

Humberto Cassio Nagato Fonseca

A utilização de computação natural para otimização de portfólios e seleção de carteiras

Vila Velha, ES

2020

Humberto Cassio Nagato Fonseca

A utilização de computação natural para otimização de portfólios e seleção de carteiras

Trabalho de Conclusão de Curso apresentado
ao Curso de Ciência da Computação da Uni-
versidade Vila Velha, como requisito parcial
para obtenção do Grau de Bacharel em Ci-
ência da Computação.

Universidade Vila Velha – UVV
Centro Tecnológico
Curso de Ciência da Computação

Orientador: Prof. D.Sc. Saulo Pereira Ribeiro

Vila Velha, ES
2020

HUMBERTO CASSIO NAGATO FONSECA

A utilização de computação natural para otimização de portfólios e seleção de carteiras

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Vila Velha, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Aprovado em _____ de _____ de 2020

COMISSÃO EXAMINADORA

Prof. D.Sc. Saulo Pereira Ribeiro
Orientador
Universidade Vila Velha

Prof. M.Sc. Erlom Pinheiro
Universidade Vila Velha

Vila Velha, ES
2020

dedico este trabalho a todos os que me ajudaram ao longo desta caminhada.

Agradecimentos

A Deus por ter me dado saúde e força para superar as dificuldades

Aos professores do curso de Ciência da Computação que me forneceram todas as bases necessárias para a realização deste trabalho, agradeço com profunda admiração pelo vosso profissionalismo.

Ao Professor D.Sc. Saulo Pereira Ribeiro, pela orientação, apoio e confiança.

Agradeço a minha mãe Fukuyo Nagato, heroína que me deu apoio, incentivo nas horas difíceis, de desânimo e cansaço.

“Não basta ensinar ao homem uma especialidade, porque se tornará assim uma máquina utilizável e não uma personalidade. É necessário que adquira um sentimento, um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto. ”
(Albert Einstein)

Resumo

Esse estudo busca analisar qual dos dois algoritmos selecionados da computação natural pode contribuir, de forma mais efetiva, para a solução de problemas de otimização de carteira. Nesse sentido a presente pesquisa tem como objetivo geral analisar o desempenho de dois algoritmos da computação natural, o algoritmo genético e o algoritmo de enxame de partículas no problema de otimização de portfólio, e comparar os resultados para obter a solução mais satisfatória para esse problema. A análise foi feita usando uma amostra composta de dados diários de cotações de fechamento que consiste de 5 anos de dados históricos de todas as ações listadas pelo S&P 500 de 08/02/2013 – 07/02/2018. A principal conclusão desta pesquisa foi que o algoritmo de enxame de partículas dentro da metodologia sugerida, obteve resultados superiores ao algoritmo genético. Tal resultado motiva o uso do método sugerido em aplicações reais no mercado financeiro, No entanto, deve-se observar que a pesquisa é baseada em alguns pressupostos necessários, que podem afetar esse resultado em aplicações práticas

Abstract

This study seeks to analyze which of the two selected algorithms of natural computing can contribute, in a more effective way, to the solution of portfolio optimization problems. In this sense, the present research has as general objective to analyze the performance of two algorithms of natural computation, the genetic algorithm and the particle swarm algorithm in the portfolio optimization problem, and to compare the results to obtain the most satisfactory solution to this problem. The analysis was performed using a sample composed of daily closing price data consisting of 5 years of historical data for all stocks listed by the S&P 500 from 02/08/2013 - 07/02/2018. The main conclusion of this research was that the particle swarm algorithm within the suggested methodology, obtained results superior to the genetic algorithm. This result motivates the use of the method suggested in real applications in the financial market. However, it should be noted that the research is based on some necessary assumptions, which can affect this result in practical applications.

A utilização de computação natural para otimização de portfólios e seleção de carteiras

Humberto Cassio Nagato Fonseca

2020

Sumário

	Sumário	13
	Lista de tabelas	14
	Lista de ilustrações	15
1	INTRODUÇÃO	17
1.1	Objetivo Geral	17
1.2	Objetivo Específico	17
1.3	Motivação e justificativa	18
1.4	Estrutura do trabalho	18
2	TRABALHOS RELACIONADOS	19
3	FUNDAMENTAÇÃO TEÓRICA	21
3.1	Investimentos Financeiros	21
3.1.1	Introdução	21
3.1.2	Renda Fixa	21
3.1.3	Renda Variável	22
3.1.4	Selic	22
3.2	Otimização de Portfólio	23
3.2.1	Introdução	23
3.2.2	Otimização	23
3.3	Markowitz Modelo de Média Variança	23

3.3.1	Retorno e risco do portfólio	24
3.4	Computação Natural	24
3.4.1	Introdução	24
3.4.2	Computação evolucionária	25
3.4.3	Inteligência de Enxame	26
3.5	Algoritmo Genético	27
3.5.1	Introdução	27
3.5.2	Representação do Gene	28
3.5.3	População Inicial	28
3.5.4	Seleção	29
3.5.5	Crossover (recombinação)	29
3.5.6	Mutação	31
3.5.7	Elitismos	31
3.5.8	Condição de parada	32
3.6	Algoritmo de Otimização de Enxame de Partículas	32
3.6.1	Introdução	32
3.6.2	Topologia	33
3.6.3	Partículas	33
3.6.4	Princípios Socio-Métricos	34
3.6.5	Função Fitness	34
3.6.6	Equação do Algoritmo	34
3.6.7	Pseudocódigo do PSO	35
4	MODELAGEM	37
4.0.1	Preparação dos dados	37
4.0.2	Modelagem do Problema	37
4.0.3	Código Fonte	38
5	ANALISE DOS ALGORITMOS	39
5.0.1	Analisando Resultados do PSO	39
5.0.2	Analisando Resultados do GA	39
6	CONCLUSÃO	43
6.0.1	Conclusões	43
6.0.2	Trabalhos Futuros	43
	REFERÊNCIAS	45

