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

ESTUDO COMPARATIVO DE MODELOS E TÉCNICAS PARA OTIMIZAÇÃO DE PORTFÓLIOS COM RESTRIÇÃO DE CARDINALIDADE

FERNANDO GARCIA DINIZ CAMPOS FERREIRA

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

BELO HORIZONTE
FEVEREIRO DE 2018

FERNANDO GARCIA DINIZ CAMPOS FERREIRA

ESTUDO COMPARATIVO DE MODELOS E TÉCNICAS PARA OTIMIZAÇÃO DE PORTFÓLIOS COM RESTRIÇÃO DE CARDINALIDADE

Dissertação 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 Mestre 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
FEVEREIRO DE 2018

Esta folha deverá ser substituída pela cópia digitalizada da folha de aprovação fornecida.

Agradecimentos

Gostaria de agradecer aos docentes do curso de pós-graduação em Modelagem Matemática e Computacional do CEFET-MG. Agradecimento especial aos professores Rodrigo Tomás Nogueira Cardoso, que me orientou neste trabalho, Adriano César Machado Pereira, Felipe Dias Paiva e Flávio Vinícius Cruzeiro Martins, pelas sugestões oferecidas e correções feitas no projeto de qualificação e na dissertação. Agradeço, por fim, a agência de fomento CAPES pela bolsa concedida durante o período da pesquisa.

Resumo

A teoria moderna do portfólio introduz o conceito de risco para investimentos financeiros, iniciando um problema de decisão em uma relação de compromisso entre o risco e retorno, que são características conflitantes em um investimento. Recentemente, estudos vêm utilizando otimização multiobjetivo para tratar esse problema, mas modelos e técnicas ainda estão sendo aprimorados. Deseja-se, portanto, contribuir para o desenvolvimento desse processo de otimização, especificamente com restrição de quantidade de diferentes ativos em um investimento, propondo modificações em modelos e técnicas já propostos. Para isso, quatro experimentos foram planejados, estabelecendo comparação de técnicas e modelos para otimização, simulações para validação desses e desenvolvimento de técnicas paralelas objetivando a redução do tempo computacional desses métodos de otimização, respectivamente. Resultados dos dois primeiros experimentos sugerem superioridade do modelo e técnica proposta em relação a outros presentes na literatura. O terceiro experimento valida o modelo e técnica escolhidos nos experimentos anteriores em uma simulação de negociações no mercado de ações, e o quarto experimento apresenta melhorias à técnica que apresentou melhor desempenho, melhorando a qualidade das soluções geradas com um aumento mínimo de tempo.

Palavras-chave: Otimização de portfólios. Algoritmos evolutivos multiobjetivos. Otimização combinatória. Programação paralela.

Abstract

The modern portfolio theory introduces the concept of risk in financial investments, initiating a decision problem in a trade-off between risk and return, which are conflicting characteristics in an investment. Recently, studies have been using multiobjective optimization to address this problem, but models and techniques are still being improved. It is therefore desired to contribute to the development of this optimization process, specially with the constraint of number of different assets in an investment, proposing modifications in models and techniques already existing. In this way, four experiments were planned, establishing comparison of optimization techniques and models, simulations for their validation and development of parallel techniques in order to reduce the proposed methods runtime, respectively. Results of the first and second experiments suggest superiority of the proposed model and technique in relation to others existing in the literature. Third experiment validates the model and technique selected in the previous experiments in a simulation of stock market trading and the fourth experiment presents improvements to the technique wich presented better performance, improving the quality of the solutions generated with a minimum increase of time.

Keywords: Portfolio optimization. Multiobjective evolutionary algorithms. Combinatorial optimization. Parallel programming.

Lista de Figuras

Lista de Tabelas

Lista de Quadros

Lista de Algoritmos

Lista de Abreviaturas e Siglas

CVaR	*Conditional Value-at-Risk*
CPU	*Central Processing Unit*
GCC	GNU *Compiler Collection*
GNU	Gnu *is Not* Unix
GPU	*Graphics Processing Unit*

Lista de Símbolos

$\prec$	Domina
$\not\prec$	Não domina

Sumário

1 Introdução

1.1 Definição do problema

O mercado financeiro é o mecanismo que viabiliza, por meio de seus componentes, a transição de recursos entre aqueles que querem aplicar seus capitais e aqueles que necessitam do recurso. É, portanto, fundamental para o correto funcionamento da economia, ajustando o capital dos poupadores às necessidades dos tomadores. (BRUNI, 1998).

No mercado financeiro é possível a realização de investimentos financeiros, que são quaisquer instrumentos ou meios financeiros que podem gerar algum retorno em longo prazo. Esses investimentos são realizados por meio de ativos financeiros, que são títulos representativos de parte patrimonial ou dívida, que garante ao seu proprietário parte desse rendimento patrimonial. A combinação de todos os ativos financeiros de um investidor é denominada carteira de investimento ou portfólio financeiro.(CÂMARA et al., 2014).

O primeiro modelo matemático para otimização de carteiras de investimento foi desenvolvido por Markowitz (1952). Esse modelo leva em consideração o retorno e risco do investimento na seleção de ativos. Para mensurar o retorno, Markowitz utilizou a medida estatística de valor esperado e, para mensurar o risco, utilizou a variação da distribuição dos retornos gerados por determinado ativo.

Câmara et al. (2014) ressalta que a grande preocupação dos investidores, ao montar sua carteira de investimento, após a publicação da teoria de portfólios de Markowitz, passa a ser minimizar seus riscos e, ao mesmo tempo, maximizar seu rendimento. O autor ainda enfatiza que a diversificação da carteira de investimento é importante para essa tarefa, já que ela reduz os riscos do investimento, uma vez que o comportamento das ações de determinada empresa não afeta, necessariamente, o comportamento das ações de outra empresa.

Com o passar do tempo, foram criados novos modelos, que utilizam diferentes formas de mensurar o risco de portfólios, já que a gestão de risco, na década de 90, se tornou o principal segmento de estudo da teoria de finanças devido à grande volatilidade apresentada pelos principais indicadores financeiros e econômicos (ASSAF NETO, 2003).

Muitos estudiosos criticaram a utilização da variância como forma de mensurar o risco, já que a variância leva em consideração tanto os valores acima quanto abaixo da média. Dessa forma, surgiram modelos que medem risco com base apenas nos desvios negativos (denominados modelos da classe *downside risk*), tendo como exemplo o modelo de Rockafellar e Uryasev (2000), que utiliza a medida de risco CVaR (*conditional value-at-risk*), que define o risco como a média dos retornos nos $\alpha\%$ piores cenários, sendo α um

dado grau de confiança. A partir de então, o CVaR tem sido muito utilizado por ser a medida de risco mais pessimista, já que é definida de acordo com as piores perdas do investimento.

Juntamente com os avanços em relação aos modelos matemáticos, a teoria de finanças sofreu outras grandes alterações nas últimas décadas, segundo Bruni (1998), incorporando técnicas computacionais de otimização e identificação de padrões, além de técnicas ligadas às áreas de matemática e física.

Atualmente, para resolver muitos desses modelos de otimização que surgiram após o modelo proposto por Markowitz (1952), são utilizadas técnicas de otimização multiobjetivo, já que esses modelos levam em consideração dois objetivos: maximizar o retorno e minimizar o custo. Pantuza Júnior (2011) divide as técnicas para solução de problemas de otimização multiobjetivo em técnicas matemáticas determinísticas, técnicas heurísticas e meta-heurísticas, tendo algoritmos genéticos como exemplo mais utilizado.

Técnicas determinísticas, apesar de encontrarem as melhores respostas para um problema multiobjetivo, possuem alto tempo de processamento para determinadas classes de problemas, impossibilitando a utilização única dessa abordagem. Técnicas heurísticas são processadas rapidamente, mas podem retornar respostas ruins e, por isso, normalmente são utilizadas para refinar uma resposta ou gerar uma população inicial em um algoritmo genético, mas raramente utilizadas como forma única para se obter a resposta desejada. Por fim, meta-heurísticas podem ser executadas em tempo polinomial, independente do problema e, apesar de não garantir soluções ótimas, na prática geram boas soluções a partir da exploração do espaço de busca e refinamento das soluções (PANTUZA JÚNIOR, 2011).

1.2 Trabalhos relacionados

A partir da formulação do modelo de média-variância por Markowitz (1952), que será melhor explicado na subseção 2.3.1, a otimização de portfólios passou a ser objeto de estudo de diversos trabalhos científicos. Inicialmente, o modelo de otimização (modelo média-variância padrão) utilizava, como função objetivo, a minimização do risco (variância dos retornos da carteira) sujeito à duas restrições. A primeira restrição definia que o retorno da carteira deveria ser maior do que um determinado limiar e a segunda definia que cada ativo deveria possuir sua proporção de investimento na composição da carteira, de forma que a soma das proporções de todos ativos seja 100%. Vale ressaltar que, nesse modelo, as variáveis de decisão, proporções dos ativos, assumem valores contínuos.

O artigo de Chang et al. (2000) faz algumas modificações ao modelo média-variância padrão, adicionando restrição de cardinalidade, definindo uma quantidade de ativos a compor uma carteira e restrição de proporção, que define limites inferior e superior para a proporção de investimento de cada ativo que compõe a carteira. Utilizando algoritmo

genético, busca tabu e um algoritmo de *simulated annealing*, como técnicas para solucionar o modelo proposto, por meio dos gráficos da fronteira eficiente, mostrou-se que a inclusão das restrições fez com que o problema se tornasse mais complexo em relação ao modelo padrão devido à descontinuidade da fronteira eficiente causada pelas restrições. Apesar de não provar, testes empíricos sugerem que tal restrição transforma o problema em NP-completo.

Rockafellar e Uryasev (2000) propuseram um modelo que utiliza o CVaR (melhor explicado na seção 2) como medida de risco, ao contrário do modelo padrão de média-variância, que utiliza a variância dos retornos históricos da carteira. A partir desse trabalho, outros também começaram a utilizar o CVaR como medida de risco. A partir de então, percebe-se modelos de otimização de portfólios usando cada vez mais essa medida em lugar da variância, de Markowitz.

O trabalho de Roman, Mitra e Darby-Dowman (2007) apresenta um modelo que utiliza tanto a medida de risco variância de Markowitz, quanto o CVaR. Nesse caso, a abordagem multiobjetivo foi transformada em um modelo monobjetivo que possui, como objetivo, a minimização do risco variância e utiliza valores limites de retorno esperado e de risco CVaR como restrições para o problema. Utilizando um algoritmo de programação quadrática para solucionar o problema, foram selecionados cinco diferentes portfólios, sendo que cada um deles fora obtido definindo-se diferentes valores limites para risco CVaR, mas mantendo o retorno esperado igual para todos os casos. Os resultados apontaram pouca diferença nos retornos obtidos pelas diferentes carteiras em um ano de simulação.

Uma maior preocupação com a aplicação prática do modelo pode ser percebida em Soleimani, Golmakani e Salimi (2009). Esse trabalho, como o trabalho de Chang et al. (2000), acrescenta restrições ao modelo média-variância padrão, que são: restrição de cardinalidade, definindo a quantidade de ativos que devem compor uma carteira; restrição de lotes mínimos, definindo uma quantidade mínima de lotes que devem ser investidos em um ativo, caso esse ativo componha uma carteira; restrição de setores de capitalização, definindo uma proporção do capital investido em ativos de cada setor empresarial. O trabalho faz parte de poucos que utilizam variáveis de decisão inteiras, o que justifica o uso de restrição de lotes mínimos, demonstrando uma maior preocupação com a aplicação prática, já que ativos são vendidos em lotes. Um algoritmo genético foi utilizado para solucionar o problema e concluiu-se que a modelagem proposta é adequada para problemas reais, devido à comercialização de quantidades inteiras de ações, como já mencionado e, também como esperado, a técnica adotada foi capaz de realizar a otimização para uma grande quantidade de ativos financeiros.

Anagnostopoulos e Mamanis (2011) é um artigo que faz um estudo da otimização de portfólios, com ênfase maior nas meta-heurísticas para solucionar o modelo média-variância multiobjetivo (maximização do retorno e minimização do risco) com restrição

de cardinalidade, que faz com que a fronteira eficiente não seja contínua. Foi realizada a comparação de cinco algoritmos evolutivos multiobjetivos: *Niched Pareto Genetic Algorithm 2* (NPGA2), *Non-dominated Sorting Genetic Algorithm II* (NSGA-II), *Pareto Envelope-based Selection Algorithm* (PESA), *Strength Pareto Evolutionary Algorithm 2* (SPEA2) e *e-MultiObjective Evolutionary Algorithm* (e-MOEA). Por meio dos gráficos de fronteira eficiente e métricas de qualidade, constatou-se que o SPEA2 obteve os melhores resultados, seguido pelo NSGA-II, apesar do SPEA2 apresentar maior custo computacional.

Pode-se perceber um maior desenvolvimento da técnica de otimização em relação à criação do modelo em Deb et al. (2011). Nesse trabalho, o modelo proposto é o média-variância multiobjetivo com restrição de cardinalidade, definindo um intervalo mínimo e máximo para a quantidade de ativos em uma carteira, e restrição de proporção de investimento, definindo uma proporção mínima e máxima para investimento em um ativo, caso o ativo componha a carteira. Para otimização do modelo, o trabalho propõe um algoritmo evolutivo baseado no *Non-dominated Sorting Genetic Algorithm II* (NSGA-II). Com os resultados obtidos, concluiu-se que o algoritmo foi capaz de solucionar problemas de tamanho grande, com até 1000 ativos, apresentando uma eficiência satisfatória.

Beasley (2013) utiliza o modelo proposto por Soleimani, Golmakani e Salimi (2009) e adiciona, a esse modelo, o custo de transação do portfólio, além do processo de rebalanceamento, que consiste em levar em consideração um investimento passado para calcular custos de operações necessários para obter-se um novo portfólio otimizado, já que a quantidade e a quantia das operações para a transição de carteiras pode variar de acordo com esse investimento prévio. Dessa forma, esse modelo consegue representar aspectos práticos da compra e venda de ações. Os resultados sugerem que as restrições, incluindo a inclusão do custo de transação e o rebalanceamento, descrevem melhor operações reais.

Percebe-se que, além de Chang et al. (2000), nos demais trabalhos que consideram aspectos práticos do investimento, principalmente a restrição de cardinalidade, esse aumento da complexidade é percebido, de forma que métodos heurísticos são escolhidos para a solução desses modelos, dando mais credibilidade à conclusão de Chang et al. (2000), de que tal restrição aumenta a complexidade do problema e impossibilita o uso de métodos exatos em tempo polinomial.

Um algoritmo híbrido, que utiliza um algoritmo memético junto a uma técnica exata para modelos de otimização quadráticos foi proposto em Ruiz-Torrubiano e Suárez (2015). O algoritmo foi desenvolvido para solucionar um problema de seleção de carteiras que possui uma função objetivo que combina o retorno esperado com o risco, considerando medidas de média e variância de Markowitz (1952). As restrições consideradas são de limite de cardinalidade e limites mínimo e máximo da proporção de investimento por ativo. Como Beasley (2013), este trabalho considera custos de transação e o processo de rebalanceamento. A partir dos resultados, notou-se a diferença de se utilizar custos de transação,

proporcionando resultados mais reais e concluiu-se que o rebalanceamento e a limitação da cardinalidade podem proporcionar ganhos financeiros.

Hanaoka (2014) propôs dois modelos para a otimização de portfólios. O primeiro modelo proposto é parecido com o modelo de Soleimani, Golmakani e Salimi (2009), mas não possui a restrição de setores de capitalização e inclui uma restrição de capital de investimento, ou seja, um modelo média-variância biobjetivo com restrições de cardinalidade, lotes mínimos e de capital de investimento, estipulando um capital máximo disponível para o investimento. O segundo modelo utiliza as mesmas restrições do primeiro e o mesmo objetivo de maximização do retorno (definido como média histórica dos retornos da carteira), mas nesse modelo, a medida de risco utilizada é a CVaR de Rockafellar e Uryasev (2000) no lugar da variância de Markowitz (1952). Ambos os modelos utilizam variáveis de decisão inteiras. Para a otimização do modelo, o trabalho utilizou duas versões multiobjetivas do algoritmo evolutivo de evolução diferencial (ED): *Differential Evolution for Multi-objective Optimization* (DEMO) e *Pareto Differential Evolution Algorithm* (PDEA) e os resultados mostram que os dois algoritmos encontraram resultados satisfatórios, proporcionando lucro em simulações de um ano, para ambos os modelos.

Enquanto que a maior parte dos trabalhos que abordam a otimização de portfólios utilizam variáveis de decisão contínuas, poucos trabalhos propõem o uso de variáveis inteiras. Ainda assim, não foi encontrada na literatura, comparação entre essas diferentes modelagens. Acreditando que a utilização de variáveis inteiras seja mais eficiente, na prática, devido à natureza do problema, este trabalho propõe um modelo de otimização inteira e o compara com modelos contínuos, tentando auxiliar futuros trabalhos na decisão do modelo a ser utilizado.

Cheng e Gao (2015) estudam a restrição de cardinalidade em um modelo de média-CVaR monobjetivo, ou seja, um modelo de otimização que possui a minimização do risco CVaR como objetivo e, como restrição, um valor mínimo para o retorno esperado da carteira. Nesse trabalho foi visto que, fixando uma quantidade determinada de ativos, quando aumenta-se o número de ativos, o tempo de solução do modelo aumenta de forma exponencial e, com isso, conclui-se que o modelo média-CVaR com restrição de cardinalidade é um problema NP-completo.

Outro trabalho que utiliza modelagem multiobjetivo é o artigo de Ibrahim, El-Beltagy e Khorshid (2016). Nesse trabalho, seis modelos são divididos em dois grupos: modelos de média, que utilizam a média histórica da carteira como medida de retorno (proposto por Markowitz (1952)) e modelos de mediana, que utilizam a mediana da carteira como retorno (proposto por Benati (2015)). Dentre os modelos de média estão:

- modelo média-variância-obliquidade, que propõe a maximização da média, minimização da variância e a maximização da obliquidade dos retornos históricos do portfólio;
- modelo média-variância-obliquidade-curtose que adiciona, ao modelo anterior, o

objetivo de minimização da curtose dos retornos;
- modelo média-variância-obliquidade-curtose para retorno e liquidez, que possui os mesmos objetivos do anterior, mas leva em consideração a liquidez dos ativos em um determinado período.

Dentre os modelos de mediana estão:

- modelo mediana-MAD, com os objetivos de maximizar a mediana dos retornos e minimizar o MAD (*Median Absolute Deviation Model*), medida de risco simétrica;
- modelo mediana-VaR, que utiliza, como medida de risco, o VaR (*Value-at-Risk*), medida de risco assimétrica que deu origem ao CVaR;
- modelo mediana-CVaR, que utiliza o CVaR como medida de risco. Para resolver os modelos foi utilizado o algoritmo NSGA-II e simulações mostraram que, no geral, os modelos de mediana proporcionaram maiores retornos, com destaque para o modelo mediana-CVaR.

Vale observar que o modelo mediana-MAD proporcionou retornos parecidos com os modelos de média.

Cesarone, Moretti e Tardella (2016) mostrou que, geralmente, os portfólios que possuem os menores riscos são compostos de poucos ativos. No trabalho, foram utilizados modelos monobjetivos que visam a minimização do risco sujeito à restrição de cardinalidade, que define o número de ativos que uma carteira deve possuir. Foram utilizadas diferentes medidas de risco, como variância, CVaR e semi-MAD e, para todas, foi constatado que existe uma cardinalidade crítica que minimiza o risco e que, a partir dela, aumentar a quantidade de ativos significa aumentar o risco. Geralmente, essa cardinalidade crítica é menor que 15. Além disso, foi constatado que, quase sempre, existe um portfólio com 10 ou menos ativos tal que a diferença entre seu risco e o menor risco dentre todos os portfólios é menor que 1%.

A dissertação de Oliveira (2016) utiliza modelos de média biobjetivos com restrição de cardinalidade, com variáveis de decisão solucionados pelos algoritmos NSGA-II e SPEA2. O trabalho estuda a utilização de várias medidas de risco para esse modelo, tais como variância, semivariância, VaR, CVaR e GARCH e testes *out-of-sample* revelaram que, utilizando apenas ativos cujos retornos seguiam distribuição normal, as medidas de risco que proporcionaram maiores retornos acumulados foram semivariância, CVaR e GARCH, com destaque para semivariância, enquanto que, para ativos cujos retornos não seguem distribuição normal, o CVaR e o GARCH apresentaram melhores retornos acumulados. Além disso, como em Anagnostopoulos e Mamanis (2011), constatou-se uma superioridade do algoritmo SPEA2. Corroborando com a afirmação de Cesarone, Moretti e Tardella (2016), esse trabalho também mostrou melhor desempenho de carteiras com um menor número de ativos.

Barroso (2017) estudou a integração de análises técnicas da teoria de finanças com a otimização de portfólios financeiros. Para a otimização, foram utilizados modelos média-variância e média-CVaR biobjetivos. Foi observado, primeiramente, que a otimização conseguiu retornos superiores aos métodos de análise técnica, quando utilizados isoladamente. A otimização realizada após a pré-seleção de ativos por métodos da análise técnica proporcionou carteiras com retornos inferiores aos que a otimização pura alcançou. Por fim, métodos da análise técnica realizados após a otimização conseguiram os melhores resultados, superando, por pouco, retornos proporcionados pela otimização simplesmente. Nesse trabalho, é possível também comparar o modelo utilizando a medida de risco variância com o modelo utilizando CVaR. Constatou-se que, no geral, modelos que utilizam CVaR produzem carteiras que apresentam maior *Drawdown*, mas também maior *Drawup*. Concluiu-se que o modelo média-CVaR apresentou resultados mais satisfatórios por possuir uma maior razão entre *Drawup* e *Drawdown*.

Apoiando em resultados encontrados pelos trabalhos apresentados, o presente trabalho propõe um novo modelo que incorpora elementos considerados em vários desses trabalhos, como medida de risco CVaR, modelagem inteira e multiobjetivo, considerando restrição de cardinalidade e rebalanceamento. Vários dos parâmetros do modelo e algoritmos propostos também são baseados nesses trabalhos relacionados. Para testar a eficiência do modelo, ele é comparado com alguns modelos apresentados nessa seção.

1.3 Objetivos

O principal objetivo deste trabalho é comparar modelos e algoritmos utilizados na otimização de portfólios financeiros, tentando determinar uma modelagem e técnica que garantam maiores ganhos financeiros, na prática, para investidores.

Para alcançar o objetivo, alguns objetivos secundários deverão ser cumpridos. São eles:

(i) propor um modelo de otimização inteira de portfólios com restrições, que incorpore custos de transação e o processo de rebalanceamento, considerando, assim, aspectos que afetam a dinâmica do processo na prática. Esse modelo deverá ser, então, comparado com outros modelos existentes na literatura;

(ii) comparar o desempenho de técnicas computacionais de otimização utilizando uma versão contínua e outra discreta do modelo. Essas técnicas consistem em três algoritmos de otimização multiobjetivo, que serão posteriormente comparados entre si;

(iii) estudar a paralelização dos algoritmos, com a intenção de tornar as técnicas utilizadas mais rápidas, sem diminuir a precisão e qualidade dos resultados;

(iv) desenvolver um arcabouço computacional para interação com o usuário, em que este

pode escolher um modelo de otimização de portfólios, a técnica para sua execução e parâmetros específicos para o modelo e algoritmo desejado.

Este trabalho tenta responder algumas perguntas: O modelo inteiro é melhor que o modelo contínuo, mais utilizado, em relação aos retornos esperados e riscos dos portfólios gerados? O PDEA, utilizado para solucionar o modelo inteiro em Hanaoka (2014), é realmente a melhor meta-heurística para esse problema, levando em consideração a qualidade de suas soluções? A inclusão do processo de rebalanceamento no modelo de otimização inteira proporciona maiores ganhos financeiros para investidores? E, técnicas para otimização de portfólios podem ser rápidas o suficiente para serem utilizadas, por investidores, em investimentos de alta frequência (refeitos em curtos períodos de tempo na ordem de minutos ou segundos)?

1.4 Justificativa

Percebe-se, na literatura, uma utilização maior de modelos com variáveis de decisão contínuas nos problemas de otimização de portfólios financeiros, enquanto modelos com variáveis discretas foram pouco explorados. Entretanto, pode-se considerar o problema da otimização de portfólios como um problema discreto, já que as variáveis de decisão (quantidades de ações investidas para cada ativo) são, na prática, um valor inteiro, uma vez que ações são compradas em lotes. Além disso, não foi encontrada na literatura uma comparação entre as duas abordagens, utilizando modelos contínuos e discretos.

Alguns modelos encontrados na literatura, como o de Beasley (2013) e Ruiz-Torrubiano e Suárez (2015), incorporam custos de transação e rebalanceamento, tornando tais modelos mais práticos. Mas não foi encontrado, ainda, modelos de variáveis discretas e que considere esse efeito dos custos de transação. Desse modo, a elaboração de um modelo discreto com rebalanceamento é uma contribuição do presente trabalho.

A utilização de métodos heurísticos na solução do problema de otimização proposto se justifica no fato do modelo inteiro proposto não possuir método exato de solução em tempo polinomial, de acordo com Mansini, Ogryczak e Speranza (2015). Ainda segundo esses autores, a restrição de cardinalidade aumenta muito a complexidade do problema, sendo que quanto menor a cardinalidade máxima de um portfólio, mais tempo o algoritmo de método exato levará.

Alguns testes de Mansini, Ogryczak e Speranza (2015), utilizando o modelo média-MAD monobjetivo com cardinalidade máxima de 10 ativos e dados de entrada com centenas de ativos disponíveis para investimento, em computadores de ponta em 2015, levaram mais de 3 horas para a execução. Autores ainda afirmam que o problema se torna mais difícil de ser solucionado de forma exata se a medida CVaR for utilizada no lugar da MAD, e pode se tornar ainda mais demorado em um modelo multiobjetivo. Levando em conta a

proposta prática do trabalho, no auxílio à investidores ou analistas, o tempo se torna uma métrica muito importante e, assim, métodos heurísticos, que proporcionam bons resultados em um tempo que pode se adaptar às necessidades do usuário, foram escolhidos para a otimização do modelo.

Em relação à comparação entre técnicas computacionais para modelos discretos, Hanaoka (2014) comparou duas versões do algoritmo ED multiobjetivo, mas não se sabe se tal técnica supera o NSGA-II e o SPEA2, que apresentaram os melhores resultados em Anagnostopoulos e Mamanis (2011) e Oliveira (2016). Pela falta de comparações de diferentes técnicas para o problema proposto, determinar uma técnica eficiente para a otimização do modelo se torna uma tarefa relevante para a comunidade científica.

Por fim, sabe-se que técnicas computacionais para otimização multiobjetivo, em geral, possuem alto custo computacional, eventualmente demandando elevado tempo de processamento. Essa demora, além de limitar a quantidade de execuções da técnica meta-heurística (que é um processo estocástica), inviabiliza seu uso em operações de alta frequência (na ordem de minutos ou segundos). Dessa forma, a paralelização da técnica pode ser considerada como uma contribuição na viabilização de investimentos em uma frequência maior e no incentivo aos estudos da otimização de portfólios em alta frequência, tema pouco pesquisado atualmente.

O trabalho contribui com o desenvolvimento científico de diversas áreas acadêmicas, e em especial com a área de otimização de portfólios, devido a seu caráter multidisciplinar. As principais contribuições são a proposta de um novo modelo de otimização de portfólio e a comparação desse com outros já aceitos na comunidade científica, além da comparação de diversas técnicas da literatura e algumas modificações propostas no trabalho, identificando as melhores abordagens de acordo com a necessidade de cada investidor.

1.5 Motivação

Pode-se perceber variações muito grande dos preços dos ativos, em especial do mercado brasileiro, em curtos períodos de tempo. Essa volatilidade cria uma oportunidade para investidores obterem retornos financeiros elevados, maior do que investimentos conservadores mais conhecidos, como poupança, tesouro direto e outras opções de renda fixa. Mas, em contrapartida, essa volatilidade pode significar grandes perdas a alguns investidores, fazendo com que o investimento em bolsa apresente um risco elevado.

Nesse cenário, é importante para um investidor fazer com que seu investimento seja tão rentável quanto possível, mas que também apresente o menor risco possível. Dado que esses objetivos são conflitantes, o processo de composição de uma carteira de investimentos mais satisfatória para o investidor se torna complexo.

Dessa forma, a escolha de um estudo de otimização de portfólios, com foco na utilidade prática do método para investidores, tem como motivação uma oportunidade de contribuição para uma área ainda em aberto na teoria de finanças que mostrou possuir forte dependência de técnicas computacionais.

Além disso, recentemente, vem ocorrendo uma enorme expansão no uso de robôs de investimentos, sendo que quase todas grandes instituições de investimento utilizam robôs de investimentos altamente otimizados para negociações no mercado. Apesar disso, existe ainda a discussão não resolvida sobre qual método proporciona maiores lucros: robôs de investimento ou especialistas financeiros. Sabe-se que, em alguns casos, bons robôs proporcionam investimentos melhores que muitos especialistas, mas a grande capacidade do especialista de se adaptar à diferentes situações ainda é um desafio para esses robôs, de forma que os robôs podem ser utilizados também para auxiliar a decisão de especialistas.

Outro fator de motivação do trabalho é a ainda pequena parcela de brasileiros que investem no mercado de capitais, apesar do recente aumento desse número. O trabalho tem a pretensão de contribuir para o desenvolvimento de metodologias de investimentos que atraiam futuros investidores, aumentando assim, a quantidade desses.

1.6 Organização do trabalho

A fundamentação teórica, apresentada no capítulo 2, faz uma revisão literária dos temas necessários ao entendimento do trabalho, abordando conteúdo teórico acerca da teoria de finanças e técnicas meta-heurísticas, mais especificamente daquelas utilizadas no trabalho. O capítulo 3, referente à metodologia, explica com maior detalhes como os experimentos foram realizados, explicita materiais e métodos necessários em seu desenvolvimento. O capítulo 4, denominado resultados, informa os dados que foram utilizados e coletados, apresentando-os e discutindo-os de forma a tentar responder os questionamentos levantados pelo trabalho. Por fim, a conclusão no capítulo 5 apresenta, de forma objetiva, uma resposta para as hipóteses criadas com base nos resultados obtidos, além de sugerir trabalhos futuros para a continuação da pesquisa proposta nesse trabalho.

2 Fundamentação Teórica

Para melhor entendimento do trabalho, a compreensão de alguns conceitos relacionados é fundamental. Dessa forma, este capítulo apresenta uma revisão literária dos fundamentos teóricos nos quais este trabalho se baseia. Por ele ter um caráter interdisciplinar, são apresentados conceitos, tanto da área de finanças, quanto da área de otimização e de computação paralela.

2.1 Teoria e Métodos de Otimização

O processo de otimização está relacionado a problemas que envolvem tomadas de decisão, sendo utilizado na busca pela melhor decisão. A medida da qualidade de possíveis soluções ou decisões são definidas por funções objetivos em que deseja-se obter o maior (problemas de maximização) ou menor (problemas de minimização) valor possível. Dessa forma, teorias de otimização podem ser utilizadas em várias áreas de estudo, como engenharia, economia, computação, entre outras. Recentemente, a área de otimização tem recebido maior atenção devido ao rápido progresso da computação, contribuindo com a criação novos métodos de otimização, além de proporcionar a execução das técnicas de otimização em tempos cada vez menores (CHONG; ŻAK, 2009).

2.1.1 Modelos de Otimização

Problemas de otimização, em geral, podem ser representados pelo par (S, f), em que S é o conjunto de soluções factíveis para o problema e $f : S \rightarrow \mathbb{R}$ é a função objetivo. O conjunto de soluções factíveis corresponde a todas as respostas válidas para o problema proposto e a função objetivo representa um valor ou custo associado a uma solução factível. O maior objetivo de problemas de otimização é encontrar uma solução s^* tal que nenhuma outra solução $s \in S$ possui melhor função objetivo que essa, ou seja, $f(s^*) \leq f(s), \forall s \in S$, para um problema de otimização de minimização. Essa solução s^* é denominada ótimo global do problema (TALBI, 2009).

Outro conceito importante, em problemas de otimização, é o de ótimo local que, segundo Castro (2006), pode ser definido como uma solução x pertencente ao conjunto de soluções factíveis S e que, em uma dada vizinhança, não possua vizinhos melhores, ou seja, em relação a uma vizinhança N definida da seguinte forma: $N(x) = y \in S : dist(x,y) \leq \epsilon$, x é um ótimo local se $f(x) \leq f(y) \forall y \in N(x)$, considerando um problema de minimização, onde dist(.) é a função que determina a distância euclidiana entre as soluções x e y, e ϵ é uma constante estritamente positiva.

Modelos de otimização, segundo Talbi (2009), podem ser divididos em classes não exclusivas: modelos de programação matemática, de otimização combinatória ou não analíticos. Modelos de programação matemática podem ainda ser classificados quanto ao tipo da variável de decisão: inteiro, contínuo ou misto; e quanto às funções de objetivo e restrições: lineares ou não-lineares. Modelos lineares de programação matemática contínua são muito utilizados e podem ser representados de forma genérica da seguinte forma:

$$\min c' \cdot x$$

sujeito a:

$$A \cdot x \geq b$$

$$x \geq 0$$

onde $x \in \mathbb{R}^n$ é um vetor de variáveis contínuas de decisão, $c \in \mathbb{R}^n$ é um vetor constante que representa custos associados a cada variável de decisão, $b \in \mathbb{R}^m$ é um vetor de coeficientes constante que representa limites associados a cada restrição e $A \in \mathbb{R}^{nxm}$ é uma matriz constante de coeficientes que representa as restrições do problema. Deve-se observar que a representação genérica apresentada é utilizada para problemas de minimização. Considera-se, portanto, que problemas de maximização da função objetivo f são equivalentes aos problemas de minimização da função objetivo $-f$.

Em modelos lineares, tanto a função objetivo $c' \cdot x$ quanto as restrições $A \cdot x \geq b$ são funções lineares. A grande utilização desse modelo de otimização se deve ao fato de haver algoritmos exatos que resolvem o problema em tempo polinomial na prática, como o simplex e o algoritmo de pontos interiores. Tais métodos apresentam boa eficiência ao explorar o fato de que, em modelos lineares contínuos, o ótimo global se encontra em um dos vértices do politopo formado pela região factível do problema.

No mundo real, percebe-se muitos problemas que possuem funções objetivos ou restrições que não são lineares. Modelos não lineares contínuos, que consistem em minimizar ou maximizar a função objetivo $f : S \subset \mathbb{R}^n \to \mathbb{R}$, podem ser úteis para esses casos. Solucionar tais modelos, entretanto, é mais difícil do que solucionar os lineares. Para o caso específico dos modelos quadráticos convexos, que são os mais simples dentre os modelos não lineares, existem métodos exatos eficientes como o algoritmo de pontos interiores estendido ou método de programação quadrática sequencial, que utiliza o método de Newton ou quase-Newton para solucionar diretamente as condições de Karush–Kuhn–Tucker para o problema original (BAZARAA; SHERALI; SHETTY, 1993).

