



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

# **ANÁLISE DE INTEGRAÇÃO DA CLASSIFICAÇÃO DE TENDÊNCIAS FINANCEIRAS E OTIMIZAÇÃO DE PORTFÓLIOS**

**FERNANDO GARCIA DINIZ CAMPOS FERREIRA**

Orientador: Rodrigo Tomás Nogueira Cardoso  
Centro Federal de Educação Tecnológica de Minas Gerais

BELO HORIZONTE  
OUTUBRO DE 2022

**FERNANDO GARCIA DINIZ CAMPOS FERREIRA**

**ANÁLISE DE INTEGRAÇÃO DA CLASSIFICAÇÃO DE  
TENDÊNCIAS FINANCEIRAS E OTIMIZAÇÃO DE  
PORTFÓLIOS**

Tese apresentado ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Doutor em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Rodrigo Tomás Nogueira Cardoso  
Centro Federal de Educação Tecnológica de Minas Gerais

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL  
BELO HORIZONTE  
OUTUBRO DE 2022

F383a Ferreira, Fernando Garcia Diniz Campos  
Análise de integração da classificação de tendências financeiras e otimização de portfólios / Fernando Garcia Diniz Campos Ferreira. – 2022.  
151 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.

Orientador: Rodrigo Tomás Nogueira Cardoso

Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Finanças – Computação – Teses. 2. Títulos (Finanças) – Métodos de simulação – Teses. 3. Otimização matemática – Teses. 4. Análise de séries temporais – Teses. 5. Algoritmos – Teses. 6. Aprendizagem do computador – Teses. I. Cardoso, Rodrigo Tomás Nogueira. II. Centro Federal de Educação Tecnológica de Minas Gerais. III. Título.

CDD 003.3



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

## **“ANÁLISE DE INTEGRAÇÃO DA CLASSIFICAÇÃO DE TENDÊNCIAS FINANCEIRAS E OTIMIZAÇÃO DE PORTFÓLIOS”.**

Tese de Doutorado apresentada por **Fernando Garcia Diniz Campos Ferreira**, em 24 de novembro de 2022, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

**Prof. Dr. Rodrigo Tomás Nogueira Cardoso (Orientador)**  
Centro Federal de Educação Tecnológica de Minas Gerais

**Prof. Dr. Frederico Gadelha Guimarães**  
Universidade Federal de Minas Gerais

**Prof. Dr. Felipe Dias Paiva**  
Centro Federal de Educação Tecnológica de Minas Gerais

**Prof. Dr. Arthur Rodrigo Bosco de Magalhães**  
Centro Federal de Educação Tecnológica de Minas Gerais

**Prof. Dr. Adriano César Machado Pereira**  
Universidade Federal de Minas Gerais

Visto e permitida a impressão,

**Profª. Drª. Elizabeth Fialho Wanner**  
Presidenta do Colegiado do Programa de Pós-Graduação em  
Modelagem Matemática e Computacional

# Agradecimentos

Gostaria de agradecer aos docentes do curso de pós-graduação em Modelagem Matemática e Computacional do CEFET-MG, especialmente ao professor Dr. Rodrigo Tomás Nogueira Cardoso, que me orientou neste trabalho. Agradeço, por fim, à agência de fomento CAPES pela bolsa concedida durante o período da pesquisa.

# Resumo

A área de Finanças Computacionais é marcada por profundas mudanças nas últimas décadas provenientes do surgimento de um rápido desenvolvimento tecnológico que permitiu a incorporação de técnicas computacionais complexas para modelagem e predição dos movimentos dos preços no Mercado Financeiro. Neste cenário, este trabalho propõe um arcabouço computacional composto por diferentes modelos e técnicas de otimização de portfólios financeiros e de classificação de tendências de ativos financeiros, além de proporcionar diferentes formas de integração das duas abordagens. Assim, o arcabouço é composto por quatro modelos propostos para otimização de portfólio e um método para a solução de cada um desses modelos, além de doze algoritmos de classificação utilizando conjuntos de atributos também propostos no trabalho e, por fim, três formas de integração dos métodos de otimização e classificação são considerados. Para os métodos de otimização de portfólio, métodos exatos apresentam melhores soluções, apesar de apresentarem tempos de execução muito maiores que meta-heurísticas. Para a classificação de tendências, os resultados indicam que o Convolutional Neural Network (CNN) foi o algoritmo que apresenta o melhor desempenho, no geral, considerando os classificadores utilizados. Simulações de investimentos mostram que negociações a uma frequência maior geram custos de transação maiores, podendo inviabilizar a utilização da metodologia proposta. Combinações de otimização de portfólios e classificações financeiras podem aproveitar as vantagens de ambas as abordagens, como mostrado pelos resultados que apresentam maiores retornos acumulados para métodos combinados. Finalmente, pode-se observar que os melhores métodos de otimização e classificação dependem muito da estratégia de operação adotada e das preferências do investidor, justificando o desenvolvimento deste arcabouço computacional proposto, composto por vários métodos.

**Palavras-chave:** Finanças computacionais. Otimização de portfólio. Classificação de séries temporais. Algoritmo evolutivo. Aprendizagem de Máquina.

# Abstract

The area of Computational Finance has been marked by profound changes in recent decades as a result of a rapid technological development that has allowed the incorporation of complex computational techniques for modeling and predicting price movements in the Financial Market. In this scenario, this work proposes a computational framework composed of different models and techniques of financial portfolio optimization and trend classification of financial assets, in addition to providing different ways of combining the two approaches. Thus, the framework consists of four models proposed for portfolio optimization, with one method for solving each of these models, as well as nine classification algorithms using sets of attributes also proposed in the work and, finally, three ways of combining portfolio optimization and classification methods are considered. Regarding portfolio optimization methods, exact methods presents better solutions, although they have much longer execution times than meta-heuristics. For financial trends classification, results indicate that the Convolutional Neural Network (CNN) was the algorithm that presents the best performance, in general, considering the classifiers used. Investment simulations show that negotiations at a higher frequency generate higher transaction costs, which may make the proposed methodology unfeasible. Combinations of portfolio optimization and financial classification can take advantage of both approaches, as shown by results, since they reveal higher cumulative returns for combined methods. In conclusion, it can be observed that the best optimization and classification methods depend very much on the adopted trading strategy and investor preferences, which justifies the development of this proposed computational framework composed by several methods.

**Keywords:** Computational finance. Portfolio optimization. Time series classification. Evolutionary algorithm. Machine learning.

# Lista de Figuras

Figura 1 – Documentos por tipo . . . . .	5
Figura 2 – Documentos por ano . . . . .	6
Figura 3 – Documentos por país . . . . .	7
Figura 4 – Documentos por patrocinador . . . . .	7
Figura 5 – Processo de Otimização de Portfólio . . . . .	18
Figura 6 – Fronteira de risco-retorno . . . . .	19
Figura 7 – Classificador linear genérico . . . . .	30
Figura 8 – MLP com uma camada oculta . . . . .	33
Figura 9 – Hiperplano ótimo em problemas de classificação binários . . . . .	36
Figura 10 – Rede Neural Convolutacional para classificação de veículos . . . . .	40
Figura 11 – Exemplo de Árvore de Decisão para problema de classificação binário . . . . .	42
Figura 12 – Empilhamento de 2 níveis . . . . .	44
Figura 13 – Exemplo de dendrograma . . . . .	49
Figura 14 – Representação de um cromossomo . . . . .	50
Figura 15 – Fluxograma básico do algoritmo NSGA-II . . . . .	54
Figura 16 – Ordenação não-dominada . . . . .	55
Figura 17 – Seleção do NSGA-II . . . . .	56
Figura 18 – Exemplo de dois pais para $n = 10$ . . . . .	56
Figura 19 – Filho resultante para exemplo anterior . . . . .	57
Figura 20 – Função de mapeamento . . . . .	58
Figura 21 – Hipervolume de um conjunto de soluções arbitrário . . . . .	59
Figura 22 – Exemplos de indivíduos em Programação Genética . . . . .	60
Figura 23 – Arquitetura da Rede Neural Convolutacional . . . . .	86
Figura 24 – Divisão dos dados em uma avaliação de janelas deslizantes . . . . .	89
Figura 25 – Arcabouço computacional utilizando a Combinação 1 . . . . .	92
Figura 26 – Arcabouço computacional utilizando a Combinação 2 . . . . .	93
Figura 27 – Arcabouço computacional utilizando a Combinação 3 . . . . .	93
Figura 28 – Fronteiras de Pareto dos Métodos 1 e L1 . . . . .	97
Figura 29 – Tempo de execução dos Métodos 1 e L1 . . . . .	97
Figura 30 – Diagramas de caixa da métrica $\Delta$ e hipervolume para os Métodos 2 e L2 . . . . .	98
Figura 31 – Fronteiras de Pareto dos Métodos 2 e L2 . . . . .	98
Figura 32 – Diagrama de caixa do hipervolume para os Métodos 1, 2, 2B, 3, 4 e 4B . . . . .	99
Figura 33 – Diagrama de caixa da métrica $\Delta$ para os Métodos 1, 2, 2B, 3, 4 e 4B . . . . .	99
Figura 34 – Fronteiras de Pareto dos Métodos 1, 2 e 2B . . . . .	100
Figura 35 – Fronteiras de Pareto dos Métodos 3, 4 e 4B . . . . .	101
Figura 36 – Diagrama de caixa e teste de Tukey para métrica $\Delta$ dos Métodos 1 e 3 . . . . .	101

Figura 37 – Fronteiras de Pareto dos Métodos 1 e 3 . . . . .	102
Figura 38 – Tempo de execução para os Métodos 1 e 3 . . . . .	102
Figura 39 – Fronteira de Pareto do Método 1 colorida conforme as cardinalidades das soluções . . . . .	103
Figura 40 – Fronteira de Pareto do Método 3 colorida conforme as cardinalidades das soluções . . . . .	103
Figura 41 – Drawdown máximo e CVaR mensal dos Métodos 1 e 3 . . . . .	105
Figura 42 – Retorno mensal dos Métodos 1 e 3 . . . . .	105
Figura 43 – Retornos acumulados dos Métodos 1 e 3 . . . . .	105
Figura 44 – Retornos acumulados para diferentes valores de custos de transação .	106
Figura 45 – Cardinalidade do portfólio selecionado para diferentes custos de transação	107
Figura 46 – Retornos acumulados para os conjuntos de dados 1, 2 e <i>benchmarks</i> .	108
Figura 47 – Retorno mensal para os conjuntos de dados 1, 2 e <i>benchmarks</i> . . . . .	108
Figura 48 – <i>Drawdown</i> máximo mensal para os conjuntos de dados 1, 2 e <i>benchmarks</i>	109
Figura 49 – CVaR mensal para os conjuntos de dados 1, 2 e <i>benchmarks</i> . . . . .	109
Figura 50 – Retornos acumulados anuais para os conjuntos de dados 1, 2 e <i>benchmarks</i>	110
Figura 51 – Teste de Tukey de precisão para os tamanhos dos conjuntos de teste . .	113
Figura 52 – Teste de Tukey de revocação para os tamanhos dos conjuntos de teste .	114
Figura 53 – Teste de Tukey de revocação para os tamanhos dos conjuntos de treino	114
Figura 54 – Teste de Tukey de F1 para os tamanhos dos conjuntos de teste . . . . .	115
Figura 55 – Teste de Tukey de F1 para os tamanhos dos conjuntos de treino . . . . .	115
Figura 56 – Desempenho dos métodos de seleção de atributos . . . . .	117
Figura 57 – Acurácia dos algoritmos para regra 1 e conjunto de dados de entrada 1	119
Figura 58 – Gráfico de Precisão por revocação para regra 1 e conjunto de dados de entrada 1 . . . . .	119
Figura 59 – Acurácia dos algoritmos para regra 1 e conjunto de dados de entrada 2	120
Figura 60 – Gráfico de precisão por revocação para regra 1 e conjunto de dados de entrada 2 . . . . .	120
Figura 61 – Acurácia dos algoritmos para regra 2 e conjunto de dados de entrada 1	121
Figura 62 – Teste de acurácia para regra 2 e conjunto de dados de entrada 1 . . . . .	121
Figura 63 – Gráfico de precisão por revocação para regra 2 e conjunto de dados de entrada 1 . . . . .	122
Figura 64 – Acurácia dos algoritmos para regra 2 e conjunto de dados de entrada 2	122
Figura 65 – Teste de acurácia para regra 2 e conjunto de dados de entrada 2 . . . . .	123
Figura 66 – Gráfico de precisão por revocação para regra 2 e conjunto de dados de entrada 2 . . . . .	123
Figura 67 – Acurácia dos algoritmos para regra 3 e conjunto de dados de entrada 1	124
Figura 68 – Acurácia dos algoritmos para regra 3 e conjunto de dados de entrada 2	124
Figura 69 – Teste de acurácia para regra 3 e conjunto de dados de entrada 2 . . . . .	125

Figura 70 – Gráfico de precisão por revocação para regra 3 e conjunto de dados de entrada 2 . . . . .	125
Figura 71 – Acurácia dos algoritmos para regra 4 e conjunto de dados de entrada 1 . . . . .	126
Figura 72 – Gráfico de precisão por revocação para regra 4 e conjunto de dados de entrada 1 . . . . .	126
Figura 73 – Acurácia dos algoritmos para regra 4 e conjunto de dados de entrada 2 . . . . .	127
Figura 74 – Teste de acurácia para regra 4 e conjunto de dados de entrada 2 . . . . .	127
Figura 75 – Gráfico de Precisão por revocação para regra 4 e conjunto de dados de entrada 2 . . . . .	127
Figura 76 – Acurácia dos algoritmos para regra 5 e conjunto de dados de entrada 1 . . . . .	128
Figura 77 – Teste de acurácia para regra 5 e conjunto de dados de entrada 1 . . . . .	129
Figura 78 – Gráfico de precisão por revocação para regra 5 e conjunto de dados de entrada 1 . . . . .	129
Figura 79 – Acurácia dos algoritmos para regra 5 e conjunto de dados de entrada 2 . . . . .	130
Figura 80 – Teste de acurácia para regra 5 e conjunto de dados de entrada 2 . . . . .	130
Figura 81 – Gráfico de precisão por revocação para regra 5 e conjunto de dados de entrada 2 . . . . .	131
Figura 82 – Capital acumulado para conjunto de dados 1 . . . . .	132
Figura 83 – Retorno mensal das regras de saída para conjunto de dados 1 . . . . .	132
Figura 84 – CVaR mensal das regras de saída para conjunto de dados 1 . . . . .	132
Figura 85 – Drawdown máximo mensal das regras de saída para conjunto de dados 1 . . . . .	133
Figura 86 – Capital acumulado para conjunto de dados 2 . . . . .	133
Figura 87 – Retorno acumulado dos métodos . . . . .	135
Figura 88 – Comparação de retorno mensal . . . . .	136
Figura 89 – Comparação de drawdown máximo mensal . . . . .	136
Figura 90 – Comparação de CVaR mensal . . . . .	137
Figura 91 – Mapa de calor da seleção de ativo por método . . . . .	138
Figura 92 – Retorno mensal dos métodos . . . . .	139
Figura 93 – CVaR mensal dos métodos . . . . .	140
Figura 94 – Drawdown máximo mensal dos métodos . . . . .	140
Figura 95 – Retorno acumulado dos métodos . . . . .	141

# Lista de Tabelas

Tabela 1 – Tamanhos dos conjuntos de treino para cada tamanho do conjunto de teste	90
Tabela 2 – Porcentagem de utilização dos atributos . . . . .	118

# Lista de Quadros

Quadro 1 – Comparações dos classificadores dos trabalhos relacionados . . . . .	16
Quadro 2 – Ativos selecionados . . . . .	63

# Lista de Algoritmos

Algoritmo 1 – <i>Branch-and-cut</i> genérico . . . . .	29
Algoritmo 2 – Algoritmo de reparo para o caso 1 . . . . .	72
Algoritmo 3 – Algoritmo de reparo para o caso 2 . . . . .	72

# Sumário

<b>1 – Introdução</b>	<b>1</b>
1.1 Definição do Problema	2
1.2 Objetivos	3
1.3 Justificativa	3
1.4 Organização do trabalho	4
<b>2 – Trabalhos Relacionados</b>	<b>5</b>
2.1 Otimização de Portfólios Financeiros	8
2.2 Classificação de Séries Temporais Financeiras	11
2.3 Combinação de Otimização de Portfólios Financeiros e Classificação de Séries Temporais Financeiras	14
<b>3 – Fundamentação Teórica</b>	<b>17</b>
3.1 Otimização de Portfólios Financeiros	17
3.1.1 Desempenho do portfólio	18
3.1.1.1 Variância	20
3.1.1.2 Drawdown	20
3.1.1.3 Value-at Risk	21
3.1.1.4 Conditional Value-at-Risk	21
3.1.2 Modelo média-variância	22
3.1.3 Modelo média-CVaR	23
3.1.4 Restrição de cardinalidade	24
3.1.5 Custos de transação e rebalanceamento	24
3.1.6 Modelos de Otimização de Portfólio com variáveis de decisão inteiras	26
3.2 Otimização Linear Inteira	27
3.2.1 Branch-and-cut	28
3.3 Algoritmos de Classificação	28
3.3.1 Regressão Logística	30
3.3.1.1 Regularização L1	32
3.3.1.2 Regularização L2	32
3.3.2 Perceptron Multicamadas	32
3.3.3 Máquina de Vetores de Suporte	34
3.3.4 Rede Neural Convolutacional	38
3.3.5 Combinação de classificadores	40
3.3.5.1 Ensacamento	41
3.3.5.2 Empilhamento	43

3.3.5.3	Reforço	44
3.3.6	Medidas de desempenho	45
3.3.6.1	Acurácia	46
3.3.6.2	Precisão	46
3.3.6.3	Revocação	46
3.3.6.4	Métrica F1	46
3.3.7	Seleção de atributos	47
3.3.7.1	Seleção de Atributos Baseada em Correlação	47
3.3.7.2	Análise de Componentes Principais	48
3.3.7.3	Agrupamento Hierárquico	48
3.4	Algoritmo Genético	50
3.4.1	Cruzamento	51
3.4.2	Mutação	52
3.4.3	Seleção	53
3.4.4	Algoritmo Genético Multiobjetivo	53
3.4.4.1	NSGA-II aplicado à Otimização de Portfólio	55
3.4.4.2	Medidas de desempenho	57
3.5	Programação Genética	60
<b>4</b>	<b>– Metodologia</b>	<b>62</b>
4.1	Dados de entrada	62
4.2	Otimização de portfólio	63
4.2.1	Modelos de otimização	64
4.2.1.1	Modelo 1	64
4.2.1.2	Modelo 2	65
4.2.1.3	Modelo 3	66
4.2.1.4	Modelo 4	69
4.2.2	Métodos de otimização	70
4.2.2.1	Método 1	70
4.2.2.2	Método 2	73
4.2.2.3	Método 2B	73
4.2.2.4	Método 3	73
4.2.2.5	Método 4	74
4.2.2.6	Método 4B	74
4.2.3	Experimentos	74
4.2.4	Experimentos <i>in-sample</i>	74
4.2.5	Experimentos <i>out-of-sample</i>	75
4.3	Classificação de séries temporais financeiras	75
4.3.1	Atributos de entrada	76
4.3.1.1	Tratamento dos atributos de entrada	81

4.3.2	Seleção de atributos	81
4.3.3	Regras de saída	82
4.3.3.1	Regra 1	82
4.3.3.2	Regra 2	83
4.3.3.3	Regra 3	83
4.3.3.4	Regra 4	83
4.3.3.5	Regra 5	83
4.3.4	Algoritmos de classificação	84
4.3.4.1	Regressão Logística com regularização <i>Ridge</i>	84
4.3.4.2	Floresta Aleatória	84
4.3.4.3	Máquina de Vetores de Suporte	84
4.3.4.4	Perceptron Multicamadas	85
4.3.4.5	Rede Neural Convolutacional	85
4.3.4.6	Adaboost	86
4.3.4.7	XGBoost	86
4.3.4.8	Programação Genética	87
4.3.4.9	Votação	87
4.3.4.10	Empilhamento	88
4.3.5	Avaliação dos modelos de classificação	88
4.3.6	Experimentos	89
4.3.6.1	Experimentos de análise de desempenho	90
4.3.6.2	Experimentos de análise de ganhos	91
4.4	Combinações de otimização de portfólio e classificação de séries financeiras	91
4.4.1	Classificação seguida de otimização	91
4.4.2	Otimização com restrição de classificação	92
4.4.3	Otimização seguida de classificação	92
4.4.4	Experimentos	93
4.5	Ambiente computacional	94
<b>5</b>	<b>Análise e Discussão dos Resultados</b>	<b>95</b>
5.1	Otimização de portfólio	95
5.1.1	Experimentos <i>in-sample</i>	96
5.1.1.1	Comparação dos algoritmos propostos com algoritmos da literatura	96
5.1.1.2	Análise dos métodos heurísticos e exatos	98
5.1.1.3	Análise do efeito da cardinalidade dos portfólios	102
5.1.2	Experimentos <i>out-of-sample</i>	104
5.1.2.1	Análise dos métodos heurístico e exato	104
5.1.2.2	Efeito dos custos de transação nos ganhos financeiros	106

5.1.2.3	Efeito das diferentes granularidades da série histórica nos ganhos financeiros . . . . .	107
5.2	Classificação de séries temporais financeiras . . . . .	111
5.2.1	Análise de desempenho . . . . .	112
5.2.1.1	Ajuste dos tamanhos dos conjuntos de dados de treino e teste	112
5.2.1.2	Análise dos métodos de seleção de atributos . . . . .	116
5.2.1.3	Regra 1 . . . . .	117
5.2.1.4	Regra 2 . . . . .	119
5.2.1.5	Regra 3 . . . . .	122
5.2.1.6	Regra 4 . . . . .	124
5.2.1.7	Regra 5 . . . . .	128
5.2.2	Análise de ganhos financeiros . . . . .	129
5.3	Combinações de otimização de portfólio e classificação de séries financeiras	134
5.3.1	Classificação seguida pela otimização . . . . .	134
5.3.2	Otimização com restrição de classificação . . . . .	134
5.3.3	Otimização seguida pela classificação . . . . .	134
5.3.4	Análise de ganhos financeiros das combinações . . . . .	134
5.3.5	Análise de cardinalidade dos métodos . . . . .	137
5.3.6	Análise do efeito do custo de transação no ganho financeiro . . . . .	139
<b>6 – Conclusão</b>	. . . . .	<b>142</b>
6.1	Trabalhos futuros . . . . .	144
<b>Referências</b>	. . . . .	<b>145</b>

# 1 Introdução

A análise do Mercado Financeiro é uma tarefa que tem sido realizada há bastante tempo por diversos pesquisadores que tentam, principalmente, obter ganhos financeiros a partir dos movimentos do mercado. Apesar de todo esse esforço, o comportamento do mercado tem se mostrando muito complexo e a possibilidade de sua predição tem sido muito questionada.

De acordo com [Murphy \(1999\)](#), o estudo do Mercado Financeiro é feito considerando duas principais escolas de pensamento: a Análise Técnica e a Análise Fundamentalista. Para a Análise Técnica, o mercado de ações pode ser analisado com base apenas em informações de preço e volume de negócios. Essa ideia se baseia na premissa de que tudo que pode afetar o preço de uma ação, como fatores políticos, sociais ou culturais, é incorporado pelo mercado e refletido em seu preço. A abordagem técnica também pressupõe as ideias de que, no mercado financeiro, os preços movem em tendência, ou seja, o sentido do movimento do preço tende a continuar até que a tendência se inverta, apresentando sinais de tal reversão, e de que a história se repete, o que justifica a tentativa de identificação de padrões no movimento desses preços. Ainda segundo [Murphy \(1999\)](#), a Análise Fundamentalista tenta entender a causa dos movimentos no mercado financeiro, enquanto a Análise Técnica se concentra na análise dos efeitos desses movimentos. Dessa forma, é natural que a Análise Técnica seja utilizada em problemas de predição de preços ou tendências no Mercado Financeiro em que a causa dos movimentos não é relevante.

Em conflito com a Análise Técnica, existe a Hipótese de Mercado Eficiente, criada pela comunidade acadêmica, que defende a ideia de que os movimentos no Mercado Financeiro são descritos pelo Movimento Browniano e, dessa forma, as alterações nos preços seriam temporalmente independentes. No entanto, como argumenta [Murphy \(1999\)](#), apesar do intenso debate acadêmico sobre o tema, tal hipótese nunca foi totalmente provada ou refutada e, assim, a realização de experimentos práticos e observações empíricas podem ser mais úteis que a tentativa direta de prová-la ou refutá-la.

Dada a dificuldade e alto risco envolvido no processo de investimento financeiro, a busca pela minimização desse risco é muito desejável. [Markowitz \(1952\)](#) desenvolve a Teoria Moderna do Portfólio, que defende a diminuição do risco do investimento pela diversificação do investimento. Para isso, o conceito de risco é definido como sendo a variância da série histórica de retornos de um dado investimento, enquanto o retorno esperado é estimado pela média da série. Essa medida de risco sofreu algumas críticas devido ao fato da variância considerar tanto variabilidades de retorno abaixo quanto acima da média. Surgiram, então, algumas medidas de risco que consideram apenas os piores retornos, ou as maiores perdas,

como o *Value-at-Risk* (VaR), posteriormente aprimorado para o *Conditional Value-at-Risk* (CVaR), medida de risco considerada coerente e pessimista (ROCKAFELLAR; URYASEV, 2000).

A partir do advento e popularização do computador, surgiram vários modelos matemáticos complexos e técnicas computacionais para analisar o comportamento do mercado, de forma que sistemas financeiros modernos se tornaram fortemente ligados à área de computação (LYUU, 2002). Em se tratando de negociações e investimentos financeiros, mudanças estruturais e o aumento da volatilidade do Mercado Financeiro, propiciam a utilização de técnicas quantitativas, que processam uma grande quantidade de informação. Lyuu (2002) afirma que a maioria das instituições de investimento fornecem modelos matemáticos complexos e técnicas computacionais para seus investidores.

Para o processamento dessa grande quantidade de dados, o uso de técnicas de Inteligência Artificial se torna fundamental, independente da forma como se analisa o mercado. Heurísticas, por exemplo, podem ser utilizadas em tomadas de decisão a partir de modelos matemáticos. Técnicas de Aprendizado de Máquina podem ser utilizadas para predição de preços ou de tendências do mercado, a partir de sua série temporal de preços ou retornos. De acordo com Bishop (2006), técnicas de Aprendizado de Máquina podem ser utilizadas para identificar padrões em sistemas de difícil descrição por meio de modelos matemáticos analíticos, como na análise e predição de séries temporais.

## 1.1 Definição do Problema

Apesar dos grandes esforços na tentativa de explicar o comportamento do mercado a partir de informações do passado, a grande complexidade dessa tarefa impediu que houvesse uma resposta definitiva sobre a possibilidade dessa predição. No entanto, na prática, alguns investidores alcançam altos ganhos financeiros em seus investimentos, enquanto pesquisadores desenvolvem modelos e técnicas cada vez mais complexos no contexto da crescente área de pesquisa de Finanças Computacionais.

Assim, este trabalho aborda, de forma multidisciplinar, os problemas de predição da série temporal e seleção de investimentos no Mercado de Ações, especificamente, uma vez que Murphy (1999) afirma que, inicialmente, ferramentas da Análise Técnica eram aplicadas apenas ao Mercado de Ações e, apenas posteriormente, foram adaptadas para os Mercados Futuro e de Opções. Uma das vantagens da operação no Mercado de Ações é a possibilidade de se recuperar o investimento após uma compra em um momento de descida do preço do ativo comprado, bastando manter o ativo até que seu preço suba. A estratégia comprar e manter (*buy and hold*) não se aplica ao Mercado Futuro, por exemplo, já que futuros são negociados em margens e, como consequência, o impacto do ganho ou perda é amplificado, métricas e indicadores se tornam menos confiáveis e a margem de

erro para o momento exato de negociar se torna muito menor. Dessa forma, o Mercado Futuro não é abordado neste trabalho.

O problema de seleção do conjunto (portfólio) de ativos no mercado, neste trabalho, se baseia nos conceitos de risco e retorno esperados, de forma que, utilizando-se técnicas de Pesquisa Operacional, é possível, a partir de um modelo matemático, obter investimentos com níveis desejáveis de retorno e risco. O problema de predição, neste trabalho, analisa a série temporal de retornos de ativos, com técnicas de Aprendizado de Máquina, na tentativa de classificar sua tendência de movimento no futuro, possibilitando a seleção de ativos com classificação de tendência de alta.

## 1.2 Objetivos

O objetivo geral do trabalho é propor diferentes formas de integrar métodos de classificação de tendência de ativos financeiros com otimização de portfólios e compará-los com integrações presentes na literatura, com o intuito de propor uma metodologia que proporcione ganhos financeiros tanto em investimentos de longo (com duração de meses ou anos) quanto de curto prazo (com duração de horas ou dias) e que supere ganhos econômicos proporcionados por integrações presentes na literatura e pela classificação e otimização utilizadas separadamente.

Para isso, objetivos específicos devem ser alcançados. O trabalho propõe um modelo de otimização de portfólios bem ajustado às restrições práticas de investimento em ações, propõe uma meta-heurística para solucionar esse método e compara diferentes modelos de classificação de tendências.

Os métodos de otimização de portfólio ou de classificação de tendências financeiras devem ser comparados tanto em relação às métricas de desempenho adequadas à avaliação de cada abordagem quanto em relação ao ganho financeiro que eles proporcionam em simulações de operações no Mercado de Ações, de forma que os melhores métodos possam ser utilizados nas análises de combinação de otimização e classificação financeiras.

Finalmente, a comparação das formas de integração propostas neste trabalho com as duas abordagens (classificação de séries e otimização de portfólio financeiro) utilizadas separadamente e com outras formas de integração encontradas na literatura também é um objetivo específico deste trabalho.

## 1.3 Justificativa

A pesquisa proposta contribui, de forma acadêmica, com várias áreas de pesquisa, por seu caráter multidisciplinar. Na área de Finanças, a proposição de novos modelos matemáticos que descrevem e estimam comportamentos de portfólios financeiros, além

da realização de novos experimentos que confrontam a Hipótese do Mercado Eficiente, constituem contribuições relevantes. O desenvolvimento de algoritmos meta-heurísticos específicos para o problema abordado, além do modelo matemático proposto, é uma inovação na área de Pesquisa Operacional. Já o preprocessamento de dados, proposto na classificação de tendência com a comparação de vários modelos para o problema específico deste trabalho, contribuem para a área de Aprendizado de Máquina.

Por fim, as grandes mudanças atuais no Mercado Financeiro brasileiro, como a constante queda na taxa de retorno de investimentos considerados seguros, como a taxa Selic e o Tesouro Direto, desde 2016, trazem também profundas alterações na forma com que a população realiza investimentos financeiros, com a consequência direta de maior interesse em negociações no Mercado de Ações. A análise desse mercado e a implementação de novas técnicas podem auxiliar, de forma prática, a realização dessa forma mais arriscada de investimento, que vem se tornando cada vez mais popular.

## 1.4 Organização do trabalho

O restante do trabalho é composto pelos seguintes capítulos: o Capítulo 2 apresenta os trabalhos relacionados, divididos conforme os temas em comum com este trabalho, sendo eles a otimização de portfólios financeiros, classificação de séries temporais financeiras e a integração dos dois temas. O Capítulo 3 discorre acerca das áreas de conhecimento abordadas no trabalho, necessárias para o entendimento deste, a saber: otimização de portfólios financeiros, otimização linear inteira, algoritmo genético, algoritmos de classificação e programação genética.

O Capítulo 4 descreve a metodologia utilizada para o desenvolvimento do arcabouço computacional proposto e os experimentos realizados a partir deste, além dos dados utilizados para sua alimentação (na Seção 4.1). Esse arcabouço é dividido em três etapas: otimização de portfólio, referente à Seção 4.2, classificação de séries temporais financeiras, referente à Seção 4.3, e a combinação das duas primeiras etapas, Seção 4.4. Os resultados obtidos para os testes propostos, no Capítulo 5, também são divididos entre as etapas de otimização, classificação e integração dos métodos, discutidos nas Seções 5.1, 5.2, e 5.3, respectivamente. Por fim, o Capítulo 6 sintetiza os resultados obtidos até o momento, analisando suas implicações e contribuições acadêmicas e tecnológicas, além de indicar os próximos passos para trabalhos futuros, na Seção 6.1.

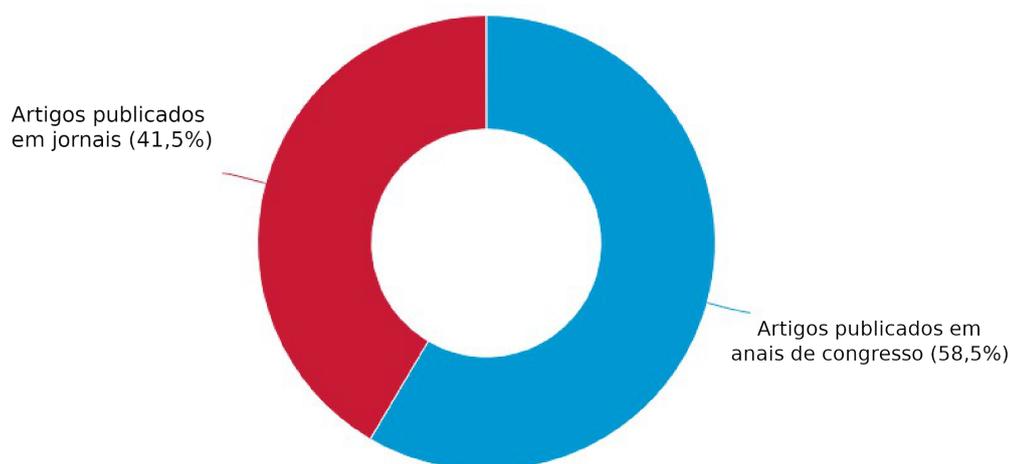
## 2 Trabalhos Relacionados

Os trabalhos relacionados presentes na literatura são analisados e expostos em seções, agrupados de acordo com temas abordados em comum com este trabalho, a saber: Otimização de Portfólios Financeiros, Classificação de Séries Temporais Financeiras e a combinação das duas abordagens.

Primeiramente, para uma análise dos trabalhos que tratam da aplicação da inteligência artificial aos investimentos financeiros, foi considerada uma amostra de trabalhos retirados da Scopus. A consulta, realizada em 27 de maio de 2020, retornou um total de 2326 documentos, incluindo artigos de periódicos e conferências publicados entre 1995 e 2019.

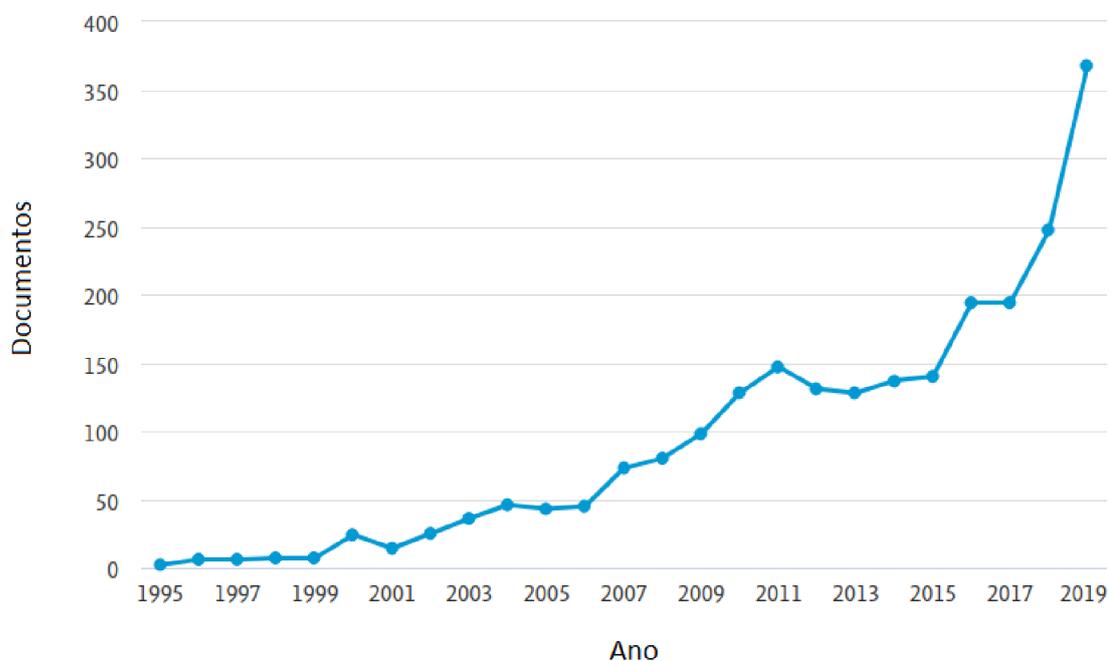
O gráfico da Figura 1 divide os documentos em dois tipos: artigos publicados em periódicos e artigos publicados em jornais ou revistas. É evidente que os trabalhos de conferência são predominantes, mas a diferença entre eles é relativamente pequena.

Figura 1 – Documentos por tipo



O gráfico de documentos por ano, apresentado na Figura 2 mostra o rápido crescimento na quantidade de pesquisas de inteligência artificial para investimentos financeiros, provocado tanto pelo avanço dessa área da computação quanto pelo interesse em investimento no mercado financeiro. Esse crescimento sugere que o tema é atual e ainda possui grande potencial de desenvolvimento acadêmico.

Figura 2 – Documentos por ano



A Figura 3 mostra um gráfico de quantidade de pesquisas por país, indicando a China como o mais produtivo, seguido pelos Estados Unidos. O Brasil, em sétimo lugar, demonstra o interesse em investimentos e o potencial do país para realização de pesquisas nessa área. A posição do Brasil no gráfico espelha o crescimento do interesse das pessoas pelo mercado financeiro e pode ser um indicativo de que haja um crescimento ainda maior de investidores, dada a pequena porcentagem de pessoas que investem nesse mercado atualmente no país.

Pode-se verificar pelo gráfico da Figura 4 que o patrocinador chinês *National Natural Science Foundation of China* é o maior patrocinador para pesquisas dessa área, com uma quantidade de pesquisas muito maior que os outros patrocinadores. Patrocinadores brasileiros ficam na quinta e sexta posições, sendo eles o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Coordenação de Aperfeiçoamento de Nível Superior (CAPES), respectivamente. Dessa forma, a importância das instituições de fomento a pesquisa no Brasil é perceptível.

Figura 3 – Documentos por país

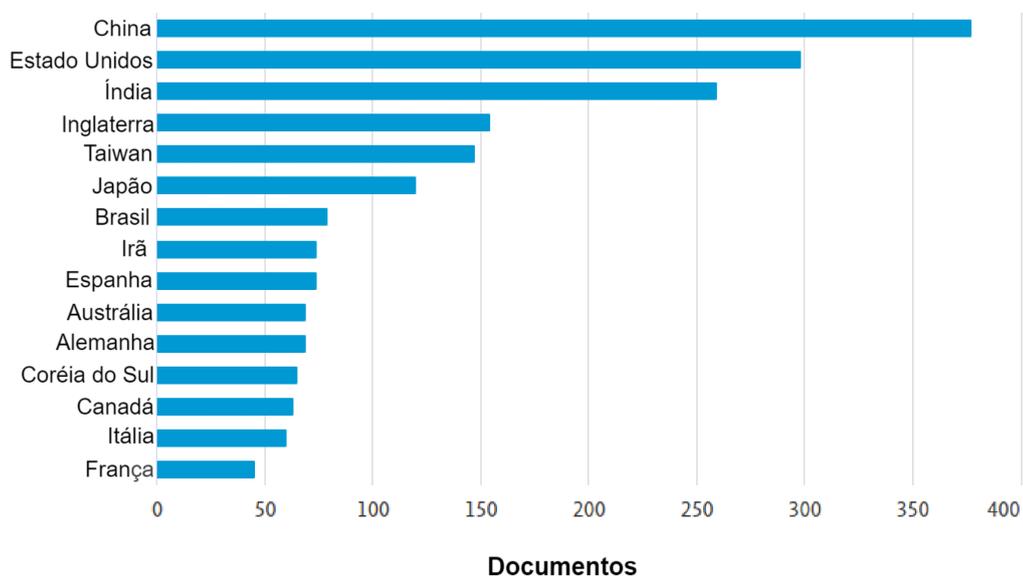
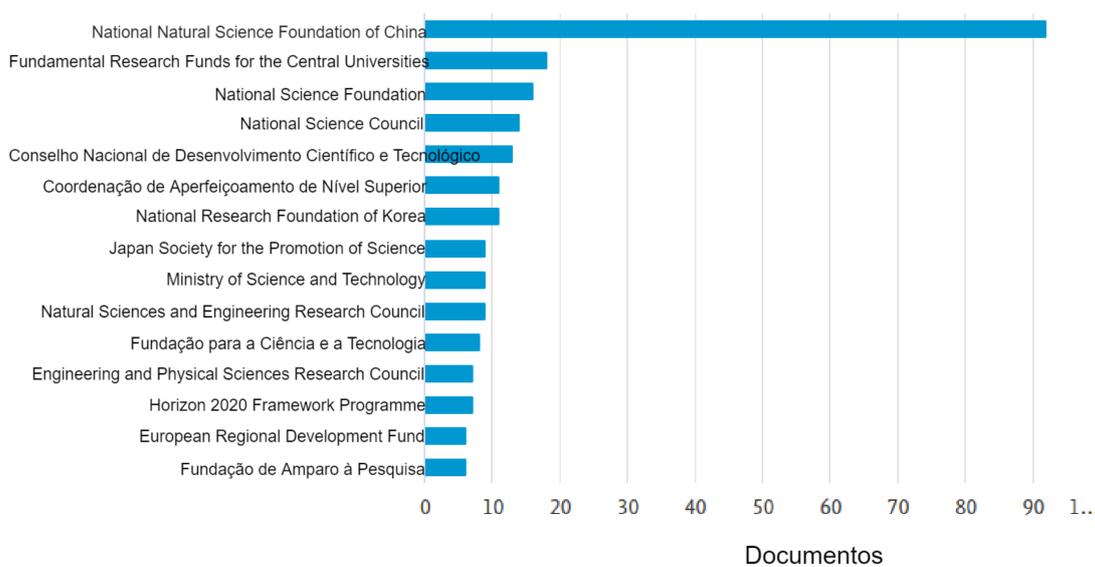


Figura 4 – Documentos por patrocinador



## 2.1 Otimização de Portfólios Financeiros

Deb et al. (2011) propõem um algoritmo NSGA-II personalizado para solucionar o modelo bi-objetivo média-variância de Markowitz (1952) acrescido de restrições de cardinalidade e de proporções de investimentos, que impõem limites inferiores e superiores à quantidade de ativos em um portfólio e à proporção de investimento em cada um desses ativos, respectivamente. De acordo com Deb et al. (2011), a inclusão dessas restrições causa descontinuidade no conjunto de valores factíveis para as variáveis de decisão, aumentando a complexidade do problema e justificando o uso de uma meta-heurística, ainda que ele seja um problema de Programação Quadrática. O algoritmo proposto é constituído por um operador de cruzamento que é uma forma limitada do *Simulated Binary Crossover* (SBX) (AGRAWAL; DEB; AGRAWAL, 1995), que produz soluções com valores entre dois extremos, que correspondem aos limites inferior e superior da proporção de investimento; o operador de mutação polinomial (DEB, 2001) é utilizado para transformar ativos presentes no portfólio original, garantindo que suas proporções de investimento continuem na faixa factível, entre os dois extremos; outro operador de mutação troca uma quantidade determinada de ativos em um portfólios por uma mesma quantidade de ativos que o portfólio original não possui, mantendo a cardinalidade do portfólio inalterada; após a execução de cada um dos operadores, é realizado um procedimento de reparação, que garante que o somatório das proporções de investimentos de cada ativo em um portfólio seja próxima de 1, com uma determinada tolerância. Deb et al. (2011) comparam, graficamente, as soluções encontradas pelo algoritmo proposto e algumas soluções obtidas por programação quadrática do modelo média-variância, sem as restrições de cardinalidade e proporção de investimento, e concluem que o método proposto é eficiente para este tipo de problema, já que as soluções do NSGA-II proposto se aproximam muito das soluções obtidas por um método exato, ainda que este desconsidere a restrição de cardinalidade.

Cheng e Gao (2015) propõem solucionar um modelo de otimização de portfólio média-CVaR com restrição de cardinalidade utilizando um algoritmo de programação matemática. Cheng e Gao (2015) afirmam que o problema de otimização do modelo média-CVaR com restrição de cardinalidade é um problema NP-difícil, cuja complexidade aumenta com a quantidade de ativos e o número de períodos considerados e, por isso, a utilização de um método de otimização inteiro-misto pode apresentar custos computacionais inviáveis para problemas práticos. Assim, propõe-se solucionar um modelo aproximado, resultado da relaxação lagrangiana da restrição de cardinalidade. Uma vez que as variáveis de decisão são variáveis contínuas, que representam as proporções de investimento em cada ativo, e tanto a função objetivo quanto as restrições são funções lineares, um algoritmo de programação matemática linear para problemas de variáveis reais, como o Simplex, pode ser utilizado. Resultados mostram que a relaxação lagrangiana realmente limita a quantidade de ativos no portfólio e que a abordagem proposta pode ser utilizada em problemas práticos de grandes

dimensões, com centenas de ativos e centenas de milhares de períodos. O trabalho, porém, não compara os resultados obtidos com resultados da otimização inteira-mista do modelo média-CVaR original, mesmo para problemas de pequena escala, ou com resultados obtidos por meta-heurísticas.

[Soleimani, Golmakani e Salimi \(2009\)](#) é um dos poucos trabalhos que utilizam quantidades de lotes dos ativos como variáveis de decisão, sendo que a maioria dos trabalhos de otimização de portfólios optam por utilizar a proporção de investimento dos ativos. O trabalho parte do modelo média-variância monobjetivo, com objetivo de minimização da variância do portfólio e restrição de retorno esperado mínimo. Ao modelo, são acrescentadas restrição de cardinalidade fixa e restrição de investimento conforme a capitalização de mercado em cada setor, de forma que o portfólio deve conter uma maior proporção de investimento em setores de maior capitalização. Para a solução do modelo proposto, é desenvolvido um algoritmo genético, que executa o operador de cruzamento *Random Assorting Recombination* (RAR) ([RADCLIFFE, 1991](#)) até que soluções factíveis sejam geradas; posteriormente, é aplicado um operador de mutação não descrito no trabalho; e para a próxima geração, seleciona-se a metade mais apta dos filhos gerados, completando a população selecionando pais de forma aleatória. [Soleimani, Golmakani e Salimi \(2009\)](#) consideram um problema de pequena dimensão, com poucos ativos, para comparar a melhor solução obtida pelo algoritmo genético com o ótimo global obtido utilizando o *software* LINGO e concluem que as soluções obtidas pelos dois métodos são muito próximas, sem, no entanto, realizar alguma análise estatística para apoiar tal afirmação, já que o algoritmo genético é um algoritmo estocástico. Por fim, o método proposto se mostra capaz de oferecer soluções para problemas de grande escala, considerando até 2000 ativos disponíveis para investimento.

O trabalho de [Hanaoka \(2014\)](#), assim como o de [Soleimani, Golmakani e Salimi \(2009\)](#), utiliza quantidades de lotes dos ativos como variáveis de decisão em seu modelo média-variância de otimização de portfólios, mas considera apenas a restrição de cardinalidade fixa e possui, como diferencial, a inclusão de custos de transação. O modelo proposto é multiobjetivo e para solucioná-lo, são utilizados os algoritmos *Pareto Differential Evolutionary Approach* (PDEA) ([MADAVAN, 2002](#)) e *Differential Evolution for Multiobjective Optimization* (DEMO) ([ROBIČ; FILIPIČ, 2005](#)), versões multiobjetivo do Algoritmo de Evolução Diferencial (ED) ([STORN; PRICE, 1995](#)), com operadores próprios deste mesmo algoritmo e métodos de reparo propostos, diferenciando-se apenas por seus mecanismos de seleção. [Hanaoka \(2014\)](#) realiza uma simulação de investimento utilizando seu método, com a estratégia de comprar conforme os resultados obtidos e manter o investimento por um ano. O investimento simulado proporciona lucro e supera o índice Bovespa durante todo o período considerado. Porém, pode-se fazer uma crítica em relação ao algoritmo utilizado, que possui operadores próprios para problemas com variáveis reais, enquanto o modelo proposto possui variáveis de decisão inteiras.

Barroso (2017) compara dois modelos de otimização de portfólio. O primeiro é o modelo proposto por Deb et al. (2011) e o segundo é o modelo média-CVaR, que utiliza as mesmas restrições e variáveis de decisão que o primeiro, diferenciando-se apenas pelas medidas de riscos utilizadas (variância e CVaR, respectivamente). O algoritmo proposto para solucionar os modelos é uma modificação do NSGA-II proposto em Deb et al. (2011), que inclui um terceiro operador de mutação que pode alterar a cardinalidade de um portfólio, diferente dos operadores de mutação do algoritmo original. Testes estatísticos são realizados para comparar os portfólios obtidos na solução de cada modelo e, apesar de não serem detectadas diferenças entre as médias de retorno, risco, *Drawdown* ou *Drawup* mensais para cada modelo, simulações de investimento no mercado revelam maiores ganhos quando a medida CVaR é utilizada.

O custo de transação em problemas de otimização de portfólios é um tema muito abordado no trabalho de Mitchell e Braun (2013), no qual são propostos modelos com rebalanceamento de acordo com diversas formas de se representar esses custos. Os modelos propostos são variações do modelo média-variância monobjetivo de Markowitz (1952), incluindo diferentes tipos de custos de transação: convexo, linear por partes, quadrático e quadrático por partes. Também é proposto uma nova medida de risco (risco escalado), que consiste em dividir a variância do risco pela quantia realmente investida, desconsiderando os custos de transação envolvidos. É importante notar que, como o trabalho utiliza proporções de investimento dos ativos como variáveis de decisão, custos de transação fixos, em unidade monetária, não podem ser diretamente considerados nesses modelos.

Ferreira (2018) propõe um modelo multiobjetivo média-CVaR de otimização de portfólios com restrição de cardinalidade fixa, incluindo de custos de transação fixos e proporcionais, além do processo de rebalanceamento. Pode-se destacar que, diferente do trabalho de Mitchell e Braun (2013), a utilização de quantidades de lotes dos ativos como variáveis de decisão no trabalho de Ferreira (2018) permite a inserção do custo fixo, representado em unidade monetária. Para a solução do modelo, são propostos algoritmos NSGA-II e SPEA 2, com operador de ponto único para cruzamento, operador de ponto para mutação e métodos de reparo propostos para o modelo específico. É realizada a comparação dos algoritmos propostos com o PDEA, proposto por Hanaoka (2014) e, considerando as métricas *coverage*, *spacing* e hipervolume, as análises de variância seguidas de testes de comparações múltiplas apontam o NSGA-II como o melhor método e o PDEA como o pior. Também é proposto um modelo linear aproximado, solucionado utilizando-se o método exato *branch-and-cut*, que proporciona soluções melhores que as meta-heurísticas propostas, mas com uma diferença muito pequena e tempo de execução muito maior. Uma crítica que pode ser feita ao trabalho de Ferreira (2018) é que, por utilizar restrição de cardinalidade fixa, realiza todas as análises para valores arbitrários de cardinalidade dos portfólios.

Recentemente, o trabalho de [Chen e Wei \(2019\)](#) propõe um modelo multiobjetivo de otimização robusta baseado no modelo média-variância de [Markowitz \(1952\)](#) e elabora um algoritmo *Particle Swarm Optimization* (PSO) ([KENNEDY; EBERHART, 1995](#)) multiobjetivo para o problema específico. [Silva, Herthel e Subramanian \(2019\)](#) também utilizam um algoritmo PSO multiobjetivo para a otimização de portfólios financeiros. Nesse caso, é proposto um PSO com *ranks* para solucionar o modelo média-variância com restrição de cardinalidade variável. Já [Kaucic, Moradi e Mirzazadeh \(2019\)](#) criam um NSGA-II e um SPEA 2 com operadores específicos para três diferentes modelos multiobjetivos que pretendem maximizar o retorno e minimizar o risco. Os modelos se diferem em relação à medida de risco utilizada, sendo elas a semi-variância, CVaR e uma combinação de ambos. A grande quantidade de trabalhos recentes que abordam o problema de otimização de portfólios financeiros é uma indicação de que esse ainda é um problema em aberto. Percebe-se também que, apesar de ser um tema já discutido por muitas décadas, a otimização de portfólios é uma área ainda em desenvolvimento, com muitas contribuições recentes.

## 2.2 Classificação de Séries Temporais Financeiras

[Krauss, Do e Huck \(2016\)](#) utilizam Redes Neurais Artificiais Profundas, *Gradient-Boosted Tree* (GBT) ([FRIEDMAN, 2001](#)) e *Random Forest* na tentativa de identificar padrões e prever tendências nos movimentos de preço do índice S&P 500, composto por ativos da bolsa do *New York Stock Exchange* (NYSE) e NASDAQ. Para isso, são treinados dados históricos diários que consistem de retornos espaçados. Quarenta atributos são utilizados para cada período e 510 períodos são utilizados no treinamento. A saída é composta por duas classes: classe 1, caso o retorno de um ativo esteja acima da mediana dos retornos de todos os ativos no próximo período e classe 0, caso contrário. Três combinações desses classificadores foram propostas: a primeira realiza uma votação ponderada pela probabilidade de cada ativo de pertencer a cada uma das classes; a segunda também consiste em uma votação que pondera, além das probabilidades, o índice de Gini das predições de cada classificador; o terceiro se difere do segundo por utilizar a ordem dos classificadores em relação ao índice de Gini apresentado, em vez de usar o valor absoluto do índice. Simulações de investimentos guiados pelas decisões dos classificadores indicam que o *Random Forest* proporciona melhores ganhos financeiros entre os classificadores individuais e todas as combinações superam todos os classificadores individuais. O trabalho não realiza validação dos hiperparâmetros, análises estatísticas dos ganhos financeiros obtidos e faltam comparações dos classificadores com base em métricas como precisão e acurácia.

[Hsu et al. \(2016\)](#) compara algoritmos de Aprendizagem de Máquina (classificadores) e métodos econométricos. Para o primeiro são utilizados Redes Neurais Artificiais (RNA) e SVM, enquanto que para o segundo são utilizados os modelos de autoregressão (AR),

Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (ENGLE, 1982) e *Autoregressive Integrated Moving Average* (ARIMA) (BOX; JENKINS, 1970). Testes estatísticos apontam maiores ganhos quando se utiliza métodos de Aprendizagem de Máquina. Além disso, o SVM apresenta acurácia maior que a RNA e comparando treinamentos utilizando dados diários e a cada hora, testes sugerem que apesar de não haver diferença entre as acurácias apresentadas por cada abordagem, a utilização de uma periodicidade menor (em horas) proporciona maiores ganhos financeiros. Como entrada dos classificadores são utilizados o preço máximo, mínimo, de abertura e de fechamento, além de doze indicadores técnicos, nos quais testes estatísticos não detectaram diferenças em serem ou não utilizados. Para a saída, duas classes são consideradas: classe 1, caso o preço de um ativo em um dado período seja menor que o preço no próximo período e classe 0, caso contrário.