Lista de tabelas

Tabela 1 – Detalhes dos indivíduos na Figura 1 O indivíduo mais apto é o indivíduo nº 5.	30
--------------------------------------------------------------------------------------------------	----

Lista de ilustrações

Figura 1 – Gráfico 1:Mecanismo da roleta no AG. O melhor indivíduo (nº 5) tem a maior parte da roleta, enquanto o pior indivíduo (nº 4) tem a menor parte.	30
Figura 2 – Duas técnicas populares de crossover no AG: ponto único e ponto duplo. No crossover de ponto único, os cromossomos de duas soluções pai são trocados antes e depois de um único ponto. No crossover de ponto duplo, existem dois pontos de cruzamento e os cromossomos entre os pontos apenas são trocados.	30
Figura 3 – 2 - Topologias do PSO	33
Figura 4 – Variação da melhor proporção de Sharpe com iterações para PSO . . .	40
Figura 5 – Variação da proporção média de Sharpe com iterações para PSO . . .	40
Figura 6 – Variação da melhor proporção de Sharpe com iterações para GA	41

1 Introdução

Um dos problemas mais importantes na área econômico-financeira é a composição de uma carteira (ou portfólio) de ativos.

A primeira contribuição no campo da econômico-financeira foi feita na década de 1950 por Harry Markowitz, que desenvolveu uma teoria para a alocação de ativos financeiros de famílias e empresas, a chamada teoria da escolha do portfólio. Essa teoria analisa como a riqueza pode ser idealmente investida em ativos que diferem em relação ao retorno e risco esperados e, portanto, também como os riscos podem ser reduzidos. (NOBELPRIZE.ORG, 2020)

Segundo, (MANKIW, 2004), um dos dez princípios básicos da economia é que o ser humano sempre optará pela melhor oportunidade para si. Deste modo consideramos que, no geral todo investidor racional busca maior quantidade de retorno possível, dado certo nível de risco que está disposto a assumir. As carteiras que conseguem atingir o maior lucro possível respeitando as condições de risco impostas, são denominadas carteiras eficientes e são obtidas através de modelos de otimização de portfólio.

De acordo com autores como (HAUGEN, 1995) alguns estudos demonstram que os modelos utilizados atualmente na seleção de carteiras não geram carteiras eficientes e por esse motivo têm tido seus resultados questionados nos últimos anos.

A proposta deste trabalho é propor uma alternativa a abordagem tradicional já utilizada, analisando dois algoritmos da computação natural o algoritmo de enxame de partículas e o algoritmo genético.

1.1 Objetivo Geral

O objetivo geral do presente trabalho é realizar uma análise de dois algoritmos da computação natural no problema de otimização de portfólio, e comparar os resultados para obter a solução mais satisfatória para esse problema.

1.2 Objetivo Específico

- Conceituar o problema de seleção de carteiras de investimento (otimização de portfólio)
- Conceituar algoritmo genético
- Conceituar algoritmo de enxame de partículas.
- Analisar e comparar os resultados dos dois modelos

1.3 Motivação e justificativa

Quando há uma redução da Selic, os títulos públicos indexados a ela e as aplicações em Renda Fixa passam a oferecer uma remuneração menor, ou seja os efeitos da mudança da Selic são sentidos por todos os brasileiros, bancos e até investidores estrangeiros.

Discutir sobre a eficiência de um modelo híbrido para a Seleção de Carteiras em um cenário de juros no piso histórico justifica-se pelo fato de que será cada vez mais difícil encontrar bons retornos sem acrescentar um pouco de risco à carteira Assim, o presente trabalho partiu da necessidade de entender quais os resultados da utilização de algoritmos da computação natural no mercado de ações.

1.4 Estrutura do trabalho

Este trabalho divide-se em quatro estruturas principais, sendo elas:

O segundo capítulo foi reservado para os trabalhos relacionados aos temas discutidos.

O terceiro capítulo é voltado para o referencial teórico, onde se encontram todos os conceitos relacionados aos recursos técnicos e teóricos estudados para construir a solução proposta.

O quarto capítulo foi reservado para a modelagem do problema.

O quinto capítulo é voltado para a análise dos resultados dos dois algoritmos utilizados na pesquisa.

E por fim, o sexto capítulo, onde está localizada a conclusão do trabalho, abordando tudo que foi desenvolvido ao longo da pesquisa.

2 Trabalhos Relacionados

Neste capítulo é feita uma análise de todos os trabalhos já feitos aplicando computação natural na economia e o que cada trabalho contribuiu para a resolução do problema proposto por cada um.