Para os problemas de otimização contínuos não lineares irrestritos, de acordo com Bazaraa, Sherali e Shetty (1993), podem ser utilizados vários métodos de busca local, como o método do Gradiente, método de Newton ou quase-Newton, sendo que para problemas convexos, o ótimo local corresponde ao ótimo global. Os problemas restritos,

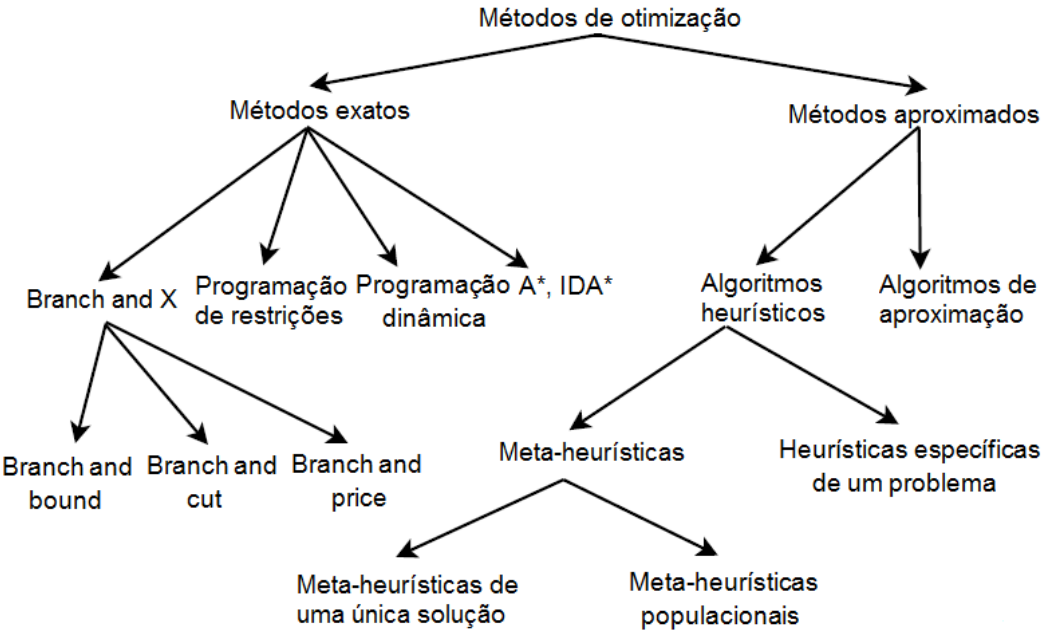
mais complicados, podem ser solucionados por métodos de penalização e barreia ou por métodos de Gradiente reduzido generalizado.

Modelos de otimização inteira são aqueles que possuem variáveis de decisão inteiras e, quando um modelo possui variáveis tanto inteiras quanto contínuas, é denominado problema de otimização inteira mista. Problemas de otimização inteira que possuem espaço de busca finitos são chamados de problemas de otimização combinatória. Modelos de otimização inteira e inteira mista são mais difíceis de resolver que modelos contínuos. Nesse caso, são utilizados algoritmos de enumeração, como o *branch-and-bound* para instâncias pequenas, já que tais métodos não são eficientes. Para instâncias maiores, o uso de técnicas heurísticas é recomendado (TALBI, 2009).

2.1.2 Métodos de Otimização

Como discutido brevemente na seção anterior, diferentes tipos de problemas possuem diferentes métodos de solução. Talbi (2009) afirma que o tipo de método a ser empregado depende da complexidade do problema. Para problemas NP-completos, métodos exatos gastam tempo não polinomial em relação ao tamanho da instância, de forma que, para instâncias de tamanho grande, métodos aproximados podem ser uma alternativa mais viável. Problemas de otimização linear contínuo pertencem à classe P de problemas, ou seja, podem ser solucionados em tempo polinomial por algoritmos exatos, enquanto que o problema do caixeiro viajante, coloração de grafos e problema de roteamento de veículo são exemplos de problemas da classe NP-completo.

Figura 1 – Métodos clássicos de otimização



Fonte: Adaptado de Talbi (2009)

Métodos exatos são aqueles que garantem encontrar o ótimo global do problema, enquanto que métodos aproximados podem gerar soluções de boa qualidade, mas não há a garantia de se encontrar o ótimo global e, apesar dessa desvantagem, métodos aproximados possuem tempo de execução em tempo aceitável, na prática, enquanto métodos exatos podem gastar tempo inviável dependendo do problema. Figura 1 apresenta ramificações dos métodos de otimização que partem desses dois principais tipos: métodos exatos e aproximados. Os exatos são divididos, então, da seguinte forma:

- Programação dinâmica utiliza a estratégia "dividir para conquistar", dividindo o problema, recursivamente, em problemas menores e o resultado depende de uma sequência de decisões parciais. O método evita a enumeração do espaço de busca podando divisões que não levarão ao ótimo global;
- Os algoritmos da família do *branch and bound* e do A* baseam-se na enumeração implícita das soluções no espaço de busca. Em ambas as famílias são construídas árvores cuja raiz representa o problema e as ramificações são divisões que levam a possíveis soluções (folhas). *Branch and bound* e algoritmos de sua família também evitam a enumeração completa realizando podas em nós que não levam à solução ótima;
- Programação de restrições é um paradigma de programação que envolve conceitos de árvore de busca e implicações lógicas. Relações de restrição são formadas entre variáveis, de forma que um modelo de otimização pode ser modelado como um conjunto de variáveis ligado por um conjunto de restrições que relacionam tais variáveis.

Métodos aproximados proporcionam boas soluções, mas não garantem encontrar o ótimo global e existem duas subclasses de algoritmos: os de aproximação e os heurísticos. Os algoritmos de aproximação garantem soluções que estejam longe do ótimo global s em, no máximo, um limite de ϵ vezes. Esses algoritmos devem possuir complexidade computacional polinomial e gerar uma solução α tal que, num problema de minimização, as seguintes sentenças sejam verdadeiras (TALBI, 2009):

$$\alpha \leq \epsilon \cdot s, se\, \epsilon > 1$$
$$\epsilon \cdot s \leq \alpha, se\, \epsilon < 1.$$

Algoritmos heurísticos se dividem em heurísticas específicas e meta-heurísticas, sendo que ambos objetivam proporcionar boas soluções em um tempo razoável, mesmo para problemas de grande escala. De acordo com Talbi (2009), a palavra heurística tem origem na palavra grega *heuriskein*, que significa "arte de descobrir novas estratégias para solucionar problemas". Heurísticas são, geralmente, metodologias que encontram ótimos locais, mas existem heurísticas específicas para alguns problemas que podem proporcionar soluções satisfatórias. Ainda segundo Talbi (2009), o sufixo meta também possui origem

grega e significa "metodologia superior". Meta-heurísticas podem ser consideradas metodologias superiores em relação às heurísticas pela possibilidade de se encontrar não apenas um ótimo local, aumentando a chance de encontrar melhores soluções e, eventualmente, o ótimo global.

Meta-heurísticas possuem dois objetivos conflitantes: diversificação e intensificação. Diversificação consiste na boa exploração do espaço de busca, mecanismo que possibilita que o método não se estabilize ou estagne após encontrar um ótimo local. Intensificação, por outro lado, consiste na boa exploração de espaços próximos às melhores soluções encontradas, à procura exatamente de um ótimo local. Meta-heurísticas de soluções únicas transformam uma solução durante o processo de busca e possuem a intensificação mais acentuada. Já meta-heurísticas populacionais envolvem uma população inteira que interage e transformam-se durante o processo de busca, possuindo o mecanismo de diversificação mais acentuado.

Algumas meta-heurísticas são inspiradas em fenômenos da natureza e são chamadas de evolutivas. Castro (2006) afirma que a computação evolutiva surgiu da observação de estratégias encontradas na própria natureza para a solução de problemas complexos, tais como procura por alimentos, fuga de predadores, organização dos lares e busca por ambientes com melhores condições climáticas. Algoritmos evolutivos são inspirados em estratégias presentes na natureza, incorporando processos que imitam esses fenômenos naturais, mas sem a preocupação da modelagem acurada dos fenômenos observados. Alguns exemplos são: o algoritmo Colônia de Formigas, inspirado na busca de formigas por alimentos; Otimização por Nuvem de Partículas, inspirado no comportamento de um bando de pássaros em voo; Algoritmo Genético, inspirado na teoria de evolução genética de Darwin; e Algoritmo Imunológico, inspirado no funcionamento do sistema imunológico dos mamíferos.

Algoritmos evolutivos podem ser utilizados em problemas de otimização e são indicados para problemas cuja solução exata não pode ser obtida em um tempo viável utilizando algoritmos disponíveis atualmente. A maioria dos algoritmos evolutivos são meta-heurísticas populacionais baseadas na teoria de evolução de Darwin e reproduzem três processos análogos aos que a teoria propõe: reprodução de uma população de indivíduos, variação genética dos novos indivíduos gerados e seleção natural dos indivíduos mais aptos. Como na teoria de Darwin, espera-se que a população possua indivíduos cada vez mais aptos ao seu ambiente (CASTRO, 2006).

Tendo em vista que algoritmos evolutivos são métodos meta-heurísticos populacionais, Castro (2006) afirma que tais algoritmos podem ser vistos como sistemas baseados em agentes, que pode ser qualquer coisa que percebe seu ambiente por meio de sensores e age sobre esse ambiente.

Em algoritmos baseados na teoria de seleção natural de Darwin, os agentes são

indivíduos e seu conjunto forma uma população. Cada indivíduo é representado geneticamente por um cromossomo composto por genes, sendo que cada um desses genes podem ser valores binários, reais, *strings*, vetores ou outra forma abstrata de dados que represente uma variável de decisão num modelo de otimização. Alelos correspondem ao conjunto de valores que cada gene pode assumir. Algoritmos evolutivos fazem, portanto, uma metáfora dos processos de recombinação, mutação e seleção natural a partir dessa representação genética computacional dos indivíduos (CASTRO, 2006).

O Algoritmo 1, de Castro (2006), representa uma forma genérica para um algoritmo evolutivo, sendo que os parâmetros de entrada são a probabilidade de cruzamento pc e a probabilidade de mutação pm e a saída é a população de agentes ou indivíduos P. Em um problema de otimização, o método "inicializa" pode atribuir diferentes soluções para os componentes da população, o método "avalia" atribui valor a cada solução com base na função objetivo, privilegiando as melhores soluções. O método "reproduza" combina informações de diferentes soluções para gerar outras novas com probabilidade pc, o "perturba" altera algumas dessas novas soluções com uma probabilidade pm e, por fim, "seleciona" escolhe as soluções para continuar na população de acordo com a avaliação atribuída a cada solução.

Algoritmo 1: ALGORITMO EVOLUTIVO GENÉRICO

Entrada: pc, pm
Saída: P
início
  INICIALIZA P
  $f \leftarrow$ AVALIA(P)
  $P \leftarrow$ SELECIONA(P, f)
  $t \leftarrow 1$
  enquanto *critério de parada não satisfeito* **faça**
    $P \leftarrow$ REPRODUZA(P,f,pc)
    $P \leftarrow$ PERTURBA(P,pm)
    $f \leftarrow$ AVALIA(P)
    $P \leftarrow$ SELECIONA(P, f)
    $t \leftarrow t + 1$
  fim
fim
retorna P

2.1.2.1 Busca Local

De acordo com Talbi (2009), a busca local é, provavelmente, o método heurístico mais antigo e simples. Consiste em melhorar uma solução inicial, parâmetro do método. De acordo com uma vizinhança definida, o método troca a solução corrente por uma solução vizinha melhor e termina quando nenhum vizinho é melhor que a solução corrente. A seleção

do vizinho que deverá tomar o lugar da solução corrente depende da estratégia adotada, sendo que as mais utilizadas são:

- *Best improvement*: nessa estratégia, deve-se selecionar o melhor vizinho da solução corrente para tomar seu lugar. Essa estratégia pode gastar muito tempo dependendo da vizinhança adotada;
- *First improvement*: seguindo essa estratégia, ao explorar os vizinhos de uma solução corrente, o primeiro melhor vizinho da solução corrente passa a ser a solução corrente e assim sucessivamente, até que se encontre uma solução que não possui vizinhos melhores. Nesse caso, todos seus vizinhos são explorados, como no *best improvement*. O fato dessa estratégia não avaliar todos os vizinhos de todas as soluções faz com que ela seja mais rápida na maioria dos casos e, no pior caso, igual o *best improvement*.

2.1.2.2 Algoritmo Genético (AG)

AG é um algoritmo evolutivo baseado na teoria da evolução de Darwin e, por isso, utiliza o vocabulário relacionado à área de genética. Dessa forma, o algoritmo utiliza uma população de indivíduos, representados por cromossomos, que por sua vez são compostos por genes. Foi proposto inicialmente por Holland (1975) e seus procedimentos podem ser representados pelo algoritmo 1.

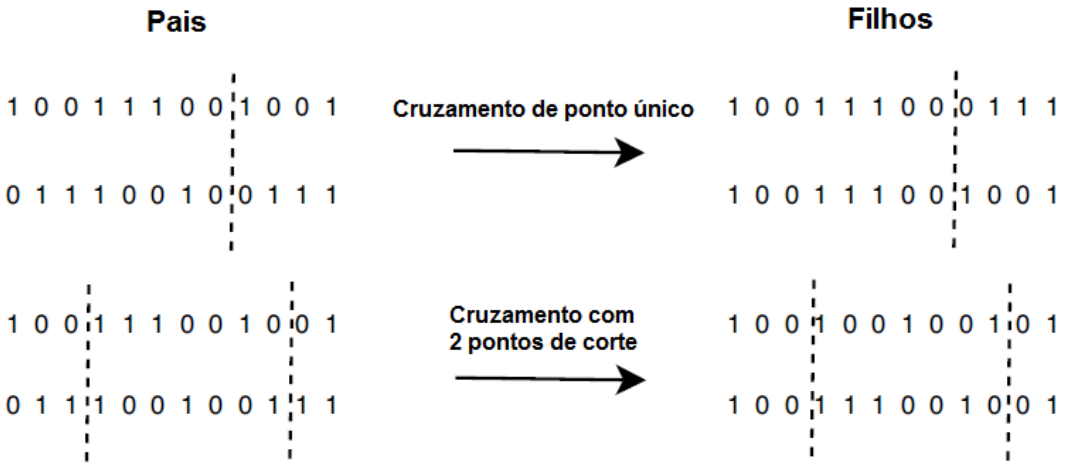
A **inicialização** pode ser realizada de forma aleatória, atribuindo diferentes soluções aos indivíduos por meio de uma função aleatória que leva em conta restrições e domínios das variáveis do problema. Atribuir aos indivíduos soluções igualmente espaçadas no espaço de soluções é outra forma de inicializar a população inicial. E uma terceira forma de inicializar a população é utilizando um método heurístico que gere boas soluções, que serão atribuídas aos indivíduos (TALBI, 2009).

Avaliação atribui aos indivíduos da população valores de aptidão (*fitness*) relacionados à função objetivo, podendo ser inclusive o próprio valor dessa função. Uma forma muito utilizada de avaliar os indivíduos é atribuindo probabilidades p de reproduzir para cada indivíduo, sendo $p_i = f_i / \sum_{j=1}^{N} f_j$, onde f_i é a função objetivo aplicada ao indivíduo i e N é o tamanho da população. Após a atribuição dessas probabilidades um método determinado roleta seleciona N indivíduos para a reprodução. É importante observar que o mesmo indivíduo pode ser escolhido mais de uma vez para reproduzir (CASTRO, 2006).

Reprodução ou **cruzamento** de indivíduos é um processo análogo ao cruzamento biológico, no qual dois indivíduos (pais) combinam seus materiais genéticos para gerarem dois novo indivíduos (filhos). Primeiramente, foi proposto o cruzamento de ponto único, em que um ponto do cromossomo é escolhido aleatoriamente e o primeiro filho é formado com os genes à esquerda daquele ponto do primeiro pai e genes à direita daquele ponto do segundo pai, enquanto que o segundo filho é formado pelos genes complementares

aos genes dos pais que geraram o primeiro filho. Esse operador é um caso específico do operador de n pontos de corte, em que n pontos do cromossomo é escolhido aleatoriamente e os genes dos pais são atribuídos ao primeiro e segundo filho intercalando a cada ponto de corte, de forma que os filhos sejam formados, um em relação ao outro, sempre pelos genes complementares dos pais (TALBI, 2009). A Figura 2 ilustra o cruzamento de ponto único e o cruzamento com dois pontos de corte para um problema em que os genes podem assumir apenas valores binários (zero ou um).

Figura 2 – Operadores de cruzamento com 1 e 2 pontos de corte



Fonte: Adaptado de Talbi (2009)

Outro operador de cruzamento muito utilizado é o cruzamento uniforme, em que cada gene do primeiro pai é designado aleatoriamente ao primeiro ou segundo filho e o filho que não foi selecionado recebe o gene do segundo pai na mesma posição daquele gene no primeiro pai. Independente do operador, cada par de pais reproduz com uma probabilidade pc, sendo que aqueles que não reproduzem são simplesmente copiados para a próxima geração (TALBI, 2009). A Figura 3 mostra o funcionamento desse operador para um problema em que as variáveis de decisão são binárias (podem assumir valor zero ou um).

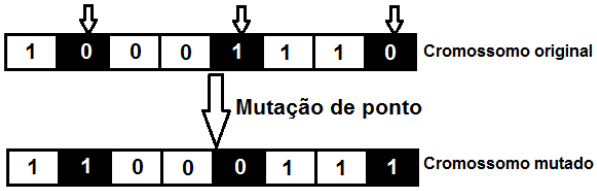
Figura 3 – Operadores de cruzamento uniforme

Fonte: Adaptado de Talbi (2009)

Em algoritmos genéticos, a pertubação é realizada através da **mutação**, processo análogo a mutação genética, podendo incorporar características aos indivíduos cujos pais

não possuem, garantindo diversidade à população. A mutação de ponto, muito utilizada, modifica o valor de cada gene de um indivíduo com uma probabilidade pm (CASTRO, 2006). Para um gene com alelo binário, por exemplo, caso seja selecionado para sofrer mutação, seu valor é trocado pelo outro (zero é alterado para um e um é alterado para zero), como ocorre no exemplo mostrado pela Figura 4.

Figura 4 – Mutações de ponto para um cromossomo de tamanho 8



Para a realização da **seleção**, muitos métodos podem ser utilizados. O mais básico deles é escolher todos os indivíduos filhos para a próxima iteração ou geração. Outro método que pode ser utilizado é a roleta, da mesma forma que foi mencionada para a seleção dos pais no cruzamento. Independente do método utilizado na seleção, pode-se optar por utilizar o elitismo, ou seja, um número arbitrário das melhores soluções de uma geração, sempre são copiados para a população da próxima geração (TALBI, 2009).

Também existem várias formas de estabelecer um **critério de parada**, sendo que utilizar um limite máximo para as gerações é a forma mais simples, porém muito utilizada. Outros exemplos são: encontrar o valor ótimo, atingir um tempo máximo de execução ou várias execuções sem que haja melhora na população. Podem ser utilizadas combinações desses critérios ou vários outros não citados (TALBI, 2009).

Talbi (2009) também afirma que o algoritmo genético pode ser utilizado junto com o método heurístico de busca local, formando um método meta-heurístico híbrido denominado algoritmo memético.

2.1.2.3 Algoritmo de Evolução Diferencial (ED)

O ED foi proposto por Storn e Price (1995) e consiste em um algoritmo evolutivo baseado na combinações de vetores com o objetivo de guiar os vetores da população em direção às boas soluções. Foi proposto inicialmente para problemas de variáveis contínuas e se mostrou muito eficiente para esse grupo de problemas.

Os procedimentos desse algoritmo seguem o pseudocódigo do algoritmo 1, exceto pelo fato do ED realizar a perturbação (mutação) antes da recombinação (cruzamento). A população inicial é gerada da mesma forma que no algoritmo genético, mas cada indivíduo é considerado um vetor de pontos flutuantes (TALBI, 2009).

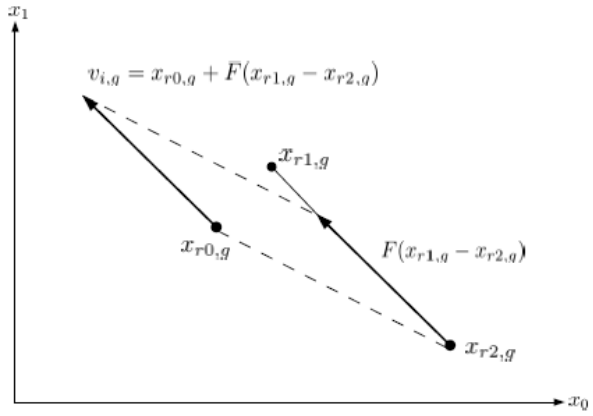
Após a geração da população inicial, seguida de sua avaliação e seleção, o algoritmo executa a **mutação**, utilizando operadores diferentes dos operadores clássicos utilizados

no AG. Seu operador realiza o seguinte procedimento, para cada novo indivíduo v_i gerado pela mutação (vetor mutante), com i variando de 1 até N (tamanho da população), em uma dada geração g (PRICE; STORN; LAMPINEN, 2005):

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \tag{1}$$

onde $F \in [0,2]$ é um fator de escala e parâmetro do algoritmo, $r0$, $r1$ e $r2$ são diferentes índices de vetores da população corrente, $x_{r0,g}$ é denominado vetor base e $(x_{r1,g} - x_{r2,g})$, vetor diferencial. Esse procedimento é ilustrado pela Figura 5.

Figura 5 – Mutação no algoritmo ED



Fonte: Price, Storn e Lampinen (2005)

De acordo com Talbi (2009), a forma com que o vetor base é selecionado depende da estratégia adotada no desenvolvimento do ED. A estratégia é definida pela notação DE/x/y/z, em que x determina a forma de seleção do vetor base, que pode ser aleatória, por exemplo, y determina a quantidade de vetores diferenciais, que pode ser 1, como na equação 1, e z determina o esquema de cruzamento, que possui o binomial como padrão. Um exemplo de estratégia muito utilizada é a DE/rand/1/bin, que seleciona o vetor base de forma aleatória, utiliza 1 vetor diferencial e realiza o cruzamento binomial.

O cruzamento no ED também utiliza operadores diferentes do operadores clássicos do AG. Um vetor experimental u_i é criado em cada geração e combina elementos dos vetores originais com os vetores mutantes. O operador de cruzamento depende da estratégia adotada, sendo que a estratégia "bin" (cruzamento binomial) determina o seguinte procedimento, dado um par composto pelo indivíduo original $x_{i,g}$ e seu respectivo vetor mutante $v_{i,g}$:

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{se } rand_j \leq CR \text{ ou } j = rand \\ x_{j,i,g}, & \text{caso contrário} \end{cases}$$

onde j indica a posição de um elemento do vetor, $rand_j \in \{0,1\}$ é um número aleatório para cada posição j dos vetores e $rand$ é um valor aleatório que define uma posição do vetor experimental que deverá conter, obrigatoriamente, o elemento do vetor mutante naquela mesma posição.

A seleção dos indivíduos para a próxima geração é realizada da seguinte forma, considerando um problema de minimização:

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{se } f(u_{i,g}) < f(x_{i,g}) \\ x_{i,g}, & \text{caso contrário} \end{cases}$$

onde CR, parâmetro do algoritmo, é a constante de *crossover*, $f(.)$ representa a função objetivo, $u_{i,g}$ é o vetor experimental de índice i na geração g e $x_{i,g}$ é o vetor original de índice i da população na geração g.

2.2 Otimização Multiobjetivo

Modelos de otimização multiobjetivo possuem várias funções objetivos, cada um representando um critério diferente, sendo que eles são, na maioria das vezes, conflitantes. Como no problema de produção em que deseja-se minimizar o custo e, ao mesmo tempo, maximizar a qualidade dos produtos fabricados. Nesse caso, não existe uma única solução ótima, mas um conjunto de soluções não dominadas chamado de conjunto Pareto-ótimo. Uma solução pertence a este conjunto se não existe outra solução que a domine, ou seja, que a supere em pelo menos um objetivo e que não seja dominada por ela em nenhum objetivo.

2.2.1 Conceitos básicos

Um problema de otimização multiobjetivo, considerando objetivos de minimização, pode ser representado genericamente da seguinte forma (TALBI, 2009):

$$\min F(x) = (f_1(x), f_2(x), ..., f_n(x))$$

sujeito a:

$$x \in S$$

onde $F(x)$ é o conjunto de tamanho n, com $n \geq 2$, de funções objetivos do problema, $x = (x_1, x_2, ..., xk)$ é o vetor de variáveis de decisão e S é o conjunto de todas soluções factíveis, levando em conta as restrições e domínios do problema.

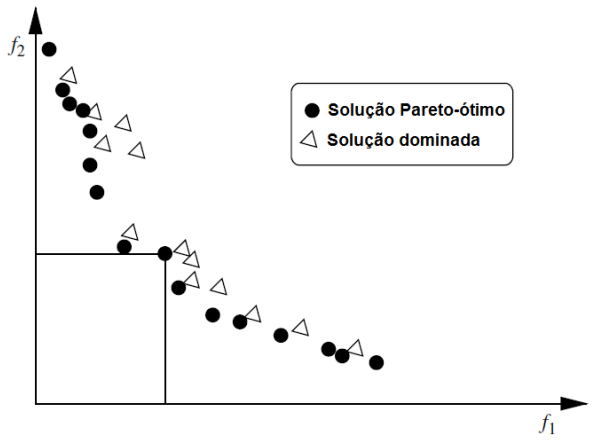
O entendimento do conceito de **dominância** é importante para a compreensão de problemas multiobjetivos. Pode-se dizer que uma solução s_1, com vetor de funções

objetivos $u = \{u_1, ..., u_n\}$, é melhor que outra solução s_2, com vetor de funções objetivos $v = \{v_1, ..., v_n\}$, se u domina v ($u \prec v$), ou seja, nenhuma função objetivo de v possui valor menor que seu respectivo valor em u e, além disso, pelo menos para um objetivo, o valor da função objetivo de u é menor que o de v. Formalmente, $\forall i \in \{1, ..., n\} : u_i \leq v_i \wedge \exists i \in \{1, ..., n\} : u_i < v_i$.

Dominância fraca de u sobre v ($u \preceq v$) ocorre quando nenhum valor de função objetivo de v é menor que o valor dessa função para u, ou ainda, $\forall i \in \{1, ..., n\} : u_i \leq v_i$. Pode-se observar que, sempre que u domina v, também ocorre a dominância fraca de u sobre v, mas pode ocorrer dominância fraca de u sobre v sem que u domine v e, nesse caso, u e v são não-dominados entre si. Também existe a dominância forte ou estrita, que ocorre quando um vetor possui os valores menores que de um outro vetor para todos os objetivos. Ou seja, u domina estritamente v ($u \prec\prec v$) se $\forall i \in \{1, ..., n\} : u_i < v_i$ e, consequentemente, u domina v.

Utilizando o conceito de dominância, Talbi (2009) define o **conjunto Pareto-ótimo** P^* como o conjunto que possui todas as soluções $x^* \in S$, tais que nenhuma outra solução de S a domine, isto é, $P^* = \{x^* \in S : \forall x \in S, F(x) \not\prec F(x^*)\}$. A **fronteira de Pareto** FP^* é a imagem do conjunto Pareto-ótimo em relação aos seus valores de funções objetivos, ou seja, $FP^* = \{F(x^*), x^* \in P^*\}$. A Figura 6 mostra uma fronteira de Pareto, formada pelas soluções representadas por círculos, para um problema de minimização biobjetivo, além de soluções dominadas, representadas por triângulos. Na imagem pode-se observar que para cada solução dominada existe pelo menos um círculo que possui valores de f_1 e de f_2 menores que os seus. Verifica-se também que as soluções do conjunto Pareto-ótimo não são dominadas por nenhuma outra e se dispõem de forma parecida com uma curva em que pontos acima dessa curva são dominados.

Figura 6 – Fronteira de Pareto e soluções dominadas



Fonte: Adaptado de Talbi (2009)

2.2.2 Meta-heurísticas para otimização multiobjetivo

A complexidade computacional do problemas de otimização multiobjetivos aumentam muito com o tamanho do problema. Assim, problemas com dificultadores como funções não lineares ou variáveis inteiras e que, principalmente, não possuem métodos exatos de solução em tempo polinomial de execução, se tornam ainda mais difíceis de serem solucionados quando o problema é multiobjetivo. Dessa forma, a utilização de meta-heurísticas, que proporcionam boas soluções em tempo aceitável na prática, é muito frequente na solução de tais problemas (TALBI, 2009).

Primeiramente serão apresentadas métricas de avaliação de soluções geradas por técnicas meta-heurísticas em problemas multiobjetivos e, posteriormente, algumas dessas técnicas serão apresentadas.

2.2.2.1 Medidas de desempenho

Em grande parte dos problemas de otimização multiobjetivo, as meta-heurísticas possuem os objetivos de aproximarem as soluções o máximo possível da fronteira de Pareto (boa cobertura), além de espalhar essas soluções de forma mais uniforme possível no espaço de soluções (boa diversidade). Dessa forma, existem algumas métricas que avaliam o desempenho das soluções geradas por uma meta-heurística em relação aos dois objetivos: cobertura e diversidade das soluções.

Métricas de diversidade, de acordo com Jiang et al. (2014), medem a qualidade do espalhamento e distribuição das soluções no conjunto ótimo proporcionado pelo algoritmo que está sendo avaliado. O espalhamento das soluções se baseia na distância entre os extremos do conjunto e a distribuição tem como referência o espaçamento entre as soluções vizinhas, sendo que deseja-se um conjunto com maior espalhamento possível e soluções vizinhas com espaçamento mais uniforme possível.

Um exemplo de métrica de diversidade é a **métrica** Δ, que é definida da seguinte forma (DEB et al., 2000):

$$\Delta(S) = \sum_{i=1}^{|S|-1} \frac{(d_i - \overline{d})}{|S| - 1}$$

onde S é o conjunto de soluções ótimas gerado pelo algoritmo avaliado, ou seja, o conjunto formado por todas as soluções que não são dominadas por nenhuma outra solução gerada, d_i é a distância Euclidiana entre as soluções consecutivas de S e $\overline{d}$ é a média dos valores de $d_i, \forall i \in 1, ..., |S| - 1$. Quanto menor é o valor da métrica Δ, mais bem espalhadas as soluções estão e, portanto, mais diversificadas elas são, sendo que o valor zero indica que o espaçamento entre todas as soluções consecutivas é exatamente o mesmo.

Métricas de capacidade indicam a quantidade ou proporção de soluções que não são dominadas. Métricas mais simples utilizam o próprio tamanho do conjunto de soluções ótimas ou a proporção desse valor em relação a todas as soluções geradas pelo algoritmo avaliado. Apesar dessas métricas que avaliam apenas soluções de um único algoritmo proporcionarem algumas informações tal como a convergência do algoritmo, ainda não são capazes de avaliar a cobertura desse algoritmo, ou seja, quão perto as soluções geradas estão do conjunto Pareto-ótimo. Como na maioria dos casos não se conhece o conjunto Pareto-ótimo, algumas métricas de capacidade comparam as soluções geradas por um algoritmo com um conjunto de referência ou com o conjunto de soluções gerado por outro algoritmo para, assim, avaliar a cobertura do algoritmo estudado em relação a outro (JIANG et al., 2014).

A **métrica C** (*Coverage of two sets*) mede a cobertura do conjunto de soluções ótimas S_1 de um algoritmo em relação ao conjunto S_2 de outro algoritmo da seguinte forma (ZITZLER; THIELE, 1998):

$$C(S_1, S_2) = \frac{|\vec{s_2} \in S_2 | \exists \vec{s_1} \in S_1 : \vec{s_1} \preceq \vec{s_2}|}{|S_2|}.$$

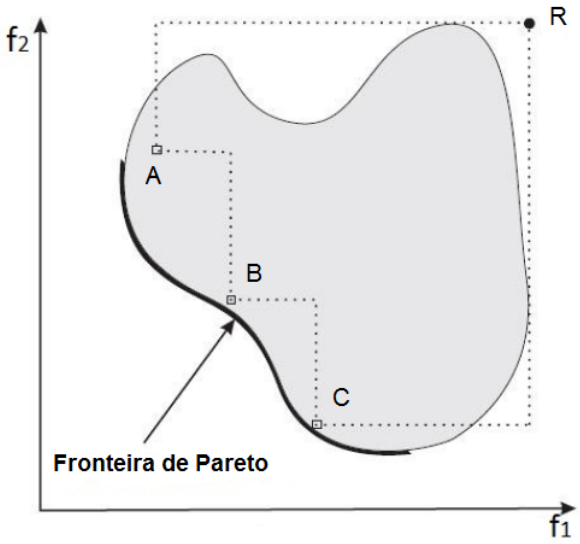
Essa métrica indica a fração das soluções do conjunto S_2 que são dominadas por alguma solução do conjunto S_1. Dessa forma, quanto menor o valor dessa métrica, melhor é a cobertura das soluções do primeiro algoritmo em relação ao segundo.

A **métrica HV** (hipervolume), proposta por Zitzler e Thiele (1998), mede o volume do espaço dominado pela fronteira de Pareto gerada pelo algoritmo de otimização multiobjetivo. Então, quanto mais próximas da fronteira Pareto-ótimo as soluções estão e quanto mais uniformemente distribuídas estão, maior o valor dessa métrica. Assim, HV é uma medida que pode analisar tanto a cobertura quanto a diversidade das soluções. O indicador também garante 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. Essa métrica, portanto, pode ser melhor definida da seguinte forma:

$$HV(S,R) = volume(\bigcup_{i=1}^{|S|} v_i)$$

onde S é o conjunto de soluções ótimas gerado pelo algoritmo, R é uma solução utilizada como referência, normalmente dominada por todas as soluções de S e v_i é o hipercubo formado por uma solução $\vec{s_i} \in S$ e a solução de referência R. Um exemplo é mostrado na Figura 7, considerando $S = \{A,B,C\}$ para um problema de minimização para os dois objetivos e, por isso, R é um ponto com valores que superam os demais em ambos os objetivos. Nesse caso, em que há apenas duas funções objetivos, o hipercubo é um retângulo e o valor do hipervolume é a área do espaço formado pela união desses retângulos.

Figura 7 – Hipervolume de um conjunto de soluções arbitrário



Fonte: Adaptado de Jiang et al. (2014)

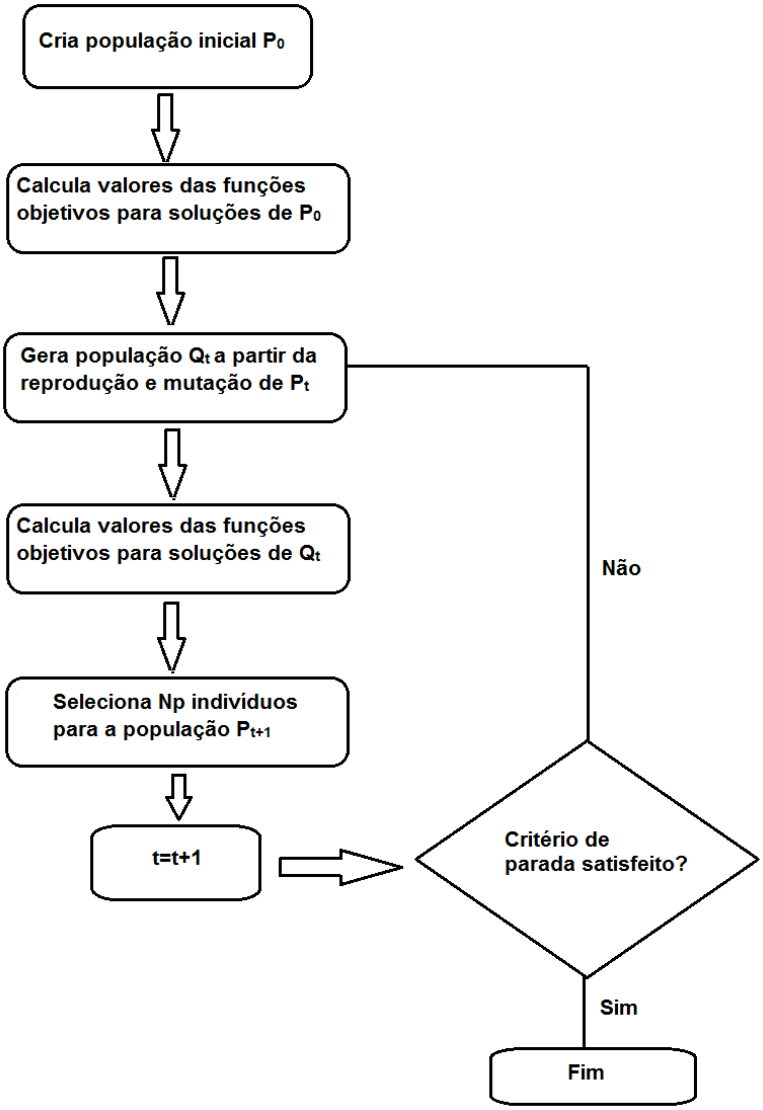
2.2.3 NSGA-II (*Non-dominated Sorting Genetic Algorithm II*)

Proposto por Deb et al. (2000), o NSGA-II é um algoritmo genético elitista para problemas de otimização multiobjetivo. Seu caráter elitista se deve ao fato do algoritmo beneficiar as melhores soluções no processo de seleção. O fluxograma do algoritmo é mostrado na Figura 8, que inicia com a criação da população inicial P de tamanho N_p, segue com o cálculo das funções objetivos dos indivíduos ou soluções. Posteriormente, o algoritmo entra num laço de repetição que inicia com reprodução e mutação da população corrente, utilizando os mesmos operadores utilizados no algoritmo genético, gerando uma população Q também de tamanho N_p. Esse laço segue com o cálculo de funções objetivos dos novos indivíduos criados e, por último, a seleção dos N_p melhores indivíduos para a próxima geração. O laço é interrompido quando o critério de parada é satisfeito.