Marcelino (2016) utiliza algoritmos de regressão e de classificação para a seleção de ativos financeiros na composição de um portfólio. Como entrada dos algoritmos são utilizados 127 indicadores disponíveis na plataforma Bloomberg e a saída é definida pela seguinte regra: classe 1, se o retorno de um ativo superar um quarto dos retornos de todos os ativos no próximo período e classe 0, caso contrário. O algoritmo de classificação utilizado é o SVM, que apresenta acurácia superior a 80% em alguns casos. Os 25% ativos com maiores probabilidades de pertencerem à classe 1 são selecionados para comporem o portfólio. Marcelino (2016) também elabora um modelo de regressão, utilizando o algoritmo *Support Vector Regression* (SVR) (DRUCKER et al., 1997) e, como na classificação, os 25% melhores ativos são selecionados para o portfólio. Baseado em uma simulação de negócios utilizando ambas as abordagens, conclui-se que o modelo de regressão proporciona ganhos financeiros maiores que o modelo de classificação, considerando apenas o ganho apresentado ao final de um período determinado arbitrariamente. A escolha da métrica acurácia para avaliar o método de classificação não condiz com a estratégia adotada de selecionar os melhores ativos para a composição do portfólio, sendo que a precisão poderia ter sido mais interessante. Ademais, um método que apenas classifique todos ativos como classe 0 apresentaria acurácia de 75%, aproximando-se muito da acurácia obtida pelo classificador. Por fim, a utilização de um método de otimização de portfólio poderia aprimorar os ganhos proporcionados.

Thakur e Kumar (2018) utilizam o classificador *Weighted Multicategory Generalized Eigenvalue Proximal Support Vector Machine* (WMGEP SVM) (KUMAR; THAKUR, 2016) na predição de séries temporais financeiras. São considerados 25 atributos entre informações de preço e volume e indicadores técnicos, selecionados utilizando um classificador *Random Forest*, que possui a propriedade de indicar a importância de cada atributo e, assim, os melhores atributos podem ser selecionados. Três classes são definidas como saída: classe 1, atribuída aos ativos em que, em um dado período, apresentam preço máximo menor que o fechamento do próximo período e preço mínimo menor que o preço mínimo do próximo período; classe -1, para os ativos que, em um dado período, apresentam preço mínimo

maior que o preço de fechamento do próximo período e preço máximo maior que o preço máximo do próximo período; e classe 0, caso contrário. Resultados apresentam apenas ganhos obtidos em simulações de investimento em índices como S&P 500, NASDAQ e Dow Jones. Métricas de avaliação do classificador, como acurácia e precisão não são apresentadas.

O trabalho de Ghazanfar et al. (2017) compara vários classificadores para predição de tendências no *Karachi Stock Exchange* (KSE). São utilizados o SVM, MLP, *Naive Bayes* (KONONENKO, 1989), *K-Nearest Neighbors* (KNN) (COVER; HART et al., 1967), *AdaBoost* (FREUND; SCHAPIRE, 1996) e a rede neural *Radial Basis Function* (RBF) (ORR et al., 1996). Foram utilizados apenas cinco atributos como entrada para os classificadores: preço máximo, mínimo, de abertura, de fechamento e volume. Três classes de saída foram consideradas: classe A, para ativos com retorno maior que zero, classe B, para ativos com retorno menor que zero e classe C, para os demais ativos. Apenas os três melhores classificadores, de acordo com Ghazanfar et al. (2017), têm seus resultados apresentados: o MLP, o *AdaBoost* e a Rede Bayesiana. Os ganhos financeiros proporcionados, em simulações, por cada classificador e suas respectivas acurácias são apresentados separados por setores de mercado e, para alguns setores, as acurácias apresentadas chegam a 100%. A validação realizada no trabalho é a validação cruzada *2-fold*, incompatível com a classificação de séries temporais por utilizar dados passados para validar o treinamento em um período futuro.

Combinações de classificadores para a predição de séries financeiras são utilizadas em Fu et al. (2018), que propõe modelos de classificadores com atributos selecionados por um Algoritmo Genético, considerando um total de 244 indicadores fundamentalistas e técnicos obtidos de séries diárias da bolsa de valores chinesa. Os classificadores individuais considerados são a Regressão Logística, *Random Forest* (RF) e *Redes Neurais Artificiais Profundas*. A técnica de combinação dos classificadores é o *stacking*, utilizado no trabalho para combinar os classificadores individuais dois a dois, de forma que três metaclassificadores são desenvolvidos, no total. Para a saída, é proposta uma regra que, calculadas as razões de retorno sobre volatilidade para cada ativo durante um período determinado, os Q% ativos com maior valor são atribuídos à classe 1, os Q% ativos com menor valor, atribuídos à classe 0 e os demais ativos são desconsiderados. O melhor resultado reportado é referente ao metaclassificador que utiliza o RF e *Redes Neurais Profundas*, com acurácia média de 0,927 e precisão de 0,919. Apesar de Fu et al. (2018) concluírem que esse metaclassificador é melhor que os classificadores individuais e os demais metaclassificadores, nenhum teste estatístico é realizado para validação dessa afirmação e, aparentemente, os altos valores obtidos de acurácia e precisão obtidos são resultados da desconsideração de grande parte dos ativos que ficam entre as classes 0 e 1, que provavelmente seriam mais difíceis de serem preditos e, conseqüentemente, reduziriam as taxas de acerto.

Assis (2019) propõe modelos de classificação de tendência para nove ativos da bolsa de valores brasileira B3 utilizando, primeiramente, os classificadores RF, SVM, MLP, Árvore de Decisão C4.5 (QUINLAN, 1993), Otimização Mínima Sequencial (PLATT, 1998) e Redes Bayesianas (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997). As entradas consideradas são compostas por informações diárias de preços de abertura, de fechamento, máximo e mínimo, além do volume de negociação e a saída define classe 1 para os ativos em que, em um dado período, seu preço de fechamento é menor que o fechamento do próximo período e classe 0, caso contrário. Em seguida, Assis (2019) propõe seis metaclassificadores utilizando a técnica de combinação *stacking*, sendo que cada metaclassificador possui, como classificador de nível 2, um dos classificadores individuais considerados no trabalho. Para a comparação de classificadores, é escolhida a métrica acurácia. Por fim, os resultados obtidos permitem a conclusão de que os metaclassificadores apresentam, em média, acurácia maior que o melhor classificador individual. A maior acurácia média apresentada é a do metaclassificador com classificador nível 2 SVM, que apresentou um valor de 54,33%. O trabalho, porém, falha em não validar a escolha de hiperparâmetros, sendo que estes são escolhidos de forma arbitrária.

## 2.3 Combinação de Otimização de Portfólios Financeiros e Classificação de Séries Temporais Financeiras

Duarte (2018) propõe a combinação da classificação de séries financeiras com uma posterior otimização de portfólios financeiros que considera os resultados obtidos pelo processo de classificação. A classificação utiliza, como entrada, 22 atributos entre log-retornos espaçados e indicadores técnicos, obtidos das informações diárias de preços de abertura, fechamento, máximo e mínimo, além do volume de negociações de 137 ativos da B3 em um período de 11 anos e 6 meses. Para a saída, são classificados positivamente os ativos que apresentam um retorno desejado no próximo período e, negativamente, os demais ativos. Os algoritmos utilizados foram o SVM e o MLP, com diferentes subconjuntos de entrada e tamanhos de janela de treinamento. Para a otimização de portfólios, foram utilizados os modelos de Black-Litterman (BL) e o modelo média-variância monobjetivo, solucionados por métodos exatos de programação matemática. Para o modelo BL, os resultados das classificações foram introduzidos como visões do investidor e no modelo média-variância, são considerados apenas os ativos classificados positivamente em cada período. Para os classificadores, o SVM aparentemente obtém precisões melhores que o MLP, em média. O modelo BL apresenta retornos acumulados muito próximos do modelo média variância, utilizando-se o classificador MLP, mas proporciona ganhos pouco maiores quando se utiliza SVM, considerando todo o período analisado. Em todos os casos, a combinações das duas abordagens mostra-se mais eficiente do que cada abordagem utilizada separadamente, em termos de ganhos financeiros proporcionados.

O trabalho de [Hanaoka \(2019\)](#) também aplica a otimização de portfólios após a classificação de séries financeiras. Para a classificação, são considerados 30 atributos entre diferentes log-retornos e indicadores técnicos obtidos a partir de dados diários de 115 ativos que participaram da B3 entre janeiro de 2004 e dezembro de 2016. A regra de saída da classificação é a mesma utilizada em [Duarte \(2018\)](#) e os algoritmos utilizados são SVMs com diferentes valores de hiperparâmetros e um metaclassificador que realiza uma votação considerando todos esses classificadores. A otimização de portfólio considera o modelo média-CVaR monobjetivo, com restrição de cardinalidade variável, tendo proporções de investimento dos ativos como variáveis de decisão. Para avaliar os classificadores, é selecionada a métrica precisão e mostra-se que a combinação de todos os classificadores não supera o melhor dos classificadores individuais, mas se aproxima muito dele. É possível concluir, a partir de simulações de negociações, que a combinação da classificação de séries financeiras e otimização de portfólio supera cada uma das abordagens utilizadas separadamente. [Hanaoka \(2019\)](#) apresenta testes estatísticos que apontam a superioridade, em média, dos retornos diários obtidos quando se utiliza a abordagem combinada proposta no trabalho.

O quadro 1 indica a presença de alguns elementos nos classificadores dos trabalhos relacionados comuns aos classificadores desenvolvidos no presente trabalho. Esses aspectos do processo de classificação comparados no quadro dizem respeito aos algoritmos de classificação utilizados em cada trabalho, à presença ou ausência de metaclassificadores, à granularidade ou periodicidade dos dados utilizados no treinamento dos classificadores e às regras utilizadas para definir as saídas ou as variáveis alvo dos classificadores, sendo a regra 1 aquela descrita para o trabalho de [Hsu et al. \(2016\)](#) e [Assis \(2019\)](#), regra 2, utilizada por [Duarte \(2018\)](#) e [Hanaoka \(2014\)](#), regra 3 utilizada por [Thakur e Kumar \(2018\)](#), regra 4 por [Krauss, Do e Huck \(2016\)](#) e a regra 5 em [Marcelino \(2016\)](#). O "x" em uma célula indica a presença da característica daquela coluna no trabalho da respectiva linha e a ausência do "x", a ausência da respectiva característica. Por fim, as linhas com índice em negrito representam trabalhos que combinam classificadores com otimização de portfólio. Em relação aos trabalhos representados por números nas linhas do quadro, têm-se:

1. ([KRAUSS; DO; HUCK, 2016](#));
2. ([HSU et al., 2016](#));
3. ([MARCELINO, 2016](#));
4. ([THAKUR; KUMAR, 2018](#));
5. ([GHAZANFAR et al., 2017](#));
6. ([FU et al., 2018](#));
7. ([ASSIS, 2019](#));
8. ([DUARTE, 2018](#));
9. ([HANAOKA, 2019](#));
10. Presente trabalho.



## 3 Fundamentação Teórica

A fundamentação teórica necessária para o devido entendimento do trabalho é apresentado em cinco diferentes seções, conforme a área a que pertencem, sendo essas: Otimização de Portfólios Financeiros, Otimização Linear Inteira, Algoritmos de Classificação, Algoritmo Genético e Programação Genética.

### 3.1 Otimização de Portfólios Financeiros

Inicialmente, alguns conceitos sobre finanças devem ser entendidos para que se possa compreender o conteúdo da Otimização de Portfólios. Ativo é um conceito importante, que representa qualquer instrumento financeiro negociável específico, como ações de uma empresa. Essas negociações são realizadas por investidores, que possuem uma certa quantia de dinheiro, denominada capital, para investir em um conjunto de ativos, os ativos disponíveis. A escolha de proporções do capital ou quantidade de lotes investidos em cada ativo forma o portfólio financeiro do investidor.

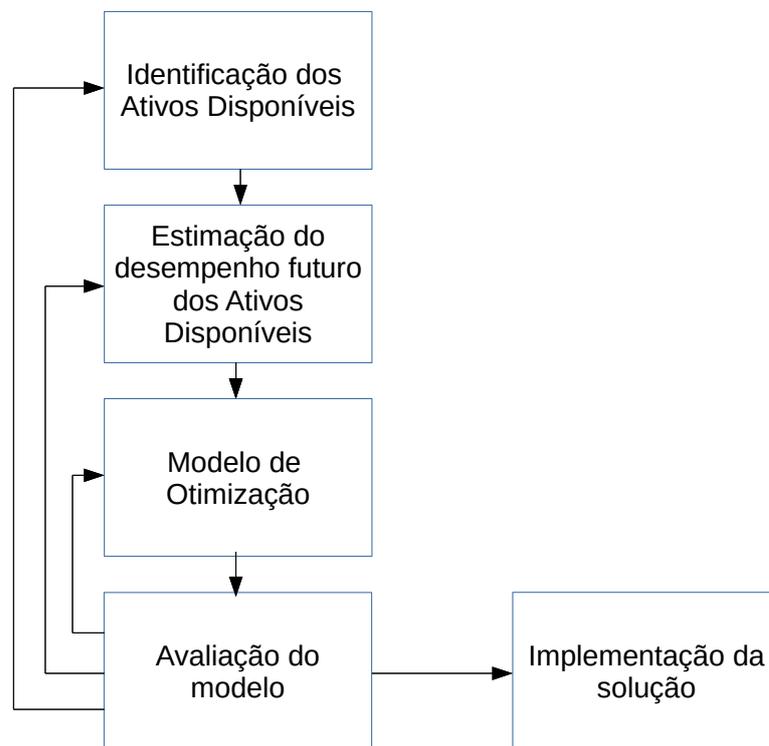
O ativo é caracterizado por sua cotação ou preço de mercado, que representa o seu valor econômico de compra ou venda. Denominando  $p_t$  esse preço de mercado ou cotação de um ativo em um tempo  $t$ , o retorno  $r_t$  desse ativo no período de  $t - 1$  a  $t$  representa o seu ganho proporcional sendo definido como  $r_t = (p_t - p_{t-1})/p_{t-1}$ . Pode-se imaginar, então, que ativos que apresentam bons históricos de retornos, possuam boas chances de manter esse alto retorno no futuro e, assim, o investimento de todo o capital no ativo de melhor histórico seria razoável. Porém, como defende [Markowitz \(1952\)](#), a incerteza do mercado faz com que esse portfólio de um único ativo possua um risco muito elevado, se comparado com portfólios diversificados, ou seja, compostos por vários ativos. Dessa forma, a Otimização de Portfólios deve considerar tanto o potencial ganho ou retorno esperado do investimento quanto seu risco.

[Mansini, Ogryczak e Speranza \(2015\)](#) divide todo o processo de otimização de portfólio em cinco partes diferentes: identificação dos ativos disponíveis, estimação do desempenho futuro desses ativos, modelo de otimização, avaliação do modelo e implementação da solução. Até a fase de implementação, as fases podem seguir caminhos cíclicos, como descrito na Figura 5.

Na fase de identificação dos ativos disponíveis, o investidor decide os tipos de mercado financeiro em que deseja investir, a estimativa do desempenho futuro dos ativos consiste em estabelecer métricas estatísticas para caracterizar os ativos com base em dados históricos de cotações, o modelo de otimização guiará a forma com a qual o conjunto

de ativos para investimento e suas respectivas quantidades ou proporções é selecionado, a validação do modelo é feita, geralmente, por simulações de negociações no mercado guiadas por esse modelo e a implementação da solução é o investimento, realizado na prática. Esse investimento deve ter um tempo de duração estipulado, sendo a estratégia nesse período denominada comprar e manter (*buy and hold*, em inglês). Este capítulo se concentrará, principalmente, na fase de modelo de otimização, devido à sua grande importância para o trabalho.

Figura 5 – Processo de Otimização de Portfólio



Fonte: Adaptado de [Mansini, Ogryczak e Speranza \(2015\)](#)

### 3.1.1 Desempenho do portfólio

Portfólios de alto desempenhos são aqueles com grande potencial de ganhos financeiros e baixa probabilidade de proporcionar ganhos abaixo do esperado. Percebendo a variação de preço do portfólio como um evento aleatório, pode-se modelar o ganho ou retorno  $R_{\mathbf{x}}$  de um portfólio  $\mathbf{x}$ , em cada período, como uma variável aleatória, de forma que  $R_{\mathbf{x}} = \sum_{j=1}^n R_j x_j$ , com  $j = 1, \dots, n$  representando cada um dos  $n$  ativos disponíveis e  $x_j$  sendo a proporção do capital investida no ativo  $j$ . Assim, o retorno esperado de um portfólio pode ser definido como o valor esperado de retorno, estimado pela média histórica  $\mu$  dos seus retornos que, por sua vez, é uma média ponderada das médias históricas  $\mu_j$  de cada

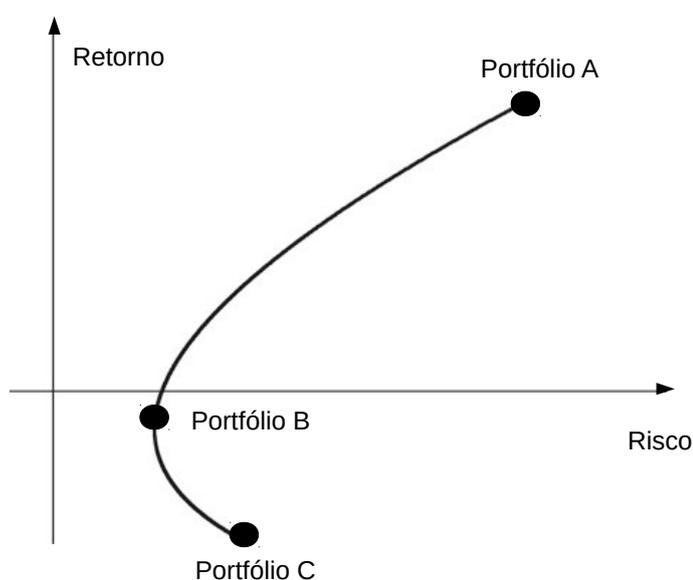
ativo disponível  $j$  (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$\mathbb{E}(R_{\mathbf{x}}) = \mu(\mathbf{x}) = \sum_{j=1}^n \mu_j x_j.$$

Enquanto o potencial ganho futuro de um portfólio é geralmente estimado pela média histórica dos retornos desse portfólio, a incerteza ou risco de um portfólio pode ser estimado por várias métricas propostas para tal finalidade. Independente da métrica utilizada, existe uma fronteira eficiente de retorno e risco, em que, para cada valor de risco, o portfólio pertencente a esta fronteira, denominado portfólio não dominado, possui o maior valor de retorno esperado entre todos os portfólios que apresentam risco menor ou igual a esse. Os portfólios que não pertencem à fronteira eficiente são ditos dominados.

A Figura 6 considera um exemplo simples em que existem apenas dois ativos disponíveis. Nesse caso, o Portfólio A, portfólio de maior retorno, é composto de um desses ativos, apenas, enquanto o Portfólio C, portfólio de menor retorno, é composto apenas pelo outro ativo. A curva que liga esses dois portfólios é composta por portfólios com diferentes proporções de cada um desses ativos. O Portfólio B é uma determinada combinação dos dois ativos que produz o portfólio de menor risco. Assim, a fronteira eficiente é composta pelos portfólios que estão entre os Portfólios A e B e, conseqüentemente, entre os Portfólios B e C estão os portfólios dominados.

Figura 6 – Fronteira de risco-retorno



Este capítulo apresenta algumas medidas de risco utilizadas no trabalho. Porém, são diversas as medidas de risco presentes na literatura. Informações adicionais sobre essas medidas podem ser obtidas em Mansini, Ogryczak e Speranza (2015).

### 3.1.1.1 Variância

Proposta por [Markowitz \(1952\)](#), a variância  $\sigma^2(\mathbf{x})$  de um portfólio  $\mathbf{x}$  é uma medida de incerteza do retorno esperado desse portfólio:

$$\sigma^2(\mathbf{x}) = \mathbb{E}\{(R_{\mathbf{x}} - \mathbb{E}(R_{\mathbf{x}}))^2\} = \mathbb{E}\{(R_{\mathbf{x}} - \mu_{\mathbf{x}})^2\}.$$

Assim, [Markowitz \(1952\)](#) utiliza o conceito de covariância dos retornos dos ativos, par a par, de forma que a variância do portfólio se torne uma função quadrática em função pesos ou proporções dos ativos. A covariância  $\sigma_{ij}$  entre os ativos  $i$  e  $j$  pode ser escrita da seguinte forma:

$$\sigma_{ij} = \mathbb{E}\{(R_i - \mu_i)(R_j - \mu_j)\}.$$

Por fim, [Markowitz \(1952\)](#) desenvolve as equações propostas, chegando na seguinte equação para variância de um portfólio  $\mathbf{x}$ :

$$\sigma^2(X) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j.$$

Críticas a essa medida de risco são feitas devido à variância aumentar tanto com retornos muito baixos, quanto com retornos muito altos que, na prática, é desejável aos investidores. Surgem, posteriormente, medidas de risco que consideram apenas a cauda referente aos menores ganhos, ou maiores perdas, na distribuição de retornos.

### 3.1.1.2 Drawdown

*Drawdown* é um importante conceito em finanças que indica perdas financeiras em um determinado período. Para uma série histórica de cotações, *drawdown* de um portfólio  $\mathbf{x}$  é o valor da perda entre o preço máximo naquele período e o preço em um tempo  $t$  após a ocorrência desse máximo. Sua fórmula é ([LAVRADO, 2015](#)):

$$DD_t(\mathbf{x}) = \max\left(0, \frac{p_{max}(\mathbf{x}) - p_t(\mathbf{x})}{p_{max}(\mathbf{x})}\right)$$

onde  $p_t(\mathbf{x})$  é o preço do portfólio  $x$  no tempo  $t$  e  $p_{max}(\mathbf{x})$  é o valor do preço máximo do portfólio em um tempo  $t' \leq t$ .

Para uma série histórica de  $T$  cotações, cada tempo ou cenário  $t \in \{1, \dots, T\}$  possui um valor de *drawdown*. Em finanças, é comum a utilização do *drawdown* médio (ADD) ou do *drawdown* máximo (MDD), sendo que  $ADD(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T DD_t(\mathbf{x})$  e  $MDD(\mathbf{x}) = \max\{DD_t(\mathbf{x})\}$ . A partir desses conceitos, alguns trabalhos propõem medidas de risco, como o *Maximum Drawdown at Risk* ([LAVRADO, 2015](#)).

### 3.1.1.3 Value-at Risk

O VaR (*Value-at-Risk*) é uma medida de risco que, diferente da variância, considera apenas os menores retornos ou maiores perdas. Assim, a medida de risco VaR considera valores de perda dos portfólios, que são valores opostos do retorno, de forma que  $L_{\mathbf{x}} = -R_{\mathbf{x}}$ , com  $L_{\mathbf{x}}$  representando as perdas do portfólio  $\mathbf{x}$ . A medida VaR é o valor que divide as  $\beta\%$  piores perdas das demais perdas, ou seja, a probabilidade  $P(L_{\mathbf{x}} > \alpha)$  de uma perda do portfólio  $\mathbf{x}$  exceder o risco VaR  $\alpha$  deve ser menor ou igual a  $\beta$  e a probabilidade  $P(L_{\mathbf{x}} \leq \alpha)$  da perda estar abaixo do VaR deve ser maior ou igual a  $\beta$  (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$P(L_{\mathbf{x}} > \alpha) \leq \beta \leq P(L_{\mathbf{x}} \leq \alpha).$$

Definindo de outra forma, o VaR é o menor valor de perda que faz com que a função de distribuição acumulada seja superior a  $\beta$ . Assim, o VaR pode ser definido como o  $\beta$ -quantil das perdas do portfólio  $\mathbf{x}$ , dado um nível de confiança  $\beta \in [0,1]$  (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$\alpha_{\beta}(\mathbf{x}) = \inf\{\eta : F_{\mathbf{x}}(\eta) \geq \beta\},$$

com  $F_{\mathbf{x}}(\eta) = P(R_{\mathbf{x}} \leq \eta)$  sendo a função de distribuição acumulada das perdas e  $\inf\{r : A\}$ , o menor limite de  $r$  dado um evento  $A$ .

### 3.1.1.4 Conditional Value-at-Risk

O CVaR (*Conditional Value-at-Risk*) é uma medida de risco proposta a partir da definição do VaR, mas que, diferente desta, considera a distribuição das perdas na cauda, incorporando a média ponderada das piores perdas, dado um nível de confiança  $\beta$ , em seu valor.

Considerando um vetor aleatório  $\mathbf{t}$ , que representa os fatores incertos do mercado (parâmetros do mercado), a probabilidade da perda de um portfólio  $\mathbf{x}$ , dados os fatores  $\mathbf{t}$  com densidade de probabilidade  $p(\mathbf{t})$ , não exceder um limite  $\alpha$  é dada por (ROCKAFELLAR; URYASEV, 2000):

$$P(L_{\mathbf{x}}(\mathbf{t}) \leq \alpha) = \int_{L_{\mathbf{x}}(\mathbf{t}) \leq \alpha} p(\mathbf{t}) d\mathbf{t}.$$

Assumindo que  $P(L_{\mathbf{x}}(\mathbf{t}) \leq \alpha)$  é uma função contínua, para efeito de simplificação, Rockafellar e Uryasev (2000) deduzem o CVaR da seguinte forma:

$$\phi_{\beta}(\mathbf{x}) = (1 - \beta)^{-1} \int_{L_{\mathbf{x}}(\mathbf{t}) \geq \alpha_{\beta}(\mathbf{x})} L_{\mathbf{x}}(\mathbf{t}) p(\mathbf{t}) d\mathbf{t}.$$

Na prática, o número de cenários a ser considerado é finito e, portanto, a forma discretizada do CVaR, para  $T$  cenários, é (ROCKAFELLAR; URYASEV, 2000):

$$\phi_{\beta}(\mathbf{x}) = (1 - \beta)^{-1} \sum_{t=1}^T [L_{\mathbf{x}}(t) - \alpha]^+ P(t),$$

onde  $[a]^+ = \max\{0, a\}$ . Rockafellar e Uryasev (2000) propõem uma nova definição de risco CVaR, que consiste na adição do VaR ( $\alpha$ ) ao valor CVaR ( $\phi$ ). No caso mais simples, em que todos os  $T$  cenários do portfólio  $\mathbf{x}$  possuem a mesma probabilidade de ocorrer ( $p = 1/T$ ), esse risco é definido como segue:

$$F_{\beta}(\mathbf{x}, \alpha) = \alpha + \frac{1}{T(1 - \beta)} \sum_{t=1}^T [L_{\mathbf{x}}(t) - \alpha]^+.$$

De acordo com Mansini, Ogryczak e Speranza (2015), este risco CVaR é considerado uma medida coerente por apresentar as propriedades de monotonia, subaditividade, homogeneidade para valores positivos, de translação equivariante e risco relevante. Mais informações sobre medidas coerentes de risco e suas propriedades podem ser encontradas em Mansini, Ogryczak e Speranza (2015).

### 3.1.2 Modelo média-variância

O modelo média-variância, desenvolvido a partir dos conceitos propostos por Markowitz (1952), responsável pela criação da Teoria Moderna do Portfólio. As métricas de desempenho de portfólio consideradas nesse modelo são a média histórica dos retornos, como estimativa para o retorno esperado e a variância do portfólio como medida de risco.

Originalmente, Markowitz (1952) considera as porcentagens  $x_j$  do capital investido em cada ativo  $j$  na caracterização do portfólio. Assim, a soma dessas proporções deve ser menor ou igual a 1, que representa todo o capital disponível para investimento, como descreve a restrição 3, considerando  $n$  ativos disponíveis. Em sua forma monobjetiva, como proposto originalmente, o modelo possui a variância como função objetivo a ser minimizada (função 1) e uma restrição de retorno mínimo, que garante um valor de retorno esperado acima de um valor determinado  $\mu_0$  (restrição 2). Por fim, a restrição 4 do modelo média-variância, apresentado a seguir, garante que as proporções de investimento não podem ser negativas.

$$\min_{\mathbf{x}} \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j \quad (1)$$

sujeito a:

$$\sum_{j=1}^n \mu_j x_j \geq \mu_0 \quad (2)$$

$$\sum_{j=1}^n x_j = 1 \quad (3)$$

$$x_j \geq 0, \forall j, j = 1, \dots, n \quad (4)$$

### 3.1.3 Modelo média-CVaR

O modelo média-variância serviu de base para o desenvolvimento de vários outros modelos de otimização de portfólios, tendo o modelo média-CVaR, proposto por [Rockafellar e Uryasev \(2000\)](#), como um exemplo. O modelo média-CVaR se difere do modelo média-variância por utilizar o risco CVaR como medida de risco, no lugar da variância, de forma que sua função objetivo é representada pela função 5.

$$\min_{\mathbf{x}} \alpha + \frac{1}{T(1-\beta)} \sum_{t=1}^T [L_{\mathbf{x}}(t) - \alpha]^+. \quad (5)$$

Pode-se observar que tanto o VaR ( $\alpha$ ) quanto a função condicional  $[\cdot]^+$  tornam essa medida de risco uma função não linear. Tendo em vista que modelos lineares podem ser solucionados mais facilmente, demandando menores custos computacionais, pode ser interessante utilizar a forma linear do CVaR, formada pelo seguinte modelo ([MANSINI; OGRYCZAK; SPERANZA, 2015](#)):

$$\min_{\eta, \mathbf{d}, \mathbf{x}} \eta + \frac{1}{T(1-\beta)} \sum_{t=1}^T d_t \quad (6)$$

sujeito a:

$$d_t \geq y_t - \eta, \forall t, t = 1, \dots, T \quad (7)$$

$$y_t = - \sum_{j=1}^n x_j r_{jt}, \forall t, t = 1, \dots, T \quad (8)$$

$$d_t \geq 0, \forall t, t = 1, \dots, T \quad (9)$$

A função 6 substitui o VaR ( $\alpha$ ) por uma variável auxiliar  $\eta$  sem restrições e a função não linear  $[L_{\mathbf{x}} - \alpha]^+$  pela variável auxiliar  $d$ ,  $y_t$  representa a própria função perda  $L_{\mathbf{x}}(t)$ , que corresponde ao oposto da média dos retornos  $r_{jt}$  de cada ativo  $j$  no cenário  $t$ , ponderada pela sua proporção  $x_j$ , como descreve a equação 8. Ao ser minimizado,  $\eta$  possui o valor do próprio VaR  $\alpha$ . As restrições 7 e 9 garantem que o valor mínimo de  $d_t$  será a perda do

portfólio em  $t$  menos o VaR, caso esse valor seja maior que zero, e zero, caso contrário, exatamente como a função não linear original (função 5) descreve. Ao adicionar as restrições 2, 3 e 4, esse se torna um modelo linear média-CVaR.

### 3.1.4 Restrição de cardinalidade

Restringir a quantidade de ativos possíveis em um portfólio, ou sua cardinalidade, é uma prática que traz algumas vantagens. Um limite mínimo de ativos garante que apenas portfólios diversificados serão incluídos como soluções do processo de otimização, enquanto que limites máximos reduzem os custos de monitorar e rebalancear portfólios após o processo de otimização (MANSINI; OGRYCZAK; SPERANZA, 2015).

No modelo, a restrição de cardinalidade consiste em garantir que a quantidade de ativos selecionados para o portfólio, ou seja, com proporções de investimento maiores que zero, esteja dentro de um intervalo caracterizado pela cardinalidade mínima  $k_{min}$  e pela cardinalidade máxima  $K_{max}$ . Mansini, Ogryczak e Speranza (2015) aponta a necessidade de se utilizar um conjunto de  $n$  variáveis auxiliares  $z_j$ , indicando a presença do ativo  $j$  no portfólio ótimo. Assim, as seguintes restrições devem ser adicionadas ao modelo de otimização de portfólio, sintetizando a restrição de cardinalidade:

$$\begin{cases} z_j = 1, \text{ se } x_j > 0 \\ z_j = 0, \text{ caso contrário} \end{cases} \quad \forall j, j = 1, \dots, n \quad (10)$$

$$k_{min} \leq \sum_{j=1}^n z_j \leq k_{max} \quad (11)$$

Observa-se que  $z_j$ , para qualquer ativo  $j$ , assume valores contidos em um conjunto binário  $\{0,1\}$  e nota-se que a restrição 10 não é linear. Assim, a presença da restrição de cardinalidade em um modelo, o torna um modelo não linear de otimização inteira mista. A restrição de cardinalidade fixa representa um caso particular dessas restrições, em que  $k_{min} = k_{max}$  na expressão 11.

### 3.1.5 Custos de transação e rebalanceamento

Custos de transação são valores cobrados por instituições financeiras ou corretoras para intermediar negociações de ações no Mercado Financeiro. A consideração desses custos em modelos de otimização de portfólio é importante, já que eles reduzem o retorno líquido do investidor e seu capital disponível para futuras negociações (MANSINI; OGRYCZAK; SPERANZA, 2015).

Para inserir esse aspecto no modelo de otimização, é importante definir variáveis  $X_j$ ,  $j = 1, \dots, n$ , que representam a quantidade de capital investida no ativo  $j$ , de forma

que  $X_j = x_j C$ , onde  $C$  é o capital total disponível para investimento e  $K_j(X_j)$  é o custo de transação associado a essa quantia investida em  $j$ . Existem diferentes estruturas de custos de transação e, assim, diferentes formas de expressá-los. O custo de transação proporcional, mais utilizado na prática de acordo com [Mansini, Ogryczak e Speranza \(2015\)](#), para um ativo  $j$ , é composto pelo valor total investido em  $j$ ,  $X_j$ , multiplicado por um fator  $\gamma_j$ , definido para cada ativo, ou seja:

$$K_j(X_j) = X_j \gamma_j.$$

Considerando os custos de transação, a restrição de retorno esperado (restrição 2) é alterada para:

$$\sum_{j=1}^n \mu_j X_j - \sum_{j=1}^n K_j(X_j) \geq \mu_0 C.$$

Considerando os custos de transação, outro problema surge quando deseja-se efetuar o processo de otimização de portfólios seguidas vezes, ou quando o investidor já conta com um portfólio prévio. Os motivos para desejar mudar os investimentos são vários, como a disponibilidade de mais capital para investimento, a necessidade de parte do capital investido ou o fato do portfólio investido sair da fronteira eficiente com o passar do tempo.

Nessa situação, a forma com que o custo de transação é calculado deve considerar esse investimento prévio, pois o custo de obter três lotes de determinado ativo é menor quando já se tem um ou dois lotes do que quando não se tem nenhum lote desse ativo. Esse problema é denominado problema de rebalanceamento de portfólio ([MANSINI; OGRYCZAK; SPERANZA, 2015](#)).

O modelo com rebalanceamento, então, deve incluir variáveis  $X_j^0$ , representando a quantia investida no ativo  $j$  nesse portfólio prévio. Quando  $X_j > X_j^0$ , ações do ativo  $j$  são compradas, vendas são realizadas quando  $X_j < X_j^0$  e a quantidade do ativo se mantém quando  $X_j = X_j^0$ . A restrição de retorno esperado, então, será:

$$\sum_{j=1}^n \mu_j X_j - \sum_{j=1}^n K_j(\delta_j) \geq \mu_0 C,$$

sendo que  $\delta_j$  representa o módulo da diferença entre  $X_j$  e  $X_j^0$ . De forma linear,  $\delta_j \geq (X_j - X_j^0)$  e  $\delta_j \geq -(X_j - X_j^0)$ .

Além do custo proporcional apresentado, [Mansini, Ogryczak e Speranza \(2015\)](#) apresentam outras estruturas de custos de transação, menos utilizadas na prática, com suas respectivas formas de rebalanceamento e suas restrições em um modelo de otimização de portfólios.

### 3.1.6 Modelos de Otimização de Portfólio com variáveis de decisão inteiras

Na Bolsa de Valores, os ativos são vendidos em lotes de ações, que representam a quantidade mínima de ações que podem ser negociadas e determinam que a quantidade de ações de um ativo deve ser um número múltiplo desse lote de ações para esse ativo. Apesar desse aspecto prático, a maioria dos modelos propostos na literatura utilizam variáveis de decisão reais, representando as proporções de investimento para cada ativo. Dado que apenas valores múltiplos dos lotes mínimos  $m_i$  de negociação de cada ativo  $i$ , de um portfólio, podem ser investidos, alguns trabalhos propõem modelos em que quantidades de lotes mínimos investidos em cada ativo de um portfólio são as variáveis de decisão, representadas por valores inteiros.

Um modelo biobjetivo média-CVaR é proposto por [Ferreira \(2018\)](#), onde um portfólio  $\mathbf{x} = x_1, x_2, \dots, x_n$  é definido pelas variáveis inteiras  $x_i$ , que representa as quantidades de lotes mínimos investidos em cada ativo  $i = 1, \dots, n$ , considerando  $n$  ativos disponíveis. Este modelo também inclui uma restrição de cardinalidade fixa e rebalanceamento dos custos de transação.

A primeira função objetivo é a medida de risco CVaR, a ser minimizada, conforme mostra a expressão 12, onde  $\zeta$  representa o valor VaR,  $\alpha$  é o nível de significância considerado,  $p_t$  é a probabilidade do cenário ou período  $t$  ocorrer e  $r_{xt}$  o log retorno do portfólio  $\mathbf{x}$  no cenário  $t$ . A segunda função, descrita pela expressão 12, é o retorno esperado, definido pela proporção  $w_i$  de investimento de cada ativo  $i$  e por sua média de log retornos histórica  $\mu_i$ .

O modelo de [Ferreira \(2018\)](#) considera a restrição de cardinalidade fixa e, para isso, são utilizadas variáveis auxiliares binárias  $z_i$  que indicam a presença de cada ativo  $i$  no portfólio. Nessa restrição,  $k$  representa o valor de cardinalidade do portfólio.

A restrição de capital máximo de investimento incorpora o rebalanceamento, considerando um portfólio prévio  $\mathbf{x}^{(0)} = x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ . A restrição de capital também considera valores fixos e proporcionais de custos de transação, onde  $f$  representa um custo fixo, aplicado quando a solução do modelo é diferente do portfólio prévio, e  $\beta$  representa o custo proporcional ao valor negociado para a obtenção dos novos valores de investimento em cada ativo. Para isso, utiliza-se a variável binária auxiliar  $v$ , que indica se o novo portfólio  $\mathbf{x}$  difere do portfólio prévio  $\mathbf{x}^{(0)}$ . Essa restrição ainda considera o capital total  $c$  disponível para investimento e o custo  $c_i$  de cada ativo  $i$ , dado pelo preço de fechamento de  $i$  no período  $T$ , último período considerado na série histórica.

Assim, [Ferreira \(2018\)](#) propõe o seguinte modelo multiobjetivo de otimização inteira

não linear:

$$\min_{x_1, \dots, x_n} \zeta + (1 - \alpha)^{-1} \sum_{t=1}^T p_t [-r_{\mathbf{x}t} - \zeta]^+ \quad (12)$$

$$\max_{x_1, \dots, x_n} \sum_{i=1}^n w_i \mu_i \quad (13)$$

$$\text{sujeito a: } \left\{ \begin{array}{l} w_i = \frac{m_i c_i x_i}{\sum_{j=1}^n m_j c_j x_j}, \forall i, i = 1, \dots, n \quad (14a) \\ \sum_{i=1}^n z_i = k \quad (14b) \\ \sum_{i=1}^n m_i c_i [(x_i - x_i^{(0)}) + \beta |x_i - x_i^{(0)}|] + f v \leq C \quad (14c) \\ z_i = \begin{cases} 0, & \text{se } x_i = 0 \\ 1, & \text{caso contrário} \end{cases}, \forall i, i = 1, \dots, n \quad (14d) \\ v = \begin{cases} 0, & \text{se } \sum_{i=1}^n |x_i - x_i^{(0)}| = 0 \\ 1, & \text{caso contrário} \end{cases} \quad (14e) \\ x_i \in \mathbb{N}, \forall i, i = 1, \dots, n \quad (14f) \end{array} \right.$$

onde as proporções de investimento de cada ativo são definidas pela equação 14a, as equações 14b e 14d compõem a restrição de cardinalidade fixa, as expressões 14c e 14e compõem a restrição de capital máximo com rebalanceamento e a expressão 14f indica que as quantidades de lotes dos ativos deve ser um valor inteiro.

## 3.2 Otimização Linear Inteira

Como visto na seção anterior, a presença da restrição de cardinalidade em um modelo de otimização de portfólio, o torna um modelo de otimização inteira ou inteira mista. Por isso, essa seção trata de métodos de programação para essa classe de problemas.

De acordo com Schrijver (1986), problemas lineares inteiros são problemas NP-completos, ou seja, problemas em que não são conhecidas soluções que podem ser executadas em um tempo polinomial em relação ao tamanho da variável de entrada. Dessa forma, encontrar sua solução ótima, ou a melhor solução possível para o problema, para instâncias de grandes dimensões desses problemas, pode ser uma tarefa inviável, na prática. Uma alternativa para a solução desses problemas é a utilização de heurísticas, que não garante a obtenção do ótimo global, mas pode proporcionar boas soluções factíveis.

Relaxações são aplicadas aos problemas lineares inteiros, com a intenção de obter soluções aproximadas e informações sobre o problema original. A relaxação linear consiste em retirar as restrições de integralidade do problema. Considerando um problema de minimização, com um vetor de custos  $c$ , definindo os coeficientes das variáveis de decisão  $x$  (de dimensão  $n$ ) da função objetivo, uma matriz  $A$  e um vetor  $b$ , representando os coeficientes e os termos independentes das restrições do problema, respectivamente, a relaxação linear de  $\min\{cx \mid Ax \leq b, x \in \mathbb{N}^n\}$  resulta no problema de programação linear  $\min\{cx \mid Ax \leq b, x \geq 0, x \in \mathbb{R}^n\}$  (SCHRIJVER, 1986).

A relaxação Lagrangeana também pode ser aplicada em problemas de programação linear inteira e consiste em remover algumas restrições do problema original e adicioná-las à função objetivo, em forma de penalidade, com pesos para cada variável de decisão, denominados multiplicadores de Lagrange (SCHRIJVER, 1986).

### 3.2.1 *Branch-and-cut*

O algoritmo *Branch-and-cut* é um método de programação linear inteira e inteira mista, que pode proporcionar ótimos globais para essa classe de problemas. Para isso, são realizados, de forma iterativa, cortes no espaço de busca, excluindo soluções reais para variáveis inteiras e a solução do problema, pelo algoritmo Simplex, após a aplicação de relaxações lineares em restrições inteiras.

Mitchell (1999) descreve o método, para um problema de minimização, por meio do Algoritmo 1. Nesse algoritmo,  $L$  é um conjunto de problemas lineares inteiros,  $\bar{z}$  é o limite superior do valor de função objetivo,  $z_l$  e  $x_l$  são, respectivamente, o valor dessa função para um problema  $l \in L$  e a sua solução. Seu funcionamento básico consiste em solucionar um conjunto de problemas lineares reais, ramificados a partir do problema inteiro original (linha 31 do algoritmo) e podar os problemas cujas relaxações lineares resultem em soluções piores do que alguma solução inteira obtida em outra partição (linha 29).

## 3.3 Algoritmos de Classificação

Problemas de classificação são problemas de aprendizagem supervisionada, em que existem dados históricos ou dados de treinamento cujos vetores de entrada, formados pelos atributos ou características relevantes ao problema, possuem vetores de saída ou vetores alvos correspondentes. Dentre esses problemas supervisionados, os de classificação são aqueles cujo objetivo é atribuir, a cada elemento do vetor de entrada, uma etiqueta ou categoria dentre um conjunto discreto finito (BISHOP, 2006). Esta seção apresentará alguns algoritmos para solucionar esse tipo de problema, em casos específicos de duas classes ou categorias.

**Algoritmo 1: Branch-and-cut** genérico

---

**Input:** o problema linear inteiro original  $PLI^0$   
**Output:** solução ótima do problema  $x^*$ , valor da função objetivo para o ótimo  $z^*$

```

1  $L \leftarrow \{PLI^0\}$ 
2  $\bar{z} \leftarrow \infty$ 
3  $z_l \leftarrow -\infty$ 
4 while  $L \neq \emptyset$  do
5     seleciona e remove um problema  $PLI^l$  de  $L$ 
6     solucionado  $\leftarrow 0$ 
7     while solucionado = 0 do
8         resolva a relaxação linear do problema  $PLI^l$ 
9         if relaxação de  $PLI^l$  é infactível then
10              $z_l \leftarrow \infty$ 
11             solucionado  $\leftarrow 1$ 
12         else
13              $x_l \leftarrow$  solução da relaxação de  $PLI^l$ 
14              $z_l \leftarrow$  valor da função objetivo de  $x_l$ 
15             if  $z_l = \infty$  then
16                  $z_l \leftarrow -\infty$ 
17             end
18             procure por planos de corte que são violados por  $x_l$ 
19             if algum plano de corte foi encontrado then
20                 adicione os planos encontrados à relaxação de  $PLI^l$ 
21             else
22                 if  $z_l \geq \bar{z}$  then
23                     solucionado  $\leftarrow 1$ 
24                 else
25                     if  $x_l$  é uma solução inteira then
26                          $\bar{z} \leftarrow z_l$ 
27                          $z^* \leftarrow z_l$ 
28                          $x^* \leftarrow x_l$ 
29                         remova de  $L$  todos os problemas  $l$  em que  $z_l > \bar{z}$ 
30                     else
31                         particione o problema em novos problemas com
32                         diferentes regiões de factibilidade e adicione-os a  $L$ 
33                     end
34                 solucionado  $\leftarrow 1$ 
35             end
36         end
37     end
38 end

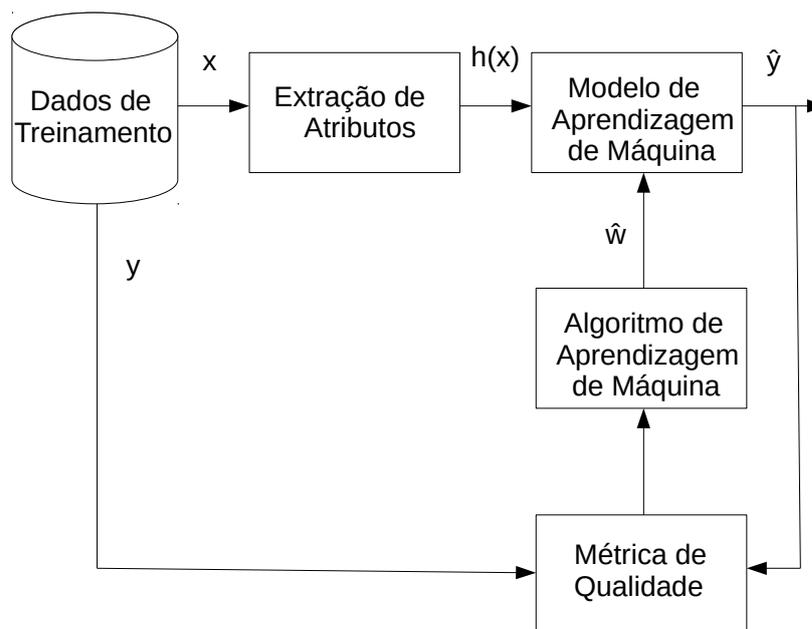
```

---

### 3.3.1 Regressão Logística

A Regressão Logística é um classificador linear, e como tal, utiliza dados de treinamento para aprender ou ajustar pesos a cada atributo do vetor de entrada. Representando o vetor de entrada por  $\mathbf{x}$ , os atributos são formados por transformações  $\mathbf{h}(\cdot)$  aplicadas em  $\mathbf{x}$ ,  $\mathbf{y}$  é o vetor de categorias associadas a cada elemento de  $\mathbf{x}$ ,  $\hat{\mathbf{w}}$  corresponde aos pesos associados a cada atributo e  $\hat{y}$  é o vetor de classes preditas para cada entrada. A Figura 7 apresenta um fluxograma para um classificador linear genérico. Pode-se observar que o modelo de Aprendizagem de Máquina produz previsões com base nos atributos e seus pesos. O algoritmo de classificação utiliza uma métrica de qualidade para avaliar as categorias preditas em relação às categorias reais de cada instância de entrada e, conforme o erro das previsões, novos pesos  $\hat{\mathbf{w}}$  são gerados. Esse ciclo se mantém até que alguma condição de parada seja satisfeita.

Figura 7 – Classificador linear genérico



No caso de um problema de classificação de duas classes, cada elemento do vetor alvo  $y$  pode assumir apenas dois valores possíveis, geralmente 0 ou 1. O vetor  $\mathbf{x} = x_1, x_2, \dots, x_n$  é um vetor de  $n$  instâncias, sendo que cada instância  $\mathbf{x}_i = x_{i1}, x_{i2}, \dots, x_{id}$  possui  $d$  entradas. O vetor de transformações  $\mathbf{h}(x) = h_1(x), h_2(x), \dots, h_d(x)$  possui uma transformação específica para cada entrada e a transforma em um atributo. Pode-se definir, então, uma função *score* da seguinte forma (BISHOP, 2006):

$$\text{score}(\mathbf{x}_i) = w_0 + w_1 h_1(x_{i1}) + \dots + w_d h_d(x_{id}) = \hat{\mathbf{w}}^T \mathbf{h}(\mathbf{x}_i).$$

A predição para uma instância  $i$ , portanto, é dada por:

$$\hat{y}_i = \text{sinál}(\text{score}(\mathbf{h}(\mathbf{x}_i))),$$

onde *sinál* é uma função que retorna um valor 0 ou 1, baseado no valor da função *score*.

A Regressão Logística utiliza a probabilidade de cada instância pertencer a uma determinada classe na predição de sua classe.  $P(1|\mathbf{x}_i)$  representa a probabilidade de a instância  $i$  pertencer à classe 1, dado que ela possui as entradas  $x_{i1}, x_{i2}, \dots, x_{id}$ . Essa probabilidade é dada pela função *sinál* sigmoide aplicada a  $\text{score}(\mathbf{h}(\mathbf{x}_i))$ :

$$P(1|\mathbf{x}_i) = \frac{1}{1 + e^{-\text{score}(\mathbf{h}(\mathbf{x}_i))}},$$

que também pode ser escrita como (BISHOP, 2006):

$$P(1|\mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{h}(\mathbf{x}_i)}}.$$

A probabilidade dessa instância pertencer à classe 0 é dada por  $P(0|\mathbf{x}_i, \mathbf{w}) = 1 - P(1|\mathbf{x}_i, \mathbf{w})$ . Utilizando a métrica de qualidade verossimilhança, deseja-se encontrar  $\mathbf{w} = \hat{\mathbf{w}}$ , que maximiza a função de verossimilhança  $l(\mathbf{w})$  (BISHOP, 2006):

$$l(\mathbf{w}) = \prod_{i=1}^n P(y_i|\mathbf{x}_i, \mathbf{w}).$$

Encontrar o valor  $\mathbf{w}$  que maximiza a função de verossimilhança corresponde a encontrar o valor  $\mathbf{w}$  que minimiza a função de entropia cruzada  $E(\mathbf{w})$ , sendo que (BISHOP, 2006):

$$E(\mathbf{w}) = -\ln l(\mathbf{w}) = -\ln \left\{ \prod_{i=1}^n P(y_i|\mathbf{x}_i, \mathbf{w}) \right\}. \quad (15)$$

Segundo Bishop (2006), a minimização da entropia cruzada é feita por algoritmos de otimização iterativos, como o gradiente descendente ou métodos quasi-Newton, que conseguem alcançar esse mínimo global por meio do gradiente da entropia cruzada  $\nabla E(\mathbf{w})$ , já que sua função é côncava e possui um único mínimo. Esse gradiente é dado por (BISHOP, 2006):

$$\nabla E(\mathbf{w}) = \sum_{i=1}^n (P(1|\mathbf{x}_i, \mathbf{w}) - y_i) \mathbf{h}(\mathbf{x}_i).$$

Após a obtenção de  $\hat{\mathbf{w}}$ , pode-se aplicá-lo em instâncias não conhecidas para obter sua probabilidade de pertencimento em cada classe. Normalmente, a classe de uma instância é predita como sendo aquela de maior probabilidade de pertencimento.

Regularizações são procedimentos aplicados em algoritmos de Aprendizagem de Máquina com a intenção de evitar que os pesos aprendidos fiquem excessivamente ajustados aos dados de treinamento e, dessa forma, percam a capacidade de generalização. Consistem em adicionar uma penalidade à função erro a ser minimizada (BISHOP, 2006).

### 3.3.1.1 Regularização L1

A regularização L1, ou LASSO, adiciona uma penalidade aplicada aos valores absolutos dos pesos  $w$  ao erro entropia cruzada, de forma que a minimização dessa nova função custo  $J$  opte por menores valores desses pesos. Apesar de a aplicação do valor absoluto ou norma L1 não ser uma função diferenciável, pode-se utilizar métodos de otimização iterativos de ajuste de pesos, considerando o parâmetro de regularização  $\lambda$  como restrição de limite para cada peso, sendo que os pesos que violarem essa restrição, terão seus valores alterados para 0. Assim, essa regularização também funciona como um método de seleção de variáveis, sendo que valores maiores de  $\lambda$  removem uma maior quantidade de atributos (BISHOP, 2006). A nova função de custo se torna:

$$\min_{\mathbf{w}, \lambda} \{J(\mathbf{w}, \lambda)\} = \min_{\mathbf{w}, \lambda} \{E(\mathbf{w}) + \lambda \|\mathbf{w}\|\}.$$

### 3.3.1.2 Regularização L2

Utilizando a regularização L2, ou Ridge, a norma L2 do vetor de pesos  $\|\hat{\mathbf{w}}\|^2$ , multiplicada por um parâmetro de regularização  $\lambda$ , que define a importância a ser dada para a regularização em relação ao erro de treinamento (BISHOP, 2006). Adicionando esse termo à função de erro a ser minimizada, cria-se uma nova função custo  $J$  a ser minimizada:

$$\min_{\mathbf{w}, \lambda} \{J(\mathbf{w}, \lambda)\} = \min_{\mathbf{w}, \lambda} \{E(\mathbf{w}) + \lambda \|\mathbf{w}\|^2\}.$$

E o gradiente dessa função, a ser utilizado em algoritmos de otimização, é:

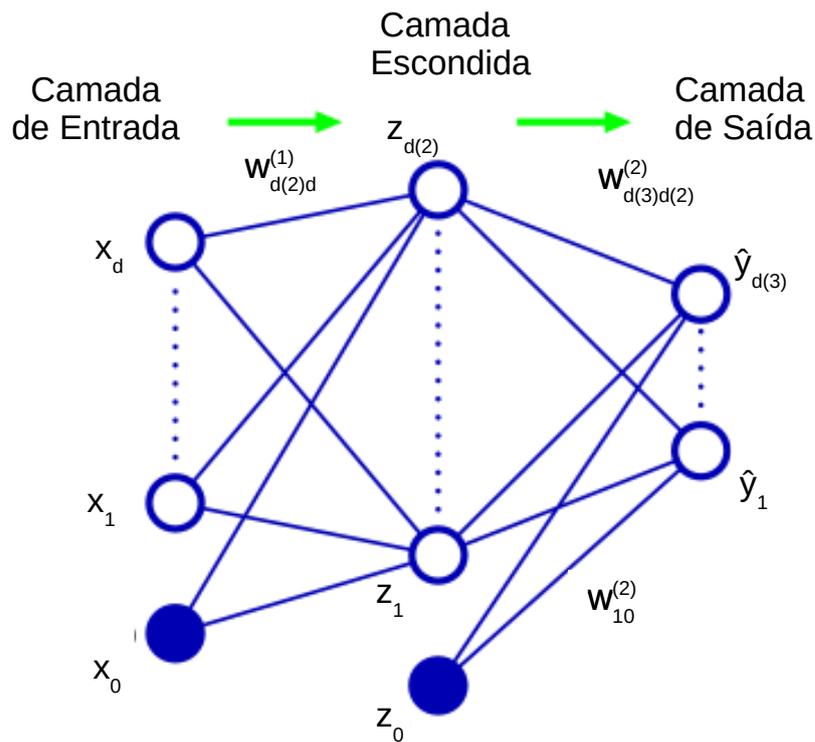
$$\nabla J(w, \lambda) = \sum_{i=1}^n (P(1|\mathbf{x}_i, \mathbf{w}) - y_i) \mathbf{h}(\mathbf{x}_i) - 2\lambda \mathbf{w}.$$

## 3.3.2 Perceptron Multicamadas

O Perceptron Multicamadas ou MLP (do inglês, *Multilayer Perceptron*) é uma Rede Neural Artificial, modelo computacional proposto em McCulloch e Pitts (1943), inspirado no Sistema Nervoso Central e que faz analogia ao processo de aprendizagem humano. Compostos por sistemas básicos denominados neurônios, o MLP possui uma estrutura multicamadas ilustrada pela Figura 8, que mostra um Perceptron Multicamadas de uma camada escondida genérico. A camada de entrada é composta por  $d$  neurônios, a camada

escondida, por  $d(1)$  e a de saída,  $d(2)$ . As saídas da camada escondida são representadas por valores  $z$  e  $\mathbf{w}$  são os pesos associados a saída de uma camada, para compor a entrada da próxima camada. Por fim, as setas indicam o sentido de movimento da informação na produção da saída  $\hat{y}$ .

