computer science?
#2	Is there a concentration of heuristic methods applied in a specific quantitative modelling framework?
#3	Has there been a growth in the number of non-heuristic methods applied to the index tracking problem?
#4	Has there been a growth in the number of heuristics/metaheuristics applied to the index tracking problem?
#5	Are heuristic approaches more used than non-heuristic approaches for index tracking problems?
#6	Do heuristic approaches have more cite impact than non-heuristic approaches for index tracking problems?
#7	Is there a prevalence of using a specific heuristic/metaheuristic in index tracking problems?
#8	Is there an integration between heuristic and general-purpose solvers?
#9	Is there a prevalence of using specific evaluation metrics for heuristic approaches?
#10	Is there a prevalence of using a specific solution method to compare with heuristic approaches?
#11	Is there a prevalence of solving for a specific tracking error objective function when using heuristic approaches?
#12	Is there a prevalence of solving for specific practical constraints when using heuristic approaches?
#13	Which data sources were most adopted in heuristic approaches?

Table 1 – Research questions for index tracking systematic literature review

The set of research questions was divided into two parts. The first part refers to general solution approaches for the index tracking problem and comparison among two groups: heuristic and non-heuristic methods. The production of both approaches grew in the last three decades, also, non-heuristic methods were more adopted than their counterparts. On the other hand, heuristics obtained the best performance when citation impact is taken into consideration, considering the metrics adopted in this work.

The second part refers to a specific analysis of heuristic/metaheuristic approaches applications developed for this problem. The first part of this analysis consisted of investigating the main heuristics, if there were hybridized heuristics/metaheuristics, comparison against other methods, and the associated evaluation metrics. Next, the model structure of the models solved by approximated methods was studied, taking into consideration objective functions, constraints, and data used in the problem. Table 2 summarizes the answers to each research question.

Researchers may refer to the published systematic review (SILVA; FILHO, 2023) for insights into advancements in index tracking problems, particularly when employing approximate solution methods. The review also provides detailed information on state-of-the-art model structures and commonly used algorithms, offering valuable guidance for studies focused on model and algorithm development within this domain.

RQ	Findings
#1	Index tracking solution methods are more relevant to journals focusing on operations research and computer science
#2	The vast amount of the developed heuristics/metaheuristics solutions were applied to mathematical programming formulations more often
#3	There has been a growth in the number of non-heuristic methods applied to the index tracking problem
#4	There has been a growth in the number of heuristics/metaheuristics applied to the index tracking problem
#5	Heuristic approaches are not more used than non-heuristic approaches for index tracking problems
#6	Heuristic approaches have more cite impact than non-heuristic approaches for index tracking problems
#7	There is a prevalence of using Differential Evolution and Genetic algorithms in index tracking problems
#8	Solvers are integrated with heuristics. A total of 11 hybridized heuristics were found
#9	There is a prevalence of using specific evaluation metrics for heuristic approaches. The most used metrics were RMSE and MSE
#10	Yes heuristics are more compared against other heuristics or the CPLEX solver
#11	There is a prevalence of solving for a specific tracking error objective function when using heuristic approaches. A good part of the works adopted RMSE and MSE
#12	No. There is no prevalence of solving for specific practical constraints when using heuristic approaches
#13	The most used databases were OR-library and datastream

Table 2 – Summary of the answers to each research question of the index tracking systematic literature review

This thesis makes a significant contribution by introducing uncertainty into the traditionally deterministic index tracking model through a stochastic programming approach.

This approach leverages synthetic market data and employs heuristics for problem-solving (Chapter 3). Furthermore, the thesis integrates investor preferences into the model, enabling the identification of tracking portfolios customized for various investor profiles (Chapters 4 and 5). This is achieved via a simulated interaction process that not only extracts investor preferences effectively but also ensures their satisfaction over time through heuristic-based adjustments.

2.6 CHAPTER CONCLUSION

In this chapter, some optimization and computational intelligence methods that can be applied to portfolio optimization were presented. An introduction to index tracking was provided, along with a discussion of the findings from a published systematic review on the topic. These methods were presented to give a base knowledge of the algorithms and methods developed in this work that will be presented in the next chapters.

3 AN INDEX TRACKING APPROACH BASED ON MULTIPLE MARKET SCENARIOS THROUGH GENERATIVE ADVERSARIAL NETWORKS

Metaheuristics are still efficient for the cardinality-constrained optimization problem since portfolios with a cardinality of at least five assets can't be solved by exact algorithms in a reasonable time (GRAHAM; CRAVEN, 2021). In index tracking, some approximate approaches have been proposed to produce good solutions that operate considering two partial allocation phases: selection of the subset of assets (asset selection) and weight/resource allocation.

Pure heuristics perform the two phases simultaneously (ANDRIOSOPOULOS et al., 2013; ANDRIOSOPOULOS; NOMIKOS, 2014; GRISHINA; LUCAS; DATE, 2017; AMORIM; SILVA; FILHO, 2021; GARCIA; GUIJARRO; OLIVER, 2018) and when there are hybridizations with general-purpose solvers, the first phase is performed by the heuristic approach and the second phase is performed by the solver (STRUB; TRAUTMANN, 2019; SCOZZARI et al., 2013; SANT'ANNA et al., 2017; RUIZ-TORRUBIANO; SUAREZ, 2009; GUASTAROBA; SPERANZA, 2012; WANG et al., 2012). The advantage of pure heuristics is that they can deal with very complex nonlinear and nonconvex problems. The advantage of hybridized heuristics is that there is no need for solution repairing and constraint handling mechanisms, as these problems are circumvented by the solver in the second phase.

Various metaheuristic algorithms have been proposed for addressing the index tracking problem. Some examples include Kernel Search (GUASTAROBA; SPERANZA, 2012), *Greedy Randomized Adaptive Search Procedure* (GRASP) (SILVA; SILVA; FILHO, 2022a), *Differential Evolution* (DE) (SCOZZARI et al., 2013), and quantum heuristics (FERNáNDEZ-LORENZO; PORRAS; GARCíA-RIPOLL, 2021). Although there are many different approaches available, a majority of the pure and hybridized metaheuristics proposed for the index tracking problem within the mathematical programming framework in recent years are based on *Genetic Algorithm* (GA)s.

GAs have been proposed by Holland (1975) and since then, they have been widely studied and applied. Inspired by the principles of natural selection and genetics, GAs are a class of algorithms that mimic the process of natural evolution to solve optimization problems. Due to the possibility of parallelizing the optimization process and their ability to handle complex non-linear problems without rigid mathematical assumptions or extra information like the gradient or Hessian matrix, GAs have found applications in various fields such as engineering, management, and computer science (OMIDVAR; LI; YAO, 2022; KATOCH; CHAUHAN; KUMAR, 2021; YU et al., 2022). Recently, GAs have been used in different applications, such as the optimization of complex production and logistics systems (Ahmed Bacha; BENATCHBA; Benbouzid-Si Tayeb, 2022; ARKHIPOV et al., 2020), to fine-tune epidemiological models (GHOSH; BHATTACHARYA, 2020), and feature selection in machine

learning (ZHOU; HUA, 2022; THAKKAR; CHAUDHARI, 2022).

GAs are very important in the field of portfolio optimization. Streichert's hybrid encoding (Streichert; Ulmer; Zell, 2004) enabled the development of genetic operators, constraint handling and repairing mechanisms that produce feasible financial portfolios when realistic constraints are considered, contributing to the extensive adoption of GAs in single and multiobjective portfolio optimization (ANAGNOSTOPOULOS; MAMANIS, 2010; ANAGNOSTOPOULOS; MAMANIS, 2011; SILVA; FILHO, 2021b; LIAGKOURAS; METAXIOTIS, 2018). To the best of our knowledge, Beasley et al. (BEASLEY; MEADE; CHANG, 2003) were the first to develop GAs for the index tracking problem with realistic constraints, providing a pure GA based on fitness and unfitness metrics, to solve nonlinear tracking error and excess return objective functions subject to cardinality, transaction costs and other constraints.

Andriosopoulos et al. (2013) compared the performance of pure GA and DE in building portfolios to track physical shipping markets, where portfolios produced by GA obtained the minimum tracking errors and maximum excess returns, irrespective of the rebalance frequency. Grishina, Lucas & Date (2017) studied a behaviourally based model, namely the prospect theory with index tracking model, and developed pure GA and DE approaches to solve it, where the GA approach had an advantage in terms of CPU time. Ni & Wang (2013) proposed a multiobjective index tracking model that considers accumulated excess return, tracking error and tracking error volatility. The authors combined lexicographic goal programming with a GA to obtain portfolios that minimize unwanted deviations from the most important objectives' targets.

Hybrid GAs have been proposed for index tracking, considering general-purpose solvers to perform a more objective capital allocation. Ruiz-Torrubiano & Suarez (2009) introduced a hybridized GA for a quadratic index tracking model, where the genetic operators performed the asset selection phase and an exact quadratic solver performed the capital allocation phase in the relaxed index tracking model. The hybrid approach can obtain near-optimal solutions in an acceptable time depending on the size of the index and cardinality of the portfolio.

Sant'Anna et al. (2017) proposed a hybrid heuristic incorporating kernel search (GUASTAROBA; SPERANZA, 2012) ideas, where solutions are constructed considering information about the weight of the assets in the exact solution of the relaxed index tracking model with tracking error constraints, instead of random initialization. Based on Sant'Anna et al. (2017), Anis, Costa & Kwon (2023) proposed an exploration–exploitation GA that uses multiple restarts (exploration) and focuses on a promising feasible region (exploitation). Strub & Trautmann (2019) proposed a hybrid GA to solve the index tracking model that incorporated fund regulation constraints imposed by the European Union. The hybrid GA obtained competitive performance relative to the pure GA and Gurobi.

Another direction in index tracking research concerns statistical approaches that aim

to compute solutions with a good out-of-sample performance and that produce these solutions faster than metaheuristics applied to index tracking problems based on the mathematical programming framework. In the literature, these models are commonly referred to as regularized or sparse index tracking models (BENIDIS; FENG; PALOMAR, 2018; GIUZIO; FERRARI; PATERLINI, 2016; TAS; TURKAN, 2018; WU; YANG, 2014; WU; YANG; LIU, 2014; YANG; WU, 2016; ZHAO; LIAN, 2016; SANT'ANA; CALDEIRA; FILOMENA, 2020). In the general model, the squared errors between the index and portfolio returns are minimized, generally considering budget and long-only constraints. The parameter that controls the portfolio sparsity is associated with a penalty function that approximates the exact cardinality constraint.

The real cardinality constraint is the ℓ_0 norm, which counts the number of nonzero elements in the portfolio. Since minimizing ℓ_0 is NP-hard, it is more advantageous to employ a continuous and differentiable function to approximate ℓ_0. Some motivations for the adoption of regression with regularization models are the overfit reduction and a more efficient portfolio optimization process (XU; LU; XU, 2016; FASTRICH; PATERLINI; WINKER, 2014; GIUZIO, 2017; BENIDIS; FENG; PALOMAR, 2018), becoming advantageous for large indexes, such as the S&P 500 and Russell 2000.

Convex approaches like Least Absolute Shrinkage and Selection Operator (LASSO) that uses the ℓ_1 penalty and elastic net that combines the ℓ_1 and ℓ_2 penalties do not satisfy the nonnegative weights constraint. Then, it is possible to satisfy this constraint of index tracking models with the nonnegative LASSO (WU; YANG; LIU, 2014; YANG; WU, 2016) and nonnegative Elastic net (WU; YANG, 2014; DING et al., 2023). Recently, Liu et al. (2023) proposed the nonnegative group bridge, which utilizes the efficient group bridge method to select inter-group and intra-group variables simultaneously.

Also, ℓ_q norms, where $0 < q < 1$, can also be used to satisfy the nonnegativity constraint, but are non-convex (BENIDIS; FENG; PALOMAR, 2018; GIUZIO, 2017; FASTRICH; PATERLINI; WINKER, 2014). The ℓ_q can control the diversity of the portfolio, where the lower the upper bound on the ℓ_q norm, the less diversified the portfolio (FASTRICH; PATERLINI; WINKER, 2014). Other approaches have been proposed to incorporate more realistic aspects into sparse index tracking models, such as uncertainty (GIUZIO; FERRARI; PATERLINI, 2016), transaction costs (SHU; SHI; TIAN, 2020) and quantile regression (ZHAO; LIAN, 2016). Recently, a new way to control the number of assets in this approach was proposed in (LI et al., 2022).

Recent research has been introducing deep learning in the index tracking context to incorporate the learning of more complex relationships between the available assets and the market index. Thus, this new perspective tries to provide even better solutions than those provided by the previous frameworks studied in the literature.

To the best of our knowledge, the study of Heaton et al Heaton, Polson & Witte (2018) was the first to propose an autoencoder-based index tracking framework. In their approach,

a deep autoencoder was used to detect and select stocks most similar to the benchmark by calculating the information loss during the encoding-decoding process. If a stock was closer to its auto-encoded version, its communal information, or the information content of this stock in the aggregate information of the universe of assets, was higher. The asset selection phase was performed by ranking and choosing the K stocks concerning their communal information, where K is the cardinality of the portfolios. In the second phase of the partial allocation, the weights were defined based on the equal-weighted $1/K$ scheme.

Considering the Heaton, Polson & Witte (2018) framework, Ouyang et al Ouyang, Zhang & Yan (2019) used deep autoencoders to perform the asset selection process, but also proposed the inclusion of a deep neural network to perform resource allocation without using information about the benchmark. The authors proposed a method to obtain a set of weights connecting the input and output units directly and thus capturing nonlinear relationships between the selected assets. Thus, this last network performs a dynamic weight allocation driven by the selected stocks' time series behaviour.

Kim & Kim (2020) used latent vectors from the deep autoencoder approach, inspired by Ouyang, Zhang & Yan (2019), Heaton, Polson & Witte (2018), and stacked autoencoders, to select the K most similar assets. The similarity between an asset return vector and the generated latent vector was computed according to either the correlation coefficient or the mutual information criteria. The weights were defined based on the equal-weighted $1/K$ scheme.

Zhang et al. (2020) evaluated the performance of six different autoencoders on the construction of tracking portfolios with different cardinalities K. The authors used autoencoders to select a subset of assets based on the extracted non-linear relationship between the available assets and the market index. The weights were optimized through a regularized quadratic programming index tracking model, considering long-only constraints. The autoencoders performed better when used to build small cardinality portfolios (less than 30 assets), where the best tracking portfolios were constructed with the sparse and denoising autoencoders.

As was shown in some works on autoencoder-based index tracking (OUYANG; ZHANG; YAN, 2019; HEATON; POLSON; WITTE, 2018; ZHANG et al., 2020), it is not beneficial to include too many stocks contributing with the same information in an autoencoder-based index tracking approach. It is recommended to combine the most-communal stocks and the least-communal stocks to improve tracking performance.

Recent studies went in a different direction from the autoencoder-based index tracking. They performed the asset selection phase by using a deep neural network architecture with fixed noise (Deep NNF), in which the portfolio's weights were the softmax layer probabilities generated by the fixed noise (KWAK; SONG; LEE, 2021). The authors adopted fixed noise as an input to prevent changes in the portfolio's weights before the rebalancing period as new data comes in during training. The authors used the Deep NNF model to

perform both partial allocation steps (selecting assets and allocating weights). Zheng et al. (2020) proposed a way to perform asset selection through a stochastic neural network. This neural network learns the stochastic process that samples assets in a way that minimizes the tracking error loss function through a first-order method, such as gradient descent.

There are advantages and disadvantages to using deep learning and sparse index tracking approaches. The deep learning index tracking approach can learn complex non-linear relationships between assets and the index but doesn't have the model extension capabilities of the mathematical programming models and sparse index tracking. Also, concerning autoencoder-based index tracking, there is no consensus about the best way to combine the assets based on their communal information. The sparse index tracking approach aims to minimize overfitting and is useful for large indexes, a scenario that is difficult for the mathematical programming approach. But, there are some limitations relative to model extension and these approaches are associated with the challenge of obtaining optimal penalty parameters (SHU; SHI; TIAN, 2020; LI et al., 2022).

Different from the deep learning and sparse index tracking approaches, the mathematical programming approach can handle a diversity of real-world constraints and objective functions formulations for index tracking, such as convex *Mixed-Integer non-linear programming* (MINLP) reformulation for the index tracking cointegration approach (SANT'ANA; CALDEIRA; FILOMENA, 2020), liquidity requirements (VIEIRA et al., 2021), to approach index tracking as a multiobjective problem (NI; WANG, 2013), and incorporate uncertainty through scenario-based optimization (MELLO; BAYRAKSAN, 2014; DEB; ZHU; KULKARNI, 2018).

Overall, this work advances the state-of-the-art approximate approaches for mathematical programming index tracking models and leverages the power of machine learning in index tracking, providing valuable insights and practical implications for researchers, practitioners, and investors in the field.

Concerning research on machine learning models applied in index tracking, this study proposes an innovative approach to index tracking by combining GANs with index tracking based on mathematical programming. This allows for a more realistic building process of index tracking portfolios using deep learning relative to past works. The GANs can capture complex nonlinear relationships between assets and the index and generate synthetic market scenarios. Thus, it is possible to take advantage of the model extension capabilities of mathematical programming and incorporate uncertainty using synthetic scenarios generated by GANs to produce more robust portfolios.

Approximate approaches that solve the NP-hard mathematical programming index tracking models are important to obtain good solutions in feasible time. The recently proposed metaheuristics and GAs that optimize index tracking portfolios only deal with models that consider historical data. This study proposes ways to incorporate multiple synthetic market scenario information generated by GANs to incoporate uncertainty in

single and multiobjective GAs that can address various realistic constraints in index tracking models.

3.1 INDEX TRACKING PORTFOLIOS THROUGH GANS

3.1.1 Index tracking model with simulated data

The traditional index tracking approach uses real (historical) data from an in-sample period of size b. The historical data is a matrix M_b with dimensions $N \times b$, where N is the number of available assets to invest or the size of the universe of assets. By optimizing the index tracking problem with the *Real Data Model* (RDM), formulated using Equations (3.1)-(3.5), a subset of K assets from the universe of assets is used to track a benchmark.

$$\text{Min} \qquad TE = \frac{\sum_{t=1}^{b}[R_t^p - R_t^I]^2}{b} \tag{3.1}$$

$$\text{subject to} \qquad \epsilon_i Z_i \leq w_i \leq \psi_i Z_i,\ i = 1, ..., N \tag{3.2}$$

$$\sum_{i=1}^{N} Z_i = K \tag{3.3}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{3.4}$$

$$Z_i \in \{0, 1\} \tag{3.5}$$

where $R_t^p = \sum_{i=1}^{N} w_i r_{it}$ is the constructed portfolio return at time t. The objective function (3.1) represents a mean squared error objective function to be minimized, $t = 1, 2, ..., b$ is the time index, $i = 1, 2, ..., N$ is the asset index, R_t^I is the benchmark return at time t, r_{it} is the stock i return at time t, w_i is the stock i proportion in the portfolio, Z_i indicates if stock i is included in the portfolio or not, and K = is the number of assets in the portfolio. Constraint (3.2) guarantees that if i belongs to the portfolio, then its proportion w_i lies between ϵ_i and ψ_i, otherwise, if i is not contained in the portfolio, $w_i = 0$. Finally, constraint (3.3) is the cardinality constraint that controls the size of the portfolio, and (3.4) is the budget constraint.

Next, the index tracking approach that constructs portfolios through multiple scenarios from simulated data is presented. A simulation $\hat{M_f}^s$, $s = 1, 2, ..., |S|$, which is a sample from the distribution $P(\hat{M}_f|M_b)$, is a matrix with dimensions $N \times f$, where f is the size of the out-of-sample period to be simulated. This shows that in our approach, simulations depend on the most recent historical data that would be used in RDM. The set of sampled simulations is defined as S. The elements of the matrix s are the simulated returns r_{it_s}, where $t_s = \{1, 2, .., f\}$ is the out-of-sample time index for all s. The return of the portfolio at time t_s is given by $R_{t_s}^p = \sum_{i=1}^{N} w_i r_{it_s}$.

As was already known, index tracking models aim to minimize the mean squared error portfolio return relative to the index. Thus, it is possible to use the simulations $s \in S$ to estimate the distribution $P(\varepsilon|M_b, w)$, where $\varepsilon = TE_s = \frac{\sum_{t_s=1}^{f}[R_{t_s}^p - R_{t_s}^I]^2}{f}$ is the tracking error of the constructed portfolio in a simulation $s \sim P(\hat{M}_f|M_b)$, and $R_{t_s}^I$ is the simulated benchmark return at time t_s. From $P(\varepsilon|M_b, w)$, an objective function $\theta(M_b, w)$ to be minimized can be considered in the index tracking problem through the *Synthetic Data Model* (SDM), presented below.

$$\text{Min} \quad \theta(M_b, w) \tag{3.6}$$

$$\text{subject to} \quad \epsilon_i Z_i \leq w_i \leq \psi_i Z_i,\ i = 1, ..., N \tag{3.7}$$

$$\sum_{i=1}^{N} Z_i = K \tag{3.8}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{3.9}$$

$$Z_i \in \{0, 1\} \tag{3.10}$$

For instance, one could aim to minimize the mean tracking error of the simulations, by making $\theta(M_b, w) = \mathbb{E}_{TE \sim P(\varepsilon|M_b,w)}[TE]$. After constructing a portfolio using RDM or SDM the realized tracking error is observed on the true out-of-sample data using Equation 3.11 presented below.

$$TE^{[\tau,\tau+v-1]} = \frac{\sum_{t=\tau}^{\tau+v-1}[R_t^p - R_t^I]^2}{v} \tag{3.11}$$

where τ is the initial out-of-sample time index and v is the rebalancing frequency or the frequency in which SDM or RDM are optimized. The rebalancing frequency method allows a fund to establish a discipline on how frequently it will reflect new market conditions as time goes by. If the portfolio is rebalanced every 20 time units, then $v = 20$. Given that the test data is given by a matrix M_T with dimensions $N \times T$, the mean tracking error over the test set can be computed from Equation 3.12. Of course that b columns from M_T will be used as the first block of historical data. Thus, the initial out-of-sample period is $\tau = b + 1$.

$$TE^{M_T}(v) = \frac{\sum_{k=0}^{\gamma-1} TE^{[\tau+kv,\tau+kv+v-1]}}{\gamma} \tag{3.12}$$

where $\gamma = \lceil \frac{T-b}{v} \rceil$ is the number of rebalances performed in the test set and $\tau + kv + v - 1 \leq \tau + T - 1$. A ceiling function was used to extract γ, since it is possible the ratio can result in a real number rather than an integer, then it is necessary to limit the last time index.

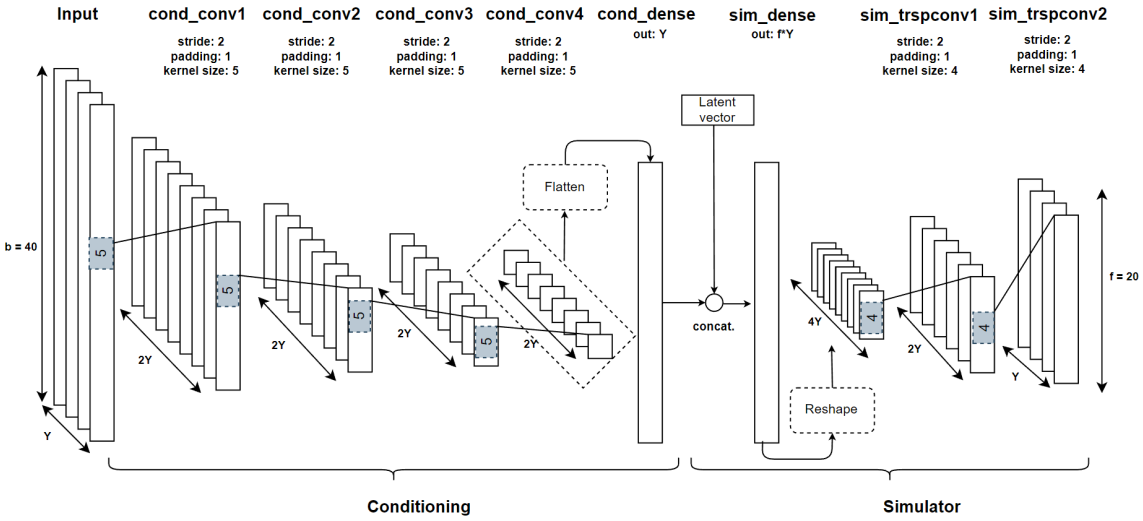
3.1.2 GAN architecture

The adopted GAN architecture is based on the WGAN-GP methodology (GULRAJANI et al., 2017), and is presented in Figure 10. The WGAN-GP methodology was proposed by Gulrajani et al. (2017) to improve WGANs. The purpose of WGAN-GP is to replace the weight clipping, used in WGANs to enforce the 1-Lipschitz constraint. A discriminator may not be forced to satisfy this constraint, but when it does, the GAN performs better than the traditional GAN framework. Weight clipping may lead to exploding and vanishing gradients, then the proposed alternative to deal with these problems is the gradient penalty (GP). GP enforces the 1-Lipschitz continuity constraint by penalizing the critic's output gradient, such that this gradient is forced to have a unitary norm.

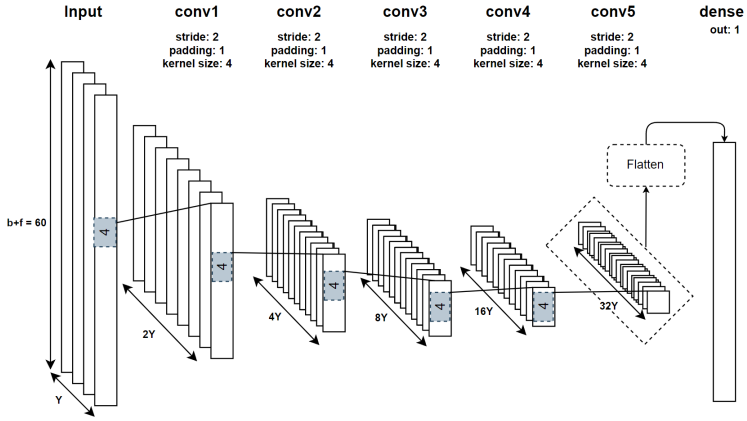
The generator (Figure 10a) considers the current market state to simulate a market scenario $\hat{M}_f$, containing a horizon of size f for Y assets. The benchmark data was also included in the input, as shown in Figure 11, thus $Y = N + 1$ is the total number of assets and the index. The generator G is constituted of two networks: conditioning and simulator. The presence of a conditioning component in the generator makes the proposed GAN a conditional GAN or cGAN (see (MIRZA; OSINDERO, 2014)), where the generator is conditioned on some extra information. In our case, the extra information presented to the generator is the compressed market state representation.

The conditioning network processes the historical data input M_b through sequential 1D convolutional layers over the time dimension. Thus, the current state of the market presented to G will have its time-series data representation compressed sequentially. To generate random scenarios, a latent vector $\lambda \sim \mathcal{N}(0, 1)$ with dimensions $2Y \times 1$ and the compressed market state are used as an input for the simulator to generate different outputs every time a new simulation is requested. The latent vector represents random events that could affect the market. The activation function of the conditioning and simulator hidden layers was $ReLu$. $Tanh$ was chosen as the activation function of the output layer of the simulator because the output belongs to $[-1, 1]$, which can better represent the returns' domain.

The discriminator D outputs a critic value for a real market realization M_f or a simulated market simulation $\hat{M}_f$. Thus, the discriminator receives as input either the concatenation of M_b and M_f or the concatenation of M_b and $\hat{M}_f$, where the representation of this concatenation is sequentially compressed by convolutional layers. $LeakyReLU$ was adopted as the activation function of the discriminator hidden layers. This GAN model will be used before each portfolio rebalancing to input simulation data into the SMD model.



(a) The generator network. This network is composed by the conditioning and simulator networks.



(b) The discriminator network. This network returns a critic value for a real or simulated (fake) market scenario.

Figure 10 – GAN architecture used in this work

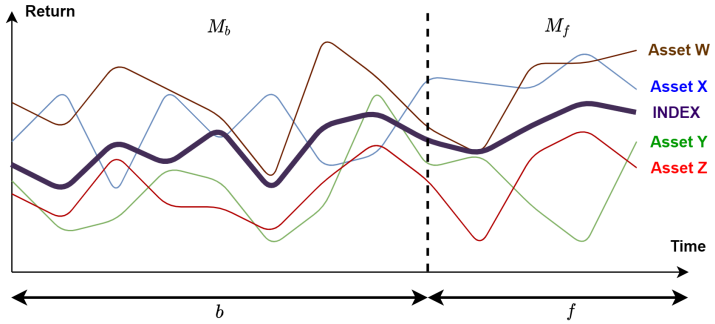


Figure 11 – GAN input representation for index tracking

3.1.3 Model evaluation

Although there are different quantitative scores to evaluate GANs that were trained to generate images (JABBAR; LI; OMAR, 2021), a human can also complement the assessment by visually inspecting the quality and diversity of the generated images (BORJI, 2019). It is more difficult to visualize the simulation quality and diversity in a portfolio optimization context, due to the visual pollution and noise associated with the superposition of multiple time series for a large number of assets. To evaluate the simulations in this context one may investigate if the simulations generated by a model m are benefic for a solver (or heuristic).

Along with the challenges associated with the direct evaluation of the quality of the simulations generated by GANs in portfolio optimization due to the lack of a quantitative or qualitative assessment method, other complications emerge due to the combinatorial nature of this problem when cardinality constraints are included. It is hard to continuously monitor the progress of the model during training since solving the index tracking optimization problem in every epoch for a specific GAN model would make the training process cumbersome and time-consuming as the universe of assets grow. Therefore, it is necessary to find a way to sample GAN models before evaluating them.

This study proposes to train a GAN model for a fixed number of epochs e_{MAX} and evaluate its performance from equal-distanced epochs (checkpoints). Each (GAN, Epoch) pair is considered to be an individual model. What must be taken into consideration is that it is unknown if the best model will be found in the last epoch and that the specific objective is to track a benchmark. The evaluation consists in comparing the performance of a heuristic SDM-h that optimizes SDM using simulated data from m to that of a heuristic RDM-h that optimizes RDM. Thus, the models are sampled in a sequence of epochs $c = \{e_0 = \Delta E, e_0 + \Delta E, e_0 + 2\Delta E, ..., e_{MAX}\}$, where $\frac{e_{MAX}}{\Delta E}$ is the number of samples per model m.

The model samples (m, e) are generated through the GAN training process. Thus, the checkpoints for each model m are saved according to ΔE. For instance, if $e_{MAX} = 1000$ and $\Delta E = 50$, then the first checkpoint for a given model m is $e = 50$ and the last is $e = 1000$, giving a total of 20 checkpoints.

Algorithm 2 was proposed to evaluate the sampled overall tracking errors of a model m, given e_{MAX} and ΔE, to get the epoch in which m performed better, with respect to the overall tracking error. It begins by initializing the current best epoch e_{best} and the current best overall tracking error $TE^{M_t}_{best}$. Next, the algorithm loops over each model sample (m, e) and evaluates the performance of SDM-h that solves SDM from the simulated data provided by a sampled model. If the resultant overall tracking error from (m, e) is lower than $TE^{M_t}_{best}$, the variables e_{best} and $TE^{M_t}_{best}$ are updated. After the loop evaluates m for all the considered epochs e, the combination (m, e) that minimized $TE^{M_t}_{best}$ is added to the set of good quality models Φ.

Finally, SDM-h and RDM-h are compared. To do this, a vector TE_{RD} is considered, where each element of this vector is an overall tracking error TE_{exec}^{MT}, $exec = 1, 2, ..., \alpha_{RD}$, obtained by running RDM-h for α_{RD} times, for a given v. From this vector, it is possible to calculate, for instance, the mean overall tracking error when using a real data model $\mu[TE_{RD}]$. Then, it is possible to compare the overall performance of Φ against the performance of TE_{RD}.

Algorithm 2: Model evaluation algorithm

Input: ΔE, e_{MAX}, m_{MAX}, M_T, v, SDM-h, α_{RD}
Output: Φ
$e \leftarrow 0$, $\Phi \leftarrow \{\}$
for m = 1:m_{MAX} **do**
  $e_{best} \leftarrow \{\}$
  $TE_{best}^{M_t} \leftarrow \infty$
  while $e \leq e_{MAX}$ **do**
    $e \leftarrow e + \Delta E$
    $GAN \leftarrow getModel(m, e)$
    $TE^{M_T} \leftarrow getSDM(GAN, \text{SDM-h}, M_T, v)$
    if $TE^{M_T} \leq TE_{best}^{M_t}$ **then**
      $TE_{best}^{M_t} \leftarrow TE^{M_T}$
      $e_{best} \leftarrow e$
    end if
    $\Phi \leftarrow \Phi \cup (m, e_{best})$
  end while
end for

A final consideration is that three conditions were adopted to compare the RDM-h and SDM-h. The first is that their search mechanisms are the same. Thus, if a GA is adopted, then the selection, crossover, and mutation operators of SDM-h and RDM-h are the same. The second is that although their objective to be optimized is the same, the only difference between them is their solution evaluation procedure (i.e. fitness calculation). The final consideration is that the in-sample data M_b used by SDM-h and RDM-h are the same for a certain rebalance period t. Thus, considering a rebalancing period t where the in-sample data is M_B, M_b is used to optimize RDM through RDM-h, and the same M_b is used as an input to the model (m, e) to generate a set of simulations S to solve SDM through SDM-h.

3.1.4 Metaheuristics to solve the multiple scenario index tracking problem

This work proposes to use GAs to solve RDM and SDM, since this kind of heuristic is widely adopted in the portfolio optimization context (KALAYCI; ERTENLICE; AKBAY, 2019; ALMEIDA-FILHO; SILVA; FERREIRA, 2021), and in many studies of the index tracking literature. The Streichert (Streichert; Ulmer; Zell, 2004) hybrid encoding was adopted. A real-valued vector $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ represent the weights of the portfolio and the assets included in the portfolio are represented by a binary vector $\mathbf{B} = \{Z_1, Z_2, ..., Z_N\}$. Also, elitism was adopted in the GAs, since one of them is based on NSGA-II (Deb et al., 2002). The genetic operators selected based on the literature and the considerations presented in subsection 3.1.3 are presented below.

- **Selection:** The binary tournament selection was adopted since it is often adopted in the portfolio optimization context (LIAGKOURAS; METAXIOTIS, 2018). In this strategy, two randomly selected solutions compete in a tournament for a place in the mating pool. When a single-objective problem is considered, the winner of the tournament is the fittest one. In a multi-objective problem, the winner is the solution contained in the best non-dominated front, or, if both solutions belong to the same front, the solution with a higher crowding-distance.

- **Crossover:** Uniform crossover was applied to the binary vector, as it was performed in many studies involving cardinality constrained portfolio selection (LIAGKOURAS; METAXIOTIS, 2018). In this operator, a single child is generated from two parents. Assets that are included in both parents will also be present in the child. Assets contained in only one of the parents have a 50% chance of being present in the child (ANAGNOSTOPOULOS; MAMANIS, 2010; ANAGNOSTOPOULOS; MAMANIS, 2011).

- **Mutation:** Sant'Anna et al. (2017) bit-flip mutation operator was applied in the binary vector of an individual with probability $p_B = \frac{1}{N}$. It exchanges one stock contained in the portfolio for another not contained in the portfolio. A Gaussian random mutation with $\sigma = 0.15$ is applied in each decision variable on the real-valued genotype. (ANAGNOSTOPOULOS; MAMANIS, 2011; Streichert; Ulmer; Zell, 2004).

After generating an initial population, or each time crossover and mutation are applied, it is necessary to satisfy the constraints (3.2)−(3.5). Thus, the repairing mechanisms described in Streichert, Ulmer & Zell (2004) were adopted to adjust the generated portfolios during the GA process.

3.1.4.1 SDM-SAAGA-GAN

The first genetic algorithm is based on the *Sample Average Approximation* (SAA) approach (MELLO; BAYRAKSAN, 2014; SHAPIRO; MELLO, 1998). The estimate of $\theta(M_b, w) = \mathbb{E}_{TE\sim P(\varepsilon|M_b,w)}[TE]$ is obtained considering a single realization of the random sample of simulations S, which gives $\hat{\theta}(M_b, w) = \frac{1}{|S|}\sum_{s\in 1,2,\ldots,|S|} TE_s$. The name of the metaheuristic is SDM-SAAGA-GAN. The proposed algorithm for SDM-SAAGA-GAN is presented in Algorithm 3.

At line 2, a new population R_g is initialized containing a total of n_I portfolios. The algorithm loops at lines 4-12 until the predefined number of generations n_G is achieved. The elitist selection is performed at lines 4−6, selecting the best individuals from R_g based on their fitness $\hat{\theta}$ and constructing the set of indexes P. In line 7, the best current solution is selected. The population of the next generation is computed in lines 8-12. At line 8 the new population R_{g+1} is initialized by using the indexes of the best individuals of the current population P. Next, at line 9, the mating pool is generated by applying the

Algorithm 3: SDM-SAAGA-GAN

Input: n_I, n_G, p_c, p_m S, N, K
Output: $x, \theta(M_b, x), R_g$
1: $g \leftarrow 1$
2: $R_g \leftarrow getInitialPop(n_I, N, K)$
3: **while** $g \leq n_G$ **do**
4: $fit \leftarrow getFitness(S, R_g)$
5: $F \leftarrow sort(fit)$
6: $P \leftarrow getCurrentPop(F, n_I)$
7: $x, \theta(M_b, x) \leftarrow selectBest(P)$
8: $R_{g+1} \leftarrow getNewPop(P, n_I)$
9: $matingPool \leftarrow selection(P, fit)$
10: $Q \leftarrow crossover(matingPool, R_g, N, p_c)$
11: $Q \leftarrow mutation(Q, n_I, N, p_m)$
12: $R_g \leftarrow concat(R_{g+1}, Q)$
13: **end while**

selection operator in R_{g+1}, considering their fitness. At lines 10 and 11, crossover, followed by mutation, are applied with probabilities p_c and p_m, respectively. The population of the next generation is assembled by concatenating the offspring Q with their parents R_{g+1}.

Algorithm 3 can solve RDM. This can be achieved by making $S = \{M_b\}$, instead of $S = \{\hat{M_f}^1, \hat{M_f}^2, ..., \hat{M_f}^{|S|}\}$. Thus the fitness will be calculated based on a unique tracking error from the performance of a portfolio in the historical data M_b, which is different from taking the mean of the tracking errors from the performance of a portfolio in all simulations $\hat{M_f}^s \in S$. When the RDM is solved with Algorithm 3, the algorithm is referred to as RDM-GA.

3.1.4.2 SDM-SBDGA-GAN

Another perspective for solving the SDM can be evaluated by transforming it into a multi-objective optimization problem, containing $|S|$ objectives, which are the tracking errors to be minimized in each scenario. MS-NSGA-II can be used to solve problems where it is necessary to find solutions with a well-balanced compromise to multiple scenarios and maintain the trade-off among multiple objectives. Algorithm 4 was proposed for the procedure of a metaheuristic based on the MS-NSGA-II, which aims to find solutions with the best compromise to multiple scenarios only. Since there is only one objective to be minimized (TE), it is not necessary to worry about the trade-off between multiple objectives.

The population R_g is initialized at lines 2−5. This initial population is constituted of the last generation populations of SDM-SAAGA-GAN obtained for each scenario taken as an individual optimization problem. For this approach, the $getFitness(S, R_g)$ function returns a vector of tracking errors for each individual, instead of a unique tracking error. In lines 7 and 8, by calculating the fitness of the portfolios that belong to R_g, it is possible to obtain the vectors that store the maximum and minimum fitness of each scenario.

After the generations loop begins, the elitist selection is performed at lines 10−13. The

Algorithm 4: SDM-SBDGA-GAN

Input: n_I, n_G, p_c, p_m S, N, K
Output: $x, \theta(M_b, x)$

$g \leftarrow 1$
$R_g \leftarrow \{\}$
for $s \in S$ **do**
    $x, \theta(M_b, x), R_s \leftarrow$ SDM-SAAGA-GAN$(n_I, n_G, p_c, p_m, s, N, K)$
    $R_g \leftarrow concat(R_g, R_s)$
end for
$fit \leftarrow getFitness(S, R_g)$
$max_s, min_s \leftarrow getScenarioExtremes(fit)$
while $g \leq n_G$ **do**
    $fit \leftarrow getFitness(S, R_g)$
    $F \leftarrow nonDominatedSort(fit)$
    $crMetric \leftarrow getCRMetric(fit, max_s, min_s)$
    $P, lastFrontIdx, fRank \leftarrow getCurrentPop(F, n_I)$
    $x, \theta(M_b, x) \leftarrow selectBest(P, R, S, crMetric)$
    $R_{g+1} \leftarrow getNewPop(P, n_I)$
    $matingPool \leftarrow selection(P, fit, fRank, crMetric)$
    $Q \leftarrow crossover(matingPool, R_g, N, p_c)$
    $Q \leftarrow mutation(Q, n_I, N, p_m)$
    $R_{g+1} \leftarrow concat(R_{g+1}, Q)$
end while

solutions from R_g are ranked according to the non-dominated sort (Deb et al., 2002), which generates a set of non-dominated fronts F. A scenario-based crowded rank procedure was proposed in (DEB; ZHU; KULKARNI, 2018). From this procedure, the computed $crMetric$ is used to maintain the diversity among individuals in NSGA-II and to maintain a diverse set of scenario-wise nondominated solutions in the multiple scenario problem. The $crMetric$ of a solution x is used to maintain the scenario-wise diversity only and compute it using Equation 3.13.

$$CR(x) = |X_{max}(x)| + \mu[fit(x)] \tag{3.13}$$

where $X_{max}(x)$ is the set of scenarios where x performed poorly. In other words, $|X_{max}(x)|$ counts in how many scenarios the tracking error of x was closer to an upper bound tracking error contained in max_s. $X_{max}(x)$ was adopted because the strategy is to find solutions with the highest number of scenarios where it performed well. To untie solutions with an equal value of $|X_{max}(x)|$ one can adopt $\mu[fit(x)] = \hat{\theta}(M_b, w)$. More detail about the computation of CR is presented in the proposed Algorithm 5.

At lines 2-14 of Algorithm 5, the CR metric is computed for each solution x of the population and the respective values are stored in the vector $crMetric$. It is considered that $n_I = fit.dim[0]$ and $|S| = fit.dim[1]$, since fit is a $n_I \times |S|$ matrix. The absolute distances of the tracking error obtained by x on scenario s relative to the lower and upper bound tracking errors of s are d_{min} and d_{max}, respectively. Using these distances, X_{max} is computed at lines 4-11. If the tracking error obtained by x in s is closer to the lower tracking error of s, then this scenario is added to X_{min}, otherwise, s is added to X_{max}.

Algorithm 5: getCRMetric

```
1   Input: fit, max_s, min_s
2   Output: crMetric
 1: crMetric ← {}
 2: for x = 1 : fit.dim[0] do
 3:     X_min ← {},X_max ← {}
 4:     for s = 1 : fit.dim[1] do
 5:         d_min ← |fit[x, s] − min_s[s]|
 6:         d_max ← |fit[x, s] − max_s[s]|
 7:         if d_min < d_max then
 8:             X_min ← X_min ∪ s
 9:         else
10:             X_max ← X_max ∪ s
11:         end if
12:     end for
13:     CR ← |X_min| + μ[fit[x, :]]
14:     crMetric ← crMetric ∪ CR
15: end for
```

Going back to Algorithm 4, at line 13 the elitist selection constructs the set of indexes of the best solutions P based on the non-dominated fronts F. The $getCurrentPop(F, n_I)$ function also returns the rank of the solutions according to F, named $fRank$, where the lower the rank, the better the solution. At line 14 the best solution x is selected based on the best $CR(x)$ (the lower the $CR(x)$ the better the solution x).

In line 14, the best current solution is selected. Since the process of computing the next generation population was already explained (lines 15-19), the binary tournament selection will be considered from the scenario-based dominance perspective. In SDM-SBDGA-GAN, the mating pool is generated by applying the selection operator in R_{g+1}, considering that the winner is the one with the best $fRank$, or if the competitors have the same rank, the solution with the best $crMetric$ wins.

The feature that distinguishes the adopted GAs from each other is their fitness calculation. Thus, it is possible to compare different genetic algorithms based on the complexity of the fitness computation. The overall time complexity to calculate the mean tracking error for a portfolio with β return samples for N assets is $O(\beta N)$. In RDM-GA, it is necessary to calculate a mean tracking error for each portfolio in the population, using b returns samples for N assets. Thus, computing the mean tracking error for each portfolio in the population results in $O(n_I bN)$.

On the other hand, in SDM-SAAGA-GAN, it is necessary to calculate the mean tracking error $|S|$ times for each portfolio in the population, using f return samples for N assets. Therefore, the complexity becomes $O(n_I|S|fN)$. In the case of SDM-SBDGA-GAN, the first step is to calculate $|S|$ mean tracking errors (or objectives) for each portfolio. Next, the algorithm uses the fast non-dominated sorting approach (Deb et al., 2002) to obtain the ranks and compute the $crMetric$ for each portfolio. Thus, the computational complexity for this algorithm is $O(n_I|S|fN + n_I^2|S| + n_I|S|)$, which simplifies to $O(n_I|S|(fN + n_I))$.

Year	1st third	2nd third	Final third
2010	X	X	X
2011	X	X	X
2012	X		
2013	X		
2014			
2015	X	X	X
2016		X	X
2017	X	X	X
2018	X	X	X
2019	X	X	X
2020	X	X	X

Table 3 – Available composition information between 2010 and 2020

3.2 MATERIALS AND METHODS

3.2.1 Dataset

The collected historical closing prices are from the Ibovespa market index and some of its components from 2010-01-12−2021-09-08. To guide the construction of the set of alternatives or universe, it was decided that it would be composed of stocks from the Brazilian market that were included in the theoretical Ibovespa portfolio at least once during this period, and had at most 5 missing values. The Ibovespa composition is rebalanced in a four-month fashion (the first, second, and final third of the year) according to the methodology of B3 (B3, 2021). To satisfy these conditions, tickers of components from the final Ibovespa composition or its composition previews were searched for every third between 2010 and 2020, as shown in Table 3.

Some compositions were not found, especially in 2014 when it was not possible to find any related data. After collecting the candidate tickers, the Alpha Vantage API (Alpha Vantage, 2021) was adopted to search historical price data for each candidate ticker. The data processing steps are presented in Figure 12. After selecting the assets and constraining all the time series for the stocks and the index to the specified period, the collected historical data contained 2884 daily closing prices per time series.

The only preprocessing step was the removal of missing data. Thus, 41 Stocks with at most 5 missing price values were filtered, and the missing prices were associated with at least one of the following five dates: 2017-10-17, 2016-09-12, 2013-12-24, 2013-07-09, and 2012-04-10. These 5 days were removed from all the time series since the missing data consist of a very low percentage of the total data ($\sim 0.2\%$) and they are non-consecutive. The final dataset includes $N = 41$ stocks and the Ibovespa index, which gives a total of 42 time series, and each time series contains 2879 daily closing prices.