A principal diferença do NSGA-II para um algoritmo genético monobjetivo é, além de possuir mais de uma função objetivo, o seu processo de seleção de indivíduos, que tenta guiar as soluções de forma a criar um conjunto com boa cobertura e diversidade. Dessa forma, dois novos conceitos são incorporados pelo algoritmo: ordenação não-dominada e distância de aglomeração.

A **ordenação não-dominada** ordena as fronteiras de Pareto formadas pelos indivíduos, de forma que a primeira fronteira é formada pelos indivíduos que não são dominados por nenhum outro. A segunda fronteira é formada por indivíduos que não são dominados por nenhum outro, exceto aqueles pertencentes à primeira fronteira, e assim sucessivamente até que todos os indivíduos estejam em uma fronteira, como ilustra a Figura 9. No NSGA-II o algoritmo utilizado para esse procedimento é o *Fast Nondominated Sorting*, também

Figura 8 – Fluxograma básico do algoritmo NSGA-II

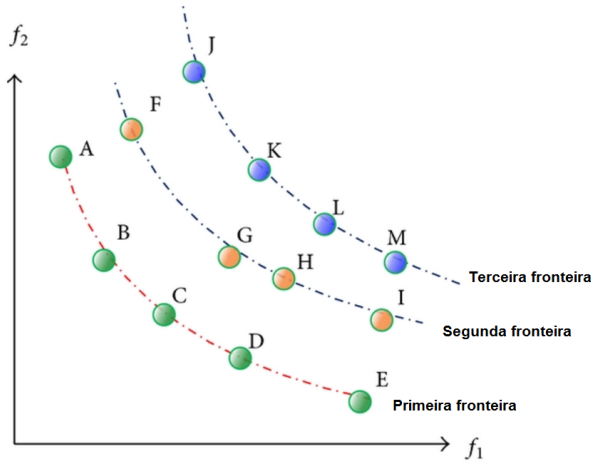


proposto por Deb et al. (2000).

A **distância de aglomeração** é utilizada para garantir diversidade do conjunto de soluções. Essa distância, para um indivíduo i, corresponde ao somatório das distâncias normalizadas entre os vizinhos $i-1$ e $i+1$, ordenados em relação a um objetivo, para todos os objetivos. É importante observar que a distância de aglomeração depende da fronteira ordenada a qual o indivíduo pertence. O algoritmo detalhado desse procedimento também pode ser encontrado no trabalho de Deb et al. (2000), que o propôs.

A partir desses conceitos, o algoritmo NSGA-II realiza a seleção de indivíduos conforme mostra a Figura 10. Primeiro, é criado uma nova população P_{t+1} vazia e começando pela primeira fronteira de P_t, é verificado se o tamanho de P_{t+1} mais o número de indivíduos da primeira fronteia excede N_p e, caso não exceda, todos os indivíduos da primeira fronteira são adicionados em P_{t+1}, sendo que o mesmo é feito com a segunda fronteira e assim por diante até que, em alguma fronteira i, a soma realizada exceda N_p. Quando isso ocorre, as

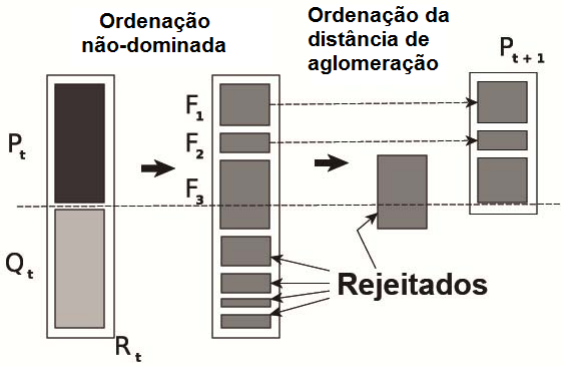
Figura 9 – Ordenação não-dominada



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

distâncias de aglomeração dos indivíduos da i-ésima fronteira são calculados e os indivíduos de maior distância são incluídos em P_{t+1} até que seu tamanho seja exatamente N_p.

Figura 10 – Seleção do NSGA-II



Fonte: Adaptado de Deb et al. (2000)

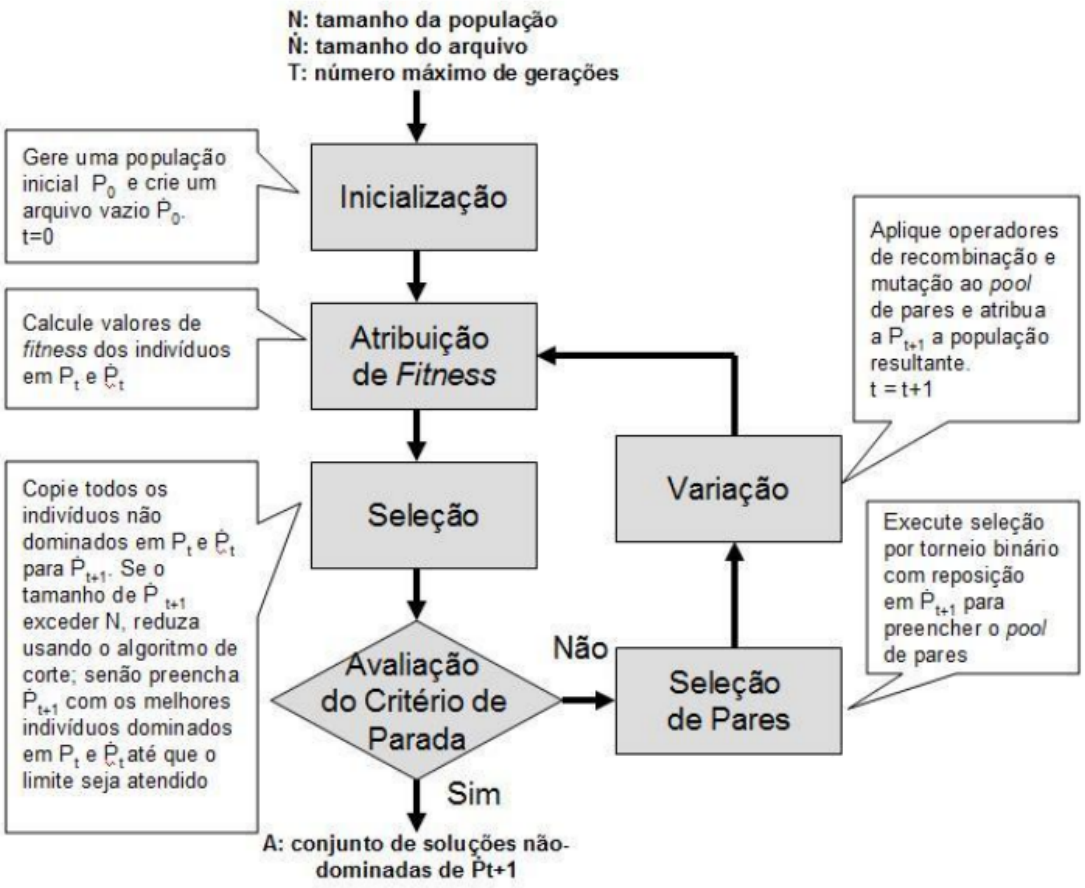
2.2.4 SPEA2 (*Strength Pareto Evolutionary Algorithm 2*)

O algoritmo SPEA2 foi proposto por Zitzler et al. (2001) e, assim como o NSGA-II, é um algoritmo genético elitista para problemas de otimização multiobjetivo. As principais diferenças entre o SPEA2 e o NSGA-II são a inclusão, no SPEA2, de um arquivo que armazena os melhores indivíduos gerados e o método de seleção que, no SPEA2, não utiliza ordenação não-dominada e distância de aglomeração, mas uma função de avaliação (*fitness*) que será apresentada posteriormente.

A Figura 11 apresenta um fluxograma do SPEA2. Nesse algoritmo, além da população de indivíduos P, de tamanho N, existe um arquivo $\overline{P}$, que armazena as $\overline{N}$ melhores

soluções obtidas.

Figura 11 – Fluxograma básico do algoritmo SPEA2



Fonte: Azuma (2011)

A **função de avaliação** ou *fitness* no SPEA2 guia as soluções ou indivíduos de forma a constituírem um conjunto com boa cobertura ou convergência ao conjunto Pareto-ótimo e boa diversidade. Para isso, são utilizados os conceitos de dominância e de densidade. Essa função é avaliada para cada indivíduo, de forma que, quanto menor seu valor, mais apto é o indivíduo. Alguns conceitos são necessários para o cálculo de tal função. Primeiramente, a força S de um indivíduo i, determinado pelo número de indivíduos dominados por ele, pode ser definida da seguinte forma:

$$S(i) = |\{j | j \in P_t \cup \overline{P_t} \wedge j \prec i\}|.$$

Outro conceito importante é o de *fitness* bruto (R) que, para um indivíduo i, corresponde ao somatório das forças de todos indivíduos que o dominam e que portanto, pode ser assim definido:

$$R(i) = \sum_{j \in P_t \cup \overline{P_t} \wedge i \prec j} S(j).$$

Para garantir a diversidade das soluções, o algoritmo utiliza o conceito de densidade:

$$D(i) = \frac{1}{dist_{ij}(k) + 2}$$

onde $dist_{ij}(k)$ é a k-ésima distância do conjunto ordenado das distâncias entre i e todas as demais soluções j de $P_t \cup \overline{P_t}$. O valor de k é um parâmetro do algoritmo, mas Zitzler et al. (2001) sugere utilizar $k = \sqrt{N + \overline{N}}$.

Por fim, a função de avaliação é definida como $F(i) = R(i) + D(i)$. Pode-se observar que o *fitness* bruto é sempre um valor inteiro enquanto que a densidade é um valor entre zero e meio, ou seja, quando dois indivíduos são comparados, a densidade é relevante apenas em caso de empate dos valores de *fitness* bruto.

Quando todos os indivíduos da população corrente e do arquivo têm sua função de avaliação calculados, o processo de seleção segue de forma que todos os indivíduos não-dominados (aqueles que possuem *fitness* bruto igual a zero) são adicionados ao novo arquivo $\overline{P_{t+1}}$. Após esse procedimento, três casos podem ocorrer:

- A quantidade de elementos adicionados é igual ao tamanho do arquivo ($\overline{N}$). Nesse caso, o processo termina.
- A quantidade de elementos adicionados é menor que $\overline{N}$. Nesse caso, o arquivo é completado com os demais indivíduos que apresentam os menores valores para a função de avaliação.
- A quantidade de elementos adicionados é maior que $\overline{N}$. Nesse caso, é utilizado um algoritmo de corte.

O algoritmo de corte exclui, iterativamente, uma solução até que a quantidade de elementos restantes (quantidade de elementos adicionais - $\overline{N}$) seja eliminada. Em cada iteração, o indivíduo a ser excluído é aquele que possui a menor distância para seu vizinho mais próximo e, em caso de empate, o que possuir a menor distância para seu segundo vizinho mais próximo e assim por diante.

2.2.5 PDEA (*Pareto-based Differential Evolution Approach*)

O algoritmo PDEA, proposto por Madavan (2002), é uma adaptação do algoritmo de evolução diferencial (ED) para problemas de otimização multiobjetivo. O algoritmo utiliza os mesmos operadores de reprodução e mutação do ED, diferenciando-se deste, principalmente, no processo de seleção de indivíduos, que no PDEA, é feito de forma

similar ao NSGA-II (utilizando ordenação não-dominada e distância de aglomeração), após a geração de novos indivíduos por meio da reprodução e mutação.

Algoritmo 2: PDEA

Entrada: f, cr
Saída: P
início
  INICIALIZA P
  $F_P \leftarrow$ AVALIA(P)
  $t \leftarrow 1$
  enquanto *critério de parada não satisfeito* **faça**
    $Q \leftarrow$ MUTAÇÃO(P,f)
    $Q \leftarrow$ REPRODUÇÃO(P,Q,cr)
    $F_Q \leftarrow$ AVALIA(Q)
    $P \leftarrow$ SELECIONA(P,Q,F_P,F_Q)
    $t \leftarrow t + 1$
  fim
fim
retorna P

O algoritmo 2 apresenta o pseudocódigo do PDEA, que possui como entrada um fator de escala f e uma probabilidade de cruzamento cr. O método inicializa atribui soluções aleatórias aos indivíduos da população, avalia calcula as funções objetivos de uma população passada por parâmetro, mutação e reprodução utilizam os mesmos operadores do ED e seleciona realiza a seleção do NSGA-II. Após o critério de parada ter sido satisfeito, a população final é retornada, finalizando o algoritmo.

2.3 Otimização de Portfólios

Antes de entrar na teoria do portfólio, é importante esclarecer alguns conceitos de finanças, tais como os **ativos**, definidos por Mansini, Ogryczak e Speranza (2015) como quaisquer instrumentos financeiros negociáveis. No mercado de capitais, um investidor, que pode ser privado ou institucional, investe uma quantidade de dinheiro denominada **capital** em um ativo ou em um conjunto de ativos, criando assim, seu **portfólio** ou **carteira** financeira. Esses ativos são vendidos em **lotes**, que são quantidades mínimas de ações que podem ser compradas de cada ativo, de forma que cada ativo só pode ser comprado ou vendido em quantidades múltiplas desse lote mínimo.

A avaliação de cada ativo no mercado de capitais pode ser avaliado de acordo com seu **preço de mercado** ou **cotação**. Para avaliar o desempenho passado de um investimento utiliza-se a taxa de **retorno** r. Se o investimento ocorreu em um tempo $t-1$, com término em t, o retorno no tempo t pode ser definido como (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$r_t = (c_t - c_{t-1})/c_{t-1}$$

onde c_{t-1} e c_t são cotações do investimento nos tempos $t-1$ e t, respectivamente. Se $r_t > 0$, significa que houve lucro no investimento e caso $r_t < 0$, significa que o investimento gerou prejuízo.

Se a cotação de um ativo esteja subindo, poderia-se esperar que seu bom desempenho continuaria no futuro e, assim, um investidor desejaria simplesmente investir todo seu capital no ativo de melhor desempenho no presente, mas a cotação de um ativo é resultado de vários fatores, como a força e estabilidade da empresa, ou sua rentabilidade, projeções futuras para seu setor, expectativa de outros investidores e do volume de negociações desse ativo, ou até mesmo de variáveis macroeconômicas, como políticas fiscais e inflações. Dessa forma, estimar cotações futuras de um ativo pode ser uma tarefa muito difícil e, consequentemente, o investimento em um único ativo se torna muito arriscado (MANSINI; OGRYCZAK; SPERANZA, 2015).

2.3.1 Modelo de Markowitz

A teoria do portfólio iniciou com o trabalho de Markowitz (1952), que foi a primeira contribuição no desenvolvimento de modelos de otimização de portfólio. O autor apresentou duas métricas de avaliação de desempenho de um portfólio, o retorno esperado e o risco, tentando prever o comportamento futuro desse portfólio. O retorno esperado expressa a ideia de que um ativo que apresenta bom desempenho no passado recente pode manter tal desempenho no futuro e, como se trata de uma previsão, o risco é a métrica proposta para modelar a incerteza do retorno.

Markowitz (1952) considera investimento em cada ativo composto por porcentagem do capital total. Ou seja, considerando n ativos disponível para investimentos, x_j, com $j \in \{1,...,n\}$, é a proporção do capital investido no ativo j. Assim, existe a restrição de que a soma dos investimentos deve ser igual ao capital disponível para investimento, ou $\sum_{j=1}^{n} x_j = 1$. Além disso, nesse modelo inicial, não era permitido operar vendido no mercado, ou seja, obter um ativo sem o capital necessário para sua compra, ou ainda $x_j \geq 0, \forall j \in \{1,...,n\}$.

Para modelar o retorno de um ativo j, Markowitz (1952) utiliza uma variável aleatória R_j, que possui média $\mu_j = E(R_j)$, onde $E(R)$ é uma função que retorna o valor esperado da variável aleatória R. Para calcular esse valor esperado para o retorno de j, deve-se conhecer seus cenários, que são as situações ou valores de cotação do ativo possíveis em um determinado tempo.

Supondo que em um determinado tempo todos os T cenários de j são conhecidos, cada cenário t, com $t = 1,...,T$, possui uma probabilidade p_t de ocorrer, de forma que

$\sum_{t=1}^{T} p_t = 1$. Em cada cenário t, o valor da variável aleatória R_j assume um valor r_{jt} e, assim, pode-se determinar o valor esperado do retorno de j: $\mu_j = \sum_{t=1}^{T} p_t r_{jt}$. Na prática, utiliza-se uma série histórica das taxas de retorno de j como seus cenários, aos quais são atribuídos probabilidades iguais. Dessa forma, o valor esperado do retorno do ativo j pode ser assim definido:

$$\mu_j = \frac{1}{T-1} \sum_{t=2}^{T} r_{jt}$$

onde r_j^t é a taxa de retorno do ativo j durante os tempos $t-1$ e t em uma série histórica formada por T cotações de j. O retorno R_X de um portfólio X, então, é a média ponderada dos retornos dos ativos que o compõe, ou seja, $R_X = \sum_{j=1}^{n} R_j x_j$ e o retorno esperado pode ser calculado como:

$$\mu(X) = E(R_X) = E(\sum_{j=1}^{n} R_j x_j) = \sum_{j=1}^{n} \mu_j x_j. \tag{2}$$

Quanto maior o retorno esperado, melhor o desempenho do portfólio, mas esse indicador apenas não considera a incerteza do retorno do portfólio, ou seja, um investidor que apenas maximiza o retorno esperado, irá investir todo seu capital no ativo que apresenta o maior desempenho no passado, desconsiderando o alto risco de tal investimento. Dessa forma, Markowitz (1952) propõe a utilização da variância do portfólio como medida de incerteza de seu retorno:

$$\sigma^2 = E\{(R - E(R))^2\}.$$

Markowitz (1952) afirma que os retornos dos ativos são muito inter-correlacionados e, dessa forma, a diversificação do portfólio pode diminuir a sua variância, mas não eliminá-la. O autor define a variância da carteira utilizando o conceito de covariância, dessa forma, fazendo com que a variância dos retornos do portfólio se torne uma função quadrática dos pesos x_j dos ativos. Tal fato pode ser observado no cálculo da covariância dos retornos de dois ativos i e j:

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

Desenvolvendo as equações propostas, Markowitz (1952) concluiu que a variância de um portfólio X pode ser definido como:

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

Por fim, o modelo clássico de Markowitz para otimização de portfólios é definido da seguinte forma (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

sujeito a:

$$\sum_{j=1}^{n} \mu_j x_j \geq \mu_0$$

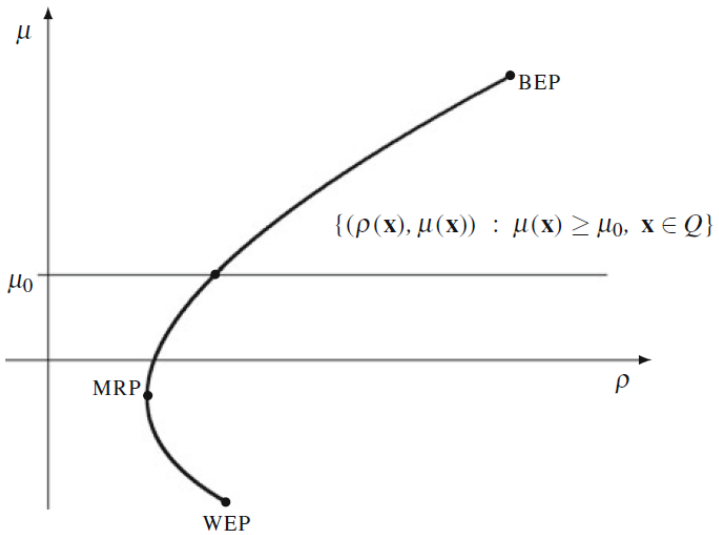
$$\sum_{j=1}^{n} x_j = 1$$

$$x_j \geq 0 \quad j = 1,...,n$$

sendo que μ_0 é um limite inferior para o retorno esperado do portfólio. Percebe-se que esse modelo é monobjetivo, quadrático e contínuo e pode, atualmente, ser solucionado por um método exato de otimização. A partir desse modelo, outros mais complexos têm sido propostos.

Em Markowitz (1959), foi proposto o conceito de fronteira eficiente, que consiste, em um problema biobjetivo de seleção de portfólios com objetivos de maximização do retorno esperado e minimização do risco, na fronteira Pareto-ótimo de portfólios para um conjunto de ativos disponíveis.

Figura 12 – Benefício da diversificação dos portfólios



Fonte: Mansini, Ogryczak e Speranza (2015)

A Figura 12 mostra uma fronteira eficiente, considerando o modelo de Markowitz com retorno esperado μ e risco ρ, em que existem apenas dois ativos disponíveis: BEP (*best expectation portfolio*) - um ativo de retorno esperado elevado, mas que também possui

alto risco e WEP (*worst expectation portfolio*) - um ativo de retorno esperado baixo, mas de baixo risco. Um conjunto de portfólios é formado, a partir do portfólio formado apenas pelo WEP, de portfólios que diminuem a proporção de WEP e aumentam gradativamente a proporção de investimento do ativo BEP. O ponto MRP (*Minimum Risk Portfolio*) é o ponto de menor risco nesse conjunto e, pode-se perceber que tal ponto não é caracterizado pelo portfólio que possui apenas o ativo de baixo risco, ou seja, a diversidade do portfólio é fundamental na minimização do risco dos portfólios. Percebe-se também que o ponto MRP domina WEP, ou seja, possui menor risco e maior retorno esperado. A fronteira eficiente, portanto, é formada pelo conjunto do ponto MRP até BEP e o parâmetro μ_0 (limite mínimo de retorno esperado) define o portfólio desse conjunto a ser escolhido no processo de otimização.

2.3.2 Medidas de Risco

Além da variância, existem outras formas de medir a dispersão da variável aleatória retorno de um portfólio, em torno de sua média. Cada forma de medida de dispersão do retorno do portfólio é também denominada medida de risco e podem substituir a função objetivo a ser minimizada do modelo de Markowitz, sendo que algumas delas possuem a vantagem de ser lineares. Mansini, Ogryczak e Speranza (2015) apresenta algumas dessas medidas como o desvio médio absoluto (MAD - *Mean Absolute Deviation*) e a diferença média de Gini (GMD - *Gini's Mean Difference*).

MAD é uma medida de dispersão que indica a média do módulo da diferença entre o valor da variável aleatória e seu valor esperado, ou seja, ela se difere da variância por utilizar o módulo do erro no lugar do quadrado do erro. GMD é outra forma de medir a dispersão de um portfólio, definida por uma função linear que mede a média dos valores absolutos das diferenças dos valores dos retornos proporcionados por cada cenário.

2.3.3 Medidas de Segurança

Do ponto de vista de investimento, variações da variável aleatória retorno acima do valor esperado são desejáveis, já que aumentam o rendimento do investidor. Dessa forma, pode ser mais interessante avaliar apenas os piores cenários, ou seja, aqueles com baixos valores de retorno.

Além das medidas de dispersão já apresentadas, denominadas de medidas de risco por Mansini, Ogryczak e Speranza (2015), existem outras medidas que avaliam somente esses piores cenários e são denominadas de medidas de segurança. Algumas dessas medidas serão apresentadas e, diferente das medidas de risco, podem ser utilizadas em um modelo de otimização de portfólio como função objetivo a ser maximizada. Essas medidas de segurança também podem ser transformadas em medidas de risco, passando a ser

chamadas de medidas de *downside risk*.

2.3.3.1 *Drawdown*

Drawdown é uma medida de segurança que indica, para uma série histórica de cotações, o valor da perda entre o máximo do investimento naquele período e o valor em um tempo t após a ocorrência desse máximo. Sua fórmula é (DRAWDOWN, 2015):

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

onde $p_t(X)$ é o preço do portfólio X no tempo t e $p_{max}(X)$ é o valor do retorno máximo do portfólio em um tempo $t' \leq t$. Seu valor oposto pode ser utilizado como uma medida de *downside risk*.

Para uma série histórica de T cotações, cada tempo $t \in \{1,...,T\}$ possui um valor de *Drawdown*. Para caracterizar toda a série, utiliza-se o *Drawdown* médio (ADD) ou o *Drawdown* máximo (MDD) dessa série, sendo que $ADD(X) = \frac{1}{T}\sum_{t=1}^{T} DD_t(X)$ e $MDD(X) = max(|DD_t(X)|)$ (DRAWDOWN, 2015). Dessa forma, essas duas medidas podem ser utilizadas em um modelo de otimização de portfólio substituindo a função objetivo da variância, proposta por Markowitz (1952). A Figura 13 mostra as cotações de um ativo durante um período de oito anos, seus respectivos *Drawdowns* e o *Drawdown* máximo dessa série. Percebe-se, portanto, que o *Drawdown* máximo indica a porcentagem da maior perda incorrida no período. É importante notar que os gráficos de preço e de *Drawdown* compartilham o mesmo eixo de tempo.

2.3.3.2 VaR

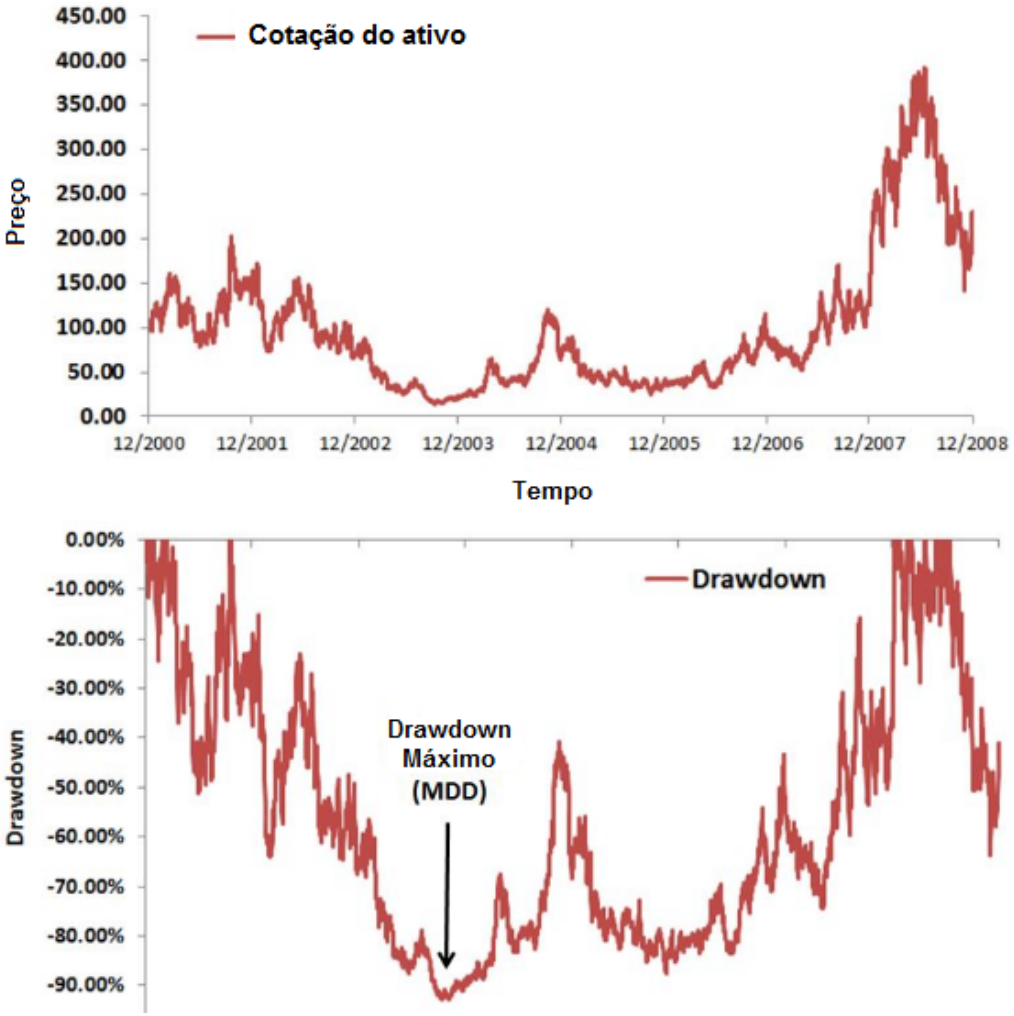
Sabendo que medidas de segurança avaliam os piores cenários de um ativo ou portfólio, é importante definir uma quantidade desses piores cenários a ser considerados. Utilizando o VaR (*Value-at-Risk*), essa quantidade é um quantil determinado por um nível de confiança $\beta \in [0,1]$. Assim o β-quantil da variável aleatória R pode ser definido por uma variável q de forma que:

$$P(R > q) \leq \beta \leq P(R \leq q)$$

onde $P(.)$ é a função probabilidade. E, assim, o valor desse quantil ou o valor da medida de segurança VaR para um portfólio X é obtido por (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$q_\beta(X) = inf\{\eta : F_X(\eta) \geq \beta\}$$

Figura 13 – Cotações e Dradowns para um ativo arbitrário



Fonte: Adaptado de Drawdown (2015)

com $F_X(\eta) = P(R_X \leq \eta)$ sendo a função de distribuição acumulada e $inf\{r : A\}$ é o menor limite de r dado um evento A. Essa medida pode ser transformada em uma medida de *downside risk* utilizando o conceito de perda no lugar da taxa de retorno, ou seja, utiliza o valor de $-R_X$ no lugar de R_X.

2.3.3.3 CVaR

Uma desvantagem da medida VaR é que não leva em consideração a cauda da distribuição, ou seja, não diferencia a distribuição dos piores retornos. Para tentar incluir tal aspecto, algumas medidas de segurança baseadas nas médias desses piores cenários vêm sendo propostas recentemente, sendo que uma dessas medidas muito utilizada é o CVaR (*Conditional Value-at-Risk*), que indica a média dos retornos dos piores cenários de um portfólio, dado um nível de confiança β. Pode-se dizer, ainda, que o CVaR é a média dos retornos abaixo do VaR do portfólio com nível de confiança igual a β.

No caso mais simples em que todos os T cenários de um portfólio X possuem a mesma probabilidade $p = 1/T$ e o nível de confiança $\beta = k/T$, a medida CVaR indica, então, a média dos retornos para os k piores cenários, definido dessa forma (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$M_\beta(X) = M_{k/T}(X) = \frac{1}{k}\sum_{s=1}^{k}\sum_{j=1}^{n} r_{js}x_j$$

em que $j = 1,...,n$ representa os índices dos n ativos que compõem o portfólio X.

Ainda de acordo com Mansini, Ogryczak e Speranza (2015), para os demais casos, o CVaR é definido pelo seguinte problema de otimização:

$$M_\beta(X) = \min_{u_t}\{\frac{1}{\beta}\sum_{t=1}^{T} y_t u_t : \sum_{t=1}^{T} u_t = \beta,\ 0 \leq u_t \leq p_t\ t = 1,...,T\},$$

onde $y_t = \sum_{j=1}^{n} r_{jt}x_j$ é a realização do cenário t e u_t, na solução ótima, vale zero, caso t não esteja entre os piores cenários e vale p_t, caso esteja. Da mesma forma que o VaR, o CVaR pode ser utilizado como uma medida de *downside risk*, utilizando perdas no lugar de retornos.

2.3.4 Modelo média-CVaR

O trabalho de Rockafellar e Uryasev (2000) foi o primeiro a utilizar a medida de risco CVaR na otimização de portfólios. O autor mostrou que minimizar o risco CVaR do portfólio X é equivalente a minimizar a função $F_\beta(X, \alpha)$, em que α é o valor do risco Var de X, com um nível de confiança β, em que:

$$F_\beta(X, \alpha) = \alpha + \frac{1}{(1-\beta)}\sum_{t=1}^{T} p_t[\sum_{j=1}^{n} -(r_{jt}x_j) - \alpha]^+$$

lembrando que a medida de risco CVaR utiliza o conceito de perda no lugar de retorno, de forma que $f(X,r_t) = \sum_{j=1}^{n} -(r_{jt}x_j)$ é a perda do portfólio X, composto por n ativos, no cenário t. Nota-se que $\sum_{j=1}^{n}[-(r_{jt}x_j) - \alpha]^+$ representa a média ponderada das perdas dos ativos de X que superaram o valor VaR no tempo t. No caso mais simples em que considera-se todos os T cenários com a mesma possibilidade $p_t = 1/T$, a função objetivo pode ser reescrita assim (ROCKAFELLAR; URYASEV, 2000):

$$F_\beta(X, \alpha) = \alpha + \frac{1}{T(1-\beta)}\sum_{t=1}^{T}[f(X,r_t) - \alpha]^+.$$

Substituindo essa função objetivo no modelo de Markowitz, obtém-se o seguinte modelo:

$$\min \alpha + \frac{1}{T(1-\beta)} \sum_{t=1}^{T} [f(X,r_t) - \alpha]^+$$

sujeito a:

$$\sum_{j=1}^{n} \mu_j x_j \geq \mu_0$$

$$\sum_{j=1}^{n} x_j = 1$$

$$x_j \geq 0 \quad j = 1,...,n$$

2.3.5 Restrição de Cardinalidade

De acordo com Mansini, Ogryczak e Speranza (2015), a teoria do portfólio induz à conclusão de que diversificar um portfólio, inserindo nesse mais ativos, é uma boa solução para a minimização de seu risco. Mas, na prática, alguns fatores como as imperfeições do mercado ou custos de transação envolvidos fazem com que exista um limite nessa diversificação de forma que, a partir desse limite, a inclusão de mais ativos diminua ganhos no investimento de tal portfólio. De fato, alguns trabalhos como o de Cesarone, Moretti e Tardella (2016) mostram que uma quantidade excessiva de ativos em um portfólio, na prática, reduz o desempenho do portfólio.

Dessa forma, pode ser desejável incorporar uma restrição de cardinalidade, ou seja, restrição para a quantidade de ativos em um portfólio. Segundo Mansini, Ogryczak e Speranza (2015), restrição de igualdade para cardinalidade só pode ser realizada utilizando variáveis binárias z_j, $j = 1,...,n$ em um problema com n ativos disponíveis, de forma que $z_j = 1$, caso o ativo j seja selecionado para o portfólio e $z_j = 0$, caso contrário. A inclusão dessa variável binária faz com que o problema se torne combinatório e, diferente do modelo contínuo, esse problema combinatório não pode ser solucionado por um algoritmo eficiente, ou com tempo polinomial de execução (MANSINI; OGRYCZAK; SPERANZA, 2015). A restrição pode ser incorporada no modelo utilizando as seguintes inequações:

$$K_{inf} \leq \sum_{j=1}^{n} z_j \leq K_{sup}$$

onde K_{inf} é a quantidade mínima de ativos que devem compor o portfólio e K_{sup} é a quantidade máxima. Em problemas de otimização de portfólio é comum utilizar um valor fixo de cardinalidade, de forma que $K_{inf} = K_{sup} = K$ (MANSINI; OGRYCZAK; SPERANZA, 2015).