Algoritmos meta-heurísticos são um dos melhores algoritmos que vem sendo discutidos recentemente. Existem numerosos estudos sobre o uso desses algoritmos no campo da otimização de portfólio. (BACANIN; TUBA, 2014) utilizaram o algoritmo de colônia de abelha para resolver a otimização do portfólio sob restrições do número de ativos envolvidos em um portfólio. Eles compararam o algoritmo com algoritmos genéticos e de vaga-lume. Os resultados anunciam a eficiência do algoritmo para otimização do portfólio. (KUO; HONG, 2013) apresentou um método de portfólio de investimentos em duas etapas, baseado em técnicas de computação flexível (CM, ou SC do inglês soft computing). O primeiro estágio utiliza a metodologia DEA (Data Envelopment Analysis) para selecionar os fundos mais lucrativos, enquanto o híbrido de algoritmo genético (AG) e otimização de enxame de partículas (PSO) é proposto para realizar a alocação de ativos no segundo estágio. A avaliação dos resultados mostra que o valor do portfólio de Sharpe com base no método proposto é superior ao valor do portfólio com base no AG, PSO e índice de mercado. O método proposto realmente pode ajudar de maneira robusta os investidores a obter ganhos. (NIGAM; AGARWAL, 2013) apresentaram uma comparação entre o algoritmo de colônia de formigas e o algoritmo genético para o fundo de índice. Além disso, no estudo empírico, a colônia de formigas teve um desempenho mais satisfatório do que o algoritmo genético. (DENG; LIN, 2010) usaram o algoritmo de colônia de formigas na bolsa de valores dos EUA, na bolsa de Londres e no Japão, Alemanha e Hong Kong. para resolver os problemas sob restrições cardinais. Os resultados apresentados revelam que o algoritmo de colônia de formigas pode ter um desempenho mais eficiente em comparação ao PSO, especialmente para portfólios com menor risco. (HAQIQI; KAZEMI, 2012) apresentaram uma abordagem sobre a ACO e usaram o conjunto de dados da bolsa de valores de Teerã para mostrar que o algoritmo é adequado para otimização de portfólio, porém de acordo com a diferença entre o portfólio que foi construído pela ACO e o valor ótimo, o método nem sempre é confiável. (DOERNER et al., 2006) escolheram o PACO (Pareto Ant Colony Optimization) e compararam PSA (Pareto Simulated Annealing), NSAG-II (Nondominated Sorting Genetic Algorithm II); além disso, eles resolveram o ACO adicionando um vetor de feromônio a uma função objetiva. Eles apresentaram que o ACO tem desempenho mais eficiente que os outros algoritmos.

3 Fundamentação Teórica

Neste capítulo é feita uma breve introdução a cinco temas centrais neste trabalho: Investimentos financeiros, otimização de portfólio, computação natural, algoritmo genético, algoritmo de otimização de Enxame de partículas. Foram analisadas as definições segundo importantes autores de cada tema bem como alguns conceitos associados à área e os paradigmas presentes nos algoritmos usados no desenvolvimento do trabalho.

3.1 Investimentos Financeiros

3.1.1 Introdução

O investimento financeiro refere-se às despesas com a compra de ativos, direitos ou mercadorias no presente para obter retornos ou benefícios futuros. Em relação ao mercado de investimentos, é possível observar uma série de investimentos com diferentes tipos de riscos e retornos, que vão desde simples aplicações financeiras como poupança e CDB. Em termos de tempo de aplicação, os investimentos podem ser divididos em três categorias. O primeiro é um investimento de curto prazo, ou seja, um investimento que busca um retorno rápido, geralmente com um pequeno retorno em um período de 01 a 24 meses. A classificação intermediária é de médio prazo quando inclui os investimentos que possuem um tempo um pouco menor de busca de retorno do que os de curto prazo, com média de 25 a 60 meses. A última categoria é o investimento de longo prazo, ou seja, o resultado só aparecerá depois de 60 meses, e geralmente trará retornos muito consideráveis aos investidores. Ressalta-se que curto, médio e longo prazos são uma classificação subjetiva, não havendo definição sobre o tema. Os títulos de renda fixa podem ser classificados em pré-fixados que são aqueles em que os juros a serem pagos são acordados no fechamento do negócio e não variam até o vencimento e os pós-fixados quando os juros a serem pagos são balizados através de uma taxa flutuante e, assim, o retorno do investimento só é conhecido no vencimento. As instituições financeiras disponibilizam aos seus clientes produtos e operações estruturados de forma que cada um tenha características próprias, principalmente considerando a análise de riscos e retornos.

3.1.2 Renda Fixa

Conforme explicado por (NETO, 2001), existem diversos títulos de renda fixa que se diferenciam essencialmente na data de vencimento (período de resgate dos títulos) e na natureza do emissor. Títulos de renda fixa são títulos em que ambas as partes usarão taxas de câmbio ou indicadores pré-determinados (neste indicador, a renda acumulada definirá

o valor total do principal mais a renda) para entender o lucro operacional da transação. Os títulos de renda fixa podem ser classificados como pré-fixados, que se referem a títulos de renda fixa que concordam em pagar juros quando a operação for concluída e não sofrerão alterações até o vencimento, bem como os pós-fixados que indicam juros a pagar por uma taxa de juros flutuante, portanto, O retorno do investimento só é conhecido no vencimento.