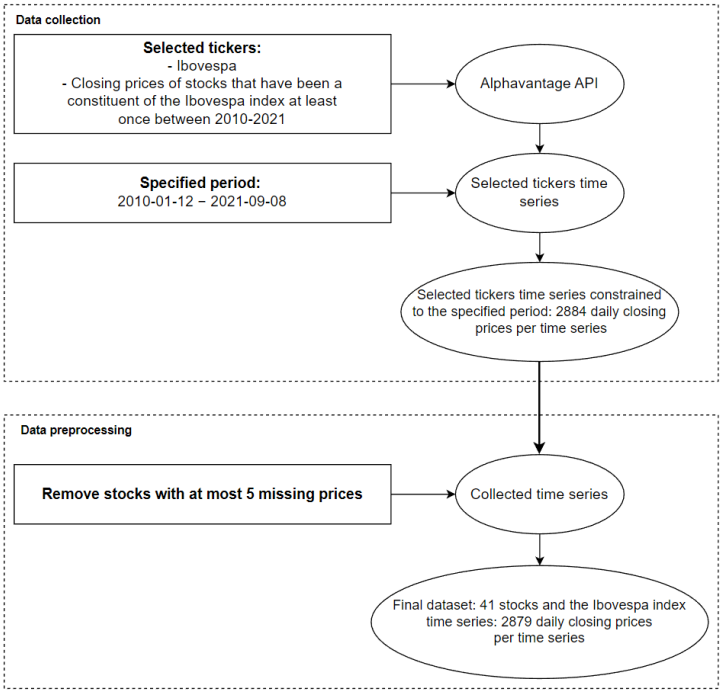


Figure 12 – Data processing steps

3.2.2 Model, rebalancing strategies and GAs hyperparameters

The adopted values for the parameters of the index tracking model were $\epsilon_i = 0$, $\psi_i = 1$ and $K = 10$. As it was already shown in Section 3.1, multi-period portfolio optimization problems were considered since there is a trade-off between transaction costs associated with portfolio rebalance frequency and tracking error (FABOZZI et al., 2007). Of course, the fund reflects the exploration of this trade-off depending on its strategy. Thus, different rebalancing frequency alternatives that could be considered by a fund manager depending on his/her willingness to incur higher transaction costs were investigated.

The following rebalancing frequency alternatives were evaluated: rebalance the portfolio every 10, 20, and 40 days. To evaluate if using an out-of-sample window smaller/bigger than the GAN simulation output would be advantageous, 10 and 40 days rebalancing frequencies were analysed. If it is advantageous it could be concluded that it is possible to train a single type of model for different rebalancing strategies, instead of one model for each rebalancing strategy by varying the horizon f of the GAN's output. For the GAs, the adopted hyperparameters were $n_I = 40$, $n_G = 100$, $p_c = 1.0$, and $p_m = 1/N$.

3.2.3 GAN training setup

As it was already shown in Figure 10, $b = 40$ and $f = 20$ were adopted for historical and simulated data, respectively. The GAN implementation was done in PyTorch (PASZKE et al., 2019). A total of 30 models were trained for 8000 epochs each using CUDA through an NVIDIA GeForce GTX 1650 (4GB). Also, it was considered that $\Delta E = 400$. The

WGAN-GP training process solver and parameters were the ones adopted in Gulrajani et al. (2017). A mini-batch size of 128 was adopted.

The period in which the GAN was trained was from 2010-01-12 until 2020-11-09. This period comprises 2673 daily returns. Since $b = 40$, then the training set contains a total of 2634 matrices of size $(N + 1) \times 40$, Each matrix represents the market state for N assets and the index, considering 40 daily returns of data, where the first row of the matrix represents the return of the index and the remaining rows represent the returns of the N assets. These matrices are stacked to form the training set. Therefore, considering $b = 40$, if $T^{INS} = 40$ returns are available in the training period, then only 1 matrix is contained in the training set. if $T^{INS} = 42$ returns are available in the training period, then 3 matrices are contained on the training set. Similarly, for T^{INS} returns in the training period, $T^{INS} - b + 1$ matrices of size $(N + 1) \times b$ are generated.

The test period was from 2020-11-10 until 2021-09-08, containing a total of 205 daily returns. To ensure that no data from the training period is used, 40 daily returns from the total of 205 are used as input for the neural network at the first time step of the test data. Thus, the out-of-sample period is from 2021-01-11 until 2021-09-08, and the total number of out-of-sample daily returns is $T = 165$. The GAN will generate simulations periodically, depending on the chosen rebalance strategy. If the rebalance strategy is $v = 60$ days, then the simulations will be generated every 60 days. The generated simulations, containing simulated future market states with $f = 20$ daily returns, will be used as input for the index tracking model.

The period from 2010 to 2021 includes various phases of the economic cycle, such as the post-2008 crisis recovery, periods of high volatility, and the COVID-19 crisis. This allows the model to be trained under different market conditions, increasing the robustness of the simulation. The choice of the test period between 2020 and 2021 includes the initial phase of the post-pandemic recovery, where the market was still experiencing significant fluctuations. This is relevant for testing the model's effectiveness in uncertain market conditions.

3.2.4 Experiment design

The adopted experiment design to evaluate the GAN model and the proposed metaheuristics are presented in Figure 13. The first step was to train $m_{MAX} = 30$ GAN models and save their states according to ΔE and the total number of epochs, and thus generate the set Φ. After training all the models, for each rebalancing frequency v, SDM-SAAGA-GAN and SDM-SBDGA-GAN are run for each model (m, e) once. Before each call to one of the metaheuristics designed for SDM, 30 simulations of the market are generated, using the respective M_b of the rebalancing period t, and thus generate S, with $|S| = 30$.

The RDM-GA was run 30 times, which is equal to the number of considered GAN models m_{MAX}, for each v to produce the mean performance it is necessary to evaluate

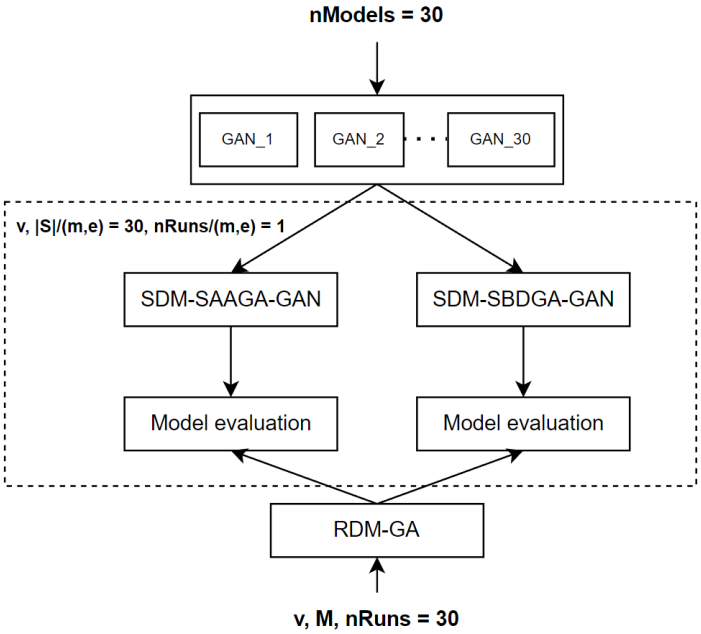


Figure 13 – The adopted experiment design

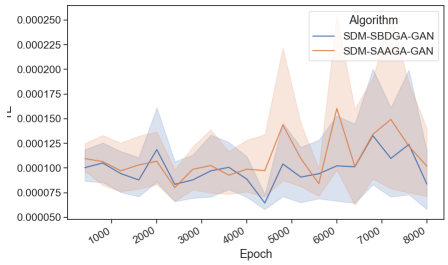
a metaheuristic that solves SDM and produces the set Φ. It is interesting to note that the same information M_b was used to generate the simulations in a rebalancing period t to solve the RDM. Thus, the performance of the metaheuristics is compared given the same amount of information. Finally, for each rebalancing frequency v, one can use the model evaluation algorithm (Algorithm 2), and compare the performance of the SDM metaheuristics against RDM-GA, and also verify which of the two SDM metaheuristics is the best for this problem.

3.3 RESULTS AND DISCUSSION

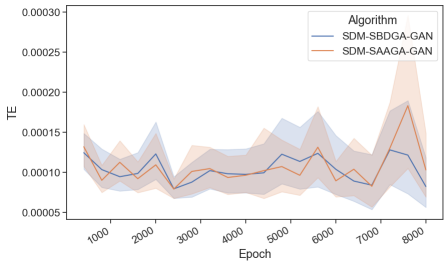
This section begins by presenting the results on the quality of GAN models and the performance of metaheuristics that use the simulations generated from these models. Subsection 3.3.2 presents a comparison between the performance of the GAs that use the simulated data from GAN and the GA that uses historical data. The last subsection brings explanations about why in some situations the simulations generated by GAN are good or bad.

3.3.1 Market simulation using GANs

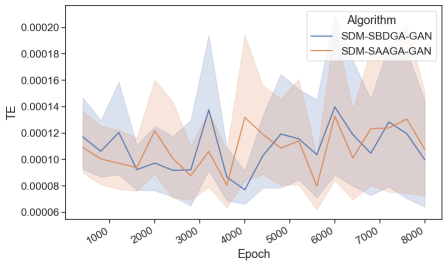
From Figure 14 it can be observed that the instability in the mean overall out-of-sample tracking error produced by the 30 GAN models over the training process. As can be seen, there is no consistent reduction of the mean out-of-sample tracking error as the training epochs get higher for both evolutionary algorithms. But, it was possible to find models (m, e) that obtained better performance than the mean tracking error obtained by the



(a) $v = 10$ days



(b) $v = 20$ days



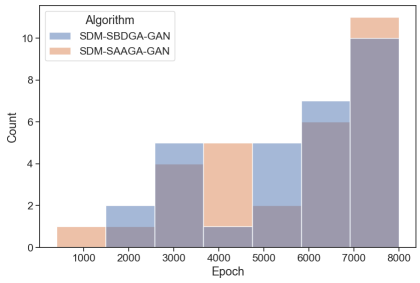
(c) $v = 40$ days

Figure 14 – Performance of the tracking error produced by 30 models for both heuristics over the epochs

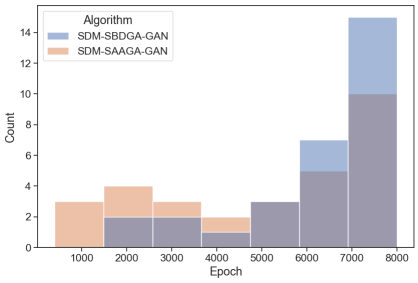
RDM-GA. The next results show the performance of the metaheuristics designed for SDM that used simulations from good quality models $(m, e) \in \Phi$.

From what is presented in Figure 15, the higher the number of epochs the more frequently better models will be produced for both metaheuristics applied to SDM. It is also possible to see, that the SDM-SBDGA-GAN approach produces good quality solutions more frequently than the SDM-SAAGA-GAN approach as the epochs get higher. This indicates that there is a difference in the model state where SDM-SBDGA-GAN can take advantage of simulated data better than SDM-SAAGA-GAN. Although the adopted GAN model was trained for a rebalancing frequency of $v = 20$ days, it can be observed that the results for $v = 40$ and $v = 10$ are very similar concerning the frequency of good solutions over the epochs.

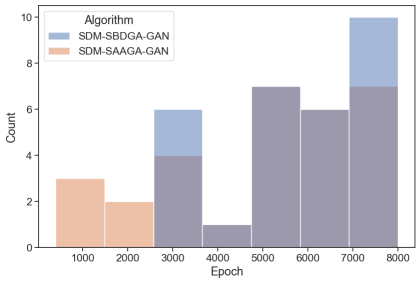
The results presented in Figure 16 concern a density heatmap of the mean overall tracking error obtained by each of the metaheuristics that use simulations from $(m, e) \in \Phi$ over the epochs. It can be observed that smaller tracking errors for both metaheuristics approaches are concentrated in a higher number of epochs, especially for SDM-SBDGA-GAN. This is also true for any of the considered rebalancing frequency values v, but it is more evident in $v = 20$, which is the rebalancing frequency for which the 30 GAN models were trained. The same cannot be said for SDM-SAAGA-GAN when $v = 40$, where the best values for tracking error were concentrated in smaller epoch values. Thus, considering the majority of the cases, as the training epochs get higher, the more frequently smaller tracking errors will be found. The results presented in Figures 15 and 16 show that the



(a) $v = 10$ days

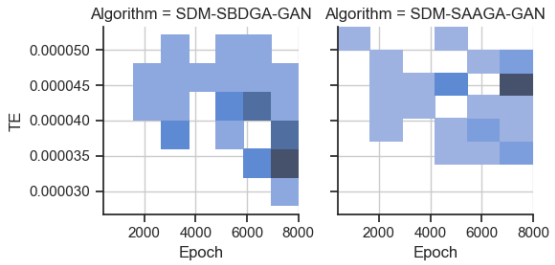


(b) $v = 20$ days

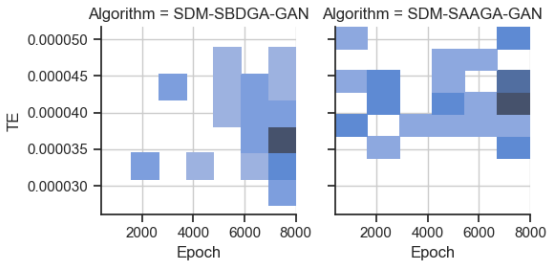


(c) $v = 40$ days

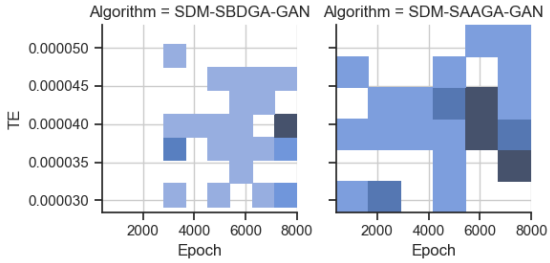
Figure 15 – Frequency of good models relative to RDM-GA for all rebalancing strategies



(a) $v = 10$ days



(b) $v = 20$ days



(c) $v = 40$ days

Figure 16 – Mean overall out-of-sample tracking error over the epochs for all the rebalancing strategies and metaheuristics

models may learn better market patterns over the epochs, and thus can potentially obtain better overall tracking error values over the epochs.

A final comment is that if it was needed to train the GAN in a future period to reflect patterns from new data, it would be more advantageous to sample models between 6000 and 8000 epochs so that it is possible to evaluate the best models and obtain better

DeltaT	Algorithm	$\mu[TE]$	$\sigma[TE]$
10	RDM-GA	4.3e-05	2.1e-04
	SDM-SAAGA-GAN	4.3e-05	5.0e-06
	SDM-SBDGA-GAN	4.1e-05	6.0e-06
20	RDM-GA	4.9e-05	3.1e-04
	SDM-SAAGA-GAN	4.2e-05	4.0e-06
	SDM-SBDGA-GAN	3.8e-05	5.0e-06
40	RDM-GA	4.6e-05	3.2e-04
	SDM-SAAGA-GAN	4.0e-05	6.0e-06
	SDM-SBDGA-GAN	3.9e-05	5.0e-06

Table 4 – Overall tracking error performance for the RDM and SDM metaheuristics in the out-of-sample period.

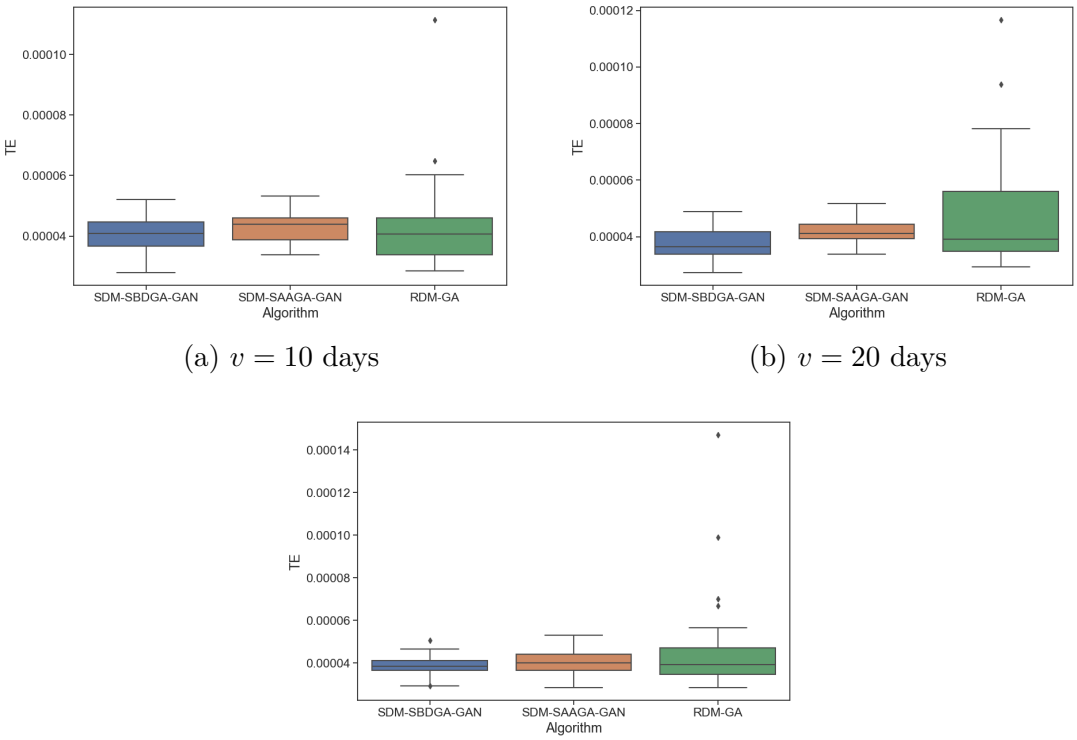
tracking errors when compared to the RDM-GA. Also, although there are models that can perform better than RDM-GA, it would be interesting to train the model for more epochs and check if the performance can be increased or decreased.

3.3.2 SDM vs RDM

Now the impact of using simulated data from the GAN instead of real historical data in the index tracking problem is compared. Thus, the performance of RDM-GA against the SDM heuristics is observed. Table 4 and Figure 18 comprises the results of the models $(m, e) \in \Phi$ with respect to the mean overall out-of-sample tracking error for each metaheuristic that solves SDM in each rebalancing strategy. Table 4 presents the overall tracking error results for each rebalancing frequency and adopted metaheuristics. Figure 18 shows the mean tracking error for SDM-SAAGA-GAN, SDM-SBDGA-GAN, and RDM-GA, obtained in each rebalancing period given a rebalancing frequency v.

In Figure 18, the leftmost figure column is analyzed first. In the Figures of this column, the solid line is the mean out-of-sample tracking error and the area is one standard deviation from the mean. It can be observed that, in the first half of the out-of-sample period, the portfolios constructed with RDM-GA contained assets that couldn't match the index in some periods. Thus, high peaks of tracking errors can be observed for all the rebalancing strategies.

These high peaks were not observed in the performance results of SDM-SAAGA-GAN and SDM-SBDGA-GAN. Thus, the portfolios built using SDM had a more stable performance over the out-of-sample period for all the rebalancing strategies. Now, observing Table 4, comparing the performance of SDM-SBDGA-GAN against SDM-SAAGA-GAN, it is possible to observe that the former was able to build portfolios with better performance for $v = 20$. In $v = 10$ and $v = 40$, it was possible to see just a slight difference between



(a) $v = 10$ days

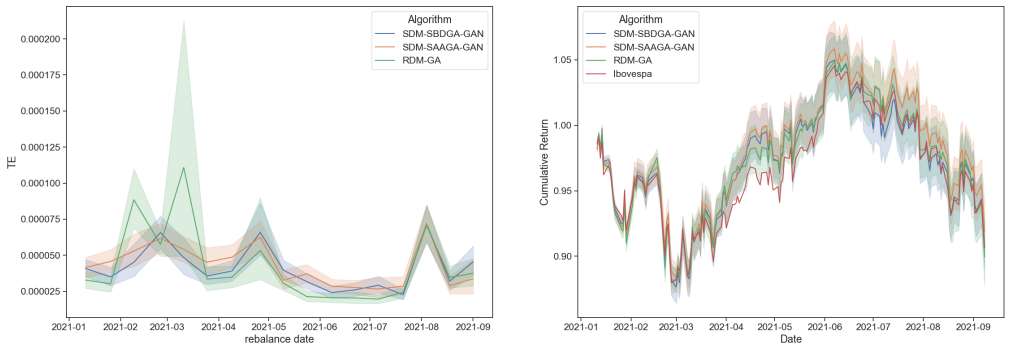
(b) $v = 20$ days

(c) $v = 40$ days

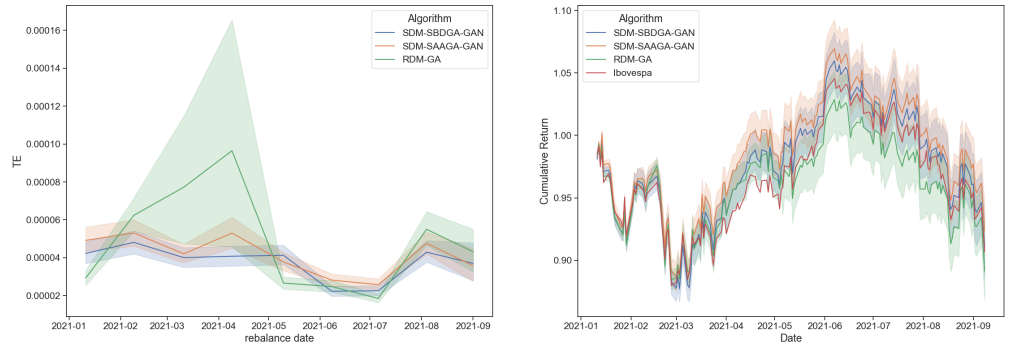
Figure 17 – Boxplots for the overall tracking error performance for the RDM and SDM metaheuristics in the out-of-sample period.

the performance of these two metaheuristics. Also, for $v = 10$ the overall tracking errors of both SDM heuristics are closer to the RDM heuristic performance. This may indicate that SDM performs poorly if the rebalancing frequency is higher than the frequency for which the GAN was trained for. More details about the results can be observed in Figure 17 which contains the boxplots for the overall tracking errors in each rebalancing strategy. The first thing to note is that the dispersion of the overall tracking error produced by SDM-SBDGA-GAN and SDM-SAAGA-GAN was smaller than the RDM-GA. Also, it can be observed that the SDM-SBDGA-GAN can obtain better minimum values than the SDM-SAAGA-GAN in $v = 10$ and $v = 20$. Also, the maximum overall tracking error values of the SDM-SBDGA-GAN are at least as good as SDM-SAAGA-GAN for all rebalancing strategies.

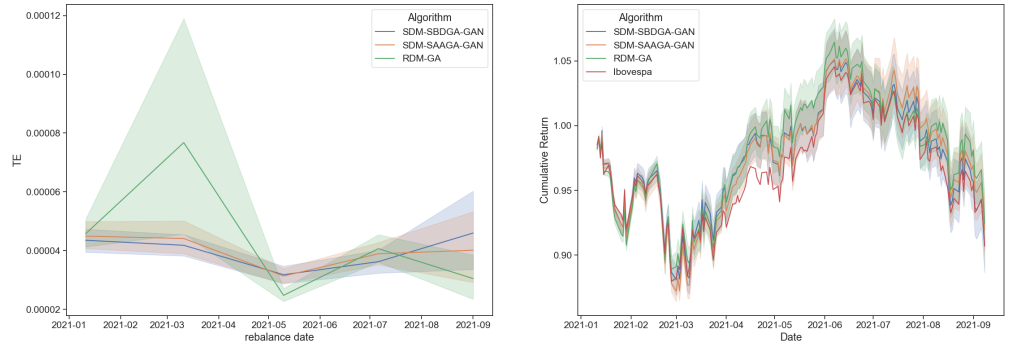
The mean trajectory of the cumulative return was also presented for all the metaheuristics in the rightmost figure column. It can be observed how close SDM-SBDGA-GAN is to the index trajectory when compared to the RDM-GA and SDM-SAAGA-GAN trajectories for $v = 20$ and $v = 40$.



(a) $v = 10$ days



(b) $v = 20$ days



(c) $v = 40$ days

Figure 18 – Comparison of the RDM-GA, SDM-SAAGA-GAN, and SDM-SBDGA-GAN mean out-of-sample tracking error for the respective rebalancing periods (left), and the mean trajectory of the cumulative return (right) for the three rebalancing strategies.

v	Model	Epoch	$\mu[TE]$	t_{best}	t_{worst}
10	GAN-19	7200	2.8e-05	2021-02-09	2021-08-04
20	GAN-26	8000	2.7e-05	2021-06-08	2021-05-10
40	GAN-15	7200	2.9e-05	2021-05-10	2021-03-11

Table 5 – Performance of the best models for each rebalancing strategy with respect to the mean overall out-of-sample tracking error. The respective best and worst rebalancing period is also presented

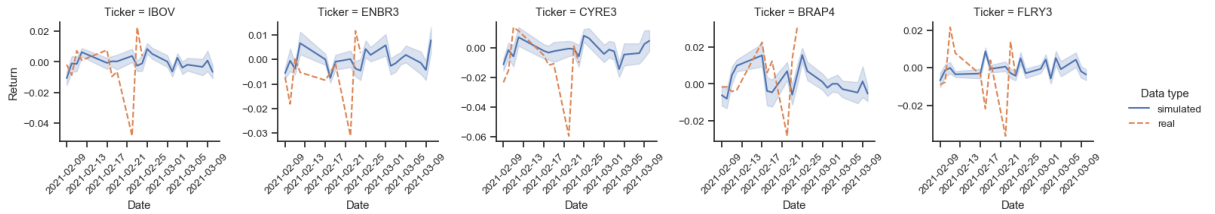
3.3.3 Comparison between simulated and realized data

The next results, presented in Table 5 concern the analysis of the models that offered the best simulations for SDM-SBDGA-GAN in each rebalancing strategy. SDM-SBDGA-GAN was analyzed because it performed well concerning the mean overall out-of-sample tracking error for the main rebalancing strategy ($v = 20$) and slightly better for $v = 10$ and $v = 40$. More specifically, it was necessary to understand why the models performed well or badly in some periods. To do that, for the best model of rebalancing frequency v, the rebalancing period t^v_{best} in which it achieved the best performance and t^v_{worst}, which is the rebalancing period in which it achieved the worst performance, were evaluated. The simulations for the index to be tracked (IBOV) and the top 4 highest weight assets of the portfolio are presented. The results are presented in Figures 19, 20, and 21.

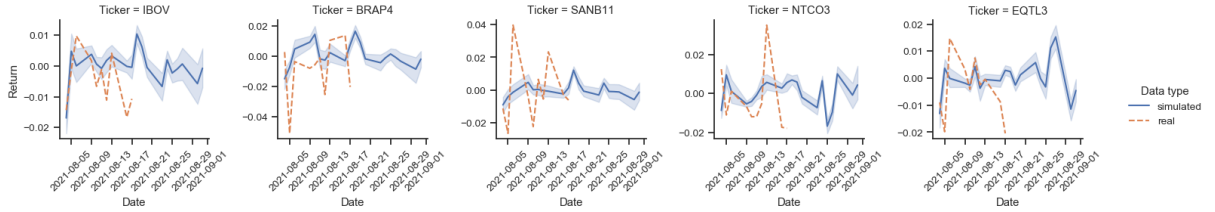
Figure 19 shows the simulations generated in the best and worst period by the best model of rebalancing frequency $v = 10$. Before looking at those results, it is necessary to remember that the SDM metaheuristic constructs the portfolio directed by the simulated index data. Then, the first thing to do is to look at the index data and see if the mean trajectory of the simulated index data presents the same trends as the realized index data. It can be observed that for the best period simulations, shown in Figure 19a, the simulated and realized IBOV data trajectories are very similar at the beginning of the out-of-sample period and diverge in the last periods. During this out-of-sample period, the index varied mostly within a return range of -0.02 and 0.02, which was also the case for the selected assets.

Now, looking at the worst period simulations, presented in Figure 19b, the simulated index mean trajectory does not diverge too much from the realized data. The problem is that the four assets with the highest weight in the portfolio had their data simulated within a return range of -0.02 and 0.02, whilst the realized trajectory of the top 3 assets varied beyond these values. Thus, instead of selecting assets with a lower variance, as was the case for the index trajectory, the metaheuristic selected assets with higher variance because the simulations indicated that these assets had less variance.

Next, the results for the best model of the rebalancing strategy for which GAN was trained are presented in Figure 20. From Figure 20a the simulated index trajectory is



(a) Mean trajectory of the return simulations for the period where the portfolio achieved its best performance



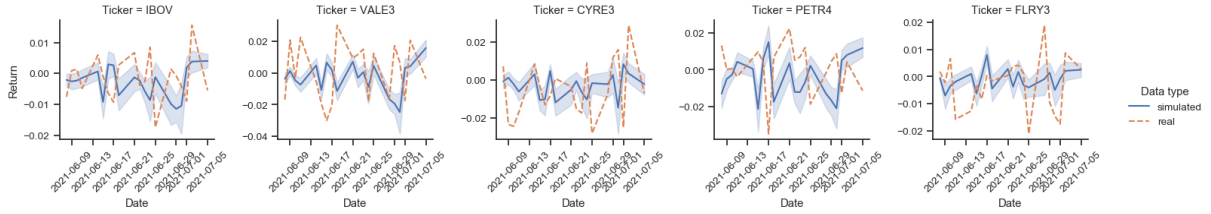
(b) Mean trajectory of the return simulations for the period where the portfolio had its worst performance

Figure 19 – Simulations generated using the best and worst out of sample periods for GAN-19, which is the model that achieved the best $\mu[TE]$ for the rebalancing strategy of 10 days. The mean trajectory of the simulated returns are for the IBOV benchmark and the stocks with the highest price in the portfolio during the rebalancing period

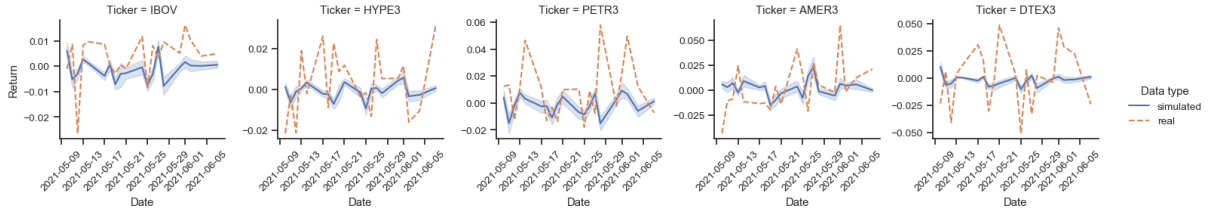
similar to that of the realized data in the best rebalancing period. The selected assets also reflect these. For the results of the worst period simulations, shown in Figure 20b, the simulated trajectory of the IBOV index had a very low variance, which was not the case for the real trajectory. This forced the heuristic to select assets that had a simulated trajectory with very low variance, but the real trajectories of these selected assets were very different from the simulated trajectories impacting the portfolio with high deviations from the index.

The final results are for the rebalancing frequency $v = 40$, presented in Figure 20. Concerning the results for the best rebalancing period, the simulated trajectory of the index does not diverge too much from the real index data, summing to the fact that the selected assets' simulated trajectories also do not diverge too much from their real trajectories, the produced tracking portfolio has good quality. Observing the worst period index simulated trajectory, it diverges from the real trajectory, since the former has a lower variability. As a consequence, the wrong assets were selected, with high amplitude oscillations, and the built portfolio was not able to keep track of the low amplitude oscillations of the real trajectory of the index.

Two problems misguide the construction of good portfolios when considering the studied GAN approach. The first is that even when the index's simulated trajectory is similar to the index's real trajectory, there is no guarantee that the selected assets will produce a good tracking portfolio. That's because the selected assets' simulated data may diverge too much from the real trajectory. The other problem occurs when the index simulations

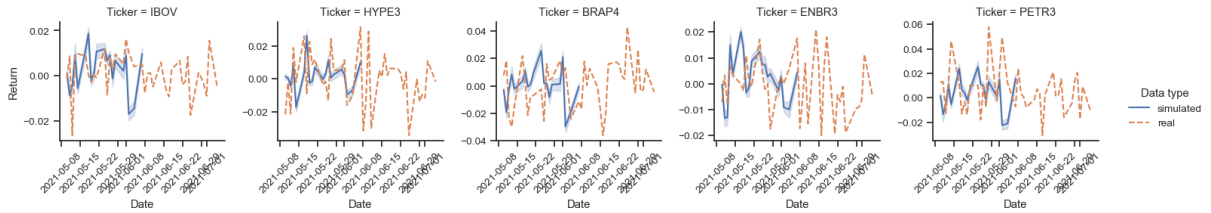


(a) Mean trajectory of the return simulations for the period where the portfolio achieved its best performance

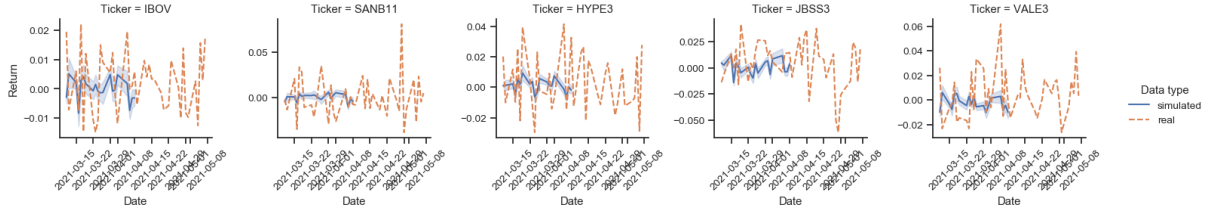


(b) Mean trajectory of the return simulations for the period where the portfolio had its worst performance

Figure 20 – Simulations generated using the best and worst out of sample periods for GAN-26, which is the model that achieved the best $\mu[TE]$ for the rebalancing strategy of 20 days. The mean trajectory of the simulated returns are for the IBOV benchmark and the stocks with the highest price in the portfolio during the rebalancing period



(a) Mean trajectory of the return simulations for the period where the portfolio achieved its best performance



(b) Mean trajectory of the return simulations for the period where the portfolio had its worst performance

Figure 21 – Simulations generated using the best and worst out of sample periods for GAN-15, which is the model that achieved the best $\mu[TE]$ for the rebalancing strategy of 40 days. The mean trajectory of the simulated returns are for the IBOV benchmark and the stocks with the highest price in the portfolio during the rebalancing period

are very poor, diverging too much from their real trajectory. In this case, the portfolio construction will be corrupted from the beginning. The metaheuristic will produce bad portfolios even if there are assets in which their associated simulations would be able to build good tracking portfolios, but since the metaheuristic will be misguided by the poor simulation of the index, they'll never be included in the portfolio. Maybe it is easier to first solve the simulated index data problem since the focus is on a single asset. This could be performed by adding extra GAN inputs concerning the current state information about the index to be tracked or by using a separate prediction model for the index to measure the quality of the index's simulations. This prediction model for the index wouldn't replace the simulations produced by GAN for this asset since they're used in the SDM but would give support in understanding if the simulations are very unrealistic or not.

3.3.4 Extending the application to hybrid heuristics

It is possible to extend other heuristics, such as hybrid heuristics, to solve the multiple-scenario index tracking model based on GANs through the SAA and SBD approaches. The hybrid heuristics for index tracking work in two phases: asset selection and weight allocation. The asset selection is performed through crossover and mutation genetic operators to search for different asset combinations to be included in the portfolio. The weight allocation phase uses general-purpose solvers to adjust the weights for the selected assets considering a nonlinear or linear index tracking model.

Four solution approaches based on Ruiz-Torrubiano & Suarez (2009) (GATOR) and Sant'Anna et al. (2017) (GASAN) hybrid GAs were adopted. For the second phase, Ruiz-Torrubiano & Suarez (2009) quadratic index tracking model (R09) and the Wang et al. Wang et al. (2012) linear index tracking model (W12) were considered. The CPLEX general-purpose solver was used to solve R09 and W12 in the second phase considering the historical data M_b as an instance of the problem.

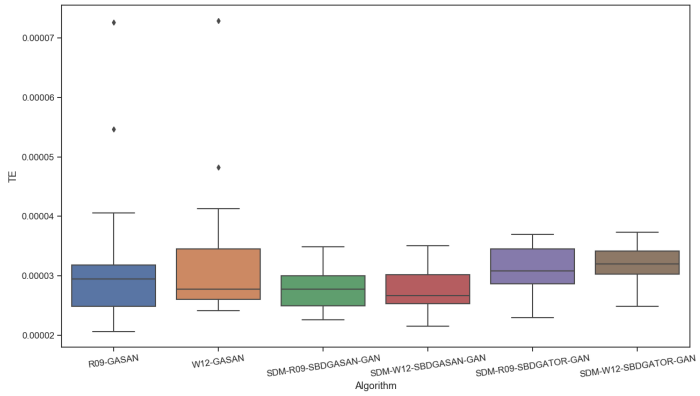
It is possible to extend the heuristics to consider the multiple-scenario optimization by considering the SAA and SDB approaches to evaluate the fitness of the solutions. Thus, it is possible to consider four groups of heuristics, giving 12 heuristics in total:

- **R09-GASAN**: R09-GASAN, SDM-R09-SAAGASAN-GAN, SDM-R09-SBDGASAN-GAN
- **R09-GATOR**: R09-GATOR, SDM-R09-SAAGATOR-GAN, SDM-R09-SBDGATOR-GAN
- **W12-GASAN**: W12-GASAN, SDM-W12-SAAGASAN-GAN, SDM-W12-SBDGASAN-GAN
- **W12-GATOR**: W12-GATOR, SDM-W12-SAAGATOR-GAN, SDM-W12-SBDGATOR-GAN

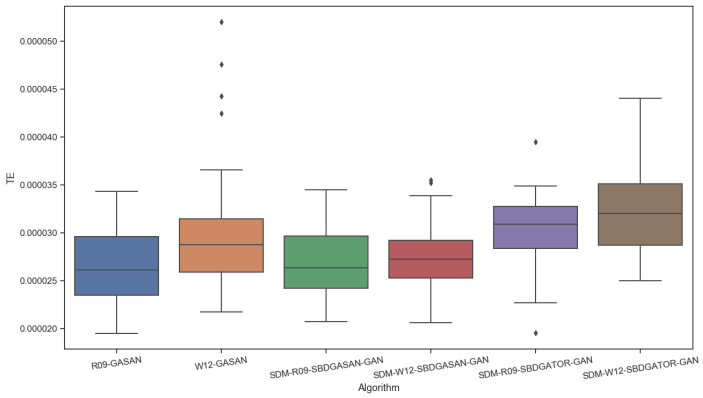
Each group contains one GA where the fitness is calculated based on historical data and two SAA and SBD GAs where the fitness is calculated based on GAN-generated markets. The same hyperparameters defined in Section 3.2.2 were adopted for these hybrid GAs.

As can be seen in Figures 40 - 43, the hybrid GAs based on SBD produces more stable portfolios than GAs based on SAA in the majority of the cases. Thus, the comparison proceeds with hybrid GAs based on historical data and GAs based on SBD. The best hybrid GAs based on historical data for all rebalancing strategies are the R09-GASAN and W12-GASAN as can be seen in Figure 44.

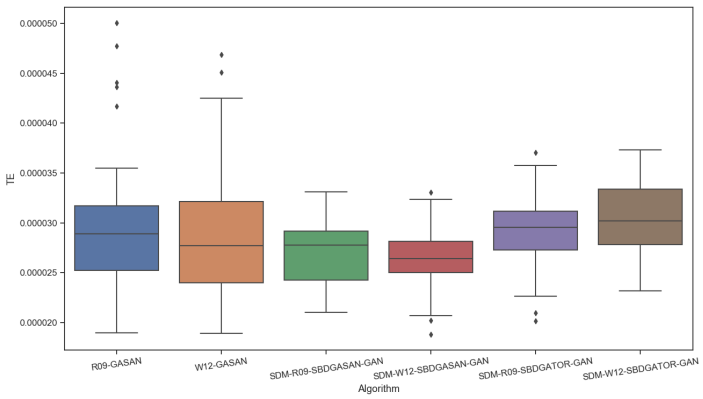
The comparison of the best hybrid GAs based on historical data against the hybrid GAs based on SBD for each rebalancing strategy is presented in Figure 22. It is possible to observe that SDM-R09-SBDGASAN-GAN and SDM-W12-SBDGASAN-GAN can produce more stable portfolios than the other historical data and SBD approaches for $v = 10$ and $v = 40$. Also, as shown in Figure 45, it is possible to observe that SBD hybrid GAs are better than the SDM-SBDGA-GAN.



(a) $v = 10$ days



(b) $v = 20$ days



(c) $v = 40$ days

Figure 22 – Comparison of historical hybrid GAs against SBD hybrid GAs for each rebalancing strategy.

3.4 CONCLUSIONS

This study analyses the construction of tracking portfolios through a GAN based on the WGAN-GP methodology. Two metaheuristics were proposed to optimize an SDM that uses simulated scenarios from GANs to optimize tracking portfolios. Due to the combinatorial nature of the problem and since there is no objective metric to measure the quality of the GAN model, an approach that simply selects the best model, based on the overall tracking error, within a subset of training epochs was proposed. The experiments involved real closing price data from the Ibovespa index and some assets of the Brazilian market between 2010 and 2021.

The experiments showed that GAN models can learn hidden patterns to generate better simulations as the number of epochs grows. It was possible to observe that one can train models to construct portfolios for a specific rebalancing strategy and use these trained models for constructing portfolios in other rebalancing strategies. It was observed that it was possible to obtain good models for lower rebalancing frequencies ($v = 40$) than the original rebalancing frequency for which GAN was trained ($v = 20$). This is important because it may not be necessary to train new models for evaluating other rebalancing strategies.

It was possible to observe that the metaheuristics used to solve the SDM have a better performance because they produce portfolios with more stable tracking errors over the out-of-sample period than the metaheuristic that solves the RDM. Also, when comparing the two SDM metaheuristics, SDM-SAAGA-GAN and SDM-SBDGA-GAN, it was possible to see that SDM-SBDGA-GAN, which considers scenario-based dominance, was better than SDM-SAAGA-GAN. More stable portfolios were also observed when extending the SBD approach to hybrid GAs for the highest and lowest rebalancing frequencies, $v = 10$ and $v = 40$, respectively. It was also possible to draw some discussions about two problems associated with how bad quality simulation data may impact portfolio construction. This study proposes to keep the focus on getting better index simulation data by feeding more input information about the index in GAN or by finding ways to measure the index simulated data quality. These potential solutions will be evaluated in future work.

This work shows the potential that synthetic data generated by GANs can have in index tracking models based on the mathematical programming framework. Thus, it is possible to add more realistic constraints, such as transaction costs, turnover, and others presented in Liagkouras & Metaxiotis (2018), which can be handled by genetic algorithms. Also, to increment the performance of this GAN-based approach to build tracking portfolios, one could first evaluate what would be the best value for b in the RDM-GA, based on the overall mean out-of-sample tracking error, then adjust the architecture of the adopted GAN network to handle this change in the size of the input, before training the GAN model.

One of the main challenges of this study is the way in which the performance of GAN

models is evaluated when applied to combinatorial portfolio optimization models. Thus, the definition of quality metrics for a set of simulations S that outputs a judgement about the capacity of S to produce good portfolios or using time series cluttering reduction approaches (ZHAO et al., 2022) to better visualize the GAN outputs in this context is one direction. The development of better methods to incorporate information associated with simulation produced by GANs in the solvers is another challenge. Another challenge associated with realistic applications of GANs in this context is to propose ways to handle their unstable training process.

Future work involves the evaluation of other formulations for multiple-scenario index tracking, and also for enhanced index tracking (GUASTAROBA; SPERANZA, 2012; YANG; HUANG; HONG, 2023), by evaluating other realistic constraints and objectives. Although the SAA approach was evaluated, there is also the possibility to investigate the performance of the Stochastic Approximation (SA) approach in this problem (MELLO; BAYRAKSAN, 2014). It is possible to use simulated scenarios to increment our previous work on preference learning in the portfolio optimization context (SILVA; FILHO, 2021b). It shouldn't be forgotten that although SDM-SBDGA-GAN performed better than SDM-SAAGA-GAN, the former is more computationally costly than the latter. Thus, proposing new ways to reduce the computational cost of SDM-SBDGA-GAN, while maintaining the quality of its solutions, is a research direction.

Finally, taking into consideration a recent survey of the field (NIKOLENKO, 2019), it would be interesting to evaluate the impact of including synthetic data in the training set, since there is a limited amount of financial assets data. Synthetic data could also be used to increment the training process of reinforcement learning approaches, which have been adopted in investment problems (SCHNAUBELT, 2022; LIN et al., 2022).

4 A SIMULATED IMO-DRSA APPROACH FOR COGNITIVE EFFORT REDUCTION IN THE CLASSICAL MULTIOBJECTIVE PORTFOLIO INTERACTIVE OPTIMIZATION

Obtaining the optimal Pareto front in multiobjective portfolio problems can be impractical when considering real-world constraints, such as portfolio cardinality. Thus, it is interesting to consider multiobjective evolutionary algorithms to solve this type of problem in a reasonable time, but without quality guarantees (SILVA; FILHO, 2021b). IMO-DRSA can reduce the search space using the preference information of the investor until the most preferable solution is found. The problem is that there is no evidence on how to reduce the number of representative portfolios to minimize DM cognitive effort during the interaction, taking the satisfaction of preferences in future distributions of portfolio components returns into account.

The objective of this work was to provide a way to support the reduction of the number of representative examples presented for the investor, while regarding out-of-sample preference satisfaction of the examples that compose a data table. A simulation approach was proposed to analyze and compare methods that select a small and robust sample of representative solutions to compose data tables.

Multiobjective portfolio optimization has been one of the main areas of research in financial MCDM (ZOPOUNIDIS et al., 2015; ALMEIDA-FILHO; SILVA; FERREIRA, 2021). Approaches that incorporate DM's preferences to solve this problem are interesting because the search can be performed in specific regions of the approximate efficient frontier (Purshouse et al., 2014; Yu; Jin; Olhofer, 2019). Recent advances involve the application of Machine learning techniques to learn preference information. Hu et al. (HU et al., 2019) proposed an MCDM approach based on dynamic feature analysis, where decision trees learn preference structures. The authors used functions to model simulated preferences and tested their algorithm capacity to learn the simulated DM's preferences. Mendonça et al. (MENDONçA et al., 2020) proposed a parallel NSGA-II to find the approximate Pareto frontier and solutions contained in the frontier are selected by a neural network. The proposed neural network compares two solutions and returns the best solution reflecting the preferences learned from the DM.

Köksalan and Şakar (KöKSALAN; ŞAKAR, 2016) developed a procedure in which the DM preferences are elicited through an interactive weighted Tchebycheff procedure, where for each iteration the DM chooses from a small set of solutions of a multi-period portfolio optimization problem with 3 criteria. Fernandez et al. (FERNANDEZ et al., 2019) used an interval-based outranking approach to capture uncertainty in the DM's preferences and simulated directly elicited preference parameters. The approach allowed searching for the most suitable portfolio by aggregating all the criteria required by the DM through fuzzy logic

in a bi-criteria optimization problem. Borovicka (BOROVIčKA, 2020) experimented with the construction of portfolios under uncertainty by using an interactive fuzzy multiobjective procedure to handle vague preference information and input data.

In IMO-DRSA, the interaction with the DM is processed using the DRSA, which is a method that learns the DM's preferences with decision examples and structure it using logical rules (GRECO; MATARAZZO; SŁOWIŃSKI, 2008; Purshouse et al., 2014; GRECO; MATARAZZO; SŁOWIŃSKI, 2010). IMO-DRSA have been customized for different problems, such as pavement maintenance (AUGERI; GRECO; NICOLOSI, 2019), space-time models for facility location (BARBATI; CORRENTE; GRECO, 2020), and portfolio optimization (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013). To the best of our knowledge, the work of Greco et al. (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013) was the first to apply IMO-DRSA in the financial portfolio optimization context using the MVO model. The study adopts meaningful quantiles of the candidate portfolio's return distribution to represent uncertainty and extract DM preferences. In this work we address Greco et al. (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013) proposed increment for IMO-DRSA, in the portfolio optimization context, to deal with distributions of returns over time.

4.1 PROPOSED SIMULATION APPROACH

The first step of IMO-DRSA is to generate an initial set S^0 of solutions performing some iterations of the evolutionary algorithm. Then, a data table is generated from S^0 and the DM can perform two actions, one is to choose a solution (stop) and the other is to select a subset of solutions and classify them as 'good' (and begin new interaction). When the DM chooses a unique solution, IMO-DRSA is terminated, otherwise, a rule induction algorithm is applied to the data table. After rule induction, the DM chooses a subset of rules and the evolutionary algorithm performs a search respecting the rule chosen by the DM. Finally, the DM chooses an action or interacts again (the loop starts over) (GRECO; MATARAZZO; SŁOWIŃSKI, 2008). We developed a way to simulate different choices of solutions, rules, and interactions of the DM. Now, the concepts will be presented along with an illustrative example of a bi-objective portfolio optimization problem, where the objective to be minimized is risk (σ) and the objective to be maximized is return (μ_r).