Figura 8 – MLP com uma camada oculta



Fonte: Adaptado de Bishop (2006)

Diferente da Regressão Logística, o Perceptron Multicamadas não se limita a definir combinações lineares de funções de bases fixas. Esse algoritmo considera um número limitado de funções de base e as adaptam aos dados de treinamento por meio de seus parâmetros adaptativos (BISHOP, 2006). Como visto na subseção 3.3.1, um classificador linear possui um modelo genérico:

$$\hat{y}_i = \text{sin}(\mathbf{w}^T \mathbf{h}(\mathbf{x}_i)), \tag{16}$$

com funções de base  $\mathbf{h}$  fixas. O MLP associa  $\mathbf{h}$  a valores de parâmetros, que podem ser ajustados no treinamento, assim como os pesos  $\mathbf{w}$ . Cada função de base é formada por combinações lineares das entradas e seus coeficientes são parâmetros adaptativos.

Para descrever um modelo básico de MLP cria-se  $d^{(2)}$  combinações lineares das entradas  $\mathbf{x}$ , para a primeira camada da rede, sendo  $d^{(2)}$  a quantidade de neurônios da segunda camada (BISHOP, 2006):

$$a_j^{(1)} = \sum_{i=1}^{d^{(2)}} \mathbf{w}_{ji}^{(1)} \mathbf{x}_i + \mathbf{w}_{j0}^{(1)},$$

para  $j = 1, \dots, n$ ,  $a_j$  chamado de função de ativação e o (1) sobrescrito representando a primeira camada da rede. Assim como a equação 16, essas funções de base são formadas por transformações dessas combinações  $a$ , de forma que cada função de base  $z_j^{(2)} = t(a_j^{(1)})$ , onde  $t(\cdot)$  é uma função de ativação não linear. Para classificação, essa função de ativação é, geralmente, a sigmoide ou a tangente hiperbólica (BISHOP, 2006).

Considerando que a segunda camada da rede seja uma camada oculta, com  $d^{(2)}$  neurônios,  $d^{(3)}$  novas combinações são criadas, sendo  $d^{(3)}$  a quantidade de neurônios da terceira camada:

$$a_j^{(2)} = \sum_{i=1}^{d^{(3)}} \mathbf{w}_{ji}^{(2)} z_i + \mathbf{w}_{j0}^{(2)}.$$

As transformações são novamente aplicadas, criando  $d^{(3)}$  saídas  $z^{(3)}$  e o procedimento se repete, para cada camada oculta a mais que a rede possuir. Considerando, porém, uma rede com apenas uma camada oculta, como mostra a Figura 8, a terceira camada é a chamada camada de saída, nesse caso com  $d^{(3)}$  neurônios. Para a camada de saída, a função de ativação utilizada para problemas de classificação é a sigmoide logística. Assim, saída  $\hat{y}$  de cada neurônio dessa camada é dada por (BISHOP, 2006):

$$\hat{y}(\mathbf{x}_j) = \text{sigmoide}(a_j^{(2)}).$$

De forma genérica, para um MLP com  $m$  camadas ocultas, a saída de cada neurônio da camada de saída é dado pelo procedimento de alimentação direta (*feedforward*), descrito pela equação:

$$\hat{y}(\mathbf{x}_j) = \text{sigmoide} \left( \sum_{i_1=1}^{d^{(m+2)}} \mathbf{w}_{ji_1}^{(m+1)} t \left( \sum_{i_2=1}^{d^{(m+1)}} \mathbf{w}_{i_1 i_2}^{(m)} t \left( \dots t \left( \sum_{i_m=1}^{d^{(2)}} \mathbf{w}_{i_{m-1} i_m}^{(1)} \mathbf{x}_{i_m} \right) \right) \right) \right).$$

A função erro, a ser minimizada, é dada pela entropia cruzada (equação 15), sendo que, para um problema de duas classes,  $P(1|\mathbf{x}_j, \mathbf{w}) = \hat{y}(\mathbf{x}_j)$  e  $P(0|\mathbf{x}_j, \mathbf{w}) = 1 - P(1|\mathbf{x}_j, \mathbf{w})$ .

### 3.3.3 Máquina de Vetores de Suporte

Máquina de Vetores de Suporte ou SVM (do inglês, *Support Vector Machine*) é um método baseado em núcleos (*kernels*) não lineares, em que a determinação dos parâmetros

de seu modelo é um problema de otimização convexo, de forma que encontrar um ótimo local corresponde a encontrar o ótimo global (BISHOP, 2006).

Para descrever o algoritmo SVM, primeiro será lembrado o modelo básico de um classificador binário linear, em que os vetores alvos para cada instância ou amostra de entrada é classificado como 1 ou  $-1$ :

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T \mathbf{h}(\mathbf{x}) + b, \quad (17)$$

onde  $b$  é o termo independente (*bias*) explícito.

Assume-se, nesse caso, que existe um hiperplano que separa as duas classes de amostras. Mas pode existir vários hiperplanos que separam essas amostras de acordo com sua classe. Uma ideia intuitiva para o problema é encontrar o hiperplano que melhor separa as duas classes de amostras, ou seja, com maior distância entre as amostras de cada classe. Para isso, é definido o conceito de margem, que, segundo Bishop (2006), é a distância entre esse hiperplano que divide as classes, também chamado de limite de decisão, e a amostra mais próxima desse hiperplano. Assim, deseja-se encontrar o limite de decisão que maximize essa margem.

A Figura 9 apresenta o hiperplano que maximiza a distância entre a amostra mais próxima representada pela reta azul. As amostras mais próximas desse hiperplanos, marcadas por circunferências vermelhas, são denominadas vetores de suporte. Os círculos verdes representam amostras de classe  $-1$  e os quadrados pretos, amostras de classe 1. Assim a linha pontilhada que passa pelos vetores de suporte dos círculos representa a reta  $\hat{y}(\mathbf{x}) = -1$ , a linha pontilhada que passa pelos vetores dos quadrados representa a reta  $\hat{y}(\mathbf{x}) = 1$  e o hiperplano ótimo, a reta  $\hat{y}(\mathbf{x}) = 0$ . A margem é a distância  $\rho$ , perpendicular ao hiperplano ótimo, entre o hiperplano e um vetor de suporte.

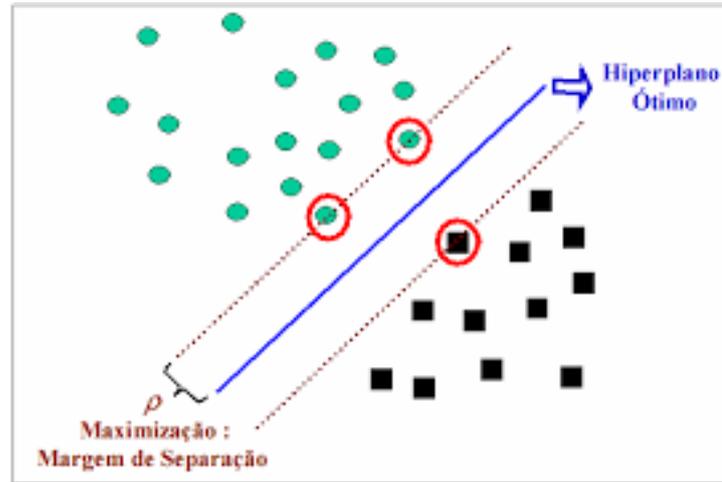
A distância perpendicular entre uma amostra  $\mathbf{x}$  e a reta  $\hat{y}(\mathbf{x}) = 0$  é dada por  $|\hat{y}(\mathbf{x})|/||\mathbf{w}||$ . Se todas as soluções são corretamente classificadas, a predição e a classe verdadeira de uma amostra possuem o mesmo sinal, ou seja,  $\hat{y}(\mathbf{x})y_i > 0$ . Então a distância perpendicular entre o hiperplano que separa as duas classes e uma amostra  $i$  é dada por (BISHOP, 2006):

$$\frac{\hat{y}(\mathbf{x}_i)y_i}{||\mathbf{w}||} = \frac{\hat{y}(\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b)}{||\mathbf{w}||}.$$

A distância perpendicular  $\rho$  entre o hiperplano e a amostra mais próxima:

$$\rho = \min_i [\hat{y}(\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b)]$$

Figura 9 – Hiperplano ótimo em problemas de classificação binários



Fonte: Andreola e Haertel (2010)

e a margem é essa distância  $\rho$  considerando o hiperplano que a torne um valor máximo (BISHOP, 2006):

$$\rho^* = \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_i [\hat{y}(\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b)] \right\}.$$

Dessa forma, surge um problema de otimização em que se deseja encontrar os pesos  $\mathbf{w}$  e o *bias*  $b$  que maximizem a distância  $\rho$ . Partindo das observações de que, na linha paralela ao hiperplano ótimo, que passa pelos vetores de suporte, o valor predito é igual ao valor real de sua classe, ou seja,  $y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$  e para todos os pontos,  $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ , já que não existem amostras entre as retas  $\hat{y}(\mathbf{x}) = -1$  e  $\hat{y}(\mathbf{x}) = 1$ . Adicionalmente, deseja-se maximizar  $\|\mathbf{w}\|^{-1}$ , sendo equivalente a minimizar  $\|\mathbf{w}\|^2$ . Assim, é proposto o seguinte modelo de otimização linear quadrática para esse problema (BISHOP, 2006):

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

sujeito a:

$$y_i(\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b) \geq 1, \quad i = 1, \dots, n.$$

Segundo Bishop (2006), é utilizado o método dos multiplicadores de Lagrange para a solução desse problema. Esses multiplicadores são denotados por  $a_i \geq 0$ , para cada restrição  $i$  do problema. Assim, a função Lagrangiana  $L(\mathbf{w}, b, \mathbf{a})$  é dada por:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n a_i \{y_i (\mathbf{w}^T \mathbf{h}(\mathbf{x}_i) + b) - 1\}.$$

Derivando a função Lagrangiana em relação a  $\mathbf{w}$  e  $b$ , são formadas as duas respectivas condições:

$$\text{textbfw} = \sum_{i=1}^n a_i y_i \mathbf{h}(\mathbf{x}_i) \quad (18)$$

$$0 = \sum_{i=1}^n a_i y_i.$$

Substituindo essas condições em  $L(\mathbf{w}, b, \mathbf{a})$  e, removendo assim  $\mathbf{w}$  e  $b$ , obtém-se a representação dual do problema, que consiste na maximização da margem (BISHOP, 2006):

$$\tilde{L}(\mathbf{a}) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

sujeito a:

$$a_i \geq 0, \quad i = 1, \dots, n$$

$$\sum_{i=1}^n a_i y_i = 0,$$

onde  $k(\mathbf{x}, \mathbf{x}') = \mathbf{h}(\mathbf{x})^T \mathbf{h}(\mathbf{x}')$  é denominada função núcleo (*kernel*).

Solucionando o problema de otimização descrito, então, encontra-se os multiplicadores de Lagrange que maximiza a margem. Para classificar uma nova amostra  $\mathbf{x}_j$  utilizando o modelo treinado, basta utilizar a equação 17, substituindo  $\mathbf{w}$  pela condição descrita pela equação 18:

$$\hat{y}(\mathbf{x}_j) = \sum_{i=1}^n a_i y_i k(\mathbf{x}_j, \mathbf{x}_i) + b.$$

O termo independente  $b$  pode ser deduzido pelos vetores de suporte, de forma que, sendo  $S$  o conjunto de todos os vetores de suporte (BISHOP, 2006):

$$b = \frac{1}{\|S\|} \sum_{i \in S} \left( y_i - \sum_{j \in S} a_j y_j k(\mathbf{x}_i, \mathbf{x}_j) \right).$$

Nota-se que a solução apresentada para o problema é válida apenas para problemas em que as amostras são linearmente separáveis. Porém, na prática, dificilmente os

dados de treinamento constituem amostras linearmente separáveis e, para solucionar tal problema, utiliza-se funções núcleo  $k$  não lineares (BISHOP, 2006). Essas funções aplicam transformações, por meio de produtos internos entre as amostras, que resultem em um conjunto linearmente separáveis. Alguns exemplos dessas funções núcleo são o polinomial, Gaussiana, sigmoide e função de base radial.

A função núcleo polinomial é definida como:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p,$$

onde  $p$  indica o grau do polinômio. A função núcleo de base radial ou RBF (*Radial Basis Function*) é dada por:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma(\mathbf{x}_i - \mathbf{x}_j)^2},$$

onde  $\gamma$  é um parâmetro de escala de distância entre as amostras. Quanto maior o valor de  $\gamma$ , mais ajustado será o hiperplano de separação às amostras de treino. A função núcleo Gaussiana é um caso específico de núcleo RBF, descrito como:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2\sigma^2}(\mathbf{x}_i - \mathbf{x}_j)^2},$$

onde  $\sigma^2$  representa a variância da distribuição das amostras de treinamento. Por fim, a função núcleo sigmoide é:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\eta \mathbf{x}_i \cdot \mathbf{x}_j + \nu),$$

onde  $\tanh(\cdot)$  é a função tangente hiperbólica,  $\eta$  é um parâmetro de escala, como  $\gamma$  da função RBF, e  $\nu$  é um termo independente.

### 3.3.4 Rede Neural Convolutiva

Rede Neural Convolutiva (LECUN et al., 1989; LECUN et al., 1998) é um algoritmo de Aprendizagem de Máquina que possui modelos de classificação invariantes a algumas transformações das entradas e, por isso, necessita de um menor esforço em pré-processamento dos dados de treinamento. Segundo Bishop (2006), esse algoritmo é muito utilizado em classificações de imagens, que normalmente é invariante às transformações de translação e rotação. O funcionamento do algoritmo também considera que regiões mais próximas em uma imagem normalmente são mais fortemente correlacionadas. Séries temporais, que também podem ser analisadas de forma gráfica, também é uma classe de problemas que pode ser lidada por esse algoritmo.

Redes Neurais Convolucionais combinam atributos de entrada em camadas de convolução e de subamostragem para extrair atributos de ordens maiores. Em uma camada de convolução, as unidades são organizadas em planos ou blocos, em que todas as unidades possuem o mesmo peso  $w$ . O conjunto de saída das unidades de um plano é denominado mapa de atributos (*feature map*). Por compartilharem um mesmo peso, as unidades de um plano aplicam uma mesma operação em cada parte da entrada, dividida espacialmente, em caso de análise de imagens, por exemplo, ou temporalmente, em caso de análise de séries temporais, por exemplo (LECUN et al., 1998).

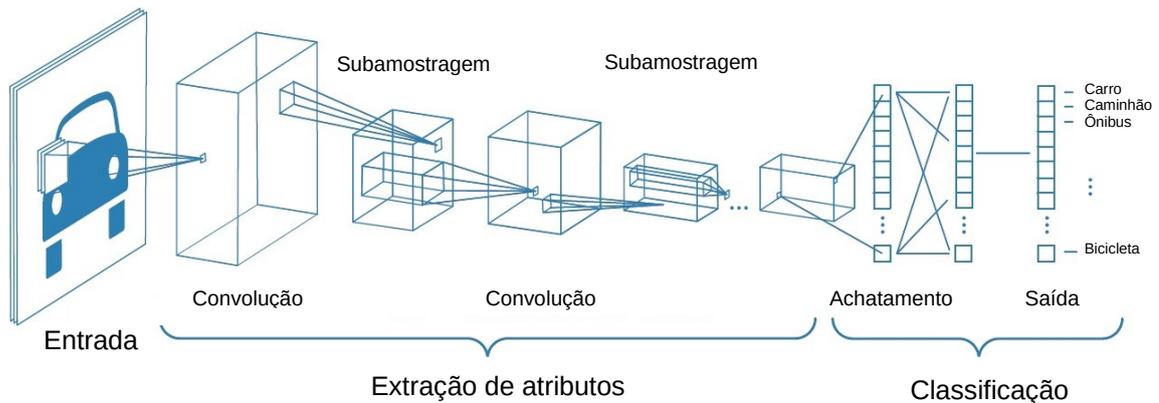
A camada de subamostragem (*pooling*) é responsável por substituir uma parte da entrada por um único valor, como o máximo ou a média de suas unidades. Em uma imagem, essa operação corresponde a diminuir a sua resolução, diminuindo a sensibilidade do algoritmo a ruídos e distorções (LECUN et al., 1998).

A convolução consiste em um mapeamento da entrada em unidades contíguas específicas do mapa de atributos. De acordo com LeCun et al. (1998), a implementação dessa operação consiste na operação de convolução, acrescida do termo independente (*bias*), o que dá origem ao nome do algoritmo. O núcleo da convolução é o conjunto de pesos de conexão usado pelas unidades em um mapa de atributos. A saída desse mapa de atributos será uma função de ativação aplicada a essa convolução. Além da usual função tangente hiperbólico, a função de ativação pode ser outras, como a sigmoide e a relu (*rectified linear unit function*), sendo que  $relu(x) = \max(0, x)$ .

A Figura 10 apresenta um esquema geral de Rede Neural Convolucional para classificação de veículos. São utilizadas sequências de camada de convolução seguida por uma camada de subamostragem, sendo que a quantidade de sequências a ser utilizada deve ser ajustada a cada problema. O conjunto de todas essas sequências formam a extração de atributos, de forma que, quanto mais profunda é uma camada de convolução, mais atributos ela consegue extrair e menor é a resolução de sua entrada. Ao final das sequências de camadas de convolução e subamostragem, os atributos extraídos são achatados, de forma que podem ser utilizados como entrada para um classificador, sendo que normalmente utiliza-se o MLP.

Considerando, no caso da Figura 10, a entrada como uma imagem de 32x32 píxeis, a primeira camada de convolução com um passo (*stride*) formado por uma área de 5x5 píxeis e produzindo, como saída, 6 mapas de atributos, então, cada mapa de atributos consistirá em unidades de atributos organizadas em uma grade de 28x28, composta por todas as combinações possíveis de imagens formadas por 5x5 píxeis. A convolução, então, aplica um filtro sobre essas partes da imagem, de forma que cada um dos 6 mapas de atributos sofrerá uma transformação diferente. Se a primeira camada de subamostragem considera uma grade de 2x2 atributos, então cada grade de 2x2, dentro da grade 28x28 do mapa de atributos, será mapeado em um único valor nessa camada de subamostragem.

Figura 10 – Rede Neural Convolutiva para classificação de veículos



Fonte: Adaptado de [MathWorks \(2017\)](#)

Esse mapeamento é feito calculando a média (no caso do *average pooling*) das saídas dessas 4 (2x2) unidades, multiplicando-a por um peso adaptativo, adicionando um *bias* e por último, aplicando uma função de ativação sigmoideal a esse valor. Dessa forma, a camada de subamostragem será composta por 6 mapas de atributos, sendo que cada mapa possui unidades dispostas em uma grade 14x14. As mesmas operações são realizadas para os demais pares de camadas de convolução e subamostragem.

Assim como no MLP, o erro da rede, para um problema de classificação binária, é a entropia cruzada, que pode ser minimizado utilizando algoritmos de atualização de pesos baseados em gradiente, como o método da retropropagação (*backpropagation*), detalhado em [Bishop \(2006\)](#).

### 3.3.5 Combinação de classificadores

A combinação de classificadores pode trazer algumas vantagens em relação ao uso de um único classificador, isoladamente. Essas combinações (*ensembles*) são, geralmente, utilizadas com a intenção de se aumentar a capacidade de predição de um conjunto de classificadores, melhorando métricas como acurácia, precisão e revocação. A principal motivação do uso dessas técnicas é o conflito que existe entre o viés e a variância em classificadores individuais. O viés (*bias*) consiste no erro do modelo devido à diferença entre sua complexidade e a complexidade da função que realmente descreve o objeto de predição. A variância é o erro relacionado à sensibilidade do modelo aos pontos individuais. Em combinações de classificadores, porém, variância e viés podem ser reduzidos sem que um deles afete, necessariamente, o outro ([BISHOP, 2006](#)).

Uma forma muito utilizada para combinar classificadores é o comitê de classificadores ou votação, procedimento que utiliza a média das predições dos classificadores

individuais para produzir uma predição final. Essa média pode ser simples, e nesse caso a combinação também é chamada de votação majoritária, ou média ponderada, sendo que nesse caso a probabilidade de cada entrada pertencer a uma determinada categoria é a média dessa probabilidade dada por todos classificadores individuais e a categoria predita é aquela de maior probabilidade média. Outras formas de combinação desses classificadores podem ser classificadas como ensacamento, reforço ou empilhamento.

### 3.3.5.1 Ensacamento

Ensacamento ou *bagging* é uma técnica de combinação de classificadores em comitês em que vários conjuntos de dados são formados aleatoriamente a partir do conjunto de dados de entrada. Assim, cada classificador base é treinado com um conjunto de dados diferente. Por fim, a predição final é dada pela votação de todos os classificadores base (BISHOP, 2006).

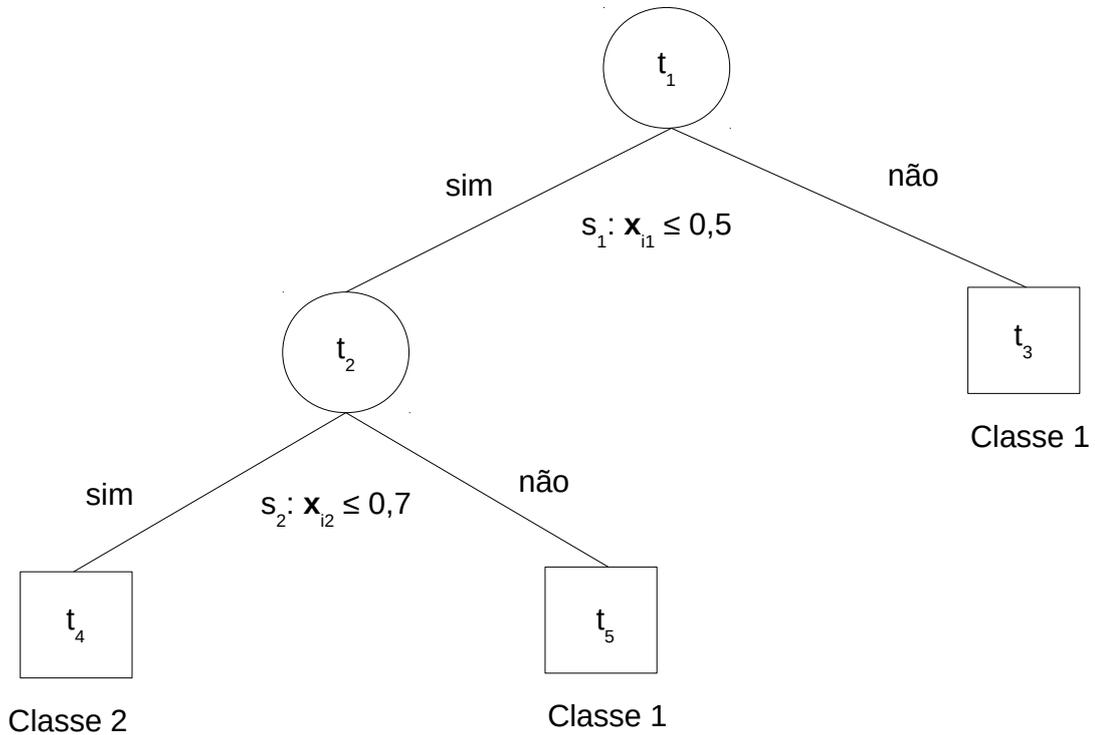
Um exemplo de ensacamento é o algoritmo Floresta Aleatória ou *Random Forest* (BREIMAN, 2001), uma combinação de Árvores de Decisão em que cada Árvore recebe um subconjunto aleatório do conjunto de amostras de entrada  $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  independente dos subconjuntos das outras Árvores, mas com mesma distribuição. A combinação das Árvores de Decisão se dá por meio do voto, sendo que, para uma entrada  $\mathbf{x}_i$ , a classe predita por mais Árvores, se torna a predição  $\hat{y}(\mathbf{x}_i)$  da Floresta Aleatória. Esse algoritmo também consegue fornecer as probabilidades de pertencimento de uma amostra a uma classe, conforme os resultados de cada Árvore, e a importância de cada atributo na predição de sua categoria.

Existem vários algoritmos que criam Árvores de Decisão de diferentes formas, sendo que um algoritmo de Árvore de Decisão muito utilizado é o CART (Classification and Regression Trees) (BREIMAN et al., 1984), cuja estrutura básica de árvores é composta por decisões binárias que dividem um nó em dois outros nós, até que sejam criados os nós folhas, que correspondem às classes do problema abordado.

A Figura 11 apresenta os elementos básicos de uma Árvore de Decisão, nós  $t_1, t_2, t_3, t_4$  e  $t_5$ , sendo que o nó raiz  $t_1$  é formado pelo conjunto de todas as amostras de treinamento  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ,  $t_2$  e  $t_3$  são divisões de  $t_1$ , de forma que  $t_1 = t_2 \cup t_3$ . Os nós  $t_3, t_4$  e  $t_5$  são denominados nós folha e cada um desses conjuntos de amostras é atribuído a uma classe. A partir do nó raiz  $t_1$ , a divisão  $s_1$ , cuja regra de decisão é transmitir para o nó filho da esquerda (sim) toda amostra  $i$  de  $t_1$  em que seu atributo 1 ( $x_{i1}$ ) seja menor ou igual a 0,5 e as demais amostras são enviadas para o nó filho da direita (não).

Cada nó terminal ou folha em uma Árvore de Decisão possui um modelo linear. Na Figura 11, a predição da Árvore pode ser descrita como:

Figura 11 – Exemplo de Árvore de Decisão para problema de classificação binário



Fonte: Adaptado de Breiman et al. (1984)

$$\hat{y}(\mathbf{x}_i) = \begin{cases} \sum_{j=1}^d \mathbf{w}^{(3)} x_{ij} + b^{(3)}, & \text{se } \mathbf{x}_i \in t_3 \\ \sum_{j=1}^d \mathbf{w}^{(4)} x_{ij} + b^{(4)}, & \text{se } \mathbf{x}_i \in t_4 \\ \sum_{j=1}^d \mathbf{w}^{(5)} x_{ij} + b^{(5)}, & \text{se } \mathbf{x}_i \in t_5 \end{cases},$$

onde  $\mathbf{w}^{(i)}$  representa os pesos treinados no nó  $i$  e  $b^{(i)}$  o termo independente (*bias*) nesse nó.

De acordo com Breiman et al. (1984), a geração da Árvore de Decisão necessita de quatro elementos: um conjunto de perguntas do tipo sim ou não, em que se divide as amostras de um nó em dois subconjuntos; uma medida de avaliação  $\phi(s,t)$  do critério de divisão, que avalia cada divisão  $s$  de um nó  $t$ ; um critério de parada para a divisão ou ramificação da árvore; e uma regra para ligar cada nó folha da árvore a uma classe.

O conjunto de perguntas deve ser criado para dividir de forma mais balanceada possível o conjunto de amostras. A partir do nó raiz, por exemplo, a divisão ideal é aquela que divide ao meio o conjunto de amostras. Para descrever a distribuição de amostras em um nó, Breiman et al. (1984) utiliza o conceito de função de impureza  $\varphi$  representada pelas porcentagens  $p_i$  de cada amostra  $i$  em um nó, em uma tupla  $(p_1, p_2, \dots, p_c)$ , para um problema com  $c$  classes, nas condições de que  $p_i \geq 0$ ,  $i = 1, 2, \dots, c$  e  $\sum_{i=1}^c p_i = 1$ . O

máximo de  $\varphi$  se encontra no ponto  $(1/c, 1/c, \dots, 1/c)$ , o mínimo se encontra nos pontos  $(1, 0, \dots, 0)$ ,  $(0, 1, 0, \dots, 0)$ , ...,  $(0, 0, \dots, 0, 1)$  e  $\varphi$  é uma função simétrica de  $p_1, \dots, p_c$ .

A partir do conceito de função de impureza, pode-se avaliar, então, a divisão de um nó  $t$  utilizando uma medida de impureza  $im(t) = \varphi(p(1|t), p(2|t), \dots, p(c|t))$ . Breiman et al. (1984) utiliza, como função de impureza, o índice de Gini:

$$\varphi(p(1|t), p(2|t), \dots, p(c|t)) = \sum_{i \neq j} p(i|t)p(j|t),$$

que, em problemas com duas classes, pode ser escrito como:

$$\varphi(p(1|t), p(2|t)) = p(1|t)p(2|t).$$

A qualidade da divisão  $s$  de um nó  $t$ , portanto, está relacionada com a diferença entre a impureza de um nó e a dos seus nós filhos, ou seja (BREIMAN et al., 1984):

$$\phi(s, t) = im(t) - p_l im(t_l) - p_r im(t_r),$$

onde  $t_l$  e  $t_r$  são nós filhos do nó  $t$  e  $p_l$  e  $p_r$ , as proporções de amostras de  $t$  enviadas para  $t_l$  e  $t_r$ , respectivamente. A melhor divisão  $s^*$ , portanto, é aquela que maximiza  $\phi(s, t)$ . Defini-se um nó  $t$  como nó folha se  $\phi(s^*, t)$  é menor que um limite  $\beta$  determinado.

Definidos os nós terminais, as regras de associação de cada um desses nós a uma categoria é determinada. No exemplo da Figura 11, a regra para classe 1 é  $(x_{i1} > 0,5)$  ou  $(x_{i1} \leq 0,5$  e  $x_{i2} > 0,7)$ . Para a classe 2, a regra é  $(x_{i1} > 0,5$  e  $x_{i2} \leq 0,7)$ .

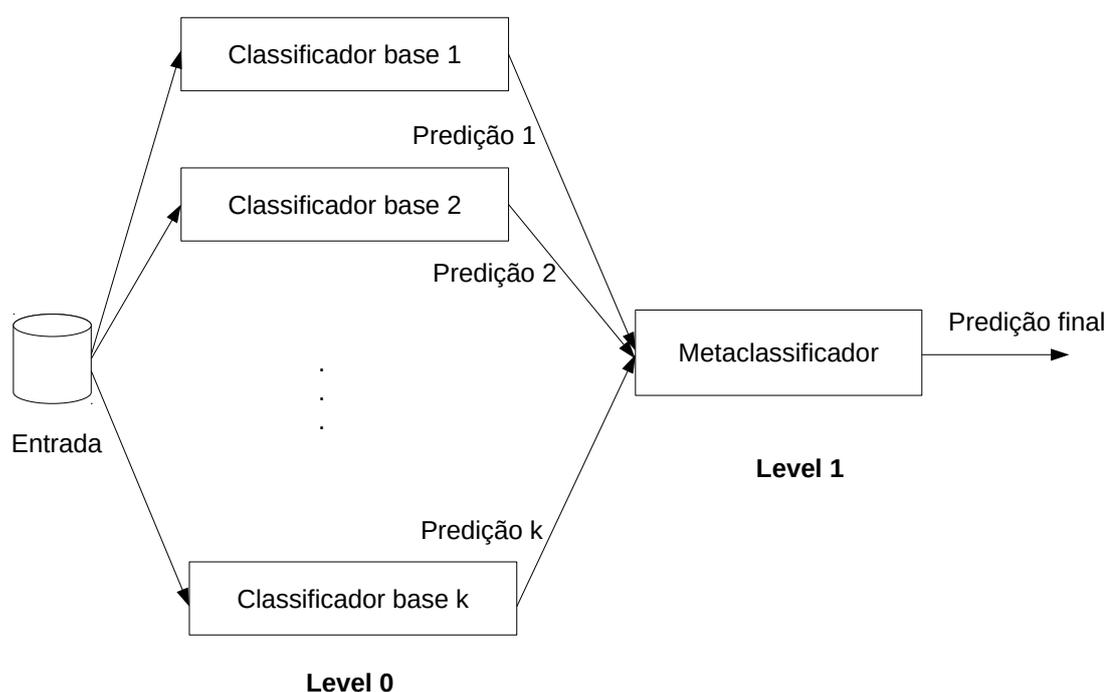
### 3.3.5.2 Empilhamento

Empilhamento ou *stacking* é um método de combinação de generalizadores, proposto por Wolpert (1992), que consiste em criar diferentes níveis de generalizadores, diminuindo o erro de generalização de algoritmos base ou individuais. Para um problema de classificação, os algoritmos classificadores que solucionam diretamente o problema de classificação abordado são denominados classificadores nível 0. Supondo que hajam  $k$  classificadores diferentes que solucionam esse mesmo problema de classificação, os classificadores nível 1 utilizam as predições desses  $k$  classificadores como atributos de entrada para cada amostra do problema original. As predições dos classificadores nível 1 também podem ser utilizadas por outros algoritmos (algoritmos nível 2) e assim por diante até que se atinja o nível de empilhamento desejado.

Um empilhamento de 2 níveis pode ser observado na Figura 12, em que  $k$  diferentes classificadores base, de nível 0, produzem predições com base em atributos de entrada.

Por haver apenas dois níveis, o segundo nível (nível 1) é formado por apenas um classificador que, por combinar predições de outros classificadores, também é denominado meta classificador. Esse meta classificador é treinado com as predições dos classificadores de nível 0 e, finalmente, apresenta a predição final para cada nova amostra.

Figura 12 – Empilhamento de 2 níveis



### 3.3.5.3 Reforço

Reforço ou *boosting* é uma técnica que combina vários classificadores, denominados classificadores base, de forma que o resultado final de todos esses classificadores apresente um desempenho superior ao de um classificador base apenas. De acordo com Bishop (2006), a técnica de reforço pode providenciar bons resultados mesmo utilizando classificadores "fracos", cujo desempenho é apenas um pouco melhor que o de um classificador aleatório.

O reforço se difere do ensacamento por treinar os classificadores base em sequência, de forma que cada classificador é treinado utilizando um conjunto de dados ponderado cujos pesos de cada entrada é determinada pelo desempenho dos classificadores prévios, sendo que as entradas classificadas incorretamente têm seus pesos aumentados para o próximo classificador. E, após o treinamento de todos os classificadores, o resultado final é dado pela votação majoritária de todos os classificadores (BISHOP, 2006).

O algoritmo de reforço mais utilizado é o AdaBoost, desenvolvido por [Freund e Schapire \(1996\)](#). Esse algoritmo é composto por três passos. Primeiro, considerando um conjunto de coeficientes  $\{w_n\}$  de pesos para cada uma das  $N$  amostras de entrada, o peso inicial  $w_n^{(1)} = 1/N$ , para toda amostra  $n \in \{1, \dots, N\}$ . No segundo passo, os  $M$  classificadores são sequencialmente aplicados aos dados de entrada multiplicados pelos seus respectivos coeficientes de peso, que são atualizados iterativamente conforme o desempenho da classificação. Para essa atualização, considera-se o erro  $\epsilon_m$  de cada classificador dado por:

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}},$$

$y_m(x_n)$  a predição do classificador  $m$  para a  $n$ -ésima amostra de entrada,  $t_n$ , o valor verdadeiro da amostra  $n$  e  $I(y_m(x_n) \neq t_n)$  igual a 1 se  $y_m(x_n) \neq t_n$  e 0, caso contrário. Assim, o coeficiente de pesos para o próximo classificador ( $m + 1$ ) será igual a:

$$w_n^{(m+1)} = w_n^{(m)} e^{\alpha_m I(y_m(x_n) \neq t_n)},$$

com

$$\alpha_m \leftarrow \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}.$$

Por fim, a predição final é dada pela votação majoritária dos classificadores.

Assim como o AdaBoost, o Reforço Gradiente, ou Gradient Boosting, é um algoritmo de reforço muito utilizado. Enquanto o AdaBoost valoriza o erro, aumentando o peso das predições erradas, o Reforço Gradiente foca apenas nos erros e ignora os acertos. Dessa forma, ele constrói uma árvore baseada apenas nos erros da árvore anterior. Para isso, o Reforço Gradiente calcula os resíduos, ou seja, a diferença entre cada predição e a resposta real, e avalia cada modelo pelo somatório desses resíduos. ([WADE; GLYNN, 2020](#))

### 3.3.6 Medidas de desempenho

Os algoritmos de classificação apresentados descrevem o funcionamento da geração de modelos de predição para cada algoritmo. Esse procedimento de treinamento do modelo, em Aprendizagem de Máquina, é realizado na fase de treinamento. Como visto na descrição dos algoritmos, cada um possui um conjunto de hiperparâmetros, cujos ajustes são realizados na fase de validação do modelo. Por fim, a avaliação final do modelo é realizada na fase de teste. Para cada uma dessas fases, diferentes amostras de entrada são utilizadas.

Para a comparação do desempenho do modelo para cada parâmetro, na validação, ou para a avaliação final do modelo, no teste, são necessárias medidas de desempenho.

Várias medidas ou métricas de desempenho podem ser utilizadas para avaliar ou comparar classificadores, sendo que algumas das mais utilizadas são a acurácia, precisão, revocação e F1. Todas essas medidas citadas são dadas em porcentagem, ou seja, variam entre 0 e 1, tendo o valor 1 como melhor valor possível.

Em um classificador binário, pode-se dizer que uma predição é positiva, quando sua categoria é 1, ou negativa, quando sua categoria é 0. Uma predição é correta quando a classe predita é igual à classe verdadeira de um vetor de entrada ou instância. Assim, para a definição dessas medidas de desempenho, são definidos os seguintes valores:  $VP$  - verdadeiros positivos, que indica a quantidade de predições positivas corretas,  $VN$  - verdadeiros negativos, que indica a quantidade de predições negativas corretas,  $FP$  - falsos positivos, que indica a quantidade de predições positivas incorretas e  $FN$  - falsos negativos, que indica a quantidade de predições negativas incorretas.

### 3.3.6.1 Acurácia

A acurácia indica o percentual de acerto de todas as predições realizadas.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}.$$

### 3.3.6.2 Precisão

A precisão indica, para uma determinada classe de interesse, a porcentagem de predições dessa classe que foram corretas. Para a classe 1 (positiva):

$$\text{Precisão} = \frac{VP}{VP + FP}.$$

### 3.3.6.3 Revocação

A revocação indica, para uma determinada classe de interesse, a porcentagem das instâncias dessa classe preditas corretamente. Para a classe 1 (positiva):

$$\text{Revocação} = \frac{VP}{VP + FN}.$$

### 3.3.6.4 Métrica F1

A métrica F1 considera tanto a precisão quanto a revocação, sendo a média harmônica de ambas:

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}.$$

### 3.3.7 Seleção de atributos

Visto que em problemas de classificação normalmente existem grandes quantidades de variáveis que podem afetar um resultado, a seleção de atributos ou a redução de dimensionalidade do problema é uma técnica utilizada na tentativa de diminuir a complexidade do problema e, principalmente, eliminar variáveis consideradas ruídos, que atrapalham o desempenho do classificador, quando consideradas.

De acordo com [Guyon e Elisseeff \(2003\)](#), os métodos de seleção de atributos podem ser classificados em três tipos: os filtros, os empacotamentos (*wrappings*) e os métodos embarcados (*embedded*). Os filtros são aplicados antes do treinamento dos classificadores e são independentes destes, os empacotamentos criam diversos subconjuntos de atributos e realiza treino com cada um desses subconjuntos, selecionando o melhor subconjunto na etapa de validação e os métodos embarcados consistem de mecanismos próprios de um modelo que o permite eliminar atributos e, assim, reduzir a dimensionalidade do problema, tendo a regressão logística com regularização L1 (subseção [3.3.1.1](#)) como exemplo.

Esta subseção descreve apenas alguns métodos de filtro de atributos, por eles serem independentes dos algoritmos de classificação e, portanto, não ajustados a nenhum deles.

#### 3.3.7.1 Seleção de Atributos Baseada em Correlação

[Hall \(2000\)](#) propõe um filtro para seleção de atributos baseado no coeficiente de correlação de Pearson. A Seleção de Atributos Baseada em Correlação ou CFS (Correlation-based Feature Selection) é um algoritmo heurístico que remove atributos de um conjunto de atributos para problemas de Aprendizagem de Máquina. O algoritmo utiliza, como métrica a ser maximizada, a correlação entre a soma dos atributos e a variável alvo, denominada mérito, dada por:

$$r_{za} = \frac{k\bar{r}_{zi}}{\sqrt{k + (k-1)\bar{r}_{ii}}},$$

onde  $k$  é a quantidade de atributos de um subconjunto,  $\bar{r}_{zi}$  representa a média das correlações entre cada atributo e a variável de saída e  $\bar{r}_{ii}$  é a intercorrelação entre atributos média. As correlações são dadas pelos coeficientes  $r$  de correlação de Pearson:

$$r_{xy} = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \sqrt{n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2}},$$

sendo  $n$  a quantidade de elementos das variáveis  $x$  e  $y$ .

São propostos, em [Hall \(2000\)](#), três estratégias de busca gulosas para encontrar um bom subconjunto de atributos: seleção para frente, que inicia com um conjunto vazio e

adiciona sempre o atributo que faz com que o novo conjunto aumente seu mérito, enquanto existir tal atributo; a eliminação para traz inicia com o conjunto formado por todos os atributos disponíveis e remove, iterativamente, o atributo que, quando removido, forma um subconjunto de mérito maior que o anterior, enquanto houver tal atributo; e a estratégia do primeiro melhor, que pode começar tanto com um conjunto vazio e, adicionar iterativamente o primeiro atributo encontrado que, se adicionado, forma um novo subconjunto com mérito superior ao anterior, ou pode começar com o conjunto de todos os atributos e, de forma análoga, remover um atributo de cada vez.

### 3.3.7.2 Análise de Componentes Principais

A Análise de Componentes Principais (ACP) é uma técnica de redução de dimensionalidade que consiste na substituição de informações em uma matriz ou Quadro de Objetos Variáveis (QOV), composta por  $n$  objetos ou amostras e  $p$  variáveis, por outra matriz com  $k \leq p$  variáveis sintéticas não correlacionadas, que possuam quase toda a informação contida na matriz original. Em problemas compostos por muitas variáveis, espera-se que a ACP proporcione um novo conjunto de dados com  $k$  variáveis, sendo  $k$  muito menor que  $p$  e com a menor perda de informação possível (DUARTE, 1998).

Cada uma dessas novas variáveis, denominadas Componentes Principais (CP), são as combinações lineares das variáveis originais que possuem variância máxima. Os coeficientes são os autovetores da matriz original (QOV), ordenadas pelos autovalores dessa matriz (DUARTE, 1998).

Os autovalores  $\lambda_i (i = 1, 2, \dots, p)$  representam as variâncias dos objetos captadas por cada CP  $i$ . Como os autovalores são ordenados de forma decrescente,  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ . E a variância total dos componentes, a ser maximizada, é estimada por:

$$s^2 = \sum_{i=1}^p \lambda_i.$$

No caso em que a matriz de dados original é normalizada (média 0 e variância 1), esse valor corresponde ao traço de sua matriz de covariâncias (DUARTE, 1998).

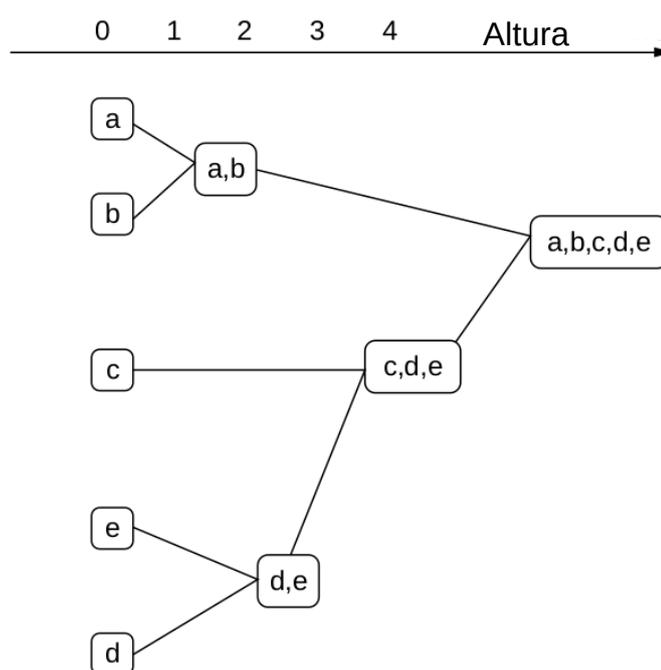
### 3.3.7.3 Agrupamento Hierárquico

O Agrupamento Hierárquico, de acordo com Everitt et al. (2011), é uma técnica que realiza uma série de agrupamentos nos dados de entrada, que variam de um único objeto ao agrupamento de todas os  $n$  objetos de entrada. Os demais agrupamentos são sucessivas fusões realizadas sobre os atributos ou objetos, criando grupos conforme as semelhanças entre esses objetos. Essas aglomerações são representadas por um diagrama de árvore denominado dendrograma. O nó raiz da árvore representa o agrupamento de todos os objetos e se encontra na posição de altura máxima da árvore, os nós terminais

são formados por cada um dos objetos separadamente e estão na altura 0 da árvore. Os agrupamentos intermediários dessa árvore unem dois outros grupos já formados e a altura desses agrupamentos é proporcional à afinidade entre os dois grupos que eles unem.

A Figura 13 mostra um exemplo de dendrograma para um Agrupamento Hierárquico dos objetos  $a$ ,  $b$ ,  $c$ ,  $d$  e  $e$ . O agrupamento  $(a,b)$ , que une os grupos  $a$  e  $b$ , é o agrupamento de menor altura, indicando que os objetos  $a$  e  $b$  são os objetos mais próximos.

Figura 13 – Exemplo de dendrograma



Fonte: Adaptado de [Everitt et al. \(2011\)](#)

Existem diferentes formas de se definir a distância entre dois grupos ou objetos. A forma mais simples, porém muito utilizada, é denominada ligação única e define a distância entre dois grupos como a menor distância entre dois objetos de cada grupo. Outras abordagens consideram os objetos mais distantes de cada grupo (ligação completa), a utilização da média ou mediana entre cada par de objetos dos dois grupos ([EVERITT et al., 2011](#)).

O Agrupamento Hierárquico pode reduzir a dimensionalidade de um problema de classificação, substituindo os  $n$  atributos originais por  $k$  agrupamentos, em uma certa altura do dendrograma. Segundo [Kaufman e Rousseeuw \(1990\)](#), a decisão da melhor altura ou melhor número  $k$  de grupos a se considerar pode ser feita utilizando diferentes métodos, como o método da Silhueta Média, que consiste em calcular o valor da Silhueta Média da árvore truncada em diferentes quantidades de grupos e decidir pelo  $k$  que maximiza esse valor.

Para definir o valor de Silhueta, pode-se considerar a distância média entre cada objeto  $i$  para os demais objetos de seu grupo  $c_i$ , da seguinte forma:

$$a(i) = \frac{1}{|c_i| - 1} \sum_{j \in c_i, j \neq i} d(i, j),$$

onde  $d(i, j)$  é a distância entre as variáveis  $i$  e  $j$ . A média de distância entre um objeto  $i$  e um agrupamento  $c_k$  diferente do agrupamento de  $i$ , é dado por:

$$b(i) = \min_{k \neq i} \frac{1}{|c_k|} \sum_{j \in c_k} d(i, j)$$

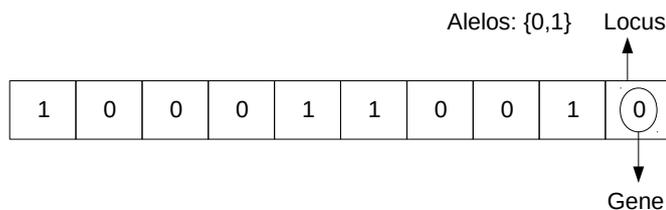
e, finalmente, a Silhueta de uma variável  $i$  é dada por (KAUFMAN; ROUSSEEUW, 1990):

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{se } a(i) < b(i) \\ 0, & \text{se } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{se } a(i) > b(i) \end{cases} .$$

### 3.4 Algoritmo Genético

Algoritmos Genéticos (AG) são algoritmos baseados na Teoria da Evolução de Darwin e seus operadores são análogos aos processos de evolução genética. As estruturas de dados de um algoritmo genético, denominadas cromossomos, representam os indivíduos da população. Cada unidade do cromossomo é denominada gene, está contida em uma posição ou locus do cromossomo e pode assumir diferentes valores dentre um conjunto de valores possíveis, denominados alelos. Originalmente, os algoritmos genéticos utilizavam apenas conjuntos binários  $\{0,1\}$  para a representação dos indivíduos, como ilustrado na Figura 14, que apresenta um cromossomo com dez posições ou locus e genes com alelos 0 ou 1 (binários) (CASTRO, 2006).

Figura 14 – Representação de um cromossomo



Fonte: Adaptado de Castro (2006)

Além dos valores binários, os alelos podem ser constituídos por valores reais, inteiros ou por palavras (*strings*). Cada indivíduo ou cromossomo é também uma potencial solução

para o problema abordado, caracterizado por uma função objetivo utilizada para avaliar esses indivíduos.

Primeiramente, em um Algoritmo Genético, é realizada a etapa de geração das soluções iniciais ou inicialização da população de indivíduos. Normalmente, esses indivíduos são formados de forma aleatória dentro do domínio do problema. Outras formas de realizar essa inicialização são criar indivíduos espalhados dentro do espaço de busca ou utilizar outras heurísticas para que as soluções iniciais já sejam inicializadas com boas avaliações. Após essa fase inicial, realizada uma única vez, os procedimentos realizados pelo algoritmo genético são uma evolução iterativa dessas soluções iniciais, utilizando os métodos de cruzamento e mutação, análogos aos procedimentos presentes na Teoria da Evolução, e a seleção dos indivíduos para a próxima iteração, ou geração. Essas iterações se encerram quando um critério de parada é satisfeito, como o atingimento de uma quantidade máxima de gerações, de um tempo máximo de execução ou de uma quantidade determinada de gerações sem alterações na melhor solução obtida. Em seguida, cada uma dessas etapas de um AG serão explicadas com maiores detalhes.

### 3.4.1 Cruzamento

O cruzamento ou recombinação, em um AG, é um procedimento que troca informações dos genes de dois cromossomos para a geração de novos cromossomos, fazendo analogia ao processo de reprodução sexuada. O par de indivíduos ou cromossomos que realiza a recombinação denomina-se pais e os novos indivíduos gerados são denominados filhos. A seleção desses pais pode ser realizada de várias formas, sendo que alguns exemplos são a seleção aleatória, em que os pais são selecionados por procedimentos aleatórios, e a seleção por torneio, em que o indivíduo de melhor avaliação é selecionado entre alguns indivíduos escolhidos aleatoriamente. De forma análoga ao processo biológico de reprodução, em que nem toda tentativa de reprodução resultam em filhos, operadores de cruzamento no AG utilizam um parâmetro  $pr$  (probabilidade de recombinação), indicando a probabilidade de cada par de pais gerarem novos indivíduos [Talbi \(2009\)](#).

Para problemas de variáveis reais, o operador de cruzamento SBX (*Simulated Binary Crossover*), proposto por [Agrawal, Deb e Agrawal \(1995\)](#) é muito utilizado. O nome do operador, Cruzamento Binário Simulado, em português, demonstra a intenção dos autores em desenvolver um operador de cruzamento com as mesmas propriedades do cruzamento binário, de forma apropriada para variáveis reais. Essas propriedades que se desejou manter foram: a média dos valores dos filhos devem ser iguais à média dos valores dos seus pais (representada por  $\bar{pai}$ ), para cada variável representada no cromossomo; e a distribuição de probabilidade do fator de espalhamento  $\Gamma$ ,  $dp(\gamma)$ , descrito por funções similares às funções do cruzamento binário:

$$dp(\Gamma) = 0,5(n_c + 1)\Gamma^{n_c}, \Gamma \leq 1$$

$$dp(\Gamma) = 0,5(n_c + 1)\Gamma^{n_c} \frac{1}{\Gamma^{n_c+2}}, \Gamma > 1,$$

com  $n_c$  sendo um parâmetro que regula a distância entre valores dos pais e filhos, de forma que valores maiores de  $n_c$  produzem filhos com valores mais próximos dos pais, e

$$\Gamma = \left| \frac{\text{filhox} - \text{filhoy}}{\text{paix} - \text{paiy}} \right|.$$

O operador SBX consiste no seguinte procedimento: primeiramente um número  $u \in [0,1]$  é escolhido aleatoriamente e, posteriormente, seleciona-se o valor  $\bar{\Gamma}$  que estabelece uma área abaixo do gráfico de distribuição de probabilidade  $dp(\bar{\Gamma})$  igual a  $u$ , para um  $n_c$  determinado. Por fim, os filhos são gerados da seguinte forma:

$$\text{filhox} = \bar{\text{pai}} - 0,5\bar{\Gamma}(\text{paix} - \text{paiy})$$

$$\text{filhoy} = \bar{\text{pai}} + 0,5\bar{\Gamma}(\text{paix} - \text{paiy}).$$

### 3.4.2 Mutação

Em algoritmos genéticos, a diversificação da população é realizada por meio da mutação, processo análogo à mutação genética, em que um alelo de um gene é alterado por outro. A mutação polinomial proposta por Deb (2001) é aplicada em variáveis reais. Esse operador causa uma perturbação, garantindo que os novos valores no cromossomo mutado esteja entre um limite inferior  $li_j$  e um limite superior  $ls_j$ , para uma determinada variável  $j$ . O efeito dessa perturbação depende de um parâmetro  $n_m$ , de forma que quanto maior esse parâmetro, menor é a perturbação realizada. Utilizando esse operador, primeiramente, um valor  $u \in [0,1]$  é selecionado aleatoriamente e os cromossomos mutantes são formados da seguinte forma:

$$cm_j = co_j + \bar{\delta}_l(co_j - li_j), \text{ se } u \leq 0,5$$

$$cm_j = co_j + \bar{\delta}_r(ls_j - co_j), \text{ se } u > 0,5,$$

onde

$$\bar{\delta}_l = (2u)^{1/(1+n_m)} - 1, \text{ se } u \leq 0,5$$

$$\bar{\delta}_r = 1 - (2(1 - u))^{1/(1+n_m)}, \text{ se } u > 0,5.$$

### 3.4.3 Seleção

Após os procedimentos de cruzamento e mutação, uma nova população é formada, a partir da população original de determinada geração. A seleção é o procedimento responsável por determinar a porção de indivíduos, dentre os indivíduos originais e os novos, que fará parte da população original da próxima geração. O método de seleção mais básico consiste em escolher todos os novos indivíduos para a próxima iteração ou geração. Um conceito importante na etapa de seleção é o elitismo, sendo que uma seleção com alto grau de elitismo é aquela que transfere grande parte dos melhores indivíduos, de acordo com seus valores de avaliação (*fitness*), para a próxima geração (TALBI, 2009).