3.1.3 Renda Variável

A diferença entre o investimento em renda variável e o investimento em renda fixa é que, ao adquirir um ativo, nenhuma parte envolvida no processo sabe o retorno futuro. Os mercados de renda variável mais famosos são as ações, que representam apenas a menor parte do capital de uma empresa. Quem compra ações de uma empresa pretende tornar-se sócio dessa empresa, independentemente do valor do seu investimento. Quando uma empresa deseja captar recursos por meio do mercado de ações, ela realiza uma oferta pública inicial (IPO), que consiste na venda das ações da empresa ao mercado, denominado mercado principal. Após a oferta pública inicial, quem deseja negociar ações da empresa passa a utilizar o mercado secundário, e as forças da oferta e da demanda atuam no mercado secundário. Preços das ações. As ações podem ser divididas em duas categorias, uma são ações ordinárias (ON), que dão aos acionistas o direito de voto nas assembleias da empresa, e a outra são ações preferenciais (PN) que dão aos acionistas uma prioridade na escolha de dividendos. Esses dividendos são os lucros da empresa divididos pelas ações da empresa, que são os lucros da empresa para os acionistas. O investimento em ações apresenta maior risco do que a renda fixa, mas isso não significa que o retorno nessa categoria seja necessariamente maior. Ao comprar ações de uma determinada empresa, o investidor tem que assumir certos riscos, portanto, é recomendável que o investidor tenha conhecimento avançado ou contrate serviços especializados neste mercado.

3.1.4 Selic

O governo federal precisa de recursos para financiar sua dívida, emitindo e vendendo títulos públicos por meio do Tesouro e do BACEN. Esses títulos geram juros pagos aos detentores desses documentos. Isso significa que o governo busca recursos no mercado aberto e oferece prêmios aos compradores dessas dívidas. As negociações de compra e venda de títulos da dívida pública federal interna conduzidas pelo Ministério da Fazenda no mercado de câmbio do Brasil são controladas por um sistema eletrônico de custódia e liquidação (conhecido como Sistema Especial de Compensação e Custódia, SELIC), que administra os negócios em Jogue seu valor neste segmento tempo real Segundo (FORTUNA, 2008), a taxa Over Selic é a taxa de juros referencial do mercado, que regula as operações diárias dos títulos públicos federais, pois o preço médio diário ajusta o preço

unitário (PU) dos títulos públicos todos os dias. representante o Banco Central compra e vende títulos do governo federal ao definir a política monetária. É decidido pela reunião de rotina do COPOM. Devido à natureza das duplicatas e dos títulos do governo, a taxa de juros SELIC é aceita como taxa livre de risco na economia brasileira e é uma referência importante para a formação das taxas de juros de mercado (NETO, 2003).

3.2 Otimização de Portfólio

3.2.1 Introdução

Um dos problemas com os que a maioria dos investidores enfrentam é usar a melhor combinação de risco e retorno para obter a melhor diversificação do portfólio. A diversificação por si só não é capaz de resolver o problema. Alguns investidores têm uma quantidade específica de capital para investir. Alguns deles precisam de suas carteiras para conter ativos específicos. Sempre há uma necessidade de diversificação otimizada. Essa otimização pode se tornar realidade usando métodos diferentes, que podem encontrar a melhor combinação de ativos para atender às metas dos investidores. Em relação a este estudo, ele se baseia na demanda dos investidores em restrições de capital e ativos específicos. O estudo visa alcançar uma fronteira eficiente para um número especificado de ativos em um portfólio e realizar a comparação entre os dois algoritmos trabalhados.

3.2.2 Otimização

Otimização matemática é uma técnica direcionada para escolher o melhor componente possível de uma coleção. Se há uma restrição ou não, podemos selecionar métodos diferentes. Embora encontrar a solução ideal seja muito complicado, modelos heurísticos podem ser usados para encontrar uma solução. A solução encontrada pelos modelos heurísticos podem não ser perfeitos e completamente ideais, mas é suficiente e nos leva ao nosso objetivo. Além disso, modelos heurísticos nos levam à solução em um ritmo mais rápido. No caso de ter grandes conjuntos de dados, podemos usar modelos meta-heurísticos. Esses modelos podem não nos levar globalmente às soluções ideais; no entanto, as soluções são suficientes. A metodologia desses algoritmos é tal que eles encontram algumas amostras; Em seguida, eles examinam essas amostras para encontrar a solução ideal local.

3.3 Markowitz Modelo de Média Variância

Inicialmente, a análise de média variância gerou relativamente pouco interesse; entretanto, após um curto período de tempo, a comunidade financeira adotou a tese. Hoje, modelos financeiros são constantemente reinventados para incorporar novas descobertas baseadas nesses mesmos princípios (FABOZZI et al., 2007). O papel mais importante da teoria

de Markowitz é estabelecer a melhor combinação de risco e retorno para as decisões dos investidores. Markowitz construiu uma abordagem matemática definindo o risco como um critério quantitativo.

3.3.1 Retorno e risco do portfólio

Em um problema de otimização de múltiplos objetivos, as funções de múltiplos objetivos precisam ser otimizados simultaneamente (CHAHARSOOGHI; KERMANI, 2008). Seja N o número de diferentes ativos, r_i o retorno esperado do ativo i ($i = 1, \dots, N$), σ_{ij} representa a covariância entre ativos i e j ($j = 1, \dots, N$), a variável de decisão x_i representa a proporção ($0 \leq x_i \leq 1$) do portfólio investido em ativos i usando esta notação, nós podemos apresentar isso (GOETZMANN et al., 2014):

$$\max R_p = \sum_{n=1}^N r_i x_i \quad (3.1)$$

$$\min \sigma_p = \sqrt{\sum_{n=1}^N \sum_{n=1}^N x_i x_j \sigma_{ij}} \quad (3.2)$$

$$\sum_{n=1}^N x_i = 1 \quad (3.3)$$

A equação (3.1) garante que todo o capital disponível seja investido.