Baseando-se no modelo de Markowitz (1952), Deb et al. (2011) propuseram um modelo multiobjetivo, que visa minimizar a variância e maximizar o retorno esperado, incluindo a restrição de cardinalidade como descrito a seguir:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{3}$$

$$\max_{x_1,\ldots,x_n} \quad \sum_{i=1}^{n} x_i \mu_i \tag{4}$$

$$sujeito\,a: \begin{cases} \sum_{i=1}^{n} z_i = k & \text{(5a)} \\ \sum_{i=1}^{n} x_i = 1 & \text{(5b)} \\ z_i = \begin{cases} 0, \text{ se } x_i = 0 \\ 1, \text{ caso contrário} \end{cases}, \forall i,\ i = 1,\ldots,n & \text{(5c)} \\ x_i \in [0,1],\ \forall i,\ i = 1,\ldots,n & \text{(5d)} \end{cases}$$

Nesse modelo, as expressões 5a e 5c representam a restrição de cardinalidade, modelada pela variável binária z, que indica os ativos não nulos presentes no portfólio. Assim, a adição dessa restrição resultou em um modelo não-linear misto.

2.3.6 Custos de Transação

Custos de transação são comissões e outras cobranças feitas por instituições financeiras que intermediam as operações entre o investidor e o vendedor do ativo. Esses custos podem possuir diferentes estruturas, podendo ser um valor fixo por operação, uma função proporcional do valor total dos ativos comprados ou vendidos, funções exponenciais ou logarítmicas do custo desses ativos negociados, entre outras formas (MANSINI; OGRYCZAK; SPERANZA, 2015).

Para modelar esses custos serão utilizadas variáveis c_j, com $j = 1,..,n$ representando os custos totais de cada ativo em um portfólio. O custo total do portfólio é representado por $\overline{C}$ de forma que a proporção de um ativo j no portfólio é dado por $x_j = c_j/\overline{C}$. O custo de transação associado a quantidade total do custo do ativo j no portfólio é representado por $K_j(c_j)$. Por fim, o custo de transação total para um portfólio X é dado por:

$$K(X) = \sum_{j=1}^{n} K_j(c_j).$$

$K(.)$, dependendo da estrutura do custo de transação, pode assumir diferentes funções, como função constante, linear, côncava, convexa, entre outras.

Utilizando custos de transação, duas novas restrições devem ser alteradas no modelo de Markowitz. A restrição de capital, $\sum_{j=1}^{n} x_j = 1$ deveria ser trocada por:

$$\sum_{j=1}^{n} c_j = \overline{C}$$

e a restrição de retorno $\sum_{j=1}^{n} \mu_j x_j \geq \mu_0$, trocada por:

$$\sum_{j=1}^{n} \mu_j c_j \geq \mu_0 \overline{C}.$$

Custos de transação podem ser modelados de diferentes formas, sendo que as mais utilizadas são o custo de transação fixo, proporcional, convexo linear por partes e côncavo linear por partes. Um **custo de transação fixo** é cobrado por ordem, independente do valor negociado, de acordo com a seguinte função:

$$K_j(c_j) = \begin{cases} f_j, \text{ se } c_j > 0 \\ 0, \text{ caso contrário} \end{cases}$$

onde f_j é um valor constante cobrado na compra ou venda do ativo j.

Para esse tipo de custo de transação, um modelo de otimização deve conter variáveis binárias z_j, com $j = 1,...,n$, que assume valor zero quando j é selecionado para compor o portfólio e um caso contrário (MANSINI; OGRYCZAK; SPERANZA, 2015).

Custos proporcionais são funções lineares da quantia investida de cada ativo. Dessa forma, o custo associado à quantia total investida em um ativo j é dada por (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$K_j(c_j) = k_j c_j$$

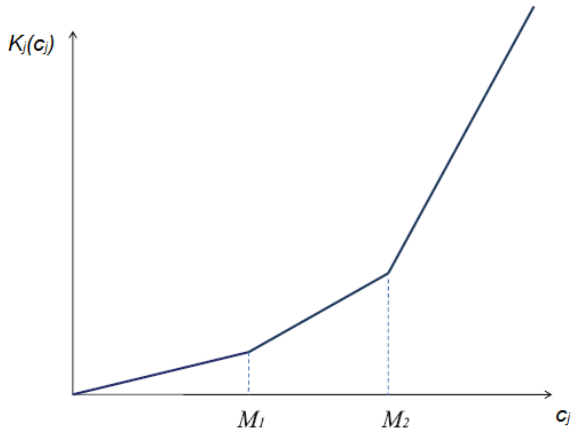
onde k_j é uma constante definida para o ativo j. Segundo Mansini, Ogryczak e Speranza (2015), esse tipo de modelagem do custo de transação é o mais utilizado, na prática.

Custo de transação convexo linear por partes é uma forma de modelar custos de transação pouco utilizada na prática, sendo definido para diferentes intervalos de quantia investida. Nesse caso, uma taxa constante k_{ji} é atribuída a cada intervalo $i \in I$, da seguinte forma (MANSINI; OGRYCZAK; SPERANZA, 2015):

$$K_j(c_j) = \begin{cases} k_{j1}c_j & \text{se } 0 \leq c_j \leq M_1 \\ k_{j2}(c_j - M_1) + k_{j1}M_1 & \text{se } M_1 \leq c_j \leq M_2 \\ k_{j3}(c_j - M_2) + k_{j1}M_1 + k_{j2}(M_2 - M_1) & \text{se } M_2 \leq c_j \leq M_3 \\ ... & \\ k_{j|I|}(c_j - M_{|I|-1}) + k_{j1}M_1 + k_{j2}(M_2 - M_1) & \\ +... + k_{j,|I-1|}(M_{|I|-1} - M_{|I|-2}) & \text{se } c_j \geq M_{|I|-1} \end{cases}$$

com $M_1, ..., M_{|I|-1}$ sendo os intervalos em que as quantias de investimento no ativo j são divididas, $|I|$ representa a quantidade total de intervalos e k_{ji} são taxas crescentes, ou seja, $k_{j1} < k_{j2} < ... < k_{j|I|}$. A Figura 14 mostra um exemplo de custo convexo linear por partes com três intervalos: $(0,M_1)$, (M_1,M_2) e $(M_2,\overline{C})$.

Figura 14 – Uma função de custo convexo linear por partes

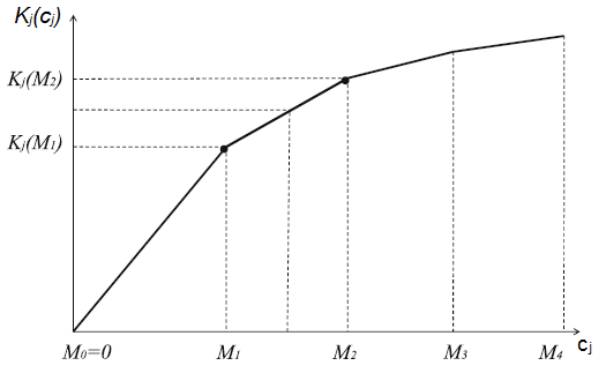


Fonte: Adaptado de Mansini, Ogryczak e Speranza (2015)

De acordo com Mansini, Ogryczak e Speranza (2015), nesse tipo de modelagem, devem ser adicionadas variáveis binárias z_{ji} que indicam se o ativo j tem sua quantidade investida c_{ji} no intervalo i diferente de zero.

O **custo côncavo linear por partes** possui mesma modelagem que o custo convexo, sendo que a única diferença dessa modelagem é que as taxas c_{ji} são decrescentes nesse caso, ou seja, $c_{j1} > c_{j2} > ... > c_{j|I|}$. A Figura 15 mostra um exemplo do custo côncavo e pode-se observar que enquanto que no custo convexo há um aumento do custo para maiores quantias investidas, no custo côncavo, esse custo diminui com o aumento dessa quantia (MANSINI; OGRYCZAK; SPERANZA, 2015).

Figura 15 – Uma função de custo côncavo linear por partes



Fonte: Adaptado de Mansini, Ogryczak e Speranza (2015)

2.3.7 Rebalanceamento

Alguns problemas surgem quando, na otimização de portfólio, considera-se que o investidor já possui um investimento prévio. Nesse caso, esse investimento ou portfólio prévio pode influenciar na decisão do investidor. Supondo, por exemplo, que esse portfólio não está apresentando um bom desempenho, ou que o investidor deseja incluir mais capital no investimento, ou ainda retirar parte do capital investido, talvez vender todos os ativos e comprar um novo portfólio não seja a melhor opção, considerando os custos de transação envolvidos. Tal problema é denominado problema de rebalanceamento de portfólios (MANSINI; OGRYCZAK; SPERANZA, 2015).

O modelo com rebalanceamento, então, deve incluir variáveis c_j^0, representando a quantia investida no ativo j nesse portfólio prévio. Sabe-se então que há uma compra do ativo j quando $c_j > c_j^0$, uma venda quando $c_j < c_j^0$ e a quantidade do ativo se mantém quando $c_j = c_j^0$.

Dessa forma, o custo de transação fixo passa a ser $f_j z_j$, onde f_j é um valor constante e z_j é uma variável binária com valor zero, quando $c_j^0 = c_j$ e um, caso contrário. E o custo proporcional, será $k_j|c_j - c_j^0|$ (MANSINI; OGRYCZAK; SPERANZA, 2015).

2.4 Programação Paralela

Programação paralela é uma técnica que permite que uma tarefa computacional seja dividida em tarefas menores que possam ser executadas ao mesmo tempo, reduzindo assim, o tempo total de execução daquela tarefa. Se a tarefa for procurar um objeto em uma coleção, a tarefa pode ser dividida em duas tarefas menores, por exemplo, sendo que a primeira tarefa deverá procurar o objeto na primeira metade da coleção e a segunda tarefa, procurar o objeto na segunda metade (WILKINSON; ALLEN, 2005).

2.4.1 Processos

De acordo com Tanenbaum (2010), um processo é uma unidade básica que pode ser entendida como um programa em execução em um sistema operacional, armazenando além do código desse programa, informações gerais sobre sua execução. Esse processo consome, além do tempo de processamento, recursos computacionais.

Atualmente, quase todos os computadores são multiprogramados, ou seja, permitem a execução de vários processos ao mesmo tempo, intercalando em fatias pequenas de tempo, o processo em execução no processador. Isso faz com que em um período maior de tempo, vários processos tenham tido trechos de código executados, criando a ilusão de paralelismo. Alguns computadores possuem mais de um processador e, nesse caso, existe também o paralelismo real, já que vários processos são executados ao mesmo tempo, cada processo em um processador (TANENBAUM, 2010).

A criação de processo em um sistema operacional acontece quando um outro processo, já em execução, solicita a criação de um novo processo por meio de chamadas do sistema. O processo já existente passa a ser denominado processo pai e o novo processo, processo filho. Dessa forma, os processos em um sistema operacional possuem uma estrutura de árvore, em que um processo pai cria alguns processos filhos que, por sua vez, criam mais alguns processos.

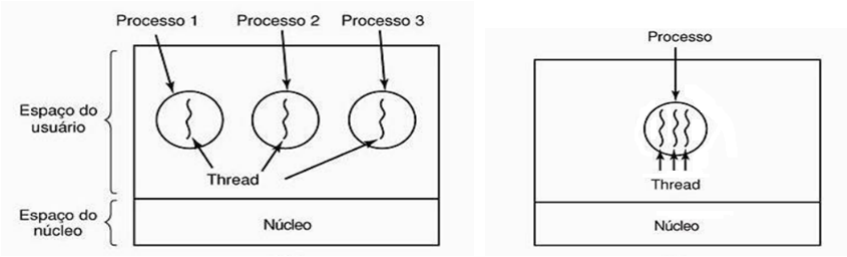
2.4.2 *Threads*

Um processo pode ser dividido em várias tarefas utilizando as *threads*, que permitem a execução concorrente ou paralela de tarefas como se fossem processos separados, mas dividindo o mesmo espaço de endereçamento. Normalmente, um processo possui apenas uma *thread* de controle, mas para algumas aplicações é desejável a utilização de várias *threads* em um mesmo processo (TANENBAUM, 2010).

Existem diversos motivos para que uma aplicação utilize múltiplas *threads*. A principal razão é que algumas aplicações possuem muitas atividades independentes que podem ser executadas ao mesmo tempo e, além disso, algumas dessas atividades podem ser eventualmente bloqueadas e, em alguns casos, criar vários processos, com espaços de endereçamento separado não funcionará. A segunda razão é a facilidade de criação e destruição de *threads* em relação aos processos, que possuem custos mais elevados para essas operações. A terceira razão é o desempenho maior que as *threads* proporcionam para aplicações com grande quantidade de operações de entrada e saída de dados. Por último, existe o fato de que as *threads* possibilitam o paralelismo real em sistemas com múltiplos processadores, reduzindo o tempo de execução da aplicação (TANENBAUM, 2010).

O modelo *monothread* é ilustrado pela Figura 16a, em que cada processo possui um único *thread* de controle e o modelo *multithread* é ilustrado pela Figura 16b, em que

um processo possui três *threads* de controle. De acordo com Tanenbaum (2010), nos dois casos existem três *threads*, sendo que os modelos diferem entre si pelo fato de que no modelo *multithread*, as *threads* compartilham o espaço de endereçamento e no modelo *monothread* o espaço de endereçamento é diferente em cada *thread*.



(a) Modelo *monothread* (b) Modelo *multithread*

Figura 16 – Diferença entre os modelos *monothread* e *multithread*

2.4.3 Arquiteturas de Processamento Paralelo

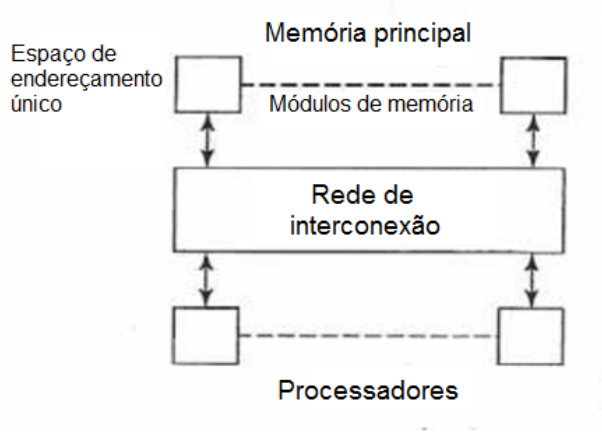
Para haver paralelismo real em uma aplicação, deve-se utilizar uma arquitetura de *hardware* que possibilite tal técnica. Os tipos de sistemas que permitem a utilização do paralelismo são divididos, principalmente, em dois grandes grupos: sistemas multiprocessados de memória compartilhada e sistemas distribuídos.

2.4.3.1 Sistemas Multiprocessados de Memória Compartilhada

Um sistema de memória compartilhada possui vários processadores e vários módulos de memória, sendo que cada processador pode acessar qualquer um dos módulos de memória por meio de um barramento ou rede de interconexão. (WILKINSON; ALLEN, 2005). A estrutura desse tipo de sistema pode ser observada na Figura 17, que mostra a ligação dos módulos de memória, formando a memória principal e sua conexão com a rede, que as ligam aos múltiplos processadores, que proporcionam o paralelismo real ao sistema.

Nesse tipo de sistema, a aplicação a ser executada deve estar na memória compartilhada e seu código fonte deve explicitar que mais de um processador deve ser utilizado ao mesmo tempo. Isso pode ser feito criando *threads* ou processos e, nesse caso, todas tarefas possuem acesso aos dados que estão na memória compartilhada, ou seja, o compartilhamento de dados acontece de forma rápida. A principal desvantagem desse tipo de sistema é a pouca escalabilidade que ele proporciona, ou seja, a dificuldade em aumentar o número de processadores ou a capacidade de memória compartilhada desse sistema (WILKINSON; ALLEN, 2005).

Figura 17 – Sistema multiprocessado de memória compartilhada típico

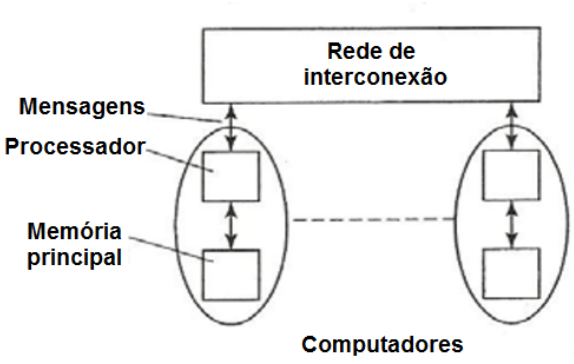


Fonte: Adaptado de Wilkinson e Allen (2005)

2.4.3.2 Sistemas Distribuídos

Na arquitetura de sistemas distribuídos, como descrito por Wilkinson e Allen (2005), existem vários computadores, denominados nós, que possuem acesso exclusivo à sua memória principal. Conectando esses nós, existe uma rede de interconexão, que possibilita que os computadores se comuniquem, enviando dados residentes em sua memória e recebendo dados residentes na memória de outros nós. Todo esse processo de comunicação entre os nós deve ser comandado pelo programa que será executado. Esse sistema é ilustrado na Figura 18, onde pode-se observar que os diversos processadores presentes no sistema possibilitam paralelismo real à aplicação.

Figura 18 – Sistema distribuído típico



Fonte: Adaptado de Wilkinson e Allen (2005)

O programa que se pretende executar nesse tipo de arquitetura deve dividir as tarefas da aplicação de forma que cada parte possa ser executado simultaneamente por um computador. O programa deve determinar também como as tarefas devem se comunicar entre os nós para o correto funcionamento da aplicação. Além de todas essas responsabili-

dades, o programador da aplicação enfrenta outras dificuldades como a complexidade do mapeamento de estruturas de dados globais, que são utilizadas em sistemas de memória compartilhada. A grande vantagem de um sistema distribuído é sua alta escalabilidade, ou seja, a facilidade de aumentar o número de nós do sistema com baixo custo. Dessa forma, sistemas distribuídos conseguem suportar enormes quantidades de processadores (WILKINSON; ALLEN, 2005).

2.4.4 Ganho de Desempenho Computacional

O principal objetivo da programação paralela é reduzir o tempo de execução de uma aplicação em relação ao tempo de execução dessa mesma aplicação em um sistema com um único processador. Assim, uma métrica muito importante em sistemas multiprocessados é o ganho, que indica quanto o sistema com múltiplos processadores é mais rápido do que um sistema monoprocessado para uma mesma aplicação. O ganho de uma aplicação na utilização de programação paralela, utilizando-se p processadores, pode ser calculado conforme a seguinte equação (WILKINSON; ALLEN, 2005):

$$S(p) = \frac{t_s}{t_p},$$

onde t_s é o tempo de execução dessa aplicação num sistema monoprocessado e t_p, o tempo de execução para um sistema multiprocessado com p processadores.

Um sistema multiprocessado possui tempo de execução de, no mínimo, o tempo gasto em uma execução sequencial, dividido pelo número de processadores. Isso ocorreria se a paralelização da aplicação não gerasse gastos computacionais e se as partes sendo executadas em cada processador fossem independentes, ou seja, não comunicassem uma com as outras. Mas na prática, a paralelização gera gastos computacionais, que aumentam o tempo de execução. Dessa forma, existe um ganho máximo, que é igual ao número de processadores, como mostrado a seguir:

$$S(p) \leq \frac{t_s}{t_s/p} \quad \text{ou} \quad S(p) \leq p.$$

Existem vários fatores que limitam o ganho em um sistema multiprocessado, como tarefas que precisam esperar algum evento e, por isso, ficam algum período ociosas, código extra para que a aplicação se torne paralelizável, comunicações entre processos e gasto computacional para criar *threads* ou processos. Esses fatores representam, na aplicação, uma parte que não pode ser paralelizada, ou seja, uma porção do programa que deve ser executada, necessariamente, em sequencia. Seja f essa parte não paralelizável de um programa, pode-se escrever a equação de ganho de forma mais completa do que

foi mostrado anteriormente, fazendo $t_p = ft_s + (1-f)t_s/p$. Assim, chega-se à seguinte equação, conhecida como lei de Amdahl (WILKINSON; ALLEN, 2005):

$$S(p) = \frac{t_s}{ft_s + (1-f)t_s/p} = \frac{p}{1+(p-1)f}.$$

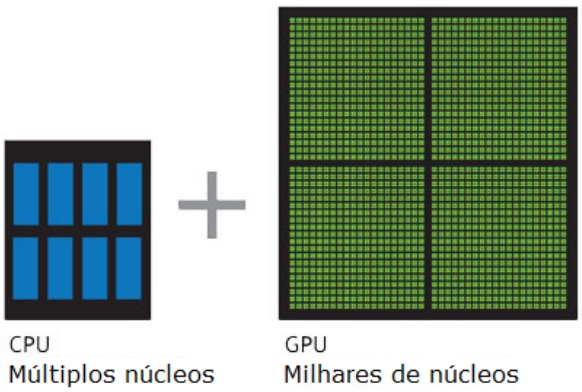
De acordo com essa lei, quanto mais se aumenta o número de processadores, mais lento é o aumento do ganho. Além disso, mesmo que fosse possível a utilização de infinitos processadores, o ganho ainda seria limitado pela porção não paralelizável f da aplicação. Isso pode ser percebido pela equação:

$$\lim_{p\to\infty} S(p) = \frac{1}{f}.$$

2.4.5 GPU (*Graphics Processing Unit*)

A GPU possui grande poder de paralelização, pois, enquanto a CPU pode possuir alguns núcleos de processamento que podem processar dados de forma paralela, a GPU possui milhares de unidades de processamento que podem executar de forma concorrente ou paralela. Para utilizar esse poder de processamento a CPU pode trabalhar junto à GPU para produzir programas de alto desempenho, enviando dados para a GPU e requisitando seu processamento sobre esses dados (NVIDIA, 2014). A Figura 19 mostra a diferença entre a arquitetura da CPU e da GPU, ressaltando o fato da GPU possuir milhares de processadores enquanto, atualmente, é comum que uma CPU possua até oito núcleos ou processadores.

Figura 19 – Arquiteturas da CPU e GPU



Fonte: Adaptado de NVIDIA (2014)

A hierarquia de memória da GPU é listada e explicada a seguir, de acordo com CUDA... (2014):

- Memória global: espaço de memória que possui maior capacidade de armazenamento, mas acesso mais lento. Variáveis nesse espaço de memória são “enxergadas” por qualquer *thread*;
- Memória compartilhada: espaço de memória armazenado no chip gráfico e que possui menor capacidade em relação à memória global, mas possui um acesso muito mais rápido. Pode ser utilizada para *threads* em um mesmo bloco;
- Memória local: espaço de memória que possui acesso mais rápido que a memória compartilhada, mas menor capacidade de armazenamento. Além disso, variáveis nessa memória podem ser utilizadas apenas pela *thread* que a criou.

2.4.6 Meta-heurísticas Paralelas

Sabendo que o paralelismo pode reduzir drasticamente o tempo de execução de algumas aplicações, meta-heurísticas podem ser muito bem aproveitadas por técnicas de paralelismo devido ao alto custo computacional desses algoritmos em alguns problemas reais aliado ao grande potencial paralelo de técnicas como algoritmos evolutivos (CASTRO, 2006).

Meta-heurísticas populacionais, em específico, podem ser paralelizadas em três diferentes níveis: nível de algoritmo, de iteração e de solução.

No paralelismo em **nível de algoritmo**, pode haver a execução paralela de diferentes heurísticas ao mesmo tempo, a mesma heurística pode ter várias execuções simultâneas para diferentes entradas, de forma que essas meta-heurísticas são executadas de forma independente e a qualidade das soluções finais devem ser equivalentes àquelas proporcionadas pela execução sequencial dessas meta-heurísticas (TALBI, 2009).

Além dessa execução independente, pode haver meta-heurísticas executando paralelamente de forma cooperativa, ou seja, trocando soluções entre elas para atingirem um melhor resultado global. Nesse caso, a qualidade das soluções pode ser afetada pelo paralelismo. Um exemplo dessa abordagem é o algoritmo evolutivo em modelo de ilhas, em que cada ilha é um algoritmo evolutivo que desenvolve (reproduz, modifica e seleciona) separadamente seus indivíduos, sendo que essas ilhas trocam seus indivíduos periodicamente (TALBI, 2009).

Paralelismo em **nível de iteração**, segundo Talbi (2009), pode ser utilizado em meta-heurísticas populacionais e de soluções únicas e, neste caso, pode-se paralelizar tanto um mecanismo de busca do algoritmo quanto a avaliação da vizinhança. Em um método de busca local, utilizando a estratégia de *best improvement*, por exemplo, a avaliação de toda a vizinhança pode ser dividida em tarefas menores, em que cada tarefa poderia avaliar parte dessa vizinhança de forma independente.

Meta-heurísticas populacionais, tais como algoritmos evolutivos podem se beneficiar

mais com esse tipo de abordagem, já que são algoritmos iterativos. Nesse caso, pode-se paralelizar tanto operadores de reprodução de indivíduos, quanto os métodos de avaliação das soluções ou, até mesmo, de seleção dos indivíduos, sendo que normalmente os métodos de avaliação são os que possuem os maiores custos computacionais (TALBI, 2009).

Paralelismo em **nível de solução** é utilizada especialmente em meta-heurísticas populacionais, que utilizam um conjunto de soluções ao mesmo tempo. No modelo sequencial, cada solução é criada e avaliada de forma uma de cada vez. A paralelização de tal processo implica no desenvolvimento e avaliação dessas várias soluções ou indivíduos de forma paralela, ou seja, ao mesmo tempo, possibilitando reduções drásticas no tempo de execução (TALBI, 2009).

3 Metodologia

A otimização de portfólios é um processo que envolve a utilização de um modelo matemático de otimização aliada a uma técnica de otimização para solucioná-lo. Dessa forma, o detalhamento do modelo e método propostos, da coleta e tratamento dos dados, além dos experimentos elaborados, serão apresentados nesse capítulo.

3.1 Modelo Proposto

Com base no trabalho de Soleimani, Golmakani e Salimi (2009), que propõe um modelo de otimização de portfólios incorporando restrições que atendem a aspectos práticos do investimento, Hanaoka (2014) propôs modelos multiobjetivos mantendo as restrições e a utilização de variáveis de decisão inteira. O modelo proposto neste trabalho parte de um desses modelos, que utiliza o CVaR como medida de risco. O modelo de Hanaoka (2014) pode ser escrito da seguinte forma:

$$\min_{x_1,\ldots,x_n} \quad \alpha + (1-\beta)^{-1} \sum_{t=1}^{T} p_t [f(x,y_t) - \alpha]^+ \tag{6}$$

$$\max_{x_1,\ldots,x_n} \quad \sum_{i=1}^{n} w_i \mu_i \tag{7}$$

$$sujeito\,a: \begin{cases} \sum_{i=1}^{n} z_i = k & \text{(8a)} \\ \sum_{i=1}^{n} m_i c_i x_i \leq C - ct & \text{(8b)} \\ z_i = \begin{cases} 0, \text{ se } x_i = 0 \\ 1, \text{ caso contrário} \end{cases}, \forall i,\ i = 1,\ldots,n & \text{(8c)} \\ x_i \in \mathbb{N},\ \forall i,\ i = 1,\ldots,n & \text{(8d)} \end{cases}$$

onde x_i é uma variável de decisão que representa o número de lotes do ativo i que foram selecionados para o portfólio e z_i é uma variável binária auxiliar que indica se o ativo i está presente nesse portfólio e w_i, outra variável auxiliar que representa a porcentagem do capital investido gasta no ativo i, de forma que:

$$w_i = \frac{m_i c_i x_i}{\sum_{i=1}^{n} m_i c_i x_i}, i = 1,\ldots,n.$$

Em relação aos parâmetros, n é o número de ativos disponíveis para investimento, k é a cardinalidade escolhida para o portfólio, C é o capital disponível para investimento, ct é o custo de transação para se obter o portfólio otimizado e considera valores de custódia, emolumentos e corretagem, m_i é a quantidade de ações que compõem um lote, c_i define o custo para a compra de uma ação do ativo i, μ_i representa a média histórica dos retornos apresentados pelo ativo i, α é o valor da medida de risco VaR aplicada ao portfólio, β é o nível de confiança para o β-quantil desejado na medida CVaR, T representa a quantidade de cenários utilizados ou o tamanho da série histórica de retornos, p_t é a probabilidade do cenário t ocorrer e $f(x,y_t)$ é a função perda do portfólio x para o cenário t, sendo que a perda pode ser considerada como o oposto do retorno.

A partir desse modelo multiobjetivo de otimização inteira não linear, o trabalho propõe outro modelo, levando em consideração outros aspectos práticos relacionados ao custo de transação. Na prática, o custo de transação pode assumir valores proporcionais ao capital investido em cada ativo e, nem sempre, é apenas um valor fixo, como no modelo de Hanaoka (2014). Além disso, o custo de transação para a obtenção de um portfólio depende dos investimentos prévios do investidor. Se um investidor possui 2 lotes de um ativo "A", por exemplo, o custo de transação que ele terá para inteirar 4 lotes desse ativo pode ser menor do que o custo que um outro investidor que não possui nenhum lote desse ativo terá. O processo de rebalanceamento, como utilizado em Beasley (2013), considera um investimento prévio para o cálculo dos custos de transação e quando incluído no modelo discreto de otimização de portfólios, dá origem ao modelo proposto apresentado a seguir:

$$\min_{x_1,\ldots,x_n} \quad \alpha + (1-\beta)^{-1} \sum_{t=1}^{T} p_t [f(x,y_t) - \alpha]^+ \tag{9}$$

$$\max_{x_1,\ldots,x_n} \quad \sum_{i=1}^{n} w_i \mu_i \tag{10}$$

$$sujeito\,a: \begin{cases} \sum_{i=1}^{n} z_i = k & \text{(11a)} \\ \sum_{i=1}^{n} m_i c_i [(x_i - x_i^{(0)}) + \gamma |x_i - x_i^{(0)}|] \leq C - Fv & \text{(11b)} \\ z_i = \begin{cases} 0, \text{ se } x_i = 0 \\ 1, \text{ caso contrário} \end{cases}, \forall i,\ i = 1,\ldots,n & \text{(11c)} \\ v = \begin{cases} 0, \text{ se } \sum_{i=1}^{n} |x_i - x_i^{(0)}| = 0 \\ 1, \text{ caso contrário} \end{cases} & \text{(11d)} \\ x_i \in \mathbb{N},\ \forall i,\ i = 1,\ldots,n & \text{(11e)} \end{cases}$$

onde $x_i^{(0)}$ representa a quantidade de lotes do ativo i presente em um portfólio prévio, γ é um valor entre zero e um que indica a proporção do capital investido em um ativo que será incorporado ao custo de transação, considerando a mesma proporção para todos os ativos. F é um valor fixo de custo de transação, cobrado em adição ao valor proporcional, para uma operação, e v é uma variável binária auxiliar que indica se deve haver alguma operação para se obter o portfólio otimizado. É possível observar que quando o portfólio prévio é igual ao otimizado, o custo proporcional $\gamma|x_i - x_i^{(0)}|$ para todos os ativos é zero, e o custo fixo Fv também é zero, já que v assumirá valor nulo.

Em relação ao modelo de Hanaoka (2014), apenas duas restrições são diferentes do modelo proposto, sendo que a primeira é representada pela inequação 11b, que adiciona, à expressão 8c, o custo proporcional $\gamma|x_i - x_i^{(0)}|$ e modifica o custo fixo Fv para modelar o processo de rebalanceamento. A restrição 11d garante que v terá valor um sempre que o portfólio prévio for diferente do otimizado e valor zero, caso contrário. Pode-se dizer que o modelo de Hanaoka (2014) é um caso particular desse modelo proposto, em que o portfólio prévio é sempre composto por nenhum ativo, ou seja, $x_i^{(0)} = 0, \forall i,\ i = 1,...,n$ e sem custo de transação proporcional ($\gamma = 0$).

Tanto a restrição de cardinalidade, quanto o rebalanceamento tornam o modelo não linear e suas soluções descontínuas, como afirmam Cheng e Gao (2015). Por isso, visando proporcionar boas soluções em tempo aceitável, na prática, foram utilizados métodos meta-heurísticos para a otimização do modelo proposto.

3.2 Algoritmos Utilizados

Para resolver o modelo proposto na seção anterior, três algoritmos evolutivos foram utilizados, sendo eles o PDEA, NSGA-II e o SPEA2. Enquanto que o SPEA2 foi utilizado por Hanaoka (2014), o NSGA-II e o SPEA2 ainda são considerados algoritmos estado da arte para problemas de otimização de portfólios com restrição de cardinalidade, além do PESA-II (*Pareto Envelope-based Selection Algorithm*) e do PAES (*Pareto Archived Evolution Strategy*), de acordo com Lwin, Qu e Kendall (2014). Foram desenvolvidos métodos de reparação específicos para garantir a factibilidade das soluções geradas. Cada solução representa um portfólio e é codificada computacionalmente em dois vetores: *vetor de ativos*, com uma cardinalidade k fixa e *vetor de lotes*, com o mesmo tamanho do vetor de ativos, onde cada elemento do vetor de ativos possui seu lote correspondente nesse vetor.

A **geração da população inicial**, em todos os três algoritmos, é feita preenchendo, aleatoriamente, o vetor de ativos e lotes de todos os indivíduos, de forma que o custo total da carteira, somado aos custos de transação, não ultrapasse o capital disponível para investimento. O Algoritmo 3 descreve tal processo, onde os parâmetros são: k, a cardinalidade definida para o portfólio, c, o capital disponível para investimento, t, a quantidade de indiví-

duos da população e D a lista contendo todos os n ativos disponíveis para investimento. A saída do algoritmo é a população inicial, representada por duas matrizes: a matriz de ativos A, que armazena a lista de ativos de todos os t indivíduos e a matriz de lotes L, que armazena a lista de lotes de todos os t indivíduos da população. A função ATUALIZA_CUSTO calcula o custo do portfólio considerando os custos de transação a partir dos ativos desse portfólio e seus respectivos lotes e, no modelo, é representado da seguinte forma:

$$custo = \sum_{i=1}^{n} m_i c_i[(x_i - x_i^{(0)}) + \gamma|x_i - x_i^{(0)}|] + Fv.$$

Algoritmo 3: GERAÇÃO DA POPULAÇÃO INICIAL

Entrada: k, c, t, D
Saída: A, L
início
  A: matriz contendo a lista de ativos de tamanho k para todos os t indivíduos
  L: matriz contendo a lista de lotes de tamanho k para todos os t indivíduos
  $custo \leftarrow 0$
  para *cada indivíduo* i *da população* **faça**
    para *cada ativo* j *do indivíduo* i **faça**
      $A_{ij} \leftarrow$ elemento aleatório de D que ainda não esteja em A_i
      $L_{ij} \leftarrow 1$
    fim
    $custo \leftarrow$ ATUALIZA_CUSTO(A_i,L_i)
  fim
  enquanto $(custo \leq c)$ **faça**
    $x \leftarrow$ ativo aleatório de i
    $L_{ix} \leftarrow L_{ix} + 1$
    $custo \leftarrow$ ATUALIZA_CUSTO(A_i,L_i)
  fim
  $L_{ix} \leftarrow L_{ix} - 1$
fim

Os três algoritmos utilizam um **critério de parada** formado por duas condições, de forma que o procedimento é finalizado quando uma dessas condições é satisfeita. A primeira condição de parada ocorre quando o número de gerações do algoritmo atinge um valor máximo G. A segunda condição de parada é satisfeita quando o indicador de hipervolume se torna menor que um limiar ϵ, normalmente próximo de zero. Esse indicador, descrito em Guerrero et al. (2010), é avaliado diversas vezes em um algoritmo evolutivo multiobjetivo e é definido da seguinte forma:

$$I_{HV}(A,B) = \begin{cases} HV(B) - HV(A), \text{ se } \forall x^2 \in B, \exists x^1 \in A : x^1 \succ x^2 \\ HV(A \cup B) - HV(A), \text{ caso contrário} \end{cases}$$

onde B é a população corrente e A corresponde à população d gerações antes de B. Quanto menor a distância d, maior seu custo computacional global, já que a avaliação desse indicador será realizada muitas vezes. Mas quanto maior o valor de d, mais gerações desnecessárias podem ser realizadas, também aumentando o tempo de execução do algoritmo. Por isso, essa distância tem que ser bem ajustada ao algoritmo, de forma a minimizar seu tempo de execução.