The simulation approach is based on the IMO-DRSA loop. The finite set of outcomes containing all possibilities of DMs and solutions is represented by a tree, shown in Figure 23. The white node S^0 represents the initial set of solutions shown to the DM at the beginning of the first interaction and black nodes are solutions obtained by the evolutionary algorithm after some interaction has been performed.

For instance, S^{E1} is a set of solutions generated by the evolutionary algorithm, where its non-dominated solutions will be presented to the DM at the beginning of interaction 4 after he/she chooses rule AW at the end of interaction 1, rule CY at the end of interaction 2 and rule E1 at the end of interaction 3.

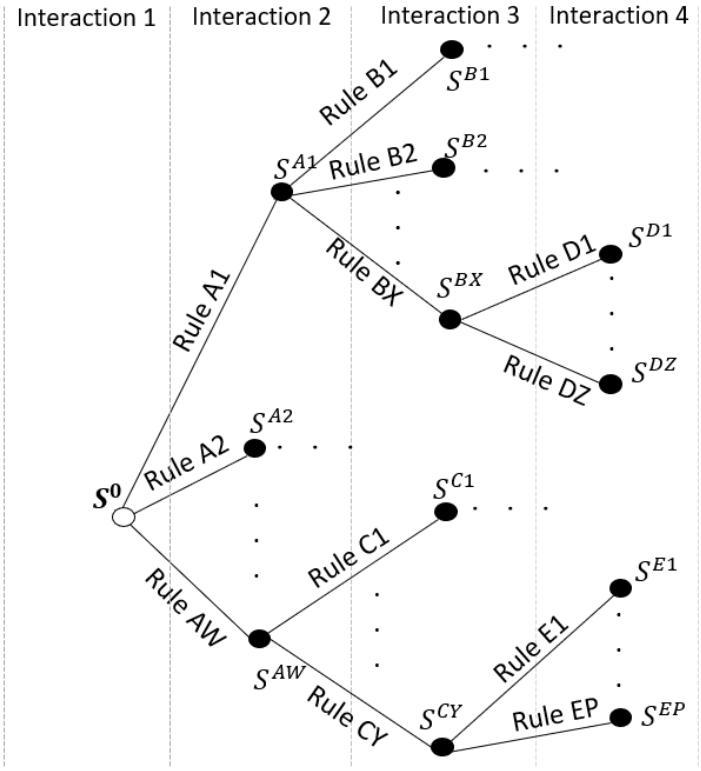


Figure 23 – Proposed DM sampling approach.

A **sample-path** is a path that begins in S^0 and terminates in a chosen non-dominated solution $x \in S^y$ or extends to S^{y+1}. A representative set of non-dominated solutions from S^{y-1} is presented for the DM in the form of a data table DT^y. If the DM chooses a subset of 'good' solutions instead of a unique solution from DT^y, a set of rules is induced and he/she chooses $rule^y$ to represent his/her preferences. The sample-path is defined by its maximum interaction level y_{max}, the sequence of rules (interaction choices) induced according to the type of investor DM_{type} in past interactions, and the final most preferable solution.

Illustrations of sample-paths with $y_{max} = 4$ for two different types of investors are shown in Figure 24. The blue and orange lines could represent risk-averse and risk-prone investor interaction choices, respectively. We can see that for each type of investor and interaction level y, a data table DT^y was generated from $X_{DT^y} \subset S^{y-1}$, a $rule^y$ was induced from DT^y and S^y was generated by the evolutionary algorithm constrained by $rule^y$. Considering the blue line, if $|DT^4| = 15$, then there are 15 sample-paths (or 15 simulated risk-averse investors). The only difference between these 15 sample-paths is the solution choice at the end of interaction 4.

Each S^{y-1} contains $|X_{DT^y}|$ representative non-dominated solutions, which represent $|X_{DT^y}|$ different DMs. Our approach implies supporting the reduction of X_{DT^y}, before showing it to the DM in interaction y. Thus, what we want is to show a small subset of good solutions. When we say good, we mean a portfolio with a high probability of satisfying DM rules (obtained in the last interaction) in the out-of-sample period.

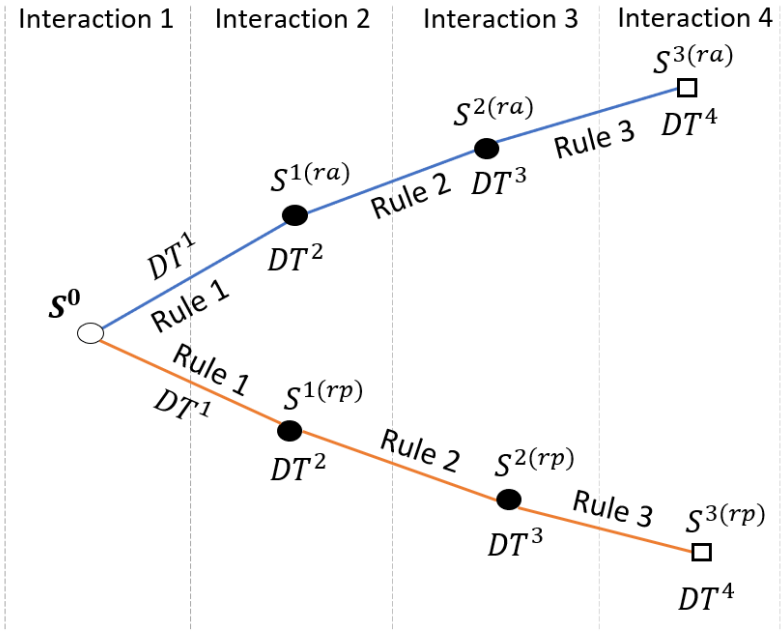


Figure 24 – Two sample-paths originated from S^0.

This study proposed algorithm 6 to enable the generation of branches from each node of the IMO-DRSA process tree presented in Figure 23. Each run of this algorithm implies simulating, for each level of interaction y, unique choices of X_{DT^y}, and $rule^y$, starting from a node S^{y-1} that was not branched yet. Also, this algorithm simultaneously simulates choices of solutions that compose the data table X_{DT^y}, which represent terminations of the IMO-DRSA procedure ($|X_{DT^y}|$ different simulated investors).

Figure 25 illustrates this algorithm generating S^1 starting from S^0, considering a bi-criteria portfolio optimization problem and its associated theoretical true efficient frontier. DT^y elements will be selected and classified by the simulated DM in lines 1-10, where y is the current interaction level, and DT_{size} is the size of the data table. If $DT_{size} = 6$, we could have 2 solutions classified as 'good' and 4 solutions classified as 'others', which gives 6 elements constituting the data table.

If $y = 1$ X is equal to the initial random solution set S^0. Next, a random subset of non-dominated feasible solutions of X, named X_{DT^y}, containing DT_{size} elements is generated by using the $randSubset(X, DT_{size})$ function. We adopted a random subset because we consider that no a priori information of the DM is collected before the initialization of IMO-DRSA.

If $y > 1$, S^{y-1} is assigned to be the initial solution X, which is a node (any black node of the tree presented in Figure 23) that was not branched yet. The function $genDataTable(S^{y-1}, DT_{size})$ is defined by the analyst and uses a data table construction method to select a subset $X_{DT^y} \subset S^{y-1}$ of robust solutions, where $|X_{DT^y}| = DT_{size}$. Finally, if y is not the last interaction, DT^y is generated by classifying elements from X_{DT^y} according to DM_{type} at line 10.

Lines 11 and 12 use DRSA to induce a set of rules from DT^y and sample one rule from

Algorithm 6: simulateInteraction

Input: v_0, DM_{type}, γ, $pars_{evol}$, DT_{size}, y, S^{y-1}, y_{max}
Output: S^y, DT^y, $rule^y$, D

$X = S^{y-1}$
if $y == 1$ **then**
  $X_{DT^y} = randSubset(X, DT_{size})$
else
  $X_{DT^y} = genDataTable(X, DT_{size})$
end if
$dataTable = \{\}$
$rule = \{\}$
if $y < y_{max}$ **then**
  $dataTable = getEval(X_{DT^y}, DM_{type})$
  $rules = getRules(dataTable)$
  $rule = sample(rules, 1)$
  $X^f_{rule} = \{\}$
  while $|X^f_{rule}|/|X| < \gamma$ **do**
    $X = evolutionaryAlgorithm(1, pars_{evol}, rule)$
    $X^f_{rule} = feasible(X, rule)$
  end while
end if
$S^y = X$
$DT^y = dataTable$
$rule^y = rule$
$D = X_{DT^y}$

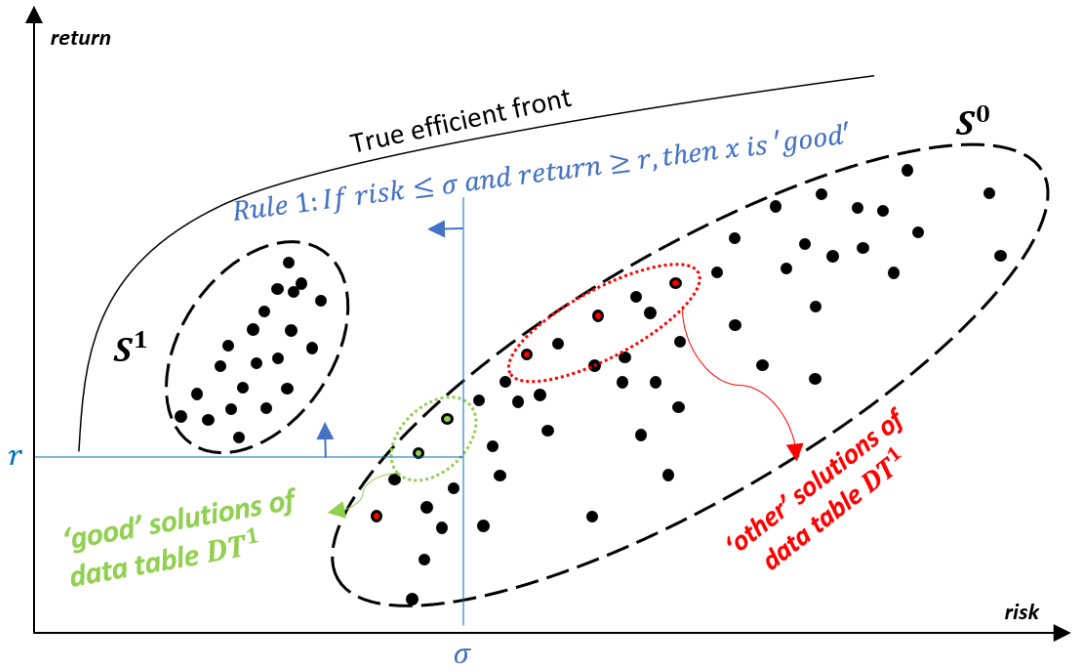


Figure 25 – Generating S^1 using DT^1.

this set, respectively. The set of non-dominated feasible solutions $X^f_{rule} \subset S^y$ is initialized in line 13. The while loop (lines 14-17) runs one generation of the evolutionary algorithm constrained by $rule^y$ until the proportion of non-dominated feasible solutions in S^y is bigger than γ. We use this condition to stop the evolutionary algorithm since we want a good number of non-dominated feasible solutions in S^y. The storage of outputs that will be used to produce a new S^{y+1} in the next interaction occurs at lines 19-21. The simulated DMs, or equivalently, the available choice alternatives X_{DT^y} in data table DT^y, are stored at line 22.

Algorithm 7 was proposed in this work and used to run IMO-DRSA simulations. The following parameters must be specified: v_0, γ, y_{max}, $pars_{evol}$, DT_{size}, DM_{types}, $nRuns$. All these parameters were already presented previously, except $nRuns$, which is the number of different random origins S^0 of the sample-paths. This algorithm writes sample-paths starting from interaction 1 and finishing in interaction y_{max} for all types of investors. Figure 24 shows the sample-paths of two types of investors generated by lines 3-14 of this algorithm.

First, an initial set of feasible solutions $S = S^0$ is generated through v_0 iterations of the multiobjective evolutionary algorithm with its set of parameters $pars_{evol}$. Then, the algorithm starts to generate the path from S^0 for each type of investor. The simulation results can be structured in .txt files stored in paths defined by the DM_{type} and $exec$. In lines 7-14, the function $simulateInteraction()$ is called and returns the results of the interaction, which are stored in the disk.

Algorithm 7: Simulated IMO-DRSA

Input: v_0, γ, y_{max}, $pars_{evol}$, DT_{size}, DM_{types}, $nRuns$
Output: simulated data tables, rules and solution choices

```
1:  exec = 1
2:  for exec ≤ nRuns do
3:     S^0 = evolutionaryAlgorithm(v_0, pars_evol)
4:     for each d ∈ DM_types do
5:        S = S^0
6:        y = 1
7:        for y ≤ y_max do
8:           S, DT, rule, D = simulateInteraction(
9:           d, γ, pars_evol, DT_size, y, S)
10:          write(S, 'd/exec/Sy.txt' )
11:          write(DT, 'd/exec/DTy.txt' )
12:          write(rule, 'd/exec/ruley.txt' )
13:          write(D, 'd/exec/solChoicesy.txt' )
14:          y = y + 1
15:       end for
16:    end for
17:    exec = exec + 1
18: end for
```

4.1.1 Data table generation analysis

The proposed simulation approach can be used to investigate different data table generation methods through the $genDataTable(S^{y-1}, DT_{size})$ function. These methods will assign non-dominated feasible solutions from S^y to the data table DT^{y+1}, such that the size of this data table is small, to reduce cognition, and the alternatives contained in it are robust, in the sense that they will, potentially, fulfill the investor's preferences ($rule^y$) over a good portion of the out-of-sample period. We will demonstrate how this investigation can be performed by using two simple data table generation methods.

First, we will define f_{ref}, a reference point in the multiobjective space that will be used to compute the quality of the non-dominated feasible solutions in both methods The evaluation of a solution $x \in S^y$ in an objective m is given by $f^m(x)$. S^y contains solutions constrained by $rule^y$, and this rule contains a maximum number of M conditions. Each condition defines an upper (Eq. 4.1) or lower (Eq. 4.2) bound constraint on objective m.

$$f^m(x) \leq ub_m \tag{4.1}$$

$$f^m(x) \geq lb_m \tag{4.2}$$

We refer to any of these two types of bound values as B_m. The reference point is defined as follows:

$$f_{ref} = (f^1_{ref}, ..., f^M_{ref}) \tag{4.3}$$

where $f^m_{ref} = \underset{f^m(x)}{\text{argmax}} |f^m(x) - B_m|$, $x \in feasible(S^y)$. $feasible(S^y)$ is the set containing solutions $x \in S^y$ whose objective evaluations satisfy $rule^y$ conditions. This is simply establishing that the reference point will be constituted of each objective value farthest from each rule condition that bounds it.

The reference point is illustrated in Figure 26. In this figure, the solutions are constrained by two rules in a bi-objective portfolio optimization problem. In this illustration we use the term 'auxiliary points' to refer to f^σ and f^r. Now, the data generation methods are presented.

- Method 1 (*closer*): this method chooses the solutions whose objective evaluations are most closer to the reference point f_{ref} to compose the data table DT^{y+1}.
- Method 2 ($farther$): this method chooses the solutions whose objective evaluations are most farther from the reference point f_{ref} to compose the data table DT^{y+1}.

The distance of the solution's objective evaluation $f(x) = (f^1(x), ..., f^M(x))$ to the reference point can be measured by any distance metric, such as an euclidian norm. Solutions are added to $X_{DT^{y+1}}$ until $|X_{DT^{y+1}}| = DT_{size}$.

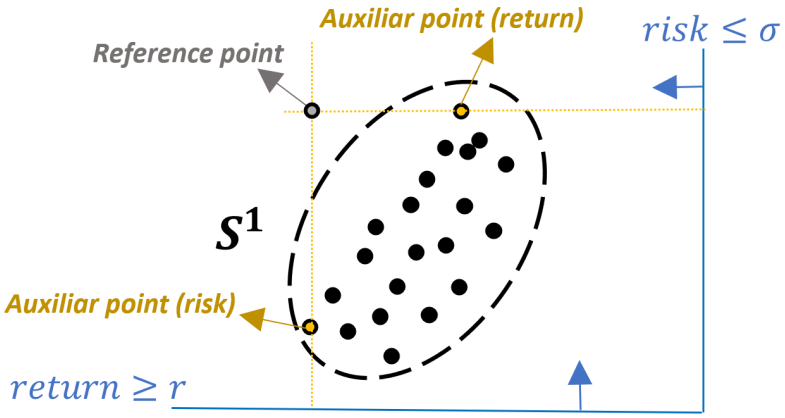


Figure 26 – Reference point computation.

A data table generation method can be evaluated through metrics concerning the data table's solutions performance in the out-of-sample period t. We define two here: mean number of feasible solutions (NMFS_t^y) and mean infeasibility (MI_t^y) at evaluation period t, and interaction level y.

$$\text{NMFS}_t^y = \frac{\sum_{w=1}^{nRuns} |feasible(X_{DT^y,w})|}{nRuns} \tag{4.4}$$

$$\text{MI}_t^y = \frac{\sum_{w=1}^{nRuns} \sum_{x \in X_{DT^y,w} - feasible(X_{DT^y,w})} I(x)}{nRuns} \tag{4.5}$$

The objective of NMFS_t^y is straightforward, since we want to evaluate the method capacity to select robust solutions that satisfy DM's preferences in the out-of-sample period. MI_t^y is used to analyze on average how bad the infeasible solutions of $X_{DT^y,w}$ are, where $I(x) = \sum_{m=1} -|f^m(x) - B_m|$ is the infeasibility of solution x with respect to $rule^{y-1}$. These metrics are only calculated for $y > 1$, since the rules are induced after interaction 1.

4.1.2 Analysis with different types of investors

In this simulation approach, the analyst may also include different types of investors using the $getEval(X_{DT^y}, DM_{type})$ function, which simulates data table classifications. Consider the data table constructed by some of the methods described in subsection 4.1.1 showed in Table 6.

In this data table, $DT_{size} = 6$, and it was required that the simulated investor indicated three 'good' solutions and three 'other' solutions. Two types of investors were considered: the risk-averse investor will always choose the subset of least risky assets, whereas the risk-prone investor will choose the subset of most profitable assets. Columns 4 and 5 of Table 6 contain the result of the classification of the decision examples of this data table performed by simulated risk-averse and risk-prone investors, respectively. The analyst may explore different investors instead of these extreme types.

Table 6 – Classification performed by two types of simulated investors

solution	risk	return	risk-averse	risk-prone
x_1	0.0001	0.002	good	other
x_2	0.0003	0.005	good	other
x_3	0.0004	0.006	good	other
x_4	0.0005	0.010	other	good
x_5	0.0009	0.013	other	good
x_6	0.0012	0.020	other	good

4.2 CASE STUDY: EXPERIMENTAL SETUP

In this section, the simulation approach configurations, the chosen data set containing stock data, the portfolio optimization model, the evolutionary algorithm, and the evaluation metrics used for the case study are presented.

4.2.1 Data and software

Data is from OR-library (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files>), containing 291 weekly price data for the stocks of Nikkei 225 index. We adopted an in-sample period of 240 weeks to perform the optimization and evaluated the portfolios in an out-of-sample period of 50 weeks so that we can calculate the evaluation metrics. Portfolio's risk and return were recalculated in each out-of-sample week. Rules were inducted by the VC-DOMLEM algorithm which is available in software jMAF(BŁASZCZYŃSKI et al., 2013).

4.2.2 Portfolio optimization model

MVO formulations are largely applied in evolutionary portfolio optimization studies, where the most commonly used risk measure is variance. Incorporating practical constraints to MVO, a CCPO model is assembled in its multiobjective form and shown below.

$$\text{minimize} \quad w^T \Omega w \tag{4.6}$$

$$\text{maximize} \quad \mu^T w \tag{4.7}$$

$$\text{subject to} \quad \epsilon Z_i \leq w_i \leq \psi_i Z_i \text{ for each } i \in 1, ..., N \tag{4.8}$$

$$\sum_{i=1}^{N} Z_i = K \tag{4.9}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{4.10}$$

$$Z_i \in \{0, 1\} \tag{4.11}$$

where p_{it} is the price of asset i a time t, where $t \in \{1, ..., T\}$, the return of asset i after d periods is defined as $r_{it} = \ln\left(\frac{p_{it}}{p_{i(t-d)}}\right)$ (CHANG et al., 2000; BEASLEY; MEADE; CHANG, 2003). Threshold/holding/floor-ceiling (4.8) and cardinality (4.9) constraints can be included in the model to reduce undesirable extra transaction costs (FABOZZI et al., 2007), where Z_i are binary variables. $Z_i = 1$ if asset i is included in the portfolio and $Z_i = 0$ if i is not included. Equation (4.8) ensures that if an asset i belongs to the portfolio, then its proportion w_i must lie between ϵ_i and ψ_i, otherwise, if i is not contained in the portfolio, $w_i = 0$. Equation (4.9) restrict the number of assets contained in the portfolio to K. We adopted $\epsilon_i = 0$, $\psi_i = 1$ and $K = 10$.

4.2.3 Multiobjective evolutionary algorithm

We choose **NSGA-II** (Deb et al., 2002), a non-dominated sorting-based multiobjective algorithm, since it is usually adopted to solve constrained multiobjective portfolio selection problems and has competitive performance (LIAGKOURAS; METAXIOTIS, 2018; KALAYCI; ERTENLICE; AKBAY, 2019). The time complexity of NSGA-II is $O(I^2M)$.The hybrid encoding proposed by Streichert (Streichert; Ulmer; Zell, 2004) was used. The weights of a portfolio are represented by a real-valued vector $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ and the included assets are defined by a binary vector $\mathbf{B} = \{Z_1, Z_2, ..., Z_N\}$. The number of individuals adopted was 250. The selected genetic operators are described below. We choose **NSGA-II** (Deb et al., 2002), a non-dominated sorting-based multiobjective algorithm, since it is usually adopted to solve constrained multiobjective portfolio selection problems and has competitive performance (LIAGKOURAS; METAXIOTIS, 2018; KALAYCI; ERTENLICE; AKBAY, 2019). The hybrid encoding proposed by Streichert (Streichert; Ulmer; Zell, 2004) was used. The weights of a portfolio are represented by a real-valued vector $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ and the included assets are defined by a binary vector $\mathbf{B} = \{Z_1, Z_2, ..., Z_N\}$. The number of individuals I adopted was 250. The selected genetic operators are described below.

- **Selection:** Binary tournament selection is the most commonly used selection strategy in the portfolio selection field (LIAGKOURAS; METAXIOTIS, 2018). This operator concerns the competition of two solutions in a tournament for a place in the mating pool. The winner of the tournament is the solution contained in the best non-dominated front, or, if both solutions belong to the same front, the solution with a higher crowding-distance.

- **Crossover:** Uniform crossover was applied to the binary vector, as it was performed in many studies involving cardinality constrained portfolio selection (LIAGKOURAS; METAXIOTIS, 2018). In this operator, a single child is generated from two parents. Assets that are included in both parents will also be present in the child. Assets contained in only one of the parents have a 50% chance of being present in the child.

We adopted $p_C = 1.0$ (ANAGNOSTOPOULOS; MAMANIS, 2010; ANAGNOSTOPOULOS; MAMANIS, 2011).

- **Mutation:** Sant'Anna et. al (SANT'ANNA et al., 2017) bit-flip mutation operator was applied in the binary vector of an individual with probability $p_B = \frac{1}{N}$. It exchange one stock contained in the portfolio by another not contained in the portfolio. A gaussian random mutation with $\sigma = 0.15$ is applied in each decision variable on the real-valued genotype. (ANAGNOSTOPOULOS; MAMANIS, 2011; Streichert; Ulmer; Zell, 2004).

4.2.4 Simulation configurations

The input parameter used for the simulation approach were: $v_0 = 50$, $y_{max} = 3$, $DT_{size} = 6$, $DM_{types} = \{risk-averse, risk-prone\}$, $nRuns = 30$. For y = 1, $\gamma = 0.35$, and for y = 2, $\gamma = 0.05$. It was necessary to decrease γ because it was noted in preliminary tests that as y grows, it takes much more CPU time to get the same proportion of non-dominated feasible solutions of S^{y-1} in S^y. Evolutionary algorithm's $pars_{evol}$ was defined in Subsection 4.2.3.

For simplification, the rules were induced using data tables from the generation Method 1 (closer) only, but the solutions X_{DT^y} were generated using the two methods. Then each call to function simulateInteraction() returned $X_{DT^y} = (X^{closer}_{DT^y}, X^{farther}_{DT^y})$.

4.3 CASE STUDY: RESULTS AND DISCUSSION

The case study results involve two parts, one concerning the comparison between data table generation methods, and another concerning the impact of the interaction level in rule satisfaction over time. In these two parts, the results of both types of investors have also been compared.

4.3.1 Comparison of data table generation methods in the out-of-sample period

Figure 27 shows NMFS^y_t (solid line) and the estimated 95% confidence interval of $|feasible(X_{DT^y,w})|$ for Method 1 (closer) and Method 2 (farther) for the risk-averse investor. It can be observed that the NMFS^y_t of both data generation method is practically the same for this type of investor. The overall performance shows that the mean number of feasible solutions is almost constant from $t = 0$ until $t = 10$ and then decreases. Figure 28 shows MI^y_t (solid line) and the estimated 95% confidence interval of $\sum_{x \in X_{DT^y,w} - feasible(X_{DT^y,w})} I(x)$, and the infeasibility is also equivalent for both methods.

The NMFS^y_t for the risk-averse type, is presented in Figure 29 and the performance for both methods is similar, but the overall performance showed that NMFS^y_t increases over time, in contrast with the risk-averse portfolios. The infeasibility over time is shown in 30 and is also reduced over time. Due to the selection of high-risk stocks by the risk-prone

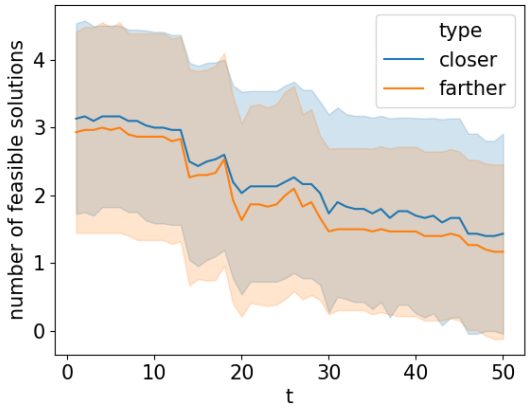


Figure 27 – NMFS_t^y over the out-of-sample period for the risk-averse investor. Solutions obtained with 2 interactions.

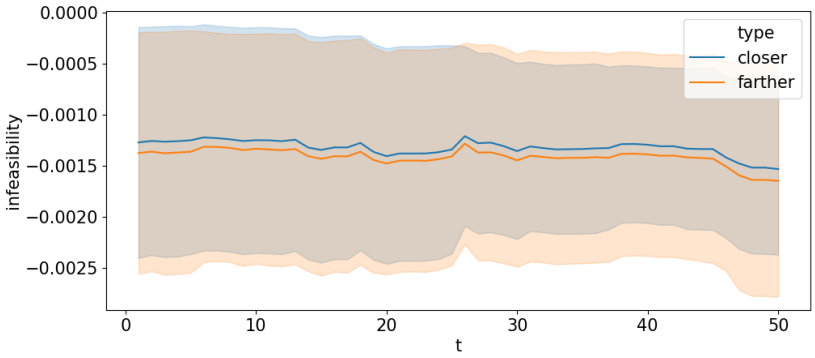


Figure 28 – MI_t^y over the out-of-sample period for the risk-averse investor. Solutions obtained with 2 interactions.

investor, the overall behavior of NMFS_t^y and MI_t^y over the out-of-sample period is more oscillatory in comparison with the risk-averse investor. It can be seen that the selected portfolios' return increased over time and therefore, the mean number of feasible solutions increased since the constraint's bound on return is fixed.

4.3.2 Analysis of the impact of the interaction level on out-of-sample rule satisfaction

The next results concern the following question: what happens with preference satisfaction over time when more interactions are performed? Observing Figure 31, NMFS_t^y is higher and more stable when the interaction level is higher, which shows that, for the risk-averse investor, choosing to interact again and spend more time running the evolutionary algorithm to search for low-risk portfolios provide better rule satisfaction over time. Figure 32 shows that more interaction does not decrease or increase MI_t^y over time in comparison to performing less interaction.

Observing Figure 33, NMFS_t^y behaviour for performing more interaction is similar to

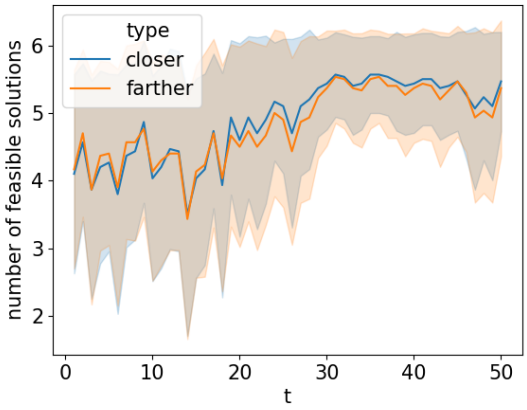


Figure 29 – NMFS_t^y over the out-of-sample period for the risk-prone investor. Solutions obtained with 2 interactions.

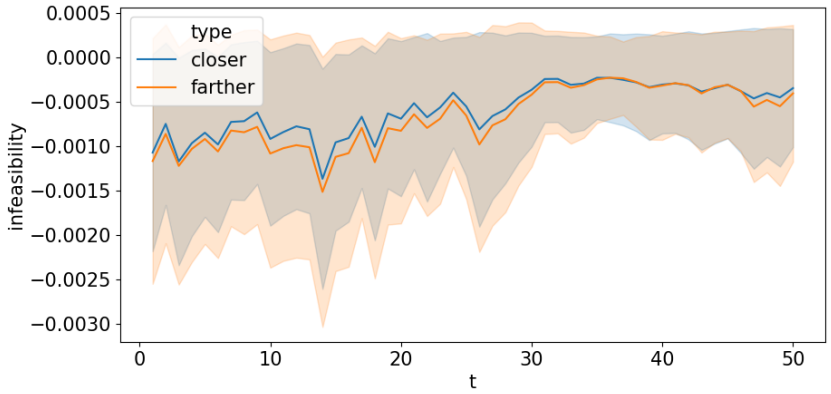


Figure 30 – MI_t^y over the out-of-sample period for the risk-prone investor. Solutions obtained with 2 interactions.

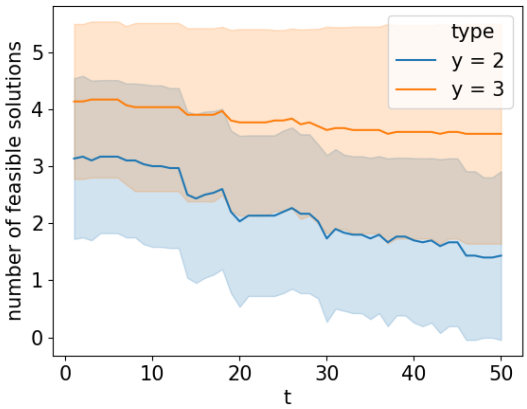


Figure 31 – NMFS_t^y over the out-of-sample period for the risk-averse investor using the Method 1 (closer).

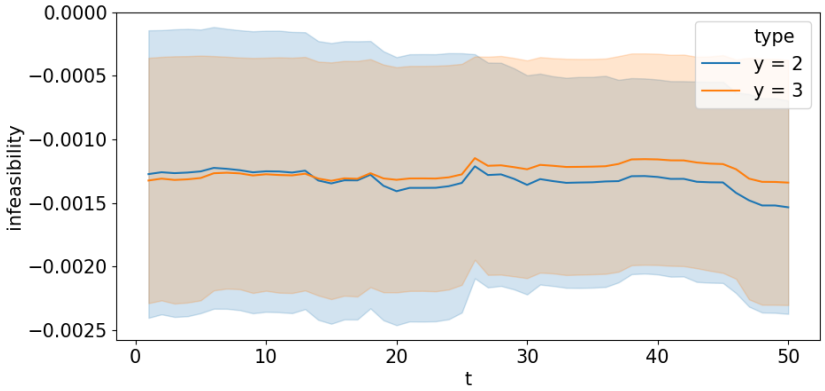


Figure 32 – MI_t^y over the out-of-sample period for the risk-averse investor using the Method 1 (closer).

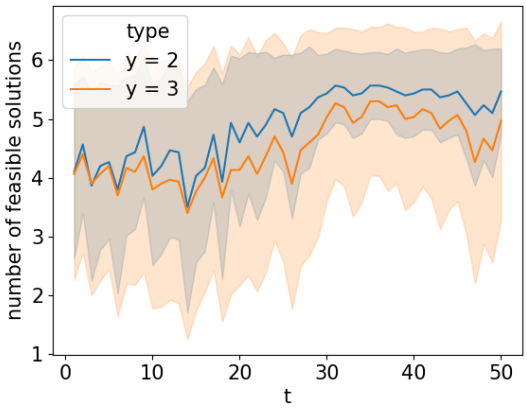


Figure 33 – $NMFS_t^y$ over the out-of-sample period for the risk-prone investor using the Method 1 (closer).

performing less interaction. But, there is a difference concerning the standard deviation of the number of feasible solutions contained in the generated data tables, where more interaction implied a more volatile preference satisfaction. 34 shows the same result for infeasibility, where more interaction is associated with volatile infeasibility. For the risk-prone investor, there exists some uncertainty in the two performance metrics when more interaction is performed.

The results showed that both data table generation methods have similar performance in both metrics. But, when comparing the overall performance of these metrics for different types of investors, we can see that for one type of investor it can decrease over time, and for another type, it may even increase over time. For instance, the mean number of feasible representative solutions of the risk-prone investor gets higher than the risk-averse investors over time. Finally, not all types of investors were benefited from performing more interactions to optimize the considered portfolio selection problem.

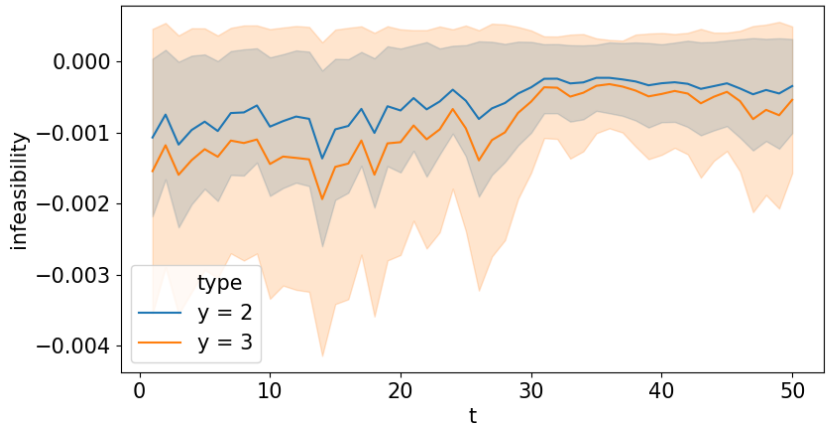


Figure 34 – MI_t^y over the out-of-sample period for the risk-averse investor using the Method 1 (closer).

4.4 CHAPTER CONCLUSION

This chapter proposes a simulation approach to investigate methods that reduce the number of representative solutions that compose the data tables presented to a DM in the IMO-DRSA procedure. The performance of representative solutions is analyzed after some executions of the proposed simulation approach in future distributions of the stock returns after the portfolio optimization phase using IMO-DRSA. It is possible to work with different interaction levels and types of investors.

The case study showed that the simulation approach proposed here can be used to test the performance of methods that select solutions that compose the data table. The comparison is based on the performance of the portfolios selected in the data table generated by each method. Also, one can investigate how the increase in interaction levels can impact constraint satisfaction over time. This is interesting because it permits us to investigate the gains of performing more interactions.

Future work may involve experiments with the use of a subset of rules instead of only one rule per simulated DM to represent its preferences. Another direction is to evaluate the proposed approach in more complex settings of the portfolio selection problem, i.e. more than two objectives. Also, since rule-constrained objectives imply populations containing a high proportion of infeasible solutions as the number of interactions grows, future work involves the search and application of more efficient constraint handling approaches and a comparison between different multi-objective evolutionary algorithms to be hybridized with IMO-DRSA. Another point to be investigated is to measure gains in other variables with experiments on humans when using this approach to select a data table reduction method.

The simulation approach proposed in this chapter enhances the evaluation of IMO-DRSA, enabling an assessment of how the interaction process can be designed. Additional

aspects and experiments are explored in the next chapter.

5 INTERACTIVELY LEARNING ROUGH STRATEGIES THAT DYNAMICALLY SATISFY INVESTOR'S PREFERENCES IN MULTIOBJECTIVE INDEX TRACKING

Multiobjective index tracking models optimize portfolios considering investors' desire to replicate or outperform a market index. Obtaining the optimal Pareto front can be impractical as the index size grows. Thus, it is interesting to consider EMO approaches to find good fronts in a reasonable time. We propose a way to learn a rough approximation of the investor's preference model to guide the EMO search for the most preferable portfolio and to perform preference-driven portfolio updates. This model can be obtained using Interactive Multiobjective Optimization using IMO-DRSA, which is able to guide evolutionary algorithms using a rule-based model that is refined in each interaction with the investor. A simulated IMO-DRSA was adopted and extended to analyse how the number of interactions, criteria considered in the interaction, and methods for cognitive effort reduction affect the capacity of an evolutionary algorithm to produce good portfolios for different types of investors during interactions and to maintain their goodness over time.

Consider a universe of assets of size N and that there are $T+1$ historical prices for each asset, where p_{it} is the price of asset $i = 1, 2.., N$ in time $t = 0, 1, 2, ..., T$. The return of asset i in t is defined as $r_{it} = \frac{p_{it} - p_{i(t-1)}}{p_{i(t-1)}}$. We want to optimize the proportions of a portfolio $w = \{w_1, w_2, ..., w_N\}$ using the available return data in $t = 1, .., T$. The portfolio return in t is given by $R_t^p(w) = \sum_{i=1}^{n} w_i r_{it}$ and R_t^I is the index return in t. The tracking error between the selected portfolio and the index in t is computed as absolute deviations $TE_t = |R_t^p - R_t^I|$ (FILIPPI; GUASTAROBA; SPERANZA, 2016). The excess return of the selected portfolio w.r.t the index in t is given by $ER_t = R_t^p - R_t^I$, and it can measure the fund manager's capacity to beat the target index (FILIPPI; GUASTAROBA; SPERANZA, 2016; BRUNI et al., 2015).

5.0.1 Multiobjective index tracking

Although some multiobjective index tracking research is associated with adding more practical features in TE and/or TE variance minimization, such as the minimization of transaction costs (CHIAM; TAN; MAMUN, 2013; GARCíA; GUIJARRO; MOYA, 2011) and objectives to control the tracking dynamics (LI; BAO; ZHANG, 2014), most of the works focus on *Enhanced index tracking* (EIT), which considers the trade-off between TE and ER. Li, Sun & Bao (2011) proposed a *Multiobjective Immune Algorithm* (MIA) to find the Pareto front of the bi-objective TE-ER model, and selected the solution with the highest ER/TE ratio. Then, the authors applied their MIA approach in a multiobjective EIT with additional objectives to control the time-scale features, extracted with empirical mode

decomposition, of the positive part and negative part of the ER time series (LI; BAO, 2014).

Filippi, Guastaroba & Speranza (2016) proposed a bi-objective kernel search for the TE-ER model and studied how to choose solutions from the front according to different market conditions. Bruni et al. (2015) transformed the trade-off between average ER and its downside risk in a linear model that constructs a portfolio that maximizes ER, respecting the investor's downside risk tolerance. Ni and Wang (NI; WANG, 2013) proposed a model that considers TE and TE volatility, accumulated ER, and combined lexicographic goal programming with a genetic algorithm to obtain portfolios that minimize unwanted deviations from the most important objectives' targets.

Some works considered fuzzy approaches to handle imprecise information. Bilbao-Terol, Arenas-Parra & Cañal-Fernández (2012) proposed a portfolio construction method based on *Fuzzy Goal Programming* (FGP) that handles both the preferences of the investor concerning portfolio performance metrics, ethical investment and environment, and expectations associated with uncertain parameters: future returns and ER. Wu and Tsai (WU; TSAI, 2014) compared three FGP models that consider TE and ER as fuzzy goals, where the preferences of the investor in each objective were represented by a nonlinear exponential membership function. The worst out-of-sample performance was from the max-min FGP and the best was from the additive FGP. Different from past related works, we provide the interactive estimation of a more transparent preference model, without parameter specification requirements, to find the most preferable portfolio.

5.0.2 Rebalancing process of index tracking portfolio

In a buy-and-hold strategy, it is possible to use a *Single-period buy-and-hold strategy* (SBH) where the optimized portfolio's position remains unchanged during all the out-of-sample period or to use a *Multi-period buy-and-hold strategy* (MBH), where the portfolio positions are adjusted to follow the market trends more precisely over time. Many studies adopt the fixed period strategy in MBH, where the market state can be reflected by updating the portfolio more or less frequently (daily, weekly, quarterly, or annually) considering the trade-off between rebalance frequency and transaction costs (SANT'ANNA et al., 2017; BILBAO-TEROL; ARENAS-PARRA; CAñAL-FERNáNDEZ, 2012; BRUNI et al., 2015).

It is also possible to consider a more adaptive strategy in MBH, named event-driven rebalance, which controls the updating frequency depending on the market conditions in a give period. A simple strategy is to use a TE threshold based on in-sample data (CHIAM; TAN; MAMUN, 2013). Another strategy is to use control charts to monitor the portfolio, such as the approach proposed by Sant'Anna et al. (2019), which combines the portfolio's TE and index volatility control charts information to update the portfolio. We propose a preference-driven rebalancing strategy. Since IMO-DRSA learns rules to approximate the investor's preference model, we evaluate if these rules can guide the evolutionary algorithm to maintain or even increase his/her satisfaction in the out-of-sample period.

5.0.3 Multiobjective interactive portfolio optimization

Research on portfolio optimization with MCDM methods includes the development of real-world constraints, risk metrics, and preference learning approaches (KALAYCI; ERTENLICE; AKBAY, 2019; ALMEIDA-FILHO; SILVA; FERREIRA, 2021; ZOPOUNIDIS et al., 2015; FERNANDEZ et al., 2019; SILVA et al., 2021). Preference learning has become a key element in supporting multicriteria decision-making (GRECO; KADZIńSKI, 2018). According to Mousseau & Slowinski (1998), in many cases, it is unrealistic to assume that DMs can define a large number of parameters precisely. Thus, the use of preference disaggregation techniques (JACQUET-LAGRèZE; SISKOS, 2001), incorporating features of mathematical programming and artificial intelligence, to enable systems to learn from decision examples, is present in many variations of multicriteria methods (GRECO; EHRGOTT; FIGUEIRA, 2016).

An approach to assess the capability of a preference learning method in identifying the decision maker's utility with respect to the considered objectives is to use simulated preference functions or simulated decision makers, as demonstrated in recent studies on interactive multiobjective optimization (KADZIńSKI; TOMCZYK; SłOWIńSKI, 2020; HU et al., 2019).

In the classification problem (ROY, 1996), a set of alternatives must be allocated to predefined classes. One can also distinguish between ordinal classification (sorting), nominal classification, and clustering. While in sorting problems, the classes are predefined and ordered in terms of preferences, classification problems deal with predefined but unordered classes, and in clustering problems, the classes are neither predefined nor ordered (OUENNICHE; PÉREZ-GLADISH; BOUSLAH, 2018). In preference disaggregation methods used for sorting, decision examples are allocations of alternatives (real or hypothetical), called references, to the predefined classes. Some examples of methods are UTADIS (PARDALOS; SISKOS; ZOPOUNIDIS, 1995), DRSA (GRECO; MATARAZZO; SLOWINSKI, 2001), the adaptation of ELECTRE-TRI (MOUSSEAU; SLOWINSKI, 1998), and PDTOPSIS-Sort (de Lima Silva; FERREIRA; de Almeida-Filho, 2020).

Different interactive preference learning approaches have been proposed. Interactive fuzzy procedures can obtain Pareto-optimized portfolios by requiring the DMs to express uncertain parameters, such as the relative criteria weights, in each interaction (BOROVIčKA, 2020; MOHEBBI; NAJAFI, 2018). Shen, Lo & Tzeng (2022) used the *Analytical Hierarchy Process* (AHP) to obtain a preference order over the criteria, and interactively relaxed the imprecise fuzzy expectations on less preferable criteria until the DM selected a Pareto-optimal portfolio. Other procedures can extract preference information by pairwise comparison of portfolios. Karakaya and Şakar (KARAKAYA; SAKAR, 2021) used an algorithm to estimate the relative criteria weight to optimize new portfolios, by asking the DM to compare pairs of solutions in each interaction, and tested it in simulated preference functions.

Explainable machine learning approaches have been used to interactively learn rule-based preference structures using information about the classification of representative portfolios. Hu et al. (2019) proposed a rule-based approach based on dynamic feature analysis. In each interaction, a random set of non-dominated portfolios were labelled according to DM preferences, then a random forest was applied to reduce unimportant criteria and the preference structure with the most important criteria was learned by a decision tree. The authors evaluated their algorithm capacity to adapt to different simulated preference functions in two cases: stable and variable preferences. Another type of rule-based approach is DRSA, which assumes that the class labels are ordered.

Let DT be a data table, which is a tuple $< U, Q, V, g >$, where U is a finite set of objects or decision examples, $Q = C \cup D$, where C is the finite set of the conditional attributes, and D is a decision attribute (here we consider a total of $|D|$ classes or groups), V_q is the value set of attribute q, $V = \cup_{q \in Q} V_q$, and the information function $g(x, q)$ is defined as $g : U \times Q \rightarrow V$, such that $g(x, q) \in V_q$ for each $q \in Q$, $x \in U$. It is possible to transform the patterns of the classification of an object into a language that the DM can understand more easily by using a rule induction algorithm (GRECO; MATARAZZO; SŁOWIŃSKI, 2008). The general structure of a decision rule is shown below.

IF $a_{u,1}$ satisfies $h_{r,1}$ and $a_{u,2}$ satisfies $h_{r,2}$ and ... and $a_{u,c}$ satisfies $h_{r,c}$; **THEN** u belongs to d_r.

where $u \in U$, and $h_{r,c}$ is the c-th criteria threshold , where $c \leq |C|$, for rule $r \in R$, which defines the conditional part of the rule, and $d_r \in D$, defines the decision part for rule r. In IMO-DRSA, the interaction with the DM is processed using the DRSA approach, where the induced decision rules will guide the evolutionary algorithm to her/his most preferable region of the search space Ω (GRECO; MATARAZZO; SŁOWIŃSKI, 2008).

IMO-DRSA can learn to guide optimization algorithms to the most preferable feasible solutions by interacting with a decision maker. Let y denote the interaction level, where $y = 1, ..., y_{max}$. In the first interaction, $y = 1$, IMO-DRSA generates an initial set S^0 of non-dominated solutions performing some iterations of the evolutionary algorithm. Then, a data table DT^1, containing DT_{size} examples, is generated from S^0 and the DM can perform two actions, one is to choose a solution (stop) and the other is to select a subset of solutions and classify them as 'good' (and begin new interaction).

To the best of our knowledge, the work of Salvatore, Matarazzo & Słowiński (2013) was the first to use IMO-DRSA in the financial portfolio optimization context using the MVO. The study adopts meaningful quantiles of the candidate portfolio's return distribution to extract DM preferences based on stochastic dominance. Most interactive portfolio optimization approaches did not consider combinatorial portfolio optimization problems, which require the use of heuristics, such as evolutionary algorithms. Silva and de

Almeida-Filho (SILVA; FILHO, 2021b) studied the interactive optimization of an NP-hard portfolio optimization problem. A simulated IMO-DRSA was applied to the cardinality constrained MVO, using NSGA-II, to evaluate the out-of-sample behaviour of portfolios produced by different investor types, the number of interactions, and frontier filters. This work expands (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013; SILVA; FILHO, 2021b) by generalizing the application of IMO-DRSA in the combinatorial portfolio optimization context, considering more instances, more objectives, and studying the influence of different factors in the evolutionary algorithm performance in the in-sample and out-of-sample period w.r.t. preference satisfaction.