Tradicionalmente, a seleção é realizada pelo método da roleta. Considerando que a função de avaliação produza apenas valores positivos, sendo que valores maiores indicam maior adaptabilidade de um indivíduo (maior valor para função objetivo em problema de maximização e menor valor, para problema de minimização), o método da roleta seleciona indivíduos com uma probabilidade proporcional aos seus valores de avaliação diante dos valores de todos os indivíduos da população (CASTRO, 2006).

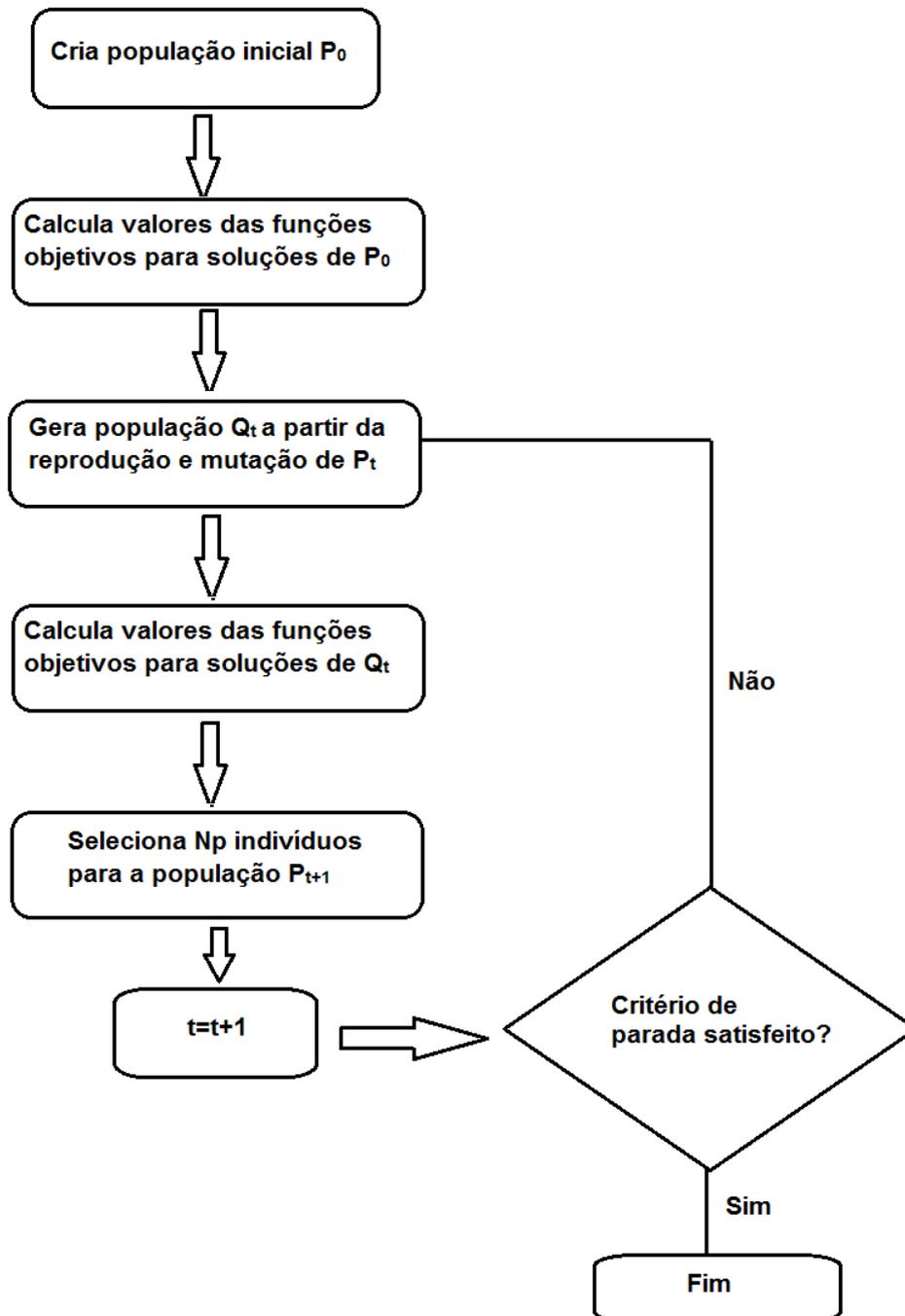
### 3.4.4 Algoritmo Genético Multiobjetivo

Tradicionalmente, o Algoritmo Genético é utilizado em problemas de decisão monoobjetivo, ou seja, compostas por uma única função objetivo que determina a avaliação das soluções. No entanto, alguns Algoritmos Genéticos para problemas multiobjetivo foram propostos, como o NSGA-II (*Non-dominated Sorting Genetic Algorithm II*), proposto por Deb et al. (2000).

A Figura 15 mostra o fluxograma de um NSGA-II com  $N_p$  indivíduos na população. Na figura, a população original de indivíduos, em uma geração  $t$ , é denominada  $P_t$  e a nova população, formada por indivíduos recombinados e mutados, representada por  $Q_t$ . Pode-se notar que o funcionamento do NSGA-II é muito semelhante ao funcionamento de um AG tradicional, exceto pela forma com que são realizadas a avaliação e a seleção dos indivíduos. A avaliação é realizada em dois procedimentos, sendo eles a ordenação não-dominada e a medida da distância de aglomeração.

A ordenação não-dominada ordena os indivíduos da população em fronteiras, com base no conceito de dominância. Pode-se dizer que uma solução é dominada em relação a outra se a primeira possui um valor inferior que a outra para alguma função objetivo e não possui, para nenhuma outra função objetivo, um valor maior que a outra. Soluções que não são dominadas por nenhuma outra são chamadas de soluções não dominadas. Na ordenação de fronteiras, a primeira fronteira é formada pelos indivíduos que não são dominados (também chamada de fronteira de Pareto), a segunda, por indivíduos dominados apenas por indivíduos da primeira fronteira, e assim sucessivamente, até que todos os

Figura 15 – Fluxograma básico do algoritmo NSGA-II

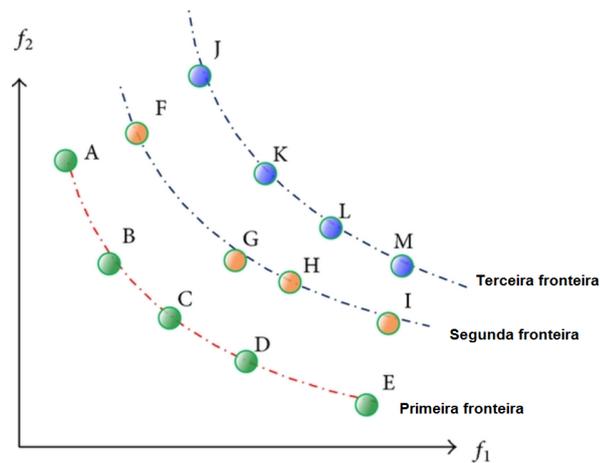


Fonte: [Ferreira \(2018\)](#)

indivíduos estejam em uma fronteira, como mostra a Figura 16. [Deb et al. \(2000\)](#) propõe o algoritmo *Fast Nondominated Sorting* para realizar essa ordenação.

A distância de aglomeração, para cada indivíduo, consiste na distância entre seus vizinhos em uma fronteira não-dominada. A medida dessa distância é feita ordenando, para cada fronteira, seus indivíduos de acordo com uma função objetivo e, posteriormente, somando, para cada indivíduo  $i$ , as distâncias normalizadas entre as funções objetivos dos vizinhos  $i - 1$  e  $i + 1$  ([DEB et al., 2000](#)).

Figura 16 – Ordenação não-dominada



Fonte: Adaptado de Wang, Tu e Chen (2015)

Considerando a ordenação não-dominada de fronteiras e a distância de aglomeração, o NSGA-II realiza uma seleção elitista que escolhe os melhores indivíduos de uma geração para constituírem a população na próxima geração. Considera-se que um indivíduo é melhor que o outro se ele está em uma fronteira não-dominada de menor grau, começando pela primeira fronteira, e em caso de empate, se sua distância de aglomeração é maior que a distância do outro (DEB et al., 2000).

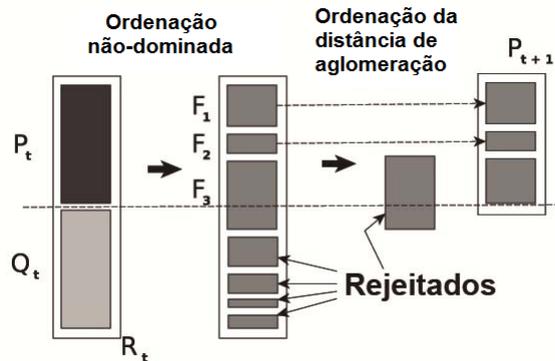
A Figura 17 descreve o processo de seleção do NSGA-II, começando pela criação de uma população  $R_t$ , formada pelas populações originais e novas na geração  $t$ .  $R_t$ , portanto, possui tamanho  $2N_p$ , devendo ser reduzida, assim, pela metade. Para isso, todos os indivíduos de uma fronteira, começando pela primeira fronteira, são adicionados à população da próxima geração  $P_{t+1}$ , se o tamanho de  $P_{t+1}$  não exceder  $N_p$ . Caso a adição de indivíduos de uma fronteira à  $P_{t+1}$  resulte em mais de  $N_p$  indivíduos para a próxima geração, como no caso da terceira fronteira da figura ( $F_3$ ), os indivíduos dessa fronteira são ordenados de forma descendente em relação à distância de aglomeração e inseridos em  $P_{t+1}$  nessa ordem até que a população da próxima geração contenha exatos  $N_p$  indivíduos.

#### 3.4.4.1 NSGA-II aplicado à Otimização de Portfólio

Deb et al. (2011) propõem um algoritmo NSGA-II particularmente interessante para este trabalho por ser desenvolvido para solucionar um modelo multiobjetivo de Otimização de Portfólio média-variância, ou seja, possui como funções objetivo, o retorno esperado a ser maximizado e o risco, definido pela variância do portfólio conforme proposto por Markowitz (1952), a ser minimizado.

Nesse NSGA-II, os cromossomos são constituídos de  $n$  valores reais, onde  $n$  é a quantidade de ativos disponíveis e cada valor  $w_i$  do cromossomo indica a proporção de

Figura 17 – Seleção do NSGA-II



Fonte: Adaptado de Deb et al. (2000)

investimento no ativo  $i$ . A geração da população inicial, no algoritmo de Deb et al. (2011), constrói  $N$  indivíduos, com  $N$  sendo o tamanho da população do NSGA-II, da seguinte forma: para cada indivíduo da população é atribuído um valor aleatório  $k$  entre  $k_{min}$  e  $k_{max}$ , indicando a cardinalidade desse indivíduo. Assim,  $k$  posições no cromossomo são selecionados aleatoriamente e para cada gene selecionado é atribuído um valor aleatório de proporção entre  $w_{min}$  e  $w_{max}$ , enquanto os demais genes são inicializados com valor 0. O algoritmo termina ao final de  $G$  gerações.

No cruzamento proposto por Deb et al. (2011), dois pais geram um único filho, de forma que esse procedimento é realizado  $N$  vezes para a geração de  $N$  filhos. Em cada procedimento, dois pais são selecionados aleatoriamente na população. Sendo  $k_1$  e  $k_2$  as cardinalidades dos pais 1 e 2, respectivamente, o filho gerado possuirá uma cardinalidade  $k$  definida de forma aleatória, com  $k$  sendo um valor entre  $\min(k_1, k_2)$  e  $\max(k_1, k_2)$ . Considerando o caso em que o pai 1 possui cardinalidade 5 ( $k_1 = 5$ ) e o pai 2, cardinalidade 6, como ilustrado na Figura 18, onde os valores  $a_i$  e  $b_i$  representam valores diferentes de 0 para genes do pai 1 e do pai 2, respectivamente.

Figura 18 – Exemplo de dois pais para  $n = 10$

Pai 1:  $a_1$  0  $a_3$  0  $a_5$  0 0  $a_8$  0  $a_{10}$

Pai 2:  $b_1$  0  $b_3$   $b_4$   $b_5$  0  $b_7$  0 0  $b_{10}$

Fonte: Adaptado de Deb et al. (2011)

Ainda seguindo o exemplo da Figura 18, o filho possuirá valor 0 nas posições dos cromossomos em que os dois pais possuem valor 0 e valor diferente de 0 nas posições em que ambos possuem valores diferentes de 0, sendo que nesse caso o filho possuirá um valor

determinado pela aplicação do operador SBX (*Simulated Binary Crossover*) limitado pelos valores  $w_{min}$  e  $w_{max}$ . No exemplo considerado, ambos os pais possuem valores diferentes de 0 nas posições 1, 3, 5 e 10, totalizando 4 valores diferentes de 0 para o filho. Como a cardinalidade selecionada para o filho é 5, outra posição deve diferir de 0. Para isso, é selecionada aleatoriamente uma posição em que apenas um dos pais possui valor diferente de 0 e, nessa posição, o filho assume o valor deste pai. Atingida a cardinalidade selecionada (5, neste exemplo), os outros genes assumem valor 0. Considerando que a posição 7 foi escolhida aleatoriamente, o filho resultante é descrito na Figura 19, onde  $?_i$  é o resultado da operação SBX para os pais na posição  $i$ .

Figura 19 – Filho resultante para exemplo anterior

Filho:  $?_1$  0  $?_3$  0  $?_5$  0  $b_7$  0 0  $?_{10}$

Fonte: Adaptado de [Deb et al. \(2011\)](#)

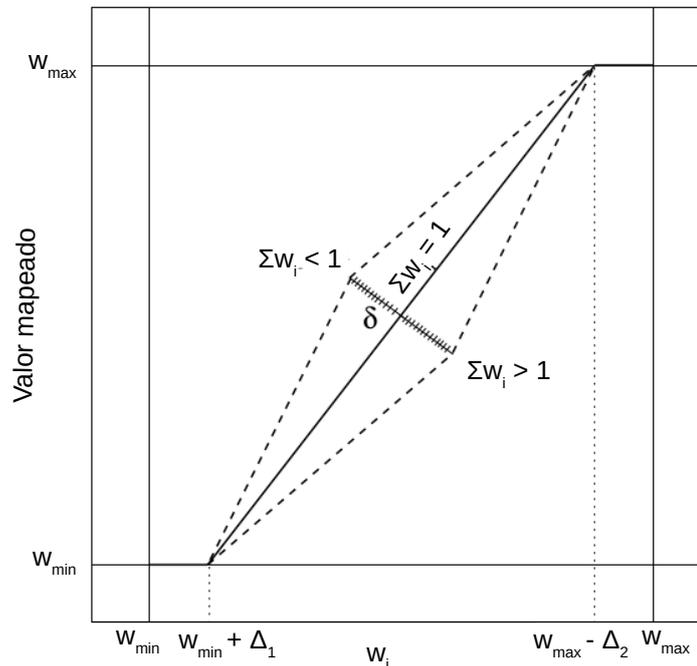
Para a mutação, [Deb et al. \(2011\)](#) propõe dois operadores. o primeiro altera valores de proporções de ativos presentes no portfólio, ou seja, perturba valores em posições não nulas de um cromossomo. Para isso, cada ativo não nulo é alterado, com uma probabilidade  $p_m$ , de acordo com o operador de mutação polinomial limitado pelos valores  $w_{min}$  e  $w_{max}$ . O segundo operador atribui os valores de proporções de  $Z$  ativos não nulos, escolhidos aleatoriamente, a  $Z$  ativos nulos, também aleatórios, e iguala as proporções dos  $Z$  ativos originalmente não nulos a zero, de forma que a cardinalidade  $k$  do portfólio ou indivíduo é mantida.

No algoritmo de [Deb et al. \(2011\)](#) é considerado a restrição de que  $\sum_{i=1}^n w_i = 1$  e, considerando que após a geração da população inicial, da operação de cruzamento e das operações de mutação a população possa ter indivíduos que violem essa restrição, é proposto um método de reparação. Primeiro, são definidos valores  $\Delta_1$  e  $\Delta_2$  de forma que valores entre  $w_{min}$  e  $w_{min} + \Delta_1$  são transformados em  $w_{min}$ , e valores entre  $w_{max} - \Delta_2$  e  $w_{max}$  são transformados em  $w_{max}$ . Posteriormente, a função de mapeamento apresentada na Figura 20 é aplicada, sendo que o valor do meio das retas de mapeamento são distanciadas em um valor  $\delta$  iterativamente até que o somatório das proporções  $w_i$  esteja arbitrariamente próxima de 1, ou seja,  $\sum_{i=1}^n w_i + \epsilon \geq 1$  e  $\sum_{i=1}^n w_i - \epsilon \leq 1$  ( $\epsilon \geq 0$ ).

#### 3.4.4.2 Medidas de desempenho

Em um AG tradicional, monobjetivo, dois algoritmos diferentes podem ser comparados com base na melhor solução gerada por cada um. No caso do NSGA-II, que proporciona

Figura 20 – Função de mapeamento



Fonte: Adaptado de [Deb et al. \(2011\)](#)

uma fronteira de soluções, a comparação entre diferentes algoritmos deve ser feita com base na cobertura das soluções, ou seja, a sua proximidade com a fronteira Pareto-ótimo (ou fronteira eficiente) e no espalhamento dessas soluções na fronteira, sendo que geralmente é desejável que os indivíduos estejam bem espalhados. O hipervolume é uma métrica muito utilizada para a comparação, tanto da cobertura, quanto do espalhamento das soluções. Já a métrica  $\Delta$  é utilizada unicamente para a avaliação do espalhamento ou diversidade dos indivíduos na fronteira.

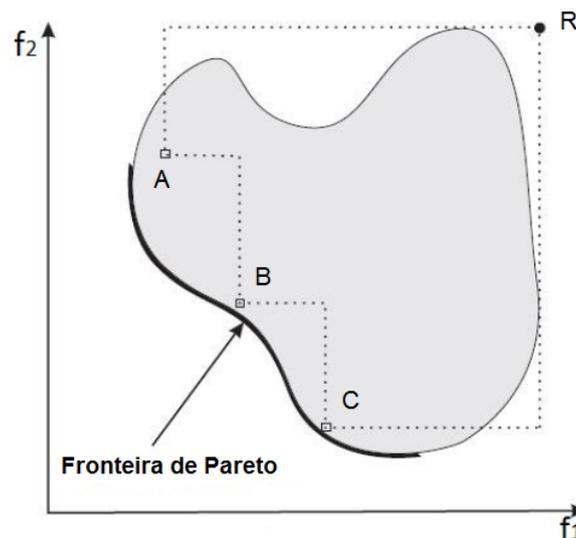
O hipervolume, também chamado de métrica  $S$ , é um indicador de qualidade de uma fronteira gerada por um algoritmo multiobjetivo. Foi proposta por [Zitzler e Thiele \(1998\)](#) e mede o volume do espaço dominado pela fronteira de Pareto gerada por um algoritmo. Dessa forma, quanto mais próximas da fronteira Pareto-ótimo e mais uniformemente distribuídas estão as soluções ou indivíduos na fronteira, maior o valor do hipervolume e, por isso, esse indicador analisa tanto a cobertura quanto a diversidade das soluções, apesar da cobertura causar maior impacto em seu valor. Uma propriedade importante dessa métrica é que, se todas as soluções de um conjunto  $A$  são fracamente dominadas por soluções de um conjunto  $B$ , o hipervolume de  $B$  será maior que o de  $A$ .

Seja  $\mathbf{S}$  o conjunto de soluções não dominadas gerado por um algoritmo multiobjetivo,  $R$  um ponto de referência, dominado por todas as soluções de  $\mathbf{S}$  e  $v_i$  o hipercubo formado pelo espaço dominado pela solução  $S_i \in \mathbf{S}$ , tendo os valores de  $R$  como limite. Assim, o hipervolume é o volume da união dos hipercubos  $v_i$  para todas as soluções  $i$ . Ou seja:

$$\text{Hiper-volume}(\mathbf{S}, R) = \text{volume}\left(\bigcup_{i=1}^{|\mathbf{S}|} v_i\right).$$

A Figura 21 apresenta um conjunto de soluções não dominadas arbitrário  $\mathbf{S} = \{A, B, C\}$ , considerando um problema com dois objetivos,  $f_1$  e  $f_2$ , a serem minimizados. Nesse caso,  $R$  é um ponto com valores que superam os demais em ambos os objetivos, ou seja, dominado por todas as soluções. A área cinza indica o espaço de busca ou espaço das soluções factíveis e a curva preta é a fronteira de Pareto-ótimo. Por haver apenas duas funções objetivos, o hipercubo formado por cada solução  $i$  é um retângulo e o valor do hipervolume é a área do espaço formado pela união desses retângulos, representada pelo polígono pontilhado.

Figura 21 – Hipervolume de um conjunto de soluções arbitrário



Fonte: Adaptado de Jiang et al. (2014)

A métrica  $\Delta$  (Delta), proposta por Deb et al. (2000) mede a diversidade das soluções não dominadas em uma fronteira gerada por um determinado algoritmo multiobjetivo. Essa diversidade avalia o espalhamento e a distribuição das soluções nessa fronteira, sendo que fronteiras cujas soluções extremas estão mais distantes e cuja distribuição das soluções intermediárias tende mais à distribuição uniforme recebem valores menores para a métrica  $\Delta$ . Geralmente, fronteiras de soluções com valores menores para essa métrica, e portanto mais diversificadas, são desejadas.

Para determinar o valor dessa métrica para um conjunto de soluções, deve-se ordenar essas soluções considerando alguma função objetivo. Assim, define-se  $d_i$  como a distância Euclidiana entre as soluções consecutivas de  $\mathbf{S}$  e  $\bar{d}$ , a média dos valores de  $d_i, \forall i \in \{1, \dots, |\mathbf{S}| - 1\}$ . Então, a métrica é definida como (DEB et al., 2000):

$$\Delta(\mathbf{s}) = \sum_{i=1}^{|\mathbf{s}|-1} \frac{(d_i - \bar{d})}{|\mathbf{s}| - 1}.$$

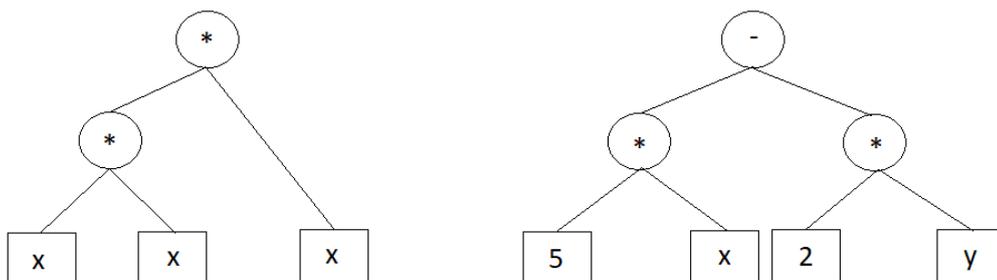
Nota-se que um conjunto de soluções com uma métrica  $\Delta$  igual a zero possui um espaçamento entre todas as soluções consecutivas exatamente iguais.

### 3.5 Programação Genética

De acordo com [Koza et al. \(2005\)](#), Programação Genética é um método que gera uma população de programas de computador e a transforma aplicando operações análogas ao processo de seleção natural, como reprodução, mutação, replicação e remoção de genes, com o objetivo de melhorar esses programas de computador para a solução de algum problema do mundo real. Assim como o Algoritmo Genético, a Programação Genética faz parte da área de Computação Evolucionária e, por isso, compartilham muitas características.

Os programas (ou indivíduos) são representados em um formato de árvore, compostos por nós terminais ou folhas interligados por funções ou operações (sendo muitas vezes operações aritméticas), formando assim, várias ramificações. Dois exemplos dessa estrutura são ilustrados na Figura 22, sendo que o indivíduo da esquerda é a árvore que representa a expressão aritmética  $x^3$  e a da direita, representa a expressão  $5x - 2y$ .

Figura 22 – Exemplos de indivíduos em Programação Genética



Para a execução do método, os passos preparatórios são a identificação dos nós terminais (que correspondem às variáveis independentes do problema, constantes aleatórias e as funções sem argumento) de cada ramificação do programa a ser gerado, das funções primitivas ou operações das ramificações, a definição da medida de aptidão (*fitness*), que representa um valor que mede o desempenho de um programa em relação aos demais, hiperparâmetros para a execução da programação genética e a definição do critério de parada para a execução ([KOZA et al., 2005](#)).

Após os passos preparatórios, a população inicial é formada aleatoriamente composta pelos nós terminais e operações definidas. Essa população inicial é modificada

iterativamente seguindo os passos: avaliação da função aptidão (*fitness*), seleção, cruzamento, reprodução e mutação e seleção. Na seleção, dois indivíduos são selecionados (chamados de pais), com uma probabilidade proporcional ao valor de sua função aptidão. A recombinação cria dois novos programas (chamados de filhos) recombinao aleatoriamente partes dos dois programas selecionados no passo anterior. A mutação cria, com uma probabilidade pequena, uma perturbação ou pequena mudança nos programas gerados pelo passo anterior. E, por fim, o passo de operação de alteração de arquitetura seleciona uma operação de alteração de arquitetura dentre as operações disponíveis e a aplica ao melhor indivíduo da população, que fará parte da população inicial da nova iteração, juntamente aos programas filhos gerados nessa iteração ou geração.

## 4 Metodologia

O arcabouço computacional proposto no trabalho e os experimentos provenientes de suas execuções são compostos por três partes distintas: a otimização de portfólio, a classificação de séries temporais financeiras e as combinações das duas partes. Esse capítulo tratará separadamente cada uma dessas partes, além das especificações dos dados coletados para a alimentação deste sistema.

### 4.1 Dados de entrada

Os dados de entrada são compostos por séries históricas de informações sobre ativos financeiros que participaram da B3, Bolsa de Valores brasileira, durante o período de janeiro de 2008 a dezembro de 2017. São selecionados, para isso, os 47 ativos com maiores quantidades de negociações nesse período, listados no Quadro 2. A coleta dos dados foi feita utilizando a plataforma Bloomberg®.

Os valores são coletados já com ajustes para desdobramentos e agrupamentos. Assim, empresas resultantes do desdobramento de uma empresa em duas passam a ter seus valores de ativos, em períodos anteriores ao desdobramento, igual à metade dos valores dos ativos da empresa original. Já as empresas resultantes de agrupamento de duas outras empresas, possuem valores de ativo igual à soma das duas empresas originais, para períodos anteriores ao agrupamento. Os dados também possuem ajustes de dividendos e juros.

Dois conjuntos de dados são utilizados, sendo um composto por séries históricas de granularidade diária (denominado conjunto de dados 1), de forma que as informações são coletadas em cada dia de funcionamento da B3 e outro composto de séries históricas com granularidade horária (conjunto de dados 2), de forma que as informações são coletadas 8 vezes por dia, correspondendo ao tempo de funcionamento da Bolsa de Valores de 8 horas diárias. Dessa forma, um total de 2474 períodos de dados são considerados para o conjunto 1, e 19792 períodos, para o conjunto 2.

As informações utilizadas para cada ativo são o preço de abertura, preço de fechamento, preço máximo, preço mínimo e volume de negociação. Para períodos em que não houve negociação para um ativo, são utilizados os valores da última negociação daquele ativo. Assim, as informações podem ser representadas por conjuntos tridimensionais, sendo que suas dimensões são os ativos financeiros, os períodos de coleta de dados e as informações de preço e volume dos ativos. O conjunto de dados 1, portanto, é representado por um conjunto de tamanho  $47 \times 2474 \times 5$  e o conjunto 2, por um conjunto de tamanho  $47 \times 19792 \times 5$ .

Quadro 2 – Ativos selecionados

<b>Código</b>	<b>Empresa</b>
ABEV3	Ambev
BBAS3	Banco do Brasil
BBDC3	Banco Bradesco
BBDC4	Banco Bradesco
BRAP4	Bradespar
BRFS3	BRF
BRKM5	Braskem
BRML3	BRMalls
CCRO3	CCR
CIEL3	Cielo
CMIG4	Companhia Energética de Minas Gerais
CPFE3	CPFL Energia
CSAN3	Cosan
CSNA3	Companhia Siderúrgica Nacional
CYRE3	Cyrela Brazil Realty
ECOR3	EcoRodovias
EMBR3	Embraer
ENBR3	EDP Brasil
EQTL3	Equatorial Energia
ESTC3	Estácio Participações
FIBR3	Fibria Celulose
GGBR4	Gerdau
GOAU4	Metalúrgica Gerdau
HYPE3	Hypera Pharma

<b>Código</b>	<b>Empresa</b>
ITSA4	Itaúsa
ITUB4	Itaú Unibanco
JBSS3	JBS
KROT3	Kroton Educacional
LAME4	Lojas Americanas
LREN3	Lojas Renner
MRFG3	Marfrig
MRVE3	MRV Engenharia
MULT3	Multiplan
NATU3	Natura
PCAR4	Companhia Brasileira de Distribuição
PETR3	Petróleo Brasileiro
PETR4	Petróleo Brasileiro
RADL3	RaiaDrogasil
RENT3	Localiza Hertz
SANB11	Banco Santander Brasil
SBSP3	Companhia de Saneamento Básico do Estado de São Paulo
TIMP3	TIM Brasil
UGPA3	Grupo Ultra
USIM5	Usiminas
VALE3	Vale S.A.
VIVT4	Telefônica Brasil
WEGE3	WEG S.A.

## 4.2 Otimização de portfólio

A otimização de portfólio é a parte do arcabouço computacional, composto por modelos e métodos para a solução desses modelos, que recebe como entrada as séries históricas de retornos e valores dos preços de fechamento dos ativos e proporciona, como saída, um conjunto de portfólios eficientes para um dado período.

Em lugar do retorno linear, apresentado na subseção 3.1.1, este trabalho utiliza o log retorno para os modelos de otimização. O log retorno  $r_{it}$ , do ativo  $i$  em um período  $t$ , é definido, a partir da série de preços de fechamento de  $i$ , como:

$$r_{it} = \ln \left( \frac{p_{it}}{p_{i,t-1}} \right) = \ln(p_{it}) - \ln(p_{i,t-1}),$$

em que  $p_{it}$  é o preço de fechamento do ativo  $i$  no período  $t$  e  $\ln(\cdot)$  é a função logaritmo natural.

O log retorno é uma transformação que suaviza ganhos e perdas discrepantes nos valores de mercado dos ativos. Outra vantagem de seu uso é o fato de que o log retorno acumulado, de um período  $t$  a um período  $t + \delta$ , corresponde ao somatório dos log retornos desse intervalo, enquanto que para encontrar o retorno linear acumulado, um produtório envolvendo os retornos lineares consecutivos deve ser aplicado.

#### 4.2.1 Modelos de otimização

O trabalho propõe uma melhoria do modelo de [Ferreira \(2018\)](#) descrito na Subseção 3.1.6, substituindo a restrição de cardinalidade fixa (equação 14b) por uma restrição de cardinalidade variável, de forma que os portfólios podem ser formados por um número de ativos entre um valor  $k_{min}$  e um valor  $k_{max}$ . Utilizar a restrição de cardinalidade variável, em lugar da cardinalidade fixa, dispensa a execução exaustiva da otimização para vários valores de cardinalidade considerados desejáveis. É importante observar que, de acordo com a expressão 14d, os ativos que formam um portfólio são aqueles que possuem um investimento igual ou maior a um lote mínimo de suas ações.

Outra mudança do modelo proposto, em relação ao modelo de [Ferreira \(2018\)](#), é a adição de uma restrição de ativos obrigatórios, que obriga a inclusão de alguns ativos no portfólio. Esta restrição ganha importância em algumas formas de combinação de otimização de portfólios e classificação de ativos propostos, conforme são descritos mais adiante, na seção 4.4.

O modelo proposto neste trabalho é denominado Modelo 1. Outros três modelos, descritos nessa subseção, são acrescentados ao arcabouço computacional e utilizados em experimentos propostos.

##### 4.2.1.1 Modelo 1

O modelo proposto substitui a expressão 14b pelas inequações  $\sum_{i=1}^n z_i \geq k_{min}$  e  $\sum_{i=1}^n z_i \leq k_{max}$ , permitindo um intervalo de valores para a cardinalidade do portfólio. A restrição de ativos obrigatórios garante a inclusão de  $Q$  (com  $Q \leq k_{max}$ ) ativos no portfólio, cada um dos  $j$  ativos ( $j = 1, \dots, Q$ ), com uma quantidade mínima  $q_j$  de lotes investidos, de forma que  $x_j \geq q_j$ . Este modelo proposto pode ser descrito da seguinte forma:

$$\min_{x_1, \dots, x_n} \quad \zeta + (1 - \alpha)^{-1} \sum_{t=1}^T p_t [-r_{\mathbf{x}t} - \zeta]^+ \\ \max_{x_1, \dots, x_n} \quad \sum_{i=1}^n w_i \mu_i$$

$$\text{sujeito a: } \left\{ \begin{array}{l} w_i = \frac{m_i c_i x_i}{\sum_{j=1}^n m_j c_j x_j}, \forall i, i = 1, \dots, n \\ x_j \geq q_j, \forall j, j = 1, \dots, Q \\ k_{min} \leq \sum_{i=1}^n z_i \leq k_{max} \\ \sum_{i=1}^n m_i c_i [(x_i - x_i^{(0)}) + \beta |x_i - x_i^{(0)}|] + f v \leq C \\ z_i = \begin{cases} 0, & \text{se } x_i = 0 \\ 1, & \text{caso contrário} \end{cases}, \forall i, i = 1, \dots, n \\ v = \begin{cases} 0, & \text{se } \sum_{i=1}^n |x_i - x_i^{(0)}| = 0 \\ 1, & \text{caso contrário} \end{cases} \\ x_i \in \mathbb{N}, \forall i, i = 1, \dots, n \end{array} \right.$$

#### 4.2.1.2 Modelo 2

O Modelo 2 se difere do modelo proposto (Modelo 1) por representar um portfólio  $\mathbf{w} = w_1, w_2, \dots, w_n$  pelos pesos  $w_i$  dos ativos disponíveis, sendo permitido a um portfólio possuir valores que não são múltiplos do número de lote mínimo de um ativo investidos neste. Esses pesos ou proporções são valores não negativos, cuja soma é igual a 1. Os ativos que compõem o portfólio devem possuir proporções em um intervalo definido por valores mínimo ( $w_{min}$ ) e máximo ( $w_{max}$ ). As restrições de cardinalidade variável, de ativos obrigatórios e de capital máximo, com rebalanceamento e custos de transação, se mantêm. Dessa forma, o Modelo 2 pode ser representado como:

$$\begin{aligned} \min_{w_1, \dots, w_n} \quad & \zeta + (1 - \alpha)^{-1} \sum_{t=1}^T p_t [-r \mathbf{w} t - \zeta]^+ \\ \max_{w_1, \dots, w_n} \quad & \sum_{i=1}^n w_i \mu_i \end{aligned}$$

$$\text{sujeito a: } \left\{ \begin{array}{ll} w_j \geq q'_j, \forall j, j = 1, \dots, Q & (19a) \\ k_{min} \leq \sum_{i=1}^n z_i \leq k_{max} & (19b) \\ C \sum_{i=1}^n (w_i - w_i^{(0)} + \beta |w_i - w_i^{(0)}|) + fv = C & (19c) \\ z_i = \begin{cases} 0, & \text{se } w_i = 0 \\ 1, & \text{caso contrário} \end{cases}, \forall i, i = 1, \dots, n & (19d) \\ v = \begin{cases} 0, & \text{se } \sum_{i=1}^n |w_i - w_i^{(0)}| = 0 \\ 1, & \text{caso contrário} \end{cases} & (19e) \\ w_i \geq w_{min} z_i, \forall i, i = 1, \dots, n & (19f) \\ w_i \leq w_{max}, \forall i, i = 1, \dots, n & (19g) \\ w_i \geq 0, \forall i, i = 1, \dots, n & (19h) \end{array} \right.$$

O Modelo 2, por utilizar as próprias proporções de investimento como variáveis de decisão, não necessita da equação 14a, que transforma quantidades de lotes em proporções de investimento. A expressão 19h garante que as proporções dos ativos em um portfólio são valores reais não negativos. A restrição 19a, de ativos obrigatórios, substitui os valores mínimos de lotes  $q_j$  por valores mínimos de proporção  $q'_j$ , com  $j = 1, \dots, Q$ , sendo que  $q'_j$  deve ser um valor entre  $w_{min}$  e  $w_{max}$ , limites mínimo e máximo descritos em 19f e 19g. A restrição de cardinalidade variável é representada da mesma forma que no Modelo 1, com variáveis binárias  $z_i$ . As expressões 19c e 19e compõem a restrição de capital máximo, com rebalanceamento e custos de transação. Na expressão 19c, a multiplicação das proporções dos ativos pelo capital disponível  $C$  permite que o custo de transação fixo  $f$  possa ser dado na unidade monetária de  $C$ . Com a utilização de variáveis reais ( $w_i$ ) e inteiras ( $z_i$ ), o Modelo 2 é um modelo de otimização multiobjetivo inteira mista não linear.

#### 4.2.1.3 Modelo 3

O Modelo 3 é um modelo linear aproximado do Modelo 1. A utilização de um modelo linear se deve à maior eficiência, em termos de custos computacionais, dos métodos de programação lineares. Para isso, é utilizada a forma linear do risco CVaR e as expressões 14a, 14c, 14d e 14e são substituídas por expressões lineares. Para a utilização de métodos de programação lineares, o modelo multiobjetivo proposto (Modelo 1) é transformado em um modelo monobjetivo de minimização do risco, com a restrição de retorno esperado mínimo, de forma que o Modelo 3 é um modelo monobjetivo de otimização linear inteira.

Na equação 14a, que define as proporções de investimento, o somatório dos custos de todos os ativos ( $m_i c_i x_i$  representa o custo do ativo  $i$ ) é substituído pelo capital total  $C'$ ,

que representa a soma de  $C$  com o custo total do portfólio prévio. Por isso, define-se um limite mínimo de investimento  $\rho C$ , em que  $\rho$  é um fator entre 0 e 1, que multiplicado por  $C$  define um limite mínimo para investimento, enquanto que  $C$  é o limite máximo. Quanto mais próximo  $\rho$  é de 1, mais preciso se torna o valor aproximado das proporções  $w_i$ .

Na expressão 14c, que define o limite máximo dos custos dos ativos do portfólio, o módulo da diferença entre as quantidades de lotes do novo portfólio e do portfólio prévio ( $|x_i - x_i^{(0)}|$ ) é substituído por uma variável  $d_i$ , que possui as restrições de ser maior que  $x_i - x_i^{(0)}$  e  $x_i^{(0)} - x_i$ , sendo portanto, maior que o máximo das duas diferenças. Por fim, uma outra restrição limita o valor de  $d_i$  em  $x_i + x_i^{(0)}$ .

As variáveis binárias  $z_i$ , utilizadas na restrição de cardinalidade variável, são originalmente definidas por uma função condicional, de forma que  $z_i$  vale 1 se o ativo  $i$  faz parte do portfólio ótimo e 0, caso contrário, conforme representado na expressão 14d. Essa função é substituída por três restrições lineares. A primeira define o domínio de  $z_i$  como sendo os valores 0 e 1. A segunda restrição garante que  $z_i$  é sempre menor ou igual à quantidade de lotes  $x_i$ , de forma que  $z_i$  será 0 quando  $x_i$  for 0. A terceira garante que  $z_i$  será 1 quando  $x_i$  for maior que 0, ou seja, maior ou igual a 1, já que a quantidade de lotes é um valor inteiro e para isso, basta fazer com que  $z_i$  seja maior ou igual a  $x_i$  multiplicado por um fator  $l$  suficientemente pequeno, de forma que  $x_i l$  nunca seja maior que 1. As restrições, portanto, são:

$$z_i \in \{0,1\}, \forall i, i = 1, \dots, n$$

$$z_i \leq x_i, \forall i, i = 1, \dots, n$$

$$z_i \geq x_i l, \forall i, i = 1, \dots, n.$$

Da mesma forma que  $z_i$  é definida por restrições lineares, a variável binária  $v$ , que indica se o portfólio ótimo (novo) é diferente do portfólio prévio, também é definida por três restrições que substituem a expressão 14e:

$$v \in \{0,1\}$$

$$v \leq \sum_{i=1}^n d_i$$

$$z_i \geq l_2 \sum_{i=1}^n d_i,$$

tal que  $l_2$  é um valor suficientemente pequeno que, multiplicado pelo somatório de todos  $d_i$ , garante que o resultado não seja maior que 1. Assim, o Modelo 3 é apresentado como:

$$\min_{\mathbf{x}, \mathbf{a}, \zeta, \mathbf{d}} \quad \zeta + \frac{1}{1 - \alpha} \sum_{t=1}^T p_t a_t \quad (20)$$

$$\begin{aligned}
 & a_t \geq 0, \forall t, t = 1, \dots, T & (21a) \\
 & a_t \geq -r_{\mathbf{x}t} - \zeta, \forall t, t = 1, \dots, T & (21b) \\
 & \sum_{i=1}^n w_i \mu_i \geq r_{min} & (21c) \\
 & x_j \geq q_j, \forall j, j = 1, \dots, Q & (21d) \\
 & \sum_{i=1}^n z_i \leq k_{max} & (21e) \\
 & \sum_{i=1}^n z_i \geq k_{min} & (21f) \\
 & \sum_{i=1}^n m_i c_i (x_i - x_i^{(0)} + \beta d_i) + f v \leq C & (21g) \\
 & \sum_{i=1}^n m_i c_i (x_i - x_i^{(0)} + \beta d_i) + f v \geq \rho C & (21h) \\
 & C' = C + \sum_{i=1}^n m_i c_i x_i^{(0)} & (21i) \\
 & d_i \geq x_i - x_i^{(0)}, \forall i, i = 1, \dots, n & (21j) \\
 & d_i \geq x_i^{(0)} - x_i, \forall i, i = 1, \dots, n & (21k) \\
 & d_i \leq x_i^{(0)} + x_i, \forall i, i = 1, \dots, n & (21l) \\
 & z_i \leq x_i, \forall i, i = 1, \dots, n & (21m) \\
 & z_i \geq x_i (L/C'), \forall i, i = 1, \dots, n & (21n) \\
 & x_i \in \mathbb{N}, \forall i, i = 1, \dots, n & (21o) \\
 & z_i \in \{0, 1\}, \forall i, i = 1, \dots, n & (21p) \\
 & v \leq \sum_{i=1}^n d_i & (21q) \\
 & v \geq \sum_{i=1}^n d_i (\beta/C') & (21r) \\
 & v \in \{0, 1\} & (21s) \\
 & w_i = \frac{m_i c_i x_i}{C'}, \forall i, i = 1, \dots, n & (21t)
 \end{aligned}$$

sujeito a:

A função objetivo descrita na expressão 20, junto com as restrições 21a e 21b proporcionam uma forma linear para a minimização do risco CVaR. A restrição de retorno mínimo (21c) considera um valor mínimo  $r_{min}$  de retorno esperado para o portfólio e a restrição 21d, de ativos obrigatórios não se modifica. Neste modelo, a restrição de cardinalidade variável é representada pelas expressões 21e, 21f, 21m, 21n e 21p. A variável  $d_i$  é descrita pelas expressões 21j, 21k e 21l,  $v$  é representado pelas expressões 21q, 21r e 21s e a expressão 21t define a proporção (aproximada) dos ativos no portfólio. As restrições

21g e 21h estabelecem, respectivamente, os limites máximos e mínimos para os custos totais do portfólio, considerando os custos de transação.

O valor  $l$ , que garante que  $lx_i$  não exceda o valor 1, é substituído por  $(L/C')$ , em que  $L$ , na prática, é um valor menor ou igual ao menor preço de fechamento no período  $T$ , considerando todos os ativos disponíveis. Para  $l_2$ , que faz com que  $l_2 \sum_{i=1}^n d_i$  não exceda 1, é utilizado o valor  $(\beta/C')$ .

#### 4.2.1.4 Modelo 4

O Modelo 4, assim como o Modelo 2, utiliza as proporções de investimento dos ativos como variáveis reais de decisão e, assim como o Modelo 3, é um modelo monobjetivo linear. Assim, este é um modelo monobjetivo de otimização linear inteira mista.

O Modelo 4 possui as mesmas restrições de valores mínimos e máximos para proporções dos ativos que compõem o portfólio do Modelo 2, e a restrição de ativos obrigatórios também permanece inalterada. Para a restrição de cardinalidade variável, a variável binária  $z_i$  é definida de forma parecida com o Modelo 3. Porém, aproveita-se do fato de que existe um valor mínimo para a proporção dos ativos do portfólio e a restrição  $z_i \leq w_i + (1 - w_{min})$  é utilizada como garantia de que  $z_i$  vale 0 se  $w_i$  também vale 0, e a restrição  $z_i \geq w_i$  garante que  $z_i$  vale 1 quando  $w_i$  possui um valor diferente (maior) de 0, considerando o domínio de  $z_i$  como os valores 0 e 1 ( $z_i \in \{0,1\}$ ).

Para a restrição de capital máximo de investimento, o módulo da diferença de  $w_i$  e  $w_i^{(0)}$ , no Modelo 2, é substituído pela variável  $d_i$ , definida de forma similar ao Modelo 3, mas considerando as variáveis  $w_i$  e  $w_i^{(0)}$  em lugar de  $x_i$  e  $x_i^{(0)}$ . Já a variável binária  $v$  é definida, de forma similar a  $d_i$ , pelas restrições:

$$\begin{aligned} v &\geq \frac{1}{n} \sum_{i=1}^n d_i \\ v &\leq \sum_{i=1}^n d_i + (1 - w_{min}) \\ v &\in \{0,1\}. \end{aligned}$$

Assim, o Modelo 4 é representado da seguinte forma:

$$\min_{\mathbf{w}, \mathbf{a}, \zeta, \mathbf{d}} \quad \zeta + \frac{1}{1 - \alpha} \sum_{t=1}^T p_t a_t$$

$$\begin{aligned}
 & a_t \geq 0, \forall t, t = 1, \dots, T & (22a) \\
 & a_t \geq -r_{\mathbf{wt}} - \zeta, \forall t, t = 1, \dots, T & (22b) \\
 & \sum_{i=1}^n w_i \mu_i \geq r_{min} & (22c) \\
 & w_j \geq q'_j, \forall j, j = 1, \dots, Q & (22d) \\
 & \sum_{i=1}^n z_i \leq k_{max} & (22e) \\
 & \sum_{i=1}^n z_i \geq k_{min} & (22f) \\
 & C \sum_{i=1}^n (w_i - w_i^{(0)} + \beta d_i) + fv = C & (22g) \\
 & d_i \geq w_i - w_i^{(0)}, \forall i, i = 1, \dots, n & (22h) \\
 & d_i \geq w_i^{(0)} - w_i, \forall i, i = 1, \dots, n & (22i) \\
 & d_i \leq w_i^{(0)} + w_i, \forall i, i = 1, \dots, n & (22j) \\
 & z_i \geq w_i, \forall i, i = 1, \dots, n & (22k) \\
 & z_i \leq w_i + (1 - w_{min}), \forall i, i = 1, \dots, n & (22l) \\
 & z_i \in \{0, 1\}, \forall i, i = 1, \dots, n & (22m) \\
 & v \geq \frac{1}{n} \sum_{i=1}^n d_i & (22n) \\
 & v \leq \sum_{i=1}^n d_i + (1 - w_{min}) & (22o) \\
 & v \in \{0, 1\} & (22p) \\
 & w_i \leq w_{max}, \forall i, i = 1, \dots, n & (22q) \\
 & w_i \geq w_{min} z_i, \forall i, i = 1, \dots, n & (22r) \\
 & w_i \geq 0, \forall i, i = 1, \dots, n & (22s)
 \end{aligned}$$

sujeito a: {

## 4.2.2 Métodos de otimização

Os dois primeiros modelos, que são multiobjetivos e não lineares, são solucionados por algoritmos evolutivos multiobjetivos e os dois últimos, pelo algoritmo de programação linear inteira e inteira mista *branch-and-cut*. Os algoritmos evolutivos são baseados no NSGA-II proposto por [Deb et al. \(2011\)](#).

### 4.2.2.1 Método 1

O Método 1 é o algoritmo proposto para solucionar o Modelo 1, baseado no NSGA-II de [Deb et al. \(2011\)](#) descrito na Subseção 3.4.4.1. Esse algoritmo proposto utiliza o mesmo processo de geração da população inicial e mesmos operadores de cruzamento e mutação

propostos por [Deb et al. \(2011\)](#), com a diferença de aplicar a função de arredondamento para o valor inteiro mais próximo nos resultados dos operadores, já que no Método 1 os cromossomos são formados por valores inteiros, que representam as quantidades de lotes investidas em cada ativo. Os limites  $w_{min}$  e  $w_{max}$ , para os valores de cada gene, também são alterados para 1 e  $\frac{C'}{Mm_i k}$ , em que  $m_i$  e  $k_j$  são respectivamente o número mínimo de lotes do ativo  $i$  e a cardinalidade do indivíduo sobre o qual o operador será aplicado.  $M$  é o valor médio de preço dos ativos no período  $T$  e o valor  $C'$  é a soma do capital disponível  $C$  com o custo total do portfólio prévio  $\mathbf{x}^{(0)}$ .

Enquanto a aplicação do arredondamento dos valores reais garante que os valores dos genes no Método 1 serão sempre inteiros, um método de reparo é proposto para garantir que a restrição de capital máximo de investimento não seja violada e que o custo total do portfólio  $\mathbf{x}$ , incluindo os custos de transação, esteja próximo do capital total  $C'$ . Esse método de reparo é aplicado após a geração da população inicial, após o operador de cruzamento e após os operadores de mutação, substituindo o método de reparo de [Deb et al. \(2011\)](#).

O método de reparo proposto para o Método 1 consiste em uma heurística gulosa que considera dois casos diferentes. No caso 1, um operador faz com que o indivíduo alterado por esse operador possua custo total superior ao capital disponível, violando a restrição [14c](#). Nesse caso, é aplicado o algoritmo de reparo para o caso 1 (Algoritmo [2](#)), que consiste em encontrar, aleatoriamente, um ativo  $j$  que possua mais que 1 lote de ações investidas e reduzir essa quantidade em uma unidade. O procedimento é repetido até que o custo desse indivíduo ou portfólio esteja abaixo de  $C'$ , de forma a não violar a restrição de capital máximo de investimento.

O caso 2 ocorre quando não há a violação da restrição [14c](#), de forma que o custo total do portfólio alterado seja menor que o capital máximo disponível. Nesse caso, o objetivo é aproximar o custo total do valor máximo, sem que haja violação da restrição [14c](#). Para isso, é utilizado o Algoritmo [3](#), que seleciona, aleatoriamente, um ativo do portfólio alterado e, caso o incremento de um lote desse ativo não resulte em uma violação da restrição de capital máximo disponível, esse incremento é realizado. O procedimento é realizado iterativamente até que seja selecionado um ativo cujo incremento em sua quantidade de lotes resulte em violação da restrição.

Um procedimento adicional é proposto para o Método 1, realizada após a etapa de seleção do NSGA-II, priorizando indivíduos que possuem baixo valor de cardinalidade. Para isso, uma proporção  $p_s$  dos indivíduos dominados (que não fazem parte da primeira fronteira ordenada) são substituídos por indivíduos com cardinalidade definida da seguinte forma:

$$k' = k_{min} + (1 - u^{0,1}) \times (k_{max} - k_{min}),$$

**Algoritmo 2:** Algoritmo de reparo para o caso 1

---

**Input:** Cromossomo do indivíduo alterado  $j$   
**Output:** Cromossomo atualizados de  $j$

```

1  $custo \leftarrow$  custo total de  $j$ 
2  $i \leftarrow$  aleatório( $1, k_j$ )
3 while  $custo > C'$  do
4   while  $x_i == 1$  do
5      $i \leftarrow$  aleatório( $1, k_j$ )
6   end
7    $x_i \leftarrow x_i - 1$ 
8    $custo \leftarrow$  custo total de  $j$ 
9 end

```

---

**Algoritmo 3:** Algoritmo de reparo para o caso 2

---

**Input:** Cromossomo do indivíduo alterado  $j$   
**Output:** Cromossomo atualizados de  $j$

```

1  $custo \leftarrow$  custo total de  $j$ 
2  $i \leftarrow$  aleatório( $1, k_j$ )
3 while  $custo + x_i c_i m_i < C'$  do
4    $x_i \leftarrow x_i + 1$ 
5    $custo \leftarrow$  custo total de  $j$ 
6    $i \leftarrow$  aleatório( $1, k_j$ )
7 end

```

---

em que  $u \in [0,1]$  é um valor definido aleatoriamente e a cardinalidade  $k$  de cada indivíduo novo gerado é determinado pelo arredondamento de  $k'$  ao valor inteiro mais próximo. O propósito desse procedimento adicional é adicionar mais ativos de menores cardinalidades, já que o cruzamento proposto por [Deb et al. \(2011\)](#) produz sempre filhos com cardinalidades intermediárias em relação às cardinalidades dos pais, de forma que indivíduos com valores extremos de cardinalidade tendem a ser eliminados rapidamente da população. No problema de otimização de portfólio, os indivíduos de menores cardinalidades são mais facilmente descartados, já que possuem maiores riscos de acordo com a teoria de [Markowitz \(1952\)](#) e possuem menores chances de selecionar ativos de maiores retornos médios, por serem compostos de poucos ativos.

Por fim, o Método 1 é um algoritmo paralelo que segue o modelo de paralelismo proposto em [Ferreira \(2018\)](#). Esta forma de paralelismo combina o modelo em ilhas, em que os indivíduos da população do NSGA-II são agrupados, em conjuntos de mesmo tamanho de diferentes indivíduos, denominados ilhas, com a ordenação paralela das funções objetivos (risco e retorno esperado) de todos os indivíduos, utilizando a unidade de processamento gráfico (ou GPU - *Graphics Processing Unit*).

Neste trabalho, as ilhas são processadas em ambientes *multithreads* em que o número de *threads* utilizadas  $N_t$  é igual ao número determinado de ilhas, sendo que cada

ilha possui  $N/N_t$  indivíduos. Em casos em que  $N$  não é divisível por  $N_t$  cada ilha é inicialmente composta por  $\lfloor N/N_t \rfloor$  (em que  $\lfloor \cdot \rfloor$  é a função piso) indivíduos e  $N - \lfloor N/N_t \rfloor$  ilhas, selecionadas aleatoriamente, recebem um indivíduo a mais. Essas ilhas são combinadas de forma a criar uma fronteira não dominada composta por no máximo  $\lfloor N/N_t \rfloor$  indivíduos, posteriormente compartilhada para todas as ilhas, a cada  $N_g$  gerações. Fronteiras que excedem o valor máximo são reduzidas, selecionando  $\lfloor N/N_t \rfloor$  indivíduos bem espaçados, correspondendo aos centróides do algoritmo de agrupamento *K-means* (algoritmo descrito em Kaufman e Rousseeuw (1990)). Quando o tamanho da fronteira combinada é menor que o número de indivíduos de uma ilha, os indivíduos a permanecer na ilha são escolhidos aleatoriamente.

#### 4.2.2.2 Método 2

O Método 2 é o algoritmo proposto para a solução do Modelo 2. Este método é muito similar ao NSGA-II de Deb et al. (2011), já que o Modelo 2 também utiliza proporções de investimentos dos ativos como variáveis de decisão. Dessa forma a geração da população inicial e os operadores de mutação e cruzamento são aqueles propostos por Deb et al. (2011). O método de reparo utiliza a mesma função de mapeamento proposta por Deb et al. (2011), mas considera a restrição de que  $\sum_{i=1}^n (w_i - w_i^{(0)} + \beta |w_i - w_i^{(0)}|) + fv/C$  deve ser arbitrariamente próximo de 1. O Método 2 também utiliza o procedimento adicional, descrito no Método 1, após a seleção e o método paralelo de execução, também descrito no Método 1.