Apesar dos algoritmos terem sido apresentados na fundamentação teórica (seção 2.2), detalhes de sua implementação, além dos métodos de reparo propostos serão apresentados, visando a reprodutibilidade dos experimentos elaborados.

3.2.1 NSGA-II

Para o NSGA-II, foram utilizados parâmetros probabilidade de cruzamento (pc) e de mutação (pm) adaptativos. Dessa forma, essas probabilidades, ao início do algoritmo possuem valores pequenos, que aumentam ao passar das gerações de forma exponencial. A parametrização adaptativa utilizada foi proposta por Ribeiro, Barbosa e Arantes (2010), tendo sido desenvolvida especialmente para algoritmos genéticos multiobjetivos.

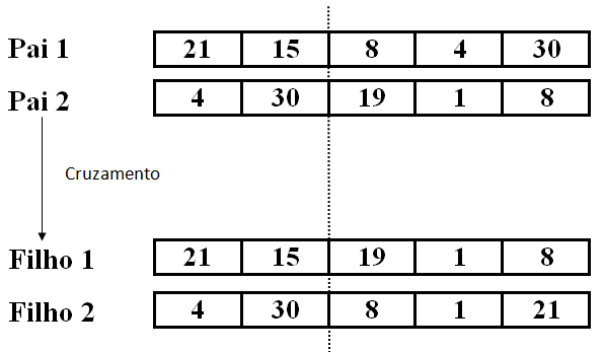
Antes da realização do cruzamento, propriamente dito, é realizada uma seleção dos indivíduos que participarão desse processo. A seleção utilizada para isso é o torneio binário, em que cada pai é definido escolhendo-se dois indivíduos aleatoriamente e, por fim, é escolhido o melhor entre eles. No NSGA-II, o melhor entre dois indivíduos é aquele que pertence à melhor fronteira não ordenada e, se eles pertencem à mesma fronteira, é melhor aquele com maior distância de aglomeração. Esse procedimento é repetido N vezes, sendo N o número de indivíduos da população e os casais de pais são formados por cada 2 pais selecionados consecutivamente. Nesse processo, um mesmo indivíduo pode ser selecionado várias vezes para se realizar um cruzamento.

Foi selecionado o **cruzamento** de ponto único (descrito na subseção 2.1.2.2), que foi modificado para evitar que um indivíduo possua o mesmo ativo em mais de uma posição de sua lista de ativos. Supondo que j seja o ponto de corte escolhido aleatoriamente em um cruzamento, a lista de ativos do primeiro filho é formado pelos ativos à esquerda da lista do primeiro pai, e o restante da lista é preenchida com os ativos à direita de j do segundo pai. Mas se algum desses ativos já estiver na lista desse filho, procura-se no primeiro pai um elemento a partir de j que ainda não esteja na lista e o insere nessa posição. O processo análogo é realizado ao segundo filho.

A Figura 20 ilustra esse processo de cruzamento em um exemplo em que a cardinalidade do portfólio é 5 e o ponto de corte escolhido aleatoriamente é aquele entre o segundo e terceiro índice do vetor de ativos, indicado pela linha tracejada. Pode-se perceber que o primeiro filho (filho 1) foi formado sem repetição de ativos e, por isso, não necessitou de um

método de reparo. O segundo filho (filho 2) teve ativos repetidos no índice 4, pois o ativo 4 já estava em sua lista na posição 1, e no índice 5, pois o ativo 30 se encontra na segunda posição de sua lista. Na posição 4, é selecionado o ativo contido nessa mesma posição do pai 2. Para a posição 5, o procedimento realizado para o índice 4 não foi possível pois o ativo 8, do pai 2, já estava na lista. Assim, procurou-se dentre os ativos do pai 1 aquele que ainda não estava na lista.

Figura 20 – Exemplo de cruzamento no NSGA-II



O próximo passo é incrementar os lotes desses filhos sem que os custos de seus portfólios excedam o capital disponível c. Portanto, o par de filhos é criado sem violar a restrição de cardinalidade. Resta, porém, avaliar se algum dos indivíduos criados viola a restrição de capital disponível. Se violar, um método de reparo do custo dos portfólios faz com que os lotes desse indivíduo sejam decrementados iterativamente e aleatoriamente até que o custo de seu portfólio, incluindo os custos de transação, não excedam c.

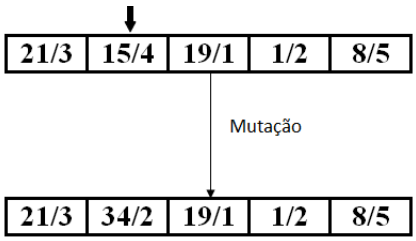
É realizado o processo de **mutação** de ponto apenas sobre as listas de ativos dos indivíduos, de forma com que alguns ativos selecionados aleatoriamente são trocados por outro ativo aleatório que não esteja em sua lista. Seu respectivo lote assume um valor aleatório entre 1 e seu valor anterior.

Utilizando o segundo filho gerado no processo de cruzamento como exemplo, a Figura 21 ilustra a mutação, em que o segundo índice do indivíduo foi selecionado aleatoriamente para sofrer mutação e, nesse caso, o ativo 15 foi substituído pelo 34, ativo escolhido aleatoriamente que não se encontra na lista de ativos. 15/4, na segunda posição, indica que ela contém o ativo 15, com 4 lotes para esse ativo e, após a mutação, a quantidade de lotes para o novo ativo (34) passa a ser um número aleatório entre 1 e 4, sendo que nesse caso o 2 foi selecionado. Nesse procedimento, soluções infactíveis podem ser geradas.

Todos os indivíduos que sofreram mutação tem seus lotes incrementados aleatoriamente sem que seus portfólios excedam c e, por fim, o método de reparo de custo dos portfólios, como aplicado no cruzamento, é também realizado nesses indivíduos que sofreram mutação.

O processo de **seleção**, utilizando os conceitos de ordenação não dominada e

Figura 21 – Exemplo de mutação no NSGA-II



distância de aglomeração, é feito conforme descrito na seção 2.2.

3.2.2 PDEA

O algoritmo PDEA utiliza, em seus procedimentos, dois parâmetros: o fator de escala f, associado ao processo de mutação, e a probabilidade de cruzamento cr, associada ao processo de recombinação. Esses parâmetros foram mantidos fixos durante a execução do algoritmo e foram escolhidos de forma empírica. A estratégia adotada foi DE/rand/1/bin (como descrita na subseção 2.1.2.3), definindo que o vetor base seja escolhido aleatoriamente, apenas 1 vetor diferencial seja utilizado e que o cruzamento binomial seja realizado. Para esse algoritmo, foram utilizados métodos de reparo baseados nos métodos propostos por Hanaoka (2014).

O processo de **mutação** foi realizado apenas sobre os vetores de ativos. Sabendo que cada elemento desse vetor é composto por um valor inteiro positivo, de 0 a $n-1$, que representa um dos ativos disponíveis para investimento, o vetor de ativos de um novo indivíduo mutante é formado utilizando-se a fórmula:

$$v_{i,g} = x_{r0,g} + f \cdot (x_{r1,g} - x_{r2,g})$$

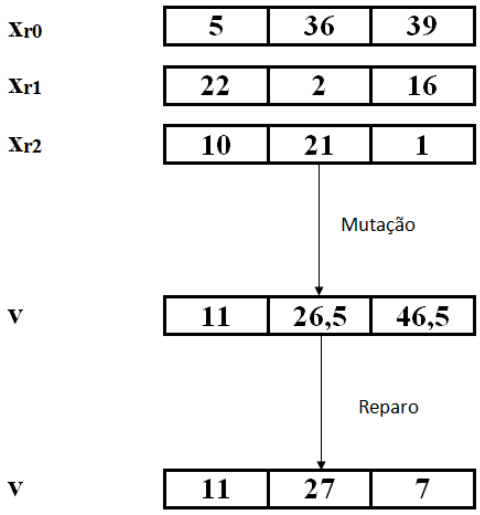
onde $r0$, $r1$ e $r2$ são índices de diferentes indivíduos da população corrente selecionados aleatoriamente.

Como a fórmula apresentada pode resultar em valores fora do intervalo desejado, existe a possibilidade de obter-se valores contínuos. O método de reparo consiste em realizar a função de teto sobre os valores resultantes para que todos os valores sejam inteiros. Posteriormente, para que todos os valores estejam no intervalo $[0,n-1]$, utiliza-se o resto da divisão desses valores inteiros por n. Garantindo que os valores sejam inteiros e estejam no intervalo desejado.

Considerando um exemplo em que o valor determinado para cardinalidade é 3, o fator de escala (f) é $0{,}5$ e a quantidade de ativos disponíveis é 40, a Figura 22 apresenta o processo de mutação para vetores base e diferencial arbitrários. Observa-se que o vetor mutante (v), gerado inicialmente, era inviável por conter valores de ponto flutuante e um

valor acima de 40. Por isso, o método de reparo elimina valores de ponto flutuante realizando a função teto, que arredonda seu valor para o número inteiro superior mais próximo (teto de $26{,}5$ é 27). Para que não haja valores acima de 40, utiliza-se o resto da divisão desses números por 40 ($47\%40 = 7$).

Figura 22 – Exemplo de mutação no PDEA



Primeiramente, o vetor de lotes dos indivíduos mutantes é preenchido com valor 1 para todos os ativos e esses lotes são incrementados aleatoriamente de forma que o custo do portfólio não exceda o capital disponível c. Esse procedimento é realizado da mesma forma como feito na geração da população inicial.

O **cruzamento**, diferente da mutação, foi realizado nos indivíduos completamente, ou seja, tanto no vetor de ativos quanto no vetor de lotes. O tipo do cruzamento realizado foi o binomial, como descrito na subseção 2.1.2.3.

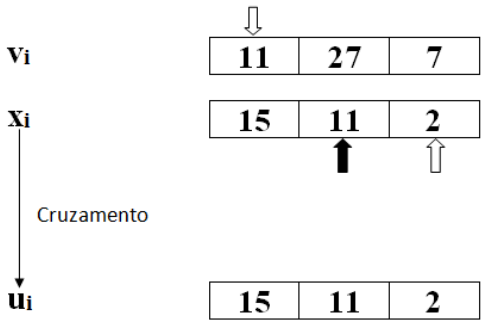
Esse operador gera um vetor experimental, combinando o vetor original i de uma geração g com seu respectivo vetor mutante e, a partir de cada elemento desses vetores, o vetor experimental é formado selecionando um desses elementos. A uma posição j do novo indivíduo experimental, pode haver duas posições na lista de ativo do novo indivíduo preenchido com o mesmo ativo, violando a restrição de cardinalidade. Quando isso ocorre, dois procedimentos podem ser aplicados:

- se a situação ocorreu ao inserir $x_{j,i,g}$, deve-se checar se $v_{j,i,g}$ já está na lista de ativos de $u_{i,g}$. Se não estiver, faz-se $u_{j,i,g} = v_{j,i,g}$. E se estiver, deve-se procurar entre $x_{1,i,g}$ e $x_{j-1,i,g}$ um ativo ainda não presente em $u_{i,g}$ e inseri-lo na posição j;
- se a situação ocorreu ao inserir $v_{j,i,g}$, deve-se checar se $x_{j,i,g}$ já está na lista de ativos de $u_{i,g}$. Se não estiver, faz-se $u_{j,i,g} = x_{j,i,g}$. E se estiver, deve-se procurar entre $v_{1,i,g}$ e $v_{j-1,i,g}$ um ativo ainda não presente em $u_{i,g}$ e inseri-lo na posição j.

Utilizando o exemplo apresentado na mutação, o cruzamento é realizado considerando o vetor mutante (v_i) resultante do processo de mutação e o vetor original x_i na mesma

posição i arbitrária. O procedimento é realizado conforme a Figura 23, sendo que, primeiramente, é realizada a seleção de um índice aleatório, indicado pela seta preta, no qual o ativo do vetor original, nessa posição, é copiado para o vetor experimental u_i. Depois disso, a partir da primeira posição, cada posição exceto a posição selecionada anteriormente, herda o ativo na posição sorteada, indicado pela seta branca. Pode-se observar que, na primeira posição, apesar do ativo 11 ser selecionado, o ativo 15, do vetor original foi herdado pelo vetor experimental, já que ele já possuía o ativo 11, herdado conforme mostra a seta preta.

Figura 23 – Exemplo de cruzamento no PDEA



Após ser gerado o vetor experimental, os lotes desse vetor são incrementados aleatoriamente de forma que o custo do portfólio não exceda o capital disponível c, como realizado na geração inicial e na mutação. Por fim, deve-se checar se o custo do portfólio supera c, tornando tal solução infactível. Caso supere, os lotes desse vetor são decrementados aleatoriamente em uma unidade até que seu custo não mais exceda o capital disponível.

A **seleção** do PDEA é feita de forma diferente da seleção do algoritmo de evolução diferencial, já que o PDEA é um algoritmo para problemas multiobjetivos. Seu método de seleção é realizado da mesma forma que a do algoritmo NSGA-II, considerando em uma geração g, a população de vetores originais x_g e a população de vetores experimentais u_g. Observa-se que essas duas populações possuem o mesmo tamanho N, que também será igual ao tamanho da população na próxima geração x_{g+1}, composta pelos melhores vetores dessas outras duas.

3.2.3 SPEA2

Os parâmetros do SPEA2 são os mesmos utilizados no NSGA-II e também é utilizado método adaptativo para determinar os valores dos parâmetros ao longo das gerações.

O **cruzamento** utilizado no SPEA2 é exatamente o mesmo utilizado no NSGA-II, cruzamento de ponto único com os métodos de reparação propostos.

Assim como o cruzamento, a **mutação** do SPEA2 também é realizada da mesma forma que no NSGA-II.

A **seleção** do SPEA2 é descrita na seção 2.2, sendo a etapa que diferencia esse algoritmo do NSGA-II, já que os outros operadores podem ser os mesmos para ambos os algoritmos, como no caso desse trabalho.

3.3 Coleta e tratamento de dados

A Bolsa de Valores brasileira oficial BM&FBOVESPA é "uma companhia que administra mercados organizados de Títulos, Valores Mobiliários e Contratos Derivativos, além de prestar serviços de registro, compensação e liquidação" (BM&FBOVESPA, 2017). O índice Bovespa ou Ibovespa é um portfólio financeiro, refeito a cada quatro meses, composto pelos ativos nacionais que são responsáveis, na média, por 80% do volume de negociação da Bolsa BMF&Bovespa nos últimos quatro meses.

Os dados utilizados no trabalho são cotações diárias do fechamento de ativos que participaram do índice Bovespa no período entre janeiro de 2010 e dezembro de 2015. Esses dados foram obtidos utilizando serviços da plataforma computacional da Bloomberg. Tal serviço fornece dados incompletos para alguns ativos desejados nesse período de 2010 a 2015 e, por isso, esses ativos não foram utilizados, restando um total de 53 ativos, os quais continham informações completas na fonte de dados utilizada. Esses ativos que foram utilizados são listados no Apêndice A.

Ajustes foram realizados às séries históricas de cada ativo utilizando o serviço da própria plataforma Bloomberg. Esses ajustes podem ser divididos em duas categorias que, de acordo com Wawrzeniak (2013), são:

- ajuste para desdobramentos e agrupamentos: quando uma ação se divide em duas (desdobramento), cada um desses novos ativos passam a ter valor igual à metade do seu valor antes do desdobramento. Quando isso ocorre, a empresa da ação que sofreu desdobramento não perde, na realidade, metade de seu valor. Dessa forma, é feito um ajuste dividindo pela metade o valor histórico dessas ações antes desse desdobramento ocorrer. O contrário ocorre em caso de agrupamentos;
- ajuste para dividendos e juros: dividendos e juros sobre capital próprio pode provocar diferenças irreais em uma série histórica de cotações, apesar de que essas diferenças não sejam tão grandes quanto no caso de desdobramentos e agrupamentos. Para remover essa diferença, são realizados três passos: o valor do dividendo é subtraído do valor do dia anterior; o resultado é dividido pelo preço do dia anterior; por fim, os preços históricos são multiplicados por este fator.

Como observa-se no modelo (seção 3.1), é utilizado o retorno dos ativos e não o preço. Dessa forma, a partir das cotações coletadas, o retorno R_t de um dado ativo, no instante t é calculado da seguinte forma:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1$$

onde P_t é o preço do ativo no instante t. Quando o intervalo entre os tempos $t-1$ e t é pequeno, e assim $R_t << 1$, sabe-se que $ln(1+R_t) \approx R_t$. Nesse caso, pode ser preferível utilizar o log-retorno devido às suas características estatísticas, tais como estacionariedade e ergodicidade, como afirmado em Portal... (2009). No tempo t o log-retorno é definido como (PORTAL..., 2009):

$$r_t = ln(1+R_t) = ln\left(\frac{P_t}{P_{t-1}}\right) = ln(P_t) - ln(P_{t-1}).$$

Outros tratamentos como normalização dos dados e remoção dos *outliers* não foram aplicados às séries históricas de retorno, já que vários trabalhos, como o de Oliveira (2016), sugerem que o desempenho da otimização de portfólios utilizando a medida CVaR não depende da normalização desses dados. Além disso, levando em consideração que a medida CVaR avalia exatamente as maiores perdas, remover os *outliers* poderia causar grandes diferenças indesejadas no valor de risco dos portfólios.

3.4 Experimentos

Uma série de experimentos foi realizada com o intuito de verificar a eficácia do modelo proposto em relação a alguns modelos presentes na literatura e outros modelos que surgem a partir da relaxação de algumas de suas restrições. Os algoritmos desenvolvidos também foram comparados de acordo com as métricas de desempenho adotadas. Outras comparações envolvem a utilização de diferentes parâmetros do modelo e dos algoritmos. Tais experimentos têm o objetivo de fundamentar possíveis escolhas de modelo, algoritmo e parâmetros, dentre os métodos pertencentes ao escopo desse trabalho, que garanta maiores lucros, na média, a um investidor que os utilize.

Cada um desses experimentos será apresentado nessa seção. Eles são compostos por testes, divididos em testes *in-sample*, aqueles que utilizam apenas dados para alimentar o modelo proposto, e testes *out-of-sample*, aqueles que utilizam dados posteriores àqueles utilizados para alimentar o modelo e que, assim, simulam a eficácia desse modelo em casos realísticos de investimento no mercado de capitais. Os testes *in-sample* que utilizam dados diários, trabalham com as cotações durante o ano de 2010 e testes *out-of-sample* trabalham com dados no período entre 2011 e 2015.

Nos testes *in-sample*, são avaliados os resultados dos algoritmos, utilizando três métricas de desempenho: métrica C (cobertura de dois conjuntos), médio (dado sempre em porcentagem), métrica Δ de espaçamento das soluções, e métrica HV (hipervolume), todos

descritos na subseção 2.2.2.1. Além dessas métricas, o tempo de execução também foi avaliado em alguns experimentos. Para facilitar, as métricas serão chamadas, respectivamente, de ACM (*Average C Metric*), DM (*Delta Metric*) e HV (hypervolume).

Nos testes *out-of-sample*, são avaliados os retornos mensais proporcionados pelos modelos utilizados, seus respectivos riscos CVaR e *Drawdown* mensais, além da evolução do retorno acumulado durante o período do teste. Esses testes não utilizam log-retorno, mas o retorno como definido na seção 2.3.

Resumidamente, o experimento 1 compara os algoritmos descritos na seção 3.2 e testa diferentes valores de cardinalidades para portfólios (parâmetro k) e determina a melhor combinação desses para utilizar nos próximos testes. O experimento 2 compara diferentes modelos com o modelo proposto e, ao fim desse experimento, é determinado o melhor modelo para o problema de otimização restrita de portfólios estudado para ser utilizado nos próximos testes. O experimento 3 realiza simulações para determinar um melhor período para a série histórica utilizada no processo de otimização e, novamente, esse parâmetro é fixado nos testes posteriores. Esse experimento também realiza simulações comparando o modelo com rebalanceamento e o modelo sem tal processo, determinando se ele deve ser utilizado nos testes posteriores. Por fim, o experimento 4 realiza melhorias aos algoritmos, paralelizando-os para uma posterior realização de simulações com frequência de transações maior que as anteriores. Essas conexões entre experimentos é ilustrada na Figura 24.

3.4.1 Primeiro Experimento

No primeiro experimento são utilizados retornos diários dos ativos selecionados durante o ano de 2010. Esses dados são, portanto, utilizados pelos três algoritmos (PDEA, NSGA-II e SPEA2) com o propósito de otimizar o modelo proposto. Nesse experimento, todos os parâmetros foram fixados, exceto a cardinalidade (k), que assumiu valores 3, 9 e 15, seguindo o trabalho de Cheng e Gao (2015), que afirma que os portfólios de maiores desempenho possuem não mais que 15 ativos. Essa variação tem objetivo apenas de encontrar um bom valor de cardinalidade para a série histórica estudada, sendo que o melhor valor para uma dada série pode ser encontrado executando paralelamente otimizações com diferentes cardinalidades. Os demais parâmetros foram fixados com base em trabalhos relacionados analisados. O período da série histórica de um ano, por exemplo, foi utilizado no trabalho de Barroso (2017).

Nesse experimento, desejou-se saber, por meio de teste *in-sample*, qual o algoritmo e qual valor de cardinalidade do portfólio apresentam melhores valores para cada uma das métricas adotadas, para que testes futuros possam utilizar uma cardinalidade e meta-heurística fixa. A Figura 25 apresenta as possíveis combinações dos algoritmos e valor estipulado de cardinalidade.

Figura 24 – Fluxograma dos experimentos

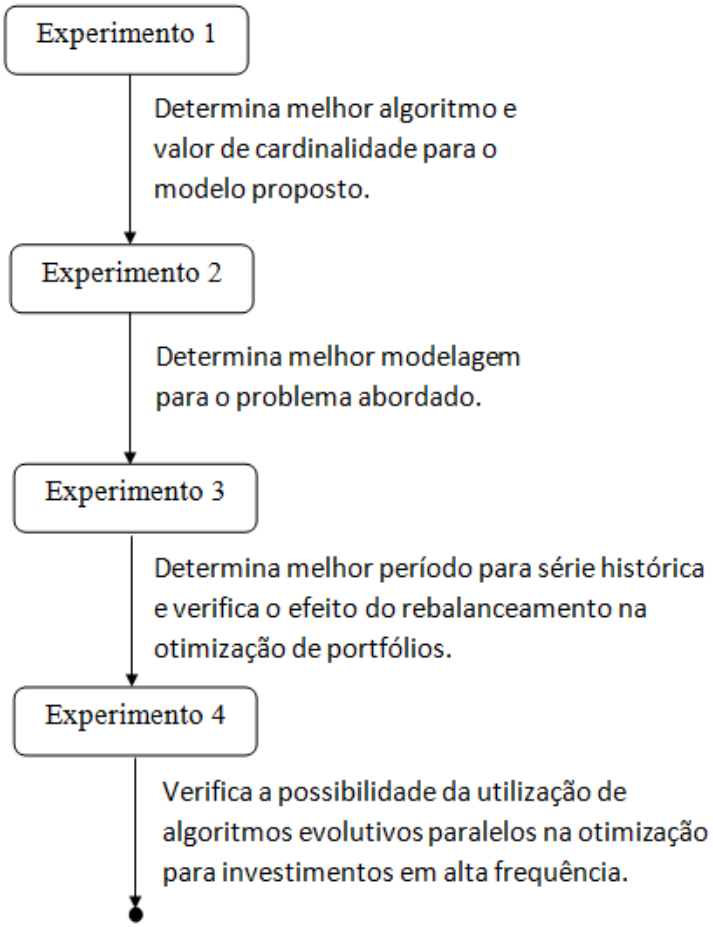
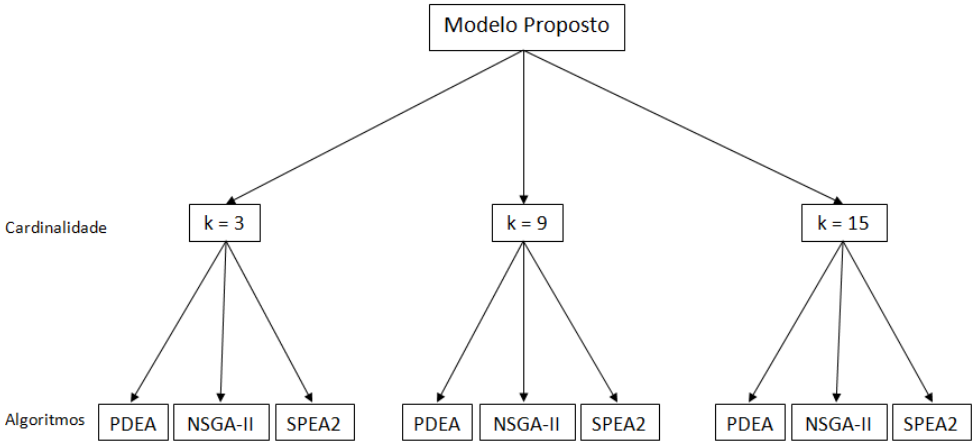


Figura 25 – Fluxograma do primeiro experimento



3.4.2 Segundo Experimento

Com testes *in-sample*, comparou-se o algoritmo multiobjetivo com melhor desempenho no teste anterior com sua versão monobjetiva, que possui os mesmos operadores e parâmetros que o multiobjetivo e que utiliza, como função objetivo, a razão entre retorno e risco de um portfólio. O objetivo desse experimento é determinar se o algoritmo monobjetivo

apresenta portfólio com índice de desempenho melhor que o melhor índice apresentado pelos portfólios dos algoritmos multiobjetivos, ou seja, se um algoritmo monobjetivo simples apresenta um desempenho melhor nessa tarefa de encontrar um único portfólio de maior índice desempenho, o que invalidaria a utilização do algoritmo multiobjetivo, com maior complexidade computacional, para esse fim.

Nesse trabalho, o índice de desempenho foi o critério utilizado para a seleção de um portfólio dentre aqueles pertencentes à fronteira eficiente gerada por algum algoritmo. Esse índice também é utilizado como critério de seleção de portfólios para algoritmos multiobjetivos em alguns trabalhos relacionados, como os de Hanaoka (2014) e Barroso (2017).

Outro teste *in-sample* comparou resultados proporcionados pela meta-heurística NSGA-II considerando o modelo proposto, conforme descrito na subseção 3.2.1 (que será denominado NSGA-II 1), com aqueles proporcionados pelo NSGA-II 2, desenvolvido por Deb et al. (2011) considerando um modelo de otimização contínuo, que é mais aceito na literatura que o modelo inteiro, conforme apresentado na subseção 2.3.5, exceto pela utilização do risco CVaR no lugar da variância.

Esse modelo contínuo pode ser comparado ao modelo discreto proposto, quando não há um investimento prévio ($x_i^{(0)} = 0, \forall i, \; i = 1,...,n$) e não há custos de transação. Tais modificações foram feitas no modelo proposto, que foi comparado com o modelo contínuo. Para isso, foi utilizado o NSGA-II também proposto por Deb et al. (2011) (chamado nesse trabalho de NSGA-II 2).

Como já mencionado, ativos são comprados em quantidades mínimas denominadas lotes e as soluções do modelo contínuo podem não corresponder a quantias múltiplas dos lotes para cada ativo. Dessa forma, um outro procedimento (denominado aqui de NSGA-II 3 por motivo de simplificação, mesmo utilizando o mesmo algoritmo de otimização que o NSGA-II 2) utiliza as soluções do NSGA-II 2 e, dividindo os valores selecionados de cada ativo pelo seu lote mínimo, foi determinado o valor de lotes que deveriam ser investidos nesse ativo, realizando sobre essa razão, a operação piso.

Dessa forma, além da comparação entre NSGA-II 1 (do modelo discreto proposto) e NSGA-II 2 (para o modelo contínuo), houve a comparação do NSGA-II 3. Deseja-se saber, portanto, a partir dos algoritmos citados, qual o melhor modelo para o problema de otimização de portfólios com restrição de cardinalidade, como definido neste trabalho, em um investimento real. Além disso, o melhor entre os três algoritmos será utilizado em testes posteriores.

Finalmente, em outro teste *in-sample*, algumas soluções obtidas por métodos exatos, que utilizam os algoritmos simplex e *branch-and-cut* (executados pelo *solver* IBM ILOG CPLEX) são comparadas, em um teste *in-sample*, com soluções obtidas pelo algoritmo

evolutivo que demonstrou melhor desempenho nos testes anteriores, desejando avaliar desempenho e tempo de execução para cada um deles. Para a aplicação do método exato, um modelo linear monobjetivo, a partir do modelo mostrado na seção 3.1, foi proposto:

$$\min_{a,\alpha,x_1,\ldots,x_n} \quad \alpha + (1-\beta)^{-1} \sum_{t=1}^{T} p_t a_t \tag{12}$$

$$sujeito\,a: \begin{cases} \sum_{i=1}^{n} w_i \mu_i \geq \mu_0 & \text{(13a)} \\ a_t \geq 0,\ \forall t,\ t = 1,\ldots,T & \text{(13b)} \\ a_t \geq f(x, y_t) - \alpha,\ \forall t,\ t = 1,\ldots,T & \text{(13c)} \\ f(x, y_t) = -\sum_{j=1}^{n} w_j r_{jt},\ \forall t,\ t = 1,\ldots,T & \text{(13d)} \\ \sum_{i=1}^{n} z_i = k & \text{(13e)} \\ z_i \leq x_i,\ \forall i,\ i = 1,\ldots,n & \text{(13f)} \\ z_i \geq x_i(\frac{P_{min}}{C}),\ \forall i,\ i = 1,\ldots,n & \text{(13g)} \\ z_i \in \{0,1\},\ \forall i,\ i = 1,\ldots,n & \text{(13h)} \\ w_i = \frac{m_i c_i x_i}{C - F},\ \forall i,\ i = 1,\ldots,n & \text{(13i)} \\ \sum_{i=1}^{n} m_i c_i [(x_i - x_i^{(0)}) + \gamma |x_i - x_i^{(0)}|] \leq C - F & \text{(13j)} \\ \sum_{i=1}^{n} m_i c_i [(x_i - x_i^{(0)}) + \gamma |x_i - x_i^{(0)}|] \geq m_i P_{max} & \text{(13k)} \\ x_i \in \mathbb{N},\ \forall i,\ i = 1,\ldots,n & \text{(13l)} \end{cases}$$

onde a_i são variáveis auxiliares para linearização da medida CVaR, μ_0 é o retorno esperado mínimo, r_{jt} representa o retorno do ativo j no cenário t, P_{min} é o preço mínimo de todos os ativos em todos os cenários e P_{max}, o preço máximo desses ativos.

Pode-se perceber, primeiramente, que o modelo mantém o objetivo de minimizar o risco CVaR, cuja medida foi substituída por funções lineares, adicionando as restrições 13b, 13c e 13d (como demonstrado por Mansini, Ogryczak e Speranza (2015)), mas para tornar o modelo monobjetivo, o retorno mínimo foi utilizado como restrição. Assim, variando-se o retorno mínimo, pode-se obter vários portfólios na fronteira eficiente. Restrições 13e, 13f, 13g e 13h representam, em funções lineares, a restrição de cardinalidade. A restrição 13i é uma função linear aproximada da função de proporção de um ativo i do modelo proposto ($w_i = m_i c_i x_i / (\sum_{j=1}^{n} m_j c_j x_j)$), sendo que a restrição 13k faz com que os custos das carteiras sejam próximas ao capital disponível, aproximando o máximo possível a definição de proporção de um ativo de ambos os modelos.

Para a otimização desse modelo utilizando o método exato, o retorno mínimo μ_0 é definido aleatoriamente em um intervalo determinado pela fronteira eficiente do método heurístico. Esse procedimento é repetido 10 vezes, gerando 10 portfólios não-dominados. Cada uma dessas soluções será comparada a duas soluções da meta-heurística, a solução com retorno imediatamente inferior e aquela com retorno imediatamente superior ao retorno da solução do método exato, já que esse retorno dificilmente será exatamente igual ao retorno de alguma solução gerada pela meta-heurística.

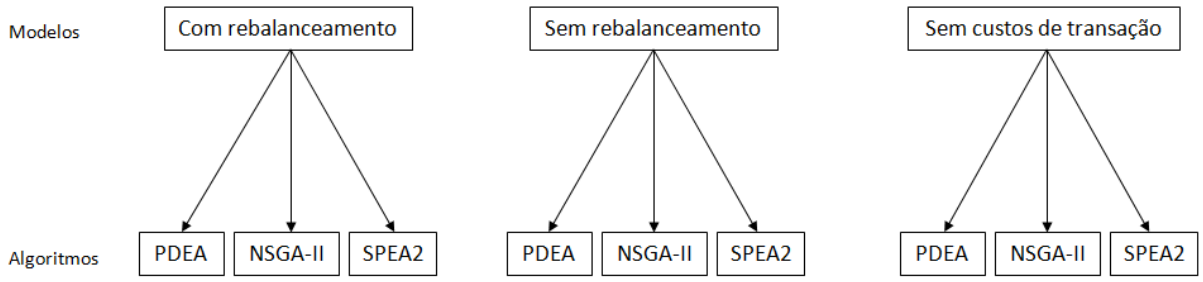
3.4.3 Terceiro Experimento

Apesar dos testes mencionados anteriormente utilizarem série histórica de preço dos ativos de um ano, um teste *out-of-sample* é realizado para tentar ajustar esse período de forma que o modelo proporcione melhores ganhos financeiros ao usuário, então varia-se esse período em valores iguais a 6 meses, 1 ano, 1 ano e 6 meses, e 2 anos. Nessa simulação, sempre seleciona-se o portfólio de maior índice desempenho dentre aqueles pertencentes à fronteira eficiente, re-selecionando-o a cada mês, para simplificação da modelagem da custódia, parte do custo de transação cobrado mensalmente. Essa simulação teve uma duração de cinco anos, iniciando em janeiro de 2011 e finalizando em dezembro de 2015.

O modelo proposto, com rebalanceamento, foi comparado com o modelo de Hanaoka (2014), sem rebalanceamento, em um teste *out-of-sample*, objetivando determinar o modelo que proporciona melhores ganhos em uma simulação de investimentos financeiros. Um modelo sem custos de transação ($\gamma = F = 0$) também foi comparado aos outros dois, na simulação considerando o período entre 2011 e 2015.

Nesse teste, todos os algoritmos foram utilizados de forma que, além de saber qual o melhor modelo, desejou-se também analisar os algoritmos em relação ao ganho que eles proporcionam em uma simulação realística. Além disso, a comparação dos algoritmos possui a finalidade de verificar a robustez dos resultados obtidos para os modelos de otimização. A Figura 26 mostra todas as combinações de modelos e algoritmos nesse teste.

Figura 26 – Fluxograma do sexto experimento



Por fim, outra simulação é realizada considerando diferentes cardinalidades com o objetivo de determinar se o valor de cardinalidade que apresentou melhor resultados *in-sample* também produz melhores ganhos no teste *out-of-sample*.