5.1 SIMULATED IMO-DRSA TO BUILD TRACKING PORTFOLIOS

Let x^y and $Rules^y$ denote, respectively, the chosen solution and the induced rule set at iteration y. When the DM chooses a unique solution x^1, IMO-DRSA is terminated. Otherwise, a rule induction algorithm is applied to the data table and generates a rule set $Rules^1$. After the induction of the rule set, the DM chooses a rule that is most representative w.r.t. its preferences to guide the evolutionary algorithm. The selected rule becomes a constraint in the search space Ω. Then, for $y = 2, ..., y_{max}$, a rule from $Rules^{y-1}$ will guide the evolutionary algorithm to search S^{y-1}, until a condition is met, which will generate DT^y. From DT^y, x^y or $Rules^y$ can be obtained.

An overview of the simulated IMO-DRSA approach is presented in Figure 35, and it's used to evaluate the impact of different factors on the robustness of the solutions w.r.t. the investor's preferences, such as the number of interactions, the type of investor and approaches to reduce the cognitive effort (SILVA; FILHO, 2021b). The simulation parameters are the size of the data table DT_{size}, the maximum number of interactions y_{max}, and the maximum runtime of the evolutionary algorithm τ^{max}. As was already discussed, in IMO-DRSA a decision maker can choose two options: choose a unique solution x^y or continue the interaction by sorting the solutions and choosing a rule r^y from the induced rule set $Rules^y$. Thus, to save computational time, our simulation process considers that, in each interaction $y = 1, ..., y_{max}$, the simulated investor chooses these two options simultaneously. Also, r^{y-1} is randomly sampled from $Rules^{y-1}$ to guide the search of the evolutionary algorithm, whenever a new interaction y begins, for $y = 2, 3, ..., y_{max}$.

In this simulation approach, we can adopt some methods that can contribute to the reduction of the cognitive effort, which is the case of the frontier filter, or the complexity of the problem solved by the evolutionary algorithm, which is the case of the data table generation method. The frontier filter is applied in a solution S^y, for $y > 2$, to reduce the size of the non-dominated frontier by creating a small subset of non-dominated solutions $X_{DT^y} \subset S^y$, where $|X_{DT^y}| = DT_{size}$. The data table generation method will use the in-sample information from portfolios contained in $|X_{DT^y}|$ to compute a set of criteria that will be used to generate a data table DT^y. Thus, the constraints of the problem's search

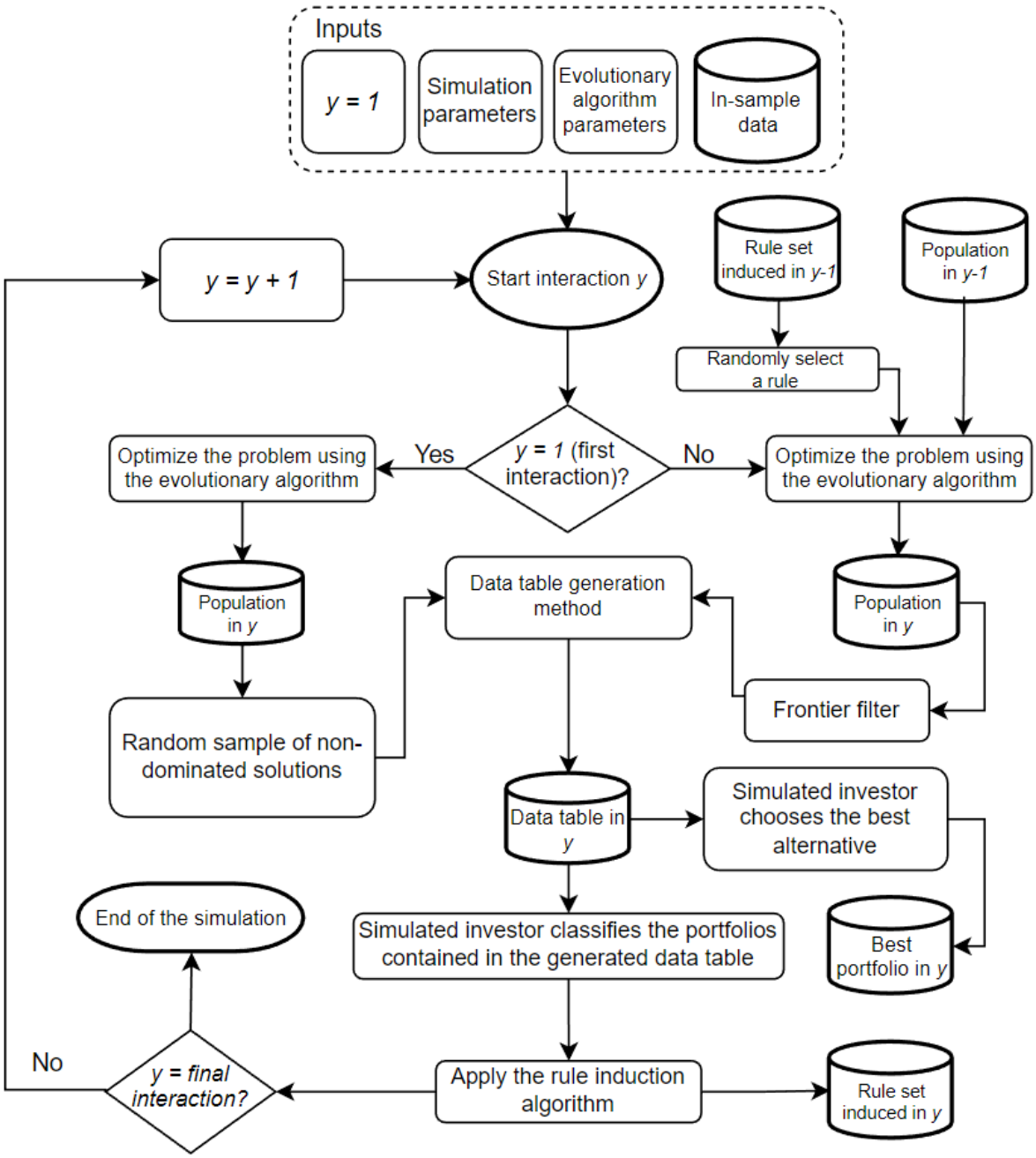


Figure 35 – The simulated IMO-DRSA approach

space will depend on the data table generation method. Next, we present the adopted multiobjective index tracking model, frontier filters, preference functions of the simulated investors, and the data table generation methods.

5.1.1 Multiobjective index tracking model

Although many other objectives can be considered, such as transaction costs, ESG, annualized return and volatility (SILVA; FILHO, 2023; CHIAM; TAN; MAMUN, 2013; BILBAO-TEROL; ARENAS-PARRA; CANAL-FERNANDEZ, 2012), to generalize the application, we propose a model that reflects the TE-ER trade-off since these types of models are most

common in the multiobjective index tracking literature.

The proposed multiobjective index tracking model is constituted by equations (5.1)-(5.8). This model is based on the model of Filippi, Guastaroba & Speranza (2016) that reflects the tradeoff between TE and ER, but also considers the minimization of the downside risk measure adopted in Bruni et al. (2015) and of the worst-case TE. These two last objectives were considered to quantify and minimize the portfolio's worst-case scenario for both TE and ER.

$$\min_{w} \quad \mu(TE) = \frac{1}{T}\sum_{t=1}^{T} TE_t \tag{5.1}$$

$$\min_{w} \quad \max_{t} TE_t \tag{5.2}$$

$$\max_{w} \quad \mu(ER) = \frac{1}{T}\sum_{t=1}^{T} ER_t \tag{5.3}$$

$$\min_{w} \quad \max_{t} (-ER_t) \tag{5.4}$$

$$\text{subject to} \quad \epsilon Z_i \leq w_i \leq \psi_i Z_i \text{ for each } i \in 1, ..., N \tag{5.5}$$

$$\sum_{i=1}^{N} Z_i = K \tag{5.6}$$

$$\sum_{i=1}^{N} w_i = 1 \tag{5.7}$$

$$Z_i \in \{0, 1\} \tag{5.8}$$

where the first objective (5.1) is the minimization of TE and the third objective (5.3) is the maximization of ER. Also, we consider the minimization of a downside risk measure (BRUNI et al., 2015) for ER and the worst-case TE, in objectives two (5.2) and four (5.4), respectively. The worst-case TE represents the maximum deviation w.r.t. the index, and the downside risk represents the maximum underperformance w.r.t. the index. The cardinality constraint (5.6) limits the size of the portfolio to K assets, where Z_i are binary variables. $Z_i = 1$ if asset i is included in the portfolio and $Z_i = 0$ if i is not included. Threshold/holding/floor-ceiling (5.5) constraints can be included in the model to reduce undesirable extra transaction costs (FABOZZI et al., 2007). Equation (5.5) ensures that if an asset i belongs to the portfolio, then its proportion w_i must lie between ϵ_i and ψ_i, otherwise, if i is not contained in the portfolio, $w_i = 0$. We adopted $\epsilon_i = 0$, $\psi_i = 1$ and $K = 10$.

5.1.2 Frontier filters

The proposed simulation approach can be used to investigate different frontier filters. For a given interaction $y = 2, ..., y_{max}$, these filters use the information contained in r^y to sample DT_{size} non-dominated feasible solutions from S^y to reduce the cognitive effort of the investor. It is important to note that the way the solutions are sampled from S^y will influence the robustness of x^{y+1}, w.r.t. the fulfilment of the investor's preferences over a good portion of the out-of-sample period. In this work, we investigate the performance of two frontier filters that were proposed in Silva and de Almeida-Filho (SILVA; FILHO, 2021b) for constraints that bound the objective space.

In this work, we consider constraints associated with the set of criteria C used in a DT. Consider a reference point g^{ref} in the search space Ω that will be used to compute the quality of the non-dominated feasible solutions in both methods. In an interaction $y+1$, S^y contains solutions constrained by r^y, which was randomly sampled from $Rules^y$. The conditions of r^y define upper or lower bounds on a subset of criteria $C^y \subseteq C$. The violation or infeasibility of a solution $x \in S^y$ w.r.t. $q' \in C^y$ is given by Equation 5.9.

$$I_{q'}(x) = \begin{cases} g(x, q') - h_{r^y, q'} & \text{if lower bound } (\geq) \\ h_{r^y, q'} - g(x, q') & \text{if upper bound } (\leq) \end{cases} \tag{5.9}$$

where $h_{r^y,q'}$ is the threshold of r^y for criterion q', and $g(x, q)$ is the evaluation of $x \in S^y$ in criterion q'. The total violation is given by $I(x) = \sum_{q' \in C^y} I_{q'}(x)\beta_{q'}(x)$, where $\beta_{q'}(x) = 1$ if $I_{q'}(x) \geq 0$, otherwise $\beta_{q'}(x) = 0$. If $I(x) = 0$, then $x \in S^y_{rule}$, otherwise, $x \notin S^y_{rule}$, where S^y_{rule} is the set containing solutions $x \in S^y$ that satisfy r^y conditions. The reference point is defined as follows:

$$g^{ref} = (g^{ref}_{q'_1}, ..., g^{ref}_{q'_{|C^y|}}) \tag{5.10}$$

where $g^{ref}_{q'} = \underset{g(u,q')}{\text{argmax}} |g(u, q') - h_{r^y,q'}|$ and $u \in S^y_{rule}$. This is simply establishing that the reference point will be constituted of each criterion q value farthest from each rule condition that bounds it.

- Method 1 (*closer*): this method chooses the solutions whose objective evaluations are closer to the reference point g^{ref} to compose the data table DT^y.
- Method 2 ($farther$): this method chooses the solutions whose objective evaluations are farther from the reference point g^{ref} to compose the data table DT^y.

We applied the euclidian distance between a solution evaluation on C^y, $g(x, C^y) = (g(x, q'_1), ..., g(x, q'_{|C^y|}))$, and the reference point g^{ref}. Solutions are added to $X_{DT^{y+1}}$ until $|X_{DT^{y+1}}| = DT_{size}$. Finally, by applying frontier filters, it is possible to exploit the preference

Table 7 – Calculation of the preference function of the simulated investors in six alternatives.

x	$\mu(TE)$	$max(TE)$	$\mu(ER)$	$max(-ER)$	$f(x,\theta^{TE})$	$f(x,\theta^{ER})$	$f(x,\theta^{EQ})$
x_1	0.0078	0.0263	0.0010	0.0181	0.0509(**2**)	0.5994(5)	0.2545(**2**)
x_2	0.0141	0.0465	0.0035	0.0347	0.0517(6)	0.5979(**1**)	0.2587(6)
x_3	0.0059	0.0221	-0.0013	0.0182	0.0509(**3**)	0.6008(6)	0.2545(**3**)
x_4	0.0115	0.0519	0.0033	0.0307	0.0515(4)	0.5980(**2**)	0.2577(4)
x_5	0.0075	0.0187	0.0010	0.0171	0.0508(**1**)	0.5994(4)	0.2543(**1**)
x_6	0.0086	0.0318	0.0020	0.0318	0.0516(5)	0.5988(**3**)	0.2579(5)

information of the investor in the out-of-sample period when considering an MBH strategy. The frontier filter is applied to the non-dominated frontier according to the rule induced for a given interaction level and the best solution according to the filter is chosen. We select the farthest solution to g_h^{ref} and the closest solution to g^{ref} if the applied filter is farther and closer, respectively.

5.1.3 Preference function for simulated investors

In this simulation approach, one can consider different types of investors using preference functions. Some works consider using a vector of weights $\theta = \{\theta_1, \theta_2, ..., \theta_M\}$ to reflect the trade-off between the objectives in a preference function (i.e. weighted sum or archiving scalarizing function) of a real or simulated decision-maker where the solution methods under study don't know this vector of weights a priori (KADZIńSKI; TOMCZYK; SŁOWIńSKI, 2020; FERREIRA et al., 2018; TOMCZYK; KADZIńSKI, 2020). We considered that the preference of an investor is modelled as a Chebyshev function, as shown in Equation (5.11). We assume that the simulated DM knows the problem very well and can precisely specify the weight of each criterion.

$$f(x,\theta^{DM}) = \max_{m=1,...,M}\{\theta_m | f_m(x) - \phi_m|\} \tag{5.11}$$

where ϕ is an ideal point defined by the DM. Here we adopted $\phi = \{0, 0, 1, -1\}$. Considering two solutions x_1 and x_2, if $f(x_1,\theta^{DM}) < f(x_2,\theta^{DM})$, then x_1 is preferred to x_2.

We consider three investors: one that is more prone to follow the index with $\theta^{TE} = \{0.6, 0.3, 0.05, 0.05\}$, one more prone to beat the index $\theta^{ER} = \{0.05, 0.05, 0.6, 0.3\}$, and another one that balances all the objectives $\theta^{EQ} = \{0.25, 0.25, 0.25, 0.25\}$. Table 7 shows how these simulated investors would evaluate some portfolios and Table 8 shows the final classification. The simulated investors classify the three best solutions, according to their preference function, as 'good' and the remaining solutions as 'other'.

Table 8 – Classification performed by the simulated investors

x	θ^{TE}	θ^{ER}	θ^{EQ}
x_1	good	other	good
x_2	other	good	other
x_3	good	other	good
x_4	other	good	other
x_5	good	other	good
x_6	other	good	other

5.1.4 Data table generation methods

Data table generation methods help the DM sort portfolios as 'good' or 'other' by presenting information in various ways. The resulting data table includes portfolio information and DM preferences for rule induction. Different methods produce distinct constraints for the model, affecting the solver's performance. An artificial DM classifies portfolios using its preference function, regardless of the data table generation method, as shown in section 5.1.3.

5.1.4.1 Stochastic dominance approach

The mean-variance approach proposed by Salvatore, Matarazzo & Słowiński (2013) adopts DRSA for decisions under uncertainty, which was derived from the stochastic dominance definition (GRECO; MATARAZZO; ROMAN, 2010). In this approach, a set of portfolios is evaluated on M criteria, where some criteria can be random variables. Thus, $g_m : S \to \mathbb{P}$, where $\mathbb{P}$ is a set of probability distributions. It is possible to consider a set of meaningful probability levels $\Pi = \{1\%, 25\%, 50\%, 75\%, 99\%\}$ and the loss (quantile) associated with each level.

In (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013) it was assumed that the returns of portfolios are normally distributed. Thus, we can compute the quantile of the portfolio return distribution corresponding to $p \in \Pi$ using the first two moments of the return distribution. Thus, we consider that TE and ER are normally distributed as a baseline, which we call the 'par_quant' data table generation approach. In this approach, we can consider, for instance, the maximum deviation w.r.t the index with 99% of probability, corresponding to $p = 99\%$, which is computed as $TE(w)_{1\%} \approx \mu(TE(w)) + 2.33\sigma(TE(w))$.

A data table that is generated by this approach, using the same portfolios from Table 7, is presented in Table 9. In DRSA, the preference direction of TE and ER associated quantiles will be considered as cost (-) and gain (+) attributes, respectively. To reduce the number of criteria in the data tables within this approach we considered the following meaningful probability levels $\Pi = \{1\%, 50\%, 99\%\}$. For TE we only considered the levels associated with the right tail of the distribution, which are 1% and 50%. This is because

Table 9 – Data table generated using the 'par_quant' method

x	$TE^{(-)}_{1\%}$	$TE^{(-)}_{50\%}$	$ER^{(+)}_{1\%}$	$ER^{(+)}_{50\%}$	$ER^{(+)}_{99\%}$
x_1	0.0219	0.0078	0.0239	0.0010	-0.0219
x_2	0.0380	0.0141	0.0433	0.0035	-0.0363
x_3	0.0161	0.0059	0.0156	-0.0013	-0.0181
x_4	0.0315	0.0115	0.0358	0.0033	-0.0292
x_5	0.0193	0.0075	0.0220	0.0010	-0.0199
x_6	0.0261	0.0087	0.0282	0.0020	-0.0243

Table 10 – Rules generated for simulated investors using the 'par_quant' data table presentation method

θ	Rule set
TE and EQ	(TE1 <= 0.0219) => (d >= good) (TE1 >= 0.0261) => (d <= other)
ER	(ER1 >= 0.0282) => (d >= good) (ER1 <= 0.0239) => (d <= other)

the 99% level may produce negative values, which would be not adequate since TE is nonnegative. Table 10 shows the rules induced using information from the generated data table and the classification performed by the simulated investors (Table 8). The sets of rules induced for the TE and EQ investors are the same since they classified the portfolios identically.

Although it was assumed that the portfolio returns are normally distributed in (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013), this framework support other distributions. We consider an approach where no assumptions are made on the distributions of tracking errors and excess returns. Therefore, we compute the sample quantile through a non-parametric approach. Given a vector V of length T, the p-th quantile of the sorted copy of V is the continuous value $\alpha(p)$, defined as $\alpha(p) = V_{(k)} + (b-k)(V_{(k+1)} - V_{(k)})$, where $V_{(k)}$ is the k-th order statistic and $b = pT + h$ is the real number between the indexes $k = \lfloor b \rfloor$ and $k+1$, and $h \in \mathbb{R}$ is an interpolation factor (HYNDMAN; FAN, 1996). This formula shows that when the quantile is different from an observation, which are the cases when $b - k > 0$, then a linear interpolation between the nearest neighbours $V_{(k)}$ and $V_{(k+1)}$ is performed. The minimum corresponds to $p = 0.0$, the median to $p = 0.5$, and the maximum to $p = 1.0$. For this approach, we considered the same meaningful probability levels as the parametric approach.

Considering the quantile definition 7 from (HYNDMAN; FAN, 1996), where $h = 1 - p$, resulting in $b = p(T-1)$, a data table that is generated by this approach is presented in Table 11. The associated induced set of rules for each investor is shown in Table 12.

Table 11 – Data table generated using the 'nonpar_quant' method

x	$TE_{1\%}^{(-)}$	$TE_{50\%}^{(-)}$	$ER_{1\%}^{(+)}$	$ER_{50\%}^{(+)}$	$ER_{99\%}^{(+)}$
x_1	0.0241	0.0056	0.0241	0.0007	-0.0177
x_2	0.0429	0.0118	0.0429	0.0038	-0.0319
x_3	0.0183	0.0048	0.0135	-0.0016	-0.0160
x_4	0.0333	0.0104	0.0333	0.0035	-0.0282
x_5	0.0180	0.0070	0.0180	0.0008	-0.0150
x_6	0.0295	0.0065	0.0290	0.0020	-0.0229

Table 12 – Rules generated for simulated investors using the 'nonpar_quant' data table presentation method

θ	Rule set
TE and EQ	(TE1 <= 0.0241) => (d >= good) (TE1 >= 0.0295) => (d <= other)
ER	(ER1 >= 0.0290) => (d >= good) (ER1 <= 0.0241) => (d <= other)

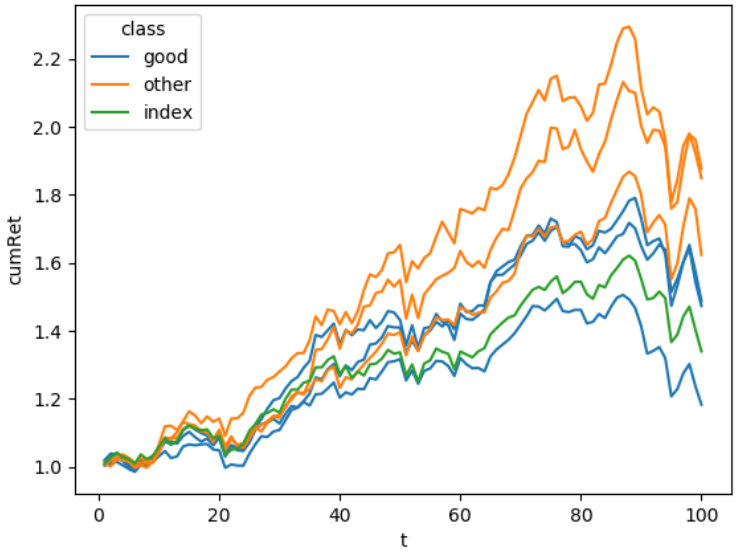
5.1.4.2 Visualization approach

Although only constraints on TE and ER quantiles were discussed, direct constraints on the objectives of the model are also possible. However, the TE and ER objectives have to be included in the data table as evaluation criteria. Following the discussion of (SALVATORE; MATARAZZO; SŁOWIŃSKI, 2013), a direct interpretation of the TE and ER probabilistic measures may be too difficult for the Decision Maker (DM), thus, another way to present portfolios to investors is required.

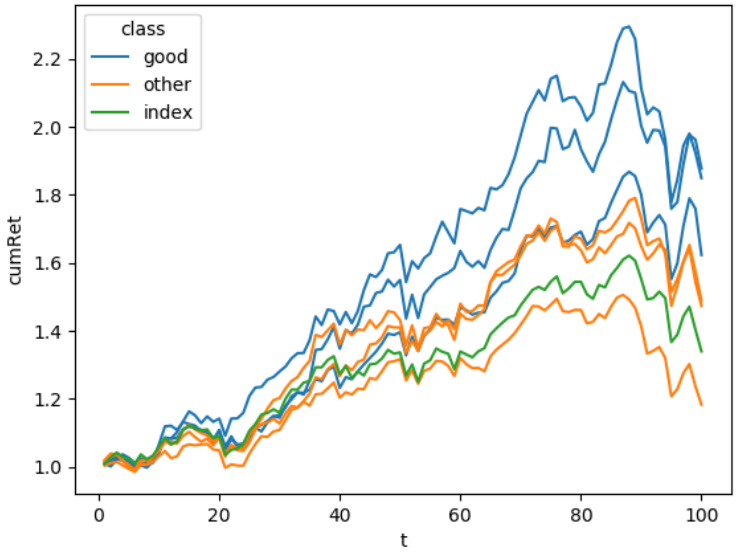
Visualization approaches in multiobjective optimization can enhance and facilitate the decision process (see (CHICA et al., 2013; CHICA et al., 2016)). A visualization tool can support the DM in identifying portfolios that are replicating or outperforming the index. A simple visualization tool could present the cumulative return trajectories of the candidate portfolios and the index and ask the DM to sort the portfolios. After the classification is performed, the complete data table is presented to the rule induction algorithm. Table 7 is an example of a data table generated by the visual approach where the criteria are the objectives of the model.

Considering the classification performed by the simulated investors (Table 8), the result of a simulated interaction with a simple visual tool that presents the cumulative returns of the portfolios of Table 7 is shown in Figure 36.

The simulated portfolio classification in the visual approach appears reasonable. Portfolios deemed 'good' by artificial TE and EQ investors show a similar cumulative return trajectory relative to the index, while portfolios deemed 'good' by artificial ER investors outperform the index. The associated induced set of rules for each investor is shown in Table 13.

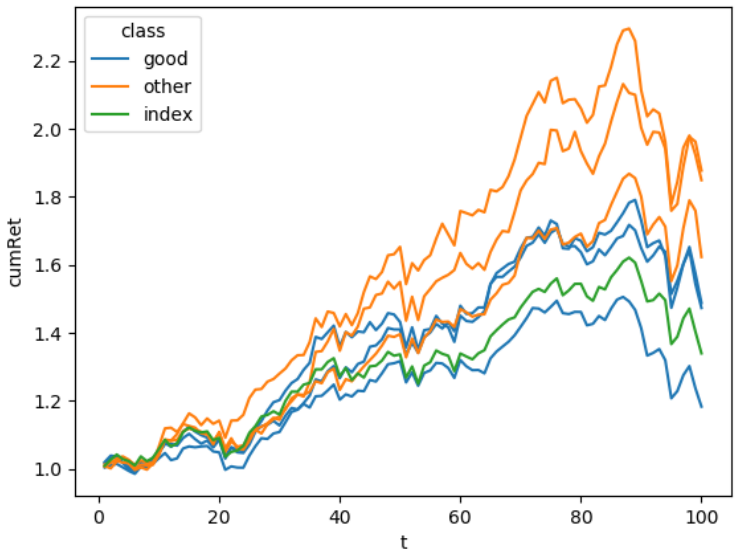


(a) θ^{TE}



(b) θ^{ER}

A challenge associated with this approach in the real world is time series cluttering. It refers to the difficulty of effectively visualizing collections of time series data as the number of series grows, leading to cluttered and overloaded plots that hinder meaningful insights. Interactive visualization tools that allow users to filter, zoom in, or interact with subsets of the time series data can help to reduce visual clutter and enable a more objective analysis (ZHAO et al., 2021).



(c) θ^{EQ}

Figure 36 – The 'visual' data table generation method. This figure presents the simulated visual classification of portfolios from Table 1 for each type of investor.

Table 13 – Rules generated for simulated investors using the 'visual' data table presentation method

θ	Rule set
TE and EQ	(meanTE <= 0.0078) => (d >= good) (meanTE >= 0.0086) => (d <= other)
ER	(meanER >= 0.0020) => (d >= good) (meanER <= 0.0010) => (d <= other)

5.2 MATERIALS AND METHODS

In this section, the simulation approach configurations, the chosen data set containing stock data, the evolutionary algorithm, and the evaluation metrics used for the case study are presented. The experiments were executed in Dell XPS 8940 with Intel(R) Core(TM) i7-10700 CPU (2.90GHz), 16GB RAM, and Windows 11.

5.2.1 Data and software

Data is from OR-library (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files>), containing different markets data, each one containing 291 weekly price data, which results in 290 returns per data set. Table 14 shows the data sets adopted to evaluate the constructed portfolios and the rebalancing strategies according to the rules induced from each simulated investor. We considered the values used by Guastaroba & Speranza (2012) for K.

We consider a rolling window scheme, where the sample size is $T = 100$. Of course that the sample size affects the strategy in the out-of-sample period, but we consider this T

Instance	N	K
Hang Seng (indtrack1)	31	10
DAX100 (indtrack2)	85	10
FTSE100 (indtrack3)	89	10
S&P100 (indtrack4)	98	10
Nikkei225 (indtrack5)	225	10
S&P500 (indtrack6)	457	40

Table 14 – Information about the considered instances for the index tracking problem

satisfactory for this analysis, which is similar to the value adopted by Filippi, Guastaroba & Speranza (2016). We considered the size of the out-of-sample period equal to $T_{oos} = 100$. After the end of the interaction process, the out-of-sample evaluation begins. In the first out-of-sample period $t^* = 101$, the four objectives are estimated using data from $t^* - T + 1$ to t^* and the rule violation is checked. If the rule is violated in t^* and an MBH strategy is used, then the portfolio is rebalanced using data from $t^* - T + 1$ to t^*. This process is repeated until the end of the data set is reached ($t^* = 200$). A total of $200 - T = 100$ out-of-sample portfolio return observations are obtained at the end of this process.

Rules were induced by the VC-DOMLEM algorithm of the jRS library, which is available in the software jMAF(BŁASZCZYŃSKI et al., 2013). Concerning the 'nonpar_quant' data table generation method, we used the default quantile algorithm from Numpy[1], which uses the quantile definition 7 from (HYNDMAN; FAN, 1996), where $h = 1 - p$, resulting in $b = p(T - 1)$.

5.2.2 Multiobjective evolutionary algorithm

NSGA-II (Deb et al., 2002), a non-dominated sorting-based multiobjective algorithm, is usually adopted to solve constrained multiobjective portfolio selection problems and has competitive performance (LIAGKOURAS; METAXIOTIS, 2018; KALAYCI; ERTENLICE; AKBAY, 2019). Thus, we consider using it with the hybrid encoding proposed by Streichert, Ulmer & Zell (2004). The weights of a portfolio are represented by a real-valued vector $\mathbf{w} = \{w_1, w_2, ..., w_N\}$ and the included assets are defined by a binary vector $\mathbf{B} = \{Z_1, Z_2, ..., Z_N\}$. We adopted a population of size 100. The same genetic operators of (SILVA; FILHO, 2021b) were adopted.

The **selection** operator was binary tournament, the uniform **crossover** was applied in the binary vector $\mathbf{B}$ with $p_C = 1.0$ (ANAGNOSTOPOULOS; MAMANIS, 2010; ANAGNOSTOPOULOS; MAMANIS, 2011), the bit-flip **mutation** operator of Sant'Anna et. al (SANT'ANNA et al., 2017) was applied in $\mathbf{B}$ with $p_B = \frac{1}{N}$, and a gaussian mutation with $\sigma = 0.15$ was applied in each decision variable on $\mathbf{w}$ (ANAGNOSTOPOULOS; MAMANIS,

[1] https://numpy.org/devdocs/reference/generated/numpy.quantile.html

2011; Streichert; Ulmer; Zell, 2004). Finally, the **constraint handling** approach proposed by Deb (DEB, 2000), which does not require the specification of penalty parameters, was adopted.

5.2.3 Simulation configurations

The input simulation parameters were: $\tau_1 = 10$s, $\tau_{y>1} = 20$s, $y_{max} = 4$, and $DT_{size} = 6$. For each combination of frontier filters ('frontier_filter'), data table generation methods ('dt_type'), and investor types ('dm_type'), we run the algorithm presented in Figure 35 30 times. We considered that the optimization time to obtain the initial solution (τ_1) was smaller because there is no specific direction to explore in the beginning. To reduce the computational burden when using MBH strategies, we considered a maximum time budget $\tau^{rebal} = 300$s. Also, the maximum optimization time to rebalance the portfolio was equal to $\tau_{y>1}$. Therefore, the maximum number of rebalances per simulation was $\frac{\tau^{rebal}}{\tau_{y>1}} = 15$.

Finally, it is possible that the evolutionary algorithm finds a unique best solution x^y, according to a given rule r^{y-1}, at a given interaction level y. Thus, it was considered that a unique solution x^y is found when the generated DT^y contains at least four copies of the unique best solution. In this case, three copies of x^y will be classified as 'good' and at least one copy of x^y will be classified as 'other'. Therefore, due to inconsistency, it is not possible to generate certain rules that classify examples as 'good'. Then, $r^y = r^{y-1}$.

5.3 RESULTS AND DISCUSSION

5.3.1 Comparison between the frontier filters and data table presentation methods in SBH

Section 2 of Appendix B contains all hypothesis tests for comparisons between frontier filters, investor types, data table presentation methods, interaction levels, in different problem instances. Figure 46 shows, for each instance, the behaviour of the preference function produced by the 'closer' and 'farther' frontier filters (rows = frontier_filter) during the interaction process when optimizing the portfolio for specific types of investors (columns = dm_type) in an SBH strategy. Although we included the results for the initial solution set (intrLvl = 1) their function is just to compare the evolution of the preference function values over the interaction process.

Table 16 show that the differences in the performance of frontier filters are significant for all variables for the 'visual' data table presentation method. Tables 17 and 18 show that for the 'nonpar_quant' and 'par_quant' methods there is no difference in performance between frontier filters in some cases, especially for TE and EQ investors. Observing Figure 46, It can be observed that the 'closer' frontier filter was able to produce better solutions in terms of the preference function values for all simulated investors and interaction levels, depending on the data table presentation method, in most of the instances.

Figure 47 shows the results on the trajectory of the mean preference function in the out-of-sample period, where the solid line is the mean and the area is the estimated 95% confidence interval. It is possible to see that although some of the 'closer' frontier filter results are aligned with the results of the in-sample period (Figure 46), the difference between the performance of the frontier filters is influenced by the interaction level and the type of investor. As the interaction level grows, the feasible region gets more constrained and non-dominated solutions associated with this region are less distant from each other, consequently, the effect of the frontier filter is diluted. We can see in intrLvl=3 and intrLvl = 4 of indtrack6 that it is even possible that solutions that produced a better preference function behaviour over time were those associated with the 'farther' filter.

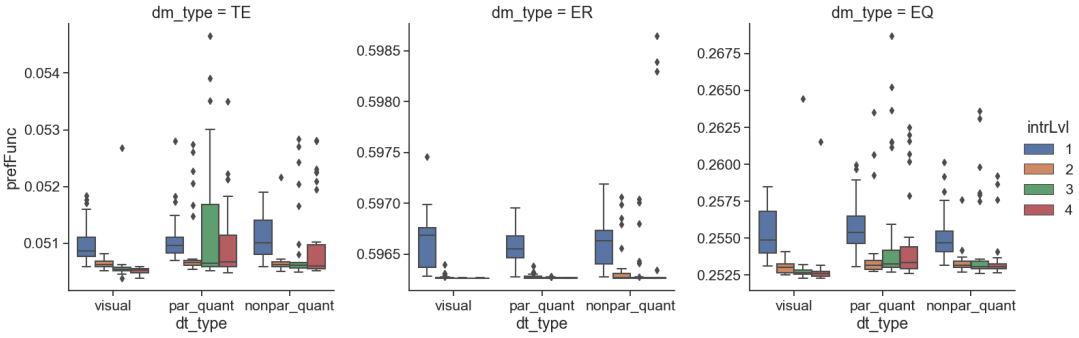
The learned rules for the ER investor guided the solver for local optima, regardless of the data table presentation method, in indtrack1-5. Thus, the out-of-sample performance of the portfolios for the ER investor for different data table presentation methods relative to preference satisfaction is very similar. But, the solutions produced by the 'closer' filter performed better than those produced by the 'farther' filter at the beginning of the out-of-sample period in indtrack1-5 and in the overall out-of-sample period in indtrack6.

It can be observed that the only instance in which the 'closer' and 'farther' frontier filters did not differ so much for TE and EQ was indtrack6. By comparing the ranges of the preference function in the in-sample and out-of-sample period of indtrack6, it can be observed that the quality of solutions produced by both frontier filters decreases very fast over time. This happens because the index dynamics in indtrack6 are defined by a relatively higher number of assets. Thus, at each time step, variations in the index components' prices have a big impact on the optimized portfolio. The out-of-sample performance can be enhanced by evaluating other EMO approaches and/or more robust metrics for the tracking error and the excess return.

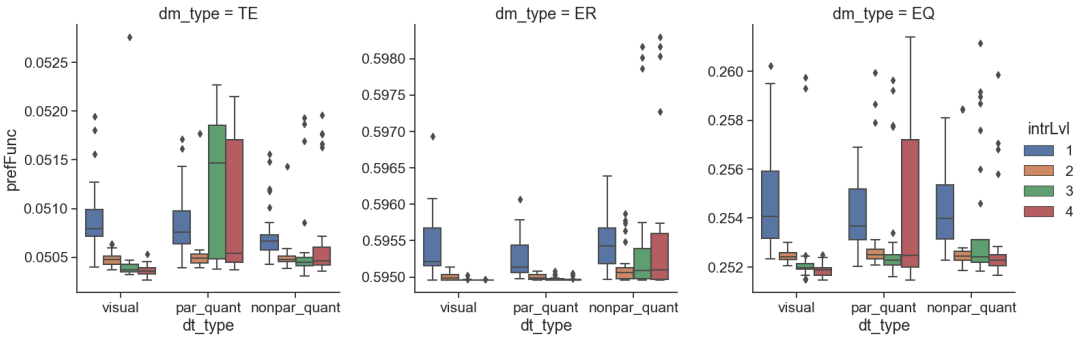
Tables 19, 20, and 21 present which pairs of interaction levels contain significant differences in preference satisfaction for the data table presentation methods combined with the 'closer' frontier filter, which was the filter that performed better. It could be observed that, in indtrack1-5, a unique solution is found almost always for the ER investor when intrLvl>2. The 'nonpar_quant' approach performs better than the 'par_quant' approach with all the investors in all instances in intrLvl=2, as shown in Tables 20 and 21. This shows that unlike 'par_quant', the 'nonpar_quant' method can guide the solver to a better search region at the beginning of the interaction process for all investors in all problem instances.

By analyzing Figure 37, it is possible to observe that the 'visual' and 'nonpar_quant' approaches can better guide the evolutionary algorithm towards an increase in the preference function of the simulated investors along the interaction process.

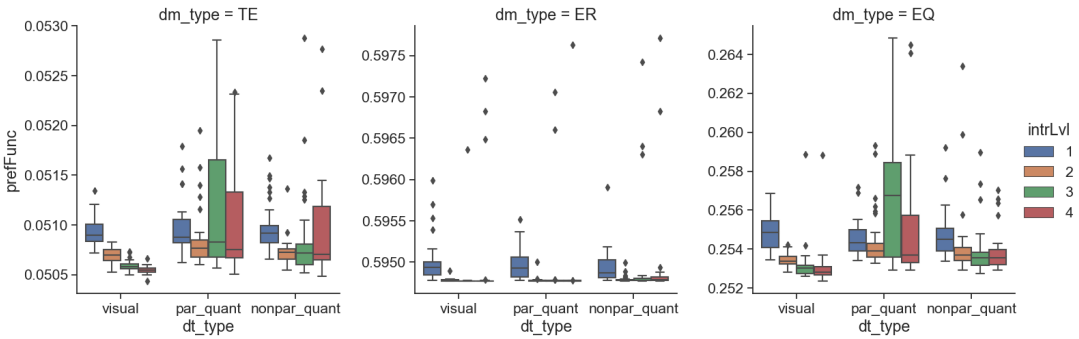
Figure 47 shows that the 'visual' approach may produce better portfolios than the 'nonpar_quant' approach, for simulated TE and QE investors, when intrLvl=2, which



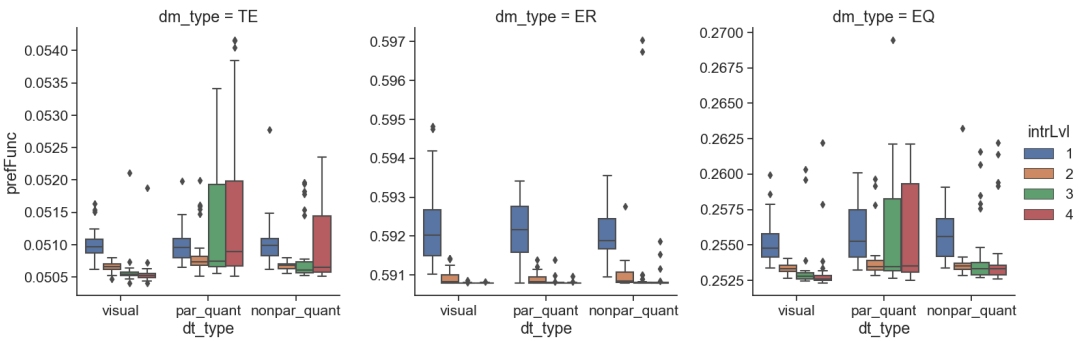
(a) indtrack1



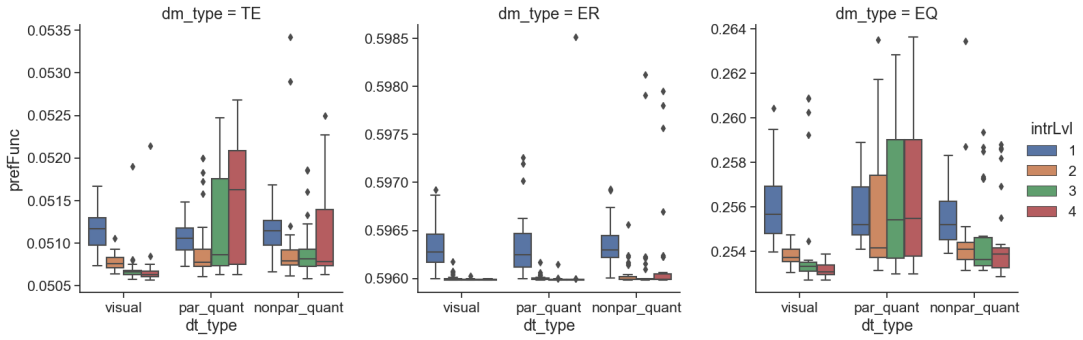
(b) indtrack2



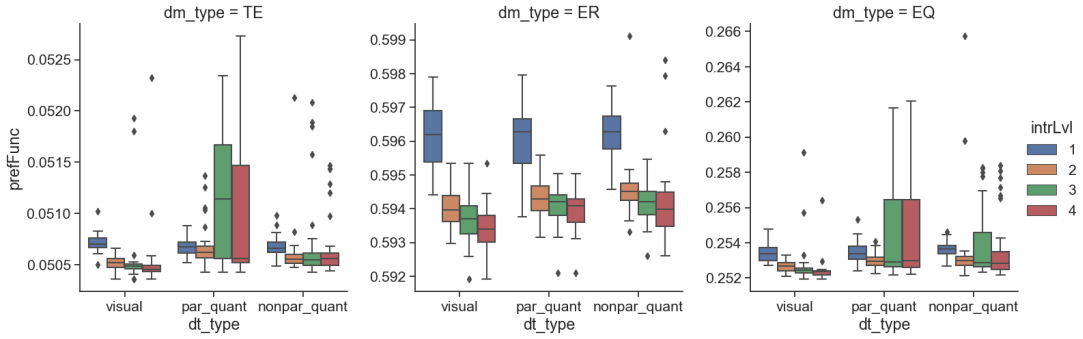
(c) indtrack3



(d) indtrack4



(e) indtrack5



(f) indtrack6

Figure 37 – Comparison of the preference function of three simulated investors with the 'closer' frontier filter and different data table presentation methods in each simulated interaction for the in-sample period of the six instances of the problem

was the most significant interaction level for the 'nonpar_quant' approach. The solutions produced using the 'visual' approach appear to be more robust than the solutions produced by the 'nonpar_quant' approach for the TE investors in indtrack2-5 and for the EQ investors in indtrack2 and intrack4-6. For ER simulated investors, the performance of the 'visual' and 'nonpar_quant' approaches is similar, whereas the 'visual' approach performed better only in indtrack6. This happens because IMO-DRSA guides the solver to local optima in indtrack1-5 at the very beginning of the interaction process, regardless of the data table presentation method, as shown in the in-sample period results for the ER investor.

More significant differences in preference satisfaction were perceived between sequential interaction levels for the 'visual' approach. Thus, it is necessary to analyze what happens with the out-of-sample performance of the produced portfolios along the interaction process. Observing Figure 38, more interactions may produce portfolios that perform slightly better at the very beginning of the out-of-sample period as can be observed for the simulated TE investors in indtrack1-3, and the simulated EQ investors in indtrack1-3 and indtrack5.

Less interaction can sometimes lead to better portfolio performance during a significant portion of the out-of-sample period. This is true for simulated TE investors in indtrack1 and indtrack4, and simulated EQ investors in indtrack4 and intrack6. For the ER investor,

there is not much difference between interaction levels because the solver quickly reaches local optima during the interaction process.

5.3.2 Importance of attributes along the interactions

Since the 'closer' frontier filter performed better in the in-sample and out-of-sample period of the SBH rebalancing strategy, we use it to investigate how the IMO-DRSA approach detects which attribute is more important for a given type of investor in each step of the interaction process. Figure 48 shows the frequency of each attribute and its preference direction in the rules induced during the interaction process.

When analyzing the results of the first interaction of all data table presentation methods, it is possible to see that attributes related to the excess return objective were prioritized for ER investors. Also, attributes related to the tracking error objective were prioritized for TE and EQ investors in the first interaction when considering the 'par_quant' and 'nonpar_quant' approaches. Both $\mu(TE)$ and $max(-ER)$ criteria were prioritized in the first interaction for the TE and EQ investors when considering the 'visual' approach.

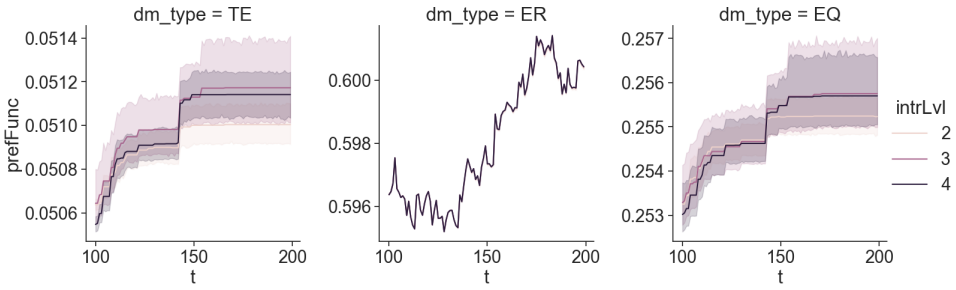
It appears that the criteria importance is clear for the VC-DOMLEM algorithm at the beginning and also that using the ER downside risk attribute in the 'visual' approach would optimize preference satisfaction for TE and EQ investors. The strategy learned by the VC-DOMLEM algorithm in the beginning of the interaction process could guide the solver to optimize the preference function in a significant way for all investor types and data table generation approaches in at least half of the instances of the problem, as shown in Tables 19-21.

For the ER investor, regardless of the data table presentation method, the VC-DOMLEM algorithm learned, in the second and third interactions, that prioritizing ER-related attributes for this type of investor was a good strategy. By adopting this approach, VC-DOMLEM guided the solver to local optima before the end of the interaction process as shown in Figure 37.

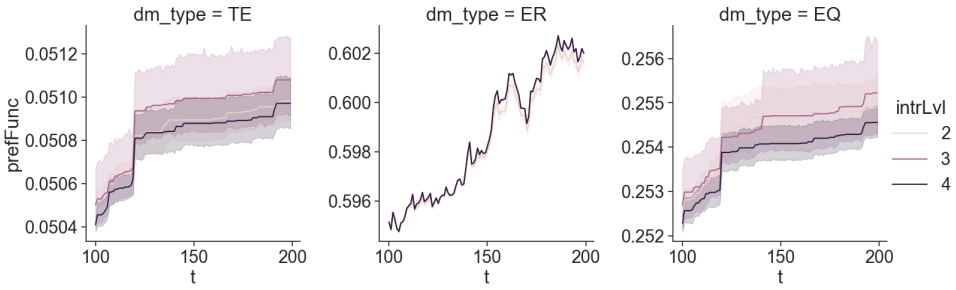
Observing the 'visual' data table presentation method in the second and third interactions, the attribute prioritized for the TE and EQ investors was mostly the ER downside risk. By prioritizing this attribute, IMO-DRSA guided the solver to optimize the preference function significantly along the interaction process when considering the 'visual' approach. Looking at the 'par_quant' and 'nonpar_quant' data table presentation methods, it can be observed that there is no specific attribute prioritization, which may lead to non-significant preference optimization over interactions two and three in the in-sample results, as shown in Tables 20 and 21.