3.4 Computação Natural

3.4.1 Introdução

A computação Natural segundo (BALLARD, 1999), são todos os sistemas de computador que são inspirados ou usam um determinado mecanismo natural ou biológico de processamento de informações. Dentro dessa área encontramos três grandes ramos, com seus respectivos subtópicos, são eles :

- Computação inspirada na natureza
 - Computação evolucionária(ou evolutiva).
 - Inteligência de enxame.
 - Redes neurais Artificiais
 - Sistemas imunológicos artificiais
- Simulação e emulação de fenômenos naturais
 - Vida artificial.

- Geometrias fractais.
- Computação utilizando meios(materiais) naturais
 - Computação quântica
 - Geometrias fractais.

Nos próximos tópicos iremos falar especificamente sobre duas subáreas da computação inspirada na natureza, a computação evolucionária(ou evolutiva) e a inteligência coletiva .

3.4.2 Computação evolucionária

Evolução é um processo de otimização cujo objetivo é melhorar a capacidade de um organismo ou sistema de sobreviver em ambientes de competição e mudança dinâmica.

No contexto da computação evolutiva, o foco está na evolução Biologia, na perspectiva de (LAURENT, 1997) e (DARWIN, 1993). A teoria evolucionária Lamarckiana é baseada na hereditariedade, ou seja, durante o tempo de vida os indivíduos transmitem os seus traços para os descendentes que continuam a se adaptar,o método de adaptação se baseia no conceito de uso e desuso.

A teoria da evolução de Darwin é baseada na seleção natural, onde em um mundo com recursos limitados e uma população estável, Os indivíduos competem com outros pela sobrevivência.Indivíduos com as melhores características ou traços têm maior chances de sobrevivência e reprodução, sendo essas características passadas para seus descendentes,essas características desejáveis são herdadas pelas gerações futuras e se tornam dominantes, ao longo do tempo, dentro da população. Durante a geração de um descendente, eventos aleatórios podem causar mudanças aleatórias nas características do descendente, se forem benéficos, as chances de sobrevivência aumentarão.

Desta forma, podemos definir a computação evolucionária como um sistema para a resolução de problemas, que usam modelos computacionais de processos evolucionários, tais como:

- Seleção natural
- Sobrevivência do mais apto.
- Reprodução

Algoritmo genético (AG) e programação genética (PG) são as duas principais áreas de pesquisa da CE. O algoritmo genético (AG) que iremos analisar nesse trabalho, foi proposto por (JOHN, 1992). Seu objetivo original era estudar fenômenos relacionados à adaptação de espécies e seleção natural ocorrendo na natureza (DARWIN, 1859), e desenvolver um meio que integra-se esses conceitos ao computador (MITCHELL, 1997). Os AGs tem

uma ampla gama de aplicações em muitos campos científicos, incluindo otimização de soluções, aprendizado de máquina, estratégia matemática e desenvolvimento de fórmulas, análise de modelos econômicos, problemas de engenharia e várias aplicações em biologia, sistema imunológico e ecossistema. , A descoberta da forma e propriedades das moléculas orgânicas (MITCHELL, 1997).

3.4.3 Inteligência de Enxame

Segundo (DORIGO; BIRATTARI, 2007), a inteligência de enxame é uma disciplina que envolve sistemas naturais e artificiais. O sistema é composto de muitos indivíduos que se coordenam por meio do controle descentralizado e da auto-organização. Em particular, o assunto enfoca os comportamentos coletivos gerados por interações locais entre os indivíduos e o ambiente circundante. Exemplos de sistemas estudados pela inteligência de enxame são colônias de formigas e cupins, cardumes de peixes, grupos de pássaros e grupos de animais terrestres. Alguns fatores humanos também pertencem ao campo da inteligência de enxame, especialmente alguns sistemas multi-robôs e certos programas de computador escritos para resolver problemas de análise e otimização de dados.

Um sistema de inteligência de enxame típico descrito por (DORIGO; BIRATTARI, 2007), tem as seguintes características:

- Composto por muitos indivíduos;
- Os indivíduos são relativamente homogêneos (ou seja, são todos iguais ou pertencem a certos tipos);
- A interação entre os indivíduos é baseada em regras comportamentais simples, que exploram apenas informações locais que os indivíduos trocam diretamente ou através do ambiente (estigmergia);
- O comportamento geral do sistema resulta da interação dos indivíduos entre si e com seu ambiente, ou seja, o comportamento do grupo é auto-organizado.