3.4.4 Quarto Experimento

De acordo com os testes realizados, foi escolhido um dos três algoritmos, que foi, neste teste, paralelizado de duas formas diferentes, sendo que os dois tipos de paralelismos são realizados a nível de algoritmo. A primeira forma de paralelismo utiliza um sistema *multithread*, enquanto que a segunda utiliza uma combinação de processamento em CPU e GPU, aproveitando a arquitetura altamente paralelizável de uma placa de vídeo.

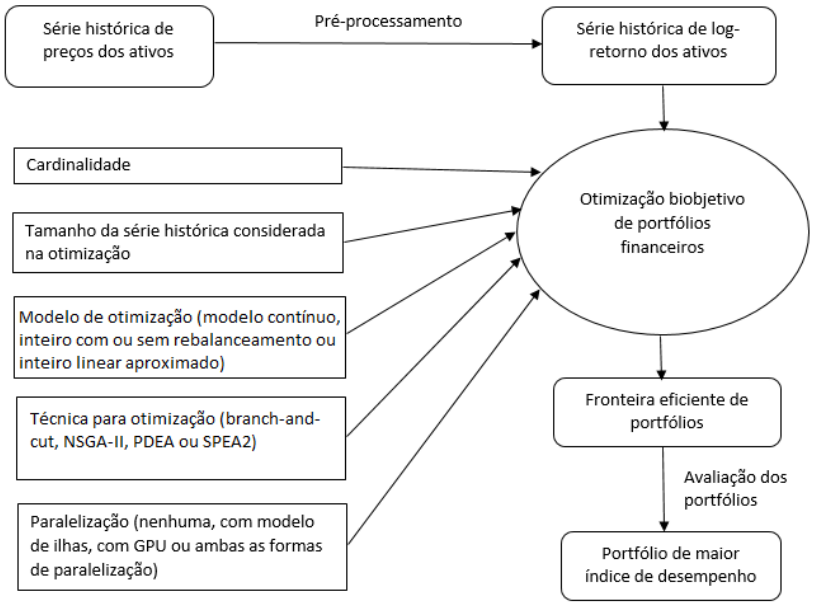
- A primeira forma utiliza um modelo de ilha na paralelização do algoritmo. Nesse modelo, os indivíduos desse algoritmo são divididos em várias ilhas, que são algoritmos evolutivos independentes com um número menor de indivíduos, que trocam soluções entre si a cada quantidade estipulada de gerações. Foi utilizada uma *thread* para cada ilha, garantindo a paralelização do algoritmo evolutivo e, para cada ilha, foi utilizado um número de indivíduos N_i igual ao número de indivíduos N do algoritmo sequencial dividido pelo número total de ilhas ou *threads*. Foram realizados experimentos utilizando números de *threads* nt iguais a 2 e 4.
- A segunda forma utiliza a GPU com a CPU para paralelizar apenas os procedimentos de ordenação do algoritmo, uma vez que ele possui alto custo computacional. A CPU delega, portanto, a ordenação para a GPU, utilizando a plataforma CUDA e a biblioteca *Thrust*, cuja documentação pode ser encontrada em NVIDIA (2017).

Teste *in-sample* é realizado, então, considerando o algoritmo sem paralelismo (sequencial), as duas configurações da primeira forma de paralelismo ($nt = 2$ e $nt = 4$) e o algoritmo que utiliza a segunda forma. Nesse teste avaliou-se as métricas de desempenho para cada algoritmo e, principalmente, o tempo de execução apresentado por cada um, avaliando se de fato a paralelização da meta-heurística pode reduzir seu tempo de execução sem piorar a qualidade das soluções geradas, possibilitando investimentos de curto prazo.

3.5 Arcabouço Computacional

Para a realização dos experimentos, foi desenvolvido um arcabouço computacional para a realização da otimização de portfólios de acordo com entradas definidas pelo usuário e as séries históricas de preços dos ativos que serviram de dados para o trabalho, como já mencionados. O arcabouço possui a estrutura ilustrada pela Figura 27, que apresenta os parâmetros de entrada representados pelos quadrados e dados coletados e gerados representados pelos quadros.

Figura 27 – Estrutura do arcabouço computacional



Os parâmetros de entrada são, portanto, a cardinalidade de ativos, o tamanho da série histórica que será considerado na execução *in-sample*, modelo de otimização, que pode ser contínuo, discreto sem rebalanceamento como proposto por Hanaoka (2014) ou discreto com rebalanceamento como proposto neste trabalho, a técnica para solução do modelo de otimização, podendo ser exata ou meta-heurística, considerando os algoritmos apresentados no trabalho e, por fim, a paralelização ou não do algoritmo, podendo-se utilizar as formas de paralelização descritas no quarto experimento.

Definidos os parâmetros, o arcabouço gera uma série de log-retorno dos ativos considerados a partir de suas séries de preços. Essa série de log-retorno é utilizada pela técnica de otimização definida para a geração da fronteira eficiente, composta por portfólios não dominados entre si em relação aos critérios de retorno esperado e risco. Testes *in-sample* analisam essas fronteiras eficientes geradas para diferentes parâmetros, enquanto que testes *out-of-sample* utilizam um único portfólio, que corresponde àquele com maior índice de desempenho (valor da razão do retorno pelo risco), em simulações de operações mensais, avaliando ganhos proporcionados por diferentes configurações dos parâmetros de entrada.

4 Análise e Discussão dos Resultados

Este capítulo apresenta e discute os resultados dos experimentos planejados, para uma configuração específica de parâmetros para modelo e algoritmos propostos, obtidos de forma empírica ou com base em trabalhos relacionados.

Os parâmetros do modelo escolhidos foram: número de ativos n igual a 53; nível de significância α de 5%; número mínimo de ações $m_i, \forall i, i = 1,...,N$ igual a 100; capital disponível C de $R\$\,100.000{,}00$; custo de transação proporcional β de $0{,}45\%$; custo de transação fixo F de $R\$\,29{,}00$.

O nível de significância e o número de ações que compõem um lote foram selecionados de acordo com trabalhos relacionados, como os de Hanaoka (2014) e Barroso (2017). Capital disponível foi escolhido de forma que uma carteira pudesse conter pelo menos um lote dos ativos mais caros. Por fim, os custos de transação, fixo e proporcional, derivam dos custos de emolumentos, corretagem e custódia, com valores retirados de Banco... (2017). Nos testes *in-sample*, foi utilizada quantidade de lotes iniciais 0 para todos os ativos ($x_i^{(0)} = 0,\ \forall\, i,\ i = 1,...,n$).

Em relação aos algoritmos, os parâmetros escolhidos foram: quantidade máxima de gerações G igual a 500; tamanho da população N_p igual a 500; para o SPEA2, o tamanho do arquivo N_a também é de 500; para o PDEA, a probabilidade de cruzamento (cr) é $0{,}3$ e o fator de mutação (f) de $0{,}5$. Os demais parâmetros são alterados de forma adaptativa, como proposto em Ribeiro, Barbosa e Arantes (2010).

Para o número máximo de gerações, foi escolhido um número relativamente grande, já que foi utilizado outro critério de parada (indicador de hipervolume) em todos os algoritmos. O número da população e do arquivo do SPEA2 foram os maiores possíveis sem que o programa sofra sobrecarga de memória. O PDEA não utiliza parâmetros adaptativos, pois o método utilizado é destinado aos algoritmos genéticos multiobjetivos, apenas. Portanto, seus parâmetros foram determinados empiricamente, sendo selecionados aqueles que apresentaram melhores resultados em testes prévios.

Cada procedimento estocástico, como a execução de um dos algoritmos para o modelo proposto, foi realizada 30 vezes. Os resultados foram analisados por meio de testes estatísticos de análise de variância ANOVA, que necessita de resíduos com as seguintes características: independência, normalidade e homocedasticidade. Foram realizados testes de Durbin-Watson, Shapiro-Wilk e Fligner-Killeen para verificar a independência, normalidade e homecedasticidade, respectivamente. Foram rejeitadas as hipóteses de normalidade e homocedasticidade, sendo que apenas a hipótese de independência não foi rejeitada. Como os dados são balanceados e o tamanho da amostra é grande o suficiente (maior que

10), Bathke (2004) afirma que, nesse caso, o teste ANOVA é robusto para as pequenas violações apresentadas de normalidade e homocedasticidade.

Todos os testes foram realizados em um computador cujo processador é um Intel® Xeon® E5-2630 v3, núcleo i7 (4 núcleos) e frequência de $2{,}4$ GHz. A memória RAM é de 8 GB. Unidade de processamento gráfico AST2500 Advanced PCIe Graphics. Foi utilizado o sistema operacional Linux Mint 18.1, linguagem de programação C e compilador GCC versão 6.1.

4.1 Primeiro Experimento

Este experimento consiste em um teste *in-sample* que compara algoritmos e parâmetros de cardinalidade. Assim como os outros testes *in-sample* que utilizam dados diários, o experimento utiliza os retornos no período entre janeiro e dezembro de 2010. Cada algoritmo (PDEA, SPEA2 e NSGA-II propostos) foi executado 30 vezes e os gráficos gerados apresentam fronteiras não-dominadas combinadas das soluções geradas em todas as execuções, ou seja, se em cada execução 500 soluções são geradas, a fronteira combinada é formada a partir de 15000 (30×500) soluções, sendo que as dominadas não pertencerão à fronteira.

Primeiramente, comparou-se os algoritmos para as diferentes cardinalidades estipuladas (3, 9 e 15). A Figura 28 apresenta as soluções proporcionadas pelos três algoritmos quando a cardinalidade é restrita ao valor 3, a Figura 29 apresenta soluções encontradas pelos algoritmos para cardinalidade 9 e, a Figura 30, para cardinalidade 15.

Observa-se que, para cardinalidade 3, todos os algoritmos apresentam soluções próximas, que sobrepõem-se no gráfico e, conforme aumenta-se a cardinalidade do portfólio, pode-se perceber que a fronteira gerada pelas soluções do NSGA-II é a melhor, apresentando soluções mais próximas da fronteira Pareto-ótimo e melhor espaçadas. O SPEA-II aparenta gerar a segunda melhor fronteira e o PDEA, a pior. A sobreposição para cardinalidade 3 deve-se, supostamente, ao fato de que, nesse caso, menos combinações são possíveis e, assim, todos os algoritmos conseguem alcançar bons resultados, mas o aumento do número de ativos no portfólio aumenta a dificuldade do problema, aumentando o número de combinações de ativos possíveis nesse portfólio e, nesse caso, algoritmos mais adequados à solução desse problema se destacam mais dos demais.

Em relação à média da métrica C (ACM), que mede quão próximo uma fronteira está da fronteira ótima em relação às outras fronteiras, percebe-se que para cardinalidades iguais a 9 e 15, o desempenho do NSGA-II é melhor que do SPEA2 que, por sua vez, supera o PDEA. Mas, para cardinalidade 3, observa-se uma sobreposição de soluções na Figura 28 e, por isso, um teste estatístico ANOVA é realizado para esse caso. A Figura 31a apresenta o *boxplot* de ACM para cada algoritmo. Como suas caixas não se sobrepõem e

Figura 28 – Fronteiras não-dominadas para cardinalidade 3

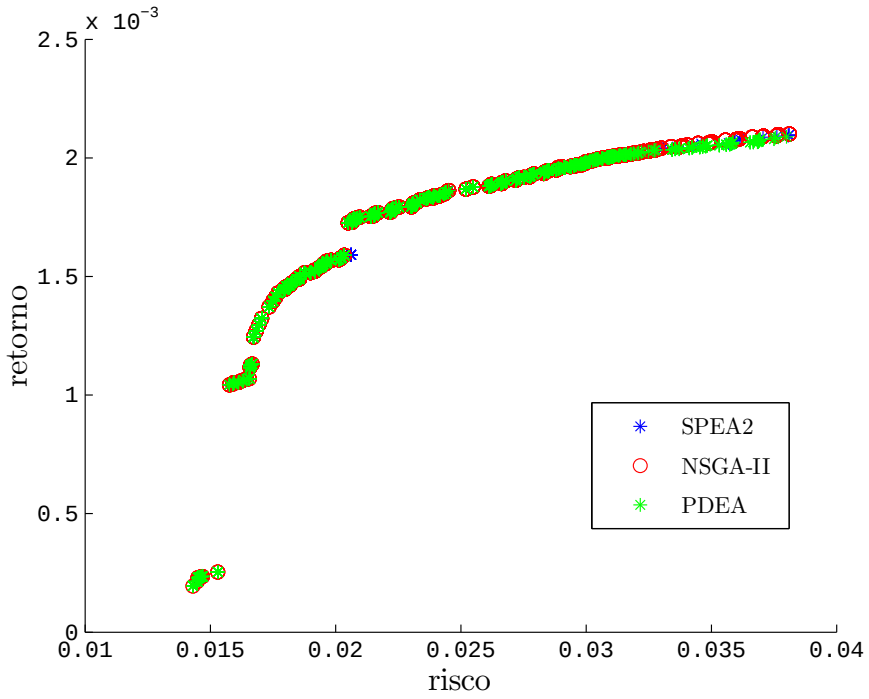
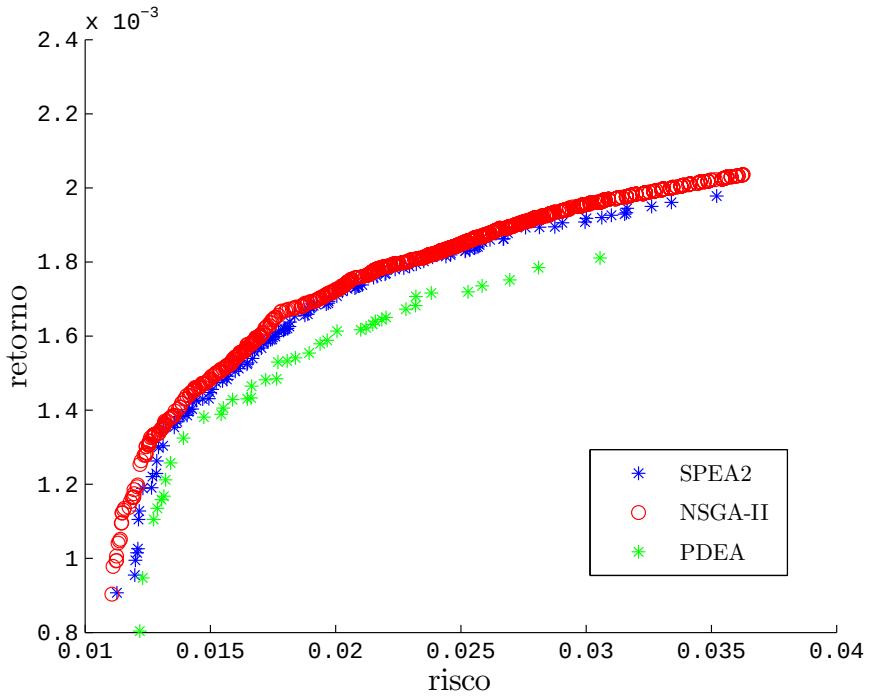
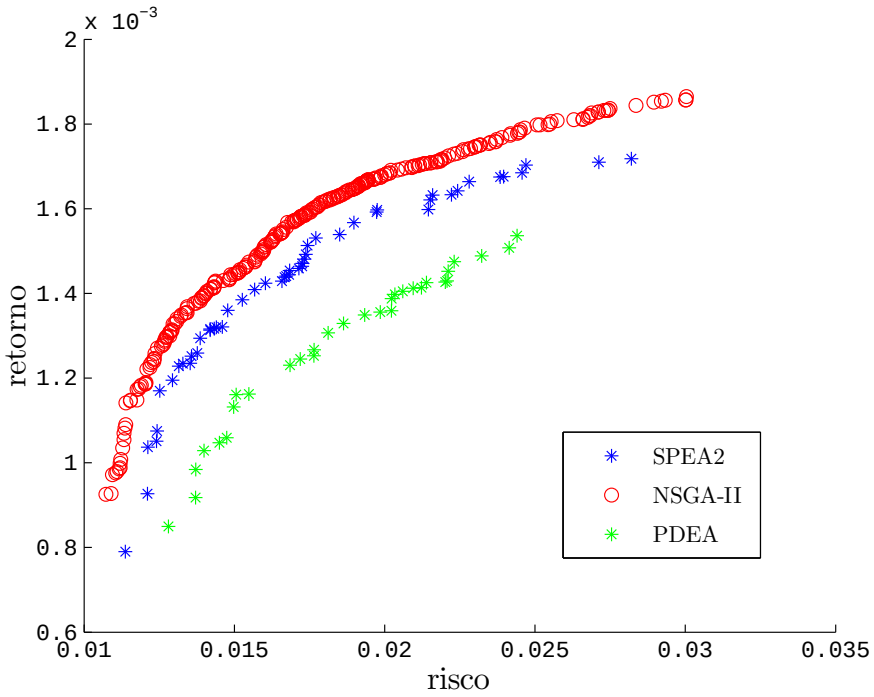


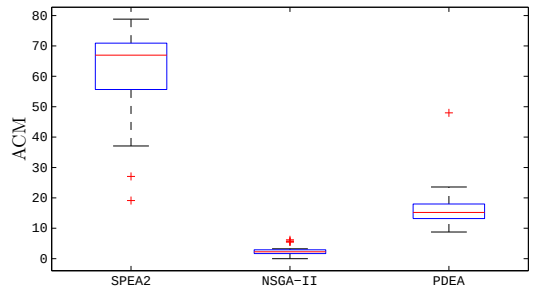
Figura 29 – Fronteiras não-dominadas para cardinalidade 9



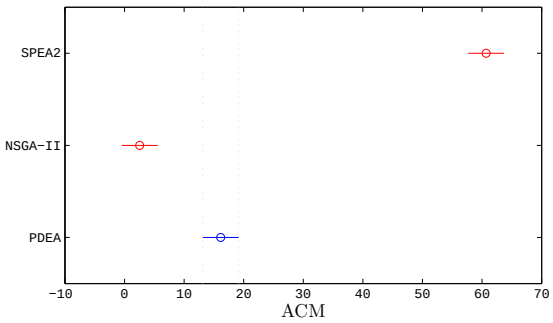
lembrando que, para ACM, quanto menor seu valor melhor é a fronteira, pode-se dizer que o desempenho do NSGA-II foi melhor que do PDEA, que foi melhor que do SPEA2, nesse caso. Mesmo assim, um teste de Tukey com nível de significância de 5% foi realizado, como mostrado pela Figura 31b e o resultado mostrado pelo *boxplot* foi confirmado.

Figura 30 – Fronteiras não-dominadas para cardinalidade 15





(a) *Boxplot* de ACM para cardinalidade 3



(b) Teste de Tukey de ACM para cardinalidade 3

Figura 31 – *Boxplot* e teste de Tukey de ACM para cardinalidade 3

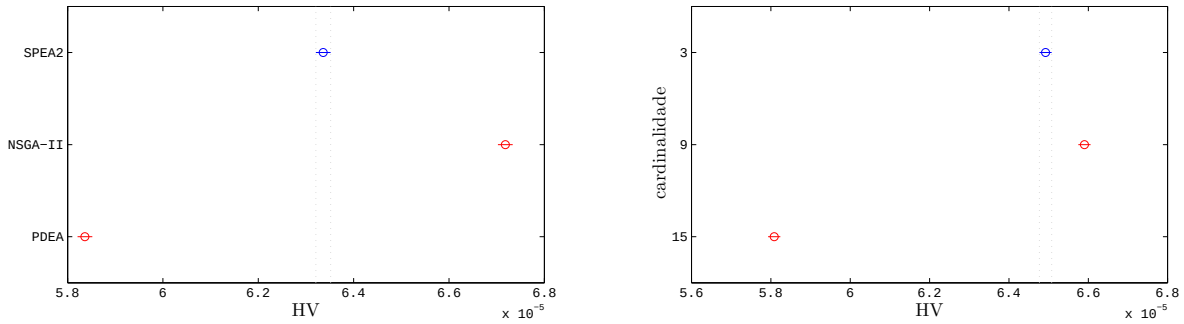
ACM é uma métrica que é calculada para uma fronteira em relação a outras e, assim, foi utilizada apenas para comparar diferentes algoritmos para um mesmo modelo. Já as métricas HV e DM são calculadas para cada fronteira de forma independente e, por isso, foram utilizadas nesse teste para comparar tanto os algoritmos quanto diferentes modelos (utilizando diferentes cardinalidades). Dessa forma, para tais métricas, foram realizados testes ANOVA com dois fatores (algoritmo e cardinalidade).

Para a métrica hipervolume (HV), o teste ANOVA, descrito pela Tabela 1, detectou diferença, tanto para algoritmos quanto para cardinalidades, além de detectar interação entre os dois fatores, ao nível de confiança de 5%. Teste de Tukey na Figura 32a revela que, para essa métrica, o NSGA-II foi superior ao SPEA2, que foi superior ao PDEA e o teste na Figura 32b mostra que o modelo com cardinalidade de 9 ativos proporciona portfólios mais

eficientes que o modelo com 3 ativos, que gera portfólios mais eficientes que o modelo com cardinalidade 15.

Tabela 1 – ANOVA 2 fatores de HV para diferentes algoritmos e cardinalidades

Fonte	SS	df	MS	F	p-valor
Coluna	$3{,}52069 \times 10^{-9}$	2	$1{,}76034 \times 10^{-9}$	2235,67	0
Linha	$3{,}26642 \times 10^{-9}$	2	$1{,}63321 \times 10^{-9}$	2074,2	0
Interação	$1{,}93486 \times 10^{-9}$	4	$4{,}83714 \times 10^{-10}$	614,32	0
Erro	$2{,}05509 \times 10^{-10}$	261	$7{,}87392 \times 10^{-13}$		
Total	$8{,}92747 \times 10^{-9}$	269			



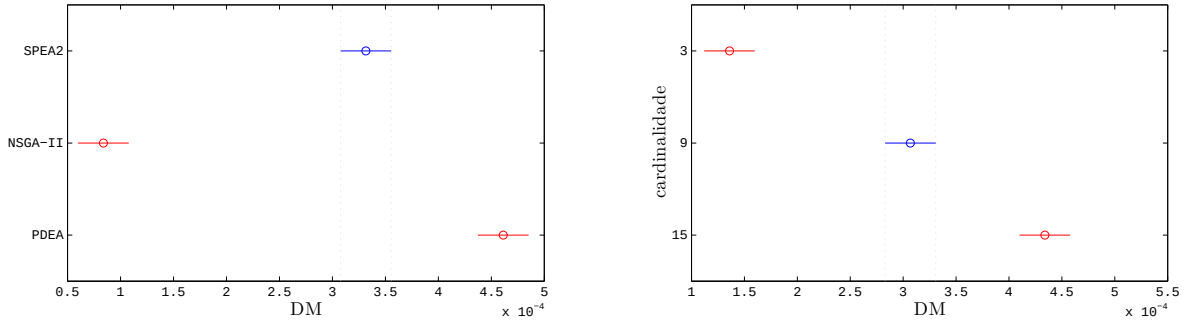
(a) Teste de Tukey de HV para diferentes algoritmos (b) Teste de Tukey de HV para diferentes cardinalidades

Figura 32 – Testes de Tukey de Hipervolume (HV)

O mesmo ocorre para a métrica Δ (DM), em que observa-se pelo teste ANOVA, mostrado na Tabela 2, que existe diferença estatística tanto entre os diferentes algoritmos quanto entre diferentes cardinalidades ao nível de significância de 5%. Posteriormente, pelo teste de Tukey mostrado na Figura 33a, verifica-se que, exatamente como ocorre para o hipervolume, a fronteira do NSGA-II apresenta um espalhamento melhor que a do SPEA2 que, por sua vez, é melhor do que a do PDEA. O teste de Tukey mostrado na Figura 33b revela que o espalhamento no modelo de cardinalidade 3 é melhor que no modelo de cardinalidade 9, que é melhor que no de 15, como esperado, já que a dificuldade do problema cresce com o aumento da cardinalidade do portfólio, diminuindo assim, as chances de se encontrar boas soluções.

Tabela 2 – ANOVA 2 fatores de DM para diferentes algoritmos e cardinalidades

Fonte	SS	df	MS	F	p-valor
Coluna	$6{,}62028 \times 10^{-6}$	2	$3{,}31014 \times 10^{-6}$	176,94	0
Linha	$4{,}02801 \times 10^{-6}$	2	$2{,}014 \times 10^{-6}$	107,66	0
Interação	$3{,}2242 \times 10^{-6}$	4	$8{,}0605 \times 10^{-7}$	43,09	0
Erro	$4{,}88269 \times 10^{-6}$	261	$1{,}87076 \times 10^{-8}$		
Total	$1{,}87552 \times 10^{-5}$	269			



(a) Teste de Tukey de DM para diferentes algoritmos (b) Teste de Tukey de DM para diferentes cardinalidades

Figura 33 – Testes de Tukey de DM (métrica Δ)

Verificou-se, portanto, que os testes estatísticos, mostrados nas Figuras 31b, 32a e 33a condizem com a afirmação de que, no geral, o NSGA-II apresenta desempenho melhor que o SPEA2 e que o PDEA apresenta o pior desempenho entre os algoritmos utilizados para o problema proposto, como os gráficos das Figuras 28, 29 e 30 sugerem. Em relação ao tempo de execução, nenhum teste estatístico foi necessário, já que a sua dispersão apresentada foi muito pequena e não houve sobreposição desses tempos em diferentes algoritmos. A média desses tempos, para o NSGA-II, SPEA2 e PDEA foi, respectivamente, $78{,}27$, $134{,}32$ e $99{,}02$ segundos, considerando as cardinalidades 3, 9 e 15.

Apesar do modelo de cardinalidade 3 apresentar melhor espalhamento e diversidade de soluções, como era esperado, o modelo de cardinalidade 9 apresentou portfólios de melhores desempenhos, ou seja, para um dado risco, os algoritmos encontram, na maior parte das vezes, portfólios de 9 ativos com retorno maior que aqueles de cardinalidade 3 ou 15.

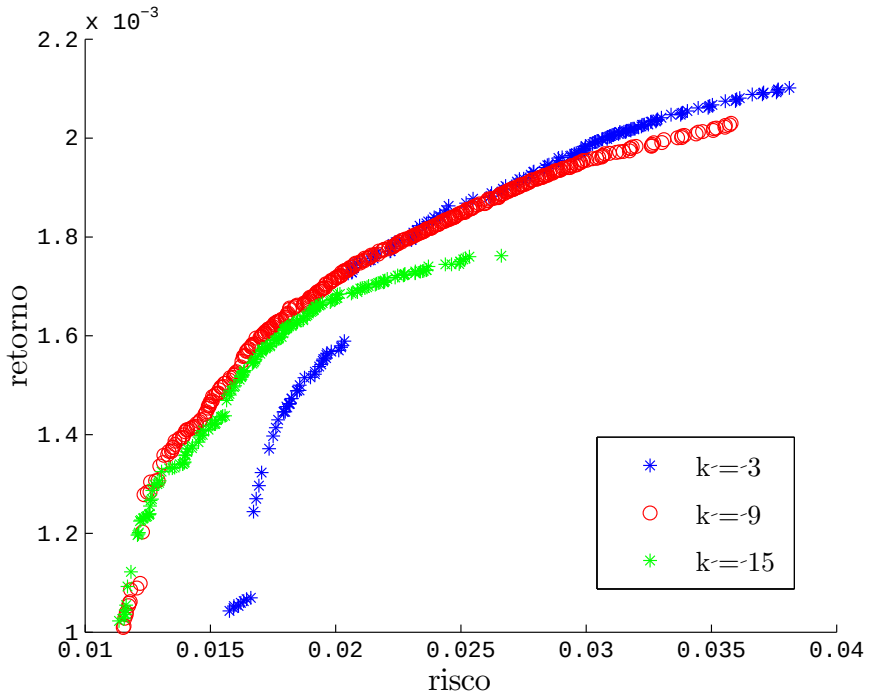
Para o algoritmo NSGA-II, por exemplo, portfólios gerados com diferentes cardinalidades podem ser observados na Figura 34, que mostra os portfólios de cardinalidade 9 quase sempre dominando os de cardinalidade 15, sugerindo que, a partir desse valor, o aumento da cardinalidade gerará portfólios cada vez piores. É importante observar que o portfólio de maior retorno gerado possui cardinalidade 3 e o de menor risco, cardinalidade 15, confirmando que a diversificação de investimento realmente diminui o risco, ao mesmo tempo que descarta investimentos de grande retorno esperado e alto risco.

4.2 Segundo Experimento

Na seção 4.1, resultados sugerem que, dentre os valores de cardinalidade utilizados, portfólios com cardinalidade 9 apresentaram, na média, melhores desempenhos e, por isso, esse e os próximos testes *in-sample* utilizarão sempre cardinalidade igual a 9.

Ainda na seção 4.1, é possível notar que os resultados apresentados indicam a

Figura 34 – Fronteira gerada pelo NSGA-II para diferentes cardinalidades



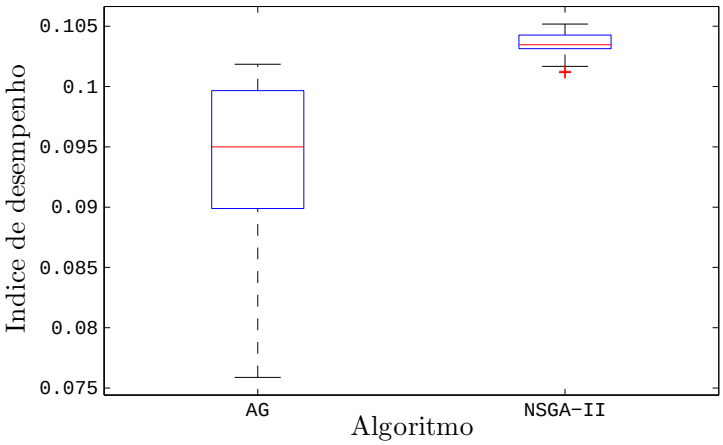
superioridade do algoritmo NSGA-II em relação aos demais para todas as métricas utilizadas. Dessa forma, nesse experimento os portfólios gerados pelo NSGA-II são comparados aos portfólios gerados por um algoritmo genético monobjetivo, tendo como função objetivo, a ser maximizada, a razão entre retorno e risco e que utiliza os mesmos operadores de cruzamento e mutação do NSGA-II.

Para o algoritmo genético monobjetivo, o método roleta (como descrito na subseção 2.1.2.2) foi utilizado na seleção dos indivíduos. Também foi utilizado o elitismo de um indivíduo, fazendo com que a melhor solução de uma dada geração também esteja na próxima geração. Parâmetros como número de indivíduos e número máximo de gerações são os mesmos utilizados no NSGA-II. Para probabilidade de cruzamento (pc) e de mutação (pm), foram utilizados os valores $0{,}8$ e $0{,}1$, respectivamente, determinados empiricamente após testes prévios.

Cada um dos algoritmos foi executado 30 vezes e para o NSGA-II, foi registrado o maior índice de desempenho (razão entre retorno e risco) dentre todos os portfólios gerados por este e, para o algoritmo genético (AG), foi registrado o índice de desempenho do melhor portfólio de sua última geração. Figura 35 mostra o *boxplot* desse índice para os dois algoritmos e, como pode-se observar, o NSGA-II proporciona portfólios com maiores índices, na média. Ainda observa-se que a dispersão apresentada pelo AG desse índice nos vários testes é muito maior que a do NSGA-II, de forma que este não precisa de várias execuções para se encontrar portfólios com bom índice de desempenho.

A partir dos resultados apresentados, pode-se afirmar que um algoritmo genético

Figura 35 – *Boxplot* do índice de desempenho para AG e NSGA-II



multiobjetivo simples apresentou melhores portfólios que um algoritmo genético monobjetivo simples, com os mesmos operadores usados no multiobjetivo, mesmo quando um único portfólio é desejado. Dessa forma, o algoritmo multiobjetivo continuará sendo utilizado nos próximos testes. É importante ressaltar que não pode-se afirmar que o modelo monobjetivo é pior que o modelo multiobjetivo. O teste realizado apenas indica que um algoritmo genético monobjetivo simples não é tão eficiente quanto o NSGA-II para o problema proposto, mesmo quando se deseja uma única solução, justificando a sua utilização.

Outros motivos para a utilização de algoritmos multiobjetivos incluem a possibilidade de proporcionar diferentes tipos de investimentos a investidores com diferentes perfis, variando de investimentos de baixo risco e menor retorno esperado até aqueles de alto risco e maior probabilidade de grandes retornos. Apesar disso, uma vantagem do algoritmo monobjetivo é o seu tempo de execução que foi, na média, $49{,}87$ segundos, enquanto que a do NSGA-II foi de $77{,}76$ segundos.

O NSGA-II, que apresentou bom desempenho nos testes anteriores, é chamado nesse teste de NSGA-II 1 e foi comparado com o NSGA-II 2 e 3. Para o NSGA-II 2, também foi utilizado tamanho de população e número máximo de gerações iguais a 500 e, para os demais parâmetros, os valores sugeridos em Deb et al. (2011) foram utilizados. Pelo mesmo motivo do experimento anterior, utilizou-se cardinalidade 9 para portfólios.

O NSGA-II 3 transforma os resultados obtidos pelo NSGA-II 2 em soluções com números inteiros de lotes. Um exemplo, na Tabela 3 mostra as soluções do NSGA-II 2, com números flutuantes de lotes. O NSGA-II 3 aplica a função piso sobre o número de lotes dessas soluções, como pode ser visto na Tabela 4.

A Figura 36 apresenta as fronteiras não-dominadas combinadas para esses algoritmos, pelas quais pode-se facilmente notar uma inferioridade do NSGA-II 3 em relação aos demais. A partir desse resultado, pode-se argumentar que, mesmo que o algoritmo

Tabela 3 – Soluções do NSGA-II 2

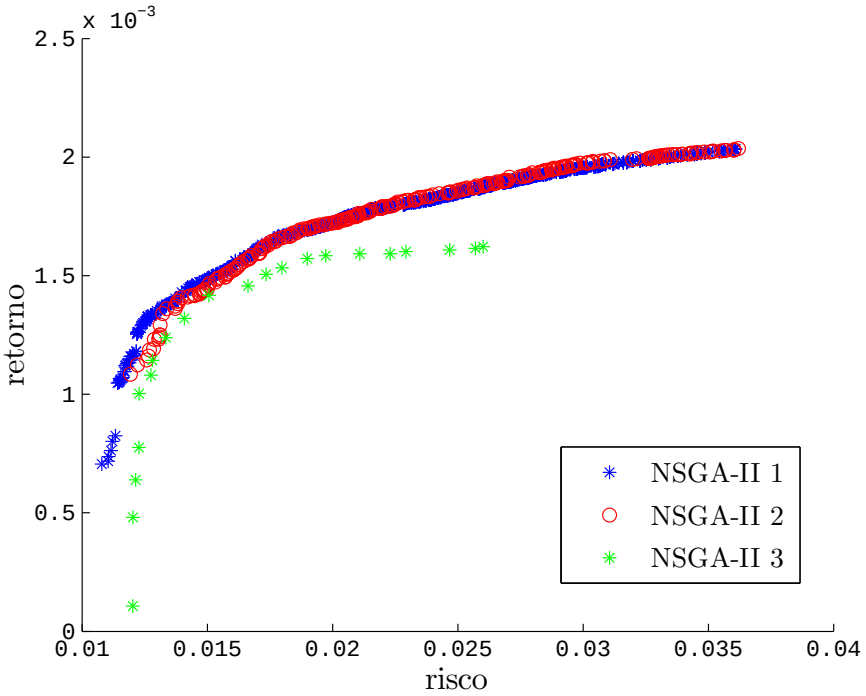
Ativo	Proporção	Lotes
ABEV3	0,168039	16,65
BRKM5	0,056685	2,78
CTIP3	0,198350	8,42
FIBR3	0,052070	1,97
LREN3	0,057481	5,09
PETR4	0,072728	2,67
RENT3	0,061494	2,40
TBLE3	0,054409	1,98
UGPA3	0,276071	10,50

Tabela 4 – Soluções do NSGA-II 3

Ativo	Proporção	Lotes
ABEV3	0,161488	16
BRKM5	0,040740	2
CTIP3	0,188432	8
FIBR3	0,026490	1
LREN3	0,056400	5
PETR4	0,054580	2
RENT3	0,051238	2
TBLE3	0,027450	1
UGPA3	0,262750	10

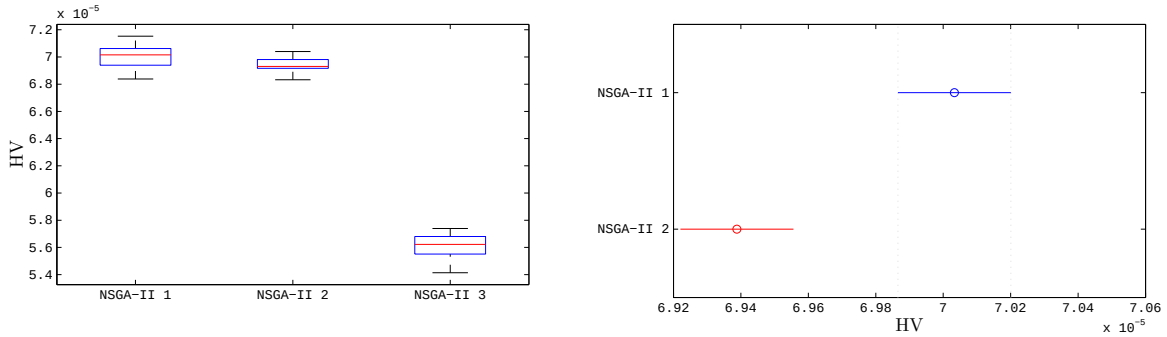
que utiliza o modelo contínuo (NSGA-II 2) for superior ao algoritmo para o modelo proposto (NSGA-II 1), para propósitos práticos, em que deseja-se valores múltiplos do custo do lote mínimo de cada ativo, o NSGA-II 1 ainda seria preferível. Mesmo assim, supondo que exista formas melhores de obter tais soluções em lotes, a partir do modelo discreto, os algoritmos foram comparados levando em consideração as métricas adotadas para testes *in-sample*.