5.3.3 Comparison between SBH and preference-driven MBH rebalancing

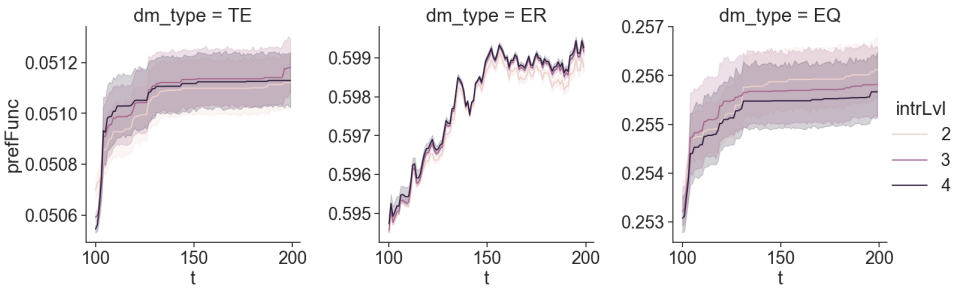
We continue the analysis considering the 'visual' and 'nonpar_quant' data table methods combined with the 'closer' frontier filter since they obtained the best performance for the



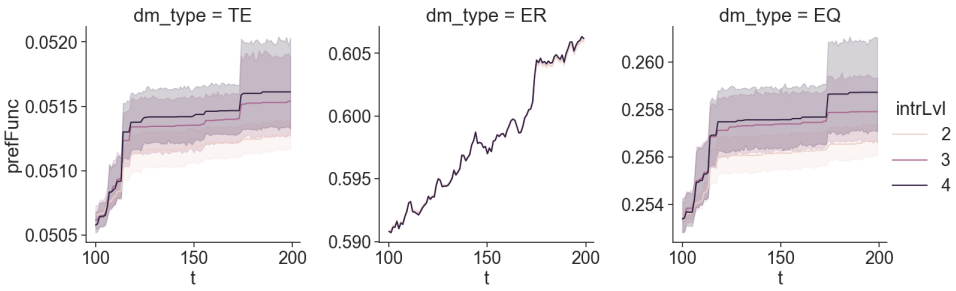
(a) indtrack1



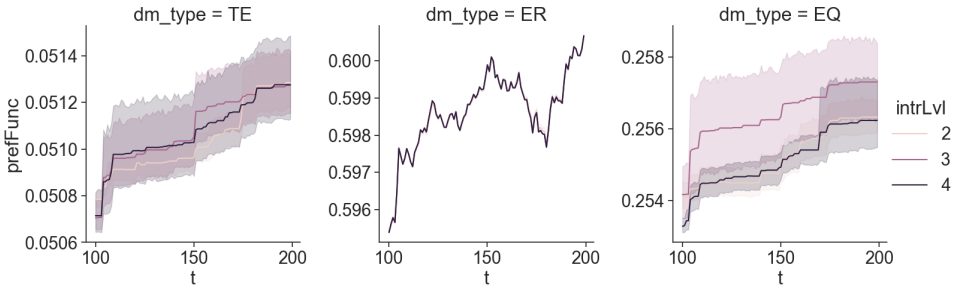
(b) indtrack2



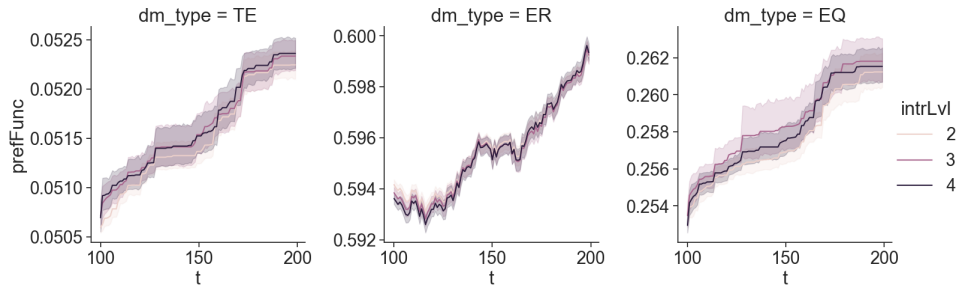
(c) indtrack3



(d) indtrack4



(e) indtrack5



(f) indtrack6

Figure 38 – Comparison of the preference function of three simulated investors with the 'closer' frontier filter and the 'visual' data table presentation methods in each simulated interaction for the out-of-sample period of the six instances of the problem

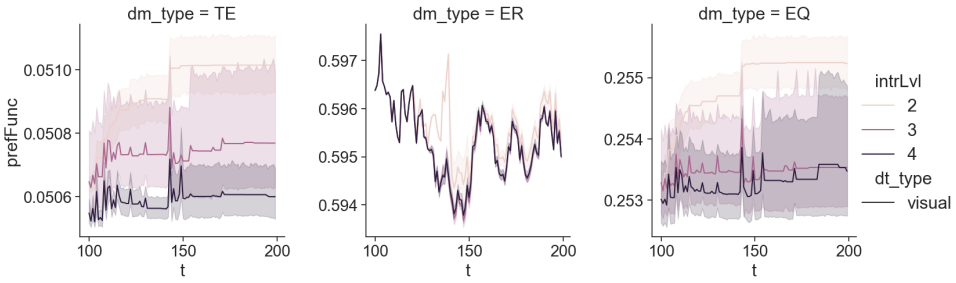
SBH rebalancing strategy. Appendix B contains all preference satisfaction evaluations for the SBH and MBH rebalancing strategies considering all frontier filters, investor types, data table presentation methods, interaction levels, in different problem instances.

Figure 49 shows the mean trajectory of the preference function evaluation in the out-of-sample period for the SBH and MBH strategies, where the solid line is the mean and the area is the estimated 95% confidence interval.

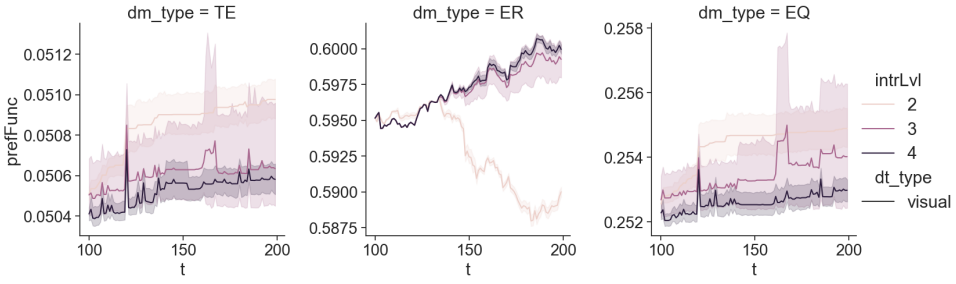
The results show that there is not much difference between the SBH and MBH rebalancing strategies when considering information obtained at the second interaction level for TE and EQ investors. This happens because the induced rules in intrLvl=2 are satisfied more easily in the out-of-sample period. As the interaction level increases the induced rules are more rigorous, and it is possible to observe the construction of portfolios that better maintain, or may even increase, the preference function value over time when adopting MBH. The 'visual' data table presentation method may induce better rules for constructing portfolios that better satisfy the investors over time when adopting MBH. The 'visual' approach could produce better solutions than the 'nonpar_quant' approach, as the interaction level increases, for TE and EQ investors in all instances.

Unlike the other investor types, for the ER investor, there is a difference between the SBH and MBH at the second interaction level. This is because the produced portfolios are more volatile and degrade preference satisfaction much faster. Thus, the portfolios of ER investors are adjusted more frequently at this early interaction stage. It can be seen that the MBH strategy with the 'visual' approach produced better portfolios than the MBH strategy with the 'nonpar_quant' in all instances at intrLvl=2.

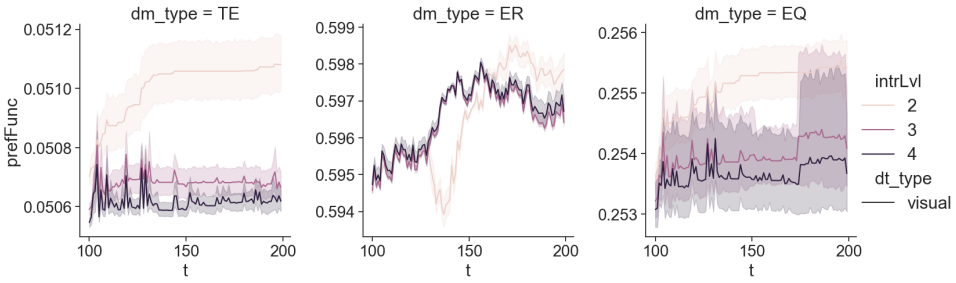
As the interaction level increases for the ER investor, the MBH strategy could produce better solutions than the SBH strategy in the majority of the instances. The MBH strategy with the 'visual' approach could perform better than the MBH strategy with the 'nonpar_quant' approach in indtrack1, indtrack3 and indtrack6 in the overall out-of-sample period and at the beginning of the out-of-sample period in indtrack2 and intrack4-5. Thus,



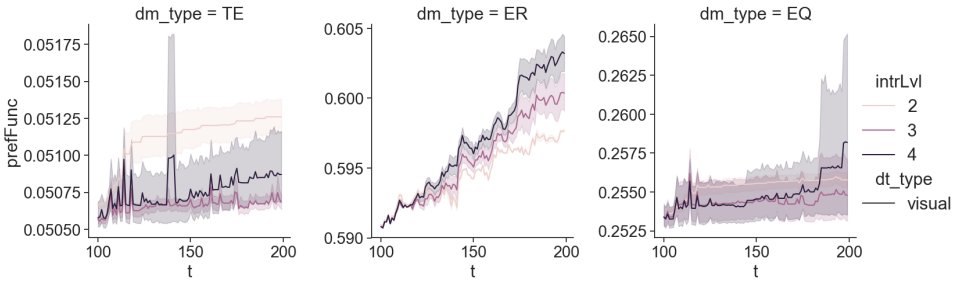
(a) indtrack1



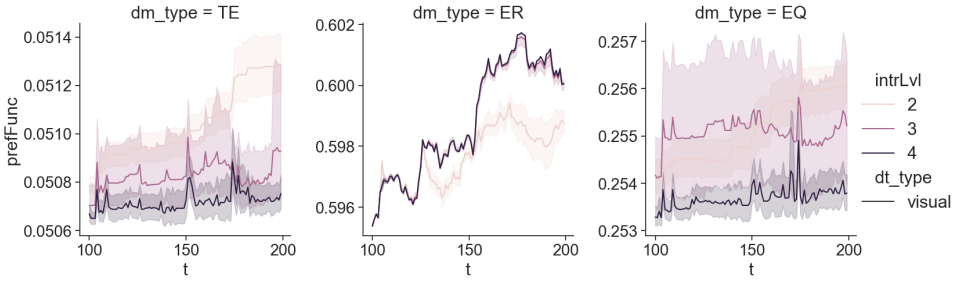
(b) indtrack2



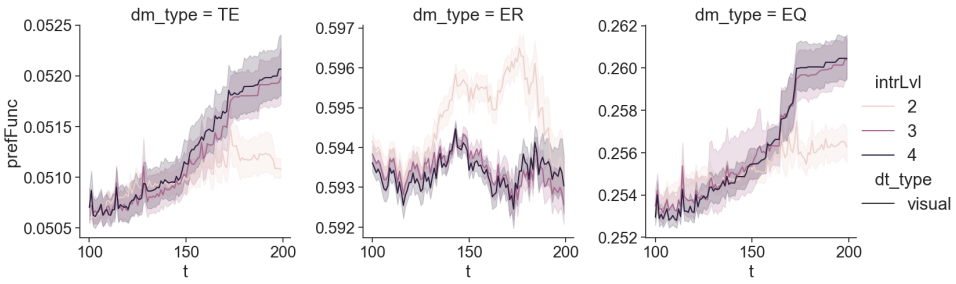
(c) indtrack3



(d) indtrack4



(e) indtrack5



(f) indtrack6

Figure 39 – Comparison of the preference function of three simulated investors when using the MBH strategy for the out-of-sample period of the six instances of the problem

although the visual approach may produce more rigorous rules, spending the rebalances in the beginning of the out-of-sample period in some cases, it may produce better portfolios at the beginning of the out-of-sample period. Thus, it may be necessary to increase the rebalance budget when dealing with ER investors to manage preference satisfaction as the interaction level increases.

Figure 39 shows the performance of the MBH strategy with the 'visual' approach as the interaction level increases. It is possible to observe that, for the TE investor, the higher the number of interactions, the better the portfolios, except for indtrack6. For the EQ investor, it is possible to observe a better performance when more interactions are performed in indtrack1-3. For the ER investor, higher interactions only generated better portfolios in indtrack1 and intrack6.

5.3.4 Comparison against NSGA-II with and without preference guidance

What would happen if the simulated investor evaluated all the solutions in a given non-dominated frontier instead of a subset of non-dominated solutions from this frontier? Would the performances be similar or different? The results are shown in Table 22 and Figure 50, where the 'closer' frontier filter and the 'visual' data table presentation approach were adopted for IMO-DRSA and compared with the NSGA-II algorithm without the rules extracted from the simulated investor through IMO-DRSA, which is identified in the plot as dt_type = 'none'. The best choice of an investor was simulated based on its preference function, from interactions with complete non-dominated frontiers generated with NSGA-II, considering the same maximum optimization time of the IMO-DRSA in the respective interactions.

It can be observed that in the considered problem instances, the preference function of the IMO-DRSA approach would be at least as good as those of the NSGA-II from at least interaction three in all the instances for the TE and EQ investors. There are some cases where IMO-DRSA performs significantly better than NSGA-II for TE and EQ investors. For TE investors, IMO-DRSA performs better for intrLvl=3 in indtrack 1-2 and indtrack4-

5, and for intrLvl=4 in indtrack1-5. For EQ investors, IMO-DRSA performs better for intrLvl=4 in indtrack 1-2 and intrack5. Concerning the ER investor, the performance of IMO-DRSA had a similar performance relative to NSGA-II from interaction level three onwards. Thus, by performing at least three interactions with only six solutions it is possible to obtain at least the same preference function values of interacting with all the non-dominated frontier solutions.

5.4 CHAPTER CONCLUSION

This study presented a new application of the IMO-DRSA to a multiobjective index tracking model. We studied how different factors would affect the IMO-DRSA performance concerning the satisfaction of the simulated investors' preferences. First, the frontier filters and data table generation methods were analyzed for three types of investors. Thus, it was possible evaluate if different ways to present the candidate portfolios would produce good or bad constraints to guide the evolutionary algorithm to optimize the candidate portfolios in a way that maximized their preferences. Also, the capacity of IMO-DRSA in detecting investor preferences was evaluated. Not only the performance of the evolutionary algorithm during the interactive process was evaluated, but also in the out-of-sample period, using the induced rules as SBH and MBH rebalancing strategies. Finally, the performance of the IMO-DRSA against NSGA-II was compared to evaluate what would be the performance w.r.t. preference satisfaction, during the interaction process, with and without preference learning during interactions.

In general, it is possible that the 'farther' frontier filter majorly produces portfolios with worse preference function distribution than the 'closer' frontier filter in the in-sample and out-of-sample periods. Also, it was possible to observe the difficulties concerning the optimization (in the in-sample period) and maintenance (in the out-of-sample period) of the preference function imposed by indexes with a higher number of assets.

The experiments showed that constraints produced by different data table generation methods affect the performance of the evolutionary algorithm. The constraints produced by the 'visual' approach can better guide the evolutionary algorithm towards the most desirable region for the simulated investors and may provide more robust solutions w.r.t. the investor's preference over time. Another advantage of the 'visual' approach is that the number of data table features is equal to the number of objectives, while in the stochastic dominance approaches, the number of features can surpass the number of objectives. The more the investor aims to achieve higher returns, the harder will be to maintain (out-of-sample) the preference function evaluation, especially when broader indexes are considered. Also, when we look at the SBH strategy, the effect of the different data table presentation methods and frontier filters is generalized for the different types of investors as the number of interactions increases.

From subsection 2.4 of Appendix B, it was observed that the 'closer' filter combined

with the 'visual' approach produced portfolios with a significantly lower variance of preference satisfaction relative to the 'farther' filter combined with the 'visual' approach. It was also possible to observe that a significant reduction of the variance of preference satisfaction occurred mostly when the interaction process shifts from interaction level 1 to interaction level 2 in the considered IMO-DRSA approach. Also, NSGA-II could only obtain significantly lower preference satisfaction variance compared to the considered IMO-DRSA approach mostly at interaction level 2.

Although the VC-DOMLEM could balance TE and ER criteria for conservative and moderate investors, it used a different strategy to satisfy aggressive investors (ER) along the interaction process, which was to prioritize ER-related attributes only. Thus, combining TE attributes and ER downside risk attributes along the interaction process was beneficial for TE and EQ investors in the 'visual' approach, whereas prioritizing ER attributes only may guide the solver to local optima during the interaction process, as was the case for ER investors. Also, when the VC-DOMLEM doesn't identify the correct attributes to prioritize, it leads the solver to non-significant preference optimization along the interaction process.

The results comparing SBH and preference-driven MBH show that by performing more interactions it is possible that preference-driven MBH maintains or even produces better satisfaction than SBH. Also, the cost to maintain satisfaction is different depending on the investor type, where an investor that is more tilted to obtain higher risk would require a higher rebalancing frequency than the conservative (TE) and moderate (EQ) investors.

This study proposes ways to reduce the cognitive effort of investors by using the simulated IMO-DRSA. First, methods like DRSA, which use decision examples as indirect preference information, require less cognitive effort from the DM (de Lima Silva; FERREIRA; de Almeida Filho, 2023; KADZIŃSKI; TOMCZYK; SŁOWIŃSKI, 2020; SILVA; de Almeida-Filho, 2018; GRECO; MATARAZZO; SLOWINSKI, 2001). Secondly, the simulated IMO-DRSA can reduce the number of interactions by assessing to what extent the number of interactions becomes redundant, considering different factors, such as the type of investor and size of the index. Thirdly, presenting fewer reference solutions to the DM will likely reduce the cognitive effort (MESQUITA-CUNHA; FIGUEIRA; BARBOSA-PÓVOA, 2022; XIN et al., 2018). Frontier filters present a subset of portfolios from the Pareto frontier to the investor based on his/her preferences. The comparison with NSGA-II showed that IMO-DRSA can be an effective way to filter good-quality portfolios from the complete frontier with the progression of the interaction process.

Another point to be investigated is to measure gains in other variables with experiments on humans when using different configurations of IMO-DRSA interaction process. Designing experiments using human decision-makers for this problem poses several challenges, such as ensuring that the participants can understand the portfolio optimization problem and designing questionnaires to evaluate preference satisfaction in the in-sample and out-of-sample periods. Another challenge is to measure and handle distinct cognitive loads

(SWELLER, 1988; CHANDLER; SWELLER, 1991; SWELLER; MERRIENBOER; PAAS, 1998) when comparing different data table generation methods.

Experiments with real DMs can be useful to further develop a more complete 'visual' approach that presents sufficient information to sort the presented portfolios into 'good' or 'bad' evaluations. Balancing detail and simplicity is a challenge when developing visualization tools for this problem. For instance, a low-detail visualization shows cumulative return trajectories and a detailed visualization would provide more information when the DM clicks a portfolio's trajectory.

Finally, future work involving this simulation approach could analyze other multiobjective optimization schemes, such as DRSA-EMO or DARWIN (GRECO; MATARAZZO; SŁOWIŃSKI, 2010), and preferences extracted through pairwise solution comparisons (GRECO; MATARAZZO; SŁOWIŃSKI, 2010; KADZIńSKI; TOMCZYK; SłOWIńSKI, 2020; BRANKE et al., 2016), Non-Compensatory Sorting models (TLILI et al., 2022), IEMO (CORRENTE et al., 2024), and other ways to choose representative sets of the pareto frontier (MESQUITA-CUNHA; FIGUEIRA; BARBOSA-PóVOA, 2022).

6 CONCLUSIONS AND FUTURE WORK

Chapter 3 explores the combination of GANs and metaheuristics to obtain more robust portfolios. This was performed by using the metaheuristics to solve the index tracking model in multiple market scenarios generated by a GAN to obtain weights less sensitive to the uncertainty of the model parameters in the out-of-sample period. Two new metaheuristics were proposed and evaluated for the index tracking model with multiple scenarios in real data. The results showed that the direct incorporation of uncertainty in the model through GAN's market simulations produces more stable portfolios when compared to portfolios constructed through historical data. Moreover, it was observed that the produced models trained for a specific rebalancing could perform well in other rebalancing strategies. Also, potential solutions that mitigate errors relative to simulated markets from GANs were discussed.

Studies on the practical application of IMO-DRSA in the portfolio optimization context were performed in Chapters 4 and 5, considering different factors and their influence on EMO approaches in producing good portfolios for different types of simulated investors. In chapter 4, a simulated IMO-DRSA was proposed to study the impact of different factors on the portfolio robustness concerning investor preferences. Ways to filter the approximate Pareto frontier using the induced rules were proposed. The experiments investigated the gains of performing more interactions for different types of investors based on one of the proposed frontier filters. In chapter 5, the simulated IMO-DRSA analysis was extended and new components were evaluated. More objectives were considered in the model and a better application generalization was performed by using more problem instances. Not only the effect of the number of interactions was evaluated, but also the effect of different frontier filters, data table presentation methods, and the way the portfolio is updated by the evolutionary algorithm.

This thesis showed the importance of computational intelligence in producing more robust portfolios and interactively adjusting the portfolio to the preferences of the investor. A summary of the characteristics of the investigation that was carried out in each chapter is presented in Table 15.

The answers to the investigated research questions are presented below:

RQ1: How can GAN-generated market scenarios improve the robustness of portfolios compared to those constructed using historical data? **Answer:** It was observed that GANs produced more robust portfolios, contingent on the rebalancing frequency.

RQ2: Are there benefits to using multiobjective metaheuristics in the index tracking model when addressing multiple market scenarios generated by GANs? **Answer:** More

Table 15 – Summary of aspects investigated in each chapter

Aspect	**Chapter 3**	**Chapter 4**	**Chapter 5**
Algorithm	GANs and GAs	IMO-DRSA and GAs	IMO-DRSA and GAs
Formulation	Index tracking	Mean-Variance	Multiobjective Index Tracking
Investor Preferences	Not addressed	Addressed through interactions	Addressed through interactions
Uncertainty	Addressed using scenarios	Not addressed	Not addressed
Practical Constraints	Cardinality and Holding	Cardinality, Holding and Investor preferences	Cardinality, Holding and Investor preferences

robust portfolios were achieved using a multi-objective approach, although there is a trade-off between portfolio robustness and computational cost.

RQ3: Given that a GAN was trained on a specific rebalancing strategy, is it possible to apply it to other rebalancing strategies without retraining? **Answer:** The GAN could be reused for other rebalancing strategies. However, its performance may vary, potentially being better or worse than for the original rebalancing strategy on which it was trained.

RQ4: Can the generators learn to produce better market simulations over the training epochs? **Answer:** Based on the observed results, it is preferable to train for more epochs, as generators facilitated the construction of more robust portfolios with longer training.

RQ5: How does the application of IMO-DRSA influence portfolio optimization in terms of cognitive effort? **Answer:** Through a simulated IMO-DRSA approach, it was observed that IMO-DRSA can potentially reduce cognitive effort, identify and sustain investor performance, depending on the investor's type.

RQ6: How do the number of interactions influence the satisfaction of the investor? **Answer:** Increasing the number of iterations and applying effective frontier filtering significantly enhance the evolutionary algorithm's ability to align with the investor's preferences over time.

RQ7: How do different methods of presenting data tables influence portfolio optimization, considering that different data presentation methods generate distinct constraints? **Answer:** It was observed that certain data table presentation approaches can impact the performance of the evolutionary algorithm, with the visual approach being the most effective as it better incorporates constraints into the search space.

RQ8: How does investor-specific preferences influence the performance of portfolio optimization methods? **Answer:** It was observed that the IMO-DRSA approach was better at handling the preferences of conservative and moderate investors compared to those of aggressive investors.

RQ9: How do frontier filters influence portfolio optimization methods? **Answer:** IMO-DRSA frontier filters reduce the cognitive effort of investors by selecting a subset of available options from the Pareto front and outperform NSGA-II as the number of iterations increases.

RQ10: How do preference-based update mechanisms impact the adaptability of portfolios over time? **Answer:** Depending on the type of investor, IMO-DRSA can identify their preference model and refine it over interactions, enabling the maintenance of preferences over time by utilizing this model.

6.0.1 Novel Contributions Relative to Existing Literature

Chapter 3 advances index tracking by combining Generative Adversarial Networks (GANs) with mathematical programming, enabling a more realistic and robust portfolio construction process. Unlike traditional deep learning and sparse approaches, mathematical programming handles real-world constraints, such as liquidity requirements, convex MINLP formulations, multiobjective optimization, and uncertainty through scenario-based models. GANs capture complex nonlinear relationships between assets and the index while generating synthetic market scenarios, which can be incorporated into approximate solutions for NP-hard index tracking models. This allows for addressing uncertainty and enhancing single and multiobjective pure/hybrid genetic algorithms (GAs).

Chapter 4 presented a simulated approach to interactive multiobjective optimization, exploring its application within the IMO-DRSA methodology. The study examined the impact of both the number of interactions and the investor type on the performance of a Genetic Algorithm (GA) during the optimization process. This work marks the first simulated approach to focus on designing the interactive process with the goal of enhancing preference extraction, thereby improving overall optimization results.

Chapter 5 expanded upon the simulation approach introduced in Chapter 4 by incorporating additional aspects of the interactive process, addressing a greater number of objectives, and applying a recent financial optimization model across multiple markets. Compared to previous studies, this chapter analyzed the impacts of various factors—such as investor types, frontier filters, the number of interactions, and different methods of presenting portfolios to investors—on the performance of a Genetic Algorithm (GA) in identifying investor preferences and sustaining their satisfaction over time. These findings represent the first comprehensive investigation of such factors in the literature within this field.

6.0.2 Future work

Future work may explore more complex models with other types of practical constraints and objectives, especially those associated with ESG. Also, it is necessary to develop new metaheuristics to increase the numerical performance of both approaches to portfolio optimization approaches. Other research directions concern experiments on the effect of different variables with real decision makers and better ways to handle the preference constraints in the interactive approach.

Future work could focus on developing robust methods for generating synthetic covariance matrices using advanced generative models, such as GANs or Variational Autoencoders (VAEs). These synthetic matrices could capture the intricate relationships between assets more accurately than traditional methods. Also, other index tracking metrics, such as those discussed in Gaivoronski, Krylov & Wijst (2005) can be adopted. Another future direction concern the tracking of other types of assets, such as Hedge Funds, to infer their positions.

In future work, it is important to consider how different training inputs, such as prices, returns, and sentiment, impact the Generative model performance. The proposed SDM heuristics should be further studied by considering statistical robustness analysis, a larger out-of-sample period, other performance metrics (VaR, Turnover, Annualized Volatilty, and others), and intraday rebalancing.

In addition, visualization tools can be developed to support the visualization of simulated scenarios to inspect GAN quality and to support preference learning in the interactive approach. Approaches to detect the convergence of Pareto fronts, particularly when there are minimal differences between the risk and return of the alternatives, which can serve as an early stopping criterion for the interaction process, will be analyzed. Another promising research direction is to evaluate the impact of integrating multiple decision rules into the preference model using IMO-DRSA.

6.1 RESEARCH DEVELOPMENTS

The developments of this thesis resulted in three articles, which correspond to chapters 3, 4, and 5. The article that constitutes Chapter 3 was published in the Applied Soft Computing Journal (SILVA; de Almeida Filho, 2023). The second article (Chapter 4) was published at the IEEE Congress on Evolutionary Computation in 2021 (SILVA; FILHO, 2021b). The last article, that was presented in Chapter 5 was published in the IEEE Transactions on Evolutionary Computation (SILVA; FILHO, 2023).

Along the PhD studies, other articles have been published. A paper concerning the use of DRSA in explaining and predicting rating assessments of sovereign bonds was published in 4OR (SILVA et al., 2021). Papers that presented metaheuristics for index tracking were published in the International Transactions in Operational Research (SILVA; SILVA; FILHO,

2022b) and Soft Computing (AMORIM; SILVA; FILHO, 2023). A systematic literature review concerning index tracking models and associated metaheuristics was published in the IMA Journal of Management Mathematics (SILVA; FILHO, 2023).

In addition, other parallel works have been developed within the GREEFO research group. The collaborative efforts of a volunteer-driven data science task force, supported by Porto Digital and the Prefeitura do Recife, culminated in the publication of significant research papers in reputable journals. These include a study published in Applied Soft Computing (SILVA et al., 2023) and another in Sustainable Cities and Society in 2021 (SILVA et al., 2021). A chapter contribution on preference learning in credit risk has been published in the book "Intelligent Decision Support Systems - Combining Operations Research and Artificial Intelligence - Essays in Honor of Roman Słowiński" in 2022 (FILHO et al., 2022).

Works that adopted computational intelligence in quantitative finance problems were also published in national and international conferences. An article that applied adversarial autoencoders for multi-class fraud detection was published in the International Joint Conference in Neural Networks (IJCNN) in 2021 (SILVA et al., 2021). The proposed simulated IMO-DRSA applied in the classical mean-variance problem was published in the IEEE Congress on Evolutionary Computation in 2021 (SILVA; FILHO, 2021b). An article that evaluated the performance of genetic algorithms to optimize tracking portfolios during the COVID-19 pandemic was published in the IEEE Congress on Evolutionary Computation in 2021 (AMORIM; SILVA; FILHO, 2021). Another conference article, that proposed the use of bayesian regression to compare districts to build new residential real estate developments, was published in IEEE SMC in 2020 (SILVA; FILHO, 2020). Other works concerning computational intelligence applied to quantitative finance were published and presented in SBPO (AMORIM; SILVA; FILHO, 2020), CSBC (SILVA; SILVA; FILHO, 2020), and INFORMS (SILVA; FILHO, 2021a).

REFERENCES

ACOSTA-GONZALEZ, E.; ARMAS-HERRERA, R.; FERNANDEZ-RODRIGUEZ, F. On the index tracking and the statistical arbitrage choosing the stocks by means of cointegration: the role of stock picking. *Quantitative Finance*, 15, n. 6, SI, p. 1075–1091, 2015. ISSN 1469-7688.

Ahmed Bacha, S. Z.; BENATCHBA, K.; Benbouzid-Si Tayeb, F. Adaptive search space to generate a per-instance genetic algorithm for the permutation flow shop problem. *Applied Soft Computing*, v. 124, p. 109079, 2022.

ALMEIDA-FILHO, A. T. de; SILVA, D. F. de L.; FERREIRA, L. Financial modelling with multiple criteria decision making: A systematic literature review. *Journal of the Operational Research Society*, Taylor & Francis, v. 72, n. 10, p. 2161–2179, 2021.

Alpha Vantage. *Alpha Vantage - API Documentation*. 2021. <https://www.alphavantage.co/documentation/>.

AMORIM, T. W. D.; SILVA, J. C. S.; FILHO, A. T. D. A. Evaluation of index tracking portfolios during the covid-19 pandemic. In: *2021 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2021. p. 1569–1576.

AMORIM, T. W. de; SILVA, J. C. S.; FILHO, A. T. de A. Comparação e avaliação de algoritmos genéticos em modelos de index tracking aplicados ao ibovespa. In: *Anais do 52º Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2020.

AMORIM, T. W. de; SILVA, J. C. S.; FILHO, A. T. de A. Assessing the interactions amongst index tracking model formulations and genetic algorithm approaches with different rebalancing strategies. *Soft Computing*, Springer Science and Business Media LLC, v. 28, n. 6, p. 4847–4860, set. 2023. ISSN 1433-7479. Disponível em: <http://dx.doi.org/10.1007/s00500-023-09185-7>.

ANAGNOSTOPOULOS, K.; MAMANIS, G. A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, v. 37, n. 7, p. 1285 – 1297, 2010. ISSN 0305-0548. Algorithmic and Computational Methods in Retrial Queues.

ANAGNOSTOPOULOS, K. P.; MAMANIS, G. The mean–variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, v. 38, n. 11, p. 14208–14217, 2011.

ANDRIOSOPOULOS, K.; DOUMPOS, M.; PAPAPOSTOLOU, N. C.; POULIASIS, P. K. Portfolio optimization and index tracking for the shipping stock and freight markets using evolutionary algorithms. *Transportation Research Part E-logistics and Transportation Review*, 52, n. SI, p. 16–34, 2013. ISSN 1366-5545.

ANDRIOSOPOULOS, K.; NOMIKOS, N. Performance replication of the Spot Energy Index with optimal equity portfolio selection: Evidence from the UK, US and Brazilian markets. *European Journal of Operational Research*, 234, n. 2, p. 571–582, 2014. ISSN 0377-2217.

ANIS, H. T.; COSTA, G.; KWON, R. H. Risk-allocation-based index tracking. *Computers & Operations Research*, v. 154, 2023.

ARAUJO, R. d. A.; NEDJAH, N.; OLIVEIRA I, A. L.; MEIRA, S. R. d. L. A deep increasing-decreasing-linear neural network for financial time series prediction. *Neurocomputing*, 347, p. 59–81, 2019. ISSN 0925-2312.

ARJOVSKY, M.; BOTTOU, L. Towards principled methods for training generative adversarial networks. In: *International Conference on Learning Representations*. [s.n.], 2017. Disponível em: <https://openreview.net/forum?id=Hk4_qw5xe>.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein generative adversarial networks. In: PRECUP, D.; TEH, Y. W. (Ed.). *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 214–223. Disponível em: <https://proceedings.mlr.press/v70/arjovsky17a.html>.

ARKHIPOV, D. I.; WU, D.; WU, T.; REGAN, A. C. A parallel genetic algorithm framework for transportation planning and logistics management. *IEEE Access*, v. 8, p. 106506–106515, 2020.

ARTZNER, P.; DELBAEN, F.; EBER, J.; HEATH, D. Coherent measures of risk. *Mathematical Finance*, 9, n. 3, p. 203–228, 1999. ISSN 0960-1627.

ASADI, S. Evolutionary fuzzification of ripper for regression: Case study of stock prediction. *Neurocomputing*, v. 331, p. 121–137, 2019. ISSN 0925-2312.

AUGERI, M. G.; GRECO, S.; NICOLOSI, V. Planning urban pavement maintenance by a new interactive multiobjective optimization approach. *European Transport Research Review*, Springer Science and Business Media LLC, v. 11, n. 1, mar. 2019.

B3. *B3 - Ibovespa*. 2021. <http://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/ibovespa.htm>. Accessed: 2021-09-10.

B3. *B3: Brasil, Bolsa, Balcão*. 2024. Disponível em: <http://www.b3.com.br/pt_br>.

BABAZADEH, H.; ESFAHANIPOUR, A. A novel multi period mean-var portfolio optimization model considering practical constraints and transaction cost. *Journal of Computational and Applied Mathematics*, v. 361, p. 313–342, 2019. ISSN 0377-0427.

BARBATI, M.; CORRENTE, S.; GRECO, S. A general space-time model for combinatorial optimization problems (and not only). *Omega*, v. 96, p. 102067, 2020. ISSN 0305-0483.

BEASLEY, J.; MEADE, N.; CHANG, T. An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research*, 148, n. 3, p. 621–643, 2003.

BEASLEY, J. E. Portfolio optimisation: Models and solution approaches. In: _____. *Theory Driven by Influential Applications*. [S.l.: s.n.], 2013. cap. Chapter 11, p. 201–221.

BENIDIS, K.; FENG, Y.; PALOMAR, D. P. Sparse Portfolios for High-Dimensional Financial Index Tracking. *IEEE Transactions on Signal Processing*, 66, n. 1, p. 155–170, 2018. ISSN 1053-587X.

BILBAO-TEROL, A.; ARENAS-PARRA, M.; CANAL-FERNANDEZ, V. A fuzzy multi-objective approach for sustainable investments. *Expert Systems With Applications*, 39, n. 12, p. 10904–10915, 2012. ISSN 0957-4174.

BILBAO-TEROL, A.; ARENAS-PARRA, M.; CAñAL-FERNáNDEZ, V. A fuzzy multi-objective approach for sustainable investments. *Expert Systems with Applications*, v. 39, n. 12, p. 10904–10915, 2012. ISSN 0957-4174.

BŁASZCZYŃSKI, J.; GRECO, S.; MATARAZZO, B.; SŁOWIŃSKI, R.; SZELĄG, M. jmaf - dominance-based rough set data analysis framework. In: *Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam*. [S.l.]: Springer Berlin Heidelberg, 2013. p. 185–209.

BŁASZCZYŃSKI, J.; GRECO, S.; SŁOWIŃSKI, R. Multi-criteria classification – a new scheme for application of dominance-based decision rules. *European Journal of Operational Research*, Elsevier BV, v. 181, n. 3, p. 1030–1044, set. 2007.

BŁASZCZYŃSKI, J.; GRECO, S.; SŁOWIŃSKI, R. Inductive discovery of laws using monotonic rules. *Engineering Applications of Artificial Intelligence*, Elsevier BV, v. 25, n. 2, p. 284–294, mar. 2012. Disponível em: <https://doi.org/10.1016/j.engappai.2011.09.003>.

BŁASZCZYŃSKI, J.; SŁOWIŃSKI, R.; SZELĄG, M. Sequential covering rule induction algorithm for variable consistency rough set approaches. *Information Sciences*, Elsevier BV, v. 181, n. 5, p. 987–1002, mar. 2011.

BORJI, A. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, v. 179, p. 41–65, 2019. ISSN 1077-3142.

BOROVIčKA, A. New complex fuzzy multiple objective programming procedure for a portfolio making under uncertainty. *Applied Soft Computing*, v. 96, 2020. ISSN 1568-4946.

BRANKE, J.; CORRENTE, S.; GRECO, S.; SłOWIńSKI, R.; ZIELNIEWICZ, P. Using choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research*, v. 250, n. 3, p. 884–901, 2016. ISSN 0377-2217.

BRUNI, R.; CESARONE, F.; SCOZZARI, A.; TARDELLA, F. A linear risk-return model for enhanced indexation in portfolio optimization. *OR Spectrum*, Springer Science and Business Media LLC, v. 37, n. 3, p. 735–759, nov. 2015.

CANAKGOZ, N. A.; BEASLEY, J. E. Mixed-integer programming approaches for index tracking and enhanced indexation. *European Journal of Operational Research*, 196, n. 1, p. 384–399, 2009. ISSN 0377-2217.

CHANDLER, P.; SWELLER, J. Cognitive load theory and the format of instruction. *Cognition and Instruction*, Informa UK Limited, v. 8, n. 4, p. 293–332, dez. 1991.

CHANG, T.-J.; MEADE, N.; BEASLEY, J.; SHARAIHA, Y. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, v. 27, n. 13, p. 1271 – 1302, 2000. ISSN 0305-0548.

CHIAM, S.; TAN, K.; MAMUN, A. A. Dynamic index tracking via multi-objective evolutionary algorithm. *Applied Soft Computing*, v. 13, n. 7, p. 3392–3408, 2013.

CHICA, M.; BAUTISTA, J.; CORDóN Óscar; DAMAS, S. A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega*, v. 58, p. 55–68, 2016.

CHICA, M.; CORDóN Óscar; DAMAS, S.; BAUTISTA, J. A robustness information and visualization model for time and space assembly line balancing under uncertain demand. *International Journal of Production Economics*, v. 145, n. 2, p. 761–772, 2013.

CORRENTE, S.; GRECO, S.; MATARAZZO, B.; SŁOWIńSKI, R. Explainable interactive evolutionary multiobjective optimization. *Omega*, v. 122, p. 102925, 2024. ISSN 0305-0483. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305048323000890>.

de Lima Silva, D. F.; FERREIRA, L.; de Almeida-Filho, A. T. A new preference disaggregation topsis approach applied to sort corporate bonds based on financial statements and expert's assessment. *Expert Systems with Applications*, v. 152, p. 113369, 2020. ISSN 0957-4174.

de Lima Silva, D. F.; FERREIRA, L.; de Almeida Filho, A. T. Preference disaggregation on topsis for sorting applied to an economic freedom assessment. *Expert Systems with Applications*, v. 215, p. 119341, 2023.

DEB, K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, v. 186, n. 2, p. 311–338, 2000. ISSN 0045-7825.

DEB, K. *Multi-Objective Optimization using Evolutionary Algorithms*. [S.l.]: John Wiley & Sons, 2001.

Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002.

DEB, K.; ZHU, L.; KULKARNI, S. Handling multiple scenarios in evolutionary multiobjective numerical optimization. *IEEE Transactions on Evolutionary Computation*, v. 22, n. 6, p. 920–933, 2018.

DING, Y.; PENG, Q.; SONG, Z.; CHEN, H. Variable selection and regularization via arbitrary rectangle-range generalized elastic net. *Statistics and Computing*, Springer Science and Business Media LLC, v. 33, n. 3, 2023.

FABOZZI, F. J.; KOLM, P. N.; PACHAMANOVA, D. A.; FOCARDI, S. M. *Robust Portfolio Optimization and Management*. Hoboken, New Jersey: John Wiley & Sons, 2007.

FAMA, E. F. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, v. 25, n. 2, p. 383, 1970.

FASTRICH, B.; PATERLINI, S.; WINKER, P. Cardinality versus q-norm constraints for index tracking. *Quantitative Finance*, 14, n. 11, p. 2019–2032, 2014. ISSN 1469-7688.

FERNANDES, B.; STREET, A.; VALLADãO, D.; FERNANDES, C. An adaptive robust portfolio optimization model with loss constraints based on data-driven polyhedral uncertainty sets. *European Journal of Operational Research*, v. 255, n. 3, p. 961–970, 2016. ISSN 0377-2217.

FERNANDEZ, E.; FIGUEIRA, J. R.; NAVARRO, J. a theoretical look at ordinal classification methods based on comparing actions with limiting boundaries between adjacent classes. *arXiv*, jul 2021. Disponível em: <http://arxiv.org/abs/2107.03440>.

FERNANDEZ, E.; NAVARRO, J.; SOLARES, E.; COELLO, C. C. A novel approach to select the best portfolio considering the preferences of the decision maker. *Swarm and Evolutionary Computation*, v. 46, p. 140 – 153, 2019. ISSN 2210-6502.

FERNáNDEZ-LORENZO, S.; PORRAS, D.; GARCíA-RIPOLL, J. J. Hybrid quantum–classical optimization with cardinality constraints and applications to finance. *Quantum Science and Technology*, IOP Publishing, v. 6, n. 3, p. 034010, jun 2021. Disponível em: <https://dx.doi.org/10.1088/2058-9565/abf9af>.

Ferreira, F. G. D. C.; Hanaoka, G. P.; Paiva, F. D.; Cardoso, R. T. N. Parallel moeas for combinatorial multiobjective optimization model of financial portfolio selection. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2018.

FERREIRA, L.; BORENSTEIN, D.; RIGHI, M. B.; de Almeida Filho, A. T. A fuzzy hybrid integrated framework for portfolio optimization in private banking. *Expert Systems with Applications*, v. 92, p. 350 – 362, 2018. ISSN 0957-4174.

FILHO, A. T. de A.; SILVA, J. C. S.; SILVA, D. F. de L.; FERREIRA, L. Preference learning applied to credit rating: Applications and perspectives. In: _____. *Intelligent Decision Support Systems : Combining Operations Research and Artificial Intelligence - Essays in Honor of Roman Słowiński*. Cham: Springer International Publishing, 2022. p. 121–137. ISBN 978-3-030-96318-7. Disponível em: <https://doi.org/10.1007/978-3-030-96318-7_7>.

FILIPPI, C.; GUASTAROBA, G.; SPERANZA, M. A heuristic framework for the bi-objective enhanced index tracking problem. *Omega*, v. 65, p. 122–137, 2016.

GAIVORONSKI, A.; KRYLOV, S.; WIJST, N. van der. Optimal portfolio selection and dynamic benchmark tracking. *European Journal of Operational Research*, 163, n. 1, p. 115–131, 2005. ISSN 0377-2217.

GARCIA, F.; GUIJARRO, F.; OLIVER, J. Index tracking optimization with cardinality constraint: a performance comparison of genetic algorithms and tabu search heuristics. *Neural Computing & Applications*, 30, n. 8, p. 2625–2641, 2018. ISSN 0941-0643.

GARCíA, F.; GUIJARRO, F.; MOYA, I. The curvature of the tracking frontier: A new criterion for the partial index tracking problem. *Mathematical and Computer Modelling*, v. 54, n. 7, p. 1781–1784, 2011. ISSN 0895-7177. Mathematical models of addictive behaviour, medicine & engineering.

GHOSH, S.; BHATTACHARYA, S. A data-driven understanding of covid-19 dynamics using sequential genetic algorithm based probabilistic cellular automata. *Applied Soft Computing*, v. 96, p. 106692, 2020.

GIUZIO, M. Genetic algorithm versus classical methods in sparse index tracking. *Decisions in Economics and Finance*, 40, n. 1-2, SI, p. 243–256, 2017. ISSN 1593-8883.

GIUZIO, M.; FERRARI, D.; PATERLINI, S. Sparse and robust normal and t- portfolios by penalized Lq-likelihood minimization. *European Journal of Operational Research*, 250, n. 1, p. 251–261, 2016. ISSN 0377-2217.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N.; WEINBERGER, K. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. v. 27. Disponível em: <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.

GRAHAM, D. I.; CRAVEN, M. J. An exact algorithm for small-cardinality constrained portfolio optimisation. *Journal of the Operational Research Society*, Taylor & Francis, v. 72, n. 6, p. 1415–1431, 2021.

GRECO, S.; EHRGOTT, M.; FIGUEIRA, J. R. (Ed.). *Multiple Criteria Decision Analysis: State of the Art Surveys*. 2nd. ed. New York, NY: Springer New York, 2016. v. 233.

GRECO, S.; KADZIńSKI, M. Feature cluster: Learning perspectives in multiple criteria decision analysis. *European Journal of Operational Research*, v. 264, n. 2, p. 403–404, 2018. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221717307774>.

GRECO, S.; MATARAZZO, B.; ROMAN, S. Dominance-based rough set approach to decision under uncertainty and time preference. *Annals OR*, v. 176, p. 41–75, 04 2010.

GRECO, S.; MATARAZZO, B.; SLOWINSKI, R. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, v. 129, n. 1, p. 1–47, 2001.

GRECO, S.; MATARAZZO, B.; SŁOWIŃSKI, R. Dominance-based rough set approach to interactive multiobjective optimization. In: _____. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 121–155. ISBN 978-3-540-88908-3.

GRECO, S.; MATARAZZO, B.; SŁOWIŃSKI, R. Dominance-based rough set approach to interactive evolutionary multiobjective optimization. In: *Preferences and Decisions*. [S.l.]: Springer Berlin Heidelberg, 2010. p. 225–260.

GRISHINA, N.; LUCAS, C. A.; DATE, P. Prospect theory-based portfolio optimization: an empirical study and analysis using intelligent algorithms. *Quantitative Finance*, 17, n. 3, p. 353–367, 2017. ISSN 1469-7688.

GUASTAROBA, G.; SPERANZA, M. G. Kernel Search: An application to the index tracking problem. *European Journal of Operational Research*, 217, n. 1, p. 54–68, 2012. ISSN 0377-2217.

GULRAJANI, I.; AHMED, F.; ARJOVSKY, M.; DUMOULIN, V.; COURVILLE, A. *Improved Training of Wasserstein GANs*. 2017.

HEATON, J. B.; POLSON, N. G.; WITTE, J. H. *Deep Learning in Finance*. 2018.

HOLLAND, J. *Adaptation in Natural and Artificial Systems*. [S.l.]: University of Michigan Press, 1975.

HU, S.; LI, F.; LIU, Y.; WANG, S. A self-adaptive preference model based on dynamic feature analysis for interactive portfolio optimization. *International Journal of Machine Learning and Cybernetics*, Springer Science and Business Media LLC, v. 11, n. 6, p. 1253–1266, nov. 2019.