#### 4.2.2.3 Método 2B

Para uma comparação com os métodos que utilizam quantidades de lotes como variáveis de decisão e, levando em conta que ativos são negociados em lotes na prática, no Método 2B, as proporções de investimento de cada uma das  $N$  soluções finais do Método 2 são transformadas em quantidades de lotes da seguinte forma, para cada ativo  $i$ :

$$x_i = \left\lfloor \frac{w_i C'}{m_i C_i} \right\rfloor.$$

#### 4.2.2.4 Método 3

O Método 3 corresponde a múltiplas execuções do *branch-and-cut* aplicado ao problema do Modelo 3, gerando uma fronteira de soluções não dominadas similar àquelas formadas pelos Métodos 1 e 2. Para isso, primeiro são encontradas as soluções de risco mínimo e retorno máximo, considerando as demais restrições do Modelo 3. Posteriormente,  $N$  valores igualmente espaçados entre o retorno esperado para o portfólio de risco mínimo e o retorno esperado para o portfólio de retorno máximo são iterativamente atribuídos a  $r_{min}$  na restrição 21c e solucionados pelo algoritmo *branch-and-cut*. Os  $N$  portfólios ou

soluções resultantes compõem a fronteira eficiente. O *branch-and-cut*, neste trabalho, utiliza o algoritmo Simplex para encontrar as soluções das relaxações lineares dos problemas derivados do problema proposto.

#### 4.2.2.5 Método 4

O Método 4 é constituído dos mesmos procedimentos do Método 3, considerando o problema descrito pelo Modelo 4. Apesar dos Métodos 3 e 4 corresponderem aos mesmos procedimentos aplicados em diferentes modelos, este trabalho separa os dois métodos por motivos de facilitação na apresentação dos resultados.

#### 4.2.2.6 Método 4B

Uma vez que o Modelo 4 utiliza proporções de investimentos como variável de decisão, o Método 4B transforma as proporções das soluções do Método 4 em quantidades de ativos, da mesma forma com que é realizada no Método 2B.

### 4.2.3 Experimentos

Os experimentos realizados para os métodos de otimização de portfólio são divididos em experimentos *in-sample* e *out-of-sample*. Os experimentos *in-sample* avaliam as fronteiras de soluções não dominadas proporcionadas por cada método, considerando as métricas  $\Delta$  e hipervolume, além de comparar o desempenho dos métodos em relação ao tempo de execução de cada um. Os experimentos *out-of-sample* comparam o desempenho de modelos e métodos em simulações de negociações no mercado financeiro, considerando os retornos, riscos e *drawdowns* mensais, além do retorno acumulado durante um determinado período, para cada configuração de modelo e método associado.

#### 4.2.4 Experimentos *in-sample*

Experimentos *in-sample* consideram os dados do conjunto de dados 1 (com granularidade diária) e a série histórica de entrada dos métodos possuem um tamanho de 247 períodos, compreendendo o ano de 2013.

Um primeiro experimento compara o algoritmo de [Deb et al. \(2011\)](#) com o Método 2, que aplica algumas alterações sobre o primeiro algoritmo para a solução de modelos similares. Posteriormente, os seis métodos propostos são comparados entre si e o melhor método é selecionado, sendo que os Métodos 2 e 4 não podem ser selecionados já que não possuem aplicação prática direta, ao contrário de suas transformações (Métodos 2B e 4B).

Nesses experimentos, o tempo de execução é uma característica importante porque determina a frequência máxima de operações no mercado. Comparando métodos com tempos de execução de mesma ordem de grandeza, a métrica hipervolume é considerada

mais importante por levar em consideração o espalhamento das soluções na fronteira eficiente e, principalmente, a cobertura dessa fronteira. A métrica  $\Delta$  pode comparar métodos que possuem valores próximos de hipervolume e pode ser utilizada para determinar se um maior valor de hipervolume para um método está associado ao espalhamento ou à cobertura de suas soluções.

Por fim, as soluções nas fronteiras eficientes proporcionadas pelos métodos propostos são analisadas em relação à posição dessas de acordo com suas cardinalidades. É esperado, de acordo com a teoria de [Markowitz \(1952\)](#), que soluções de maior cardinalidade possuam menor risco e retorno esperado, enquanto que soluções de menor cardinalidade possuam maior risco e retorno esperado.

#### 4.2.5 Experimentos *out-of-sample*

Experimentos *out-of-sample* realizam simulações de negociações no mercado financeiro. Para isso um determinado método é aplicado e, dentre as soluções da fronteira eficiente proporcionada por aquele método, é selecionado o portfólio com maior valor de retorno sobre risco. Esse portfólio é comprado e mantido por um tempo determinado e, após esse tempo, um novo portfólio é escolhido, utilizando o mesmo procedimento.

Estes experimentos utilizam os dados do conjunto de dados 1 (com granularidade diária) e 2 (com granularidade horária), ambos considerando o período entre janeiro de 2013 e dezembro de 2016. Para o conjunto de dados 1, a série histórica é composta de dados de um ano e os métodos são novamente aplicados a cada mês em um processo de janela deslizando. Durante cada período de um mês, o portfólio comprado é mantido. Para o conjunto de dados 2, a série histórica de entrada dos métodos são composta de dois meses de dados e o portfólio selecionado é mantido por 3 dias.

O primeiro experimento compara métodos heurísticos e exatos em relação ao ganho financeiro proporcionado por cada um, considerando o conjunto de dados 1. O segundo experimento é uma análise de sensibilidade que procura avaliar os efeitos da magnitude dos custos de transação nos ganhos proporcionados pelos métodos propostos, também utilizando o conjunto de dados 1. Por fim, o terceiro experimento compara os ganhos utilizando os conjuntos de dados 1 e 2, para um mesmo método de otimização de portfólio.

### 4.3 Classificação de séries temporais financeiras

Neste trabalho, a classificação de tendência de séries financeiras é binária, de forma que cada ativo financeiro disponível é classificado como ativo promissor ou não promissor, sendo que cinco regras diferentes são consideradas para essa classificação. A classificação considera os conjuntos de dados 1 e 2, utilizando todos seus períodos de dados.

A partir das cinco informações coletadas sobre cada ativo: preço de abertura  $P_A$ , preço de fechamento  $P_F$ , preço mínimo  $P_L$ , preço máximo  $P_H$  e volume de negociações  $V$ , 60 atributos de entrada são formados, entre log retornos, indicadores técnicos e proporções de seus preços. Todos os atributos utilizados são valores proporcionais ou relativos de forma que estes atributos podem ser utilizados na comparação de diferentes ativos. Além disso, todos os atributos utilizados são numéricos.

Nos experimentos que utilizam apenas a classificação como sinal de compra, os ativos classificados como promissores são comprados e os não promissores são vendidos, caso esteja presente em um portfólio prévio. A quantidade a ser comprada ou vendida de cada ativo depende da quantidade desse ativo no portfólio prévio, sendo que o investimento final em cada ativo é igual ao capital total disponível dividido pela quantidade de ativos classificados como promissores.

### 4.3.1 Atributos de entrada

Os 22 primeiros atributos são log retornos espaçados com diferentes intervalos de tempo. Esses atributos são utilizados no trabalho de [Krauss, Do e Huck \(2016\)](#).

- Atributo 1-20: cada atributo  $j$ , entre 1 e 20, é formado da seguinte forma, para cada período  $t$  de um determinado ativo:

$$\text{Atributo } j = \ln\{P_F(t)\} - \ln\{P_F(t - j)\},$$

em que  $P_F(t)$  é o preço de fechamento de um determinado ativo no período  $t$ . Dessa forma, o Atributo 1, por exemplo, é o log retorno do período anterior ao período  $j$ . Considerando granularidade diária dos dados de entrada, o Atributo 1 será o log retorno no período entre os dias  $j - 1$  e  $j$ .

- Atributo 21:

$$\ln\{P_F(t)\} - \ln\{P_F(t - 30)\}.$$

- Atributo 22:

$$\ln\{P_F(t)\} - \ln\{P_F(t - 40)\}.$$

Os atributos 23 a 36 são formados por log retornos utilizando diferentes tipos de preço e diferentes períodos de referência. Esses atributos são encontrados no trabalho de [Paiva et al. \(2018\)](#).

- Atributo 23:

$$\ln\{P_F(t - 1)\} - \ln\{P_F(t - 2)\}.$$

- Atributo 24:

$$\ln\{P_F(t - 2)\} - \ln\{P_F(t - 3)\}.$$

- Atributo 25:

$$\ln\{P_F(t-3)\} - \ln\{P_F(t-4)\}.$$

- Atributo 26:

$$\ln\{P_H(t)\} - \ln\{P_A(t)\}.$$

- Atributo 27:

$$\ln\{P_H(t)\} - \ln\{P_A(t-1)\}.$$

- Atributo 28:

$$\ln\{P_H(t)\} - \ln\{P_A(t-2)\}.$$

- Atributo 29:

$$\ln\{P_H(t)\} - \ln\{P_A(t-3)\}.$$

- Atributo 30:

$$\ln\{P_H(t-1)\} - \ln\{P_A(t-1)\}.$$

- Atributo 31:

$$\ln\{P_H(t-2)\} - \ln\{P_A(t-2)\}.$$

- Atributo 32:

$$\ln\{P_H(t-3)\} - \ln\{P_A(t-3)\}.$$

- Atributo 33:

$$\ln\{P_L(t)\} - \ln\{P_A(t)\}.$$

- Atributo 34:

$$\ln\{P_L(t-1)\} - \ln\{P_A(t-1)\}.$$

- Atributo 35:

$$\ln\{P_L(t-2)\} - \ln\{P_A(t-2)\}.$$

- Atributo 36:

$$\ln\{P_L(t-3)\} - \ln\{P_A(t-3)\}.$$

- Atributo 37:

$$\ln\{P_F(t-1)\} - \ln\{P_A(t-1)\}.$$

Os atributos 38 a 40, utilizados em [Paiva \(2014\)](#), são formados por proporções de cada parte de um *candle* ou *candlestick*, representação gráfica de movimentação dos valores de um ativo que indica seus preços de abertura, fechamento, mínimo e máximo em um determinado período  $t$ , além de informar se o preço de abertura é menor (*candle* de alta) ou maior (*candle* de baixa) que o valor de fechamento utilizando, para isso, uma cor diferente para cada tipo de *candle*.

- Atributo 38: corresponde à proporção da sombra superior do *candle* em relação ao seu comprimento total:

$$\frac{P_H - \max(P_A, P_F)}{P_H - P_L}.$$

- Atributo 39: corresponde à proporção do corpo do *candle* em relação ao seu comprimento total:

$$\frac{|P_F - P_A|}{P_H - P_L}.$$

- Atributo 40: corresponde à proporção da sombra inferior do *candle* em relação ao seu comprimento total:

$$\frac{\min(P_A, P_F) - P_L}{P_H - P_L}.$$

Os atributos 41 a 47 são indicadores técnicos utilizados no trabalho de [Paiva et al. \(2018\)](#).

- Atributo 41: indicador Momentum, considerando um intervalo de 10 períodos. Neste trabalho, foram utilizados os preços mínimos, apesar de comumente serem utilizados preços de fechamento.

$$\frac{P_L(t)}{P_L(t - 10)}.$$

- Atributo 42: Índice de Força Relativa, considerando uma série de 14 períodos.

$$100 - \frac{100}{1 + MG/MP},$$

em que  $MG$  é a média dos ganhos (retornos positivos) e  $MP$  é a média das perdas (retornos negativos), ambos considerando os retornos de 14 períodos, definidos pela série histórica dos preços de fechamento.

- Atributo 43: SAR (do inglês, *stop and reverse*) Parabólico. Neste trabalho é utilizado um fator de aceleração  $AC$  de 0,01 e valor máximo de aceleração de 0,2. Para transformar o SAR Parabólico em um valor relativo, seu valor é dividido pelo preço de fechamento do ativo avaliado no período anterior.

$$SAR(t) = SAR(t - 1) + AC(PE - SAR(t - 1)),$$

em que  $SAR(t)$  é o SAR parabólico do período  $t$  e  $PE$  é um ponto extremo de máximo, em séries com tendência de alta, ou de mínimo, em séries com tendência de baixa.

$$\text{Atributo 43}(t) = \frac{SAR(t)}{P_F(t - 1)}.$$

- Atributo 44: Intervalo Verdadeiro (TR - *True Range*).

$$TR(t) = \frac{\max(P_H(t), P_F(t-1))}{\min(P_L(t), P_F(t-1))}.$$

- Atributo 45: Intervalo Médio Verdadeiro ou ATR (*Average True Range*, em inglês), considerando um intervalo de 14 períodos de dados.

$$ATR(t) = \frac{\sum_{i=0}^{13} TR(t-i)}{14}.$$

- Atributo 46: Chaikin *Accumulation Distribution Line* (AD).

$$AD(t) = AD(t-1) + V(t) \frac{(P_F(t) - P_L(t)) - (P_H(t) - P_F(t))}{P_H(t) - P_L(t)}.$$

- Atributo 47: indicador Volume Balanceado (OBV - *On Balance Volume*).

$$OBV(t) = \begin{cases} OBV(t-1) + V(t), & \text{se } P_F(t) > P_F(t-1), \\ OBV(t-1) - V(t), & \text{se } P_F(t) < P_F(t-1), \\ OBV(t-1), & \text{se } P_F(t) = P_F(t-1) \end{cases}.$$

Os atributos 48 a 55 são indicadores técnicos utilizados em trabalhos de classificação de séries financeiras como os de [Hsu et al. \(2016\)](#) e [Kara, Boyacioglu e Baykan \(2011\)](#).

- Atributo 48: Oscilador de Preço, utilizando média móvel simples, um intervalo rápido de 12 períodos e um intervalo lento de 26 períodos.

$$\frac{MMS(P_F, 12) - MMS(P_F, 26)}{MMS(P_F, 26)},$$

em que  $MMS(P_F, i)$  é a média móvel simples de  $i$  preços de fechamento para o intervalo composto por  $t - i + 1$  e  $t$ , sendo  $t$  o período atual.

- Atributo 49: Oscilador de Preço, utilizando média móvel exponencial (*MME*), um intervalo rápido de 12 períodos e um intervalo lento de 26 períodos.

$$\frac{MME(P_F, 12) - MME(P_F, 26)}{MME(P_F, 26)},$$

- Atributo 50: indicador  $\%R$  de Williams, considerando 5 períodos de dados.

$$\%R(t) = \frac{\max_{i=0}^4 P_H(t-i) - P_F(t)}{\max_{i=0}^4 P_H(t-i) - \min_{i=0}^4 P_L(t-i)}.$$

- Atributo 51: indicador  $\%R$  de Williams, considerando 10 períodos de dados.

$$\%R(t) = \frac{\max_{i=0}^9 P_H(t-i) - P_F(t)}{\max_{i=0}^9 P_H(t-i) - \min_{i=0}^9 P_L(t-i)}.$$

Os atributos 52 e 53 são formados, respectivamente, pelas linhas  $\%K$  e  $\%D$  do Oscilador Estocástico, considerando 10 períodos de dados.

- Atributo 52: linha %K do Oscilador Estocástico.

$$\%K(t) = \frac{P_F(t) - \min_{i=0}^9 P_L(t-i)}{\max_{i=0}^9 P_H(t-i) - \min_{i=0}^9 P_L(t-i)}.$$

- Atributo 53: linha %D do Oscilador Estocástico, formada pela média móvel simples da linha %K com um intervalo de 3 períodos de dados.
- Atributo 54: Oscilador Chaikin, com intervalo rápido de 3 períodos e intervalo lento de 10 períodos de dados.

$$MMS(AD, 3) - MMS(AD, 10).$$

- Atributo 55: Bandas de Bollinger %B.

$$\frac{P_F - P_L}{P_H - P_L}.$$

- Atributo 56: Índice de Canal de *Commodities* (CCI - *Comoddy Channel Index*), com um intervalo de 14 período de dados. Para o cálculo desse indicador, é utilizado o valor médio de preços do ativo para cada período. Neste trabalho, esse valor médio é considerado como a média aritmética de  $P_H$ ,  $P_F$ ,  $P_A$  e  $P_L$ .

$$CCI = PT - \frac{MMS(PT, 14)}{0,015 \sigma},$$

em que  $PT$  é a média aritmética dos preços máximo, mínimo e médio, e  $\sigma$  é o desvio padrão de  $PT$  para os 14 períodos.

Os atributos 57 a 59 são encontrados no trabalho de [Kumar e Thakur \(2016\)](#).

- Atributo 57: Índice de Força Verdadeira, com intervalo rápido de 13 períodos e intervalo lento de 25 períodos.

$$\frac{MME(MME(DIF, 25), 13)}{MME(MME(|DIF|, 25), 13)},$$

em que  $DIF$  é a série composta pelas diferenças de preços de fechamento atuais e os preços de fechamentos anteriores.

- Atributo 58: Oscilador Definitivo, utilizando os intervalos de 7, 14 e 28 períodos de dados.

$$\frac{(A_7 \times 4)(A_{14} \times 2)A_{28}}{4 + 2 + 1},$$

em que

$$A_i = \frac{\sum_{t=1}^i P_F(t) - \min(P_L(t), P_F(t-1))}{\sum_{t=1}^i TR(t)}.$$

- Atributo 59: Índice Ulcer com intervalo de 14 períodos. Primeiro, um *Drawdown* Percentual  $DP$  é definido:

$$DP(t) = \frac{P_F(t) - \max_{i=0}^{13} P_H(t-i)}{\max_{i=0}^{13} P_H(t-i)},$$

e, assim, o Índice Ulcer ( $IU$ ) é definido como:

$$IU(t) = \sqrt{\frac{\sum_{i=0}^{13} DP^2(t-i)}{14}}.$$

- Atributo 60: Índice de Fluxo de Dinheiro ( $IFD$ ).

$$IFD(t) = PT * V(t),$$

em que o preço típico  $PT$  é a média aritmética de  $P_H(t)$ ,  $P_L(t)$ ,  $P_A(t)$  e  $P_F(t)$ .

#### 4.3.1.1 Tratamento dos atributos de entrada

Cada um dos atributos citados são transformados de forma que se tornem normalizados, com média zero e desvio padrão igual a um, para cada intervalo de dados utilizados nos treinamentos dos modelos de classificação. Essas mesmas transformações ajustadas para os intervalos de treinamento, são aplicadas em seus respectivos intervalos de validação e teste. É importante notar que, aplicando essa transformação, os atributos possuem, garantidamente, média zero e desvio padrão um apenas nos intervalos de treinamento e, para validação e teste, espera-se que a média e desvio padrão sejam, aproximadamente, os mesmos. Supõe-se, ainda, que os atributos possuam distribuições simétricas em relação à sua mediana.

#### 4.3.2 Seleção de atributos

Para a seleção de atributos, são utilizados três métodos a priori, ou filtros e dois métodos embarcados. Esses três primeiros métodos são aplicados antes da normalização dos dados, enquanto que os dois últimos são aplicados após a normalização. O arcabouço computacional considera então, além do conjunto formado por todos os atributos, cinco subconjuntos de atributos para cada etapa de treinamento do modelo de classificação. Os subconjuntos formados pelos métodos embarcados são posteriormente utilizados por outros algoritmos de classificação, sendo então, aplicados como filtros. Esses subconjuntos são formados por cada um dos cinco métodos de seleção de atributos:

- Seleção de Atributos Baseada em Correlação, descrita na Subseção [3.3.7.1](#), com a estratégia de busca eliminação reversa, que se inicia com todos os atributos e elimina atributos utilizando uma heurística gulosa até que a eliminação de outro atributo resulte em um pior subconjunto de atributos;

- Análise de Componentes Principais, descrita na Subseção 3.3.7.2, que forma um novo conjunto de atributos não correlacionados. Nesse caso, o novo subconjunto possui 95% das informações do conjunto original dos 60 atributos;
- Agrupamento Hierárquico, como descrito na Subseção 3.3.7.3, aplicado aos 60 atributos citados. O corte na árvore resultante é feito estabelecendo um número de agrupamentos  $n_a$ , em que cada um desses grupos resultantes é utilizado como um atributo. Para determinar o valor de  $n_a$ , são realizados cortes com todos os valores de  $3a60$  e é escolhida a poda que maximiza o valor de Silhueta Média dos grupos;
- O algoritmo de Regressão Logística com regularizador LASSO (ver Subseção 3.3.1.1), com diferentes valores de penalidade de regularização  $\lambda_{LASSO}$ , é utilizado para formar um subconjunto de atributos, de forma que o melhor valor de  $\lambda_{LASSO}$ , dentre todos os valores considerados, é determinado na etapa de validação e os atributos cujos pesos no modelo treinado são diferentes de zero são escolhidos para o subconjunto de atributos. Os valores considerados para  $\lambda_{LASSO}$  são: 0,001; 0,003; 0,01; 0,03; 0,1; 0,3; 1; 3; 10; 30 e 100.
- O algoritmo Floresta Aleatória descrito na Subseção 3.3.5.1 é utilizado para selecionar alguns atributos dentre os 60 atributos considerados. Para isso, a importância de cada atributo é avaliada de acordo com seu efeito sobre a variável de saída, de forma que os atributos que, quando alterados, normalmente alteram a variável de saída possuem maior importância e, assim, a cada variável é atribuída uma importância, sendo que a soma de todas essas importâncias deve se igual a 1. Utilizando esse algoritmo, é formado um subconjunto composto pela menor quantidade atributos em que soma de suas importâncias é maior ou igual a 0,8.

### 4.3.3 Regras de saída

As regras de saída definem as variáveis alvo ou classes de cada elemento do conjunto de entrada. Neste trabalho são consideradas cinco diferentes regras de saída. Nas três primeiras regras, a classe de cada ativo é determinada apenas pelos atributos daquele mesmo ativo, enquanto que as duas últimas regras definem as regras de cada ativo de acordo com os atributos de todas os ativos de um determinado período.

#### 4.3.3.1 Regra 1

Na regra 1, a variável alvo  $y_i(t)$  do ativo  $i$  no período  $t$  é definida como:

$$y_i(t) = \begin{cases} 1, & \text{se } r_i(t+1) > 0, \\ 0, & \text{caso contrário} \end{cases},$$

em que  $r_i(t)$  é o retorno linear do ativo  $i$  no período  $t$ . Vários trabalhos na literatura utilizam

esta regra, como [Kara, Boyacioglu e Baykan \(2011\)](#), [Hsu et al. \(2016\)](#), [Gerlein et al. \(2016\)](#), [Pang et al. \(2018\)](#) e [Assis \(2019\)](#).

#### 4.3.3.2 Regra 2

Na regra 2, as classes são definidas da seguinte forma:

$$y_i(t) = \begin{cases} 1, & \text{se } r_i(t+1) > 0,01, \\ 0, & \text{caso contrário} \end{cases} .$$

Esta regra é utilizada nos trabalhos de [Pommeranzenbaum \(2014\)](#), [Duarte \(2018\)](#) e [Paiva et al. \(2018\)](#), mesmo que esses trabalhos utilizem limites mínimos de retorno (também denominados alvos) diferentes de 0,01.

#### 4.3.3.3 Regra 3

A regra 3, proposta por [Thakur e Kumar \(2018\)](#), leva em consideração os preços mínimo  $P_{L_i}$ , máximo  $P_{H_i}$  e de fechamento  $P_{F_i}$  de cada ativo  $i$  e define as classes da seguinte forma:

$$y_i(t) = \begin{cases} 1, & \text{se } P_{F_i}(t+1) > P_{H_i}(t) \text{ e } P_{L_i}(t+1) > P_{L_i}(t), \\ 0, & \text{caso contrário} \end{cases} .$$

#### 4.3.3.4 Regra 4

Para a regra 4, as classes são formadas por:

$$y_i(t) = \begin{cases} 1, & \text{se } r_i(t+1) > \text{mediana de } \{r_j(t+1), j = 1, \dots, n\}, \\ 0, & \text{caso contrário} \end{cases} .$$

Os trabalhos de [Marcelino \(2016\)](#) e de [Fu et al. \(2018\)](#) utilizam a regra 4.

#### 4.3.3.5 Regra 5

A regra 5, encontrada nos trabalhos de [Krauss, Do e Huck \(2016\)](#) e de [Fu et al. \(2018\)](#), define cada classe da seguinte forma:

$$y_i(t) = \begin{cases} 1, & \text{se } r_i(t+1) > 75\% \text{ dos elementos de } \{r_j(t+1), j = 1, \dots, n\}, \\ 0, & \text{caso contrário} \end{cases} .$$

### 4.3.4 Algoritmos de classificação

Esta subseção cita cada um dos algoritmos de classificação que integram o arcabouço computacional proposto e que são utilizados nos experimentos realizados neste trabalho. São descritos, também, os hiperparâmetros que possuem um conjunto de valores alterados no treinamento, e determinados pela validação, de cada algoritmo. A validação seleciona, dentre todos os valores considerados para um determinado hiperparâmetro, aquele que faz com que as predições apresentem maior valor para a métrica F1.

Os algoritmos de classificação são comparados com dois métodos de classificação utilizados como referência (*baselines*). O primeiro *baseline* atribui todas as amostras de entrada à classe que possui maior amostras, em uma determinada regra de saída. O segundo *baseline* atribui, para cada amostra, uma classe definida aleatoriamente.

#### 4.3.4.1 Regressão Logística com regularização *Ridge*

A Regressão Logística com regularização *Ridge*, descrito na Subseção 3.3.1.2, utiliza os mesmos valores de penalidade utilizados para a regularização LASSO. Essa penalidade  $\lambda_{Ridge}$ , na regularização *Ridge*, evita que o algoritmo se torne muito ajustado aos dados de treino e percam a capacidade de generalização.

- $\lambda_{Ridge}$  : 0,001; 0,003; 0,01; 0,03; 0,1; 0,3; 1; 3; 10; 30 e 100.

Para a minimização do gradiente do erro (Entropia Cruzada), é utilizado o algoritmo de otimização Saga, proposto por Defazio, Bach e Lacoste-Julien (2014).

#### 4.3.4.2 Floresta Aleatória

Os valores dos hiperparâmetros da Floresta Aleatória (ver Subseção 3.3.5.1) foram definidos com base no trabalho de Probst, Wright e Boulesteix (2019). Os hiperparâmetros da Floresta Aleatória que têm seus valores definidos na validação são o número de Árvores de Decisão  $n_{arvores}$ , número de variáveis ou atributos  $n_{variaveis}$  considerados na busca pela melhor divisão em um nó de alguma Árvore e o número máximo de divisões  $n_{divisoes}$  das Árvores, ou a profundidade máxima destas.

- $n_{arvores}$  : 100, 500 e 1000;
- $n_{atributos}$  :  $\sqrt{d}$  e  $d/3$ , em que  $d$  é a quantidade de atributos considerada;
- $n_{divisoes}$  :  $\infty$  e  $n_{tr} + 2n_{te}$ , em que  $n_{tr}$  e  $n_{te}$  são os tamanhos do conjunto de dados de treino e de teste, respectivamente.

#### 4.3.4.3 Máquina de Vetores de Suporte

Os valores dos hiperparâmetros da Máquina de Vetores de Suporte, descrita na Subseção 3.3.3, foram definidos com base no trabalho de Rojas-Domínguez et al. (2018). A etapa de validação compara diferentes valores para os hiperparâmetros fator de penalidade

$\lambda_{SVM}$  para o erro, que evita a perda da capacidade de generalização do algoritmo, funções núcleos (*kernels*) e o coeficiente  $\gamma$  dessas funções.

- $\lambda_{SVM}$  : 0,01; 0,1; 1; 10 e 100;
- *kernels* : RBF(*Radial Basis Function*) e sigmóide;
- $\gamma$  :  $1/d$ ; 0,0001; 0,01; 1 e 10.

#### 4.3.4.4 Perceptron Multicamadas

Os hiperparâmetros do Perceptron Multicamadas (MLP), algoritmo descrito na Subseção 3.3.2, são ajustados por valores em um intervalo definido com base no trabalho de Itano, Sousa e Del-Moral-Hernandez (2018) e empiricamente, com base em alguns testes realizados. Dois diferentes modelos de MLP são utilizados: com uma e duas camadas ocultas, já que um MLP com apenas duas camadas, com um número suficiente de unidades, pode aproximar qualquer mapeamento (LIPPMANN, 1987). Os hiperparâmetros definidos pela validação são a função de ativação, a quantidade de neurônios na primeira e na segunda camada oculta (*neuronios\_c1* e *neuronios\_c2*), e o fator de penalidade  $\lambda_{MLP}$  da regularização L2.

- *ativacao* : função tangente hiperbólica (tanh) e ativação linear retificada (ReLU);
- *neuronios\_c1* : 4, 8, 12 e 16;
- *neuronios\_c2* : 0 (apenas uma camada oculta) , 4, 8 e 12;
- $\lambda_{MLP}$  : 0,0001; 0,01 e 1.

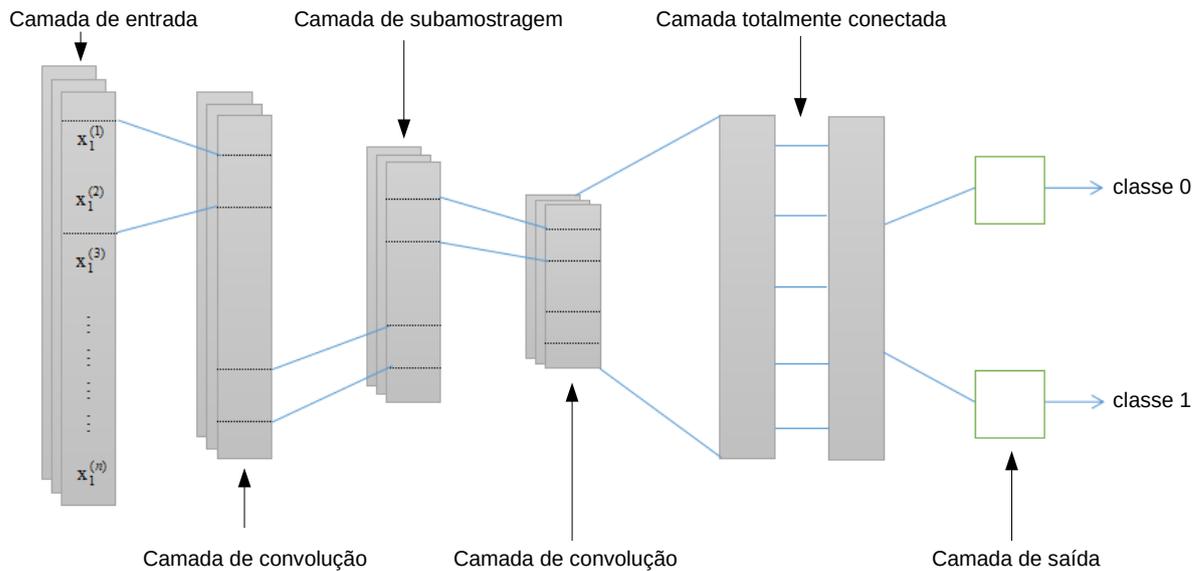
Para a minimização do gradiente do erro, é utilizado o algoritmo de otimização Adam, proposto por Kingma e Ba (2014).

#### 4.3.4.5 Rede Neural Convolutacional

O modelo e os valores dos hiperparâmetros da Rede Neural Convolutacional (Subseção 3.3.4) foram definidos com base no trabalho de Chen e He (2018). O modelo da rede é uma Rede Neural Convolutacional para uma dimensão (temporal) mostrado na Figura 23, em que cada um dos  $d$  atributos considerados são tratados como um canal e  $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}$  são os valores do atributo 1 para as  $n$  amostras, sendo 1 amostra por período de dados. Pelo fato desse modelo de uma dimensão não poder ser diretamente aplicado aos problemas que utilizam as regras de saída 4 e 5, que comparam amostras de todos os ativos, a Rede Neural Convolutacional é aplicada apenas na classificação de séries com as regras 1, 2 e 3. As camadas de entrada e saída, nessa rede, são similares às mesmas camadas na rede MLP e as camadas totalmente conectadas operam como suas camadas ocultas, utilizando as mesmas quantidades de neurônios estabelecidas para o MLP. Essas camadas também utilizam as mesmas funções de ativação e fator de regularização definidos para o MLP.

A camada de subamostragem utiliza o *average pooling*, método que reduz o tama-

Figura 23 – Arquitetura da Rede Neural Convolucional



Fonte: Adaptado de [Chen e He \(2018\)](#)

nho dos mapas de atributos, substituindo um subconjunto desses mapas de atributos pela média desses valores. Os hiperparâmetros definidos por validação são o *passo*, que representa a quantidade de períodos utilizados, por vez, na aplicação dos filtros, a quantidade de filtros aplicada e o tamanho da subamostragem.

- *passo* :  $n_{tr}/5$ ,  $n_{tr}/10$  e  $n_{tr}/20$ ;
- *filtros* : 8, 16 e 32;
- *subamostragem* : 1, 2 e 4.

#### 4.3.4.6 Adaboost

Os hiperparâmetros utilizados por [Rijn e Hutter \(2018\)](#) para o Adaboost, consideram a taxa de aprendizado  $\eta_{Adaboost}$ , tipo do algoritmo *algoritmo* e profundidade máxima  $max\_profundidade_{XGBoost}$ .

- $\eta_{Adaboost}$ : 0,01;
- *algoritmo*: SAMME e SAMME.R ([HASTIE et al., 2009](#));
- $max\_profundidade_{XGBoost}$ : 1 e 10.

#### 4.3.4.7 XGBoost

O XGBoost ([CHEN; GUESTRIN, 2016](#)) é o algoritmo de Reforço Gradiente utilizado neste trabalho. Os hiperparâmetros do algoritmo são definidos pela fase de validação e os valores considerados são os mesmos utilizados por [Probst, Wright e Boulesteix \(2019\)](#). O XGBoost utiliza Árvores de Decisão, com profundidade máxima  $max\_profundidade_{XGBoost}$

e o modelo *booster* do algoritmo é o *gbtree*. O algoritmo considera ainda a taxa de aprendizado  $\eta_{XGBoost}$ , largura mínima dos nós filhos *min\_largura*, porcentagem de subamostras das instâncias de treino *subamostras*, regularização L1 ( $\alpha_{XGBoost}$ ) e L2 ( $\lambda_{XGBoost}$ ).

- *max\_profundidade*<sub>XGBoost</sub>: 6 e 13;
- $\eta_{XGBoost}$ : 0,018;
- *min\_largura*: 1 e 2,06;
- *subamostras*: 1 e 0,839;
- $\alpha_{XGBoost}$ : 1 e 1,113;
- $\lambda_{XGBoost}$ : 1 e 0,982.

A taxa de aprendizado  $\eta$  utilizada foi de 0,018.

#### 4.3.4.8 Programação Genética

O algoritmo de Programação Genética utilizado é proposto por [Arnaldo, Krawiec e O'Reilly \(2014\)](#) e disponibilizado pelos autores. O algoritmo é baseado em árvores de decisão e utiliza como hiperparâmetros o tamanho da população *tam\_pop*, o conjunto de funções a serem aplicadas nos terminais *conj\_func*, tamanho do torneio, ou quantidade de indivíduos que participam do torneio para seleção dos pais *tam\_tor* e a taxa de mutação *tx\_mut*. O conjunto de funções padrão é utilizado, incluindo soma, subtração, multiplicação, divisão, seno e cosseno. Os demais valores dos hiperparâmetros são definidos experimentalmente, sempre incluindo o valor padrão de cada um deles:

- *tam\_pop*: 100, 200 e 500;
- *tam\_tor*: 2, 4 e 10;
- *tx\_mut*: 0.05, 0.1 e 0.15.

#### 4.3.4.9 Votação

A votação de classificadores, como mencionada em [3.3.5](#), produz uma predição com base nas predições de vários algoritmos de classificação. Para as regras 1, 2 e 3, os algoritmos de classificação são a Regressão Logística com regularização Ridge, a Floresta Aleatória, a Máquina de Vetores de Suporte, o Perceptron Multicamadas, a Rede Neural Convolutiva, o AdaBoost, XGBoost e a Programação Genética. As regras 4 e 5 utilizam os mesmos algoritmos, exceto a Rede Neural Convolutiva.

Dois votações são consideradas: a votação majoritária e a votação ponderada. Na votação majoritária a predição final, para cada amostra, é a classe com maior quantidade de predições entre os algoritmos de classificação considerados e, em caso de empate, a predição é realizada de forma aleatória. Na votação ponderada a probabilidade de pertencimento a cada classe, para cada amostra, é considerada e a predição final atribuída a classe com maior somatório de probabilidades, entre todos os algoritmos, para cada

amostra. No caso do SVM (Máquina de Vetores de Suporte), cuja solução originalmente não define diretamente os pesos de cada atributo de entrada, essa probabilidade não é utilizada e, nesse caso, é atribuído um valor de 0,25 à probabilidade de pertencimento à classe 1, para amostras atribuídas à classe 0, e 0,75 para amostras atribuídas à classe 1.

#### 4.3.4.10 Empilhamento

Empilhamentos de dois níveis são utilizados como algoritmos classificadores neste trabalho, como descrito na Subseção 3.3.5.2. Para as regras de saída 1, 2 e 3, a Regressão Logística com regularização Ridge, a Floresta Aleatória, a Máquina de Vetores de Suporte, o Perceptron Multicamadas, a Rede Neural Convolutacional, o AdaBoost, o XGBoost e a Programação Genética são utilizados como algoritmos de classificação nível 0. Todas as combinações possíveis desses cinco algoritmos são utilizadas para treinamento do algoritmo de nível 1, sendo que a melhor combinação (que proporciona maior acurácia) é selecionada na validação. Nesse método as probabilidades de predição para a classe 1 (probabilidade de ser um ativo promissor) gerados por cada algoritmo de nível 0 são entradas dos algoritmos de nível 1. Para as regras de saída 4 e 5, o mesmo procedimento é adotado, desconsiderando a Rede Neural Convolutacional.

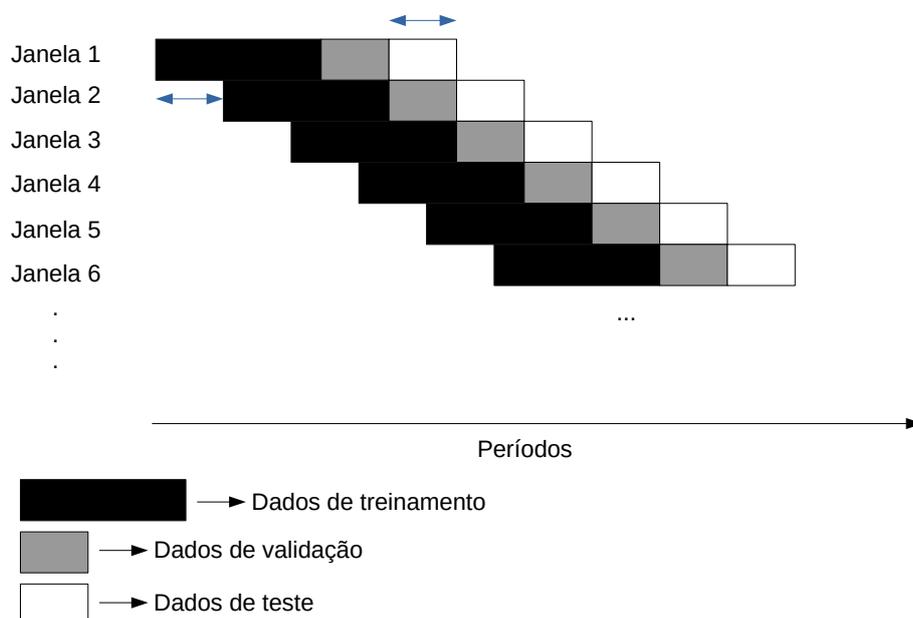
Os algoritmos utilizados para o algoritmo nível 1 são a Máquina de Vetores de Suporte e o Perceptron Multicamadas, por apresentarem bons desempenhos como algoritmos nível 1 no trabalho de Assis (2019). Os hiperparâmetros desses algoritmos de nível 1 são definidos da mesma forma que esses mesmos algoritmos utilizados como algoritmos nível 0.

### 4.3.5 Avaliação dos modelos de classificação

Para a avaliação dos modelos de classificação é utilizado um procedimento de janelas deslizantes que divide cada janela em dados de treinamento, validação e teste, como mostrado na Figura 24. A partir de uma primeira janela, a segunda janela é formada por um deslocamento de  $n_{te}$  períodos de dados, em que  $n_{te}$ , indicado pelas setas duplas na figura, é o tamanho (quantidade de períodos) dos dados de teste. Esse deslocamento garante que os dados utilizados para teste nessa segunda janela não foram utilizados na primeira janela.

Este método de avaliação assegura que a validação e o teste serão sempre realizados com dados posteriores aos dados utilizados no treinamento, assim como na prática, em que a predição é feita para dados não conhecidos, em tempo real, enquanto que o treinamento é realizado com dados passados, porém, temporalmente próximos aos dados de teste. Essa proximidade dos dados de treino, validação e teste é importante para a detecção de padrões locais, ou tendências de curta duração.

Figura 24 – Divisão dos dados em uma avaliação de janelas deslizantes



### 4.3.6 Experimentos

Assim como os experimentos para otimização de portfólios, os experimentos realizados para classificação de séries financeiras também são divididos em experimentos de avaliação do desempenho dos classificadores e experimentos de avaliação de ganhos proporcionados por esses classificadores. Os primeiros avaliam o desempenho das previsões proporcionadas pelos algoritmos. Para isso, são utilizadas as métricas acurácia, precisão, revocação e F1. Nesse trabalho, são desejáveis bons valores de precisão e revocação, o que atribui importância à métrica F1. Uma revocação mínima, porém, é necessária para que algumas combinações da classificação com otimização de portfólios sejam possíveis e, dado um valor mínimo de revocação, a precisão é considerada a métrica mais importante, por indicar a taxa de acerto de ativos promissores, que é a classe de interesse do trabalho. Por todas essas questões, a análise de todas essas métricas é necessária nos experimentos de avaliação de desempenho.

Os experimentos de avaliação de ganhos financeiros comparam os ganhos, em simulações de operações no mercado financeiro, proporcionados por previsões de cada algoritmo e regra de saída. Para isso, são analisados retornos, riscos e *drawdowns* mensais, além do retorno acumulado de todo o intervalo de simulação. Os experimentos de classificação utilizam tanto o conjunto de dados 1, como o conjunto de dados 2, considerando o intervalo entre janeiro de 2008 e dezembro de 2017.

#### 4.3.6.1 Experimentos de análise de desempenho

Um primeiro experimento varia valores do tamanho dos dados de treinamento e teste, sendo que o tamanho dos dados de validação é igual ao tamanho dos dados de teste, por motivos de simplificação. Para esse experimento, é utilizado o conjunto de dados 1 e o algoritmo Floresta Aleatória é utilizado, já que este é um algoritmo também utilizado para seleção de atributos. Todas as seis formas de seleção de atributos do arcabouço, incluindo a utilização do conjunto de todos os atributos, são considerados e a forma de seleção a ser utilizada no teste é determinada na validação.

Os tamanhos dos dados de teste utilizados neste primeiro experimento são 10, 20, ..., 100 e os tamanhos dos dados de treinamento são valores definidos de acordo com o tamanho dos dados teste, multiplicando este tamanho pelos valores 1; 1,5; 2; 4 e 10. O limite de 1000 períodos para o tamanho dos dados de treinamento é estabelecido, já que a quantidade total de dados disponível é limitado. A tabela 1 mostra os diferentes tamanhos para o conjunto de treino, formados pelos possíveis pares de tamanhos dos conjuntos de dados de teste e multiplicadores. Este experimento possui, como objetivo, determinar valores fixos para os tamanhos desses dados para a realização dos próximos experimentos, além de analisar o efeito da quantidade do tamanho dos dados de treinamento e teste no desempenho do método, uma vez que uma quantidade pequena de dados pode ser insuficiente para uma detecção de padrões e uma quantidade grande pode incluir dados muito antigos, dificultando a análise dos dados mais recentes.

Tabela 1 – Tamanhos dos conjuntos de treino para cada tamanho do conjunto de teste

Tamanho do conjunto de teste	Multiplicador				
	1x	1,5x	2x	4x	10x
<b>10</b>	10	15	20	40	100
<b>20</b>	20	30	40	80	200
<b>30</b>	30	45	60	120	300
<b>40</b>	40	60	80	160	400
<b>50</b>	50	75	100	200	500
<b>60</b>	60	90	120	240	600
<b>70</b>	70	105	140	280	700
<b>80</b>	80	120	160	320	800
<b>90</b>	90	135	180	360	900
<b>100</b>	100	150	200	400	1000

O segundo experimento considera o tempo de execução gasto na validação de todas as seis formas de seleção de atributos e, procurando reduzir esse tempo de execução, é analisado o desempenho do algoritmo Floresta Aleatória, utilizando o conjunto de dados 1, combinando diferentes formas de seleção de atributos, com o objetivo de utilizar a menor quantidade de seleção de atributos, reduzindo o tempo de treinamento, sem reduzir o desempenho do algoritmo. A estratégia utilizada consiste em utilizar apenas uma forma de

seleção de atributos e adicionar novas formas até que se atinja um desempenho similar ao desempenho do algoritmo utilizando todas as formas de seleção de atributos na etapa de validação.

O terceiro experimento considera apenas os métodos de seleção de atributos escolhidos no segundo experimento e compara o desempenho de todos os algoritmos de classificação, descritos na subseção 4.3.4 utilizando tanto o conjunto de dados 1 quanto o conjunto de dados 2, para todas as cinco regras de saída. Neste experimento, cada combinação de conjunto de dados de entrada e regra de saída é considerada um problema diferente, totalizando 10 problemas diferentes, e os classificadores são comparados apenas neste escopo. Comparações gerais (para todos os conjuntos de dados de entrada e regras de saída) não são realizadas.

#### 4.3.6.2 Experimentos de análise de ganhos

O primeiro experimento considera todas as cinco regras de saída e compara as previsões fornecidas pelos melhores algoritmos, em cada regra, utilizando os conjuntos de dados de entrada 1 e 2. É analisado, no geral, qual dos dois conjuntos de entrada proporciona melhores ganhos financeiros em simulações de investimentos financeiros utilizando as previsões dos classificadores. O segundo experimento considera apenas o melhor conjunto de entrada determinado pelo primeiro método e compara os ganhos financeiros proporcionados quando se utiliza cada uma das cinco regras de saída.

## 4.4 Combinações de otimização de portfólio e classificação de séries financeiras

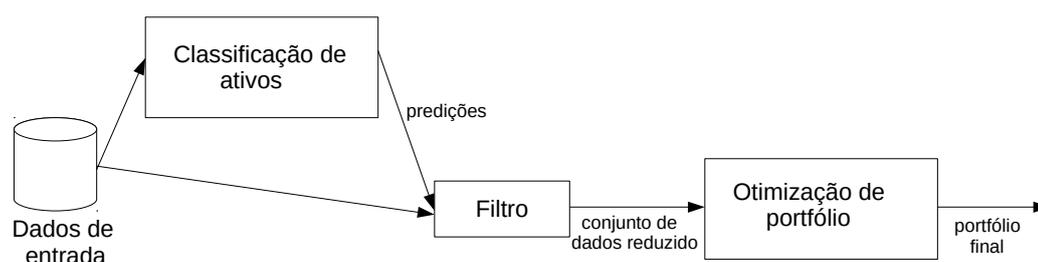
O arcabouço computacional proposto é composto de três formas diferentes de integração das abordagens de otimização de portfólios e classificação de séries financeiras. Para as duas primeiras combinações, a saída da etapa de otimização de portfólio é um único portfólio, que apresenta maior valor de retorno sobre risco dentre os portfólios não dominados resultantes dos modelos multiobjetivos. Para a terceira combinação, a saída da etapa de otimização de portfólio é constituída por esse conjunto de portfólios não dominados.

### 4.4.1 Classificação seguida de otimização

A classificação é feita antes da otimização de portfólios. A classificação das séries dos ativos é realizada com todos os dados de entrada e, após a realização das previsões, a otimização de portfólios é realizada apenas com os ativos considerados promissores (atribuídos à classe 1). Assim, as previsões da classificação atuam como um filtro de

ativos considerados na otimização de portfólio, que proporciona um portfólio financeiro que também é a saída de todo o arcabouço computacional neste caso. É importante ressaltar a importância de algoritmos com bons valores de revocação para a Combinação 1, já que maiores quantidades de ativos promissores é fundamental para a diversificação do portfólio. Utilizando esta combinação, o fluxograma do arcabouço computacional proposto é apresentado na Figura 25.

Figura 25 – Arcabouço computacional utilizando a Combinação 1



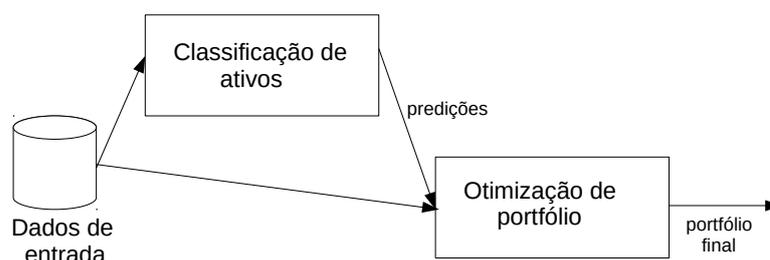
#### 4.4.2 Otimização com restrição de classificação

Esta forma de combinação também realiza a classificação dos ativos antes da otimização de portfólios, mas neste caso, alguns ativos promissores são inseridos no portfólio por meio da restrição de ativos obrigatórios, garantindo que todos esses ativos selecionados pela classificação serão, necessariamente, parte do portfólio proporcionado pela etapa de otimização. Os demais ativos a compor o portfólio são escolhidos pela etapa de otimização de portfólios considerando todos os ativos disponíveis. Neste caso, todos os dados de entrada são utilizados tanto pela etapa de classificação como pela etapa de otimização. Para este cenário, a precisão é mais importante do que a revocação, pois poucos ativos serão selecionados por esta etapa, mas os ativos selecionados devem possuir alta probabilidade de apresentar um bom desempenho futuro. Novamente, o portfólio proporcionado pela etapa de otimização de portfólios é a saída gerada pelo arcabouço computacional, quando a Combinação 2 é aplicada. Utilizando esta combinação, o fluxograma do arcabouço computacional proposto é apresentado na Figura 26.

#### 4.4.3 Otimização seguida de classificação

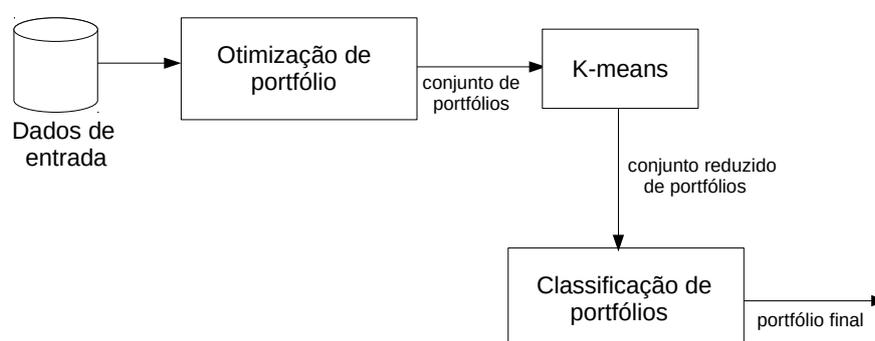
A Combinação 3, diferente das outras duas, aplica a classificação de séries financeiras após a otimização de portfólios. Neste caso, a otimização de portfólios é realizada com todos os dados de entrada e, após a geração da fronteira não dominada de portfólios, esta é primeiramente reduzida aplicando-se o algoritmo de agrupamento K-means e, posteriormente, o conjunto de portfólios reduzido é enviado à etapa de classificação que, neste caso, classifica esses portfólios em vez dos ativos financeiros considerados. Dessa forma, a

Figura 26 – Arcabouço computacional utilizando a Combinação 2



classificação escolhe o portfólio de maior probabilidade de pertencimento à classe 1 para se investir, sendo que esse portfólio selecionado é a saída do arcabouço computacional para este caso. Assim, a precisão do algoritmo de classificação é fundamental para a Combinação 3. Utilizando esta combinação, o fluxograma do arcabouço computacional proposto é apresentado na Figura 27. Os atributos e tamanhos dos dados dos conjuntos de treinamento e teste do classificador são os mesmos definidos para o método de classificação.

Figura 27 – Arcabouço computacional utilizando a Combinação 3



#### 4.4.4 Experimentos

As três combinações são comparadas em simulações de operações no mercado financeiro quando se usa cada uma delas. Essas simulações consideram retornos, riscos e *drawdowns* mensais, além do retorno acumulado para um período de 8 anos de simulação considerando dois casos diferentes, utilizando o conjunto de dados de entrada 1 e 2. Avalia-se o desempenho das combinações em cada caso, separadamente, e de maneira geral, considerando os dois conjuntos de dados de entrada.

## 4.5 Ambiente computacional

Todos os testes são realizados em um computador cujo processador é um Intel® Xeon® E5506 v3, Núcleo i7 (8 núcleos físicos) e frequência de 2,13 GHz. A memória RAM é de 48 GB e o sistema operacional utilizado é o CentOS versão 6.6.

Para os testes referentes à otimização de portfólios, é utilizada a linguagem de programação C, com compilador GCC 6.0.1. São utilizadas a biblioteca *math.h*, que fornece um conjunto de funções para operações matemáticas, e a API (*Application Programming Interface*) *CPLEX Callable Library*, que oferece algoritmos exatos para solucionar problemas de otimização. Neste trabalho, é utilizado o algoritmo *branch-and-cut* oferecido pela API.

Para os testes referentes à classificação de séries temporais financeiras, utiliza-se a linguagem de programação Python, com interpretador de versão 3.4. São utilizadas as bibliotecas NumPy, que fornece funções para operações matemáticas e de manipulação de conjunto *n*-dimensionais de dados, a Pandas, que fornece estruturas de dados de alto desempenho e ferramentas para análise de dados, o scikit-learn, que oferece ferramentas de análise e mineração de dados, o xgboost, para a implementação do algoritmo XGBoost, a ta-lib, que fornece alguns indicadores técnicos, TensorFlow, que fornece ferramentas para o desenvolvimento de algoritmos de Aprendizagem de Máquina e o Keras, que fornece ferramentas para desenvolvimento de algoritmos de Aprendizagem Profunda (*Deep Learning*). Neste trabalho, a Rede Neural Convolutiva é desenvolvida utilizando as bibliotecas Keras e TensorFlow. Os demais algoritmos de Aprendizagem de Máquina são implementados utilizando a biblioteca scikit-learn, utilizando um valor de *random-state*, semente que define a sequência de valores com distribuição uniforme (representando uma sequência de valores aleatórios), igual a 1234, para fins de reprodutibilidade. Para a execução da programação genética, foi utilizado o algoritmo MRGP (Multiple Regression Genetic Programming) disponível em <<https://flexgp.github.io/gp-learners/mrgp.html>>.

## 5 Análise e Discussão dos Resultados

Este capítulo apresenta, analisa e discute os resultados dos experimentos planejados, de forma separada para cada etapa do arcabouço computacional, como organizado no capítulo 4, que descreve esses experimentos. Configurações específicas de parâmetros para os modelos e algoritmos propostos são determinados de forma empírica ou com base em trabalhos relacionados.