Figura 36 – Fronteiras não-dominadas apresentadas pelos algoritmos NSGA-II 1, 2 e 3



Primeiramente, em relação ao hipervolume (HV), como esperado, percebe-se pelo *boxplot* da Figura 37a que o NSGA-II 3 apresenta o pior desempenho. Em relação aos outros dois algoritmos, não é possível determinar, pelo *boxplot*, a existência de diferença estatística entre eles, mas o teste ANOVA detectou tal diferença, ao nível de significância de 5% e o teste de Tukey, apresentado na Figura 37b, sugere a superioridade do NSGA-II 1

sobre o NSGA-II 2 em relação a essa métrica.



(a) *Boxplot* de HV para NSGA-II 1, 2 e 3 (b) Teste de Tukey de HV para NSGA-II 1 e 2

Figura 37 – *Boxplot* e teste de Tukey de HV (hipervolume)

Para a média da métrica C (ACM), apenas o *boxplot* apresentado na Figura 38 é suficiente para afirmar que o NSGA-II 1 é superior ao 2, que é melhor que o 3. O mesmo ocorre para a métrica Δ (DM), cujo *boxplot* pode ser observado na Figura 39.

Figura 38 – *Boxplot* de ACM para NSGA-II 1, 2 e 3

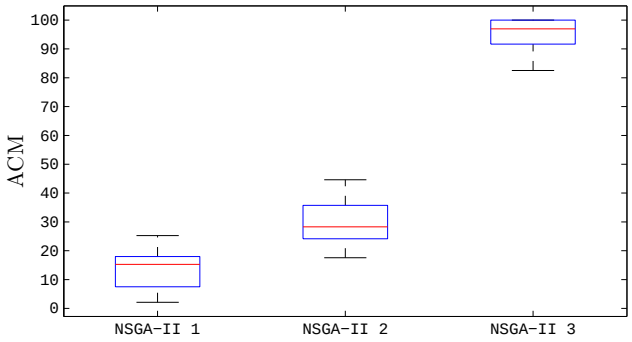
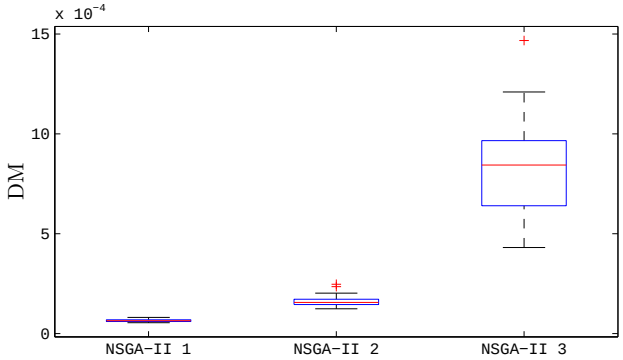


Figura 39 – *Boxplot* de DM para NSGA-II 1, 2 e 3



Resultados sugerem, então, que o algoritmo desenvolvido para o modelo proposto (NSGA-II 1), além de proporcionar portfólios mais eficientes, também proporcionam maior

diversidade desses. O NSGA-II 3, que é o outro algoritmo que proporciona portfólios que podem ser negociados, na prática, gera investimentos piores (dominados pelas soluções do NSGA-II 1) e pouco diversificadas, em relação ao espalhamento das soluções, devido à grande quantidade gerada de soluções dominadas pela sua própria fronteira. Dessa forma, esse experimento justifica a utilização do modelo proposto (discreto).

O motivo da superioridade do NSGA-II 1 (que utiliza modelo discreto) pode ser o fato de que o NSGA-II 2 (que utiliza modelo contínuo), proposto por Deb et al. (2011), consome todo o capital disponível, enquanto que o NSGA-II 1 utiliza o modelo proposto neste trabalho e sua restrição de capital disponível (restrição 11b) permite que portfólios de custos variados sejam gerados.

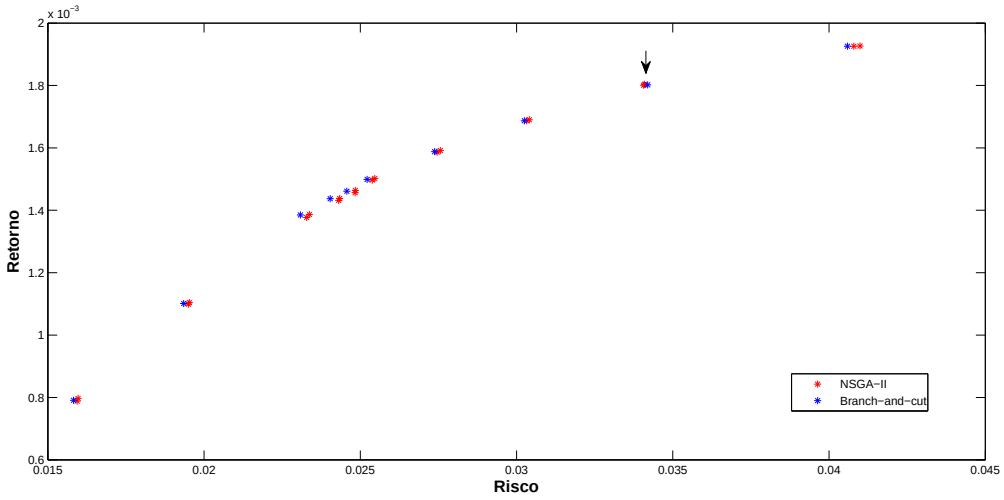
O NSGA-II 1 também se mostra superior aos demais em relação ao tempo de execução, para o qual a média apresentada pelo NSGA-II 2 foi de $213{,}52$ segundos e, como esperado, o tempo de execução do NSGA-II 3 foi quase igual ao do NSGA-II 2, já que aquele utiliza as soluções do NSGA-II 2 e, a partir dessas, realiza poucas operações. Enquanto isso, o NSGA-II 1 apresentou uma média de $75{,}92$ segundos.

O desempenho do NSGA-II ainda foi comparado com um método exato, utilizando um modelo monobjetivo linear. É importante ressaltar que todas as soluções obtidas pelo método exato *branch-and-cut* possuem *gap*, que corresponde à diferença entre os valores obtidos pelas soluções dos problemas primal e dual, igual a 0, ou seja, são soluções ótimas.

Um teste *in-sample* foi realizado e analisando as soluções geradas por cada método, como mostrado na Figura 40, pode-se notar que o método exato quase sempre gera soluções que dominam as soluções do NSGA-II. As soluções do método exato apresentaram risco inferior aos das soluções do NSGA-II com retorno imediatamente abaixo e imediatamente acima do retorno daquela solução (ponto azul à esquerda dos pontos vermelhos na figura), exceto a solução destacada pela seta na Figura 40, em que a solução do método exato foi dominada por uma solução do NSGA-II.

A ocorrência de soluções do método exato dominadas por soluções do método heurístico se deve ao fato de que o método exato utiliza um modelo aproximado do modelo proposto, já que as técnicas utilizadas nesse caso solucionam apenas modelos lineares. E, apesar do modelo exato apresentar desempenho melhor que o NSGA-II, mesmo para o modelo não-linear proposto, a distância entre essas soluções são relativamente pequenas, como pode-se observar na Figura. Já em relação ao tempo de execução, o método exato gasta em média $247{,}19$ segundos para realizar a otimização para um único portfólio, enquanto que o NSGA-II gera 500 portfólios espalhados pela fronteira não-dominada em $77{,}76$ segundos, em média. Considerando essa grande diferença no tempo de execução, então, os próximos experimentos utilizarão a meta-heurística NSGA-II e, em trabalhos futuros, algoritmos híbridos, compondo com o método exato.

Figura 40 – Fronteira não-dominada dos métodos exato e heurístico



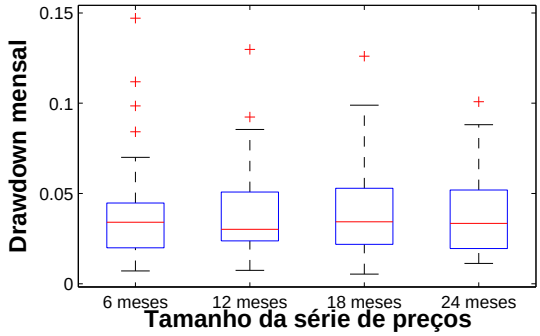
4.3 Terceiro Experimento

Esta seção apresenta todos os testes *out-of-sample* realizados neste trabalho, comparando diferentes tamanhos de série histórica *in-sample*, modelos, meta-heurísticas e cardinalidades com o objetivo de comparar os ganhos financeiros proporcionados por cada configuração de parâmetros.

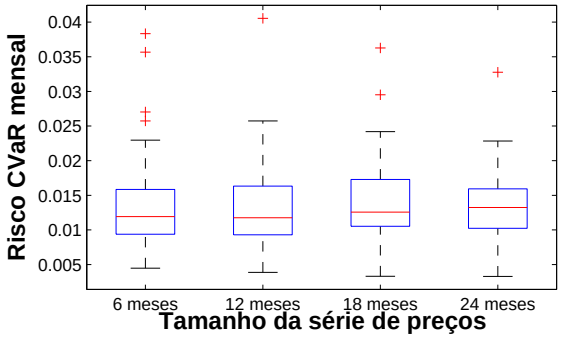
O primeiro teste *out-of-sample* foi realizado com o objetivo de analisar o efeito de diferentes tamanhos de série de preços no desempenho dos portfólios gerados. O tamanho dessa série histórica foi alterada entre os valores de 6, 12, 18 e 24 meses. Ao realizar tal simulação, verifica-se pouca diferença em relação aos valores mensais de *Drawdown* (ver Figura 41a), risco CVaR (ver Figura 41b) e retorno (ver Figura 41c). Mesmo assim, foram aplicados testes de variância que falharam em detectar diferenças estatísticas, ao nível de significância de 5%, para diferentes tamanhos de série histórica nos três casos, sendo que todos os testes indicaram um p-valor superior a $0{,}9$.

Também foi analisado o retorno acumulado utilizando cada um dos diferentes tamanhos de série histórica no período entre janeiro de 2012 e dezembro de 2015. Como pode ser visto na Figura 42, não foi possível estabelecer um padrão entre o tamanho das séries e o retorno acumulado proporcionado quando utiliza-se cada um desses para otimização de portfólio. Assim, o tamanho de série histórica de 1 ano foi definida para os próximos testes, com base em trabalhos relacionados, como o de Barroso (2017), que utiliza esse valor.

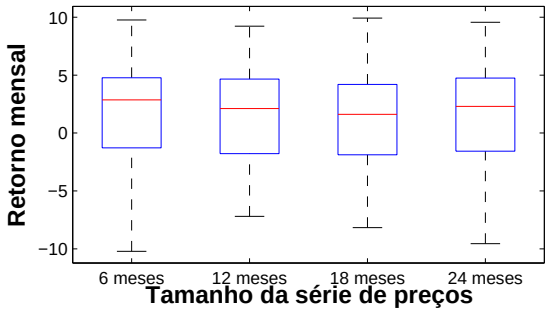
Para a comparação do modelo proposto com o modelo de Hanaoka (2014) é realizado um teste *out-of-sample* que utiliza portfólios fornecidos pelo procedimento de otimização para simulação de investimento no mercado de ações, no período entre 2011 a 2015. Além desses dois modelos, o teste considerou também o modelo proposto quando



(a) *Boxplot* de *Drawdown* mensal



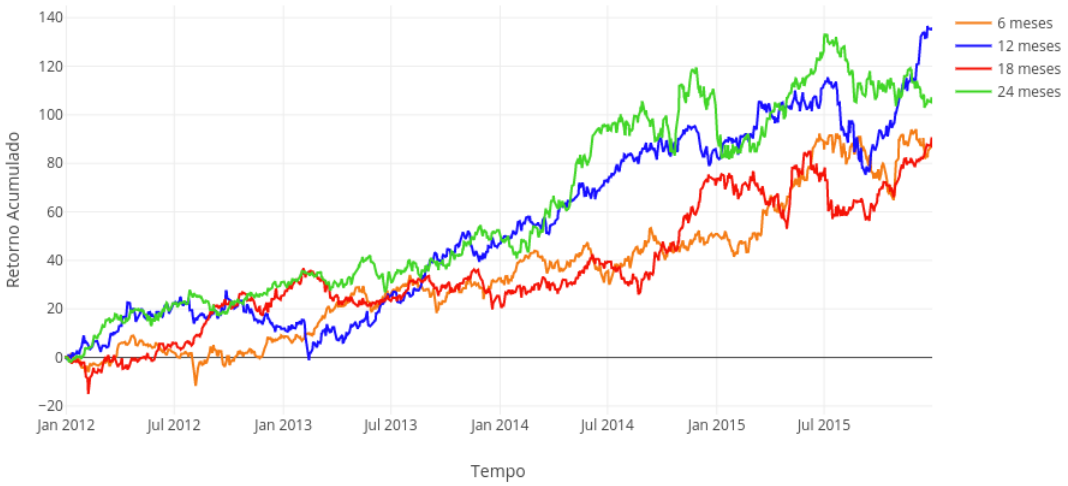
(b) *Boxplot* de risco CVaR mensal



(c) *Boxplot* de retorno mensal

Figura 41 – *Boxplots* para diferentes tamanhos de série histórica

Figura 42 – Retorno acumulado para diferentes tamanhos de série histórica



são desconsiderados os custos de transação, para avaliar os seus efeitos e o próprio índice Bovespa (Ibovespa), que reflete o desempenho geral do mercado brasileiro e servirá como *benchmark*. Assim, os modelos usados nesse experimento foram:

- modelo 1, que corresponde ao modelo proposto desconsiderando todos os custos de transação;
- modelo 2, o modelo de Hanaoka (2014), discreto e sem rebalanceamento;

- modelo 3, o modelo proposto neste trabalho, discreto e com rebalanceamento;
- modelo 4, modelo básico de investimento de todo o capital disponível no índice Bovespa, servindo como *benchmark*.

Para cada um dos modelos foram utilizados os três algoritmos (NSGA-II, PDEA e SPEA2). Dessa forma, desejando avaliar cada um dos modelos e algoritmos nessa simulação e, posteriormente, decidir a melhor combinação entre modelos e algoritmos, foi realizado uma análise de dois fatores para cada uma das métricas adotadas para esse tipo de teste (retorno mensal, risco CVaR mensal e Drawdown mensal). Uma vez que o período considerado é constituído de 5 anos e as métricas adotadas são mensais, 60 repetições são utilizadas para cada combinação.

Primeiramente, em relação ao risco Drawdown mensal, percebe-se por meio da Tabela 5 que, ao nível de significância de 5%, não foi detectada diferença estatística em relação aos diferentes algoritmos. Já para os diferentes modelos, a hipótese nula de igualdade das médias do valor Drawdown é descartada, indicando diferença entre os modelos. Posteriormente, o teste de Tukey, apresentado na Figura 43 revela que o modelo 4 apresenta valor Drawdown pior que os demais. O resultado realça a importância de se fazer uma boa otimização, considerando modelos e métodos adequados, uma vez que o modelo 4 foi superado facilmente pelos modelos de otimização de portfólio. Além disso, o teste sugere um melhor Drawdown médio para o método 1 em relação ao 3, indicando que o modelo sem rebalanceamento apresenta maiores perdas que o modelo sem custo de transação.

Tabela 5 – ANOVA 2 fatores para Drawdown

Fonte	SS	df	MS	F	p-valor
Mês	$0{,}42585$	59	$0{,}0072178$	40.1077	0
Algoritmo	$0{,}00049$	2	$0{,}0002431$	$1{,}3507$	$0{,}25980$
Método	$0{,}06764$	3	$0{,}0225458$	$125{,}2828$	0
Algoritmo:Método	$0{,}00269$	6	$0{,}0004481$	$2{,}4901$	$0{,}02173$
Erro	$0{,}11679$	649	$0{,}0001800$		

O mesmo ocorre para risco CVaR e retorno mensal, ou seja, são detectadas diferenças estatísticas apenas para modelos no teste de risco CVaR (como mostra a Tabela 6) e de retorno (como mostra a Tabela 7). Testes de Tukey mostram que, além de apresentarem menores riscos (ver Figura 44), os modelos de otimização apresentam maiores ganhos que o *benchmark* (ver Figura 45). Enquanto isso, o teste não foi capaz de detectar diferenças entre os diferentes modelos de otimização ou entre os diferentes algoritmos, mesmo sendo constatadas diferenças nos testes *in-sample*, como mostrado na seção 4.1.

Para analisar melhor, então, o comportamento dos investimentos quando utilizados cada um dos modelos, os retornos acumulados apresentados por cada um deles são

Figura 43 – Teste de Tukey para valores de Drawdown nos diferentes modelos

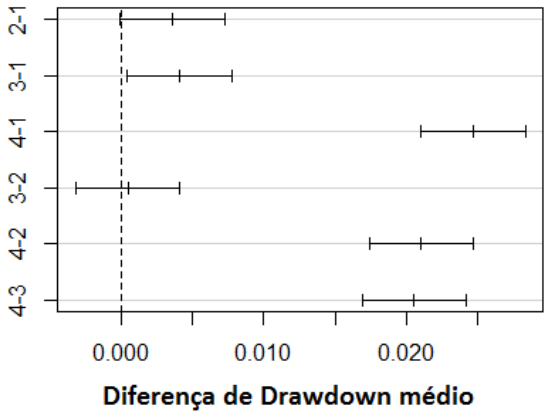
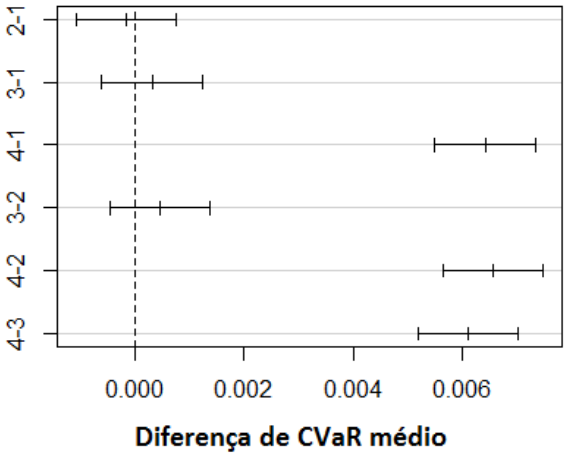


Tabela 6 – ANOVA 2 fatores para Risco CVaR

Fonte	SS	df	MS	F	p-valor
Mês	0,0252261	59	0,00042756	37,3434	0
Algoritmo	0,0000099	2	0,00000493	0,4309	0.6501
Método	0,0054848	3	0,00182827	159,6817	0
Algoritmo:Método	0,0000246	6	0,00000410	0,3577	0,9055
Erro	0,0074307	649	0,00001145		

Figura 44 – Teste de Tukey para riscos CVaR nos diferentes modelos



confrontados em gráficos. A Figura 46 apresenta esses retornos para os três modelos de otimização (modelo 1 ou M1, modelo 2 ou M2 e modelo 3 ou M3) utilizando-se o PDEA e nota-se que, apesar dos retornos acumulados estarem muito próximos no início, fazendo com que as linhas no gráfico se entrelacem, ao final o modelo de Hanaoka (2014), sem rebalanceamento, parece distanciar-se mais dos outros modelos.

Esse distanciamento ocorre de forma mais nítida nos algoritmos SPEA2 (cujo retorno acumulado é apresentado pela Figura 47) e NSGA-II (cujo retorno acumulado é

Tabela 7 – ANOVA 2 fatores para Retorno

Fonte	SS	df	MS	F	p-valor
Mês	8512,3	59	144,276	17,9133	0
Algoritmo	10,3	2	5,152	0,6397	0,5278
Método	434,2	3	144,723	17,9688	0
Algoritmo:Método	9,2	6	1,525	0,1894	0,9798
Erro	5227,1	649	8,054		

Figura 45 – Teste de Tukey para retornos nos diferentes modelos

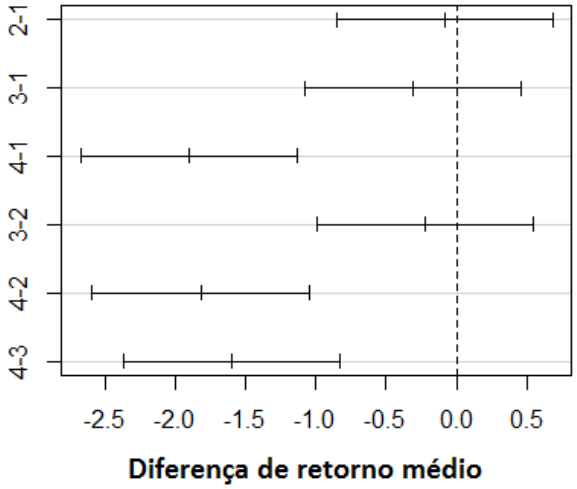
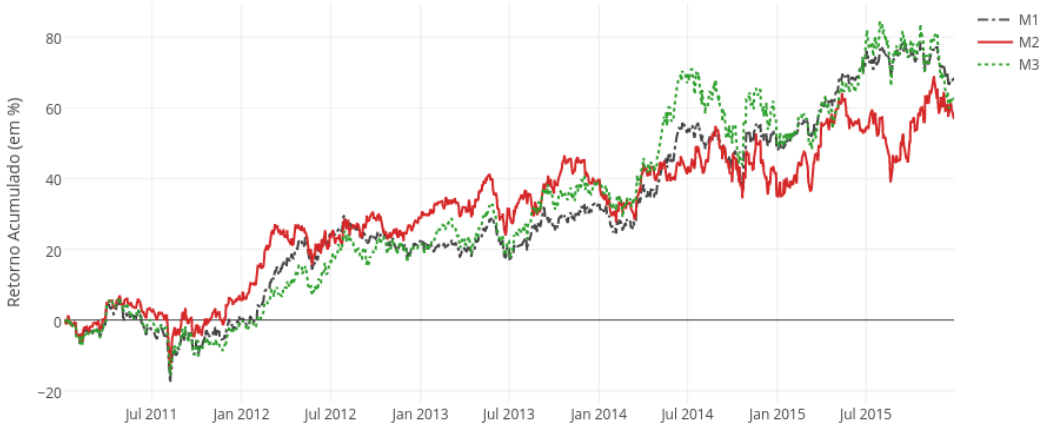


Figura 46 – Retornos acumulados para os diferentes modelos utilizando PDEA



apresentado pela Figura 48). Esse padrão pode sugerir que o modelo de Hanaoka (2014) proporciona menor ganho em relação aos outros dois, como era esperado, já que o processo de rebalanceamento do modelo proposto procura minimizar os custos de transação, que podem chegar a valores altos naquele modelo. Já o modelo sem custos corresponde a

um cenário ideal em que não existe tais custos, sobrando mais capital a ser investido. Essa sobra tende a aumentar com a quantidade de investimentos realizadas e, por isso, haveria diferenças mais significativas apenas ao final do gráfico. Assim, esperava-se também maiores ganhos proporcionados pelo modelo sem custos, o que pode ser verificado apenas quando utilizou-se o SPEA2, sendo que o modelo proposto apresentou retorno acumulado similar à este para os demais algoritmos.

Figura 47 – Retornos acumulados para os diferentes modelos utilizando SPEA2

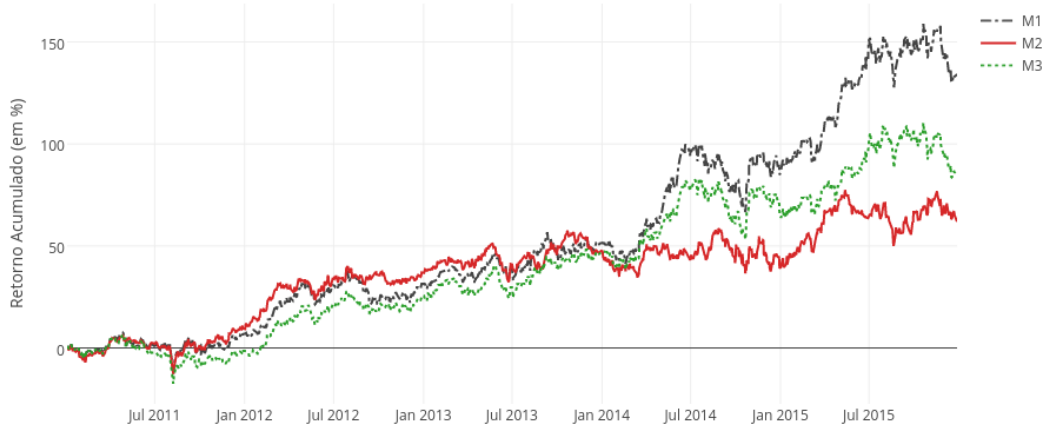
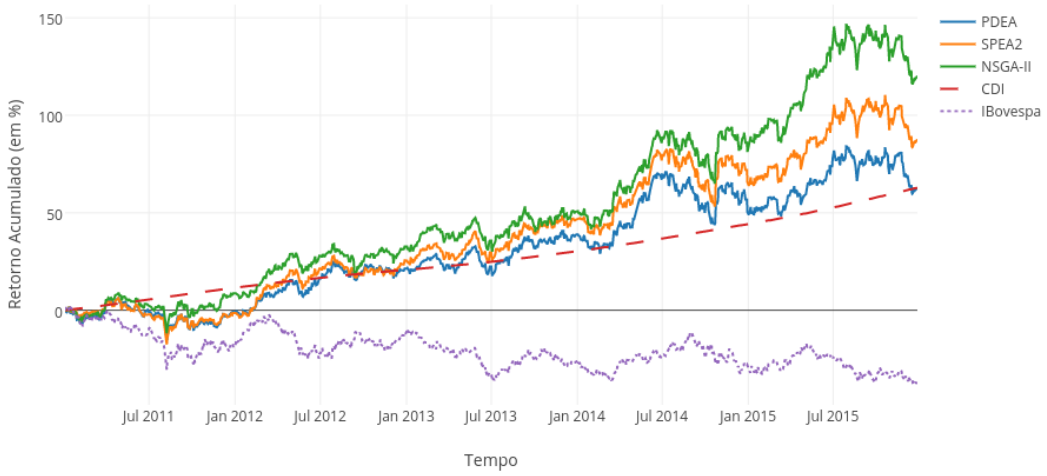


Figura 48 – Retornos acumulados para os diferentes modelos utilizando NSGA-II

O mesmo procedimento que foi realizado para analisar o comportamento dos investimentos utilizando diferentes modelos, também foi realizado para os diferentes algoritmos e, novamente, um padrão pôde ser observado. A Figura 49 mostra o comportamento do

retorno acumulado ao longo do período estudado para os três algoritmos utilizando o modelo proposto, com rebalanceamento e custo de transação. Em relação aos três algoritmos, percebe-se melhor desempenho quando utiliza-se o NSGA-II, em segundo lugar fica o SPEA2 e, por último, o PDEA, assim como visto nos testes *in-sample* da seção 4.1, e tal padrão também se repete para os outros modelos. O resultado sugere, dessa forma, uma convergência entre os resultados *in-sample* e *out-of-sample* em relação ao desempenho dos algoritmos.

Figura 49 – Retornos acumulados para os diferentes algoritmos

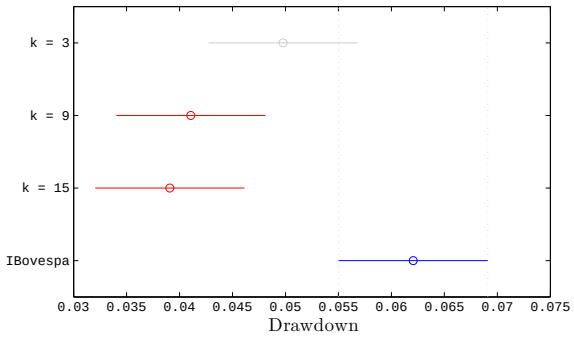


Pode-se notar, também na Figura 49, o retorno acumulado proporcionado quando todo o capital é investido no índice Bovespa durante esse período. O comportamento do Ibovespa revela o mal desempenho do mercado brasileiro nesse período e comparando o índice ao modelo proposto, nitidamente se percebe que, mesmo quando o mercado brasileiro apresenta um desempenho ruim, o modelo proposto, utilizando qualquer um dos algoritmos, proporcionou ganhos financeiros na simulação. Enquanto isso, um investimento no Ibovespa geraria apenas perdas durante o período. Espera-se, portanto, que quando o mercado brasileiro apresentar um bom desempenho e o Ibovespa proporcionar melhores ganhos, o modelo proposto também apresente resultados ainda melhores.

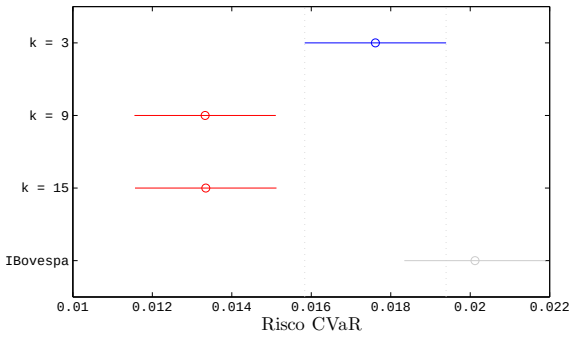
Os retornos acumulados apresentados pelo modelo proposto foram, por fim, comparados com o retorno acumulado proporcionado, nesse período, pelo indicador econômico CDI (Certificados de Depósito Interbancário), que são "os títulos de emissão das instituições financeiras, que lastreiam as operações do mercado interbancário" (CDI. . . , 2017) e, por sua taxa apresentar pequenas mudanças, o indicador é utilizado como investimento de renda fixa e como *benchmark* para comparações com outras metodologias de investimento. Ainda na Figura 49, é possível verificar, ao final do período, melhores ganhos proporcionados pelo modelo proposto, indicando a vantagem de utilizá-lo em vez de investir no CDI. É importante

observar que os ganhos do CDI foram muito maiores que do Ibovespa, destacando o mal desempenho do mercado brasileiro nesse período.

Ainda neste experimento, foi realizada uma simulação com diferentes cardinalidades para comparar os resultados obtidos com aqueles mostrados também na seção 4.1. Para isso, foi utilizado o modelo proposto com o algoritmo NSGA-II e, de acordo com as métricas Drawdown e o risco CVaR, testes de Tukey com nível de significância de 5%, realizados após a identificação de diferenças estatísticas em análises ANOVA, indicam onde estão tais diferenças. Na Figura 50a é possível observar que apenas quando utiliza-se cardinalidade 3, o modelo não apresenta Drawdown superior ao do Ibovespa. Em relação ao risco CVaR, a Figura 50b mostra um resultado ainda pior para o modelo com cardinalidade 3, já que o teste revela a superioridade do modelo com cardinalidade 9 e 15 em relação ao de cardinalidade 3 e ao *benchmark* Ibovespa. Tais resultados corroboram a teoria de Markowitz (1952), que afirma que a diversidade do portfólio reduz seu risco.



(a) Teste de Tukey de Drawdown



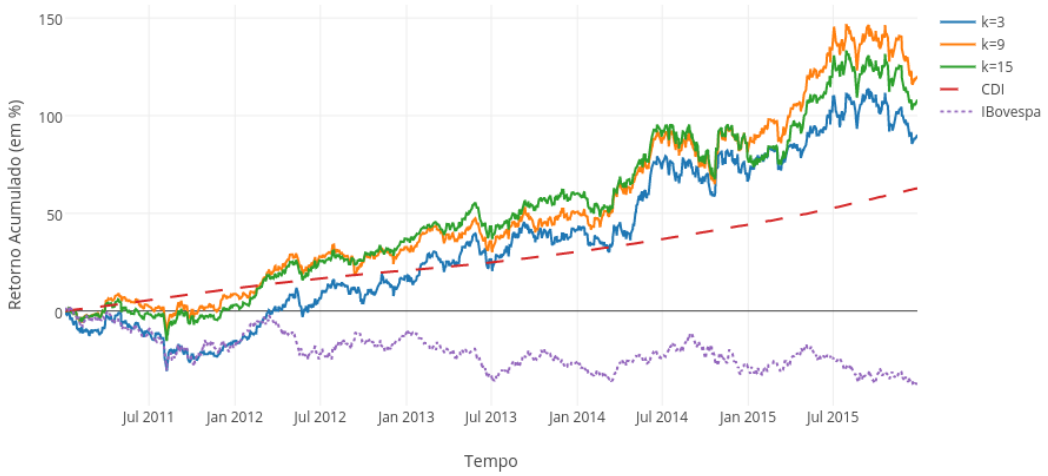
(b) Teste de Tukey de risco CVaR

Figura 50 – Teste de Tukey de riscos para cardinalidades 3, 9 e 15

Essa diferença de risco, detectada no teste de Tukey, reflete no ganho que a utilização de cada cardinalidade pode proporcionar a um investidor. A Figura 51 revela um pior retorno acumulado para o modelo de cardinalidade 3, em uma simulação de investimentos no período de 2011 a 2015. Além disso, o melhor desempenho do modelo com cardinalidade 9, condiz com os resultados dos testes *in-sample*, mostrados na seção 4.1, que mostram portfólios com cardinalidade 9 superarem os demais em termos de cobertura e diversidade das fronteiras não-dominadas produzidas.

A partir da Figura 51, também pode-se verificar a robustez do modelo, uma vez que, mesmo alterando a cardinalidade dos portfólios, o modelo proporciona ganhos que superam aqueles proporcionados pelo CDI ao fim do período estudado. Nota-se, também, uma grande distância entre os ganhos proporcionados pelo modelo proposto e pelo Ibovespa.