HYNDMAN, R. J.; FAN, Y. Sample quantiles in statistical packages. *The American Statistician*, JSTOR, v. 50, n. 4, p. 361, nov. 1996.

ISLAM, M. A.; KOWAL, M.; JIA, S.; DERPANIS, K. G.; BRUCE, N. D. B. *Position, Padding and Predictions: A Deeper Look at Position Information in CNNs*. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2101.12322>.

JABBAR, A.; LI, X.; OMAR, B. A survey on generative adversarial networks: Variants, applications, and training. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 54, n. 8, oct 2021. ISSN 0360-0300.

JACQUET-LAGRèZE, E.; SISKOS, Y. Preference disaggregation: 20 years of mcda experience. *European Journal of Operational Research*, v. 130, n. 2, p. 233–245, 2001. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221700000357>.

JORION, P. Portfolio optimization with tracking-error constraints. *Financial Analysts Journal*, 59, n. 5, p. 70–82, 2003. ISSN 0015-198X.

KADZIńSKI, M.; TOMCZYK, M. K.; SłOWIńSKI, R. Preference-based cone contraction algorithms for interactive evolutionary multiple objective optimization. *Swarm and Evolutionary Computation*, v. 52, 2020. ISSN 2210-6502.

KALAYCI, C. B.; ERTENLICE, O.; AKBAY, M. A. A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, v. 125, p. 345 – 368, 2019. ISSN 0957-4174.

KARAKAYA, G.; SAKAR, C. T. A new interactive algorithm for continuous multiple criteria problems: A portfolio optimization example. *International Journal of Information Technology & Decision Making*, v. 20, n. 01, p. 371–398, 2021.

KATOCH, S.; CHAUHAN, S. S.; KUMAR, V. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, Springer Science and Business Media LLC, v. 80, n. 5, p. 8091–8126, out. 2021.

KIM, S.; KIM, S. Index tracking through deep latent representation learning. *Quantitative Finance*, v. 20, n. 4, p. 639–652, April 2020.

KöKSALAN, M.; ŞAKAR, C. T. An interactive approach to stochastic programming-based portfolio optimization. *Annals of Operations Research*, Springer Science and Business Media LLC, v. 245, n. 1-2, p. 47–66, set. 2016.

KOLM, P. N.; TUETUENCUE, R.; FABOZZI, F. J. 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234, n. 2, p. 356–371, 2014.

KONNO, H.; YAMAZAKI, H. Mean-absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock-market. *Management Science*, 37, n. 5, p. 519–531, 1991. ISSN 0025-1909.

KWAK, Y.; SONG, J.; LEE, H. Neural network with fixed noise for index-tracking portfolio optimization. *Expert Systems with Applications*, v. 183, p. 115298, 2021. ISSN 0957-4174.

LI, K.; LIAO, M.; DEB, K.; MIN, G.; YAO, X. Does preference always help? a holistic study on preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation*, v. 24, n. 6, p. 1078–1096, 2020.

LI, Q.; BAO, L. Enhanced index tracking with multiple time-scale analysis. *Economic Modelling*, v. 39, p. 282–292, 2014. ISSN 0264-9993.

LI, Q.; BAO, L.; ZHANG, Q. L. Multi-scale tracking dynamics and optimal index replication. *Applied Economics Letters*, Routledge, v. 21, p. 252–256, 2014.

LI, Q.; SUN, L.; BAO, L. Enhanced index tracking based on multi-objective immune algorithm. *Expert Systems with Applications*, v. 38, n. 5, p. 6101–6106, 2011. ISSN 0957-4174.

LI, X. P.; SHI, Z.-L.; LEUNG, C.-S.; SO, H. C. Sparse index tracking with k-sparsity or ε-deviation constraint via ℓ_0-norm minimization. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–14, 2022.

LIAGKOURAS, K.; METAXIOTIS, K. Examining the effect of different configuration issues of the multiobjective evolutionary algorithms on the efficient frontier formulation for the constrained portfolio optimization problem. *Journal Of The Operational Research Society*, 69, n. 3, p. 416–438, 2018.

LIN, Y.-C.; CHEN, C.-T.; SANG, C.-Y.; HUANG, S.-H. Multiagent-based deep reinforcement learning for risk-shifting portfolio management. *Applied Soft Computing*, v. 123, p. 108894, 2022. ISSN 1568-4946.

LIU, Y.; LIN, Y.; SONG, X.; LIU, C.; LIU, S. Nonnegative group bridge and application in financial index tracking. *Statistical Papers*, Springer Science and Business Media LLC, 2023.

MARIANI, G.; ZHU, Y.; LI, J.; SCHEIDEGGER, F.; ISTRATE, R.; BEKAS, C.; MALOSSI, A. C. I. *PAGAN: Portfolio Analysis with Generative Adversarial Networks*. 2019.

MARKOWITZ, H. Portfolio selection. *The Journal of Finance*, v. 7, n. 1, p. 77–91, 1952.

MARKOWITZ, H. *Portfolio Selection - Efficient Diversification of Investment*. New York: Wiley, 1959.

MELLO, T. H. de; BAYRAKSAN, G. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, v. 19, n. 1, p. 56–85, 2014. ISSN 1876-7354.

MENDONçA, G. H.; FERREIRA, F. G.; CARDOSO, R. T.; MARTINS, F. V. Multi-attribute decision making applied to financial portfolio optimization problem. *Expert Systems with Applications*, v. 158, 2020. ISSN 0957-4174.

MESQUITA-CUNHA, M.; FIGUEIRA, J. R.; BARBOSA-PóVOA, A. P. New ϵ-constraint methods for multi-objective integer linear programming: A pareto front representation approach. *European Journal of Operational Research*, 2022. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221722006142>.

MIRZA, M.; OSINDERO, S. *Conditional Generative Adversarial Nets*. arXiv, 2014. Disponível em: <https://arxiv.org/abs/1411.1784>.

MISHRA, S. K.; PANDA, G.; MAJHI, R. A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. *Swarm and Evolutionary Computation*, v. 16, p. 38 – 51, 2014. ISSN 2210-6502.

MOHEBBI, N.; NAJAFI, A. A. Credibilistic multi-period portfolio optimization based on scenario tree. *Physica A: Statistical Mechanics and its Applications*, v. 492, p. 1302–1316, 2018.

MOUSSEAU, V.; SLOWINSKI, R. *Journal of Global Optimization*, Springer Science and Business Media LLC, v. 12, n. 2, p. 157–174, 1998. ISSN 0925-5001. Disponível em: <http://dx.doi.org/10.1023/A:1008210427517>.

MUTUNGE, P.; HAUGLAND, D. Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research*, 90, p. 33–41, 2018. ISSN 0305-0548.

NI, H.; WANG, Y. Stock index tracking by pareto efficient genetic algorithm. *Applied Soft Computing*, v. 13, n. 12, p. 4519–4535, 2013. ISSN 1568-4946.

NIKOLENKO, S. I. *Synthetic Data for Deep Learning*. 2019.

OMIDVAR, M. N.; LI, X.; YAO, X. A review of population-based metaheuristics for large-scale black-box global optimization—part i. *IEEE Transactions on Evolutionary Computation*, v. 26, n. 5, p. 802–822, 2022.

OUENNICHE, J.; PÉREZ-GLADISH, B.; BOUSLAH, K. An out-of-sample framework for topsis-based classifiers with application in bankruptcy prediction. *Technological Forecasting and Social Change*, v. 131, p. 111–116, 2018.

OUYANG, H.; ZHANG, X.; YAN, H. Index tracking based on deep neural network. *Cognitive Systems Research*, 57, p. 107–114, 2019. ISSN 1389-0417.

PARDALOS, P.; SISKOS, Y.; ZOPOUNIDIS, C. (Ed.). *Advances in Multicriteria Analysis*. Boston, MA: Springer US, 1995. v. 5.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In: WALLACH, H.; LAROCHELLE, H.; BEYGELZIMER, A.; ALCHé-BUC, F. d'; FOX, E.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems 32*. [S.l.]: Curran Associates, Inc., 2019. p. 8024–8035.

PAWLAK, Z. Rough sets. *International Journal of Computer & Information Sciences*, Springer Science and Business Media LLC, v. 11, n. 5, p. 341–356, out. 1982. Disponível em: <https://doi.org/10.1007/bf01001956>.

Purshouse, R. C.; Deb, K.; Mansor, M. M.; Mostaghim, S.; Wang, R. A review of hybrid evolutionary multiple criteria decision making methods. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2014. p. 1147–1154.

RADFORD, A.; METZ, L.; CHINTALA, S. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. arXiv, 2015. Disponível em: <https://arxiv.org/abs/1511.06434>.

RESENDE, M. G. C.; RIBEIRO, C. C. *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. New York: Springer, 2016.

ROCKAFELLAR, R. T.; URYASEV, S. Optimization of conditional value-at-risk. *Journal of Risk*, v. 2, p. 21–41, 2000.

ROLL, R. A Mean-variance analysis of tracking error - minimizing the volatility of tracking error will not produce a more efficient managed portfolio. *Journal of Portfolio Management*, 18, n. 4, p. 13–22, 1992. ISSN 0095-4918.

ROY, B. *Multicriteria methodology for decision aiding*. [S.l.]: Kluwer Academic Publishers, 1996. ISBN 9781441947611.

RUDOLF, M.; WOLTER, H.; ZIMMERMANN, H. A linear model for tracking error minimization. *Journal of Banking & Finance*, 23, n. 1, p. 85–103, 1999. ISSN 0378-4266.

RUIZ-TORRUBIANO, R.; SUAREZ, A. A hybrid optimization approach to index tracking. *Annals of Operations Research*, 166, n. 1, p. 57–71, 2009. ISSN 0254-5330.

SALVATORE, G.; MATARAZZO, B.; SŁOWIŃSKI, R. Beyond markowitz with multiple criteria decision aiding. *Journal of Business Economics*, Springer Science and Business Media LLC, v. 83, n. 1, p. 29–60, fev. 2013.

SANT'ANA, L. R.; CALDEIRA, J. F.; FILOMENA, T. P. Lasso-based index tracking and statistical arbitrage long-short strategies. *North American Journal of Economics and Finance*, 51, 2020. ISSN 1062-9408.

SANT'ANNA, L. R.; FILOMENA, T. P.; CALDEIRA, J. F.; BORENSTEIN, D. Investigating the use of statistical process control charts for index tracking portfolios. *Journal Of The Operational Research Society*, 70, n. 10, SI, p. 1622–1638, 2019. ISSN 0160-5682.

SANT'ANNA, L. R.; FILOMENA, T. P.; CALDEIRA, J. F.; BORENSTEIN, D. Investigating the use of statistical process control charts for index tracking portfolios. *Journal of the Operational Research Society*, Taylor & Francis, v. 70, n. 10, p. 1622–1638, 2019.

SANT'ANNA, L. R.; FILOMENA, T. P.; GUEDES, P. C.; BORENSTEIN, D. Index tracking with controlled number of assets using a hybrid heuristic combining genetic algorithm and non-linear programming. *Annals of Operations Research*, 258, n. 2, p. 849–867, 2017.

SCHNAUBELT, M. Deep reinforcement learning for the optimal placement of cryptocurrency limit orders. *European Journal of Operational Research*, v. 296, n. 3, p. 993–1006, 2022. ISSN 0377-2217.

SCOZZARI, A.; TARDELLA, F.; PATERLINI, S.; KRINK, T. Exact and heuristic approaches for the index tracking problem with UCITS constraints. *Annals of Operations Research*, 205, n. 1, p. 235–250, 2013. ISSN 0254-5330.

SHAPIRO, A.; MELLO, T. H. de. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, Springer Science and Business Media LLC, v. 81, n. 3, p. 301–325, maio 1998.

SHEN, K.-Y.; LO, H.-W.; TZENG, G.-H. Interactive portfolio optimization model based on rough fundamental analysis and rational fuzzy constraints. *Applied Soft Computing*, v. 125, p. 109158, 2022.

SHU, L.; SHI, F.; TIAN, G. High-dimensional index tracking based on the adaptive elastic net. *Quantitative Finance*, Informa UK Limited, v. 20, n. 9, p. 1513–1530, abr. 2020. Disponível em: <https://doi.org/10.1080/14697688.2020.1737328>.

SILVA, D. F. de L.; SILVA, J. C. S.; FILHO, A. T. de A. Aprendizagem de preferências através do método topsis: uma aplicação para ratings de liberdade econômica. In: *CSBC 2020 - SEMISH - Seminário Integrado de Software e Hardware*. [S.l.: s.n.], 2020.

SILVA, D. F. de L.; SILVA, J. C. S.; SILVA, L. G. O.; FERREIRA, L.; ALMEIDA-FILHO, A. T. de. Sovereign credit risk assessment with multiple criteria using an outranking method. *Mathematical Problems in Engineering*, Hindawi Limited, v. 2018, p. 1–11, set. 2018.

SILVA, J. C. S.; de Almeida Filho, A. T. Using gan-generated market simulations to guide genetic algorithms in index tracking optimization. *Applied Soft Computing*, v. 145, p. 110587, 2023. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494623006051>.

SILVA, J. C. S.; de Lima Silva, D. F.; Delgado Neto, A. de S.; FERRAZ, A.; MELO, J. L.; Ferreira Júnior, N. R.; de Almeida Filho, A. T. A city cluster risk-based approach for sars-cov-2 and isolation barriers based on anonymized mobile phone users' location data. *Sustainable Cities and Society*, v. 65, p. 102574, 2021. ISSN 2210-6707. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2210670720307927>.

SILVA, J. C. S.; de Lima Silva, D. F.; Ferreira Júnior, N. R.; de Almeida Filho, A. T. An analytical tool to support public policies and isolation barriers against sars-cov-2 based on mobility patterns and socio-economic aspects. *Applied Soft Computing*, v. 138, p. 110177, 2023. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494623001953>.

SILVA, J. C. S.; FILHO, A. T. d. A. Interactively learning rough strategies that dynamically satisfy investor's preferences in multiobjective index tracking. *IEEE Transactions on Evolutionary Computation*, p. 1–1, 2023.

SILVA, J. C. S.; FILHO, A. T. de A. Performing hierarchical bayesian regression to assess the best districts for building new residential real estate developments. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2020. p. 2411–2416.

SILVA, J. C. S.; FILHO, A. T. de A. A preference-driven imo-drsa approach to the index tracking problem. In: *INFORMS Annual Meeting*. [S.l.: s.n.], 2021.

SILVA, J. C. S.; FILHO, A. T. de A. A simulated imo-drsa approach for cognitive reduction in multiobjective financial portfolio interactive optimization. In: *2021 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2021. p. 1560–1568.

SILVA, J. C. S.; FILHO, A. T. de A. A systematic literature review on solution approaches for the index tracking problem. *IMA Journal of Management Mathematics*, n/a, n. n/a, 2023.

SILVA, J. C. S.; MACêDO, D.; ZANCHETTIN, C.; OLIVEIRA, A. L.; FILHO, A. T. de A. Multi-class mobile money service financial fraud detection by integrating supervised learning with adversarial autoencoders. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2021. p. 1–7.

SILVA, J. C. S.; SILVA, D. F. d. L.; FILHO, A. T. de A. An enhanced grasp approach for the index tracking problem. *International Transactions in Operational Research*, 2022. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.13163>.

SILVA, J. C. S.; SILVA, D. F. d. L.; FILHO, A. T. de A. An enhanced grasp approach for the index tracking problem. *International Transactions in Operational Research*, n/a, n. n/a, 2022.

SILVA, J. C. S.; SILVA, D. F. de L.; FERREIRA, L.; ALMEIDA-FILHO, A. T. de. A dominance-based rough set approach applied to evaluate the credit risk of sovereign bonds. *4OR*, Springer Science and Business Media LLC, jan. 2021.

SILVA, L. G. de O.; de Almeida-Filho, A. T. A new promethee-based approach applied within a framework for conflict analysis in evidence theory integrating three conflict measures. *Expert Systems with Applications*, v. 113, p. 223–232, 2018.

SLOWINSKI, R.; GRECO, S.; MATARAZZO, B. Rough set and rule-based multicriteria decision aiding. *Pesquisa Operacional*, FapUNIFESP (SciELO), v. 32, n. 2, p. 213–270, ago. 2012. Disponível em: <https://doi.org/10.1590/s0101-74382012000200001>.

S&P. *Standard & Poor's Global Ratings*. 2024. Disponível em: <https://www.standardandpoors.com/pt_LA/web/guest/home>.

Streichert, F.; Ulmer, H.; Zell, A. Evaluating a hybrid encoding and three crossover operators on the constrained portfolio selection problem. In: *2004 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2004. p. 932–939.

STRUB, O.; TRAUTMANN, N. A two-stage approach to the UCITS-constrained index-tracking problem. *Computers & Operations Research*, 103, p. 167–183, 2019. ISSN 0305-0548.

SWELLER, J. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, v. 12, n. 2, p. 257–285, 1988.

SWELLER, J.; MERRIENBOER, J. J. G. van; PAAS, F. G. W. C. *Educational Psychology Review*, Springer Science and Business Media LLC, v. 10, n. 3, p. 251–296, 1998.

TAS, E.; TURKAN, A. H. Regularized Index-tracking Optimal Portfolio Selection. *Economic Computation and Economic Cybernetics Studies and Research*, 52, n. 3, p. 135–146, 2018. ISSN 0424-267X.

THAKKAR, A.; CHAUDHARI, K. Information fusion-based genetic algorithm with long short-term memory for stock price and trend prediction. *Applied Soft Computing*, v. 128, p. 109428, 2022.

TLILI, A.; KHALED, O.; MOUSSEAU, V.; OUERDANE, W. Interactive portfolio selection involving multicriteria sorting models. *Annals of Operations Research*, 2022.

TOMCZYK, M. K.; KADZIńSKI, M. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation*, v. 24, n. 2, p. 320–334, 2020.

VIEIRA, E. B. F.; FILOMENA, T. P.; SANT'ANNA, L. R.; LEJEUNE, M. A. Liquidity-constrained index tracking optimization models. *Annals of Operations Research*, Springer Science and Business Media LLC, jul. 2021.

WANG, M.; XU, C.; XU, F.; XUE, H. A mixed 0-1 LP for index tracking problem with CVaR risk constraints. *Annals of Operations Research*, 196, n. 1, p. 591–609, 2012. ISSN 0254-5330.

WANG, M.; XU, F.; DAI, Y.-H. An index tracking model with stratified sampling and optimal allocation. *Applied Stochastic Models in Business and Industry*, 34, n. 2, p. 144–157, 2018. ISSN 1524-1904.

WU, L.; YANG, Y. Nonnegative Elastic Net and application in index tracking. *Applied Mathematics and Computation*, 227, p. 541–552, 2014. ISSN 0096-3003.

WU, L.; YANG, Y.; LIU, H. Nonnegative-lasso and application in index tracking. *Computational Statistics & Data Analysis*, 70, p. 116–126, 2014. ISSN 0167-9473.

WU, L. chuan; TSAI, I. chan. Three fuzzy goal programming models for index portfolios. *Journal of the Operational Research Society*, Taylor & Francis, v. 65, n. 8, p. 1155–1169, 2014.

XIN, B.; CHEN, L.; CHEN, J.; ISHIBUCHI, H.; HIROTA, K.; LIU, B. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*, v. 6, p. 41256–41279, 2018.

XU, F.; LU, Z.; XU, Z. An efficient optimization approach for a cardinality-constrained index tracking problem. *Optimization Methods & Software*, 31, n. 2, p. 258–271, 2016. ISSN 1055-6788.

YANG, T.; HUANG, X.; HONG, K. R. A new uncertain enhanced index tracking model with higher-order moment of the downside. *Soft Computing*, Springer Science and Business Media LLC, 2023.

YANG, Y.; WU, L. Nonnegative adaptive lasso for ultra-high dimensional regression models and a two-stage method applied in financial modeling. *Journal of Statistical Planning and Inference*, 174, p. 52–67, 2016. ISSN 0378-3758.

Yu, G.; Jin, Y.; Olhofer, M. References or preferences – rethinking many-objective evolutionary optimization. In: *2019 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2019. p. 2410–2417.

YU, Y.; MO, J.; DENG, Q.; ZHOU, C.; LI, B.; WANG, X.; YANG, N.; TANG, Q.; FENG, X. Memristor parallel computing for a matrix-friendly genetic algorithm. *IEEE Transactions on Evolutionary Computation*, v. 26, n. 5, p. 901–910, 2022.

ZHANG, C.; LIANG, S.; LYU, F.; FANG, L. Stock-index tracking optimization using auto-encoders. *Frontiers in Physics*, v. 8, p. 388, 2020. ISSN 2296-424X.

ZHAO, H.; CHEN, Z.-G.; ZHAN, Z.-H.; KWONG, S.; ZHANG, J. Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. *Neurocomputing*, v. 430, p. 58–70, 2021. ISSN 0925-2312.

ZHAO, K.; LIAN, H. The Expectation-Maximization approach for Bayesian quantile regression. *Computational Statistics & Data Analysis*, 96, p. 1–11, 2016. ISSN 0167-9473.

ZHAO, Y.; WANG, Y.; ZHANG, J.; FU, C.-W.; XU, M.; MORITZ, D. Kd-box: Line-segment-based kd-tree for interactive exploration of large-scale time-series data. *IEEE Transactions on Visualization and Computer Graphics*, v. 28, n. 1, p. 890–900, 2022.

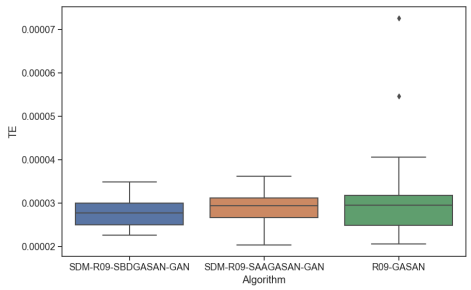
ZHENG, Y.; CHEN, B.; HOSPEDALES, T. M.; YANG, Y. Index tracking with cardinality constraints: A stochastic neural networks approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 34, n. 01, p. 1242–1249, Apr. 2020. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/5478>.

ZHOU, J.; HUA, Z. A correlation guided genetic algorithm and its application to feature selection. *Applied Soft Computing*, v. 123, p. 108964, 2022.

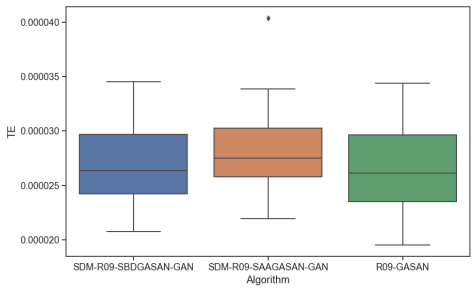
ZOPOUNIDIS, C.; GALARIOTIS, E.; DOUMPOS, M.; SARRI, S.; ANDRIOSOPOULOS, K. Multiple criteria decision aiding for finance: An updated bibliographic survey. *European Journal of Operational Research*, v. 247, n. 2, p. 339 – 348, 2015. ISSN 0377-2217.

APPENDIX A – FIGURES ASSOCIATED WITH THE GAN-BASED HYBRID GAS RESULTS

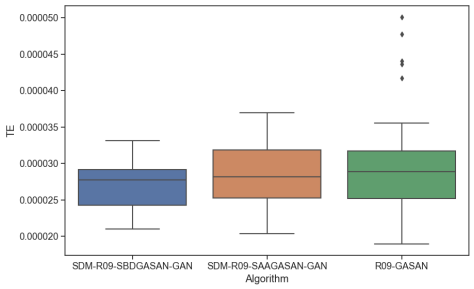
A.1 HYBRID GAS RESULTS



(a) R09-GASAN: $v = 10$ days

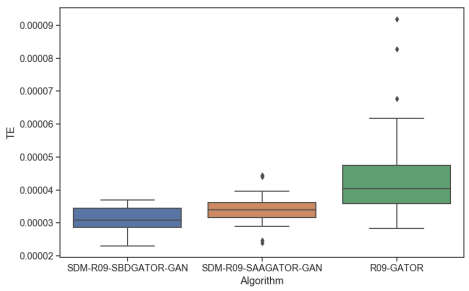


(b) R09-GASAN: $v = 20$ days

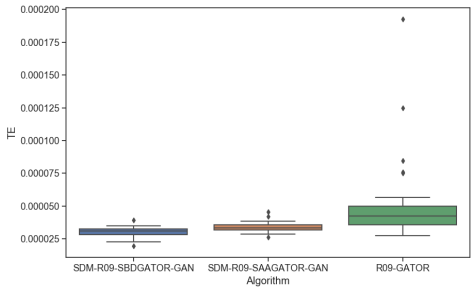


(c) R09-GASAN: $v = 40$ days

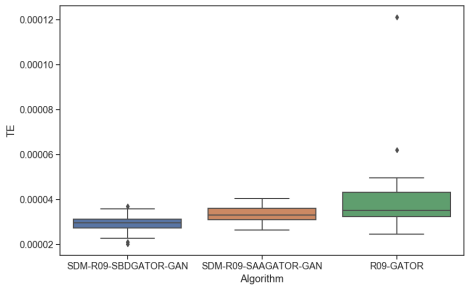
Figure 40 – Boxplots for the overall tracking error performance for the R09-GASAN hybrid GAs in the out-of-sample period.



(a) R09-GATOR: $v = 10$ days

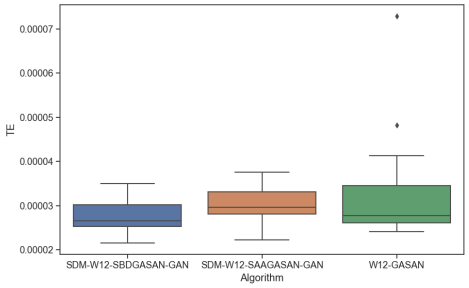


(b) R09-GATOR: $v = 20$ days

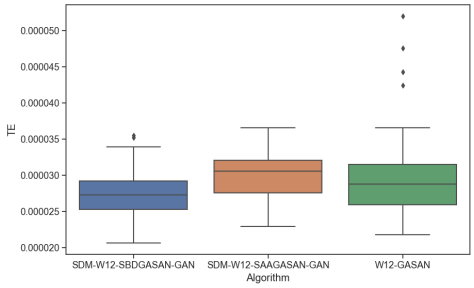


(c) R09-GATOR: $v = 40$ days

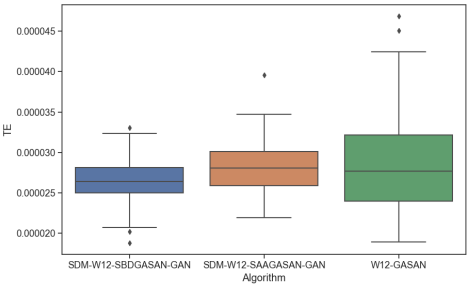
Figure 41 – Boxplots for the overall tracking error performance for the R09-GATOR hybrid GAs in the out-of-sample period.



(a) W12-GASAN: $v = 10$ days

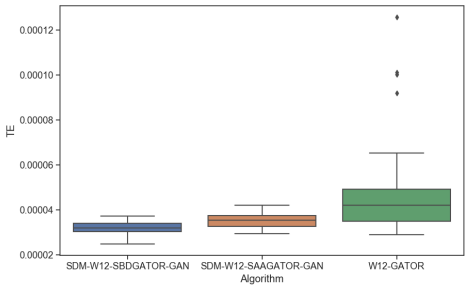


(b) W12-GASAN: $v = 20$ days

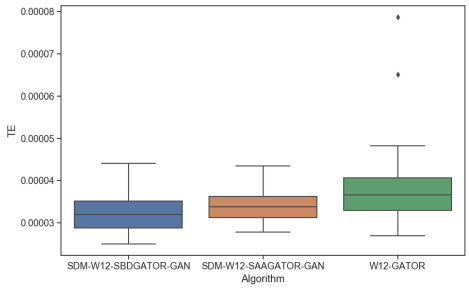


(c) W12-GASAN: $v = 40$ days

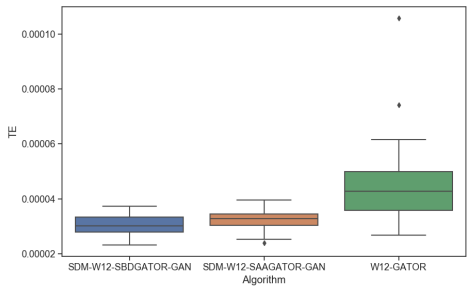
Figure 42 – Boxplots for the overall tracking error performance for the W12-GASAN hybrid GAs in the out-of-sample period.



(a) W12-GATOR: $v = 10$ days



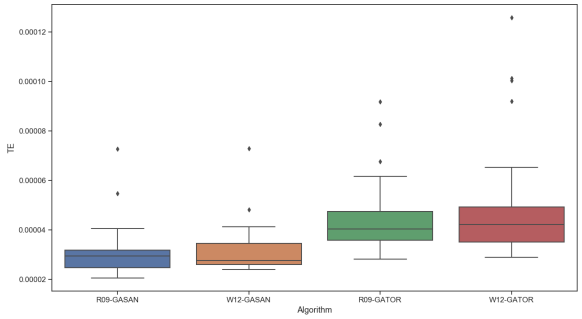
(b) W12-GATOR: $v = 20$ days



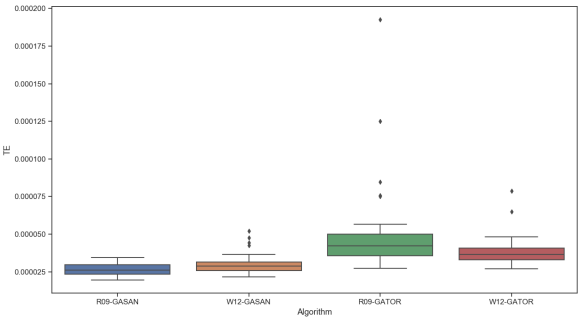
(c) W12-GATOR $v = 40$ days

Figure 43 – Boxplots for the overall tracking error performance for the W12-GATOR hybrid GAs in the out-of-sample period.

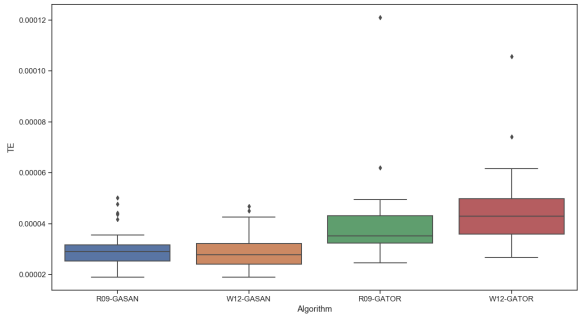
A.2 COMPARISON OF THE FOUR HYBRID GAS BASED ON HISTORICAL DATA FOR EACH REBALANCING STRATEGY



(a) $v = 10$ days



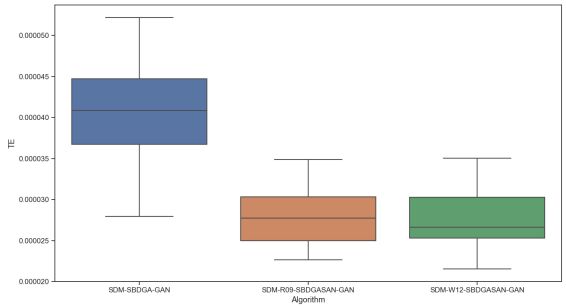
(b) $v = 20$ days



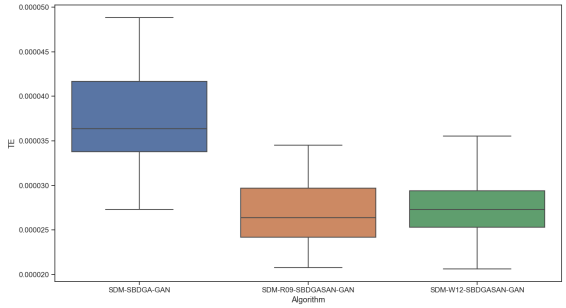
(c) $v = 40$ days

Figure 44 – Performance of hybrid GAs based on historical data for each rebalancing strategy.

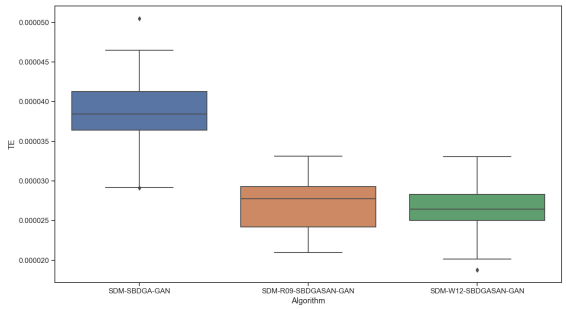
A.3 COMPARISON OF THE BEST HYBRID GAS BASED ON SBD AGAINST THE SDM-SBDGA-GAN



(a) $v = 10$ days



(b) $v = 20$ days

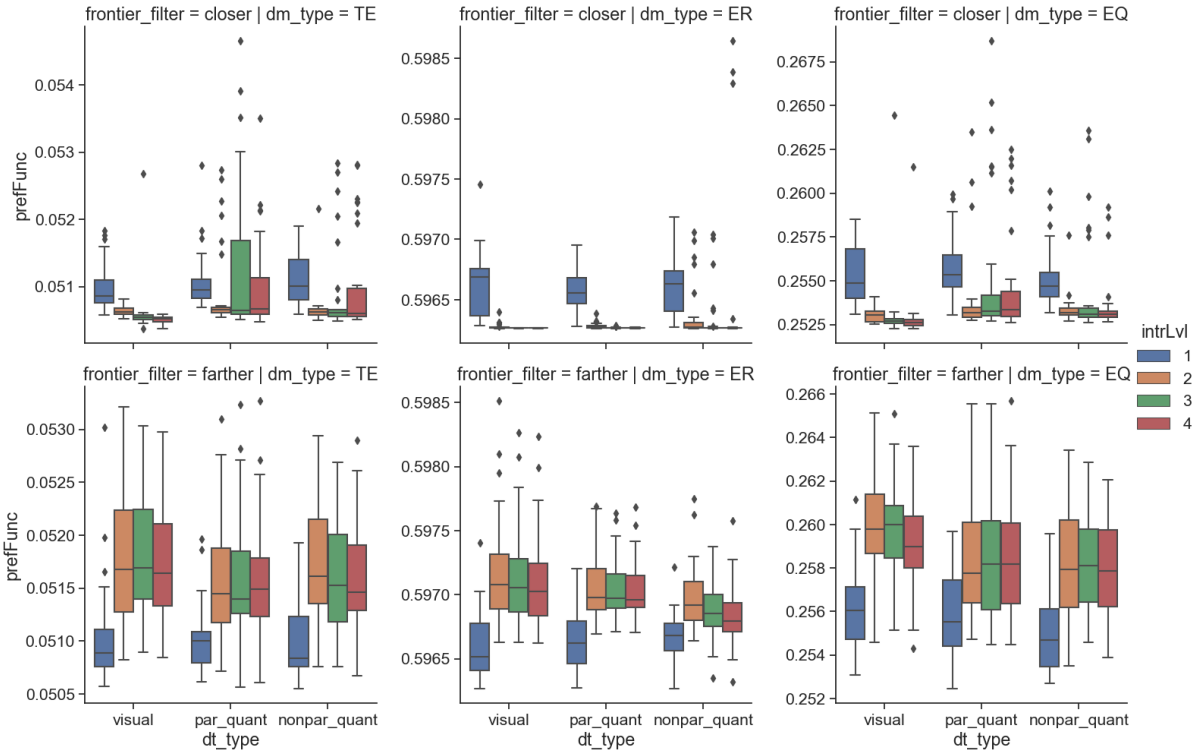


(c) $v = 40$ days

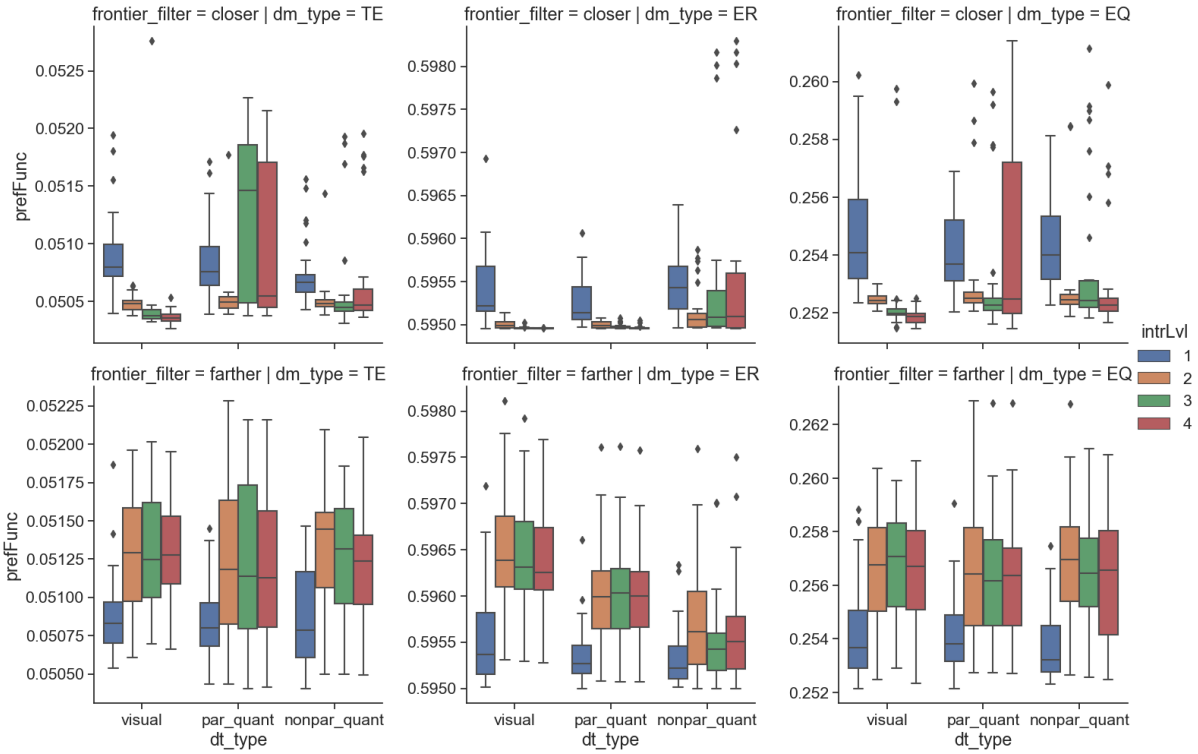
Figure 45 – Performance of the two best SBD hybrid GAs against the SDM-SBDGA-GAN for each rebalancing strategy.

APPENDIX B – FIGURES ASSOCIATED WITH IMO-DRSA COMPARISONS

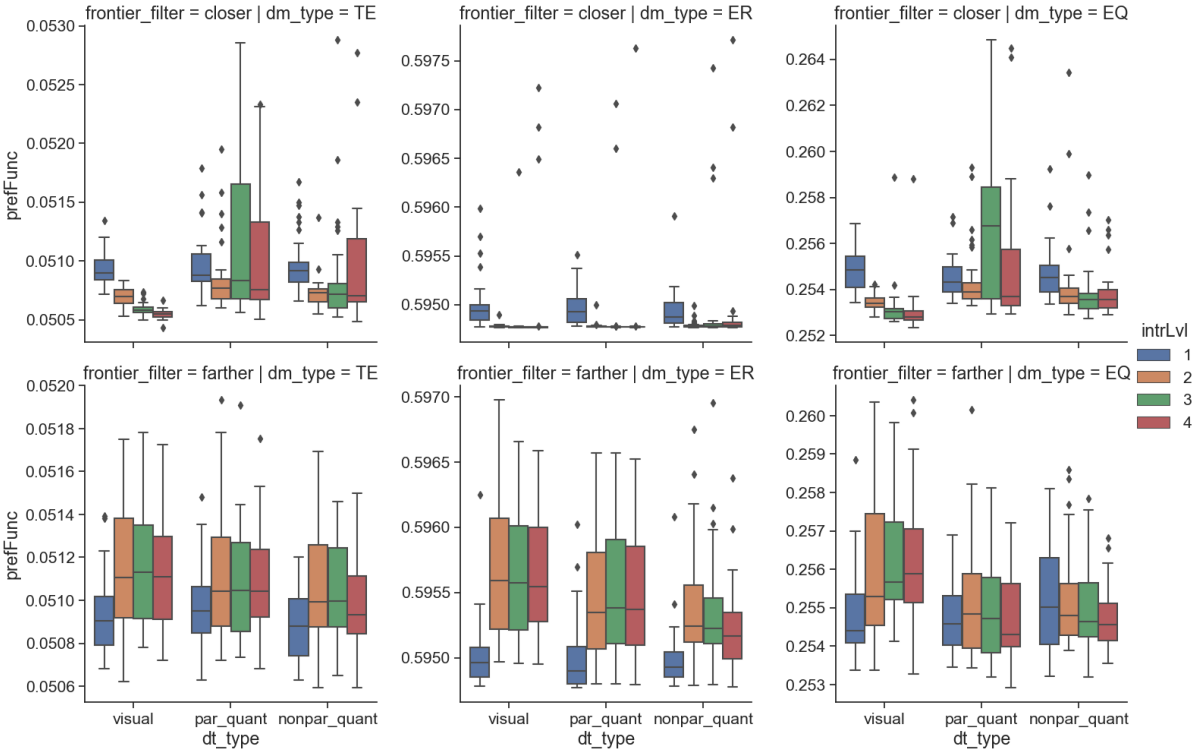
B.1 COMPARISON BETWEEN THE FRONTIER FILTERS AND DATA TABLE PRESENTATION METHODS IN SBH