### 5.1 Otimização de portfólio

Os parâmetros adotados para os modelos de otimização de portfólio são: número de ativos  $n$  igual a 47; nível de confiança  $\alpha$  de 0,95; probabilidade de cada cenário ocorrer  $p_t(t = 1, \dots, T)$  igual à quantidade total de cenários  $T$ ; número mínimo de ações  $m_i, \forall i, i = 1, \dots, N$ , igual a 100; capital disponível  $C$  de R\$ 100.000,00; custo de transação proporcional  $\beta$  de 0,015%; custo de transação fixo  $f$  de R\$ 20,00; proporção de investimento mínimo  $w_{min}$  de 0,001; proporção de investimento máximo  $w_{max}$  de 0,5; cardinalidade mínima  $k_{min}$  igual a 3; cardinalidade máxima  $k_{max}$  igual a 25; proporção mínima  $\rho$  do capital disponível  $C$  para investimento igual a 0,9.

O nível de confiança  $\alpha$  e o número de ações  $m_i$  que compõem um lote para cada ativo  $i$  foram selecionados conforme os trabalhos relacionados de Hanaoka (2014), Barroso (2017) e Ferreira (2018). Para o capital disponível  $C$  foi escolhido o mesmo valor de Ferreira (2018). Por fim, os custos de transação, fixo e proporcional ( $f$  e  $\beta$ ), derivam dos custos de emolumentos, corretagem e custódia, com os valores referentes àqueles informados pelo sistema de negociação eletrônica (*home broker*) do Banco do Brasil Banco. . . (2019). Nos testes *in-sample*, foi utilizada quantidade de lotes iniciais 0 para todos os ativos ( $x_i^{(0)} = 0, \forall i, i = 1, \dots, n$ ).

Os parâmetros utilizados para os algoritmos evolutivos (Métodos 1 e 2) são: quantidade máxima de gerações  $G$  igual a 1000; tamanho da população  $N$  igual a 500; índice do cruzamento SBX ( $n_c$ ) igual a 10 e índice da mutação polinomial ( $n_m$ ) igual a 50. Os parâmetros de probabilidade de cruzamento ( $p_c$ ) e de mutação ( $p_m$ ) são alterados de forma adaptativa, como proposto em Ribeiro, Barbosa e Arantes (2010), utilizando os mesmos valores iniciais adotados nesse trabalho. O tamanho da população é o mesmo adotado nos trabalhos de Deb et al. (2011) e Ferreira (2018), a quantidade máxima de gerações definida de maneira empírica e os índices de cruzamento e mutação, conforme o trabalho de Deb et al. (2011).

Cada método heurístico (Métodos 1 e 2), estocástico, foi executado 30 vezes. Os

resultados foram analisados por meio de testes estatísticos de análise de variância ANOVA, uma vez que a verificação visual de gráficos dos resíduos em relação à ordem da coleta dos dados e dos resíduos em relação aos valores ajustados não indicou a existência de padrões, corroborando as hipóteses de independência e homocedasticidade desses dados. O gráfico de Papel de Probabilidade não apresenta valores com grandes desvios da reta que representa as frequências de valores seguindo uma distribuição normal. Além disso, a análise visual da distribuição de frequência dos valores dos dados analisados indica uma simetria destes.

### 5.1.1 Experimentos *in-sample*

Análises *in-sample* consideram apenas o período de dados referente ao ano de 2013 e procuram avaliar o desempenho dos métodos implementados no arcabouço computacional, comparando o desempenho desses, entre eles, e com outros algoritmos encontrados na literatura.

#### 5.1.1.1 Comparação dos algoritmos propostos com algoritmos da literatura

O desempenho das meta-heurísticas propostas neste trabalho é comparado com o desempenho de meta-heurísticas propostas em trabalhos anteriores, com base nas métricas  $\Delta$ , hipervolume e tempo de execução.

Em relação ao Modelo 1, com variáveis de decisão inteiras, representando as quantidades de lotes de ações de cada ativo que compõe o portfólio, o NSGA-II proposto por [Ferreira \(2018\)](#) para solucionar modelos de otimização de portfólio com restrição de cardinalidade fixa, é utilizado. Para solucionar o Modelo 1, esse método é executado com valores de cardinalidade que iniciam em três e aumentam de um em um, compondo uma fronteira de Pareto combinada de todas as execuções anteriores, até que um dos valores de cardinalidade produza apenas portfólios dominados por algum portfólio da fronteira combinada, sendo que essa sequência de execuções do algoritmo de [Ferreira \(2018\)](#) é denominado aqui de Método L1. Essa fronteira combinada resultante do Método L1 é comparada, na Figura 28, com a fronteira proporcionada por uma execução do Método 1 proposto, projetado para solucionar o modelo com restrição de cardinalidade variável. Pela figura, nota-se que as soluções combinadas para diferentes execuções do Método L1 são visualmente próximas às soluções do Método 1.

Apesar das semelhanças entre as soluções dos dois métodos, a necessidade de se executar o Método L1 muitas vezes para se obter um resultado próximo ao do Método 1 pode ser considerada uma desvantagem desse método. A Figura 29 apresenta os diagramas de caixa (*boxplots*) que comparam os tempos de execução desses métodos e revela que o Método L1, executado 30 vezes, apresenta um tempo de execução significativamente maior que o Método 1. Para este caso, todas as execuções do Método L1 param no valor

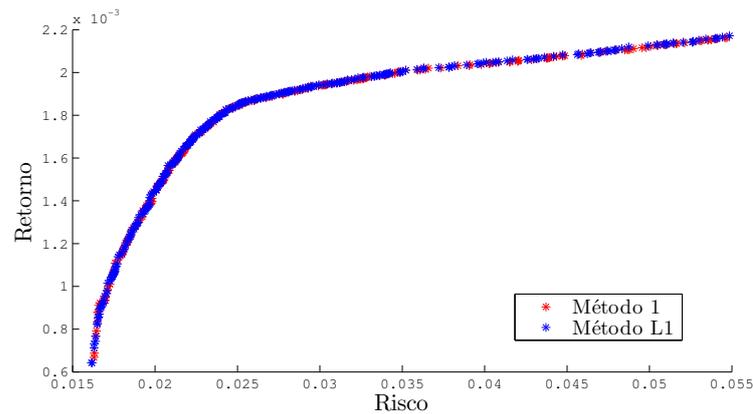


Figura 28 – Fronteiras de Pareto dos Métodos 1 e L1

de cardinalidade dez, sendo que o NSGA-II de [Ferreira \(2018\)](#) é executado oito vezes. Outra desvantagem do Método L1, em relação ao Método 1, é a impossibilidade de incluir portfólios de cardinalidade 11, por exemplo, em sua fronteira de Pareto combinada, sem que haja algum portfólio de cardinalidade 10.

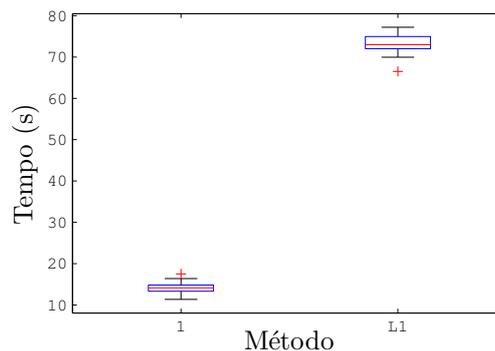


Figura 29 – Tempo de execução dos Métodos 1 e L1

Em relação ao Método 2, baseado no NSGA-II de [Deb et al. \(2011\)](#) e que propõe alterações a este, seu desempenho é comparado com o do algoritmo de [Deb et al. \(2011\)](#), ambos aplicados ao Modelo 2. Para facilitar a apresentação dos resultados, o NSGA-II proposto por [Deb et al. \(2011\)](#), da literatura, é denominado Método L2.

A Figura 30a apresenta o diagrama de caixa (*boxplot*) da métrica  $\Delta$  e a Figura 30b mostra o diagrama de caixa do hipervolume para os Métodos 2 e L2. Observa-se que o Método L2 apresenta um espalhamento das soluções na fronteira eficiente mais uniforme que as soluções do Método 2, já que a Métrica  $\Delta$  do Método L2 é menor. Combinado ao fato de que o Método 2 apresenta soluções com cardinalidade a partir de 3, enquanto o Método L2 apresenta soluções com cardinalidade de no mínimo 5 ativos, o melhor espalhamento do Método L2 pode ser explicado por uma maior quantidade de indivíduos de cardinalidades semelhantes e, portanto, mais próximos em termos de suas funções retorno e risco. Entretanto, o diagrama de caixa de hipervolume demonstra que as mudanças

propostas para o Método 2 melhoram a cobertura das soluções, de forma que o Método 2 apresenta um hipervolume maior que o L2, indicando uma maior proximidade das soluções do Método 2 à fronteira Pareto-ótima.

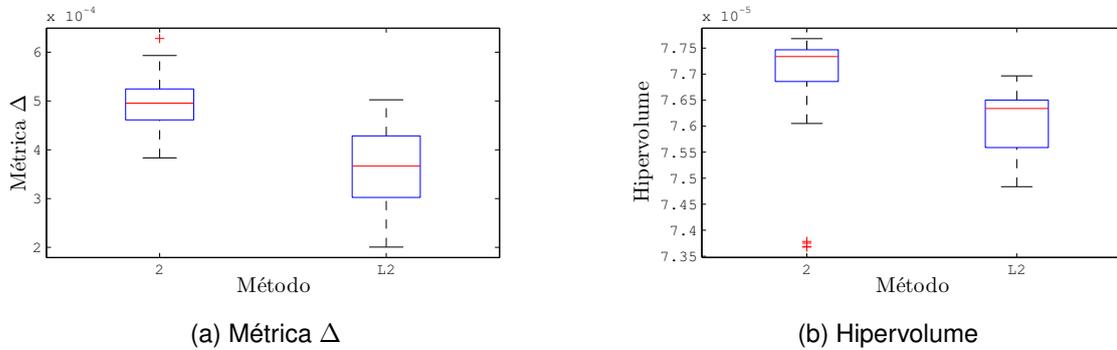


Figura 30 – Diagramas de caixa da métrica  $\Delta$  e hipervolume para os Métodos 2 e L2

Essas análises são bem representadas pela Figura 31, que mostra as fronteiras de Pareto das soluções proporcionadas por uma execução do Método 2 e uma execução do Método L2. Como esperado, as mudanças feitas no Método 2 preservam soluções de baixa cardinalidade na fronteira final, por meio da reinserção de soluções com maiores probabilidades de possuírem baixa cardinalidade. Portanto, como os portfólios de maiores retornos são os menos diversificados, sendo que o portfólio de maior retorno esperado é aquele composto apenas pelo ativo de maior média histórica de retornos, o Método 2 consegue identificar portfólios com retornos maiores que os portfólios do Método L2 e sua fronteira de Pareto se estende nessa direção. Por fim, pode-se considerar que o Método 2 proposto é mais eficiente que o Método L2 para o problema abordado neste trabalho.

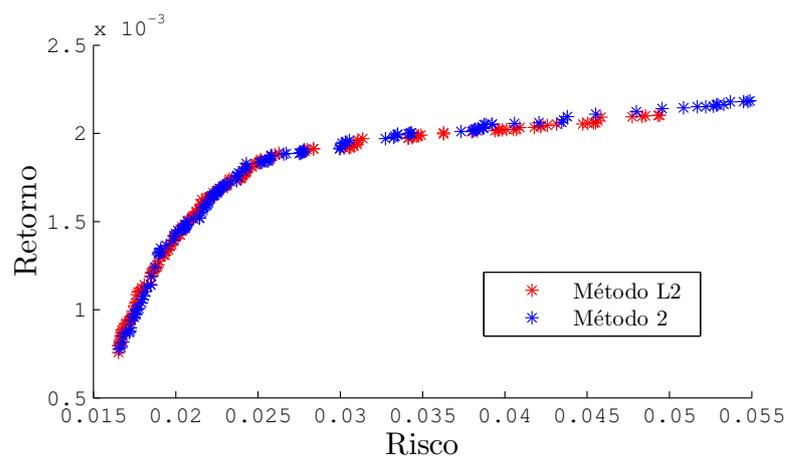


Figura 31 – Fronteiras de Pareto dos Métodos 2 e L2

### 5.1.1.2 Análise dos métodos heurísticos e exatos

Comparando o desempenho de todos os métodos do arcabouço computacional proposto, a Figura 32 apresenta os diagramas de caixa dos hipervolumes de cada um

desses métodos e a Figura 33 apresenta os diagramas de suas métricas  $\Delta$ .

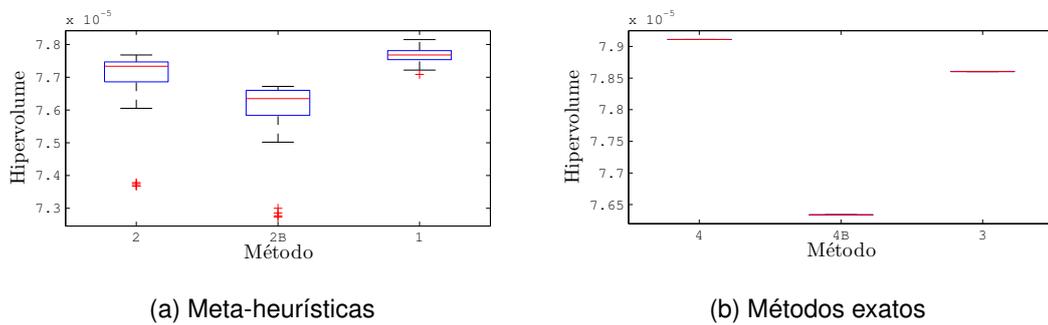


Figura 32 – Diagrama de caixa do hipervolume para os Métodos 1, 2, 2B, 3, 4 e 4B

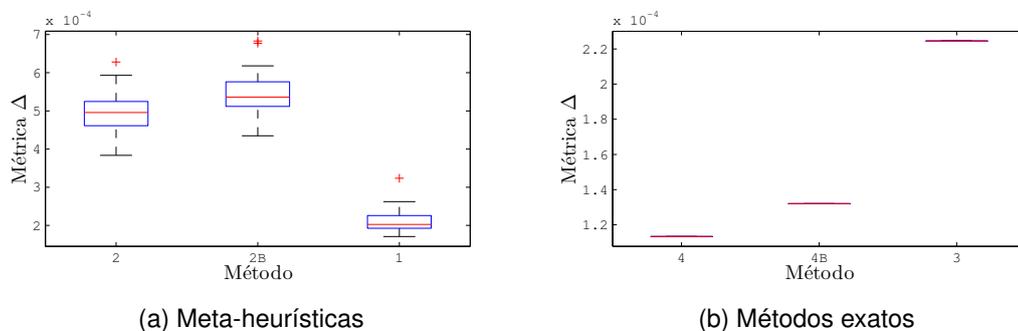


Figura 33 – Diagrama de caixa da métrica  $\Delta$  para os Métodos 1, 2, 2B, 3, 4 e 4B

Considerando as meta-heurísticas, o Método 1 que soluciona o Modelo 1, com variáveis de decisão inteiras, apresenta um hipervolume melhor (maior) que o do Método 2, possivelmente devido a um maior espaço de busca para o Modelo 2, gerando uma maior dificuldade para o método em encontrar boas soluções. Essa dificuldade em encontrar boas soluções também é expressa pela maior variabilidade da métrica apresentada pelo Método 2. Outro efeito dessa dificuldade de encontrar boas soluções é um pior valor de métrica  $\Delta$ , apresentando maiores valores e maior variabilidade destes, do Método 2, em relação ao Método 1. Além disso, a transformação das soluções contínuas em soluções inteiras aplicada pelo Método 2B piora a qualidade das soluções, tanto em relação ao espalhamento como em relação à cobertura das soluções.

Pela Figura 34 que mostra as fronteiras de Pareto proporcionadas pelas meta-heurísticas, pode-se notar visualmente que os Métodos 1 e 2 apresentam soluções próximas em suas fronteiras, apesar do Método 1 apresentar um espalhamento das soluções pouco melhor. Porém, quando o Método 2B transforma as soluções do Método 2, grande parte dessas soluções geradas se afastam muito das soluções dos outros dois métodos, dominadas por essas. Assim, comparando os Métodos 1 e 2B, cujas soluções são valores inteiros dos lotes de investimento de cada ativo, nota-se que o Método 1 é visualmente melhor que o Método 2B.

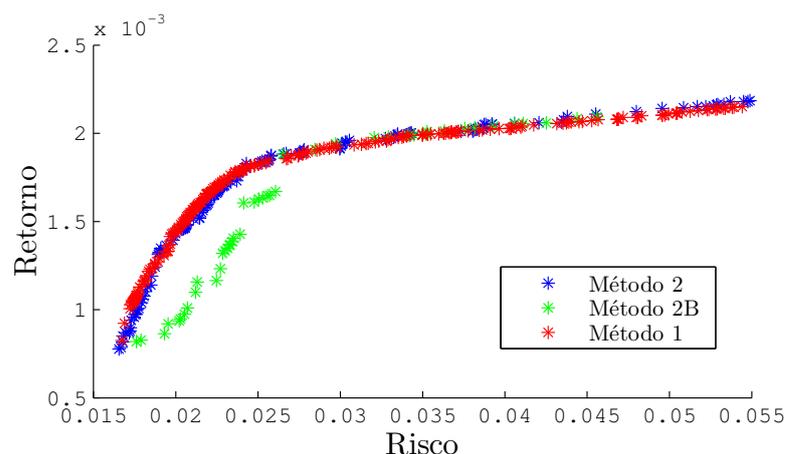


Figura 34 – Fronteiras de Pareto dos Métodos 1, 2 e 2B

Já em relação aos métodos exatos, o Método 4 com variáveis de decisão reais supera o Método 3 com variáveis de decisão inteiras, considerando tanto o hipervolume quanto a métrica  $\Delta$ . Esse desempenho melhor do Método 4 era esperado, já que o problema com variáveis inteiras (Modelo 3) é um subconjunto do problema com variáveis reais (Modelo 4).

Na prática, porém, os investimentos em ativos são realizados em lotes de ações e quando as soluções do Método 4 são transformadas pelo Método 4B em soluções inteiras, que representam quantidades de lotes, o hipervolume se torna pior que o do Método 3. Apesar disso, o Método 4B ainda apresenta uma métrica  $\Delta$  melhor que a do Método 3, mesmo apresentando uma piora em relação ao Método 4. Esse maior valor de hipervolume, mesmo com um valor pior para métrica  $\Delta$ , indica uma melhor cobertura das soluções do Método 3.

De fato, as fronteiras de Pareto dos algoritmos exatos, na Figura 35, confirmam uma dominância de parte das soluções do Método 3 em relação às soluções do Método 4B. Além disso, considerando o hipervolume mais importante que a métrica  $\Delta$ , por considerar tanto o espalhamento das soluções quanto a cobertura destas, pode-se destacar um melhor desempenho do Método 3 sobre o 4B.

Na prática, então, pode-se dizer que os algoritmos que solucionam os modelos com variáveis inteiras (Métodos 1 e 3) proporcionam soluções melhores que os métodos que solucionam modelos com variáveis reais, quando estes são transformados em lotes de ações (Métodos 2B e 4B).

Dessa forma, as soluções do melhor método heurístico (Método 1) e do melhor método exato (Método 3) são comparadas quando aplicadas ao Modelo 1, por este estar mais próximo da realidade que o Modelo 3. Em termos de métrica  $\Delta$ , o diagrama de caixa na Figura 36a mostra que a mediana para o Método 1 é menor que para o Método 3. Uma análise de variância (ANOVA) confirma uma diferença estatística entre os valores da métrica

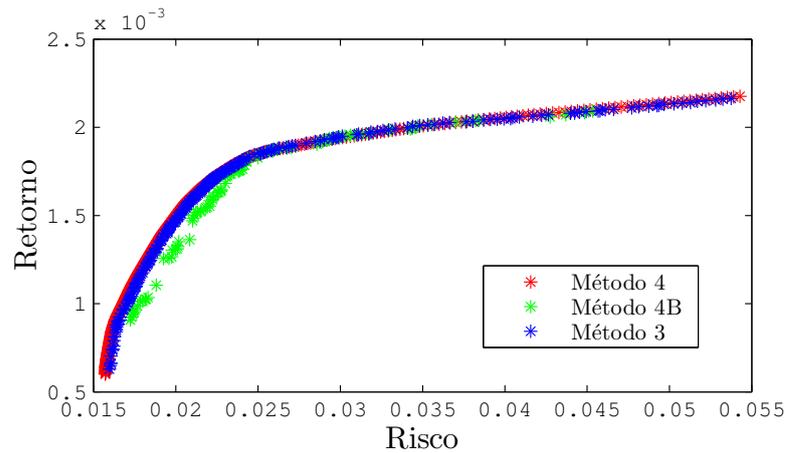


Figura 35 – Fronteiras de Pareto dos Métodos 3, 4 e 4B

para os dois métodos com um p-valor igual a 0.0154. A Figura 36b apresenta o teste de comparação múltipla de Tukey, revelando que o espalhamento das soluções do Método 3 é melhor, considerando um nível de confiança de 95%.

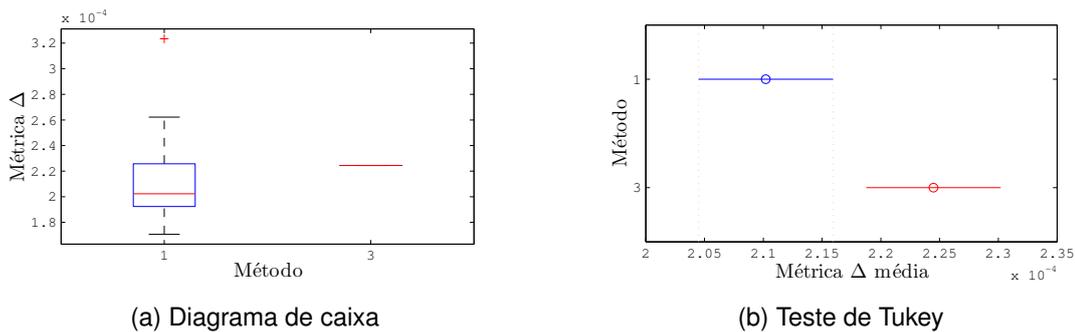


Figura 36 – Diagrama de caixa e teste de Tukey para métrica  $\Delta$  dos Métodos 1 e 3

Para o hipervolume, a Figura 32 consegue comparar os dois métodos e revelar a superioridade do método exato (Método 3). Porém, tanto o hipervolume quanto a métrica  $\Delta$  apresentadas pelos dois métodos são muito próximas, como pode ser observado na proximidade das soluções nas fronteiras de Pareto apresentadas na Figura 37, em que grande parte das soluções dos dois métodos estão visualmente sobrepostas.

É importante ressaltar que o Método 3, exato, projetado para solucionar o Modelo 3, proporciona soluções ótimas para este modelo, mas não proporciona soluções ótimas para o Modelo 1, apesar de proporcionar boas soluções também para este modelo, como indicam os resultados obtidos, uma vez que suas soluções são um pouco melhores que aquelas proporcionadas pelo Método 1, heurístico, projetado para solucionar o Modelo 1.

Considerando, porém, o tempo de execução dos métodos, uma importante métrica a ser analisada, o Método 3 também é executado 30 vezes e os tempos de execução de cada execução são registrados. O diagrama de caixa da Figura 38 mostra a grande discrepância entre o tempo gasto pelos métodos exato e heurístico. Enquanto o Método 3 possui tempos

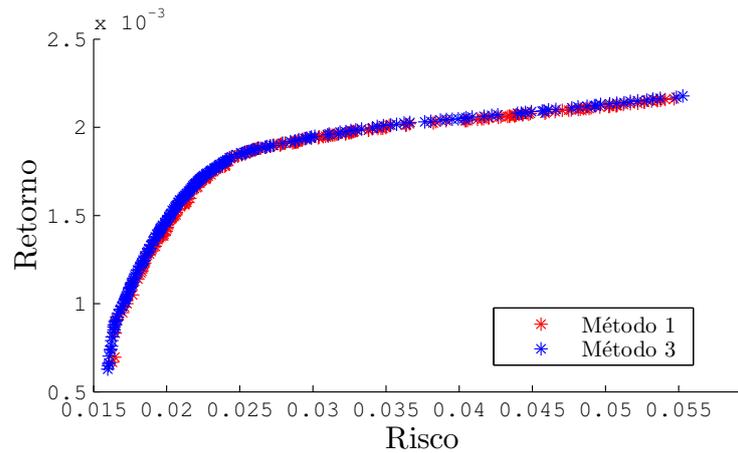


Figura 37 – Fronteiras de Pareto dos Métodos 1 e 3

de execução na ordem de dezenas de minutos, o Método 1 apresenta tempos de execução na ordem de dezenas de segundos. É importante notar que o método exato gera  $N$  soluções de forma sequencial, enquanto que o Método 1 utiliza um procedimento paralelo.

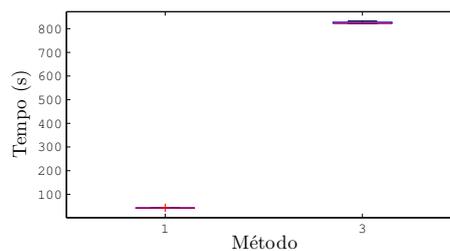


Figura 38 – Tempo de execução para os Métodos 1 e 3

Esse curto tempo de execução do Método 1 permite que ele seja utilizado em investimentos com operações de alta frequência, que requerem alterações de portfólios a cada período de poucos minutos. Além disso, mesmo em operações com frequências menores o Método 1 pode ser preferível sobre o Método 3 pela maior agilidade da execução. Por fim, constata-se que os resultados *in-sample* não são suficientes para a escolha de um desses métodos para todos os casos e, portanto, os dois métodos serão novamente comparados em testes *out-of-sample*.

#### 5.1.1.3 Análise do efeito da cardinalidade dos portfólios

A análise do efeito da cardinalidade dos portfólios na fronteira de Pareto dos métodos de otimização considerados, é feita de forma visual a partir de fronteiras de Pareto combinadas das 30 execuções de cada método analisado. Analisa-se as posições dos portfólios nessa fronteira conforme a sua quantidade de ativos.

Analisando a Figura 39, que mostra a fronteira de Pareto do Método 1 com as soluções coloridas de acordo com suas cardinalidades, observa-se que são escolhidos

portfólios com no máximo 8 ativos, mesmo que 25 ativos sejam permitidos pelo algoritmo, o que condiz com os resultados obtidos por [Cesarone, Moretti e Tardella \(2016\)](#), que conclui que os portfólios de maiores desempenho possuem não mais que 15 ativos.

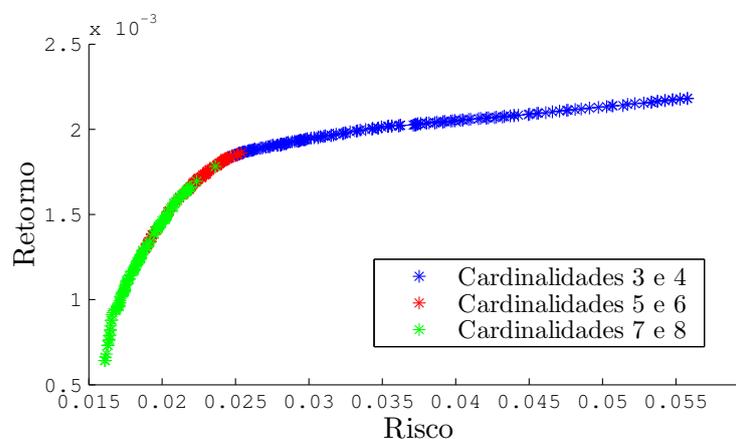


Figura 39 – Fronteira de Pareto do Método 1 colorida conforme as cardinalidades das soluções

É interessante notar também a separação de grupos de portfólios com valores similares de cardinalidades na fronteira de Pareto, de forma que os portfólios de alta quantidade de ativos (7 ou 8) tendem a estar na região de menores valores de risco e retorno na fronteira, portfólios com baixa quantidade de ativos (3 ou 4) tendem a estar na região de maiores risco e retorno e portfólios com valores medianos de cardinalidade (5 ou 6) tendem a estar em uma região intermediária da fronteira. Esse comportamento coincide com a ideia central da Teoria Moderna do Portfólio, que teve início no trabalho de [Markowitz \(1952\)](#), de que a diversidade de ativos em um portfólio reduz seu risco.

Para o Método 3, o mesmo padrão de localidade das soluções na fronteira de Pareto do Método 1 (Figura 39), de acordo com suas cardinalidades, é observado na Figura 40.

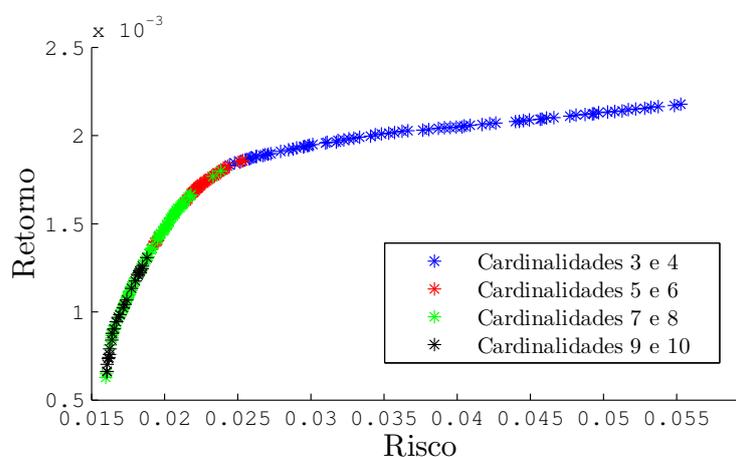


Figura 40 – Fronteira de Pareto do Método 3 colorida conforme as cardinalidades das soluções

Nesse caso, observa-se soluções de até 10 ativos, enquanto que o Método 1 proporcionou soluções de até 8 ativos, o que possivelmente se deve à maior quantidade possível de combinações para portfólios de 9 e 10 ativos, aumentando o espaço de busca de soluções nessas condições e, conseqüentemente, a dificuldade da meta-heurística em encontrar boas soluções. Contudo, não se pode afirmar que a falta de soluções de cardinalidade 9 e 10 nas soluções do Método 1 implique em sua incapacidade de produzir soluções de baixo risco. De fato, a solução com menor risco observado para o Método 3 possui cardinalidade 7 ou 8.

### 5.1.2 Experimentos *out-of-sample*

As análises *out-of-sample* visam comparar ganhos financeiros proporcionados pelos métodos heurístico (Método 1) e exato (Método 3) utilizando conjunto de dados de entrada com diferentes granularidades. Além disso, os efeitos dos custos de transação nesses ganhos são analisados. Essas análises são feitas por meio de simulações de operações no Mercado de Ações considerando os aspectos práticos do processo de investimento financeiro.

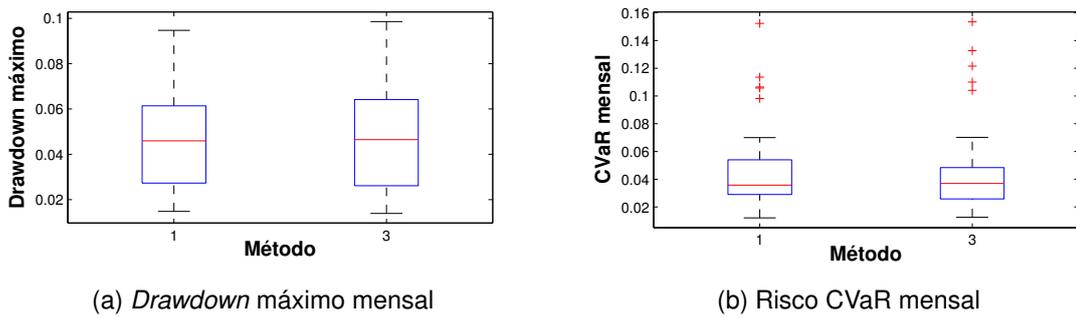
#### 5.1.2.1 Análise dos métodos heurístico e exato

Primeiramente, os desempenhos dos Métodos 1 e 3 são comparados em uma simulação de operações no Mercado de Ações utilizando o conjunto de dados de entrada 1, com periodicidade diária de informações. Para essa comparação, são considerados o *drawdown* máximo mensal, risco CVaR mensal, retorno mensal e retorno acumulado durante todo o período analisado, ou seja, de janeiro de 2014 a dezembro de 2016.

A Figura 41a apresenta os valores de *drawdown* máximo mensal dos dois métodos e a Figura 41b mostra os valores de risco CVaR mensal. As figuras mostram diagramas de caixa com diferenças muito pequenas entre os dois métodos para ambas as métricas. Por isso, testes ANOVA são aplicados para a comparação estatística dos métodos e esses testes falham em rejeitar a hipótese de igualdade da média populacional para ambas as métricas, sendo que o p-valor obtido para o *drawdown* máximo é de 0,9758 e, para o risco CVaR, 0,897.

Para o retorno mensal observado, os métodos também apresentam valores muito semelhantes, como pode ser observado na Figura 42. Novamente, o teste ANOVA aplicado falha em rejeitar a hipótese nula, que supõe a igualdade da média populacional de valores de retorno mensal dos Métodos 1 e 3. Neste caso, o p-valor apresentado é de 0,977.

Por fim, a pequena diferença no desempenho dos dois métodos também é facilmente percebida na Figura 43, que apresenta os retornos acumulados dos métodos e os compara com o retorno acumulado do índice Ibovespa, considerado o indicador mais importante



(a) Drawdown máximo mensal

(b) Risco CVaR mensal

Figura 41 – Drawdown máximo e CVaR mensal dos Métodos 1 e 3

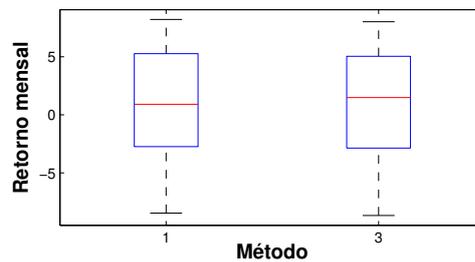


Figura 42 – Retorno mensal dos Métodos 1 e 3

do desempenho médio das cotações das ações negociadas na B3. Ainda na Figura 43, observa-se que os retornos de ambos os métodos superam o retorno do Ibovespa durante grande parte do período analisado, além de apresentarem ganhos mesmo em período de queda do Ibovespa, indicando uma boa eficiência do processo de otimização proposto.



Figura 43 – Retornos acumulados dos Métodos 1 e 3

Os resultados obtidos indicam que os Métodos 1 e 3 apresentam desempenho muito similares para as métricas analisadas nos testes *in-sample* e *out-of-sample*, exceto para o tempo de execução, em que o Método 1, heurístico, apresenta uma vantagem discrepante em relação ao Método 3. Por isso, o Método 1 é o método a ser utilizado nos próximos testes e na etapa de combinação da otimização de portfólio e classificação de tendências financeiras.

### 5.1.2.2 Efeito dos custos de transação nos ganhos financeiros

O custo de transação total, neste trabalho, é composto por uma parte fixa ou independente, expressa em unidade monetária (reais) e uma parte variável, que representa uma proporção do capital total investido no portfólio. Nota-se que a parte fixa do custo de transação se torna proporcionalmente menor, em relação à parte variável, com o aumento do capital total de negociação dos ativos. Para esta análise, ambos os custos são aumentados e os efeitos sobre o ganho proporcionado pelo Método 1, para diferentes valores de custos fixo e proporcional, são avaliados.

Os diferentes valores de custos de transação são determinados com base nos valores já definidos para o custo fixo  $f$  e variável  $\beta$ . Os valores de  $f$  e  $\beta$  utilizados em testes anteriores definem o custo de transação total denominado aqui de 1x. Os outros valores alteram esse valor de forma exponencial: o custo total dez vezes maior que o original (denominado 10x) possui custo fixo igual a  $10f$  e variável igual a  $10\beta$ , o custo 100x é cem vezes maior que o original e o custo 1000x, mil vezes maior. A Figura 44 apresenta os retornos acumulados para cada custo de transação total.



Figura 44 – Retornos acumulados para diferentes valores de custos de transação

Ao final do período analisado, os retornos proporcionados quando se utiliza custos menores, são maiores, como esperado, já que quanto maior o custo de transação, maior é a parte do capital final que será consumida por este, deixando de constituir parte do ganho financeiro. Mas, apesar dessa diferença, é possível perceber um padrão nos movimentos de subida e descida desses retornos acumulados, mesmo que o método selecione portfólios completamente diferentes para cada valor de custo de transação total.

É interessante notar também a proximidade dos retornos acumulados para custos de transação até cem vezes maiores que o original (1x). Esses padrões observados e a baixa sensibilidade do método proposto à variação do custo de transação indicam a robustez desse processo de otimização, além da eficácia do uso do rebalanceamento desses custos, como é considerado no modelo proposto.

Analisando as cardinalidades dos portfólios selecionados pelo Método 1, para os diferentes valores de custos de transação, os diagramas de caixa da cardinalidade do portfólio selecionado para cada custo, mostrados na Figura 45, também revelam similaridade dos portfólios em relação à quantidade de ativos para custos até cem vezes maiores que o original. Por outro lado, quando o custo 1000x é utilizado, os portfólios utilizam quantidades menores de ativos, apresentando uma variabilidade menor para esses valores, além de uma mediana também muito menor que as demais.

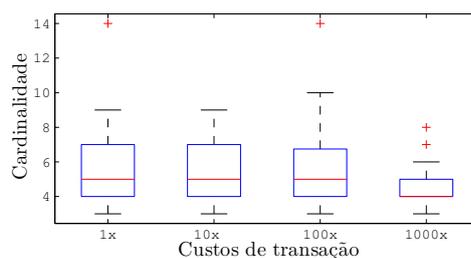


Figura 45 – Cardinalidade do portfólio selecionado para diferentes custos de transação

A diminuição da cardinalidade dos portfólios com o aumento do custo de transação é esperado, já que o custo de se trocar um ativo ou adquirir um novo pode se tornar maior que uma perda esperada de um ativo já adquirido. O teste ANOVA, porém, não rejeita a hipótese nula de igualdade da média de cardinalidade para todos os custos de transação, considerando um nível de confiança de 95% (o teste apresenta um p-valor de 0,0724).

### 5.1.2.3 Efeito das diferentes granularidades da série histórica nos ganhos financeiros

Os riscos e retornos proporcionados pelo Método 1 utilizando o conjunto de dados 1, com granularidade diária, e o conjunto de dados 2, com granularidade horária, são comparados entre si e com os valores apresentados pelos *benchmarks* Ibovespa e CDI (Certificado de Depósito Interbancário), comumente utilizado como uma taxa livre de risco.

Primeiramente, a Figura 46 apresenta o retorno acumulado do Método 1 utilizando os conjuntos de dados 1 e 2, com os retornos do Ibovespa e do CDI. A partir da figura, nota-se que durante o período específico de janeiro de 2014 até o final de 2015, os retornos acumulados da simulação utilizando o conjunto de dados 1 são maiores que os retornos acumulados quando se utiliza o conjunto de dados 2. No entanto, o retorno acumulado para o conjunto de dados 2 ultrapassa o retorno para o conjunto 1, se tornando muito maior que este ao final de 2016.

De toda forma, ambos os conjuntos de dados proporcionam um retorno acumulado superior ao do Ibovespa durante a maior parte do período analisado e o conjunto de dados 1 proporciona o melhor retorno acumulado por maior parte do período.

Em relação ao retorno mensal, os diagramas de caixa na Figura 47 mostram distri-



Figura 46 – Retornos acumulados para os conjuntos de dados 1, 2 e *benchmarks*

buições similares quando utilizados os conjuntos de dados 1 e 2, ou mesmo para o Ibovespa. Os valores de retorno mensal para o CDI, que apresentam baixa variabilidade, se encontram próximos à mediana desse risco para o método de otimização com os diferentes conjuntos de dados e para o Ibovespa. O teste ANOVA falha em detectar uma diferença estatística entre os retornos, apresentando um p-valor de 0,9379.

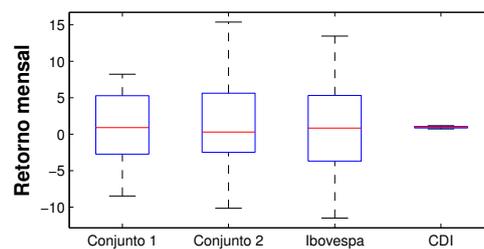


Figura 47 – Retorno mensal para os conjuntos de dados 1, 2 e *benchmarks*

Considerando o CDI como uma taxa livre de riscos, as análises de *drawdown* máximo de CVaR mensal desconsideram esse *benchmark*. Assim, a Figura 48 apresenta os diagramas de caixa de *drawdown* máximo mensal para o Método 1 utilizando os conjuntos de dados 1 e 2, e para o Ibovespa. Os diagramas apresentados na figura possuem as suas caixas sobrepostas e, por isso, são insuficientes para a comparação dos valores. Então, o teste ANOVA é aplicado e falha em detectar a diferença entre as diferentes abordagens (considerando um nível de confiança de 95%), com um p-valor de 0,0807.

Já para o risco CVaR, os diagramas de caixa, apresentados na Figure 49a, revelam menores valores e variabilidade para o conjunto de dados 1, quando comparados com o conjunto de dados 2. Tal indicação é confirmada pelo teste ANOVA, que detecta diferença estatística para os valores de risco das diferentes abordagens com um nível de confiança de 95%. Portanto, o teste de comparação múltipla de Tukey é aplicado e identifica os riscos para o conjunto 1 como sendo estatisticamente menores que os riscos apresentados pelo conjunto 2, para o mesmo nível de confiança, como mostra a Figura 49b. O método de

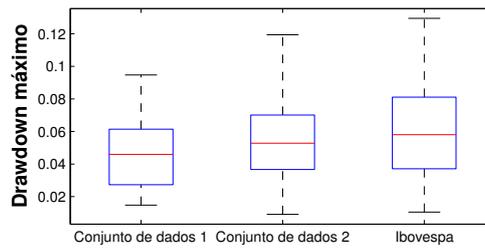


Figura 48 – Drawdown máximo mensal para os conjuntos de dados 1, 2 e benchmarks

otimização com ambos os conjuntos de dados não apresentam diferenças estatísticas para o risco CVaR em relação ao Ibovespa.

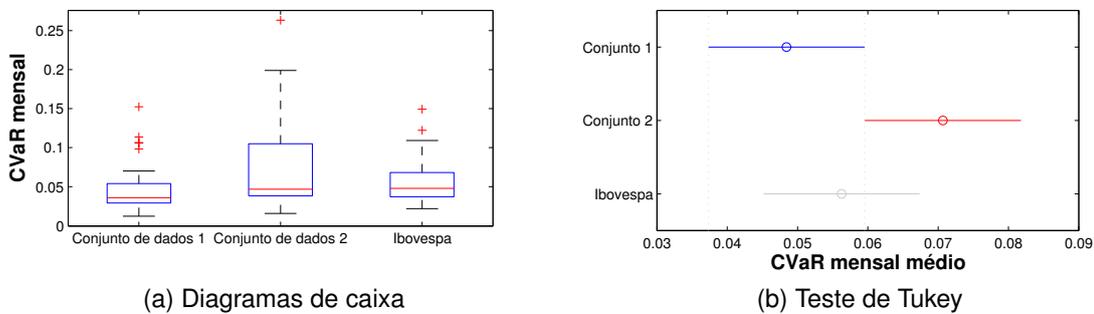


Figura 49 – CVaR mensal para os conjuntos de dados 1, 2 e benchmarks

O maior valor de risco apresentado quando se utiliza o conjunto de dados 1 se deve ao fato de que a utilização de apenas 1 mês de série histórica aumenta a chance de que esse período represente um comportamento esporádico, que não se mantenha no futuro, enquanto que o intervalo de 1 ano de série histórica, utilizado no conjunto de dados 2, tende a representar melhor comportamentos de longo prazo nos movimentos dos preços dos ativos. Esse menor risco para o conjunto 1 proporciona descidas mais suaves ao método, o que possivelmente fez com que o retorno acumulado para essa configuração superasse o método com o conjunto 2 em grande parte do período, mesmo que apresente um retorno menor ao final do período.

Esse comportamento mais incerto quando se utiliza o conjunto de dados 2 pode ser observado na Figura 50, que apresenta os retornos acumulados anuais separadamente. Em 2014, por exemplo, a Figura 50a mostra que o Método 1 utilizando o conjunto de dados 2 proporciona ganhos sempre menores que os do CDI, enquanto que em 2016 o método utilizando esse mesmo conjunto de dados proporciona ganhos maiores que os do CDI para quase todo o período, como mostra a Figura 50c.

Especificamente para o ano de 2014, a Figura 50a apresenta oscilações para o conjunto de dados 2 muito similares às do Ibovespa, enquanto que o conjunto 1 supera ambos em quase todo o ano. Já para o ano de 2015, a Figura 50b mostra uma proximidade dos

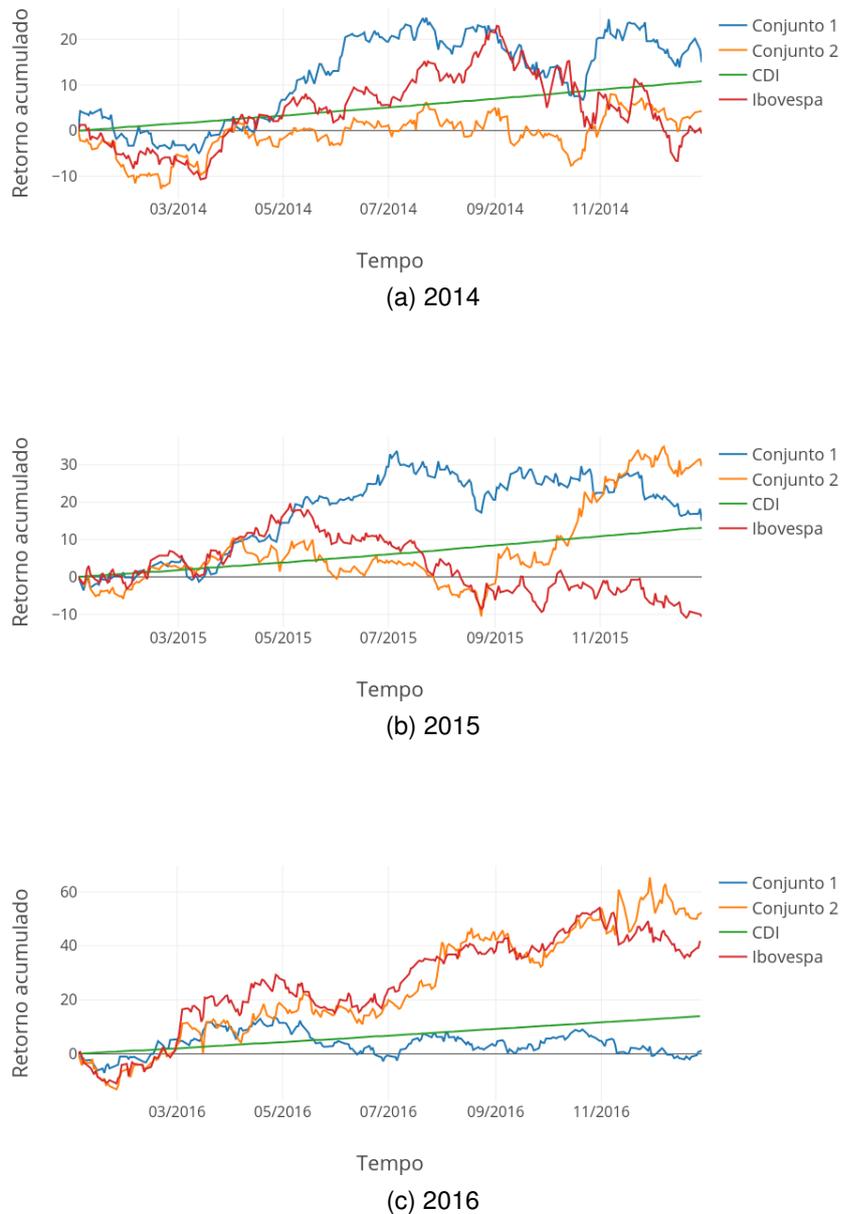


Figura 50 – Retornos acumulados anuais para os conjuntos de dados 1, 2 e *benchmarks*

retornos para o conjunto de dados 2 e Ibovespa próximos até o mês de setembro, quando o retorno acumulado para o conjunto 2 apresentam um aumento repentino, superando o retorno para o conjunto 1. Em 2016, por fim, a Figura 50c mostra um bom desempenho do retorno do Ibovespa, acompanhado por um aumento do conjunto 2 durante todo o ano, enquanto que o conjunto de dados 1 não consegue acompanhar esse aumento, apresentando um comportamento mais estável, e é muito superado pelas outras abordagens.

O comportamento menos estável quando se utiliza o conjunto de dados 2, que se deve a sua maior frequência de negociações e utilização apenas de dados mais recentes, como já mencionado, faz com que o método de otimização e de negociação consiga acom-

panhar melhor o desempenho do mercado, representado pelo desempenho do Ibovespa. Analisando a correlação dos retornos proporcionado pelo método utilizando cada um dos conjuntos de dados com os retornos do Ibovespa, nota-se que realmente o conjunto de dados 2 possui a maior correlação. O coeficiente de correlação de Pearson entre os retornos para o conjunto 2 e os do Ibovespa é de 0,302, enquanto que para o conjunto de dados 1, é de 0,117.

## 5.2 Classificação de séries temporais financeiras

Enquanto alguns hiperparâmetros dos algoritmos de classificação são determinados pela etapa de validação, que avalia o desempenho do algoritmo em um determinado período utilizando cada valor dentre o conjunto de valores considerados para o parâmetro, alguns hiperparâmetros assumem valores fixos, que correspondem aos valores padrões determinados pela biblioteca scikit-learn (SCIKIT-LEARN, 2019). Os conjuntos de valores definidos para os parâmetros determinados por validação são descritos na subseção 4.3.4.

Para a Regressão Logística (com regularização LASSO ou *Ridge*), a tolerância para o gradiente do erro (Entropia Cruzada) utilizada como critério de parada do algoritmo de otimização é de 0,0001 e o número máximo de iterações para esse algoritmo é 100.

Para a Floresta Aleatória, a quantidade mínima de amostras necessária para que um nó possa se dividir em dois outros nós é 2, a quantidade mínima de amostras presente em um nó folha é 1 e a impureza mínima para que ocorra uma divisão em um nó, igual a 0,0000001.

A Máquina de Vetores de Suporte utiliza uma tolerância do gradiente do erro para o critério de parada do algoritmo igual a 0,001 e não é determinada uma quantidade máxima de iterações para esse algoritmo.

Para o Perceptron Multicamadas, utiliza-se uma taxa de aprendizado, para o algoritmo de otimização, igual a 0,001, tolerância para o valor do gradiente do erro de 0,0001 e número máximo de iterações para esse algoritmo, igual a 200. Esses valores fixos são valores ótimos encontrados no trabalho de Itano, Sousa e Del-Moral-Hernandez (2018). A Rede Neural Convolutiva utiliza esses mesmos valores em seu algoritmo de otimização do erro.

Os classificadores são comparados com procedimentos básicos (*benchmarks*) que prediz a classe de todas as amostras de entrada como sendo aquela que possui uma maior quantidade de elementos ou de forma aleatória. Visando uma facilitação na apresentação dos resultados, os algoritmos de classificação são numerados da seguinte forma:

- Algoritmo 1: Floresta Aleatória;
- Algoritmo 2: Máquina de Vetores de Suporte;

- Algoritmo 3: Perceptron Multicamadas;
- Algoritmo 4: Rede Neural Convolutacional;
- Algoritmo 5: Regressão Logística com regularização *Ridge*;
- Algoritmo 6: Procedimento que prediz todas as amostras como sendo da classe com maior quantidade de ocorrências;
- Algoritmo 7: Procedimento que classifica as amostras aleatoriamente;
- Algoritmo 8: Comitê de classificadores utilizando votação majoritária;
- Algoritmo 9: Comitê de classificadores utilizando votação ponderada;
- Algoritmo 10: Empilhamento utilizando a Máquina de Vetores de Suporte como classificador nível 1;
- Algoritmo 11: Empilhamento utilizando o Perceptron Multicamadas como classificador nível 1;
- Algoritmo 12: Adaboost;
- Algoritmo 13: XGBoost;
- Algoritmo 14: Programação genética.

### 5.2.1 Análise de desempenho

Os experimentos de análise de desempenho dos algoritmos se inicia com ajustes de parâmetros como tamanhos dos conjuntos de dados de treinamento e teste, além da escolha dos métodos de seleção de atributos a serem utilizados. Para isso, o algoritmo de classificação, a regra de saída e o conjunto de dados de entrada são fixados, sendo que são selecionados o algoritmo Floresta Aleatória, por ser um dos algoritmos que realizam seleção de atributos, a regra de saída 1 e o conjunto de dados de entrada 1, por serem os mais utilizados em trabalhos relacionados presentes na literatura. Por fim, fixa-se os valores determinados para os tamanhos dos conjuntos de treino, teste e os métodos de seleção de atributos para que os algoritmos de classificação considerados neste trabalho sejam comparados para cada regra de saída e conjunto de dados de entrada.

#### 5.2.1.1 Ajuste dos tamanhos dos conjuntos de dados de treino e teste

Na realização do ajuste dos tamanhos dos conjuntos de treino e teste, cada tamanho de teste e cada respectivo tamanho de treino obtido por cada um dos multiplicadores são comparados em relação às métricas acurácia, precisão, revocação e F1. Para isso, os fatores tamanho do conjunto de teste e multiplicador desse tamanho na determinação do tamanho do conjunto de teste são analisados em testes ANOVA de dois fatores. Deve-se notar que valores de p-valores iguais a 0 ou 1 nesses testes, são valores arredondados para quatro casas decimais.

Em relação à acurácia, nenhum dos fatores apresentam influência sobre a métrica, para os níveis dos fatores considerados. Para o tamanho dos conjuntos de teste, o p-valor

obtido é de 0,9902 e para o multiplicador de treino, 0,5627. Também não foi constatada influência da combinação dos dois fatores, para um nível de confiança de 95%, já que é apresentado um p-valor igual a 1.

Para a precisão, é detectada uma diferença estatística entre os diferentes valores de tamanho do conjunto de teste ( $p$ -valor = 0). A Figura 51, que mostra o teste de comparação múltipla de Tukey para os tamanhos dos conjuntos de teste, revelam que a precisão utilizando 10 períodos para teste, por exemplo, apresenta precisão maior que o classificador proporciona quando utiliza 70, 90 ou 100 períodos. Os resultados também indicam que a precisão diminui com o aumento do tamanho do conjunto de teste, como esperado, já que intervalos menores de teste possuem mais proximidade com os períodos de treino e, portanto, maior chance de apresentarem os mesmos padrões presentes no treino. Não são detectadas diferenças para os diferentes valores do multiplicador que determina o tamanho do conjunto de treino ( $p$ -valor = 0,6686) e para a combinação dos fatores ( $p$ -valor = 0,7245).