A principal propriedade de um sistema de inteligência de enxame é a capacidade de atuar de forma coordenada sem a presença de um coordenador ou de um controlador externo. Muitos exemplos podem ser observados na natureza, de enxames que executam algum comportamento coletivo sem que nenhum indivíduo controle o grupo ou esteja ciente do comportamento geral do mesmo. Apesar da falta de responsáveis pelo grupo, o enxame como um todo pode apresentar um comportamento inteligente. Isso é o resultado da interação de indivíduos espacialmente vizinhos que agem com base em regras simples. Na maioria dos casos, o comportamento de cada indivíduo do enxame é descrito em termos probabilísticos: o comportamento aleatório de cada indivíduo depende da percepção local de seus vizinhos. Por causa das características acima, é possível projetar um sistema de inteligência de enxame que seja escalável, paralelo e tolerante a falhas.

- Escalabilidade significa que o sistema pode aumentar seu tamanho enquanto mantém sua funcionalidade, sem ter que redefinir a forma como seus componentes interagem. Assim como em um sistema de inteligência de enxame, a interação envolve apenas indivíduos adjacentes, e o número de interações muitas vezes não aumenta à medida que o número total de indivíduos no grupo aumenta: o comportamento de cada indivíduo é afetado apenas pela ambiguidade do tamanho do grupo. Em sistemas artificiais, a escalabilidade é interessante porque um sistema escalável pode aumentar o desempenho simplesmente aumentando seu tamanho sem qualquer reprogramação.
- Em um sistema de inteligência de enxame, o paralelismo é possível porque os indivíduos que compõem um enxame podem realizar ações diferentes em locais diferentes ao mesmo tempo. Em um sistema artificial, o paralelismo é desejável porque pode ajudar a tornar o sistema mais flexível, ou seja, ser capaz de se organizar como uma equipe que lida com diferentes aspectos de tarefas complexas ao mesmo tempo.
- A tolerância a falhas é uma propriedade inerente aos sistemas de inteligência de enxame, que é atribuída à descentralização e auto-organização de sua estrutura de controle. Uma vez que o sistema é composto de muitos indivíduos intercambiáveis e eles não são responsáveis por controlar o comportamento geral do sistema, é fácil dissolver o indivíduo com defeito e substituí-lo por um indivíduo totalmente funcional.

3.5 Algoritmo Genético

3.5.1 Introdução

Os algoritmos genéticos são métodos de otimização inspirados pela teoria da evolução de Darwin (DARWIN, 2009). O principal mecanismo simulado na AG é a sobrevivência do mais apto. Na natureza, os organismos mais aptos têm uma maior probabilidade de sobrevivência. Isso os ajuda a transferir seus genes para a próxima geração. Com o tempo, bons genes que permitem que as espécies se adaptem melhor ao meio ambiente (evitam inimigos e encontram comida) se tornam dominantes nas gerações subsequentes (LACERDA; CARVALHO, 1999). Inspirado pelos cromossomos e genes presentes na natureza, o algoritmo AG representa problemas de otimização como um conjunto de variáveis. Cada solução de um problema de otimização corresponde a um cromossomo e cada gene corresponde a uma variável do problema. Por exemplo, para um problema com 10 variáveis, o AG usa cromossomos com 10 genes. O algoritmo AG utiliza três operadores principais para melhorar os cromossomos em cada geração: seleção, cruzamento (recombinação) e mutação. Estes passos como bem como representação de problemas e população inicial são discutidos a seguir nas subseções.

3.5.2 Representação do Gene

Como discutido acima, cada cromossomo corresponde a uma possível solução de um problema de otimização. Um cromossomo é composto por múltiplos genes que simulam as variáveis do problema de otimização. Para usar o algoritmo AG, portanto, o primeiro passo é formular o problema e definir seus parâmetros como um vetor. Lá Segundo (MIRANDA, 2007), são tres variantes do AG: binário, contínuo e real. Na versão binária, cada gene pode ser atribuído com dois valores (por exemplo, 0 ou 1). Na versão contínua, cada restrição é representada como $v_{min} < A_i < v_{max}$. Os valores v_{min} e v_{max} são os limites mínimo e o máximo para o atributo A_i . Segundo (SECCHI, 2004) A versão real utiliza de representações em níveis de abstração mais altos e tem sido investigada no intuito de de comparar o desempenho dos AGs quanto ao alfabeto utilizado (por exemplo (YAO, 1993)) e vem ganhando muita força em problemas mais complexos de otimização paramétrica como estudado em (WRIGHT, 1991) . Para trabalhar com essa versão devem ser criados operadores específicos. Usando essa versão os resultados podem ser mais eficientes do que usando um AG genérico, já que através da codificação e dos operadores são incorporadas informações sobre o espaço de soluções.

Deve-se notar que os genes podem ser até caracteres ou componentes de um programa. Desde que os genes sejam alimentados em uma função de fitness e levem a um valor fitness, eles pode ser usado no algoritmo AG. No caso de usar diferentes partes de um programa de computador para cada gene, o AG é chamado de Programação Genética. Nesse capítulo, a versão binaria e continua da Ga serão usadas e discutidas.

3.5.3 População Inicial

O AG começa o processo de otimização com um conjunto aleatório de cromossomos. No caso do AG binário, a seguinte formula é usada:

$$X_i = \begin{cases} 1 & r_i < 0.5 \\ 0 & \text{caso contrário} \end{cases} \quad (3.4)$$

Onde X_n é o n-ésimo gene e r_i é um número aleatório em $[0, 1]$ gerado separadamente para cada gene. No AG contínuo, a seguinte equação é usada para inicializar aleatoriamente os genes:

$$X_i = (ub_i - lb_i) * r_i + lb_i \quad (3.5)$$

Onde X_n é o n-ésimo gene, r_i é um número aleatório em $[0, 1]$ gerado separadamente para cada gene, ub_i é o limite superior do n-ésimo gene (variável) e lb_i é o limite inferior do n-ésimo -ésimo gene (variável).