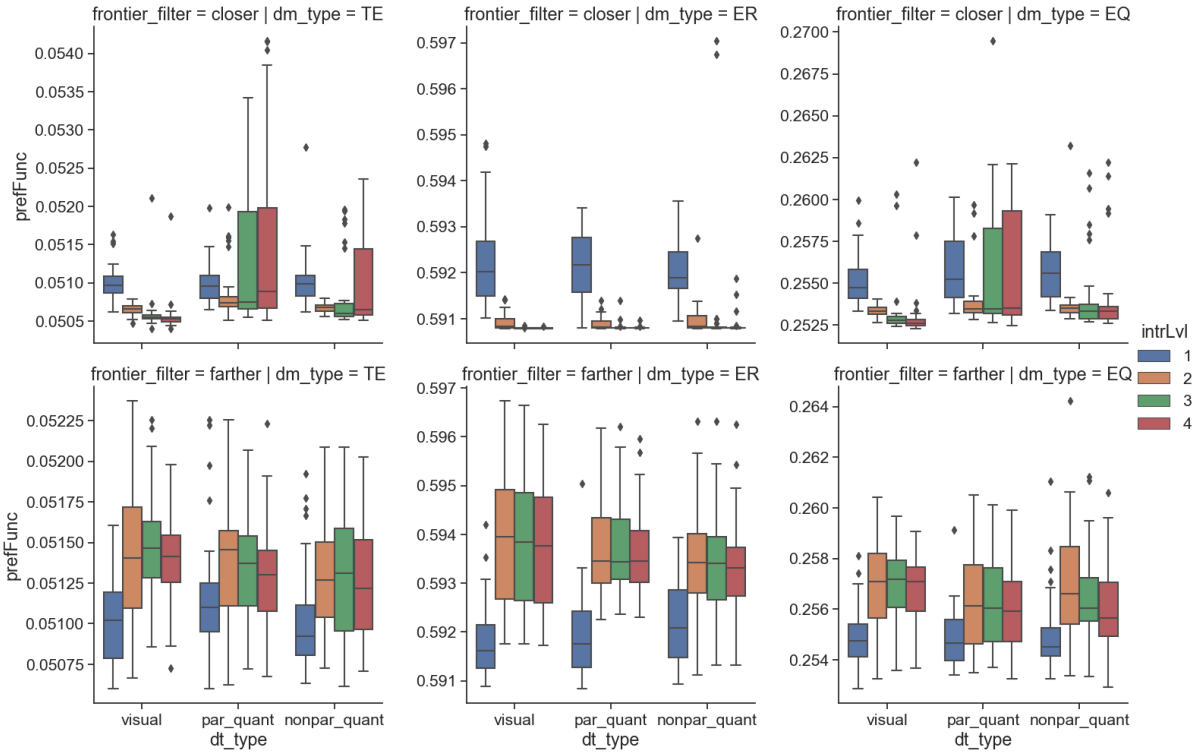
(a) indtrack1



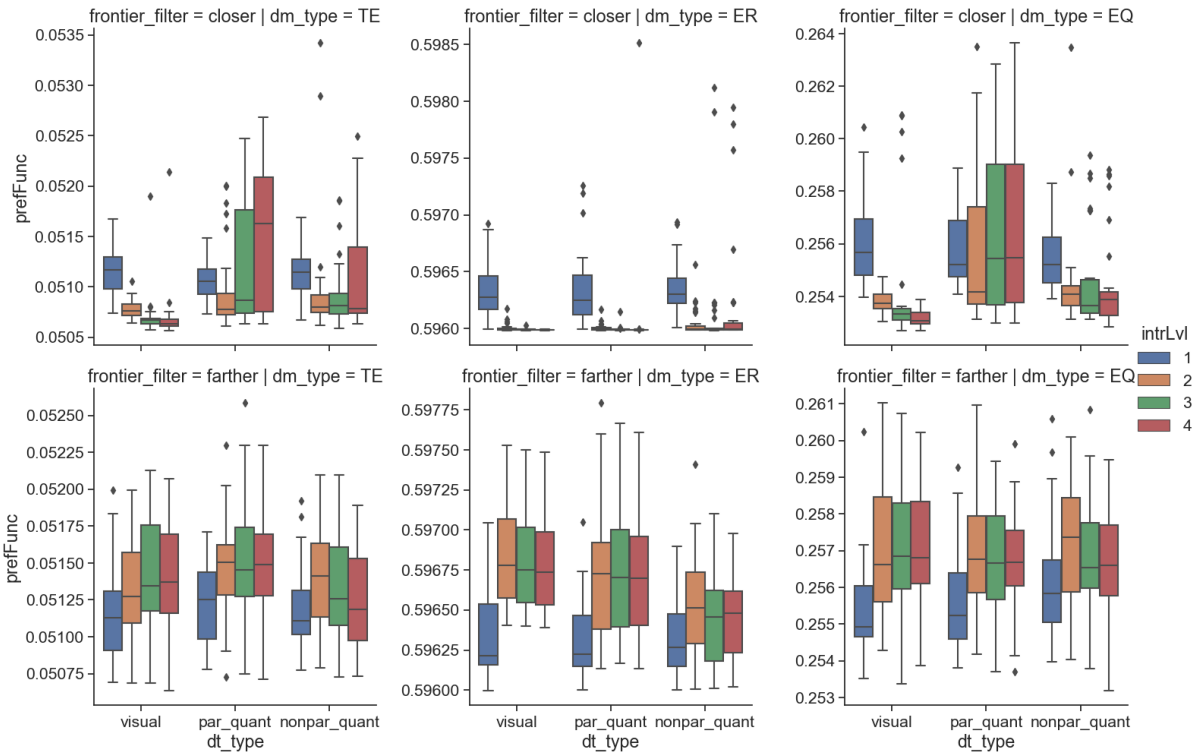
(b) indtrack2



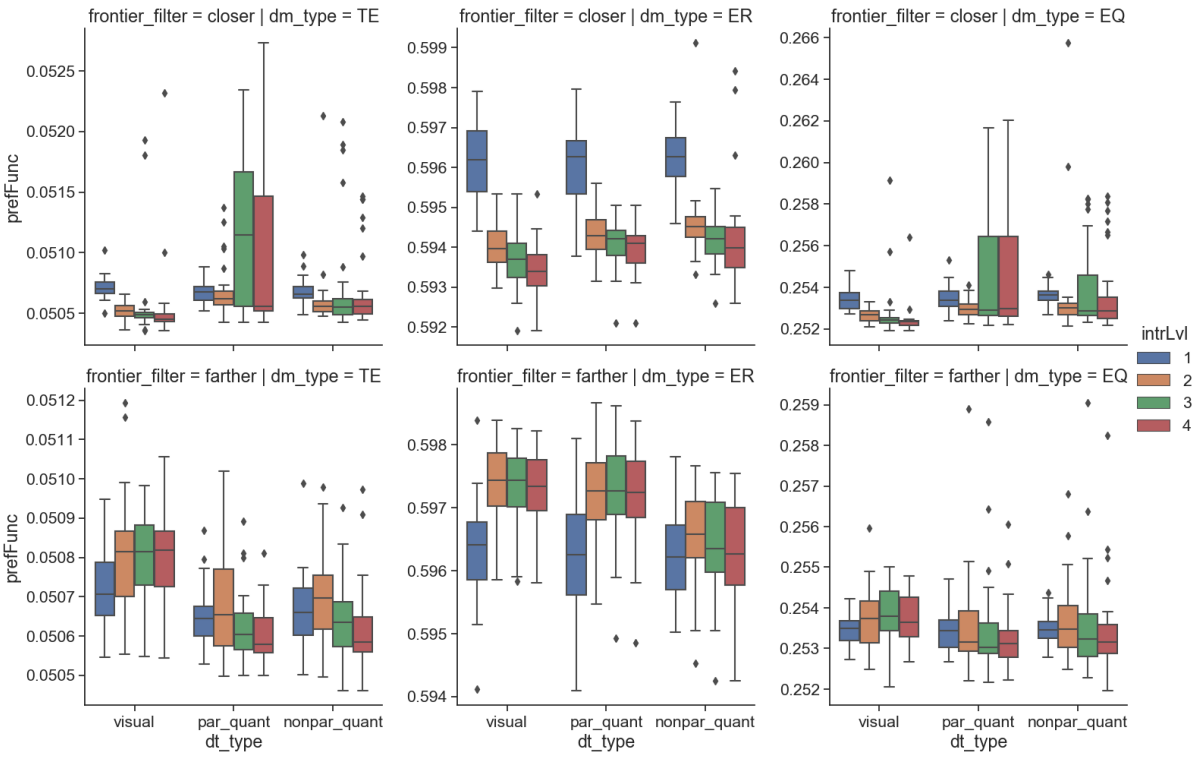
(c) indtrack3



(d) indtrack4

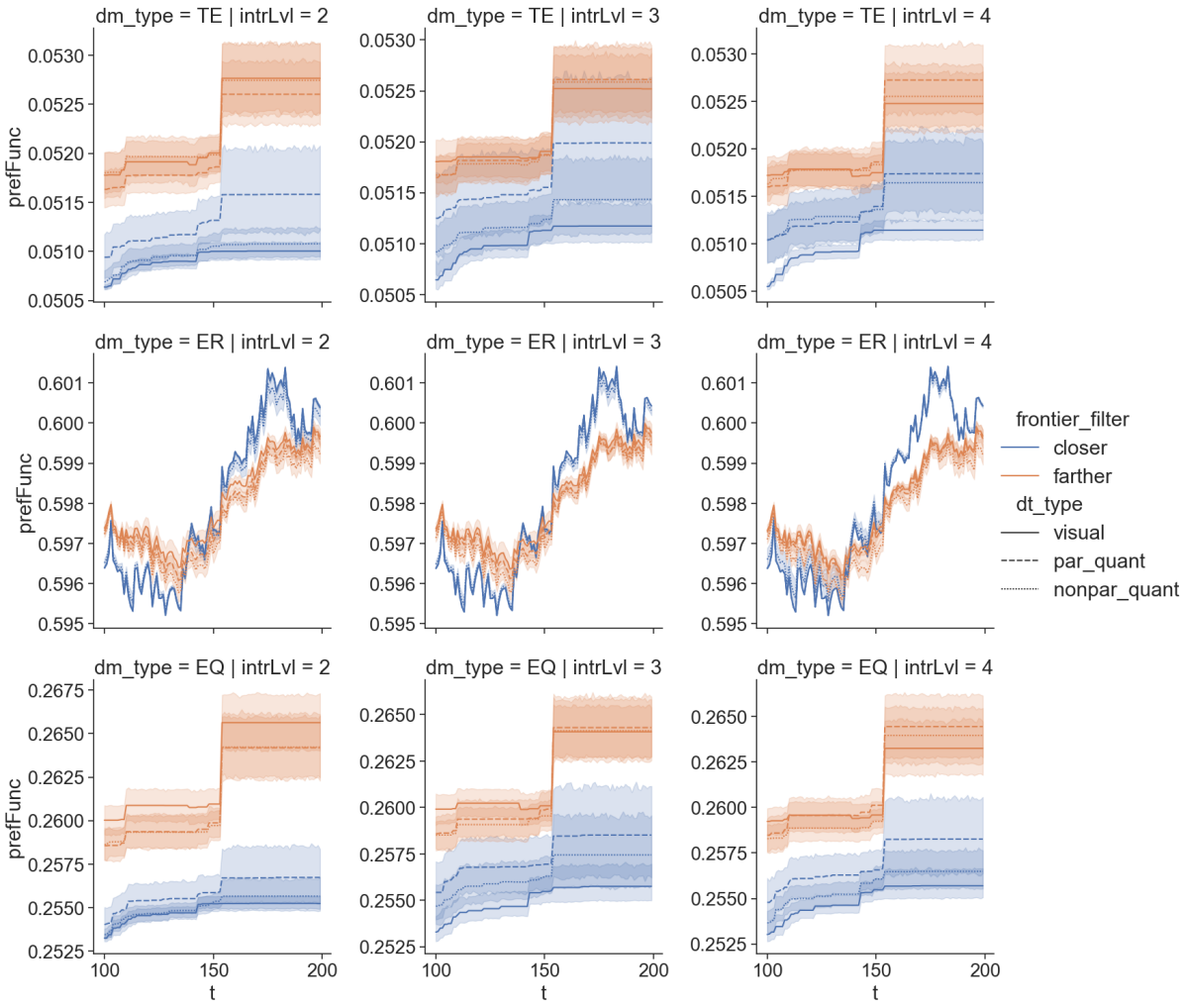


(e) indtrack5

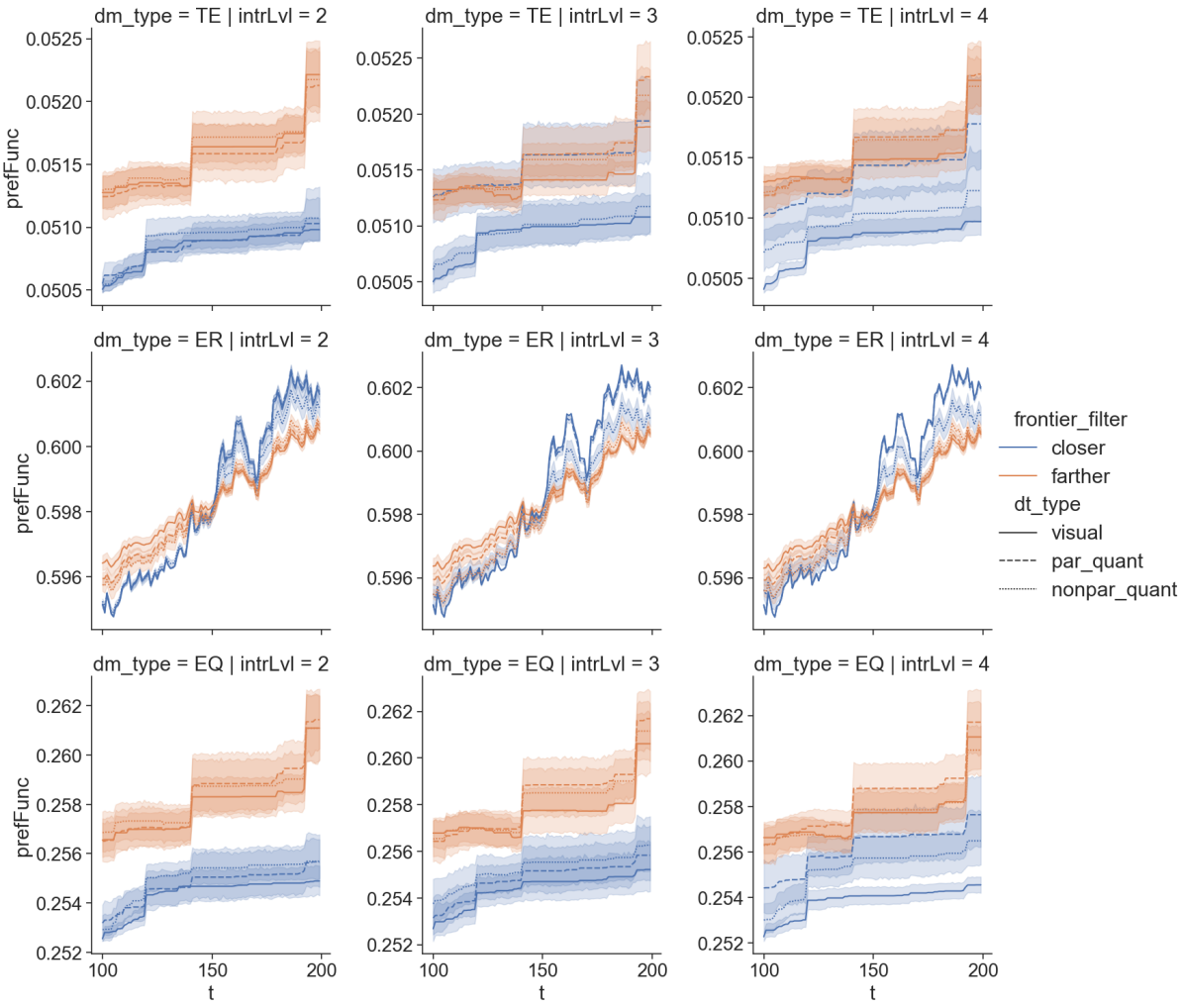


(f) indtrack6

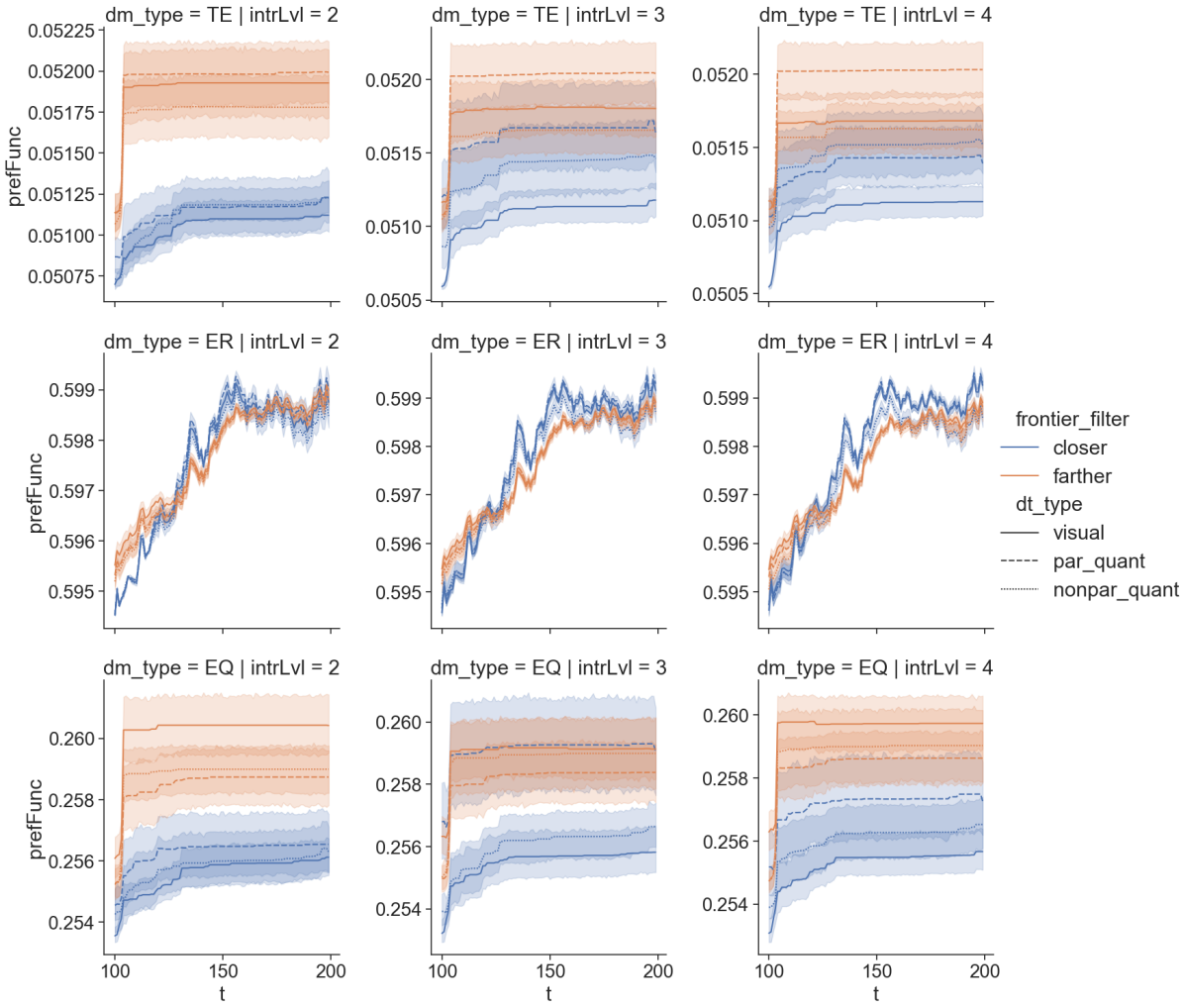
Figure 46 – Comparison of the preference function of three simulated investors with different frontier filters and data table presentation methods in each simulated interaction for the in-sample period of the six instances of the problem


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
frontier_filter
closer
farther
dt_type
visual
par_quant
nonpar_quant

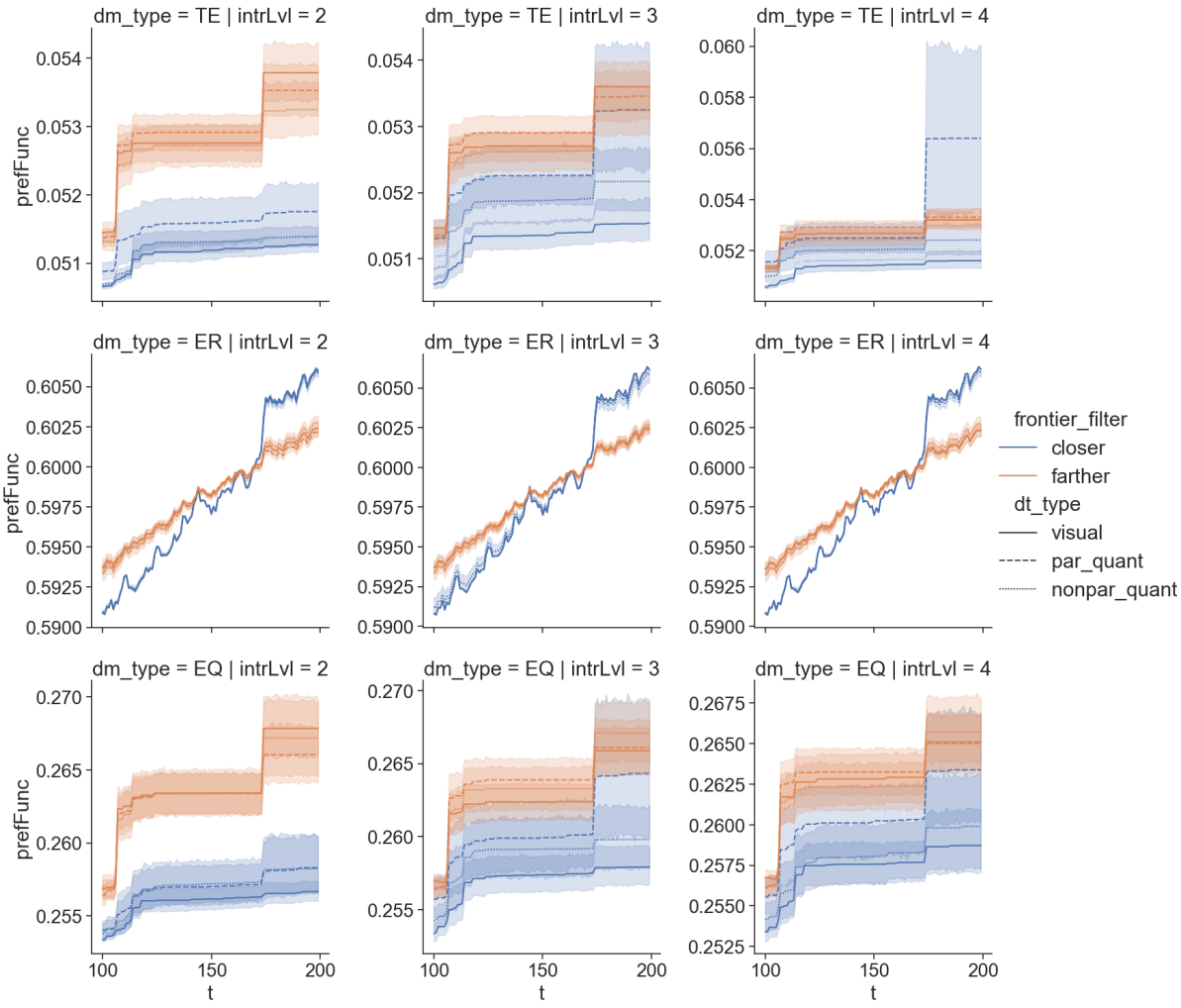
(a) indtrack1


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
frontier_filter
closer
farther
dt_type
visual
par_quant
nonpar_quant

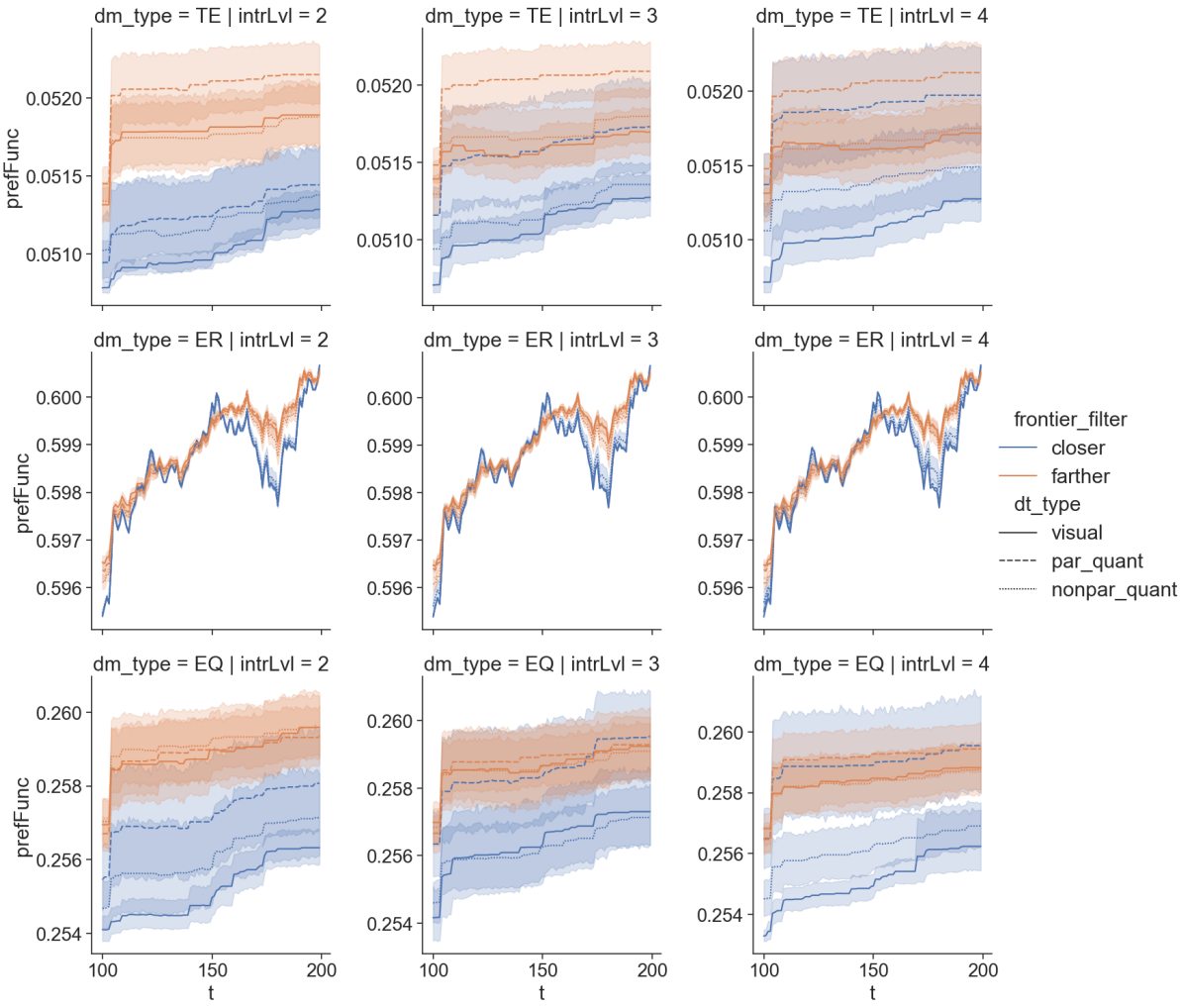
(b) indtrack2


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
frontier_filter
closer
farther
dt_type
visual
par_quant
nonpar_quant

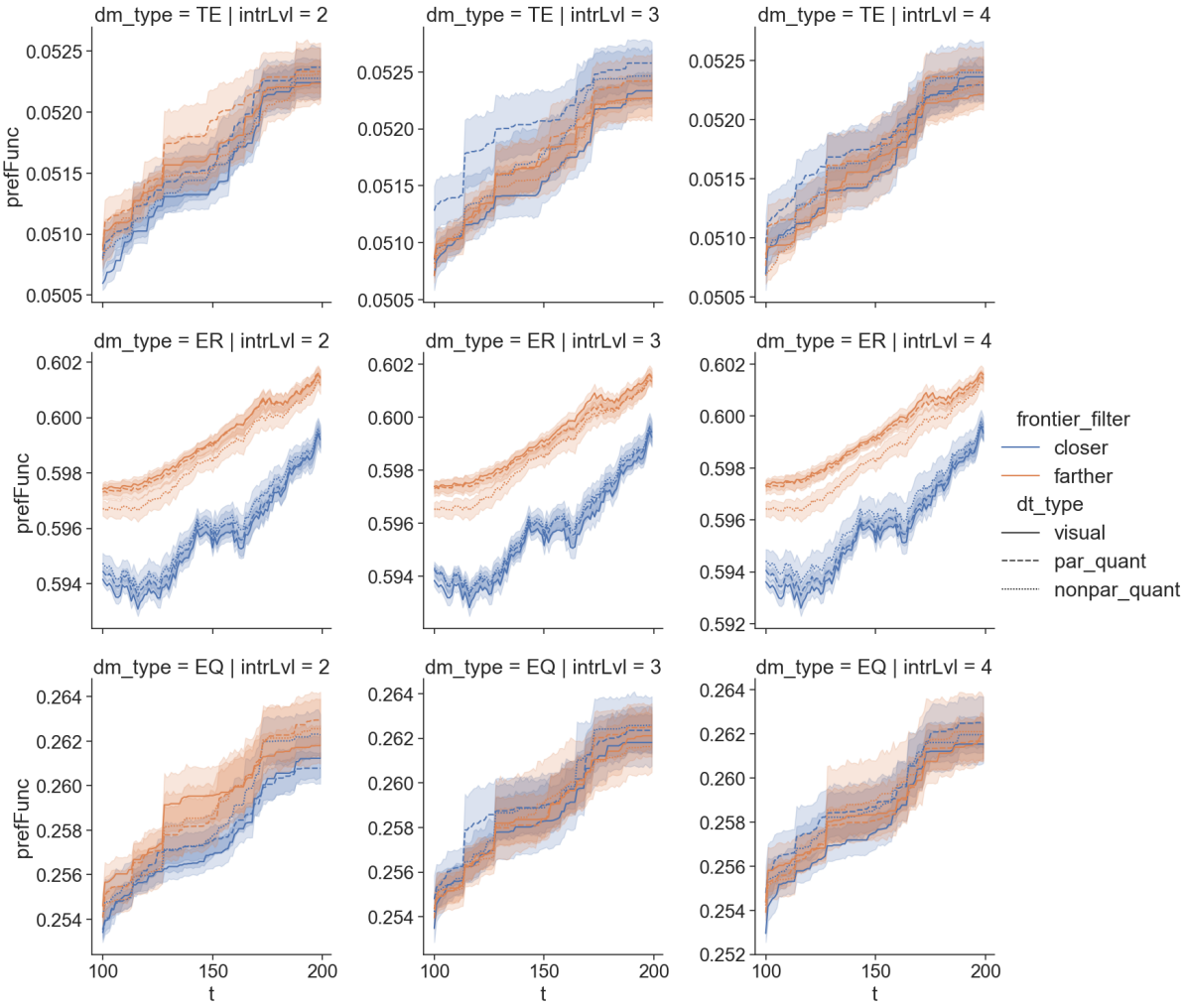
(c) indtrack3


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
frontier_filter
closer
farther
dt_type
visual
par_quant
nonpar_quant

(d) indtrack4


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
frontier_filter
closer
farther
dt_type
visual
par_quant
nonpar_quant

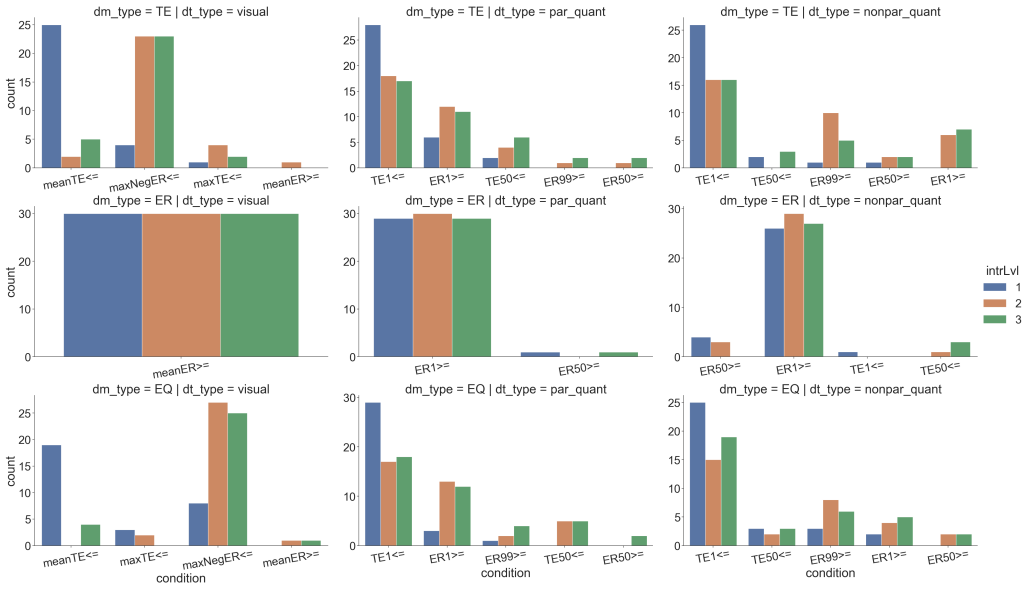
(e) indtrack5



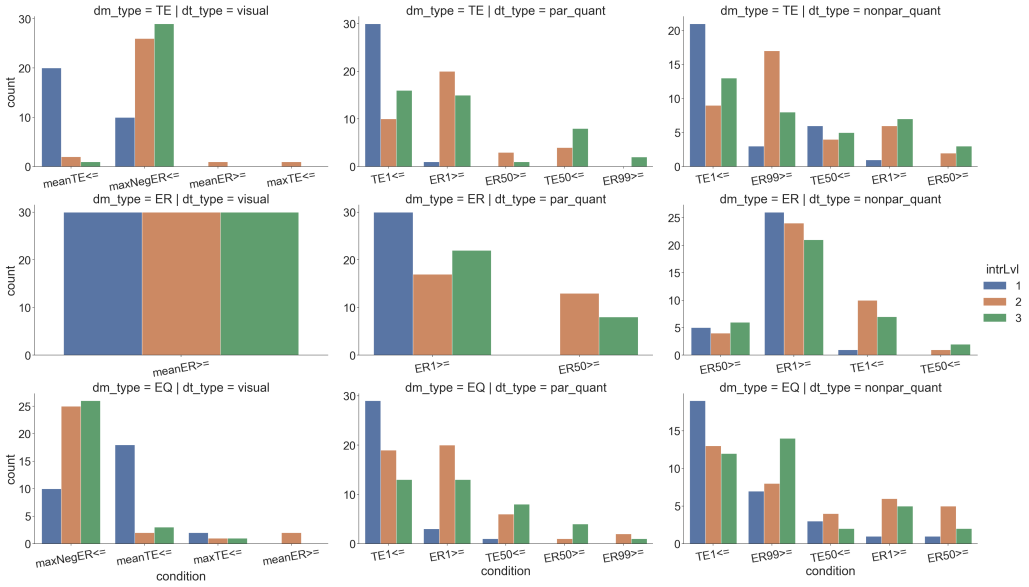
(f) indtrack6

Figure 47 – Comparison of the preference function of three simulated investors with different frontier filters and data table presentation methods in each simulated interaction for the out-of-sample period of the six instances of the problem

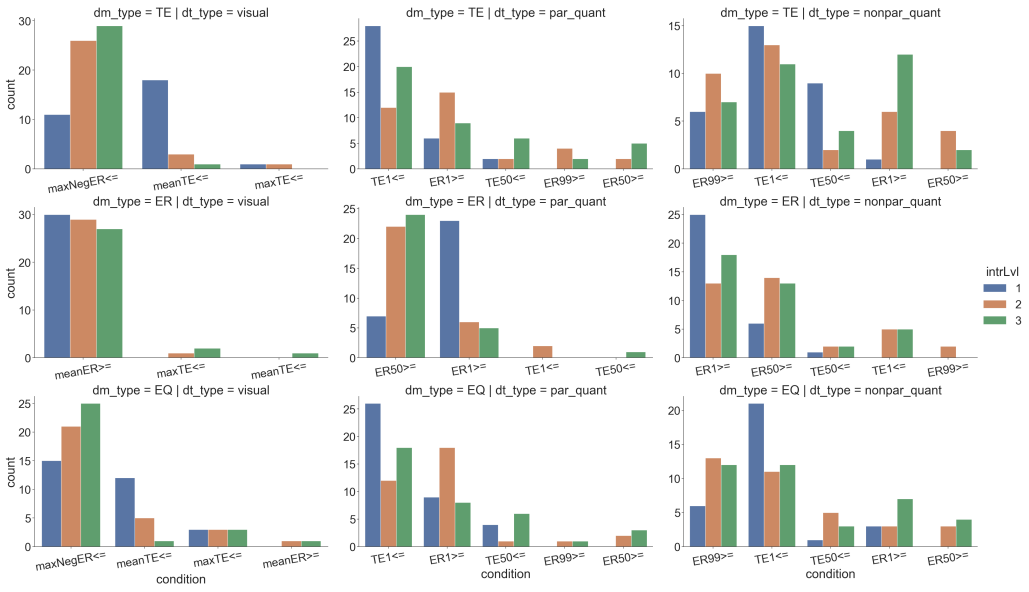
B.1.1 Importance of attributes along the interactions



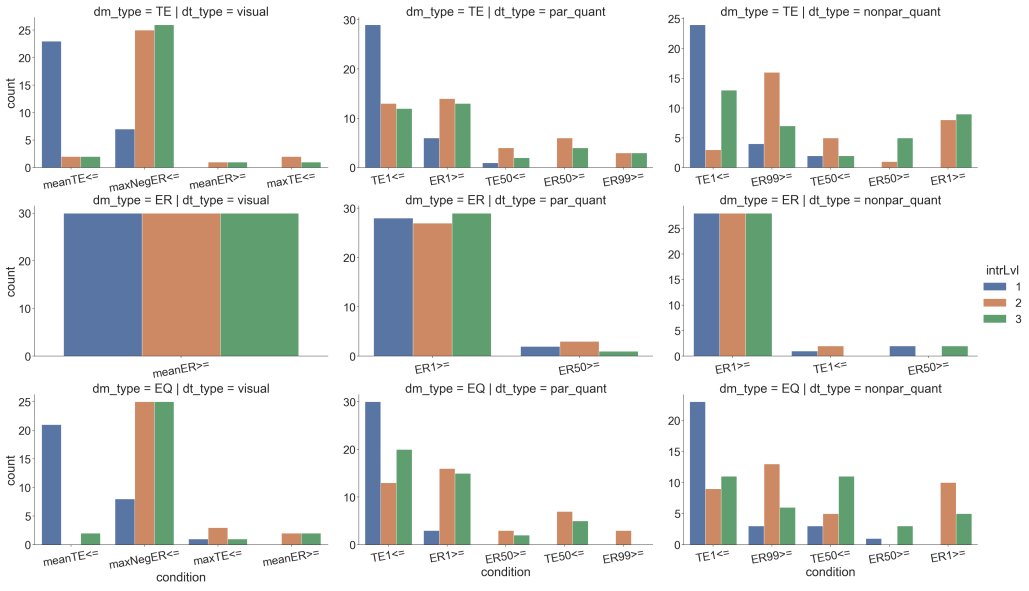
(a) indtrack1



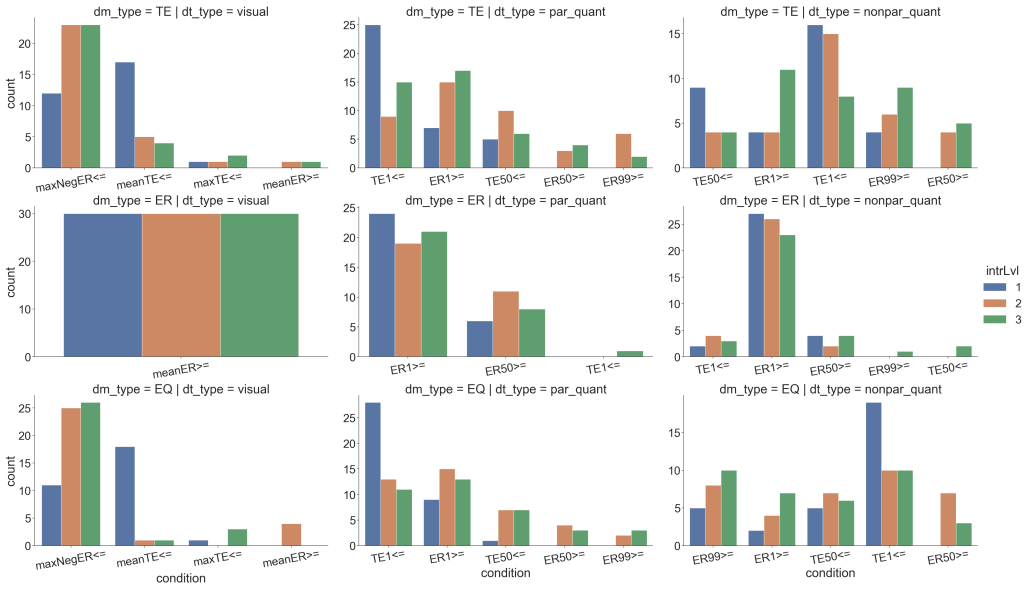
(b) indtrack2



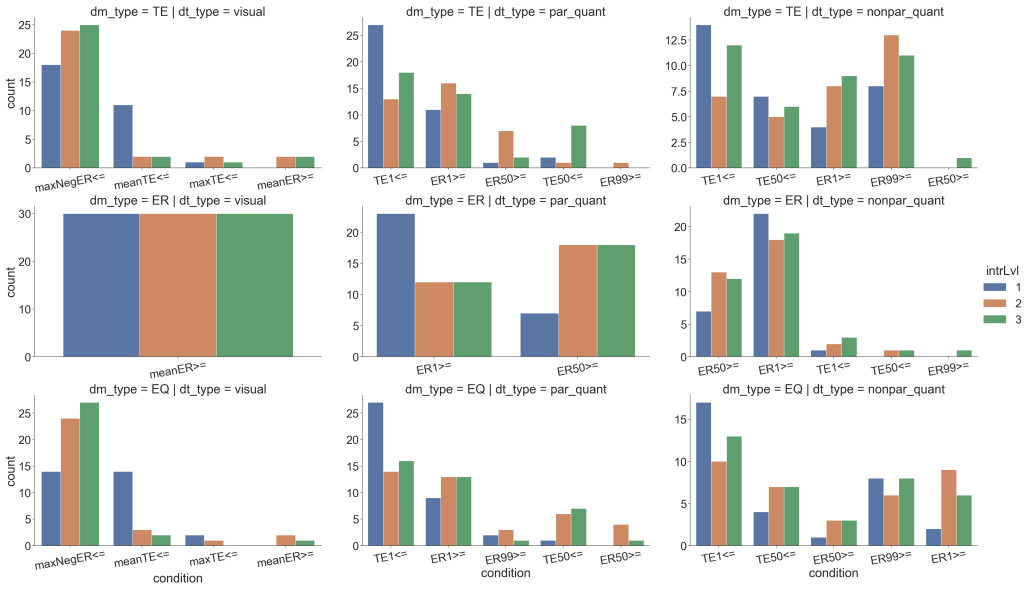
(c) indtrack3



(d) indtrack4



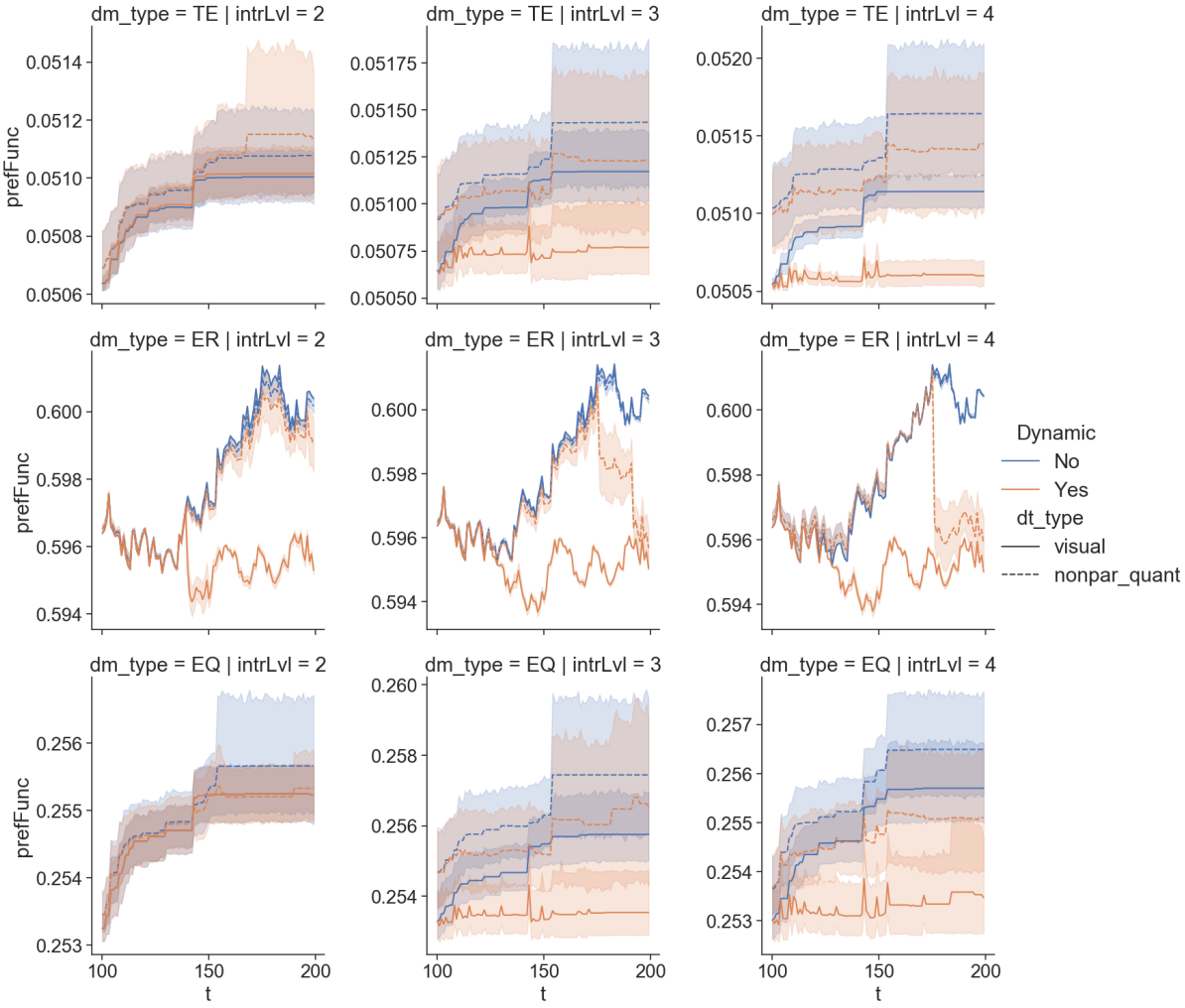
(e) indtrack5



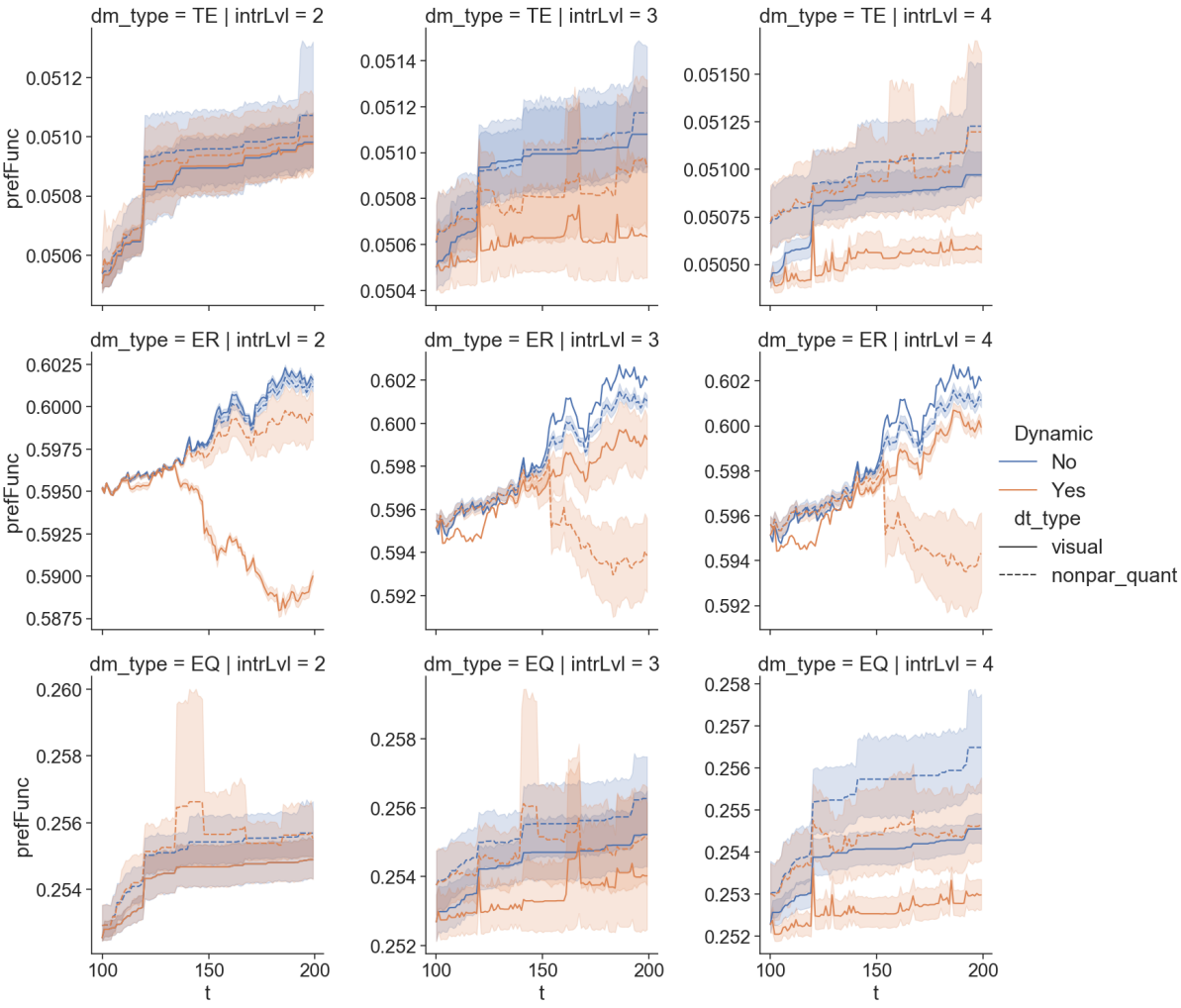
(f) indtrack6

Figure 48 – Objectives contained in rules along the interaction process when considering different types of investors for the six instances of the problem

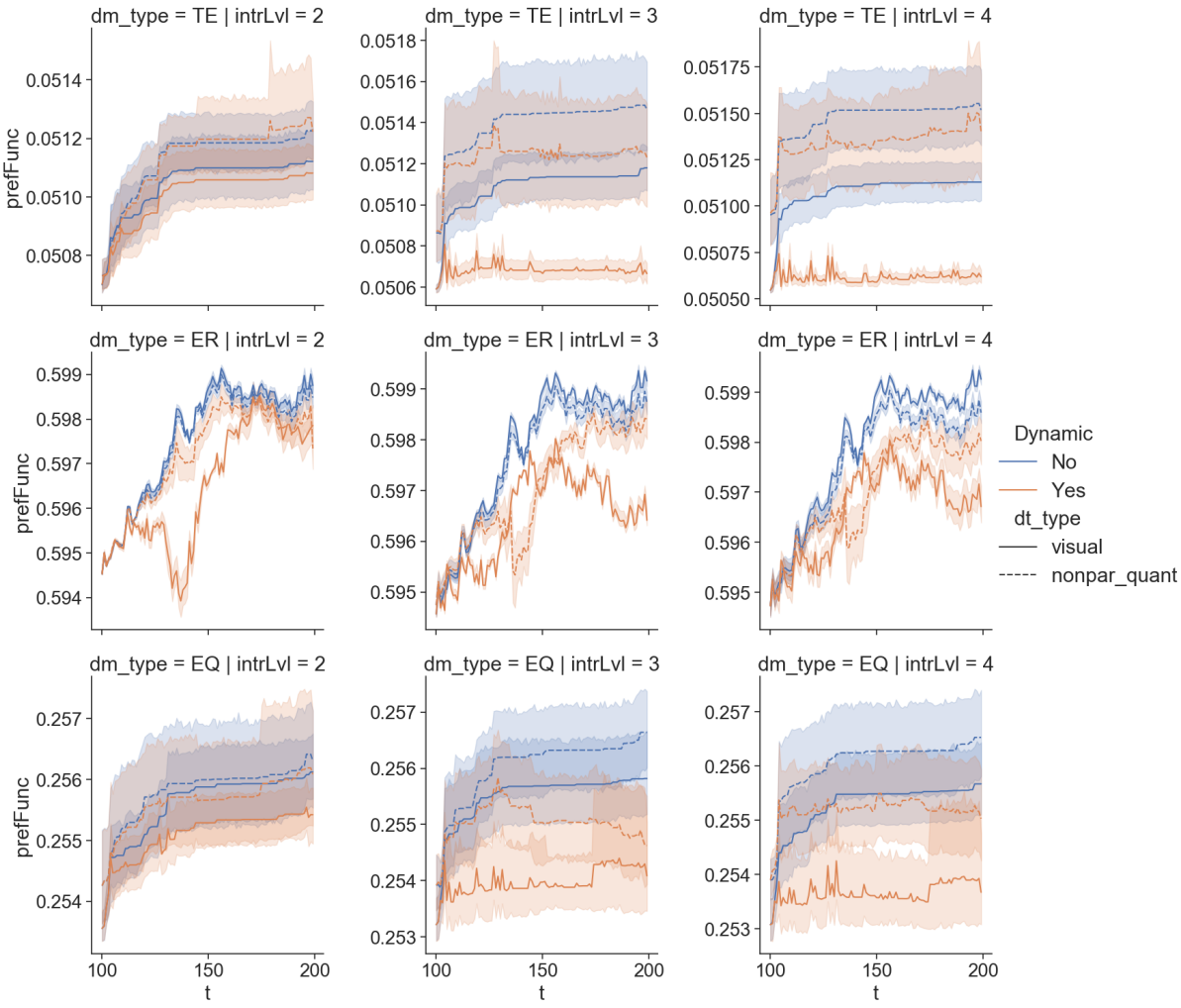
B.1.2 Comparison between SBH and preference-driven MBH rebalancing