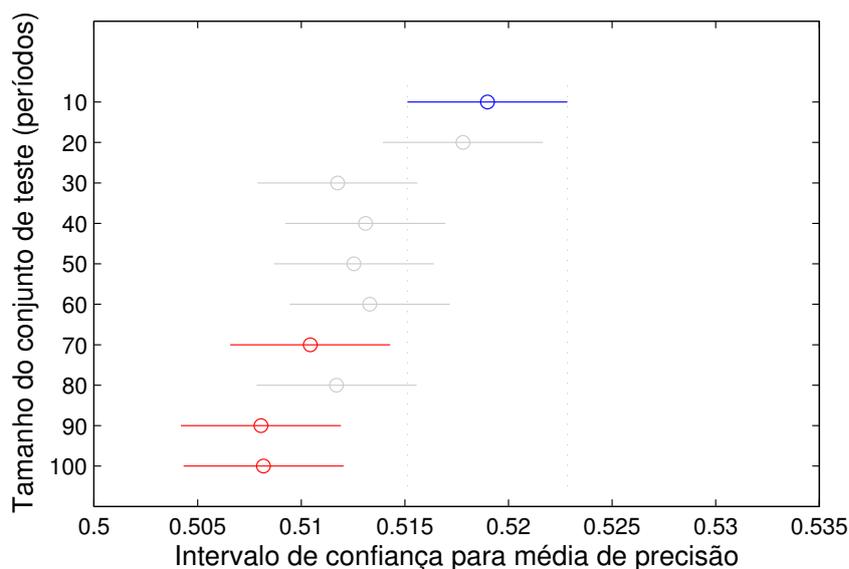


Figura 51 – Teste de Tukey de precisão para os tamanhos dos conjuntos de teste

Já para a revocação, resultados indicam diferenças estatísticas, considerando um nível de confiança de 95%, tanto para o tamanho do conjunto de teste quanto para o multiplicador que define o tamanho do conjunto de treino, com  $p$ -valor igual a 0 para ambos os fatores. A Figura 52 mostra o teste de comparação múltipla de Tukey para os tamanhos dos conjuntos de teste e, também para a revocação, quanto menor o tamanho do conjunto de teste, maior o valor da métrica. Nesse caso, observa-se que o teste com 10 períodos de dados apresenta revocação melhor que o teste com o classificador utilizando de 30 a 100 períodos. Esse comportamento também se deve à proximidade dos dados de teste com os dados de treino.

A comparação dos diferentes multiplicadores que definem o tamanho do conjunto de treino é apresentada na Figura 53 e pode-se perceber que a revocação também aumenta

conforme o aumento no tamanho do conjunto de treino, possivelmente devido à maior facilidade em detectar padrões de movimento de preços de ativos de curto prazo. Nesse teste, o multiplicador de uma vez (1x) apresenta revocação maior que tamanhos de treino definidos pelos multiplicadores de quatro (4x) ou dez vezes (10x). Por fim, não é detectada diferença para a combinação dos fatores, em relação à revocação, sendo que o teste ANOVA apresenta p-valor de 1.

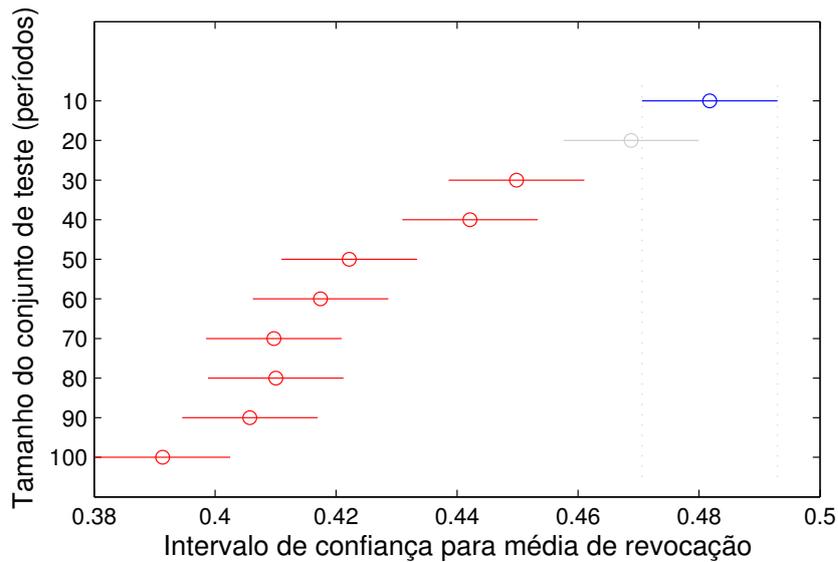


Figura 52 – Teste de Tukey de revocação para os tamanhos dos conjuntos de teste

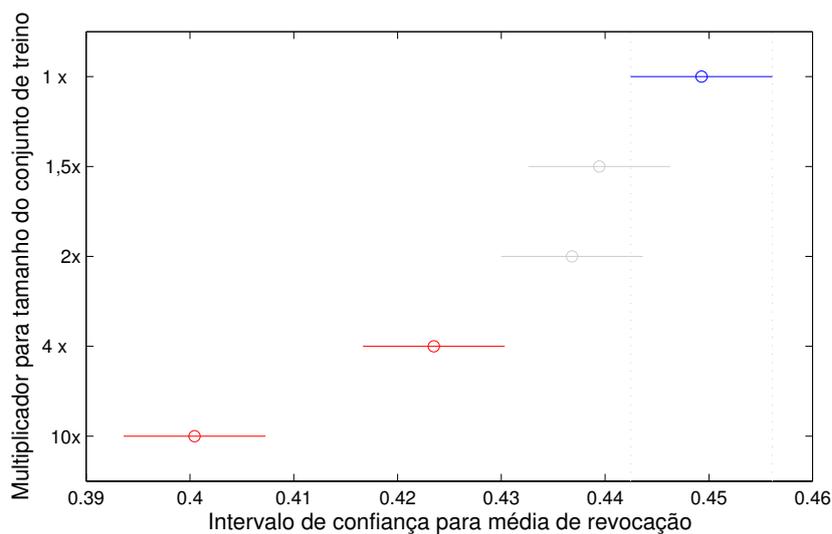


Figura 53 – Teste de Tukey de revocação para os tamanhos dos conjuntos de treino

Combinando a revocação com a precisão, é fácil observar que valores menores para os tamanhos dos conjuntos de treino e teste apresentam os maiores desempenho. Como esperado, então, a ANOVA para F1 indica diferenças estatísticas para os dois fatores, com p-valor de 0 para ambos, e os testes de comparação múltipla sugerem melhores

desempenhos para menores tamanhos de treino e teste. A Figura 54 apresenta teste de comparação múltipla para o tamanho do conjunto de teste e a Figura 55, para o multiplicador que determina o tamanho do conjunto de treino. Para a combinação dos fatores, novamente não é detectada influência sobre a métrica, já que o teste apresenta um p-valor igual a 1.

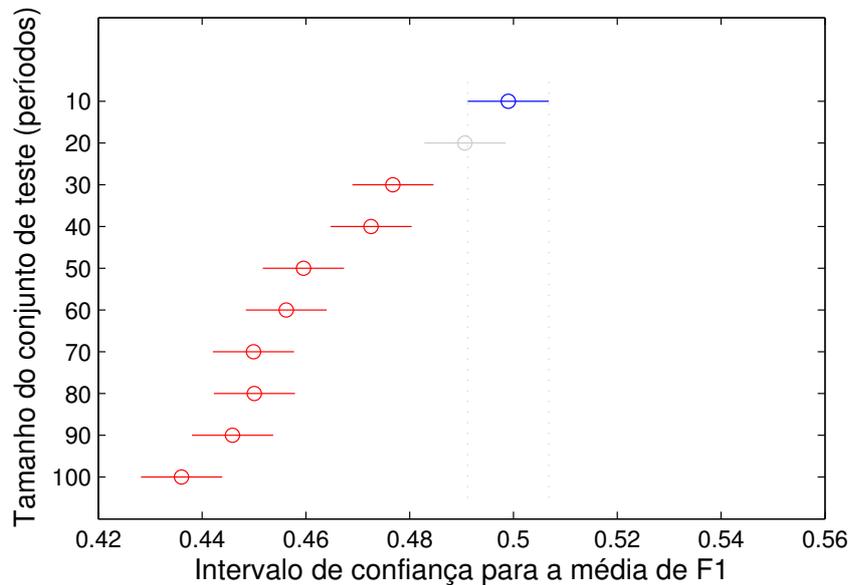


Figura 54 – Teste de Tukey de F1 para os tamanhos dos conjuntos de teste

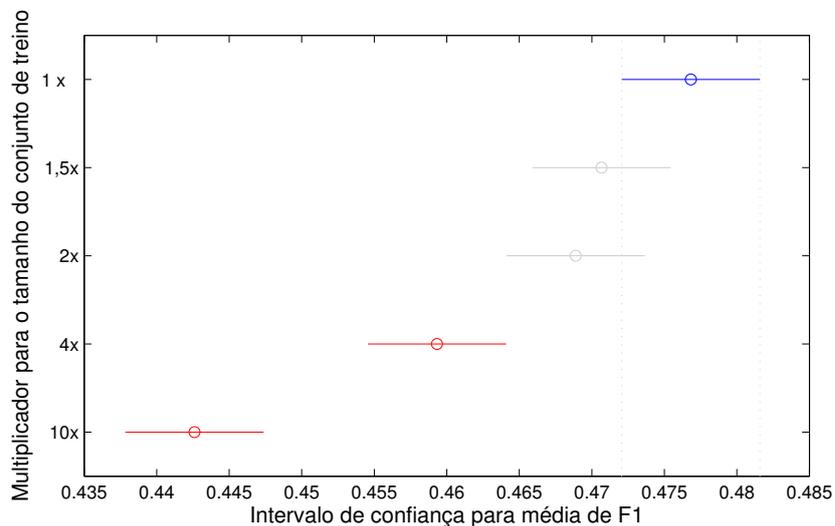


Figura 55 – Teste de Tukey de F1 para os tamanhos dos conjuntos de treino

Com base nos resultados e considerando que tanto a precisão quanto a revocação aumentam com a diminuição dos tamanhos dos conjuntos de treino e teste, são selecionados os maiores valores que não apresentam diferenças estatísticas em relação aos valores mínimos desses tamanhos, ou seja, 10 períodos de teste e o tamanho do conjunto de treino igual ao tamanho de treino. A ideia é minimizar o tempo de execução dos algoritmos de classificação sem afetar o desempenho destes. Então, é selecionado um tamanho de

conjunto de teste de 20 períodos e, um multiplicador igual a duas vezes o tamanho do conjunto de treino. Assim, os próximos testes possuem um treino constituído de 40 períodos de dados e um teste de 20 períodos. O conjunto de validação também possui um tamanho de 20 períodos.

### 5.2.1.2 Análise dos métodos de seleção de atributos

A análise dos métodos de seleção de atributos visa a comparação de várias combinações das cinco formas consideradas de seleção de atributos utilizadas na etapa de validação, com o intuito de reduzir essas combinações, reduzindo o tempo total do processo de validação. Priorizando os métodos embarcados, por serem os que mais reduzem as quantidades de atributos, diminuindo também, o tempo de treinamento do algoritmo de classificação, um primeiro teste compara três métodos de seleção de atributos:

- Método de seleção de atributos 1: utiliza todas as cinco formas de seleção de atributos, entre os filtros e métodos embarcados, além de considerar o conjunto formado por todos os 60 atributos. A melhor forma a ser utilizada em um determinado período é determinado pela validação;
- Método de seleção de atributos 2: utiliza apenas os atributos selecionados pelo método embarcado que utiliza a avaliação da importância de cada atributo pelo algoritmo da Floresta Aleatória;
- Método de seleção de atributos 3: utiliza apenas os atributos selecionados pelo algoritmo de Regressão Logística com regularização LASSO.

Para a comparação de desempenho desses três métodos, testes ANOVA são aplicados para as métricas acurácia, precisão, revocação e F1, que reportam p-valores de, respectivamente, 0,9749; 0,3368; 0,9439 e 0,7455. A Figura 56 mostra essas pequenas diferenças de desempenho utilizando os diferentes métodos de seleção de atributos, sendo que a Figura 56a apresenta os diagramas de caixa para acurácia, a Figura 56b, para precisão, a Figura 56c, para revocação e a Figura 56d para F1. Dessa forma, não se pode afirmar que existe alguma diferença estatística quando diferentes métodos são utilizados. Dessa forma, o método de seleção de atributos 3 é selecionado para os demais testes por ser o mais rápido dos três métodos e por ser um dos métodos que mais reduz a quantidade de atributos a ser utilizados.

Utilizando o método de seleção de atributos 3, os percentuais de utilização de cada atributo em todo o período considerado é mostrado na Tabela 2. Esses percentuais consideram a quantidade de vezes em que o algoritmo de Regressão Logística com regularização LASSO atribuiu um peso diferente de 0 para o atributo em relação à quantidade total de vezes em que o algoritmo foi executado.

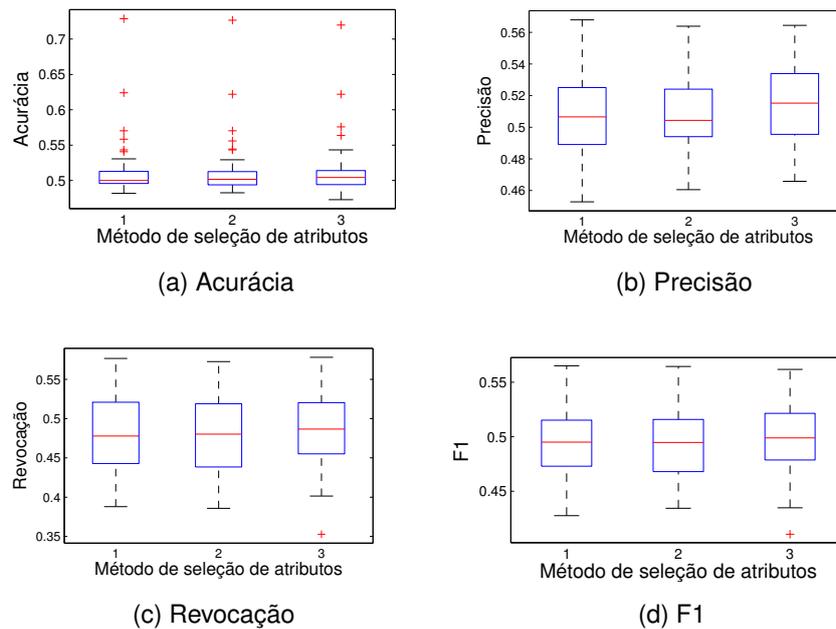


Figura 56 – Desempenho dos métodos de seleção de atributos

### 5.2.1.3 Regra 1

Tanto para a regra de saída 1, quanto para as outras regras de saída, testes ANOVA são realizados para a acurácia dos diferentes algoritmos, sendo que valores abaixo de  $10^{-4}$  são considerados iguais 0 0. A acurácia é considerada a métrica mais importante, uma vez que classificações de boa acurácia podem indicar momentos assertivos de compra e venda de ações. Quando não se identifica um algoritmo de melhor acurácia, a precisão e revocação não utilizados como métrica de comparação. Para a precisão e revocação, o *benchmark* majoritário (Algoritmo 6) não é utilizado, pois a classe 0 é a classe da maioria das amostras em todas as regras de saída. Dessa forma, o Algoritmo 6 não está presente em nenhuma comparação de algoritmos para essas métricas. A comparação de precisão e revocação é realizada utilizando um gráfico dos valores de precisão e revocação de cada algoritmo considerando cada limite de classificação, variando de 0 a 1. Esse limite determina a probabilidade a partir da qual os ativos são classificados entre 0 e 1. Para comparação de acurácia, esse limite é definido como 0,5.

Primeiramente, são apresentados os resultados utilizando o conjunto de dados de entrada 1, com periodicidade diária. Neste cenário, a acurácia apresentada por cada algoritmo de classificação é apresentada na Figura 57, que mostra que a acurácia do *benchmark* majoritário (Algoritmo 6) supera estatisticamente quase todos os algoritmos de classificação considerados. Sabendo que o algoritmo majoritário classifica todos os ativos como 0 (não promissor), pode-se afirmar que essa estratégia apresenta uma precisão igual a 0. Por isso, um teste ANOVA realizado excluindo o Algoritmo 6 apresenta um p-valor igual a 0,0586, indicando que nenhuma diferença estatística foi detectada entre os demais

Tabela 2 – Porcentagem de utilização dos atributos

Atributo	Utilização (%)	Atributo	Utilização (%)
1	31.391994	31	33.086909
2	20.158673	32	19.491525
3	40.822214	33	11.287414
4	2.921024	34	8.150018
5	45.943022	35	6.996033
6	23.692752	36	35.412910
7	49.873783	37	12.279120
8	32.365669	38	46.556076
9	40.948431	39	28.633249
10	46.574107	40	27.767761
11	34.276956	41	38.243779
12	31.806708	42	33.826181
13	24.774612	43	36.530833
14	49.098449	44	33.988460
15	21.565092	45	19.509556
16	15.506671	46	33.321313
17	56.022358	47	32.708258
18	3.858637	48	31.301839
19	57.338622	49	29.624955
20	34.547422	50	33.014785
21	59.231879	51	29.877389
22	20.176704	52	31.770646
23	36.386585	53	32.005049
24	14.424811	54	30.454382
25	56.815723	55	35.052290
26	18.121168	56	43.238370
27	57.933646	57	43.959611
28	34.835918	58	48.341147
29	60.133429	59	47.133069
30	10.728453	60	42.210602

algoritmos.

Uma vez que a acurácia não foi suficiente para determinar o melhor algoritmo, o gráfico da Figura 58 compara esses algoritmos em relação à precisão e revocação. A partir deste gráfico, nota-se que os Algoritmos 4 e 8 apresentam precisão maiores que os demais. O Algoritmo 8 é o comitê de classificadores utilizando voto majoritário e inclui vários algoritmos, incluindo o Algoritmo 4 (CNN). Assim, pode-se supor que o bom desempenho do Algoritmo 8 se deve ao bom desempenho do Algoritmo 4, escolhido como o melhor para a regra 1, por ser mais rápido que o Algoritmo 8.

Para o conjunto de entrada 2, com granularidade em horas, o Algoritmo 6 apresenta uma diferença ainda maior para os demais algoritmos, conforme mostra o gráfico da Figura 59. Mas excluindo este *benchmark*, os algoritmos não apresentam diferenças estatísticas

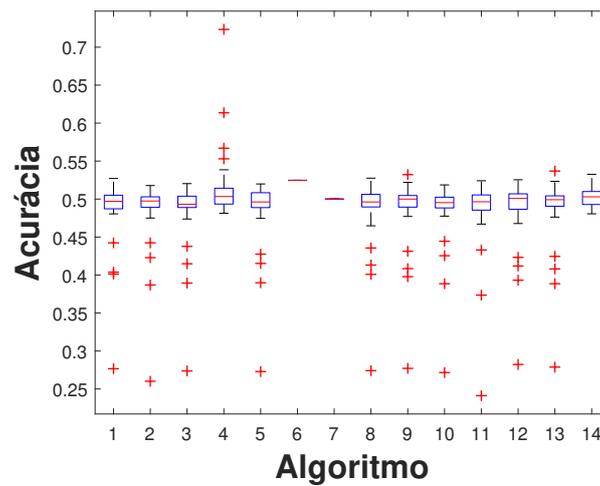


Figura 57 – Acurácia dos algoritmos para regra 1 e conjunto de dados de entrada 1



Figura 58 – Gráfico de Precisão por revocação para regra 1 e conjunto de dados de entrada 1

entre si, apresentado um p-valor de 0,4441 no teste ANOVA.

Mais uma vez, acurácia não foi suficiente para a definição do melhor algoritmo. Assim, esses algoritmos são comparados em termos de precisão e revocação e os resultados mostram uma superioridade do Algoritmo 12 (Adaboost) em relação aos demais, conforme mostrado pelo gráfico da Figura 60.

#### 5.2.1.4 Regra 2

Como para a regra de saída 1, primeiramente, são apresentados os resultados utilizando o conjunto de dados de entrada 1, para a regra de saída 2. A acurácia de cada algoritmo de classificação, apresentada na Figura 61, revela que a acurácia do

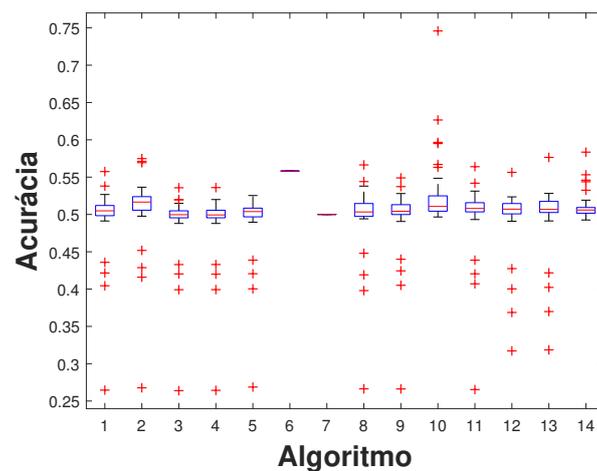


Figura 59 – Acurácia dos algoritmos para regra 1 e conjunto de dados de entrada 2



Figura 60 – Gráfico de precisão por revocação para regra 1 e conjunto de dados de entrada 2

*benchmark* majoritário (Algoritmo 6) apresenta um bom desempenho para acurácia, acima de 0,7, superando estatisticamente todos os algoritmos de classificação considerados. Considerando que poucas amostras (menos de 30%) pertencem à classe 1 nesta regra, o valor de acurácia elevado para o *benchmark* majoritário era esperado. Porém, na prática, este procedimento não é adequado já que ele não realiza nenhuma inferência para a classe 1. Para a comparação dos demais algoritmos, um teste ANOVA foi aplicado e indicou diferença estatística entre os algoritmos, com um p-valor muito baixo (abaixo de  $10^{-4}$ ).

Desconsiderando então os *benchmarks*, o Algoritmo 4 apresenta a melhor média de acurácia, conforme mostrado pelo teste de Tukey da Figura 62, que indica que o Algoritmo 4 supera os demais, estatisticamente, exceto o Algoritmo 1 (Floresta Aleatória) e o Algoritmo 13 (XGBoost).

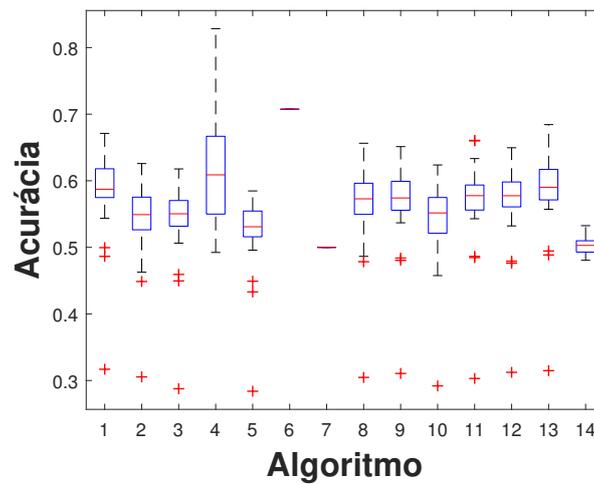


Figura 61 – Acurácia dos algoritmos para regra 2 e conjunto de dados de entrada 1

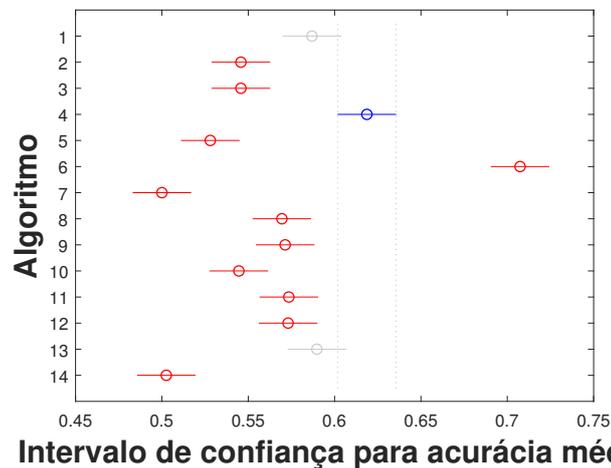


Figura 62 – Teste de acurácia para regra 2 e conjunto de dados de entrada 1

Comparando a precisão e revocação dos algoritmos, o gráfico da Figura 63 mostra um melhor desempenho da Rede Neural Convolutiva (Algoritmo 4) e do comitê de classificadores utilizando voto ponderado (Algoritmo 9). Uma vez que o comitê de classificadores depende do resultado da classificação do Algoritmo 4, sendo assim mais lento, no cenário da regra de saída 2 e conjunto de dados de entrada 1, o Algoritmo 4 também é escolhido como o melhor.

Os diagramas de caixa da Figura 64 revelam uma acurácia alta do Algoritmo 7, o que demonstra que o cenário da regra 2 e conjunto de dados de entrada 2 é muito desbalanceada, com aproximadamente 90% dos ativos sendo considerados não promissores (classe 0). Esses diagramas mostram valores melhores para os Algoritmos 1, 11 e 13, sem definir claramente uma diferença estatística entre os algoritmos.

Desconsiderando os *benchmarks*, os Algoritmos 11 (empilhamento utilizando o Perceptron Multicamadas como classificador nível 1) e 13 (XGBoost) apresentam as melhores



Figura 63 – Gráfico de precisão por revocação para regra 2 e conjunto de dados de entrada 1

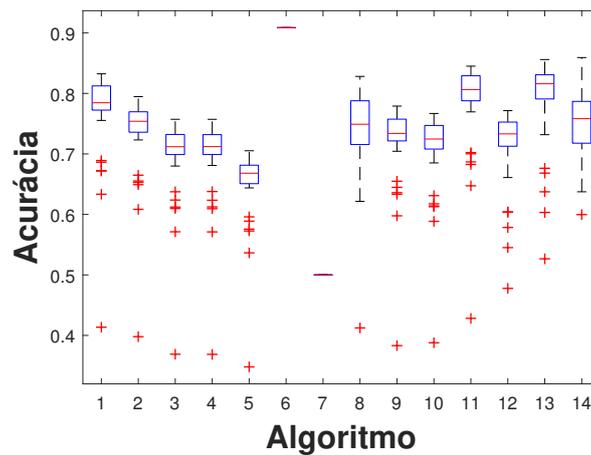


Figura 64 – Acurácia dos algoritmos para regra 2 e conjunto de dados de entrada 2

médias de acurácia, de acordo com o teste de Tukey da Figura 65, superando estatisticamente todos os algoritmos, exceto o Algoritmo 1 (Floresta Aleatória) e o Algoritmo 14 (Programação Genética).

Os Algoritmos 1 (Floresta Aleatória) e 12 (Adaboost) são os que apresentam maiores valores de precisão para diferentes valores de revocação, como mostra o gráfico da Figura 66. Como o Algoritmo 1 apresenta uma média de acurácia superior ao do Algoritmo 12, pode-se considerar o Algoritmo 1 como o melhor para este cenário.

### 5.2.1.5 Regra 3

Começando pelo conjunto de dados de entrada 1, a acurácia de cada algoritmo para a regra de saída 3 é apresentada na Figura 67 e revela que, assim como para a regra de

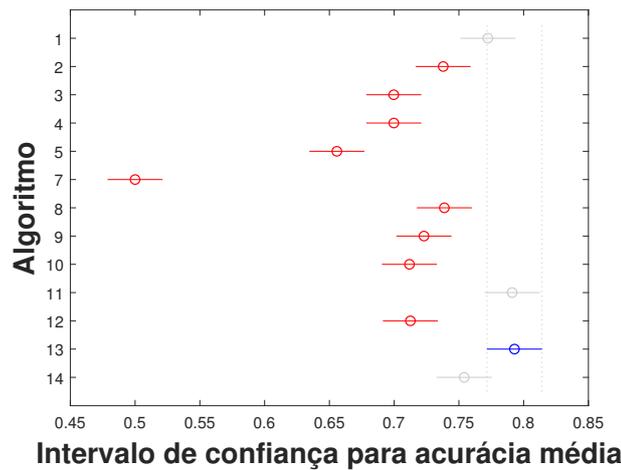


Figura 65 – Teste de acurácia para regra 2 e conjunto de dados de entrada 2

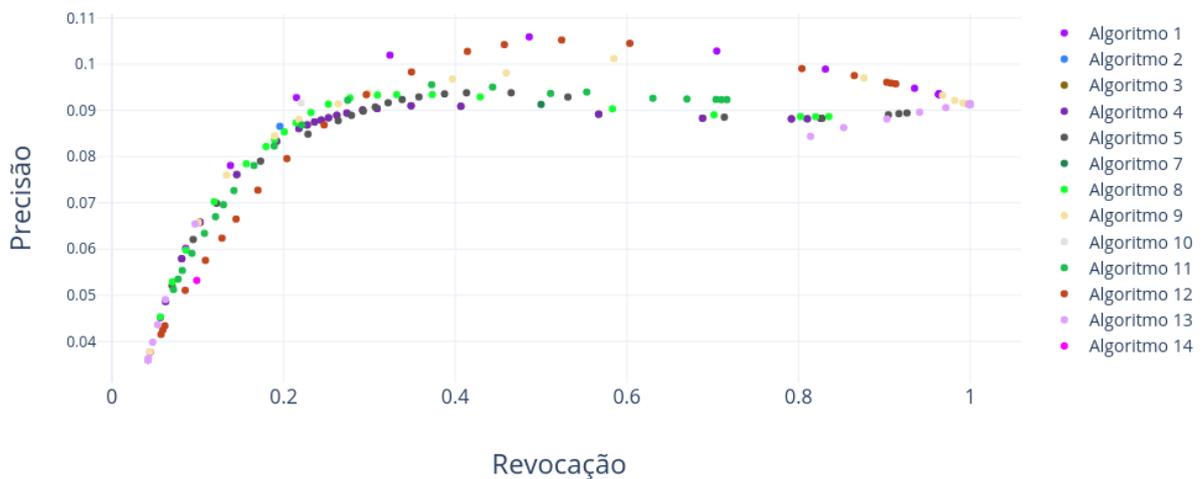


Figura 66 – Gráfico de precisão por revocação para regra 2 e conjunto de dados de entrada 2

saída 2, o *benchmark* majoritário (Algoritmo 6) apresenta um bom desempenho. Neste caso, porém, esse *benchmark* não supera o Algoritmo 4 (Rede Neural Convolutiva) e ambos superam todos os outros algoritmos, uma vez que a caixa do Algoritmo 4 não sobrepõe a de nenhum outro algoritmo. Assim, o Algoritmo 4 é considerado o melhor algoritmo de classificação para a regra 3 utilizando o conjunto de dados 1.

O diagrama de caixas da Figura 68 mostram melhores valores de acurácia para os Algoritmos 1 (Floresta Aleatória), 11 (empilhamento utilizando o Perceptron Multicamadas como classificador nível 1) e 13 (XGBoost).

Como os diagramas de caixa não são suficientes para determinar o melhor algoritmo, um teste ANOVA foi aplicado, identificando diferença de média de acurácia com p-valor

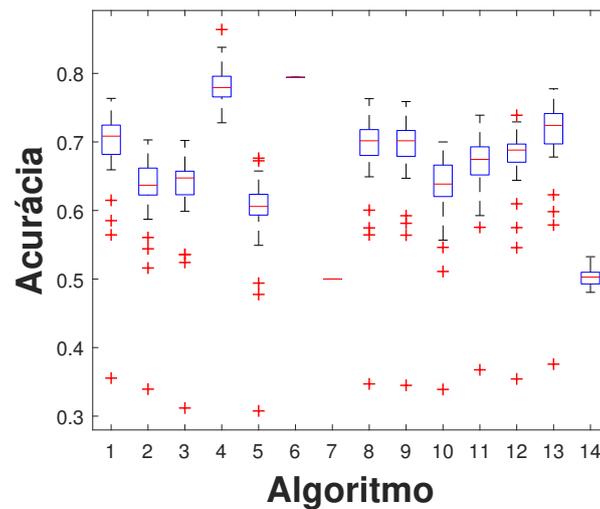


Figura 67 – Acurácia dos algoritmos para regra 3 e conjunto de dados de entrada 1

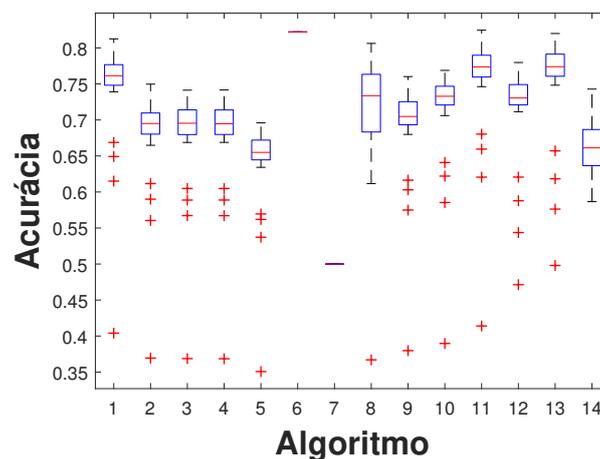


Figura 68 – Acurácia dos algoritmos para regra 3 e conjunto de dados de entrada 2

muito baixo. Posteriormente, o teste de Tukey, apresentado na Figura 69, foi aplicado e também não identifica diferença entre esses três algoritmos.

A Figura 70 mostra o gráfico de precisão por revocação dos algoritmos de classificação, indicando que o Algoritmo 11 apresenta o pior desempenho. Assim, pode-se considerar os Algoritmos 1 e 13 como os algoritmos de maior desempenho para esse cenário (regra de saída 3 e conjuntos de dados de entrada 2), sendo que o Algoritmo 1 é selecionado como o melhor por ser mais rápido que o Algoritmo 13.

#### 5.2.1.6 Regra 4

Primeiramente, para o conjunto de dados de entrada 1, a Figura 71 compara a acurácia de cada algoritmo, mostrando que o Algoritmo 14 (Programação Genética) possui a pior média entre eles. Os demais algoritmos apresentam médias muito próximas, de forma que não é identificada nenhuma diferença entre elas apenas visualizando o diagrama

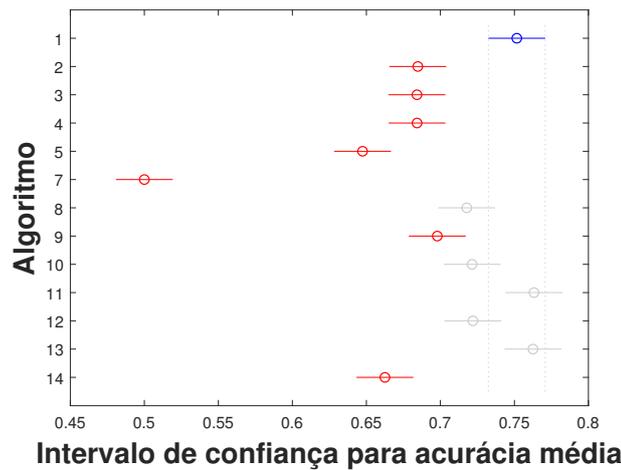


Figura 69 – Teste de acurácia para regra 3 e conjunto de dados de entrada 2



Figura 70 – Gráfico de precisão por revocação para regra 3 e conjunto de dados de entrada 2

de caixas. Assim, foi realizado um teste ANOVA que também falhou em identificar essa diferença, com um p-valor de 0,07284.

Observa-se no gráfico de precisão por revocação da Figura 72 uma precisão muito espalhada para valores de revocação baixos e valores de precisão dos algoritmos muito próximos para valores altos de revocação. Para valores menores de revocação, pode-se visualizar maiores valores de precisão do Algoritmo 1 (Floresta Aleatória).

O diagrama de caixas da Figura 73 revela pouca diferença entre as medianas de acurácia apresentada por cada algoritmo, além de um excesso de *outliers* em quase todos esses algoritmos. Dessa forma, um teste ANOVA foi realizado na tentativa de se descobrir se existe alguma diferença estatística para as médias de acurácia, apresentando um p-valor

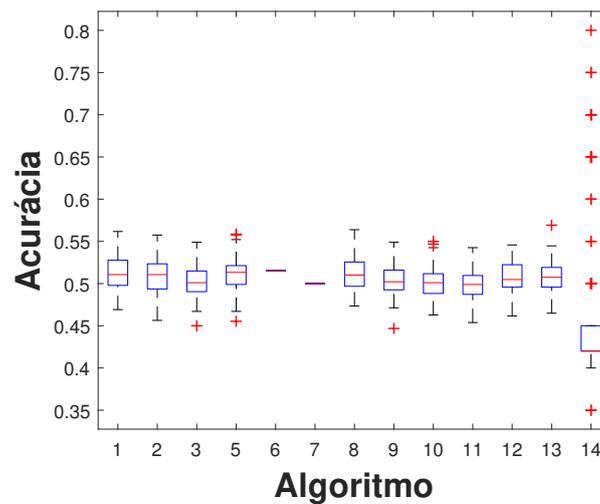


Figura 71 – Acurácia dos algoritmos para regra 4 e conjunto de dados de entrada 1

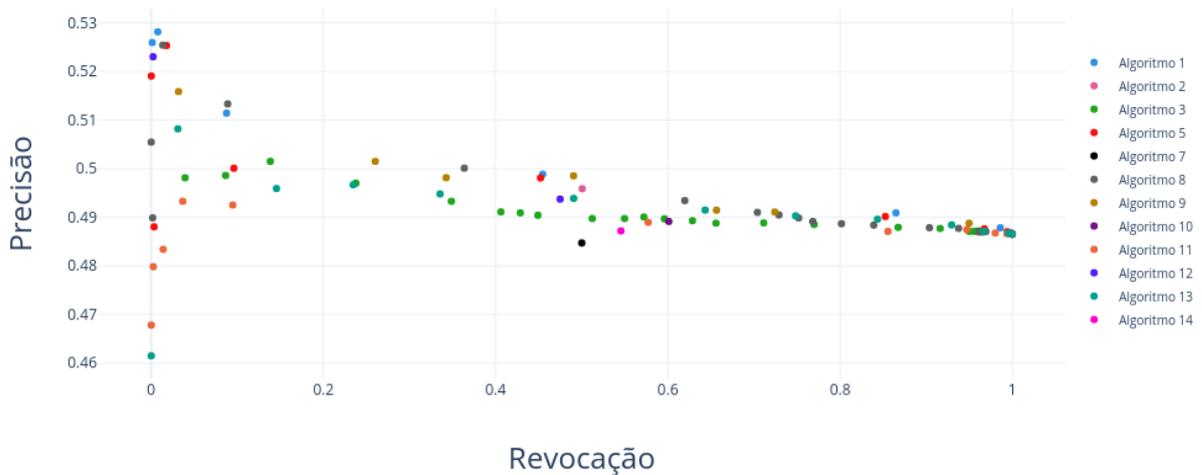


Figura 72 – Gráfico de precisão por revocação para regra 4 e conjunto de dados de entrada 1

muito baixo. Uma vez que o teste ANOVA detectou diferença para a média de acurácia entre os algoritmos, o teste de comparação múltipla (Tukey) apresentado na Figura 74 mostra que os melhores valores foram apresentados pelos Algoritmos 1, 5, 8 e 9.

Dentre os Algoritmos 1 (Floresta Aleatória), 5 (Regressão Logística), 8 (Votação Majoritária) e 9 (Votação Ponderada), o Algoritmo 8 é o que apresenta a melhor curva no gráfico de precisão por revocação da Figura 75. Assim, ele é escolhido como o melhor algoritmo de classificação deste cenário.

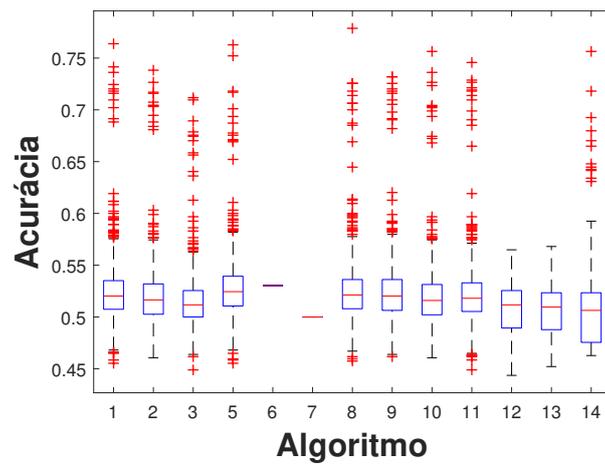


Figura 73 – Acurácia dos algoritmos para regra 4 e conjunto de dados de entrada 2

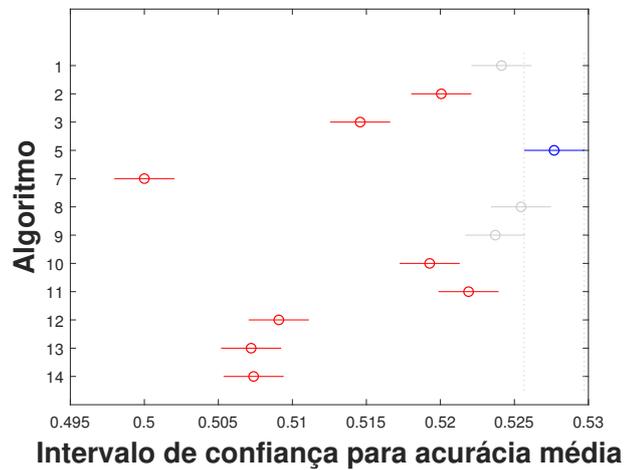


Figura 74 – Teste de acurácia para regra 4 e conjunto de dados de entrada 2

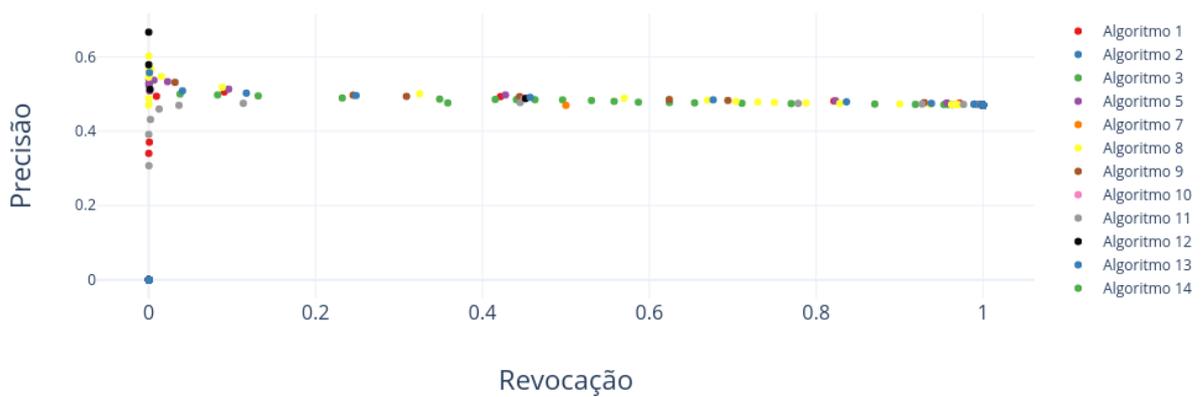


Figura 75 – Gráfico de Precisão por revocação para regra 4 e conjunto de dados de entrada 2

### 5.2.1.7 Regra 5

De forma similar à regra 4, a regra de saída 5 também classifica as amostras com base nos valores de retorno de todas as amostras. Mas, enquanto a regra 4 é bem balanceada, com metade das amostras para cada classe, a regra 5 é desbalanceada, sendo que um quarto das amostras são rotuladas como classe 0 e três quartos, como classe 1.

Considerando o conjunto de dados de entrada 1, a Figura 76 compara valores de acurácia de cada algoritmo e mostra que, para este caso, o *benchmark* aleatório é muito pior que os outros algoritmos, como esperado já que o desempenho ruim para o *benchmark* aleatório e um bom desempenho para o *benchmark* majoritário, em relação à acurácia, são característicos de problemas desbalanceados. Além dos *benchmarks*, os Algoritmos 1, 5 e 8 são os que apresentam os melhores desempenhos.

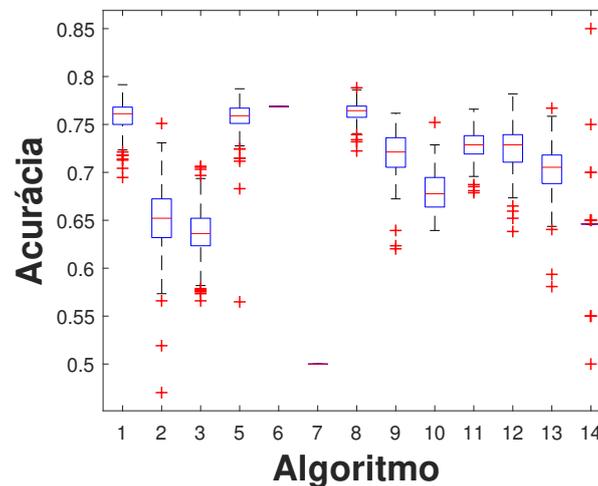


Figura 76 – Acurácia dos algoritmos para regra 5 e conjunto de dados de entrada 1

Após a obtenção de um p-valor muito baixo no teste ANOVA de acurácia para os algoritmos de classificação, o teste de Tukey da Figura 77 falha em detectar diferença entre os Algoritmos 1, 5 e 8.

O gráfico de precisão por revocação da Figura 78 apresenta poucos pontos do Algoritmo 1. Porém, esses pontos apresentam uma precisão superior a dos demais algoritmos e, por isso, é considerado o melhor algoritmo para este cenário.

Para o conjunto de dados 2, o diagrama de caixas da Figura 79 revela maiores valores de mediana e menores variações de acurácia dos Algoritmos 1, 5 e 14, considerando muitas das suas acurácias como *outliers*. Para comparar a média de acurácia desses algoritmos, um teste ANOVA foi realizado e apresentou um p-valor muito baixo. Assim, o teste de comparação múltipla de Tukey (apresentado na Figura 80) foi executado e, apesar de mostrar que os Algoritmos 1, 5 e 14 apresentam médias estatisticamente superiores às média dos demais algoritmos, falhou em detectar diferença entre os três algoritmos.

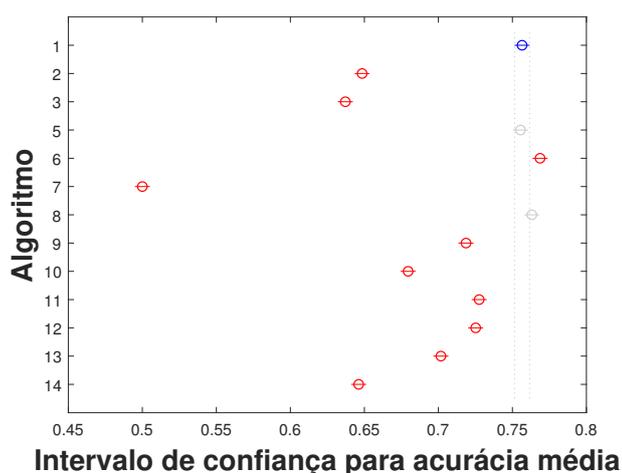


Figura 77 – Teste de acurácia para regra 5 e conjunto de dados de entrada 1

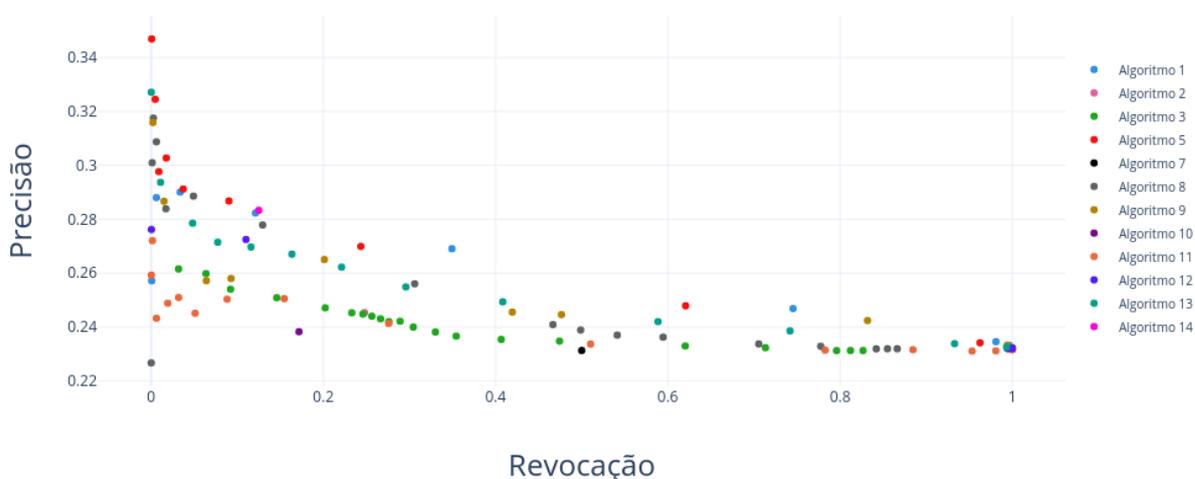


Figura 78 – Gráfico de precisão por revocação para regra 5 e conjunto de dados de entrada 1

Por fim, o gráfico de precisão por revocação da Figura 81 demonstra melhores valores do Algoritmo 1, que apresenta os melhores valores de precisão dado um valor de revocação e, portanto, é definido como o melhor algoritmo para o cenário da regra 5 e conjunto de dados de entrada 2.

### 5.2.2 Análise de ganhos financeiros

A análise de desempenho dos algoritmos possibilitaram a escolha de um melhor algoritmo de classificação para cada um dos cenários estudados. Assim, a análise de ganhos financeiros discute os resultados financeiros proporcionados pelo melhor algoritmo de cada um dos cenários. Em resumo, os melhores algoritmos para cada cenário foram:

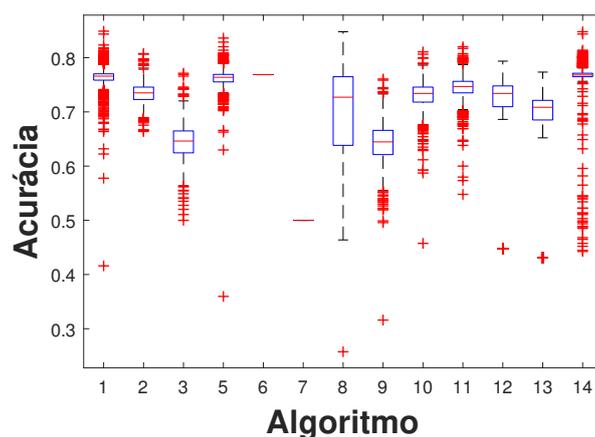


Figura 79 – Acurácia dos algoritmos para regra 5 e conjunto de dados de entrada 2

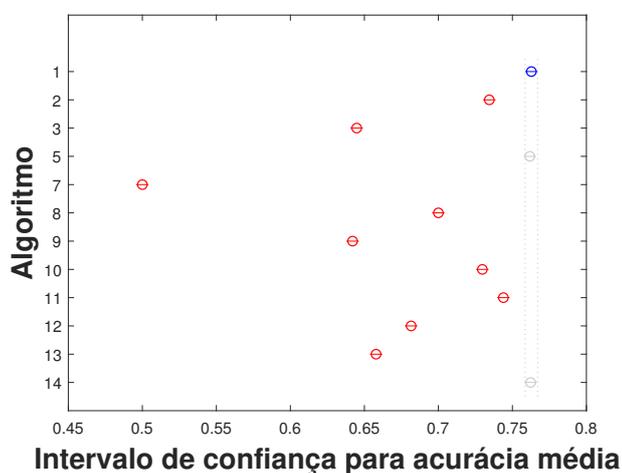


Figura 80 – Teste de acurácia para regra 5 e conjunto de dados de entrada 2

- Regra de saída 1 e conjunto de dados 1: Algoritmo 4;
- Regra de saída 2 e conjunto de dados 1: Algoritmo 4;
- Regra de saída 3 e conjunto de dados 1: Algoritmo 4;
- Regra de saída 4 e conjunto de dados 1: Algoritmo 1;
- Regra de saída 5 e conjunto de dados 1: Algoritmo 1;
- Regra de saída 1 e conjunto de dados 2: Algoritmo 12;
- Regra de saída 2 e conjunto de dados 2: Algoritmo 1;
- Regra de saída 3 e conjunto de dados 2: Algoritmo 1;
- Regra de saída 4 e conjunto de dados 2: Algoritmo 8;
- Regra de saída 5 e conjunto de dados 2: Algoritmo 1;

Utilizando o conjunto de dados 1, a comparação dos ganhos proporcionados por cada regra de saída são comparados. Para isso, a simulação das negociações dos ativos de acordo com os sinais produzidos por cada algoritmo foi realizado considerando o período entre 2009 e 2017, sendo que a cada dia, ativos com predição igual a 0 são vendidos caso

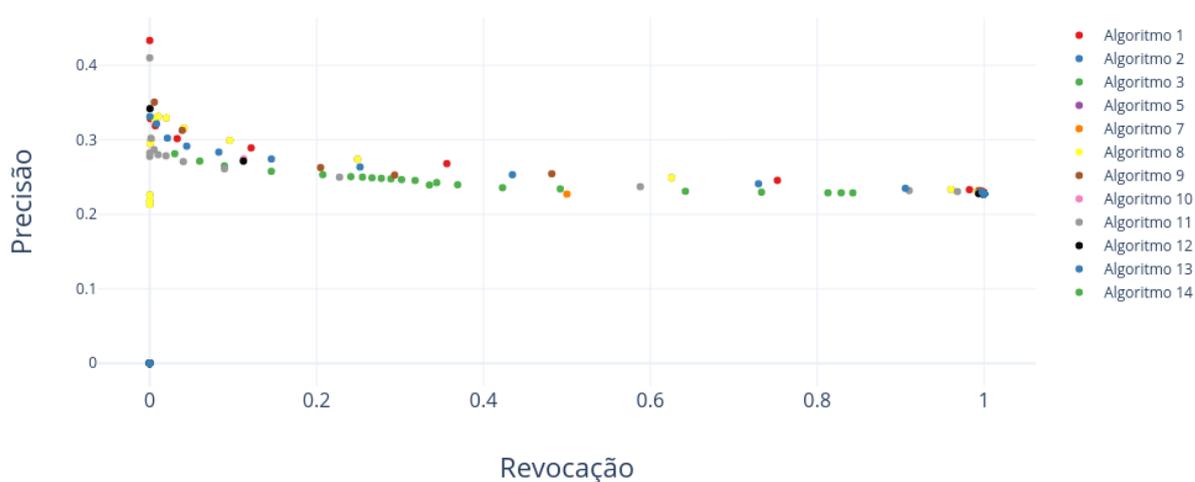


Figura 81 – Gráfico de precisão por revocação para regra 5 e conjunto de dados de entrada 2

esteja presente no portfólio e com predição igual a 1 são comprados se não estão presentes no portfólio. Começando com um valor de 100000,00 reais, o capital acumulado gerado pelo melhor algoritmo de cada regra é registrado.

O capital acumulado, em reais, proporcionado pelo melhor algoritmo de classificação em cada regra de saída utilizando o conjunto de dados 1, com granularidade de 1 dia, pode ser observado na Figura 82, que demonstra uma similaridade muito grande do comportamento do capital acumulado em relação ao tempo para cada uma dessas regras durante o período entre os anos de 2009 e 2017. Nota-se que a regra 2 apresenta um pior desempenho que as demais ao final do tempo observado, mas nenhum algoritmo apresentou um desempenho substancialmente melhor que os demais. Assim, para o conjunto de dados 1, a regra 1, utilizando o Algoritmo 4 (CNN), será utilizada nos próximos experimentos.

Para analisar estatisticamente as diferenças nos ganhos proporcionados por cada regra de saída, os indicadores de retorno, CVaR e drawdown mensais são comparados utilizando análises de variância. Os diagramas de caixa de retorno mensal para cada regra são apresentados na Figura 83, que revela pouca diferença entre as regras. Um teste ANOVA aplicado aos retornos mensais das diferentes regras também falha em detectar diferença entre as regras, apresentando um p-valor de 0,9763.

A Figura 84 mostra os diagramas de caixa dos valores de risco CVaR mensais de cada regra de saída, revelando pouca diferença na mediana desses valores. O teste ANOVA foi aplicado e o p-valor de 0,1099 indica que não foi encontrada uma diferença estatística entre as diferentes regras.