3.5.4 Seleção

O operador de seleção da AG simula a seleção natural. Na seleção natural, a chance de sobrevivência é aumentada proporcionalmente ao condicionamento físico. Ser selecionado leva à propagação de seus genes para as gerações subsequentes. Na AG, isso pode ser simulado com uma roleta. A Figura 1 ilustra um exemplo de uma roleta para seis indivíduos. Os detalhes desses indivíduos são apresentados na Tabela 1. A figura e a tabela mostram que os indivíduos com valores mais altos de aptidão têm uma chance maior de serem selecionados pela roleta. A Figura 1 ilustra um exemplo de uma roleta para seis indivíduos. Os detalhes desses indivíduos são apresentados na Tabela 1. A figura e a tabela mostram que os indivíduos com valores mais altos de aptidão têm uma chance maior de serem selecionados pela roleta.

A roleta normaliza e mapeia os valores de condicionamento físico para os valores de probabilidade. Isso significa que os limites inferior e superior da roleta são 0 e 1, respectivamente. Ao gerar um número aleatório nesse intervalo, um dos indivíduos irá ser selecionado. Quanto maior o setor no gráfico π da roleta que um indivíduo ocupa, maior a chance do indivíduo ser selecionado usando esse operador.

Uma questão importante levantada por (MIRJALILI; DONG; LEWIS, 2020) foi por que indivíduos pobres não são descartados. Na natureza, indivíduos menos aptos também podem acasalar e contribuir para a produção das gerações subsequentes. Isso depende de situações ambientais, território e competição. Um indivíduo pobre pode ter genes que produzem um excelente recurso em conjunto com genes de outro indivíduo para combater uma nova mudança no ambiente. Portanto, dar uma pequena chance de seleção a indivíduos pobres mantém uma pequena probabilidade de tirar vantagem de seus bons recursos.

Observe que o intervalo de valores de fitness muda em diferentes problemas, portanto, essas avaliações devem ser normalizadas, como pode ser ilustrado na Tabela 1. Além disso, um dos problemas do uso da roleta é que ela falha ao lidar com valores negativos de fitness. Devido ao cálculo da soma acumulada durante o processo de simulação da roleta, a escala de fitness deve ser feita para alterar todos os valores negativos de adequação para positivos.

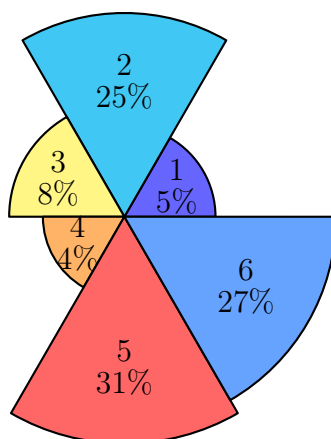
3.5.5 Crossover (recombinação)

A seleção natural apresentada na subseção anterior permite selecionar indivíduos como pais para a etapa de cruzamento. Segundo (SRINIVAS; PATNAIK, 1994) esta etapa permite a troca de genes entre indivíduos para produzir novas soluções. Existem diferentes métodos de cruzamento na literatura. O mais fácil divide os cromossomos em duas (ponto único) ou três partes (ponto duplo). Eles então trocam genes entre dois cromossomos. Um exemplo é visualizado na Figura 2.

Tabela 1 – Detalhes dos indivíduos na Figura 1 O indivíduo mais apto é o indivíduo nº 5.

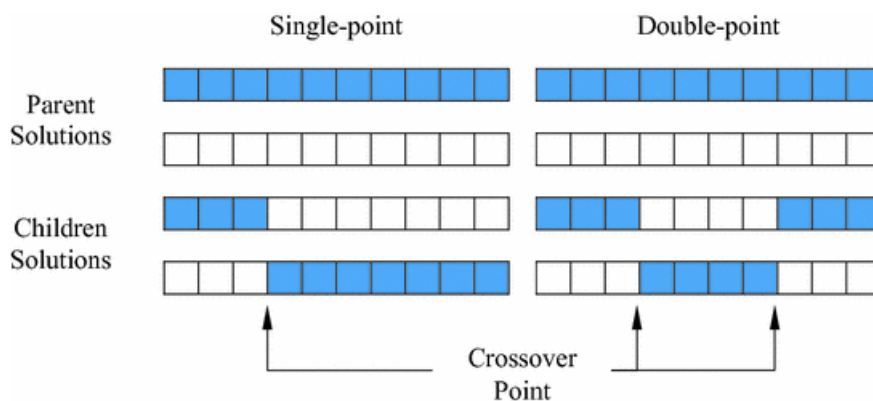
NºIndividuo	Valor Fitness	% do Total
1	12	5
2	55	24
3	20	8
4	10	4
5	70	30
6	60	26
Total	227	100

Figura 1 – Gráfico 1:Mecanismo da roleta no AG. O melhor indivíduo (nº 5) tem a maior parte da roleta, enquanto o pior indivíduo (nº 4) tem a menor parte.