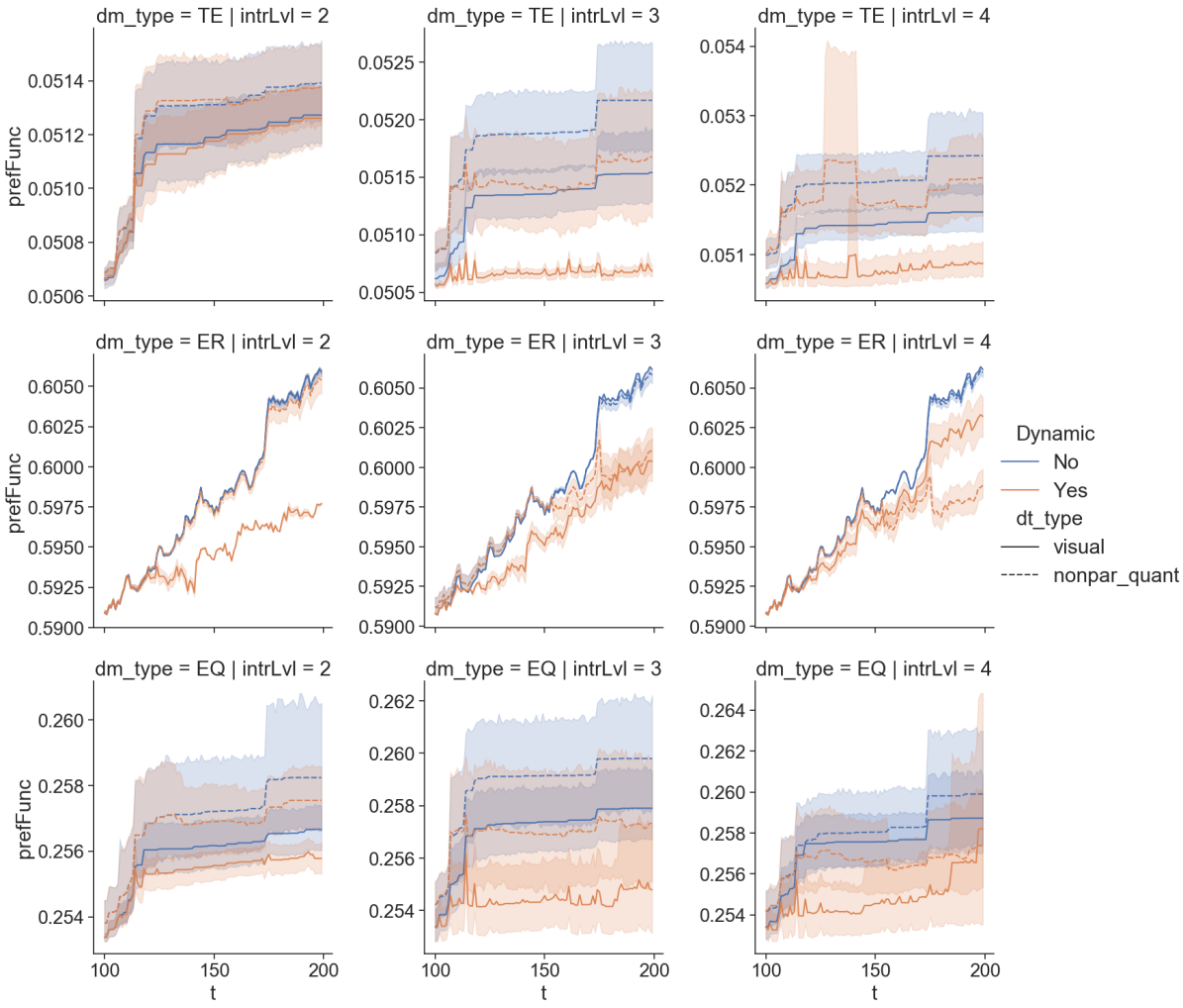
(a) indtrack1


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
Dynamic
No
Yes
dt_type
visual
nonpar_quant

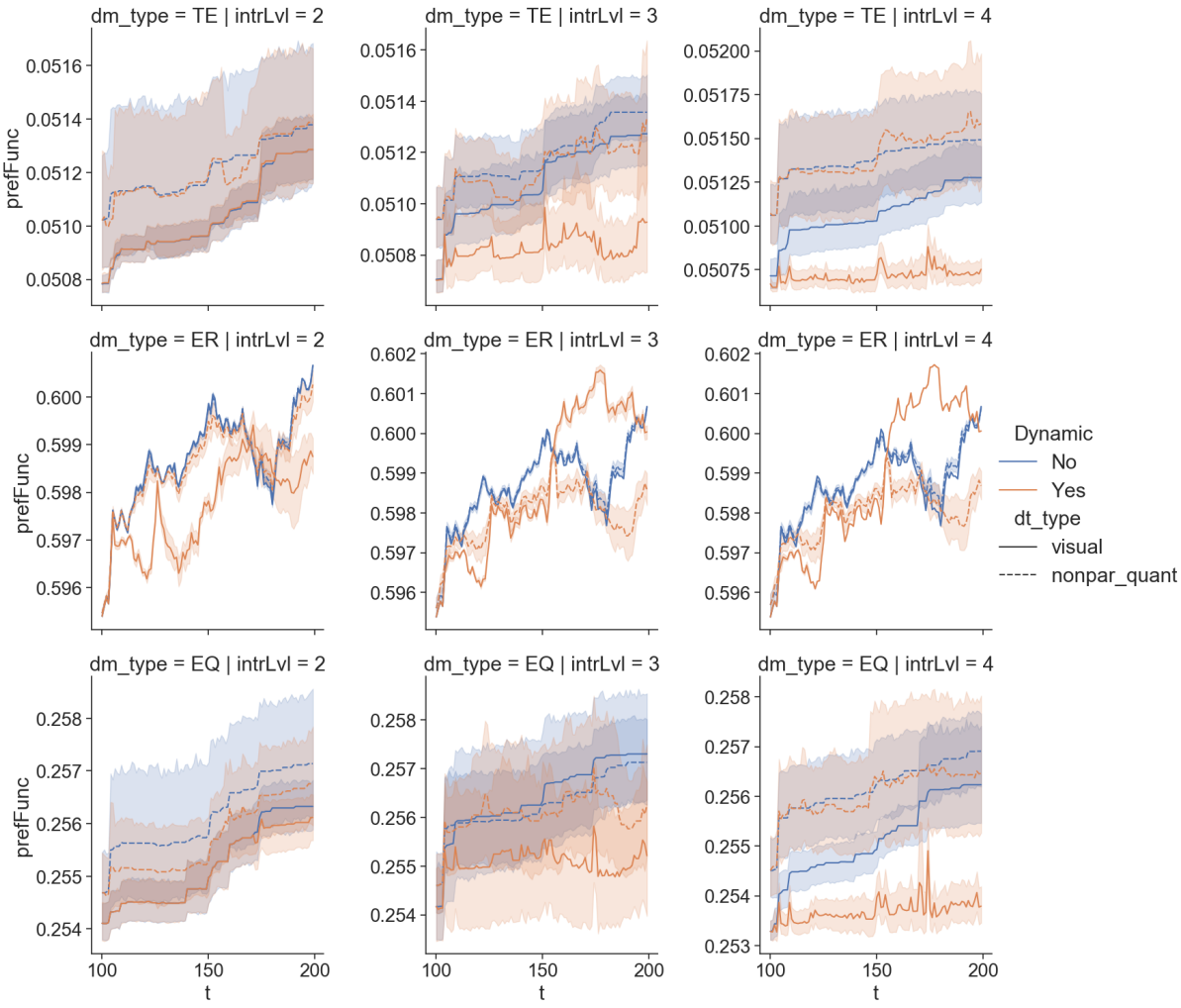
(b) indtrack2


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
Dynamic
No
Yes
dt_type
visual
nonpar_quant

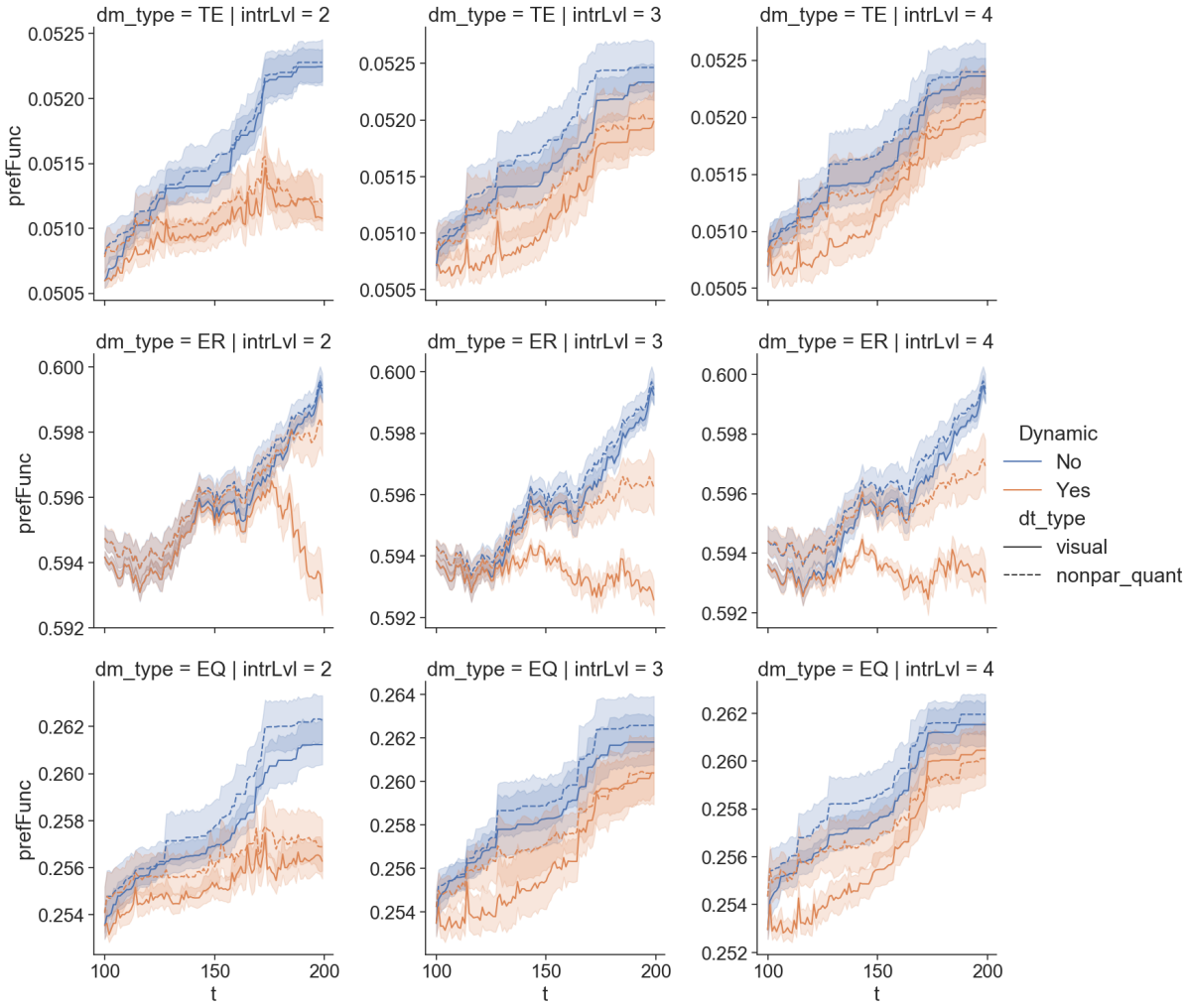
(c) indtrack3


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
Dynamic
No
Yes
dt_type
visual
nonpar_quant

(d) indtrack4


dm_type = TE | intrLvl = 2
dm_type = TE | intrLvl = 3
dm_type = TE | intrLvl = 4
dm_type = ER | intrLvl = 2
dm_type = ER | intrLvl = 3
dm_type = ER | intrLvl = 4
dm_type = EQ | intrLvl = 2
dm_type = EQ | intrLvl = 3
dm_type = EQ | intrLvl = 4
prefFunc
t
Dynamic
No
Yes
dt_type
visual
nonpar_quant

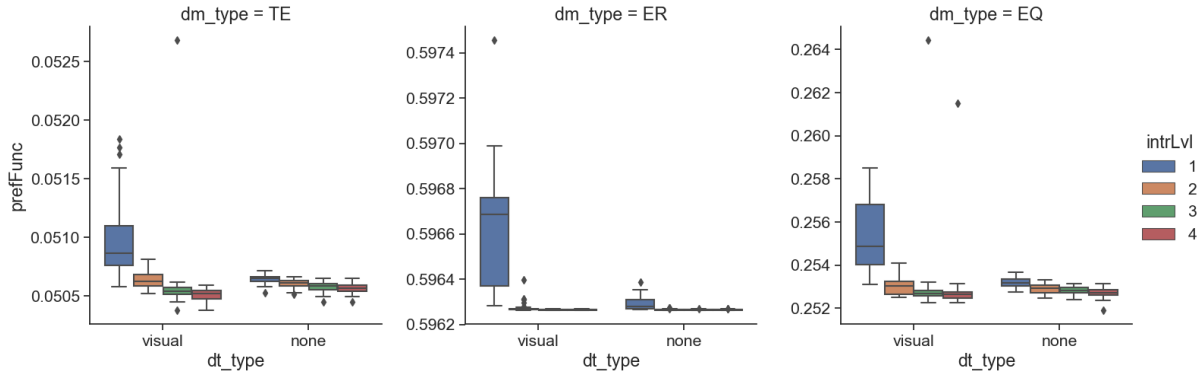
(e) indtrack5



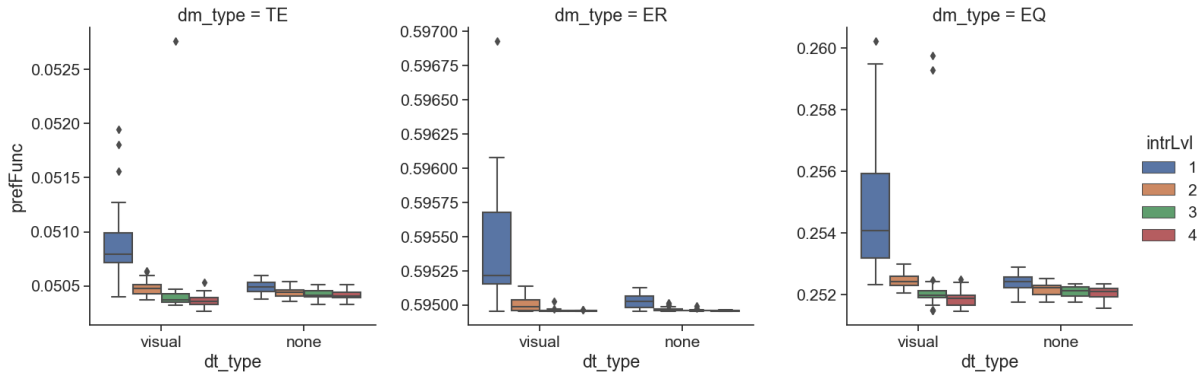
(f) indtrack6

Figure 49 – Comparison of the preference function of three simulated investors when using the SBH and MBH strategies for the out-of-sample period of the six instances of the problem

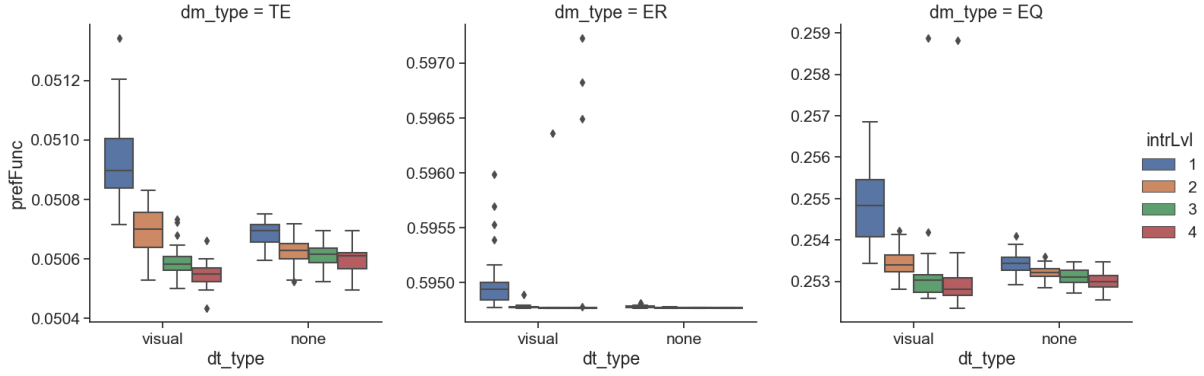
B.1.3 Comparison against NSGA-II with and without preference guidance



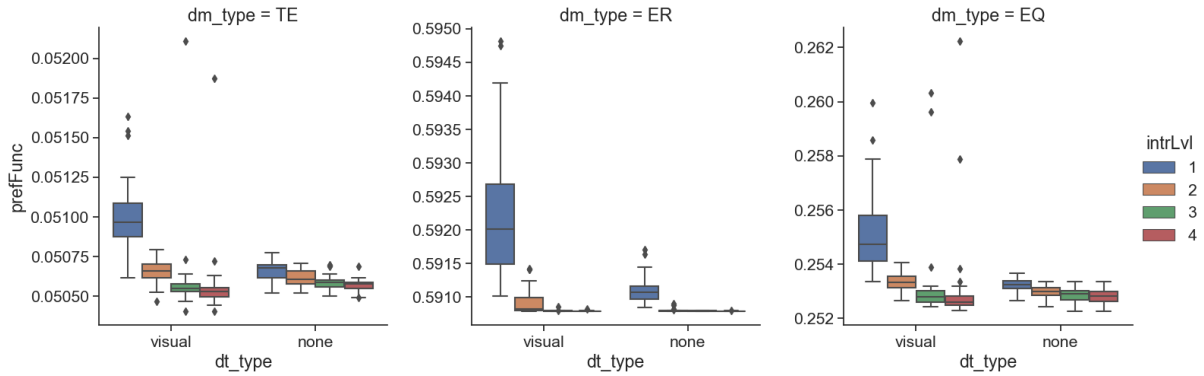
(a) indtrack1



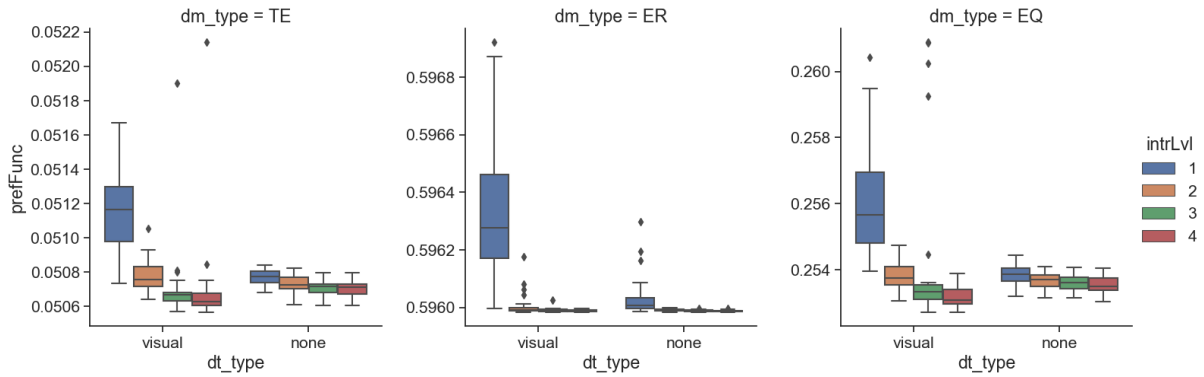
(b) indtrack2



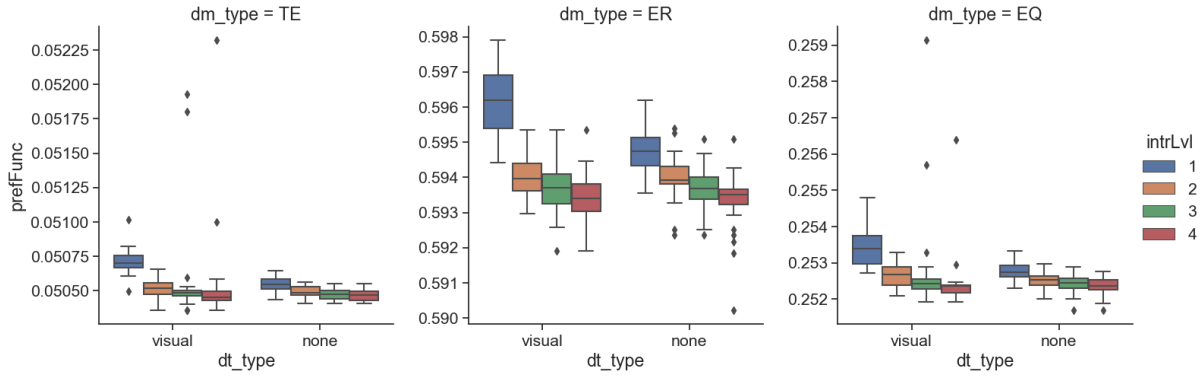
(c) indtrack3



(d) indtrack4



(e) indtrack5



(f) indtrack6

Figure 50 – Comparison of the preference function of three simulated investors with and without the preference information extracted in simulated interactions for the six instances of the problem

B.2 STATISTICAL TESTS FOR THE IN-SAMPLE RESULTS

B.2.1 Comparing the preference satisfaction for different frontier filters

The Wilcoxon signed-rank test tests the null hypothesis that the median difference between two related samples is zero. If the p-value satisfies $p < 0.05$, then the null hypothesis is rejected at a confidence level of 5%, concluding that there is a difference in the preference satisfaction between the groups. We use the Wilcoxon signed-rank test to evaluate if there is a significant difference between the performance of portfolios produced using different frontier filters in terms of preference satisfaction.

The frontier filters were compared considering different instances of the problem, investor types, and data table presentation methods. The p-values are presented for the 'visual', 'par_quant' and 'nonpar_quant' data table presentation methods in Tables 16, 17 and 18, respectively. The p-values in bold are those that satisfy $p < 0.05$.

Table 16 shows that the differences in the performance of frontier filters are significant for all variables for the 'visual' data table presentation method. Thus it can be observed from Figure 46 that the 'closer' frontier filter produces better portfolios, in terms of preference satisfaction, when combined with the 'visual' data table presentation method for all instances, investor types and interaction levels.

Table 17 shows that for the ER investor, the differences in the performance of frontier filters are significant for all variables for the 'par_quant' data table presentation method. Thus it can be observed from Figure 46 that the 'closer' frontier filter produces better portfolios, in terms of preference satisfaction, when combined with the 'par_quant' data table presentation method, for the ER investor, in all instances and interaction levels.

The difference in performance of the frontier filters for the TE investor are significant for intrLvl=2 in indtrack1-5, for intrLvl=3 in indtrack1 and indtrack5-6, and for intrLvl=4 in indtrack1. Thus, observing Figure 46, the performance of the 'closer' filter combined with the 'par_quant' data table presentation method for the TE investor is better for intrLvl=2 in all instances, but is worse for intrLvl>2 in indtrack1, indtrack5 and indtrack6.

The difference in performance of the frontier filters for the EQ investor are significant for intrLvl=2 in indtrack1-5, for intrLvl=3 in indtrack1-3, and for intrLvl4 in indtrack1-2. Thus, observing Figure 46, the performance of the 'closer' filter combined with the 'par_quant' data table presentation method for the EQ investor is better for intrLvl=2 in indtrack1-5 and for intrLvl>2 in indtrack1-2, but is worse for intrLvl>2 in indtrack3. Thus, in the majority of significant comparisons, the 'closer' filter performed better than the 'farther' filter in the 'par_quant' data table presentation method.

Table 18 shows that for the ER investor, the differences in the performance of frontier filters are significant for all variables for the 'nonpar_quant' data table presentation method, except for intrLvl> 2 in indtrack2. Thus it can be observed from Figure 46 that the 'closer' frontier filter produces better portfolios, in terms of preference satisfaction,

when combined with the 'nonpar_quant' data table presentation method, for the ER investor, in all instances for intrLvl=2, and in indtrack1 and in indtrack3-6 for intrLvl>2.

The difference in performance of the frontier filters for the TE investor are significant for intrLvl=2 in all instances, for intrLvl=3 in indtrack1-5, and for intrLvl=4 in indtrack1-2 and in indtrack4. Thus, observing Figure 46, the performance of the 'closer' filter combined with the 'nonpar_quant' data table presentation method for the TE investor is better for intrLvl=2 in all instances, for intrLvl=3 in indtrack1-5, and for intrLvl=4 in indtrack1-2 and indtrack4.

The difference in performance of the frontier filters for the EQ investor are significant for intrLvl=2 in all instances, for intrLvl=3 in indtrack1-5, and for intrLvl=4 in indtrack1-5. Thus, observing Figure 46, the performance of the 'closer' filter combined with the 'nonpar_quant' data table presentation method for the EQ investor is better for intrLvl=2 in all instances, for intrLvl=3 in indtrack1-5, and for intrLvl=4 in indtrack1-5. Thus, in the majority of significant comparisons, the 'closer' filter performed better than the 'farther' filter in the 'nonpar_quant' data table presentation method.

dm_type	intrLvl	indtrack					
		1	2	3	4	5	6
TE	2	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	3	**0.00000**	**0.00002**	**0.00000**	**0.00001**	**0.00001**	**0.00036**
	4	**0.00000**	**0.00000**	**0.00000**	**0.00001**	**0.00001**	**0.00004**
ER	2	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	3	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	4	**0.00000**	**0.00000**	**0.00036**	**0.00000**	**0.00000**	**0.00000**
EQ	2	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00001**
	3	**0.00000**	**0.00000**	**0.00000**	**0.00001**	**0.00014**	**0.00006**
	4	**0.00000**	**0.00000**	**0.00001**	**0.00002**	**0.00000**	**0.00003**

Table 16 – Resultant p-values from the comparison between frontier filters for the 'visual' data table presentation method.

dm_type	intrLvl	indtrack					
		1	2	3	4	5	6
TE	2	**0.00258**	**0.00000**	**0.00031**	**0.00004**	**0.00015**	0.53044
	3	**0.03327**	0.99179	0.78126	0.59994	**0.02067**	**0.00083**
	4	**0.00241**	0.30861	0.42843	0.50383	0.76552	0.22888
ER	2	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	3	**0.00000**	**0.00000**	**0.00019**	**0.00000**	**0.00000**	**0.00000**
	4	**0.00000**	**0.00000**	**0.00003**	**0.00000**	**0.00003**	**0.00000**
EQ	2	**0.00002**	**0.00002**	**0.03683**	**0.00002**	**0.01108**	0.15286
	3	**0.00385**	**0.00019**	**0.01108**	0.27116	0.39333	0.59994
	4	**0.00077**	**0.01175**	0.92626	0.32857	0.67328	0.41653

Table 17 – Resultant p-values from the comparison between frontier filters for the 'par_quant' data table presentation method.

dm_type	intrLvl	indtrack					
		1	2	3	4	5	6
TE	2	**0.00000**	**0.00001**	**0.00001**	**0.00000**	**0.00148**	**0.00049**
	3	**0.00008**	**0.00001**	**0.00499**	**0.00171**	**0.00033**	0.22888
	4	**0.01480**	**0.00072**	0.37094	**0.02431**	0.16503	0.81302
ER	2	**0.00000**	**0.00002**	**0.00000**	**0.00000**	**0.00000**	**0.00001**
	3	**0.00000**	0.08221	**0.00111**	**0.00011**	**0.00036**	**0.00000**
	4	**0.00296**	0.13059	**0.00039**	**0.00000**	**0.00361**	**0.00003**
EQ	2	**0.00000**	**0.00000**	**0.00111**	**0.00003**	**0.00004**	**0.00642**
	3	**0.00003**	**0.00111**	**0.00296**	**0.00211**	**0.00053**	0.58571
	4	**0.00001**	**0.00003**	**0.00241**	**0.00296**	**0.00005**	0.64352

Table 18 – Resultant p-values from the comparison between frontier filters for the 'non-par_quant' data table presentation method.

B.2.2 Comparing the preference satisfaction for different interaction levels

We use the Wilcoxon signed-rank test to evaluate if there is a significant difference between the performance of portfolios produced in sequential interaction levels in terms of preference satisfaction. The interaction levels were compared considering the 'closer' frontier filter, which performed better than the 'farther' frontier filter, different instances of the problem, investor types, and data table presentation methods. The p-values are presented for the 'visual', 'par_quant' and 'nonpar_quant' data table presentation methods in Tables 19, 20 and 21, respectively. The p-values in bold are those that satisfy $p < 0.05$.

Table 19 shows that the differences in the performance for all pairs of interaction levels are significant for the TE investor in indtrack1-4 and indtrack6, and for intrLvl<4 in indtrack5, when the 'visual' data table presentation method is adopted. Thus it can be observed from Figure 46 that the 'visual' data table presentation method produces better portfolios, in terms of preference satisfaction, as the number of interactions increases from 1 to 4 in indtrack1-4, and as the number of interactions increases from 1 to 3 in indtrack5 and indtrack6. The results for the EQ investor were very similar.

For the ER investor, the differences in the performance for all pairs of interaction levels are significant in indtrack1-2 and indtrack4-6, and for intrLvl<4 in indtrack3. But, observing Figure 46, better portfolios are produced as interaction levels increase from 1 to 2 in indtrack1-5 and from 1 to 4 in indtrack6. It was observed that when intrLvl> 2 the solver arrives in local optima almost always in indtrack 1-5, although there are some outliers. Thus, for indtrack1-5, the increments were observed only between interaction levels 1 and 2.

Table 20 shows that the differences in the performance for the first pairs of interaction levels are significant for the TE investor in indtrack1-3 when the 'par_quant' data table presentation method is adopted. Thus it can be observed from Figure 46 that the 'par_quant' data table presentation method produces better portfolios, in terms of preference satisfaction, as the number of interaction levels increases from 1 to 2 in indtrack1-3. For the EQ investor, the 'par_quant' data table presentation method produces better portfolios, with a significant difference, as the number of interactions increases from 1 to 2 in indtrack1 and in indtrack4, and as the number of interactions increases from 1 to 3 in indtrack2.

For the ER investor, the differences in the performance for all pairs of interaction levels are significant in indtrack1-4 and indtrack6, and for intrLvl<4 in indtrack5. Observing Figure 46, better portfolios are produced as interaction levels increase from 1 to 2 in indtrack1-5 and as interaction levels increase from 1 to 4 indtrack6. Similar to the 'visual' approach, it was observed that when intrLvl> 2 the solver arrives in local optima almost always in indtrack1-5, although there are some outliers. Thus, for indtrack1-5, the increments were observed only between interaction levels 1 and 2.

Table 21 shows that the differences in the performance for the first pairs of interaction

levels are significant for all types of investor in indtrack1-6 when the 'nonpar_quant' data table presentation method is adopted. Thus it can be observed from Figure 46 that the 'nonpar_quant' data table presentation method produces better portfolios for the TE and EQ investors, in terms of preference satisfaction, as the number of interaction levels increases from 1 to 2 in all instances.

For the ER investor, better portfolios are produced as interaction levels increase from 1 to 2 in all instances. Also, the 'nonpar_quant' method produces better portfolios for the ER investor, with a significant difference, as the number of interactions increases from 1 to 3 in indtrack6. Similar to the results for the simulated ER investor in the 'visual' and 'par_quant' approaches, it was observed that the solver arrives in local optima in indtrack1 and indtrack4 when intrLvl> 2 in the 'nonpar_quant' approach. Thus, for indtrack1-5, the increments were observed only between interaction levels 1 and 2.

dm_type	(intrLvl-1, intrLvl)	indtrack					
		1	2	3	4	5	6
TE	(1, 2)	**0.00001**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	(2, 3)	**0.00007**	**0.00005**	**0.00000**	**0.00003**	**0.00026**	**0.02459**
	(3, 4)	**0.00102**	**0.00010**	**0.00014**	**0.00091**	0.10804	**0.04951**
ER	(1, 2)	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	(2, 3)	**0.00013**	**0.00003**	**0.00007**	**0.00002**	**0.00009**	**0.00044**
	(3, 4)	**0.04311**	**0.00147**	0.07314	**0.00769**	**0.02771**	**0.00098**
EQ	(1, 2)	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	(2, 3)	**0.00015**	**0.00045**	**0.00019**	**0.00036**	0.05446	**0.00923**
	(3, 4)	**0.02786**	**0.00002**	**0.00322**	**0.00564**	**0.00325**	**0.00061**

Table 19 – Resultant p-values from the comparison between interaction levels for the 'visual' data table presentation method.

dm_type	(intrLvl-1, intrLvl)	indtrack					
		1	2	3	4	5	6
TE	(1, 2)	**0.02067**	**0.00004**	**0.02703**	0.06871	0.08972	0.28021
	(2, 3)	0.88235	**0.00123**	**0.01108**	0.06557	0.11535	**0.00313**
	(3, 4)	0.37577	0.08962	0.17655	0.30437	**0.02459**	0.07649
ER	(1, 2)	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
	(2, 3)	**0.00013**	**0.00020**	**0.00468**	**0.00006**	**0.00006**	**0.00098**
	(3, 4)	**0.00222**	**0.00044**	**0.00898**	**0.00769**	0.05046	**0.00065**
EQ	(1, 2)	**0.00045**	**0.00385**	0.25628	**0.00024**	0.22888	**0.00773**
	(2, 3)	0.39443	**0.01412**	**0.00128**	0.13113	0.13535	0.08648
	(3, 4)	0.36127	0.29430	**0.03186**	0.93169	0.92344	0.80776

Table 20 – Resultant p-values from the comparison between interaction levels for the 'par_quant' data table presentation method.

dm_type	(intrLvl-1, intrLvl)	indtrack					
		1	2	3	4	5	6
TE	(1, 2)	**0.00001**	**0.00013**	**0.00001**	**0.00000**	**0.00241**	**0.00071**
	(2, 3)	0.68916	0.11094	0.39486	0.49452	0.36004	0.86116
	(3, 4)	0.95443	0.70513	0.53220	0.33948	0.40042	0.66824
ER	(1, 2)	**0.00053**	**0.00114**	**0.00003**	**0.00001**	**0.00001**	**0.00001**
	(2, 3)	**0.00498**	0.70888	0.62660	**0.00325**	0.52668	**0.00236**
	(3, 4)	0.95935	0.98927	0.20122	**0.00573**	0.77643	0.27724
EQ	(1, 2)	**0.00002**	**0.00038**	**0.00316**	**0.00006**	**0.00096**	**0.00120**
	(2, 3)	0.47505	0.81987	0.06194	0.17655	0.75479	0.63251
	(3, 4)	0.30366	**0.00947**	0.73267	0.48597	0.44550	0.49263

Table 21 – Resultant p-values from the comparison between interaction levels for the 'nonpar_quant' data table presentation method.

B.2.3 Comparing NSGA-II and IMO-DRSA

We use the Wilcoxon signed-rank test to evaluate if there is a significant difference between the performance of portfolios produced by NSGA-II and IMO-DRSA from interaction levels 2 to 4 in terms of preference satisfaction. The IMO-DRSA uses the closer frontier filter and the visual data table presentation method, a combination that obtained a good overall performance. The p-values are presented in Table 22. The p-values in bold are those that satisfy $p < 0.05$.

For simulated TE investors, there is a significant performance difference between NSGA-II and IMO-DRSA, in terms of preference satisfaction, for intrLvl=2 in intrack2-4. Observing Figure 6, it is possible to observe that NSGA-II performed better in these cases. For intrLvl=3, there is a significant performance difference between the algorithms in indtrack1-2 and intrack4-5, where IMO-DRSA performed better in these cases. For intrLVL=4, there is a significant performance difference between the algorithms in indtrack1-5, where IMO-DRSA performed better in these cases.

For simulated ER investors, there is a significant difference between NSGA-II and IMO-DRSA in terms of preference satisfaction for intrLvl=2 in intrack1-5. Observing Figure 6, it is possible to observe that despite some outliers, both algorithms reached local optima in indtrack1 and indtrack3. NSGA-II achieved better performance in indtrack2, indtrack4, and indtrack5. For intrLvl=3, there is no evidence supporting a significant performance difference between the algorithms in any instance. Although there is a significant performance difference between the algorithms for intrLvl=4 in indtrack2, both algorithms arrived at local optima.

For simulated EQ investors, there is a significant difference between NSGA-II and IMO-DRSA in terms of preference satisfaction for intrLvl=2 in intrack2-4. Observing Figure 6, it is possible to observe that NSGA-II performed better in these cases. For intrLvl=3, there is no evidence supporting a significant performance difference between the algorithms in any instance. For intrLvl=4, there is a significant performance difference between the algorithms in indtrack1-2 and intrack5, where IMO-DRSA performed better in these cases.

dm_type	(intrLvl)	indtrack					
		1	2	3	4	5	6
TE	2	0.07865	**0.02564**	**0.00096**	**0.02849**	0.05710	0.18462
	3	**0.00984**	**0.01566**	0.10201	**0.01044**	**0.00727**	0.46528
	4	**0.00083**	**0.00021**	**0.00049**	**0.01657**	**0.00567**	0.81302
ER	2	**0.00411**	**0.00296**	**0.02703**	**0.00022**	**0.03327**	0.84508
	3	0.67328	0.89364	0.42843	0.76552	0.17791	0.68836
	4	0.53044	**0.03501**	0.70356	0.36004	0.23694	0.79710
EQ	2	0.08221	**0.00002**	**0.00258**	**0.00026**	0.17138	0.11093
	3	0.15286	0.42843	0.26230	0.94261	0.16503	0.84508
	4	**0.04716**	**0.00277**	0.13059	0.47795	**0.00066**	0.37094

Table 22 – Resultant p-values from the comparison between NSGA-II and IMO-DRSA with the closer frontier filter and visual approach in each interaction level.

B.2.4 Evaluating the IMO-DRSA approach further with the 'visual' data table presentation method

A more profound evaluation concerning IMO-DRSA with the 'closer' frontier filter combined with the 'visual' data table approach since it performed better among all IMO-DRSA approaches. The Levene test was applied to evaluate if there is a significant difference between the variance of the preference satisfaction performance of portfolios produced by the 'closer' and 'farther' frontier filters when using the 'visual' approach, which is represented here as $f_{\theta^{DM},dt_{type}}$, where $\theta^{DM} = \{\theta^{TE}, \theta^{ER}, \theta^{EQ}\}$ and $dt_{type} = \{'closer', 'farther'\}$. The variance differences $\sigma^2(f_{\theta^{DM},closer}) - \sigma^2(f_{\theta^{DM},farther})$ and p-values in parenthesis are presented in Table 23. The p-values in bold are those that satisfy $p < 0.05$. It is possible to see that, in the majority of significant cases, the 'closer' filter produces portfolios with a lower variance of preference satisfaction.

dm_type	intrLvl	indtrack					
		1	2	3	4	5	6
TE	2	-4.26e-07 (**0.00000**)	-1.30e-07 (**0.00000**)	-8.71e-08 (**0.00000**)	-1.76e-07 (**0.00000**)	-9.62e-08 (**0.00000**)	-1.97e-08 (**0.00256**)
	3	-1.85e-07 (**0.00014**)	5.02e-08 (**0.02374**)	-7.04e-08 (**0.00000**)	-4.61e-08 (**0.00973**)	-7.54e-08 (**0.00021**)	1.11e-07 (0.58943)
	4	-3.11e-07 (**0.00000**)	-1.14e-07 (**0.00000**)	-7.45e-08 (**0.00000**)	-2.61e-08 (**0.02355**)	-6.30e-08 (**0.00075**)	1.11e-07 (0.60543)
ER	2	-1.94e-07 (**0.00000**)	-4.56e-07 (**0.00000**)	-2.66e-07 (**0.00000**)	-1.54e-06 (**0.00000**)	-9.42e-08 (**0.00000**)	-1.71e-07 (0.22891)
	3	-1.66e-07 (**0.00000**)	-4.16e-07 (**0.00000**)	-1.38e-07 (**0.00000**)	-1.50e-06 (**0.00000**)	-7.74e-08 (**0.00000**)	-2.09e-08 (0.79926)
	4	-1.53e-07 (**0.00000**)	-3.92e-07 (**0.00000**)	1.97e-07 (0.14832)	-1.33e-06 (**0.00000**)	-7.55e-08 (**0.00000**)	-3.08e-08 (0.65081)
EQ	2	-5.34e-06 (**0.00000**)	-4.19e-06 (**0.00000**)	-3.62e-06 (**0.00001**)	-3.15e-06 (**0.00000**)	-3.55e-06 (**0.00000**)	-5.06e-07 (**0.00048**)
	3	-3.94e-07 (**0.01109**)	3.28e-07 (**0.03243**)	-1.39e-06 (**0.00437**)	1.12e-06 (0.20388)	2.47e-06 (0.48275)	1.39e-06 (0.95700)
	4	-2.14e-06 (**0.00422**)	-3.92e-06 (**0.00000**)	-2.01e-06 (**0.00176**)	1.75e-06 (0.29018)	-2.51e-06 (**0.00000**)	1.93e-07 (0.09584)

Table 23 – Resultant p-values of the Levene test from the comparison between frontier filters for the 'visual' data table presentation method.

The Levene test was also applied to evaluate if there is a significant difference between the variance of the preference satisfaction performance of portfolios produced by sequential interaction levels using the 'closer' filter combined with the 'visual' approach, which is represented here as $f_{\theta^{DM},intrLvl}$, where $\theta^{DM} = \{\theta^{TE}, \theta^{ER}, \theta^{EQ}\}$ and $intrLvl = \{2, 3, 4\}$. The variance differences $\sigma^2(f_{\theta^{DM},intrLvl}) - \sigma^2(f_{\theta^{DM},intrLvl-1})$ and p-values in parenthesis are presented in Table 24. The p-values in bold are those that satisfy $p < 0.05$. It is possible to observe that most of the significant cases are associated with interaction level 2. Thus, the produced portfolios have a significantly lower variance of preference satisfaction mostly when the interaction process shifts from level 1 to level 2 for all investors.

dm_type	(intrLvl-1, intrLvl)	indtrack					
		1	2	3	4	5	6
TE	(1, 2)	-1.18e-07 (**0.00013**)	-1.25e-07 (**0.00033**)	-1.36e-08 (**0.03700**)	-5.10e-08 (**0.00145**)	-4.89e-08 (**0.00000**)	-3.01e-09 (0.68374)
	(2, 3)	1.46e-07 (0.48422)	1.79e-07 (0.40778)	-3.10e-09 (**0.00921**)	7.59e-08 (0.54099)	4.39e-08 (0.80303)	1.18e-07 (0.27256)
	(3, 4)	-1.48e-07 (0.35068)	-1.81e-07 (0.31928)	-1.16e-09 (0.36535)	-1.97e-08 (0.93179)	2.33e-08 (0.87621)	-6.41e-10 (0.93888)
ER	(1, 2)	-6.66e-08 (**0.00000**)	-1.76e-07 (**0.00074**)	-7.79e-08 (**0.00081**)	-9.56e-07 (**0.00005**)	-6.36e-08 (**0.00000**)	-5.82e-07 (**0.00349**)
	(2, 3)	-6.70e-10 (0.05283)	-2.90e-09 (**0.00001**)	8.12e-08 (0.38134)	-3.91e-08 (**0.00061**)	-1.45e-09 (0.07328)	1.27e-07 (0.49323)
	(3, 4)	-2.87e-13 (0.81107)	-1.50e-10 (0.13322)	3.15e-07 (0.23417)	-1.04e-10 (0.46345)	-4.23e-11 (0.33135)	-1.35e-08 (0.81722)
EQ	(1, 2)	-2.72e-06 (**0.00001**)	-4.18e-06 (**0.00001**)	-8.42e-07 (**0.00003**)	-2.56e-06 (**0.00015**)	-2.21e-06 (**0.00006**)	-1.73e-07 (**0.01353**)
	(2, 3)	4.32e-06 (0.56945)	3.57e-06 (0.15940)	1.12e-06 (0.31001)	3.15e-06 (0.21945)	5.68e-06 (0.07392)	1.72e-06 (0.29875)
	(3, 4)	-1.90e-06 (0.86770)	-3.56e-06 (0.18026)	3.50e-08 (0.97952)	5.22e-07 (0.95151)	-5.78e-06 (**0.04034**)	-1.24e-06 (0.38664)

Table 24 – Resultant variance differences and p-values from the comparison between interaction levels for the 'visual' data table generation method.

Finally, the Levene test was applied to evaluate if there is a significant difference between the variance of the preference satisfaction performance of portfolios produced by the and NSGA-II and 'IMO-DRSA' using the 'closer' filter combined with the 'visual' approach, which is represented here as $f_{\theta^{DM},GA}$, where $\theta^{DM} = \{\theta^{TE}, \theta^{ER}, \theta^{EQ}\}$ and $GA = \{IMO - DRSA, NSGA - II\}$. The variance differences $\sigma^2(f_{\theta^{DM},IMO-DRSA}) - \sigma^2(f_{\theta^{DM},NSGA-II})$ and p-values in parenthesis are presented in Table 25. The p-values in bold are those that satisfy $p < 0.05$. It is possible to see that the NSGA-II approach produces portfolios with a significantly lower variance of preference satisfaction mostly at interaction level 2 for all investors.

dm_type	intrLvl	indtrack					
		1	2	3	4	5	6
TE	2	3.52e-09 (**0.00181**)	2.45e-09 (0.14866)	3.97e-09 (**0.00085**)	3.63e-09 (**0.04582**)	5.97e-09 (**0.01451**)	3.03e-09 (**0.02721**)
	3	1.49e-07 (0.31236)	1.82e-07 (0.31085)	9.70e-10 (0.44003)	7.97e-08 (0.25333)	4.99e-08 (0.30438)	1.21e-07 (0.16031)
	4	6.35e-10 (0.47151)	9.79e-10 (0.52136)	-2.74e-10 (0.61072)	6.04e-08 (0.19621)	7.30e-08 (0.32593)	1.20e-07 (0.20054)
ER	2	6.65e-10 (0.06544)	2.78e-09 (**0.00013**)	4.78e-10 (0.13038)	3.86e-08 (**0.00097**)	1.50e-09 (**0.04605**)	-7.96e-08 (0.93598)
	3	5.12e-14 (0.67722)	9.55e-11 (0.70773)	8.17e-08 (0.32291)	1.39e-10 (0.22847)	4.85e-11 (0.12057)	1.29e-07 (0.36910)
	4	3.90e-13 (0.53089)	-5.18e-12 (0.10718)	3.97e-07 (0.08094)	4.01e-11 (0.28741)	7.45e-12 (0.12694)	-3.15e-07 (0.72039)
EQ	2	1.25e-07 (**0.01223**)	1.56e-08 (0.60911)	8.37e-08 (**0.00715**)	6.76e-08 (**0.04655**)	1.43e-07 (**0.00563**)	4.00e-08 (0.06728)
	3	4.46e-06 (0.32308)	3.59e-06 (0.13911)	1.19e-06 (0.10929)	3.21e-06 (0.13912)	5.82e-06 (**0.03074**)	1.76e-06 (0.19069)
	4	2.54e-06 (0.30975)	3.19e-08 (0.15637)	1.22e-06 (0.15476)	3.72e-06 (0.16825)	3.34e-08 (0.52756)	5.16e-07 (0.56497)

Table 25 – Resultant variance differences and p-values of the Levene test from the comparison between NSGA-II and IMO-DRSA combining the 'visual' data table generation method and the 'closer' frontier filter.