Figura 51 – Retornos acumulados para as diferentes cardinalidades



4.4 Quarto Experimento

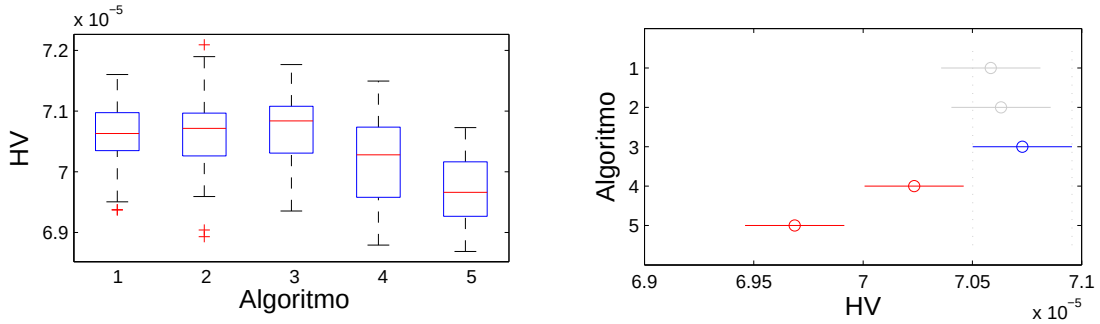
Apesar do NSGA-II apresentar, nos testes anteriores, bons resultados em comparação às outras meta-heurísticas, aproximando-se dos resultados proporcionados por técnicas exatas e com tempo de execução muito menor que essas técnicas exigem, o tempo de obtenção de soluções eficientes e diversificadas, mesmo com o NSGA-II, pode ainda estar aquém da necessidade de alguns usuários de obter uma carteira num tempo compatível com seu tipo de operação na bolsa ou com as rápidas mudanças do mercado. Na tentativa de reduzir ainda mais esse tempo de execução do NSGA-II, são propostos algoritmos paralelos utilizando modelos de ilha e computação com GPU.

Primeiramente, utilizando modelos de ilha, foram realizados testes com 2 e 4 ilhas, sendo que cada ilha é executada por uma única *thread*. Os cinco algoritmos seguintes foram comparados em um teste *in-sample*:

- algoritmo 1: NSGA-II sequencial, proposto na subseção 3.2.1, com N indivíduos;
- algoritmo 2: NSGA-II paralelo com 2 ilhas, cada uma contendo $N/2$ indivíduos;
- algoritmo 3: NSGA-II paralelo com 4 ilhas, cada uma contendo $N/4$ indivíduos;
- algoritmo 4: NSGA-II sequencial com $N/2$ indivíduos;
- algoritmo 5: NSGA-II sequencial com $N/4$ indivíduos.

O *boxplot* de hipervolume para cada um dos cinco algoritmos pode ser visto na Figura 52a, na qual observa-se valores de hipervolume dos algoritmos 4 e 5 pouco menores que os demais. Já na Figura 52b, que apresenta o teste de Tukey de hipervolume dos algoritmos com nível de significância de 5%, confirma-se uma diferença estatística entre os algoritmos 3 e 4, sendo que o 3 se mostrou superior, e entre o algoritmo 5 e os demais, sendo

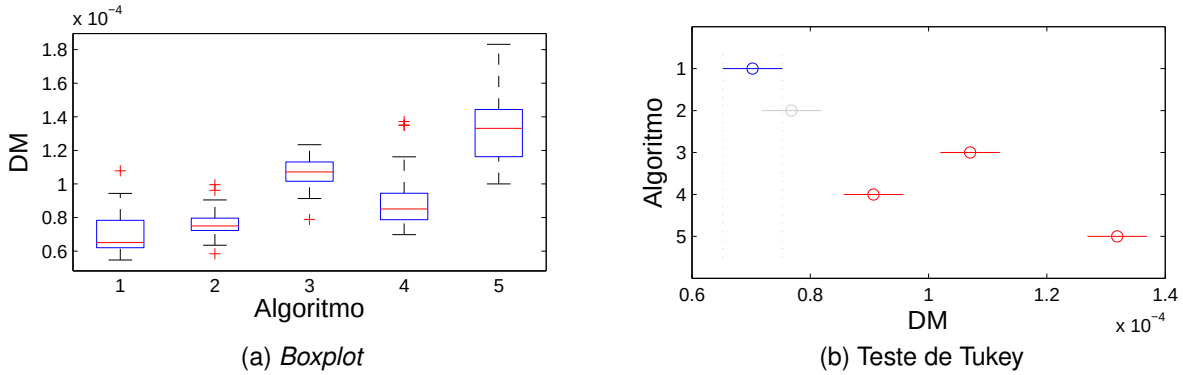
que o algoritmo 5 apresenta os piores resultados. Não foi encontrada, então, diferenças estatísticas entre os algoritmos paralelos (algoritmos 2 e 3) e o sequencial com N indivíduos (algoritmo 1).



(a) *Boxplot* (b) Teste de Tukey

Figura 52 – *Boxplot* e teste de Tukey de hipervolume para algoritmos sequenciais e paralelos

Figura 53a mostra o *boxplot* da métrica Δ (DM) para todos os cinco algoritmos. Observa-se que os algoritmos 3 e 5 apresentam espalhamento pior que os demais, enquanto que o teste de Tukey, mostrado na Figura 53b revela que, ao nível de significância de 5%, não é possível detectar diferença estatística, em média, entre a métrica Δ dos algoritmos 1 e 2, que possuem valor médio menor que a apresentada pelo algoritmo 4, que por sua vez é superada pela média do algoritmo 3 e, por fim, tem-se o algoritmo 5 com o maior valor médio para DM. A partir desses resultados, pode-se afirmar que o NSGA-II paralelo com 4 ilhas apresenta uma fronteira não-dominada com espalhamento pior que o do NSGA-II sequencial.



(a) *Boxplot* (b) Teste de Tukey

Figura 53 – *Boxplot* e teste de Tukey de métrica Δ para algoritmos sequenciais e paralelos

A partir dos resultados desse experimento, pode-se supor que a quantidade de indivíduos em cada ilha do NSGA-II paralelo com 4 ilhas é muito pequena para um bom desenvolvimento individual das ilhas. Apesar disso, percebe-se que o hipervolume e métrica Δ do NSGA-II paralelo com 2 ilhas (com $N/2$ indivíduos em cada ilha) é melhor que esses valores para o NSGA-II sequencial com $N/2$ indivíduos (algoritmo 4) enquanto

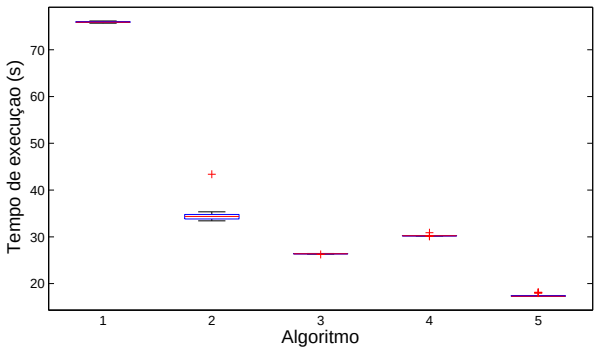
que os mesmos valores para o NSGA-II paralelo com 4 ilhas (com $N/4$ indivíduos em cada ilha) é melhor que para o NSGA-II sequencial com $N/4$ indivíduos (algoritmo 5), revelando que os algoritmos paralelos apresentam desempenho melhores que versões sequenciais com mesma quantidade de indivíduos das ilhas de cada algoritmo paralelo, apesar de um pequeno aumento no tempo de execução dos algoritmos paralelos por causa do gerenciamento das *threads* e o tempo de espera dessas para a sincronização das tarefas sequenciais, como mostrado na Tabela 8, que apresenta o tempo de execução dos cinco algoritmos.

Tabela 8 – Tempo de execução dos algoritmos sequenciais e paralelos

Algoritmo	**Menor tempo (s)**	**Tempo médio (s)**	**Maior tempo (s)**
1	75,6922	75,9203	76,1643
2	33,4330	34,5960	43,3765
3	26,2450	26,3824	26,4516
4	30,2140	30,3110	30,8696
5	17,2973	17,6435	18,0956

Essas diferenças de tempo podem ser melhor observadas no *boxplot* mostrado na figura 54, que mostra também as diferenças estatísticas presentes entre o intervalo de tempo de execução apresentado por cada algoritmo e, por isso, teste de variância não é necessário.

Figura 54 – *Boplot* dos tempos de execução dos algoritmos sequenciais e paralelos



Observa-se que mantendo um mesmo número de indivíduos em cada ilha e aumentando o número de *threads*, aumenta-se também esse *overhead* necessário para o gerenciamento e comunicação dessas *threads*. Como esperado, pode-se perceber também que o algoritmo 1, sequencial com N indivíduos possui o maior tempo de execução, o algoritmo 2, paralelo com 2 *threads*, apresentou tempos de execução maiores que o algoritmo 3, paralelo com 4 *threads*, e o algoritmo 2 possui um tempo de execução pouco maior que o algoritmo 4, sequencial com $N/2$ indivíduos, enquanto que a diferença de tempo entre

os algoritmos 3 e 5, sequencial com $N/4$ indivíduos, mostra-se maior, ou seja, o ganho de desempenho (*speedup*) do algoritmo 2 é consideravelmente maior que o do algoritmo 3.

Também foi desenvolvida outra versão paralela do NSGA-II, que consiste na paralelização dos métodos de ordenação utilizando a GPU. A paralelização da ordenação, que possui alto custo computacional, tem como intenção reduzir o tempo de execução do algoritmo evolutivo sem afetar a qualidade das soluções, como ocorreu com o NSGA-II em modelo de ilha utilizando 4 *threads*. Essa versão paralela produz resultados de mesma qualidade que aqueles gerados pela versão sequencial, já que ela apenas realiza a ordenação, método determinístico, de forma paralela.

O NSGA-II paralelo com GPU (paralelo-GPU 1), como esperado, apresentou tempos de execução menores que aqueles apresentados pelo algoritmo sequencial, como pode-se notar na Tabela 9. Apesar disso, nota-se também que a paralelização com modelos de ilhas apresentou tempos de execução ainda menores que a paralelização da ordenação com GPU, o que se deve ao fato de que outros métodos de alto custo computacional como a ordenação não-dominada de fronteiras, que ainda é executada de forma sequencial, limitam o ganho que se pode obter com essa forma de paralelismo com GPU proposta.

Supondo, então, que a paralelização dos métodos de ordenação com GPU não influencia na qualidade das soluções e selecionando o NSGA-II paralelo com 2 ilhas como melhor algoritmo utilizando modelo de ilhas, já que não apresentou resultados piores que o algoritmo sequencial e possui tempo de execução menor que esse, foi proposta, finalmente a integração desse algoritmo paralelo de 2 ilhas com a paralelização da ordenação com GPU. Esse novo algoritmo paralelo foi denominado algoritmo paralelo-GPU 2 e, após 30 execuções, seu menor tempo, tempo médio e maior tempo foram registrados e apresentados na Tabela 9. Nesse caso, a paralelização da ordenação apresenta uma menor diminuição no tempo em relação ao algoritmo sequencial porque as ilhas possuem números menores de indivíduos, fator do qual depende a complexidade do método de ordenação.

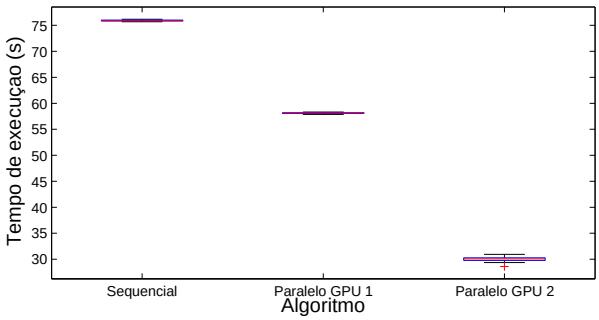
Tabela 9 – Tempo de execução dos algoritmos sequencial e paralelos com GPU

Algoritmo	**Menor tempo (s)**	**Tempo médio (s)**	**Maior tempo (s)**
Sequencial	75,6922	75,9203	76,1643
Paralelo-GPU 1	57,8890	58,1087	58,4087
Paralelo-GPU 2	28,7524	29,7364	31,0712

Novamente, a figura 55, contendo o *boxplot* dos tempos de execução desses algoritmos paralelos com GPU e do algoritmo sequencial, revela intervalos não sobrepostos desses tempos, destacando a grande diferença entre eles, como mencionado, e descartando a necessidade de teste estatístico para essa métrica.

Com os resultados apresentados, pode-se afirmar que neste trabalho foi proposto e

Figura 55 – *Boplot* dos tempos de execução dos algoritmos sequenciais e paralelos com GPU



implementado um algoritmo de otimização capaz de gerar soluções eficientes de carteiras de investimento num tempo computacional baixo (menos de um minuto, no caso deste experimento), compatível com as mudanças e oscilações próprias do mercado financeiro, o que possibilita investimentos de curto prazo, mesmo que este tipo de investimento não tenha sido contemplado neste trabalho. Na prática, um investidor que necessite de soluções confiáveis em tempos ainda menores pode utilizar o arcabouço implementado, bastando ampliar o orçamento computacional para a realização de um processamento paralelo de uma grande massa de dados.

5 Conclusão

Neste trabalho, foram propostos modelos e algoritmos para a otimização de portfólios financeiros com a intenção de revelar a melhor metodologia de investimento ou evidenciar as vantagens e desvantagens de cada uma. Para isso, foram realizados testes em quatro experimentos computacionais, sendo que o primeiro foi responsável por comparar meta-heurísticas e diferentes cardinalidades de ativos em uma otimização *in-sample*, o segundo comparou um modelo discreto com rebalanceamento proposto com modelos presentes na literatura e com um modelo linear aproximado desenvolvido para comparar técnicas exatas e meta-heurísticas, o terceiro comparou ganhos proporcionados pelo modelo com rebalanceamento em relação ao modelo sem rebalanceamento e outros *benchmarks* em simulação realística de investimento durante cinco anos e o quarto comparou meta-heurísticas sequenciais com versões paralelas dessas, verificando se a paralelização de fato diminui seu tempo de execução sem afetar a qualidade das soluções.

O primeiro experimento estudou o efeito de diferentes cardinalidade de portfólios em testes *in-sample*. O melhor desempenho verificado quando utiliza-se cardinalidade 9 corrobora as afirmações de Markowitz (1952), de que diversificar um portfólio com vários ativos reduz seu risco, e de Cesarone, Moretti e Tardella (2016), de que portfólios com não mais de 15 ativos apresentam os melhores desempenhos, apresentando fronteira não-dominada com maior cobertura e espalhamento das soluções.

O primeiro experimento também compara desempenhos de três algoritmos evolutivos multiobjetivos para a otimização inteira de portfólios com restrição de cardinalidade. Hanaoka (2014) utilizou os algoritmos DEMO e PDEA para a otimização desse modelo e alcançou bons resultados em uma simulação, principalmente utilizando o PDEA. Entretanto, neste trabalho verifica-se desempenhos melhores para o SPEA2 e, principalmente, para o NSGA-II em relação aos resultados apresentados pelo PDEA, que apresentou pior espalhamento e cobertura de soluções na média. Assim, esses testes podem ser considerados uma contribuição acadêmica, propondo técnicas possivelmente melhores do que aquela utilizada na literatura para essa modelagem inteira do problema estudado.

O modelo proposto neste trabalho é comparado com um modelo monobjetivo e com um modelo contínuo no segundo experimento. Resultados, apresentando melhor desempenho para o modelo proposto em ambos os experimentos, justificam a utilização deste e contribui academicamente para a área de otimização de portfólios, apresentando um modelo com desempenho aparentemente melhor que outros presentes na literatura. Mostrando, assim, que o modelo considera aspectos que afetam a dinâmica do processo na prática, garantindo melhor desempenho, verifica-se que ele cumpre com o objetivo proposto

para esta pesquisa.

Foram comparadas, no segundo experimento, técnicas exatas e meta-heurística. Resultados mostraram que usando meta-heurísticas pode-se obter resultados quase tão bons quanto aqueles obtidos com métodos exatos em um tempo muito menor, tornando-as adequadas para muitos casos práticos em que um alto tempo de execução é indesejável ou até inviável para o tipo de investimento.

Testes *out-of-sample* compararam o modelo proposto com o modelo de Hanaoka (2014) e, apesar das simulações realizadas não detectarem diferenças estatísticas dos dois modelos para as métricas utilizadas, gráficos de retorno acumulado sugerem que o modelo proposto proporciona maiores ganhos, o que pode ser explicado pela consideração de elementos práticos, como custos de transação, em investimentos e, por isso, pode-se considerar o processo de rebalanceamento como uma contribuição ao modelo discreto de otimização de portfólios. Simulações também revelam que tanto o modelo proposto quanto o de Hanaoka (2014) são capazes de proporcionar lucros acima de "investimentos seguros" como como a taxa SELIC (ou taxa livre de risco) mesmo em momentos de queda no mercado brasileiro.

O terceiro experimento corrobora a hipótese de que o NSGA-II é o algoritmo que apresenta melhor desempenho, apresentando melhores fronteiras eficientes em testes *in-sample* e maiores ganhos financeiros em testes *out-of-sample*. Da mesma forma, o experimento também mostrou o SPEA2 como segundo com maior ganhos nas simulações e o PDEA como algoritmo menos eficiente, proporcionando menores ganhos financeiros.

O tempo de execução do NSGA-II sequencial pode estar acima do tempo desejado por alguns investidores, dependendo do tipo de operação que ele queira realizar. Para solucionar tal problema, a paralelização se mostrou eficiente na redução do tempo de execução da meta-heurística, principalmente utilizando modelo de ilha. Testes de variância não detectaram diferenças no desempenho das soluções geradas em alguns algoritmos paralelos em relação ao algoritmo sequencial, sendo que o NSGA-II paralelo com 2 ilhas e ordenação paralelizada com GPU foi o melhor algoritmo paralelo desenvolvido, pois apresentou menor tempo de execução sem afetar a qualidade das soluções.

O modelo proposto no trabalho mostrou-se mais eficiente que outros presentes na literatura para investimento, na prática, justificando sua importância para a área de finanças computacionais e sua contribuição acadêmica. Em relação às técnicas utilizadas, pode-se concluir que a melhor técnica depende da necessidade do usuário (investidor). Para investimentos de longa duração (meses ou anos), em que a qualidade das soluções é mais relevante que o seu tempo de execução, pode-se utilizar métodos exatos, enquanto que o NSGA-II paralelo é indicado para investimentos de alta frequência (em minutos ou segundos), em que um baixo tempo de execução é desejável ou imprescindível.

Por fim, pode-se concluir que o objetivo geral do trabalho foi concluído, uma vez mostrada a eficiência do modelo proposto e distinguidas as vantagens e desvantagens das técnicas utilizadas, podendo atribuir cada uma às diferentes necessidades de cada investidor por meio do arcabouço eficiente que foi implementado neste trabalho e que, após a criação de uma interface gráfica intuitiva, estará apto para ser utilizado como ferramenta de mercado no auxílio a investidores.

5.1 Trabalhos Futuros

Modelos e técnicas não contemplados no trabalho podem ser abordados em trabalhos futuros. Modelos com três ou mais objetivos, incluindo mais de uma medida de risco, podem ser desenvolvidos e comparados com o modelo biobjetivo proposto nesse trabalho. Para as técnicas, outros algoritmos evolutivos multiobjetivo, como o NSGA-III, MOEA/D (*Multi-objective Evolutionary Algorithm Based on Decomposition*), AEMMT (*Multiobjective Evolutionary Algorithm With Many Tables*) e o AEMMD (Algoritmo Evolutivo Multiobjetivo com Múltiplas Dominâncias), podem ser utilizados ou ainda uma combinação dos algoritmos utilizados no trabalho com outras meta-heurísticas, como o *Simulated Annealing* ou o VNS (*Variable Neighborhood Search*). Outra combinação a ser testada é das meta-heurísticas com métodos exatos para busca local, buscando resultados melhores que as meta-heurísticas com um tempo de execução menor que dos métodos exatos.

Para os testes *in-sample*, a busca por um melhor ajuste de parâmetros como cardinalidade, taxas de mutação e cruzamento podem ser contemplados em trabalhos futuros, uma vez que essa não era a pretensão deste trabalho. Em relação à melhor cardinalidade, um algoritmo paralelo pode executar várias otimizações considerando diferentes valores, buscando o melhor valor para cada série histórica analisada, inclusive utilizando valores maiores que 15.

Nos testes *out-of-sample*, muitas vezes, retornos acumulados mostraram grandes diferenças entre diferentes modelos, enquanto que testes estatísticos falharam em detectar diferenças para os valores mensais apresentados por eles. Percebe-se que, no início, os retornos acumulados dos diferentes modelos tendem a permanecer próximos enquanto que as diferenças tendem a surgir ao final do período considerado. Trabalhos futuros concentrarão nesse final, realizando testes com períodos *out-of-sample* maiores.

Os algoritmos paralelos desenvolvidos tiveram número fixo para gerações de troca de indivíduos entre as ilhas. O aumento ou a diminuição desse número determinam um *tradeoff* entre uma boa qualidade das soluções e um baixo tempo de execução. Quanto maior o número, maior será o tempo de execução, mas melhor será a qualidade das soluções, e o contrário ocorre quando se diminui esse número. Trabalhos futuros podem otimizar esse número de gerações para troca de indivíduos, levando em conta a qualidade

das soluções e o tempo de execução desejados.

Nesse trabalho foi desenvolvido um arcabouço computacional que realiza otimização de portfólios de acordo com parâmetros de entrada definidos pelo usuário. No entanto, esse arcabouço ainda carece de uma interface gráfica para uma comunicação mais rápida e fácil com o usuário. Dessa forma, tal interface pode ser desenvolvida em trabalhos futuros, para que qualquer investidor possa, de forma intuitiva, comunicar-se com o arcabouço, que oferece otimizações para investimentos de curto e longo prazo, conforme a necessidade do usuário.

Apesar de terem sido selecionados, dentre as soluções não-dominadas, a solução de maior razão retorno sobre risco, outros procedimentos podem ser adotados com base em preferências do decisor, valorizando mais o retorno ou o risco. Além disso, trabalhos futuros podem utilizar as preferências desse decisor a priori para guiar a busca da meta-heurística. Por fim, algoritmos e modelos de otimização abordados nesse trabalho podem ser combinados, em trabalhos futuros, com algumas estratégias de investimento, como a inclusão de *stops* ou análise técnica, com a intenção de determinar os melhores momentos para compra e venda de portfólios.

Referências

ANAGNOSTOPOULOS, K. P.; MAMANIS, G. The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of ve multiobjective evolutionary algorithms. In: EXPERT SYSTEMS WITH APPLICATIONS, 2011. [S.l.], 2011. v. 38, n. 11, p. 14208–14217. Citado 3 vezes nas páginas 3, 6 e 9.

ASSAF NETO, A. **Finanças corporativas e valor**. [S.l.]: Atlasl, 2003. Citado na página 1.

AZUMA, R. M. **Otimização multiobjetivo em problema de estoque e roteamento gerenciados pelo fornecedor**. 2011. 121 f. Dissertação (Mestrado em Engenharia Elétrica), 2011. Citado na página 28.

BANCO do Brasil - compra e venda de ações. 2017. Disponível em: <http://www.bb.com.br/portalbb/page83,129,9128,0,1,1,9.bb>. Acesso em: 04 de maio de 2017. Citado na página 68.

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), 2017. Citado 5 vezes nas páginas 7, 61, 63, 68 e 79.

BATHKE, A. The anova f test can still be used in some balanced designs with unequal variances and nonnormal data. **Journal of Statistical Planning and Inference**, Elsevier, v. 126, n. 2, p. 413–422, 2004. Citado na página 69.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear programming - theory and algorithms (2. ed.)**. [S.l.]: Wiley, 1993. ISBN 978-0-471-55793-7. Citado na página 12.

BEASLEY, J. E. Portfolio optimisation: Models and solution approaches. p. 21, 2013. Citado 3 vezes nas páginas 4, 8 e 51.

BENATI, S. Using medians in portfolio optimization. **Journal of the Operational Research Society**, v. 66, n. 5, p. 720–731, 2015. ISSN 1476-9360. Disponível em: <http://dx.doi.org/10.1057/jors.2014.57>. Citado na página 5.

BM&FBOVESPA. **Índice Bovespa**. 2017. <http://www.bmfbovespa.com.br>. Acessado: 2017-02-15. Citado na página 59.

BRUNI, A. L. **Risco, Retorno e Equilíbrio**: uma análise do modelo de precificação de ativos financeiros na avaliação de ações negociadas na bovespa (1988-1996). 1998. 163 f. Dissertação (Mestrado em Administração), 1998. Citado 2 vezes nas páginas 1 e 2.

CÂMARA, J. B. de A. et al. Diversificação entre classes de investimentos como estratégia para minimizar riscos e aumentar a rentabilidade em aplicações financeiras. In: CONGRESSO UFSC DE CONTROLADORIA E FINANÇAS E INICIAÇÃO CIENTÍFICA EM CONTABILIDADE, 2014, Santa Catarina. [S.l.], 2014. v. 5, p. 16. Citado na página 1.

CASTRO, L. de. **Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications**. Taylor & Francis, 2006. (Chapman & Hall/CRC Computer and Information Science Series). ISBN 9781584886433. Disponível em: <https://books.google.com.br/books?id=0gT1zMLEDloC>. Citado 6 vezes nas páginas 11, 15, 16, 17, 19 e 48.

CDI - Certificados de Depósito Interbancário. 2017. Disponível em: <https://www.portalbrasil.net/indices_cdi.htm>. Acesso em: 23 de maio de 2017. Citado na página 85.

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 3 vezes nas páginas 6, 38 e 92.

CHANG, T. .-J. et al. Heuristics for cardinality constrained portfolio optimisation. In: COMPUTERS & OPERATIONS RESEARCH, 2000, Londres. [S.l.], 2000. v. 27, n. 13, p. 1271–1302. Citado 3 vezes nas páginas 2, 3 e 4.

CHENG, R.; GAO, J. On cardinality constrained mean-cvar portfolio optimization. In: CHINESE CONTROL AND DECISION CONFERENCE, 2015. [S.l.]: IEEE, 2015. v. 27, p. 1074–1079. Citado 3 vezes nas páginas 5, 52 e 61.

CHONG, E. K. P.; ŻAK, S. H. **An Introduction to Optimization**. 2. ed. [S.l.]: John Wiley & Sons, INC, 2009. Citado na página 11.

CUDA Toolkit Documentation. 2014. Disponível em: <https://developer.nvidia.com/cuda-toolkit>. Acesso em: 20 de abril de 2017. Citado na página 47.

DEB, K. et al. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: ____. **Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 849–858. ISBN 978-3-540-45356-7. Disponível em: <http://dx.doi.org/10.1007/3-540-45356-3_83>. Citado 4 vezes nas páginas 23, 25, 26 e 27.

DEB, K. et al. Bi-objective portfolio optimization using a customized hybrid nsga-ii procedure. In: ____. **Evolutionary Multi-Criterion Optimization: 6th International Conference, EMO 2011, Ouro Preto, Brazil, April 5-8, 2011. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 358–373. ISBN 978-3-642-19893-9. Disponível em: <http://dx.doi.org/10.1007/978-3-642-19893-9_25>. Citado 5 vezes nas páginas 4, 39, 63, 75 e 78.

DRAWDOWN. 2015. Disponível em: <http://breakingdownfinance.com/finance-topics/risk-management/drawdown>. Acesso em: 04 de maio de 2017. Citado 2 vezes nas páginas 35 e 36.

GUERRERO, J. L. et al. Introducing a robust and efficient stopping criterion for moeas. In: **IEEE Congress on Evolutionary Computation**. [S.l.: s.n.], 2010. p. 1–8. ISSN 1089-778X. Citado na página 53.

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), 2014. Citado 17 vezes nas páginas 5, 8, 9, 50, 51, 52, 56, 63, 65, 67, 68, 79, 80, 82, 83, 92 e 93.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. [S.l.]: The University of Michigan Press, 1975. Citado na página 17.

IBRAHIM, M. A.; EL-BELTAGY, M.; KHORSHID, M. Evolutionary multiobjective optimization for portfolios in emerging markets: Contrasting higher moments and median models. In: ____. **Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I**. Cham: Springer International Publishing, 2016. p. 73–87. ISBN 978-3-319-31204-0. Disponível em: <http://dx.doi.org/10.1007/978-3-319-31204-0_6>. Citado na página 5.

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 3 vezes nas páginas 23, 24 e 25.

LWIN, K.; QU, R.; KENDALL, G. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. **Applied Soft Computing**, v. 24, p. 757 – 772, 2014. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494614003913>. Citado na página 52.

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 29.

MANSINI, R.; OGRYCZAK, W.; SPERANZA, M. G. **Linear and Mixed Integer Programming for Portfolio Optimization**. [S.l.]: Springer International Publishing, 2015. ISBN 978-3-319-18482-1. Citado 13 vezes nas páginas 8, 30, 31, 33, 34, 35, 37, 38, 39, 40, 41, 42 e 64.

MARKOWITZ, H. Portfolio selection. v. 7, n. 1, p. 77–91, 1952. Citado 10 vezes nas páginas 1, 2, 4, 5, 31, 32, 35, 39, 86 e 92.

MARKOWITZ, H. M. **Portfolio Selection**: Efficient diversification of investments. [S.l.]: New York: Wiley, 1959. Citado na página 33.

NVIDIA. **O que é computação com aceleração de GPU?** 2014. Disponível em: <http://www.nvidia.com.br/object/what-is-gpu-computing-br.html>. Acesso em: 20 de abril de 2017. Citado na página 47.

NVIDIA. **Thrust**. 2017. Disponível em: <http://docs.nvidia.com/cuda/thrust/\#axzz4i3gqmHce>. Acesso em: 10 de maio de 2017. Citado na página 66.

OLIVEIRA, M. dos Santos de. **Análise Comparativa entre Medidas de Risco na Otimização Multiobjetivo de Carteira de Ações com Restrição de Cardinalidade**. 2016. 134 f. Dissertação (Mestrado em Modelagem Matemática e Computacional), 2016. Citado 3 vezes nas páginas 6, 9 e 60.

PANTUZA JÚNIOR, G. **Métodos de otimização multiobjetivo e de simulação aplicados ao problema de planejamento operacional de Lavra em Minas a Céu Aberto**. 2011. 103 f. Dissertação (Mestrado em Engenharia Mineral – Escola de Minas), 2011. Citado na página 2.

PORTAL Action: Retornos. 2009. <http://www.portalaction.com.br/series-temporais/51-retornos>. Acessado: 2017-03-18. Citado na página 60.

PRICE, K.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution**: A practical approach to global optimization. 1. ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 2005. Citado na página 20.

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 2 vezes nas páginas 54 e 68.

ROCKAFELLAR, R. T.; URYASEV, S. Optimization of conditional value-at-risk. **Journal of risk**, v. 2, p. 21–42, 2000. Citado 4 vezes nas páginas 1, 3, 5 e 37.

ROMAN, D.; MITRA, G.; DARBY-DOWMAN, K. Mean-risk models using two risk measures: a multi-objective approach. In: QUANTITATIVE FINANCE, 2007. [S.l.], 2007. v. 7, n. 4, p. 443–458. Citado na página 3.

RUIZ-TORRUBIANO, R.; SUÁREZ, A. **A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs**. [S.l.: s.n.], v. 36, 2015. 125–142 p. Citado 2 vezes nas páginas 4 e 8.

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. In: EXPERT SYSTEMS WITH APPLICATIONS, 2009. [S.l.], 2009. v. 36, n. 3, p. 5058–5063. Citado 4 vezes nas páginas 3, 4, 5 e 50.

STORN, R.; PRICE, K. **Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces**. 1995. Citado na página 19.

TALBI, E.-G. **Metaheuristics: From Design to Implementation**. [S.l.]: Wiley Publishing, 2009. ISBN 0470278587, 9780470278581. Citado 14 vezes nas páginas 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 48 e 49.

TANENBAUM, A. **Sistemas operacionais modernos**. Prentice-Hall do Brasil, 2010. ISBN 9788576052371. Disponível em: <https://books.google.com.br/books?id=nDatQwAACAAJ>. Citado 2 vezes nas páginas 43 e 44.

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 27.

WAWRZENIAK, D. **Como Ajustar o Histórico de Preços de Ações?** 2013. <http://blog.bussoladoinvestidor.com.br/ajustar-o-historico-de-precos-de-acoes/>. Acessado: 2017-03-18. Citado na página 59.

WILKINSON, B.; ALLEN, C. **Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers**. Prentice Hall, 2005. (An Alan R. Apt book). ISBN 9780131918658. Disponível em: <https://books.google.co.in/books?id=kzuhQgAACAAJ>. Citado 5 vezes nas páginas 42, 44, 45, 46 e 47.

ZITZLER, E. et al. **SPEA2: Improving the strength Pareto evolutionary algorithm**. [S.l.]: Tik-report, 2001. Citado 2 vezes nas páginas 27 e 29.

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 24.

Apêndices

APÊNDICE A – Ativos Utilizados

Todos os testes realizados até o momento utilizaram, como dados de entrada para o processo de otimização, 53 ativos disponíveis para investimentos. Esses ativos faziam parte do índice Bovespa durante o período de 2010 a 2015 e suas cotações foram retiradas da plataforma Bloomberg. A lista desses ativos e suas respectivas empresas podem ser vistas no Quadro 1.

Quadro 1 – Ativos utilizados

Ativo	Empresa
ABEV3	Ambev S.A.
BBAS3	Banco do Brasil S.A.
BBDC3	Banco do Bradesco S.A.
BBDC4	Banco do Bradesco S.A.
BRAP4	Bradespar S.A.
BRFS3	BRF S.A.
BRKM5	Braskem S.A.
BRML3	BR Malls Participações S.A.
BVMF3	BM&F Bovespa S.A.
CCRO3	CCR S.A.
CESP6	CESP
CIEL3	Cielo S.A.
CMIG4	CEMIG
CPFE3	CPFL Energia S.A.
CPLE6	COPEL
CSAN3	Cosan S.A. Indústria e Comércio
CSNA3	Companhia Siderúrgica Nacional
CTIP3	CETIP S.A.
CYRE3	Cyrela Brazil Realty S.A.
EMBR3	Embraer S.A.
ENBR3	EDP Energias do Brasil S.A.
EQTL3	Equatorial Energia S.A.
ESTC3	Estácio Participações S.A.
FIBR3	Fibria Celulose S.A.
GGBR4	Gerdau S.A.
GOAU4	Metalúrgica Gerdau S.A.

Continua na próxima página

Quadro 9 – Ativos utilizados: continuação

Ativo	**Empresa**
HYPE3	Hypermarcas S.A.
ITSA4	Itaúsa - Investimentos Itaú S.A.
ITUB4	Itaú Unibanco Holding S.A.
JBSS3	JBS S.A.
KROT3	Kroton Educacional S.A.
LAME4	Lojas Americanas S.A.
LREN3	Lojas Renner S.A.
MRFG3	Marfrig Global Foods S.A.
MRVE3	MRV Engenharia e Participações S.A.
MULT3	Multiplan Empreendimentos Imobiliários S.A.
NATU3	Natura Cosméticos S.A.
PCAR4	Companhia Brasileira de Distribuição
PETR3	Petróleo Brasileiro S.A.
PETR4	Petróleo Brasileiro S.A.
RADL3	Raia Drogasil S.A.
RENT3	Localiza Rent a Car S.A.
SANB11	Banco Santander Brasil S.A.
SBSP3	SABESP
SUZB5	Suzano Papel e Celulose S.A.
TBLE3	Engie Brasil Energia S.A.
TIMP3	TIM Participacões S.A.
UGPA3	Ultrapar Participações S.A.
USIM5	Usiminas
VALE3	Vale S.A.
VALE5	Vale S.A.
VIVT4	Telefônica Brasil S.A.
WEGE3	WEG S.A.