O drawdown máximo mensal também apresenta pouca diferença nas distribuições

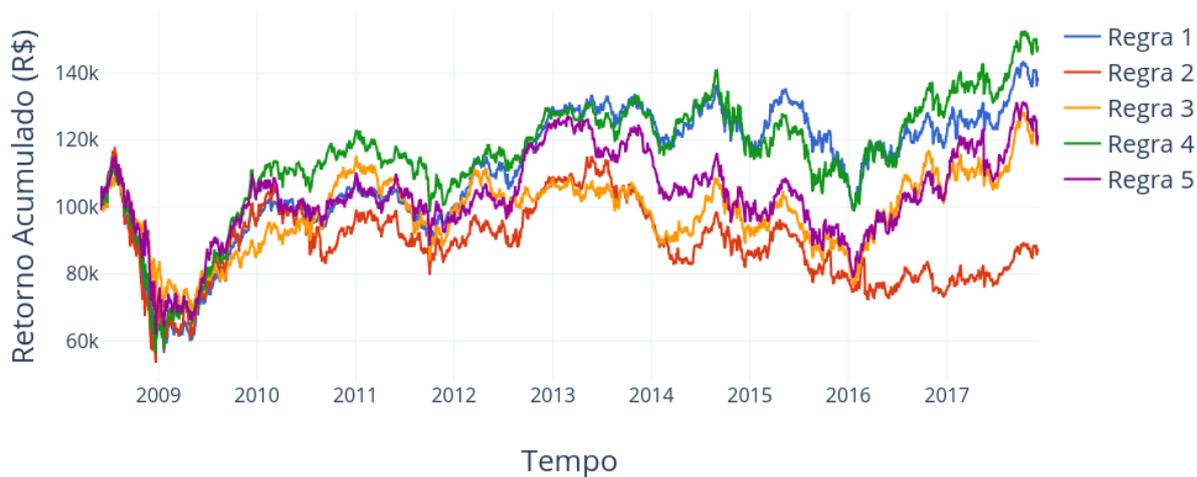


Figura 82 – Capital acumulado para conjunto de dados 1

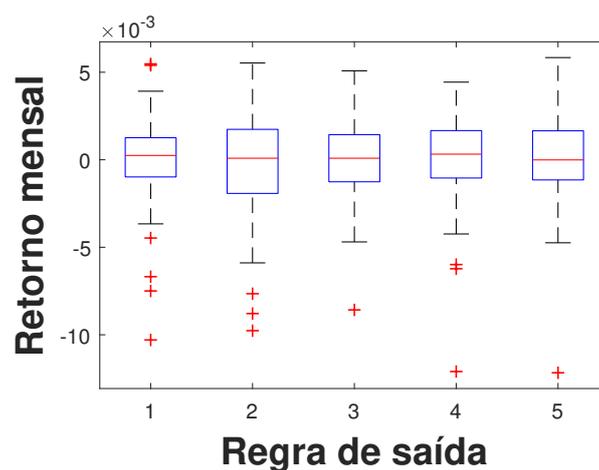


Figura 83 – Retorno mensal das regras de saída para conjunto de dados 1

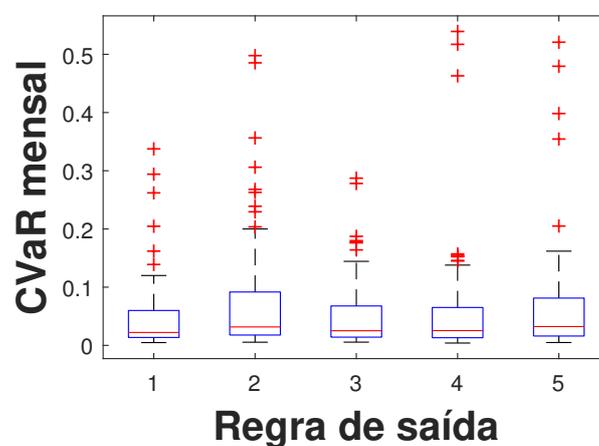


Figura 84 – CVaR mensal das regras de saída para conjunto de dados 1

dos seus valores, conforme revelado pelos diagramas de caixa da Figura 85. O teste ANOVA também falha em detectar diferença estatística entre as regras, com um p-valor de 0,0601.

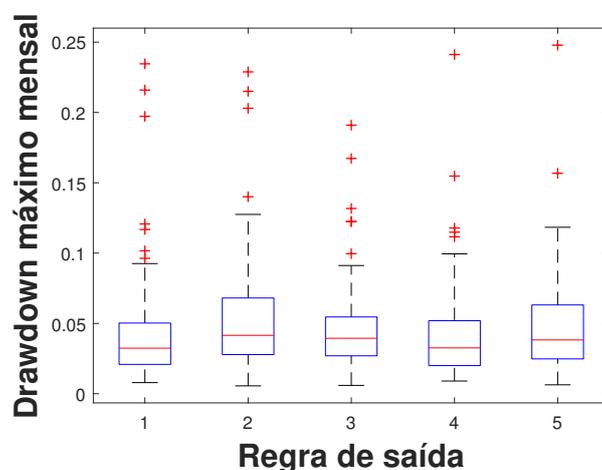


Figura 85 – Drawdown máximo mensal das regras de saída para conjunto de dados 1

Sem que uma regra de saída apresente superioridade estatística, a regra 1 é utilizada nos próximos experimentos que utilizam o conjunto de dados 1, por ser uma regra considerada mais simples e mais utilizada em trabalhos relacionados.

A Figura 86 apresenta o capital acumulado, em reais, proporcionado pelo melhor algoritmo em cada regra de saída utilizando o conjunto de dados 2, com granularidade de 1 hora. A figura mostra que o capital chegou a 0,00 real em todas as regras de saída, indicando que todo o dinheiro investido foi perdido. Essa redução linear do capital ao final das curvas de capital acumulado se deve ao custo fixo aplicado às transações financeiras.

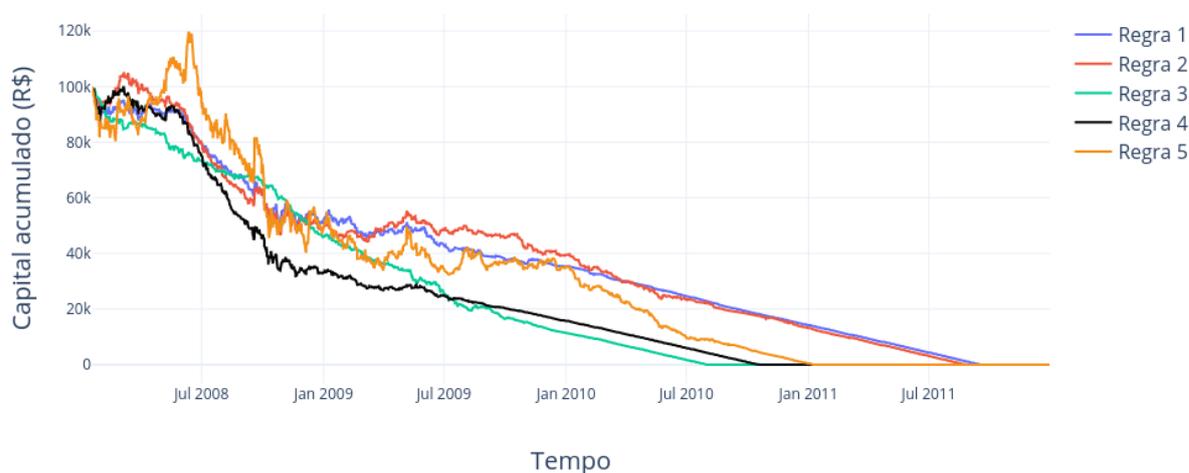


Figura 86 – Capital acumulado para conjunto de dados 2

Uma vez que em todas as regras de saída para o conjunto de dados 2 caíram ao

valor de 0,00 real após determinado tempo, apenas o conjunto de dados 1 será utilizado nos próximos experimentos.

## 5.3 Combinações de otimização de portfólio e classificação de séries financeiras

De acordo com os resultados das seções anteriores, a classificação de séries financeiras utiliza o Algoritmo 4 (Rede Neural Convolucional) com a regra de saída 1 e conjunto de dados 1, e a otimização utiliza o algoritmo NSGA-II.

### 5.3.1 Classificação seguida pela otimização

O método da classificação seguida da otimização realiza a otimização de portfólios após a classificação dos ativos, sendo que todos os ativos classificados como promissores (classe 1) são utilizados como entrada para a otimização, que seleciona um conjunto de ativos dentre esses ativos promissores. Esse método sinaliza a compra dos ativos presentes nesse portfólio e a venda dos ativos que não estão presentes no portfólio.

### 5.3.2 Otimização com restrição de classificação

A otimização com restrição realiza a otimização de portfólio considerando todos os ativos disponíveis. O modelo de otimização possui uma restrição a mais, que garante que o melhor ativo (aquele com maior probabilidade de pertencer à classe 1), esteja no portfólio com uma proporção mínima de investimento. Essa proporção foi selecionada após realizações de testes prévios e seu valor, definido como 0,5 ou 50%.

### 5.3.3 Otimização seguida pela classificação

Na otimização seguida da classificação, os portfólios Pareto-ótimos de ativos são gerados pela otimização. Em seguida, o algoritmo de agrupamento K-médias (ou K-means) é utilizado para diminuir a quantidade desses portfólios. Arbitrariamente, foi determinado um valor de 30 grupos, de forma que o K-médias seleciona 30 diferentes grupos e os núcleos de cada um desses grupos são selecionados para a etapa de classificação que, finalmente, seleciona o melhor portfólio, aquele com a maior probabilidade de pertencer à classe 1, dos portfólios promissores.

### 5.3.4 Análise de ganhos financeiros das combinações

Para comparar a eficiência das combinações propostas, são utilizados os indicadores de retorno, risco CVaR e drawdown máximo mensais avaliados para uma simulação de

negociações diárias realizada entre os anos de 2009 e 2017. Essas combinações também são comparadas com os *baselines* otimização e classificação, aplicados separadamente, e o *benchmark* índice Bovespa (Ibovespa B3), que é o principal indicador de desempenho das ações negociadas na B3. Para efeito de simplificação, as três combinações, os dois *baselines* e o *benchmark* são denominados:

- Método 1: Ibovespa B3;
- Método 2: Classificação;
- Método 3: Otimização;
- Método 4: Classificação seguida de otimização;
- Método 5: Otimização com restrição de classificação;
- Método 6: Otimização seguida de classificação;

Os gráficos da Figura 87 comparam o retorno acumulado, em porcentagem, das combinações propostas em comparação com os *baselines* e o *benchmark*. Esses gráficos revelam que, ao final do período observado, a otimização seguida de classificação (Método 6) e a otimização com restrição (Método 5) foram os algoritmos que apresentaram melhor desempenho, superando os algoritmos *baselines*, enquanto a classificação seguida de otimização (Método 4) apresentou o pior desempenho. Porém, nenhum algoritmo superou o índice Bovespa nesse período.

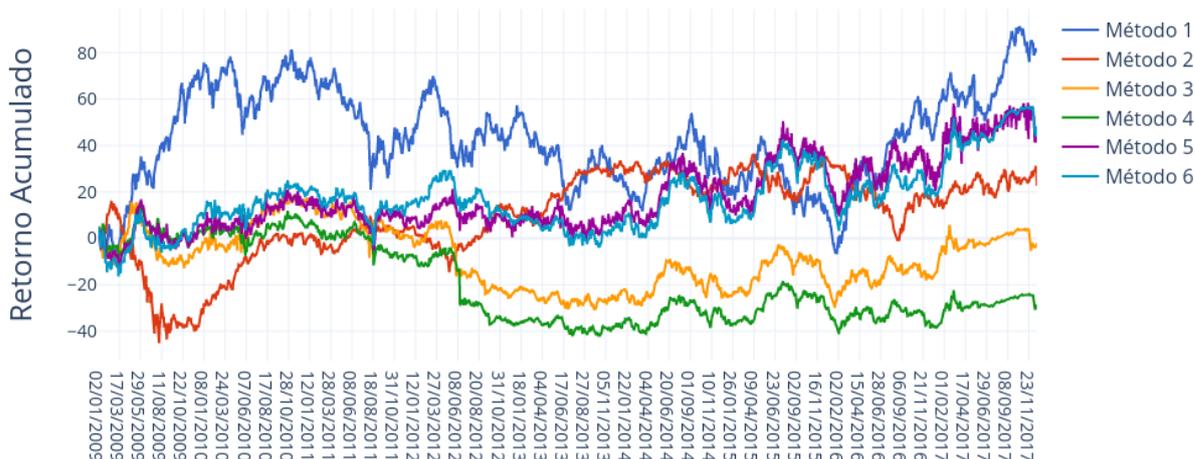


Figura 87 – Retorno acumulado dos métodos

A Figura 88 apresenta os diagramas de caixa para o retorno mensal de cada método, em que pode-se observar que as medianas dos métodos estão próximas de 0. O teste ANOVA falha em detectar diferença estatística para a média de retorno dos métodos, com um p-valor de  $8,06 \times 10^{-60}$ .

Para a métrica drawdown máximo mensal, o diagrama de caixa da Figura 89a

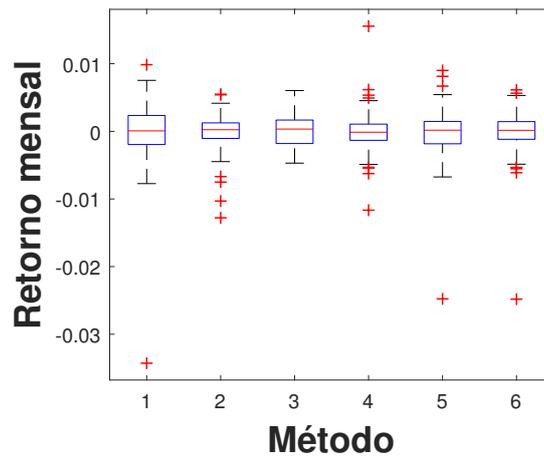


Figura 88 – Comparação de retorno mensal

revela uma mediana maior para os Métodos 3 (otimização) e 4 (classificação seguida de otimização), além de uma variabilidade maior desses métodos, uma vez que suas caixas são maiores que as demais. O teste ANOVA foi aplicado e, por apresentar um p-valor de 0,9997, detectou uma diferença estatística entre os métodos, conforme mostrada pelo teste de comparação múltipla da Figura 89b, que confirma um maior drawdown máximo do Método 4, seguido pelo Método 3.

Como o drawdown máximo é uma medida de risco, percebe-se que o Método 2, classificação de ativos financeiros, consegue reduzir os riscos, uma vez que ele seleciona um portfólio que possui, em média, metade da quantidade de ativos da B3, enquanto que o Método 3, de otimização, seleciona um portfólio com uma cardinalidade menor. Os Métodos 5 e 6, apesar de possuírem também uma cardinalidade próxima ao do Método 3, conseguiram se beneficiar da integração das abordagens para reduzir o drawdown máximo.

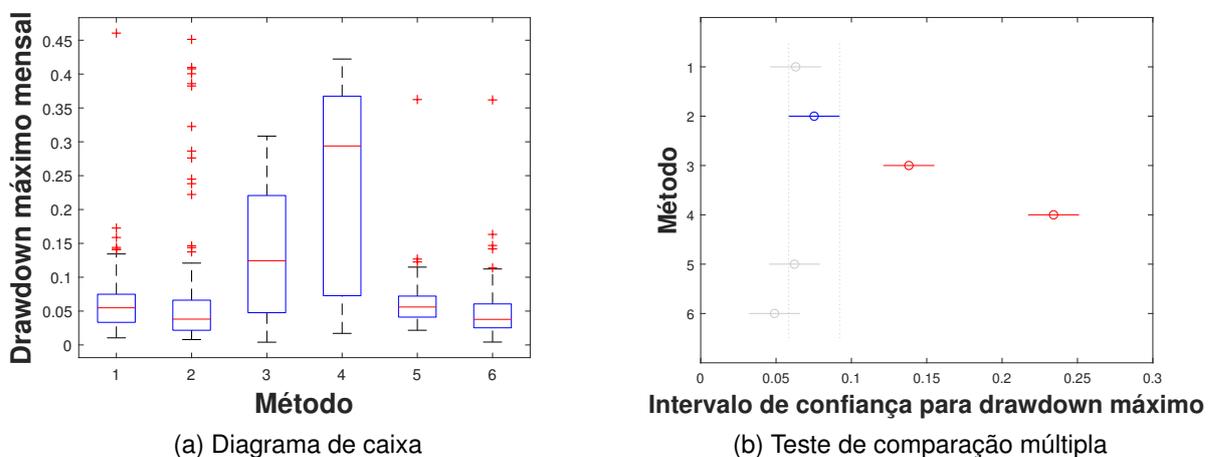


Figura 89 – Comparação de drawdown máximo mensal

O diagrama de caixa ilustrado na Figura 90a mostra uma maior variabilidade da

medida de risco CVaR para o Método 5, com uma distribuição maior dos dados para a métrica. O teste ANOVA aplicado aos métodos analisados indicou um p-valor de 0,0009 e o teste de comparação múltipla mostrado na Figura 90b confirma que o CVaR mensal médio apresentado pelo Método 5 supera, estatisticamente, os valores médios dos Métodos 2, 3 e 6. Sendo que a diferença entre os Métodos 3 e o 5 é que o Método 3 seleciona sempre o portfólio de maior retorno sobre risco dentre os portfólios na fronteira de Pareto, enquanto que o Método 5 utiliza a classificação para selecionar o melhor portfólio nessa fronteira, nota-se que a classificação aumenta o risco do investimento, uma vez que seleciona frequentemente portfólios de maior retorno em detrimento do risco.

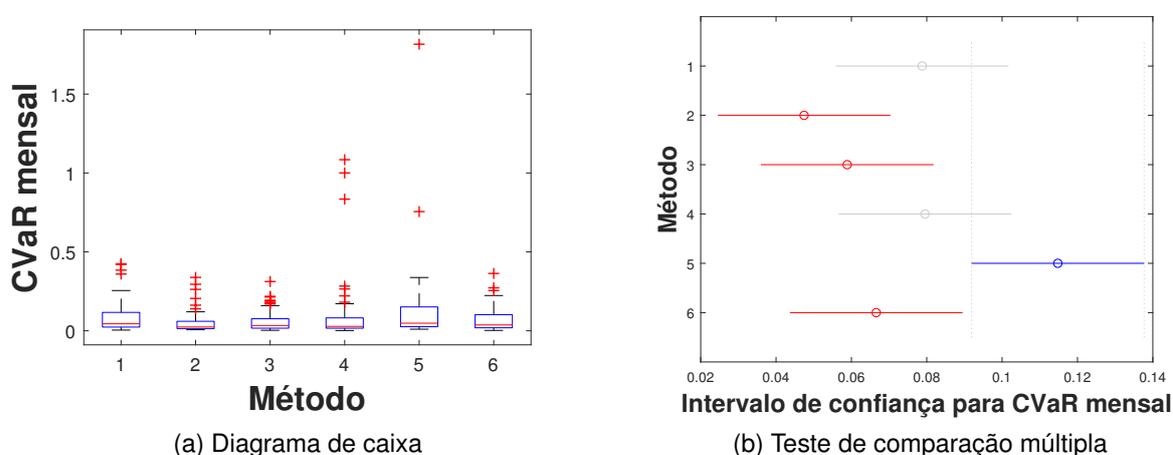


Figura 90 – Comparação de CVaR mensal

Entre as integrações de otimização e classificação (Métodos 4, 5 e 6), apenas o Método 6 não é superado por outros métodos em nenhuma métrica de desempenho e, por isso, esse é o método de integração utilizado nos próximos experimentos.

### 5.3.5 Análise de cardinalidade dos métodos

A análise de cardinalidade dos portfólios gerados pelos Métodos 2, 3 e 6 compara a quantidade de vezes em que um determinado ativo é escolhido por cada método. Além disso, é apresentado na Figura 91, o retorno de cada um desses ativos durante todo o período. Assim, é esperado que ativos com maiores retornos sejam escolhidos mais vezes. A Figura 91 apresenta um mapa de calor de cada método, sendo que a escala de cor (apresentada na figura) é aplicada independentemente para cada um dos métodos e para o retorno, facilitando a identificação dos ativos menos escolhidos e dos mais escolhidos por cada método.

Percebe-se que o Método 2 (classificação de séries financeiras) apresenta pouca variação na quantidade de vezes em que um determinado ativo é escolhido. Considerando que o período analisado é composto de 2460 dias, cada ativo é escolhido um pouco mais

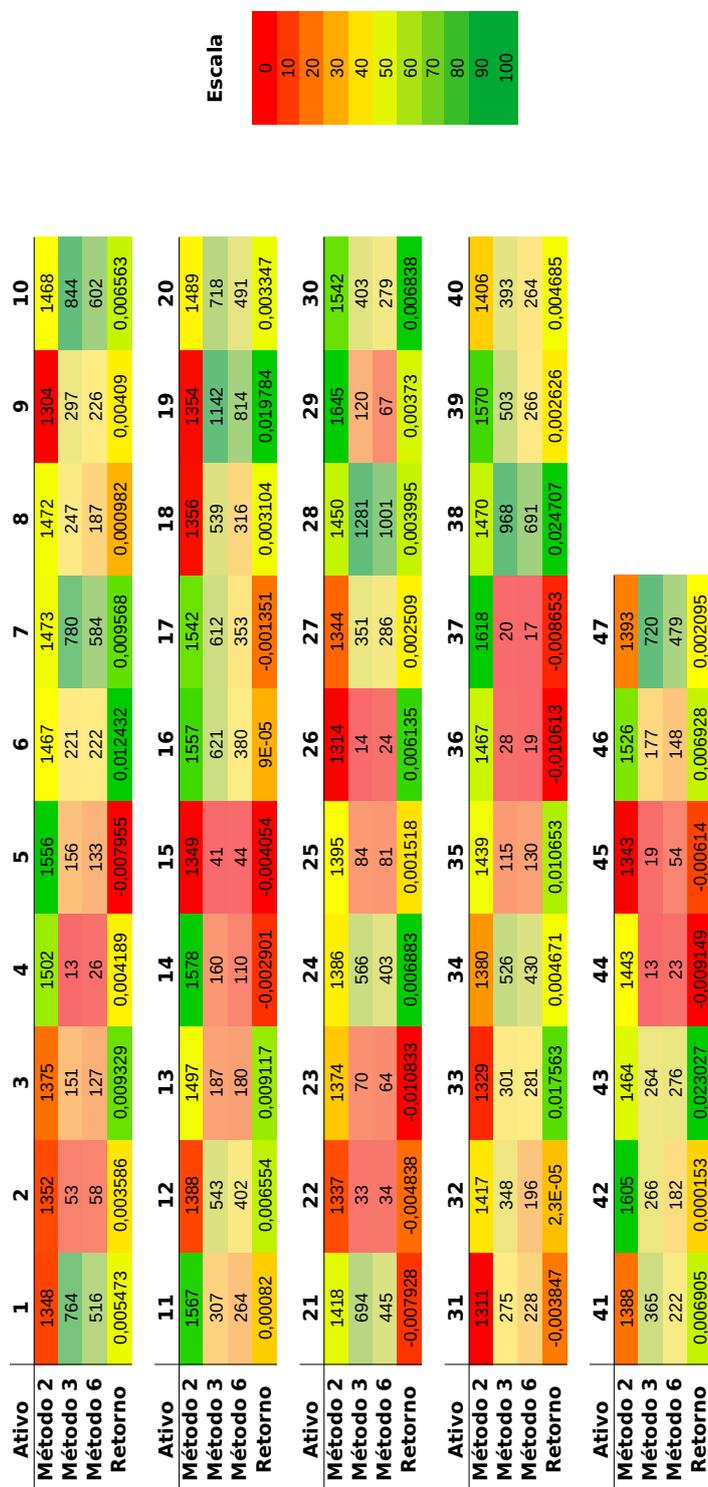


Figura 91 – Mapa de calor da seleção de ativo por método

que a metade (1230 dias) dessas vezes. Enquanto isso, os Métodos 3 e 6 (otimização de portfólios e a otimização de portfólios com restrição de classificação, respectivamente) selecionam uma quantidade muito menor de cardinalidade, na média. De fato, a cardinalidade média do Método 2 é de 30,5261261261261 ativos, enquanto que a cardinalidade média do Método 3 é 7,79864864864865 ativos e a cardinalidade média do Método 6 é de 5,68693693693694 ativos. Além disso, os métodos 3 e 6 apresentam uma correlação maior que a do Método 2 entre a quantidade de vezes que um ativo é escolhido e o retorno desse ativo.

### 5.3.6 Análise do efeito do custo de transação no ganho financeiro

Os ganhos proporcionados pelos métodos são dependentes dos custos de transação aplicados às negociações dos ativos financeiros. A fim de analisar o efeito desses custos no ganho, o Método 6, integração que apresentou os melhores ganhos e riscos, é comparado com métodos que desconsideram custos de transação, simulando ganhos em um caso hipotético em que esses gastos de negociação de ativos não existem. Dessa forma, os seguintes métodos são criados:

- Método 7: Método 2, desconsiderando custos de transação;
- Método 8: Método 3, desconsiderando custos de transação;
- Método 9: Método 6, desconsiderando custos de transação.

A Figura 92 apresenta os diagramas de bloco dos retornos mensais para os métodos sem custo de transação com o Método 6, que considera esses custos, e o Ibovespa B3, utilizado como *benchmark*. Esses diagramas mostram que as medianas dos métodos, para retorno, são próximas de 0, sendo possível notar uma menor variabilidade dos retornos do Método 7. Porém, o teste ANOVA para a comparação desses retornos mensais apresenta um p-valor de 0,9836, indicando que o teste não detectou diferença estatística entre esses valores.

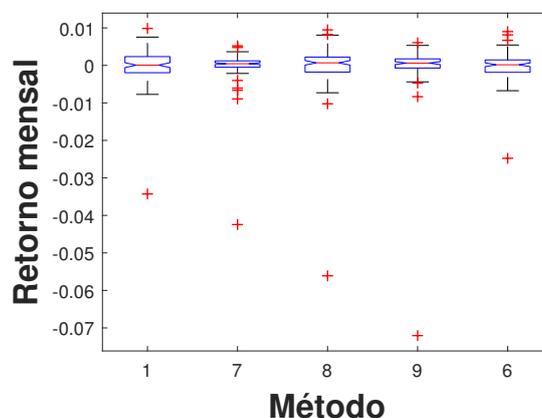


Figura 92 – Retorno mensal dos métodos

Os valores de CVaR mensais proporcionados pelos métodos, apresentados pelos diagramas de bloco na Figura 93, também revelam uma proximidade entre os métodos, com uma pequena diferença para os Métodos 6 e 7, sendo que o 6 (método com custo de transação) apresenta uma variabilidade maior e o 7, uma variabilidade menor. Novamente, o teste ANOVA falha em detectar diferença estatística entre os métodos, apresentando um p-valor de 0,0966.

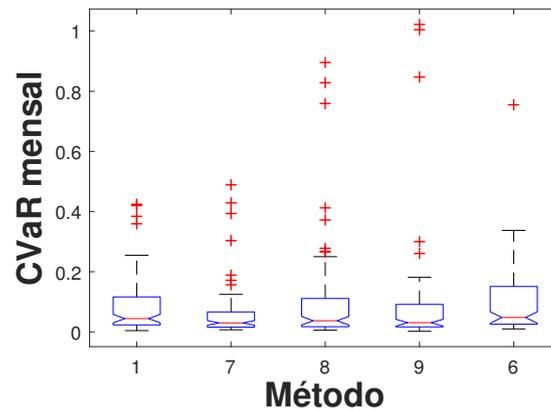


Figura 93 – CVaR mensal dos métodos

Finalmente, a comparação de drawdown máximo mensal dos métodos no diagrama de bloco da Figura 94, também indica pouca diferença entre eles. O teste ANOVA apresenta um p-valor de 0,3324, também indicando que nenhuma diferença estatística foi evidenciada.

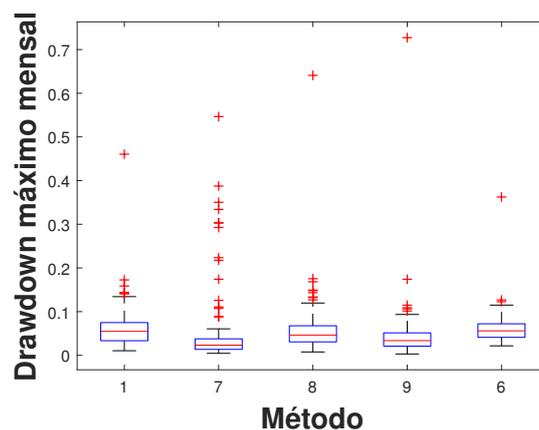


Figura 94 – Drawdown máximo mensal dos métodos

Apesar de nenhuma diferença estatística ter sido detectada para as métricas de desempenho consideradas, os gráficos de retorno acumulado, em porcentagem, da Figura 95 revelam que o Método 6, que considera os custos de transação, apresenta o menor retorno ao final do período analisado, sendo o único método que apresenta desempenho inferior ao *benchmark* Ibovepa B3 (Método 1). Percebe-se também que o Método 9, que consiste na integração da classificação com otimização apresenta o maior retorno ao final

do período, indicando que a abordagem proposta tem o potencial de superar o ganho financeiro do mercado, já que o Ibovespa B3 representa esse desempenho do mercado de ações, de forma geral.



Figura 95 – Retorno acumulado dos métodos

## 6 Conclusão

Neste trabalho é proposto um arcabouço computacional que integra métodos de classificação de séries temporais financeiras e de otimização de portfólios financeiros. Em relação à otimização de portfólios, é proposto um novo modelo e um novo algoritmo evolucionário para solucioná-lo. As comparações propostas envolvem, principalmente, quatro modelos de otimização de portfólio que diferem entre si pelas variáveis de decisão utilizadas e pela linearidade ou não das funções do modelo. Para os métodos de otimização, são comparados os métodos exatos (*branch-and-cut*) e as meta-heurísticas (NSGA-II). Resultados dos testes *in-sample* e *out-of-sample* revelam vantagens e desvantagens de cada modelo e método, apesar do fato de que um modelo e método é escolhido como sendo o melhor, neste trabalho, para serem utilizados na combinação da classificação com a otimização de portfólio.

O modelo de otimização com variáveis de decisão inteiras, restrição de cardinalidade variável e rebalanceamento apresenta resultados que permitem concluir que ele representa uma melhora em relação ao modelo com restrição de cardinalidade fixa proposto por [Ferreira \(2018\)](#). Essa conclusão decorre do fato de que o modelo com cardinalidade variável proporciona portfólios diversificados, com diferentes quantidades de ativos, em um tempo de execução muito menor que o modelo com restrição de cardinalidade fixa gastaria, permitindo a aplicação da metodologia proposta em problemas envolvendo centenas ou milhares de ativos. Além disso, os resultados indicam que, mesmo permitindo quantidades muito elevadas de ativos, os portfólios selecionados pelos métodos de otimização conseguem limitar esse valor, evitando custos elevados de manutenção desses portfólios. A proposta do algoritmo evolutivo para solucionar esse modelo proposto também pode ser considerada uma contribuição acadêmica do trabalho.

Além das contribuições acadêmicas, os experimentos relacionados à otimização de portfólio também podem ser considerados contribuições práticas, pois podem auxiliar investidores na realização de negociações no Mercado de Ações. A comparação de modelos com variáveis de decisão contínuas e inteiras, por exemplo, revela que mesmo quando os modelos com variáveis de decisão contínuas proporcionam um conjunto de portfólios melhor que o conjunto proporcionado por modelos de variáveis inteiras, quando suas variáveis são transformadas em quantidades de lotes, esses portfólios gerados por modelos contínuos perdem desempenho e se tornam piores que os portfólios proporcionados pelos modelos inteiros. Por isso, na prática, a utilização de modelos inteiros é recomendada.

Além disso, as análises *in-sample* revelam que os métodos exatos, mesmo quando aplicados a modelos aproximados, proporcionam portfólios melhores que os métodos

heurísticos. A diferença de desempenho entre eles, no entanto, é muito pequena, enquanto o tempo de execução para os métodos heurísticos são muito menores. É importante notar que, neste trabalho, apenas 53 ativos são considerados, no total, e para problemas práticos em que se considera um número muito maior de ativos, espera-se que a diferença seja ainda maior. Assim, geralmente, o uso de algoritmos heurísticos é mais interessante, na prática, exceto em casos em que o tempo de execução não seja muito relevante e que mesmo uma pequena diferença de desempenho dos portfólios seja muito desejada.

Já as análises *out-of-sample* mostram que a escolha entre a utilização de granularidades de frequência menores (diárias) ou maiores (horárias) dependem muito da preferência do investidor, sendo que séries com granularidades diárias apresentam menores riscos e, portanto, os portfólios selecionados neste caso possuem menor volatilidade, enquanto séries com granularidades horárias proporcionam portfólios que podem apresentar elevados ganhos financeiros em um curto período, apesar de apresentar também um maior risco.

Analisando os testes *in-sample*, nota-se que os portfólios com menores cardinalidades tendem a apresentar maiores riscos e retornos, enquanto os portfólios com maiores cardinalidades tendem a apresentar menores riscos e retornos, corroborando a teoria de [Markowitz \(1952\)](#), que defende a ideia de que a diversificação de portfólios financeiros reduz seus riscos. As análises *out-of-sample* também justificam a utilização do rebalanceamento, que garante uma baixa sensibilidade do comportamento de ganhos para diferentes valores de custos de transação, além de minimizar estes.

Resultados obtidos para a etapa de classificação de séries temporais dos ativos financeiros indicam que os desempenhos dos classificadores são muito relacionados ao tamanho dos conjuntos de dados de treino e teste, enquanto o método de seleção de atributos não parece afetar muito esse desempenho, considerando acurácia, precisão ou revocação, estando mais fortemente relacionado com o tempo de execução desses algoritmos.

Para a comparação dos algoritmos de classificação, a acurácia é considerada a métrica mais importante, uma vez que é desejado um bom acerto na predição dos ativos promissores, para sua compra, e dos ativos não promissores, para sua venda. Comparando a acurácia dos algoritmos de classificação para as diferentes regras de saída, não se percebe facilmente um padrão de desempenho dos algoritmos para todas as regras. Considerando, porém, o gráfico de precisão por revocação dos algoritmos, nota-se que o Algoritmo 4 (CNN), supera os demais para granularidades diárias, apresentando elevados valores de precisão para valores baixos de revocação. Além disso, os modelos propostos superam o *benchmark* aleatório (Algoritmo 7), em precisão, em todas as regras de saída analisadas, indicando que os classificadores apresentam desempenho superior ao de um palpite aleatório.

Três combinações são propostas neste trabalho: classificação seguida de otimização,

otimização seguida de classificação e otimização utilizando classificação como restrição. Resultados sugerem que classificação seguida de otimização apresenta um Drawdown superior aos demais e a otimização com restrição de classificação apresenta um maior risco CVaR, enquanto a otimização com restrição de classificação e a otimização seguida de classificação apresentaram maiores retornos acumulados no período analisado, superando a otimização e a classificação executadas separadamente, justificando a adoção dessas formas propostas de combinação das abordagens. Enquanto que na literatura são encontrados apenas integrações que utilizam classificação antes da otimização, os resultados indicam que a integração utilizando a otimização antes da classificação proporcionam melhor retorno financeiro.

Por fim, é possível afirmar que os resultados obtidos questionam os métodos encontrados na literatura e propõe um método que apresenta um desempenho superior. Além disso, o arcabouço computacional proposto constitui uma contribuição tecnológica capaz de realizar diversos testes com algoritmos da área de Finanças Computacionais e guiar investidores em suas decisões de operações financeiras no Mercado Financeiro.

## 6.1 Trabalhos futuros

Trabalhos futuros devem analisar o efeito de diferentes granularidades de dados, incluindo outras periodicidades, além de hora ou dia, comparar o desempenho de outros algoritmos de classificação, como o RNN (Recurrent Neural Network), e a implementação de outras formas de combinação das abordagens. Trabalhos futuros também devem incluir outros mercados, além do Mercado de Ações e mercados de outros países, além do Brasil. Outros aspectos a considerar em trabalhos futuros é a inclusão de indicadores fundamentalistas como atributos para os algoritmos de classificação e a utilização de novas estratégias de compra e venda, como a adição de gatilhos para a venda dos ativos com base no ganho ou perda proporcionados por este. Por fim, uma implementação a se considerar em trabalhos futuros é a integração de análise de notícias e comentários relacionados aos ativos financeiros em redes sociais para predição de uma futura apreciação ou depreciação dos ativos considerados.

# Referências

AGRAWAL, R. B.; DEB, K.; AGRAWAL, R. B. Simulated binary crossover for continuous search space. **Complex Systems**, v. 9, 06 1995. Citado 2 vezes nas páginas 8 e 51.

ANDREOLA, R.; HAERTEL, V. Classificação de imagens hiperespectrais empregando support vector machines. **Boletim de Ciências Geodésicas**, v. 16, n. 2, 2010. ISSN 1982-2170. Disponível em: <<https://revistas.ufpr.br/bcg/article/view/17682>>. Citado na página 36.

ARNALDO, I.; KRAWIEC, K.; O'REILLY, U.-M. Multiple regression genetic programming. In: **Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: Association for Computing Machinery, 2014. (GECCO '14), p. 879–886. ISBN 9781450326629. Disponível em: <<https://doi.org/10.1145/2576768.2598291>>. Citado na página 87.

ASSIS, C. A. S. de. **Predição de Tendências em séries financeiras utilizando meta-classificadores**. 2019. 103 f. Tese (Doutorado em Modelagem Matemática e Computacional) — Centro Federal de Educação Tecnológica de Minas Gerais, 2019. Citado 4 vezes nas páginas 14, 15, 83 e 88.

BANCO do Brasil. 2019. Disponível em: <<https://www.bb.com.br/portalbb/page83,129,9128,0,1,1,9.bb>>. Acesso em: 25 de agosto de 2019. Citado na página 95.

BARROSO, B. C. **Um Estudo Comparativo da Integração entre Métodos da Análise Técnica e Otimização de Portfólios**. 2017. 126 f. Dissertação (Mestrado em Modelagem Matemática e Computacional) — Centro Federal de Educação Tecnológica de Minas Gerais, 2017. Citado 2 vezes nas páginas 10 e 95.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006. Citado 14 vezes nas páginas 2, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 e 44.

BOX, G.; JENKINS, G. **Time series analysis: forecasting and control**. Holden-Day, 1970. (Holden-Day series in time series analysis). Disponível em: <<https://books.google.com.br/books?id=5BVfnXaq03oC>>. Citado na página 12.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>. Citado na página 41.

BREIMAN, L. et al. **Classification and Regression Trees**. Monterey, CA: Wadsworth and Brooks, 1984. Citado 3 vezes nas páginas 41, 42 e 43.

CASTRO, L. D. **Fundamentals of Natural Computing**: Basic concepts, algorithms, and applications. [S.l.: s.n.], 2006. ISBN 9781584886433. Citado 2 vezes nas páginas 50 e 53.

CESARONE, F.; MORETTI, J.; TARDELLA, F. Optimally chosen small portfolios are better than large ones. **Economics Bulletin**, v. 36, n. 4, p. 1876–1891, 2016. Disponível em: <<https://ideas.repec.org/a/ebl/ecbull/eb-16-00671.html>>. Citado na página 103.

- CHEN, C.; WEI, Y. Robust multiobjective portfolio optimization: a set order relations approach. **Journal of Combinatorial Optimization**, v. 38, n. 1, p. 21–49, Jul 2019. ISSN 1573-2886. Disponível em: <<https://doi.org/10.1007/s10878-018-0364-9>>. Citado na página 11.
- CHEN, S.; HE, H. Stock prediction using convolutional neural network. **IOP Conference Series: Materials Science and Engineering**, v. 435, p. 012026, 11 2018. Citado 2 vezes nas páginas 85 e 86.
- CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <<http://doi.acm.org/10.1145/2939672.2939785>>. Citado na página 86.
- CHENG, R.; GAO, J. On cardinality constrained mean-cvar portfolio optimization. In: **The 27th Chinese Control and Decision Conference (2015 CCDC)**. [S.l.: s.n.], 2015. p. 1074–1079. ISSN 1948-9439. Citado na página 8.
- COVER, T. M.; HART, P. et al. Nearest neighbor pattern classification. **IEEE transactions on information theory**, Menlo Park, v. 13, n. 1, p. 21–27, 1967. Citado na página 13.
- DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. [S.l.]: Jhon Wiley & Sons, 2001. v. 16. Citado 2 vezes nas páginas 8 e 52.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, April 2000. ISSN 1089-778X. Citado 5 vezes nas páginas 53, 54, 55, 56 e 59.
- DEB, K. et al. Bi-objective portfolio optimization using a customized hybrid nsga-ii procedure. In: TAKAHASHI, R. H. C. et al. (Ed.). **Evolutionary Multi-Criterion Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 358–373. ISBN 978-3-642-19893-9. Citado 13 vezes nas páginas 8, 10, 55, 56, 57, 58, 70, 71, 72, 73, 74, 95 e 97.
- DEFAZIO, A.; BACH, F.; LACOSTE-JULIEN, S. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. **arXiv e-prints**, p. arXiv:1407.0202, Jul 2014. Citado na página 84.
- DRUCKER, H. et al. Support vector regression machines. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1997. p. 155–161. Citado na página 12.
- DUARTE, J. ao B. **Introdução à análise de componentes principais**. 1998. Disponível em: <<https://agro.ufg.br/up/396/o/ACP.pdf>>. Acesso em: 30 de julho de 2019. Citado na página 48.
- DUARTE, W. M. **Modelo automatizado para decisões de investimento usando machine learning e a técnica de portfolio selection de black-litterman**. 2018. 78 f. Dissertação (Mestrado em Administração) — Centro Federal de Educação Tecnológica de Minas Gerais, 2018. Citado 3 vezes nas páginas 14, 15 e 83.
- ENGLE, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. **Econometrica**, [Wiley, Econometric Society], v. 50, n. 4, p. 987–1007, 1982. ISSN 00129682, 14680262. Disponível em: <<http://www.jstor.org/stable/1912773>>. Citado na página 12.

- EVERITT, B. et al. **Cluster Analysis**. Wiley, 2011. (Wiley Series in Probability and Statistics). ISBN 9780470978443. Disponível em: <<https://books.google.com.br/books?id=w3bE1kqd-48C>>. Citado 2 vezes nas páginas 48 e 49.
- FERREIRA, F. G. D. C. **Estudo comparativo de modelos e técnicas para otimização de portfólios com restrição de cardinalidade**. 2018. 118 f. Dissertação (Mestrado em Modelagem Matemática e Computacional) — Centro Federal de Educação Tecnológica de Minas Gerais, 2018. Citado 9 vezes nas páginas 10, 26, 54, 64, 72, 95, 96, 97 e 142.
- FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: **Proceedings of the Thirteenth International Conference on International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996. (ICML'96), p. 148–156. ISBN 1558604197. Citado 2 vezes nas páginas 13 e 45.
- FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001. ISSN 00905364. Disponível em: <<http://www.jstor.org/stable/2699986>>. Citado na página 11.
- FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. **Machine Learning**, v. 29, n. 2, p. 131–163, Nov 1997. Citado na página 14.
- FU, X. et al. A Machine Learning Framework for Stock Selection. **arXiv e-prints**, p. arXiv:1806.01743, Jun 2018. Citado 3 vezes nas páginas 13, 15 e 83.
- GERLEIN, E. A. et al. Evaluating machine learning classification for financial trading: An empirical approach. **Expert Systems with Applications**, v. 54, p. 193 – 207, 2016. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417416000282>>. Citado na página 83.
- GHAZANFAR, M. a. et al. Using machine learning classifiers to predict stock exchange index. **International Journal of Machine Learning and Computing**, v. 7, p. 24–29, 04 2017. Citado 2 vezes nas páginas 13 e 15.
- GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 1157–1182, mar. 2003. ISSN 1532-4435. Disponível em: <<http://dl.acm.org/citation.cfm?id=944919.944968>>. Citado na página 47.
- HALL, M. A. Correlation-based feature selection for machine learning. **Department of Computer Science**, v. 19, 06 2000. Citado na página 47.
- HANAOKA, G. P. **Seleção de Carteiras de Investimentos Através da Otimização de Modelos Restritos Multiobjetivos Utilizando Algoritmos Evolutivos**. 2014. 87 f. Dissertação (Mestrado em Modelagem Matemática e Computacional) — Centro Federal de Educação Tecnológica de Minas Gerais, 2014. Citado 4 vezes nas páginas 9, 10, 15 e 95.
- HANAOKA, G. P. **Sistema de tomada de decisão de compra e venda de ações baseado em máquinas de vetores suporte e otimização de portfólios**. 2019. 108 f. Tese (Doutorado em Modelagem Matemática e Computacional) — Centro Federal de Educação Tecnológica de Minas Gerais, 2019. Citado na página 15.
- HASTIE, T. J. et al. Multi-class adaboost. **Statistics and Its Interface**, v. 2, p. 349–360, 2009. Citado na página 86.

- HSU, M.-W. et al. Bridging the divide in financial market forecasting. **Expert Syst. Appl.**, v. 61, p. 215–234, 2016. Citado 4 vezes nas páginas 11, 15, 79 e 83.
- ITANO, F.; SOUSA, M. A. de Abreu de; DEL-MORAL-HERNANDEZ, E. Extending mlp ann hyper-parameters optimization by using genetic algorithm. In: **2018 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–8. ISSN 2161-4407. Citado 2 vezes nas páginas 85 e 111.
- JIANG, S. et al. Consistencies and contradictions of performance metrics in multiobjective optimization. **IEEE Transactions on Cybernetics**, v. 44, n. 12, p. 2391–2404, Dec 2014. ISSN 2168-2267. Citado na página 59.
- KARA, Y.; BOYACIOGLU, M. A.; BAYKAN Ömer K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. **Expert Systems with Applications**, v. 38, n. 5, p. 5311 – 5319, 2011. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417410011711>>. Citado 2 vezes nas páginas 79 e 83.
- KAUCIC, M.; MORADI, M.; MIRZAZADEH, M. Portfolio optimization by improved nsga-ii and spea 2 based on different risk measures. **Financial Innovation**, v. 5, n. 1, p. 1, Jun 2019. ISSN 2199-4730. Disponível em: <<https://doi.org/10.1186/s40854-019-0140-6>>. Citado na página 11.
- KAUFMAN, L.; ROUSSEEUW, P. J. **Finding Groups in Data: An Introduction to Cluster Analysis**. [S.l.]: John Wiley, 1990. ISBN 978-0-47031680-1. Citado 3 vezes nas páginas 49, 50 e 73.
- KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: **Proceedings of the IEEE International Conference on Neural Networks**. [S.l.: s.n.], 1995. p. 1942–1948. Citado na página 11.
- KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. **arXiv e-prints**, p. arXiv:1412.6980, Dec 2014. Citado na página 85.
- KONONENKO, I. Bayesian neural networks. **Biological Cybernetics**, v. 61, n. 5, p. 361–370, Sep 1989. Citado na página 13.
- KOZA, J. et al. **Genetic Programming IV: Routine Human-Competitive Machine Intelligence**. Kluwer Academic Publishers, 2005. (Genetic Programming). ISBN 9780387250670. Disponível em: <<https://books.google.com.br/books?id=dZwEm52UonEC>>. Citado na página 60.
- KRAUSS, C.; DO, X. A.; HUCK, N. **Deep neural networks, gradient-boosted trees, random forests**: Statistical arbitrage on the s&p 500. [S.l.], 2016. Disponível em: <<https://ideas.repec.org/p/zbw/iwqwdp/032016.html>>. Citado 4 vezes nas páginas 11, 15, 76 e 83.
- KUMAR, D.; THAKUR, M. Weighted multicategory nonparallel planes svm classifiers. **Neurocomputing**, v. 211, p. 106–116, 2016. Citado 2 vezes nas páginas 12 e 80.
- LAVRADO, R. **Medindo o risco através da perda máxima: Maximum Drawdown at Risk**. 2015. 72 f. Tese (Doutorado em Matemática) — Instituto Nacional de Matemática Pura e Aplicada, 2015. Citado na página 20.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. **Neural Computation**, v. 1, n. 4, p. 541–551, 1989. Disponível em: <<https://doi.org/10.1162/neco.1989.1.4.541>>. Citado na página 38.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers Inc., v. 86, n. 11, p. 2278–2323, 1998. ISSN 0018-9219. Citado 2 vezes nas páginas 38 e 39.

LIPPMANN, R. An introduction to computing with neural nets. **IEEE ASSP Magazine**, v. 4, p. 4–22, 1987. Citado na página 85.

LYUU, Y. **Financial Engineering and Computation: Principles, mathematics, and algorithms**. New York, NY, USA: Cambridge University Press, 2002. ISBN 0-521-78171-X. Citado na página 2.

MADAVAN, N. K. Multiobjective optimization using a pareto differential evolution approach. In: **Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on**. [S.l.: s.n.], 2002. v. 2, p. 1145–1150. Citado na página 9.

MANSINI, R.; OGRYCZAK, W.; SPERANZA, M. **Linear and Mixed Integer Programming for Portfolio Optimization**. [S.l.: s.n.], 2015. 1-119 p. Citado 8 vezes nas páginas 17, 18, 19, 21, 22, 23, 24 e 25.

MARCELINO, S. S. de F. **Formação de portfólio por meio de máquinas de suporte vetorial e redes de camadas profundas**. 2016. 129 f. Dissertação (Mestrado em Administração) — Universidade de Brasília, 2016. Citado 3 vezes nas páginas 12, 15 e 83.

MARKOWITZ, H. Portfolio selection\*. **The Journal of Finance**, v. 7, n. 1, p. 77–91, 1952. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1952.tb01525.x>>. Citado 12 vezes nas páginas 1, 8, 10, 11, 17, 20, 22, 55, 72, 75, 103 e 143.

MATHWORKS. **Convolutional Neural Network**. 2017. Disponível em: <<https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>>. Acesso em: 07 de agosto de 2019. Citado na página 40.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Citado na página 32.

MITCHELL, J. E. Branch-and-cut algorithms for combinatorial optimization problems. **Handbook of applied optimization**, v. 1, p. 65–77, 1999. Citado na página 28.

MITCHELL, J. E.; BRAUN, S. Rebalancing an investment portfolio in the presence of convex transaction costs, including market impact costs. **Optimization Methods and Software**, Taylor & Francis, v. 28, n. 3, p. 523–542, 2013. Disponível em: <<https://doi.org/10.1080/10556788.2012.717940>>. Citado na página 10.

MURPHY, J. **Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications**. New York Institute of Finance, 1999. (New York Institute of Finance Series). ISBN 9780735200661. Disponível em: <[https://books.google.com.br/books?id=5zhXEqr\\_lC](https://books.google.com.br/books?id=5zhXEqr_lC)>. Citado 2 vezes nas páginas 1 e 2.

ORR, M. J. et al. **Introduction to radial basis function networks**. [S.l.]: Technical Report, Center for Cognitive Science, University of Edinburgh, 1996. Citado na página 13.

- PAIVA, F. D. **Redes Neurais para decisões no Mercado de Ações brasileiro**. 2014. 119 f. Tese (Doutorado em Administração) — Universidade Federal de Lavras, 2014. Citado na página 77.
- PAIVA, F. D. et al. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. **Expert Systems with Applications**, v. 115, 08 2018. Citado 3 vezes nas páginas 76, 78 e 83.
- PANG, X. et al. An innovative neural network approach for stock market prediction. **The Journal of Supercomputing**, Jan 2018. ISSN 1573-0484. Disponível em: <<https://doi.org/10.1007/s11227-017-2228-y>>. Citado na página 83.
- PLATT, J. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998. Citado na página 14.
- POMMERANZENBAUM, I. R. **Redes Neurais Artificiais na predição das principais séries do índice Ibovespa e suas aplicações em sistemas automatizados de negociação**. 2014. 126 f. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, 2014. Citado na página 83.
- PROBST, P.; WRIGHT, M. N.; BOULESTEIX, A.-L. Hyperparameters and tuning strategies for random forest. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 9, n. 3, p. e1301, 2019. Citado 2 vezes nas páginas 84 e 86.
- QUINLAN, J. R. **C4.5: Programs for Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1558602402. Citado na página 14.
- RADCLIFFE, N. J. Forma analysis and random respectful recombination. In: **ICGA**. [S.l.: s.n.], 1991. Citado na página 9.
- RIBEIRO, L. de C.; BARBOSA, A. M.; ARANTES, J. ao Matheus de O. Algoritmo genético multiobjetivo: sistema adaptativo com elitismo. **Brazilian Conference on Dynamics, Control and their Applications**, v. 9, p. 6, 2010. Citado na página 95.
- RIJN, J. N. van; HUTTER, F. Hyperparameter importance across datasets. In: **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining**. ACM, 2018. Disponível em: <<https://doi.org/10.1145/2F3219819.3220058>>. Citado na página 86.
- ROBIČ, T.; FILIPIČ, B. Demo: Differential evolution for multiobjective optimization. In: COELLO, C. A. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). **Evolutionary Multi-Criterion Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 520–533. ISBN 978-3-540-31880-4. Citado na página 9.
- ROCKAFELLAR, R. T.; URYASEV, S. Optimization of conditional value-at-risk. **Journal of Risk**, v. 2, p. 21–41, 2000. Citado 4 vezes nas páginas 2, 21, 22 e 23.
- ROJAS-DOMÍNGUEZ, A. et al. Optimal hyper-parameter tuning of svm classifiers with application to medical diagnosis. **IEEE Access**, v. 6, p. 7164–7176, 2018. ISSN 2169-3536. Citado na página 84.
- SCHRIJVER, A. **Theory of Linear and Integer Programming**. New York, NY, USA: John Wiley & Sons, Inc., 1986. ISBN 0-471-90854-1. Citado 2 vezes nas páginas 27 e 28.

SCIKIT-LEARN. 2019. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 17 de setembro de 2019. Citado na página 111.

SILVA, Y.; HERTHEL, A. B.; SUBRAMANIAN, A. A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems. **Expert Systems with Applications**, v. 133, 05 2019. Citado na página 11.

SOLEIMANI, H.; GOLMAKANI, H. R.; SALIMI, M. H. Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. **Expert Syst. Appl.**, Pergamon Press, Inc., Tarrytown, NY, USA, v. 36, n. 3, p. 5058–5063, abr. 2009. ISSN 0957-4174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2008.06.007>>. Citado na página 9.

STORN, R.; PRICE, K. **Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces**. 1995. Citado na página 9.

TALBI, E.-G. **Metaheuristics: From design to implementation**. [S.l.: s.n.], 2009. v. 74. Citado 2 vezes nas páginas 51 e 53.

THAKUR, M.; KUMAR, D. A hybrid financial trading support system using multi-category classifiers and random forest. **Applied Soft Computing**, v. 67, p. 337 – 349, 2018. ISSN 1568-4946. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1568494618301224>>. Citado 3 vezes nas páginas 12, 15 e 83.

WADE, C.; GLYNN, K. **Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python**. Packt Publishing, 2020. ISBN 9781839213809. Disponível em: <<https://books.google.com.br/books?id=2tcDEAAQBAJ>>. Citado na página 45.

WANG, H. S.; TU, C. H.; CHEN, K. H. Supplier selection and production planning by using guided genetic algorithm and dynamic nondominated sorting genetic algorithm ii approaches. p. 15, 2015. Citado na página 55.

WOLPERT, D. H. Stacked generalization. **Neural Networks**, v. 5, n. 2, p. 241 – 259, 1992. ISSN 0893-6080. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0893608005800231>>. Citado na página 43.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms — a comparative case study. In: \_\_\_\_\_. **Parallel Problem Solving from Nature — PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 292–301. ISBN 978-3-540-49672-4. Disponível em: <<http://dx.doi.org/10.1007/BFb0056872>>. Citado na página 58.