No cruzamento de ponto único, os cromossomos de duas soluções pai são trocados antes e depois de um único ponto. No cruzamento de ponto duplo, no entanto existem dois pontos de cruzamento, e os cromossomos só serão trocados se estiverem entre eles.:

Figura 2 – Duas técnicas populares de crossover no AG: ponto único e ponto duplo. No crossover de ponto único, os cromossomos de duas soluções pai são trocados antes e depois de um único ponto. No crossover de ponto duplo, existem dois pontos de cruzamento e os cromossomos entre os pontos apenas são trocados.



Segundo (GOLDBERG, 1989) O principal objetivo do cruzamento é garantir que os genes sejam trocados e que as crianças herdem os genes dos pais. O crossover é o principal mecanismo de exploração no AG. Para dois pais, se assumirmos que o cruzamento é feito usando um ponto de cruzamento aleatório, o algoritmo está tentando verificar e pesquisar diferentes combinações de genes provenientes dos pais. Isso leva à exploração dessas soluções possíveis sem a introdução de um único gene novo. Observe que existe um parâmetro no AG chamado Probability of Crossover (P_c), que indica a probabilidade de aceitar um novo filho. Este parâmetro é um número no intervalo de $[0,1]$. Um número aleatório no mesmo intervalo é gerado para cada filho. Se esse número aleatório for menor que P_c , o filho será propagado para a geração subsequente. Caso contrário, o pai será propagado. Isso acontece na natureza, assim como todos os filhos não sobrevivem.

3.5.6 Mutação

Como foi discutido por (GOLDBERG, 1989) o operador crossover troca genes entre cromossomos. O problema com esse mecanismo é a falta de introdução de novos genes. Se todas as soluções se tornarem ruins (ficar presas em soluções localmente ideais), o cruzamento não levará a soluções diferentes com novos genes diferentes daqueles dos pais. Para resolver isso, o operador de mutação também é considerado no algoritmo AG.

A mutação causa mudanças aleatórias nos genes. Existe um parâmetro chamado Probabilidade de mutação (P_m) usado para todos os genes de um cromossomo filho produzido no estágio de cruzamento. Este parâmetro é um número no intervalo de $[0,1]$. Um número aleatório no mesmo intervalo é gerado para cada gene no novo filho. Se esse número aleatório for menor que P_m , o gene será atribuído com um número aleatório com os limites inferior e superior.

como citado em (NADA, a) O crossover atua como acelerador do mecanismo de busca e tira proveito das soluções mais promissoras, já a mutação funciona como operador exploratório que dispersa a população pelo espaço de busca evitando os máximos locais.

3.5.7 Elitismos

O cruzamento e a mutação alteram os genes nos cromossomos. Dependendo do P_m , há uma chance de que todos os pais sejam substituídos pelos filhos. Isso pode levar ao problema de perder boas soluções na geração atual. Para consertar isso (RUDOLPH, 1994) sugere usar um novo operador chamado Elitismo. Esse operador não foi inicialmente integrado à primeira versão do GA. No entanto, um grande número de pesquisadores o utilizou recentemente devido ao seu impacto significativo no desempenho da AG.

O mecanismo deste operador funciona de maneira muito simples. Uma porção dos melhores cromossomos da população atual é mantida e propagada para a geração subsequente sem nenhuma alteração. Isso evita que essas soluções sejam danificadas pelo

cruzamento e mutação durante o processo de criação de novas populações. A lista de elites é atualizada, basta reunir os indivíduos com base em seus valores de fitness.

3.5.8 Condição de parada

A situação ótima de um algoritmo de otimização seria que o processo terminasse assim que um ponto ótimo fosse descoberto. Porém, na prática não se pode afirmar que o ponto ótimo gerado pelo algoritmo será o ponto "ótimo-global". Por causa desse fato, geralmente o critério utilizado para a parada é um número máximo de gerações ou um tempo limite de processamento, "o que ocorrer primeiro" (KOZA & RICE, 1994 e TANOMARU, 1995).()

3.6 Algoritmo de Otimização de Enxame de Partículas

3.6.1 Introdução

Técnicas de inteligência de enxame (SI), incluindo otimização de enxame de partículas (PSO) e otimização de colônia de formigas (ACO), usam o comportamento de enxame para resolver o problema e usam o conceito de inteligência de vizinhança junto com a inteligência individual. O algoritmo de otimização de enxame de partículas (PSO) foi proposto inicialmente por (KENNEDY; EBERHART, 1995) para estudar o interessante conceito de um comportamento social (bando de pássaros ou cardume de peixes) de uma maneira simplificada na simulação. No entanto, rapidamente o potencial desta técnica foi reconhecido como uma ferramenta de otimização poderosa que pode ser aplicada com sucesso nos campos de problemas de otimização contínua e discreta, como otimização de função, escalonamento e, portanto, recebeu muitas atenções desde que foi originada. Por trás do PSO está uma teoria muito simples de cognição social. Todos na população têm sua própria experiência e podem avaliar a qualidade dessa experiência. Como os indivíduos são sociais, eles também entendem o comportamento de seus vizinhos. Esses dois tipos de informação correspondem à aprendizagem pessoal (cognitiva) e à transmissão cultural (social), respectivamente. Portanto, a probabilidade de um determinado indivíduo tomar uma decisão dependerá de seu desempenho anterior e do desempenho de certos vizinhos. (EBERHART; SHI; KENNEDY, 2001) usa três princípios para resumir o processo de adaptação cultural

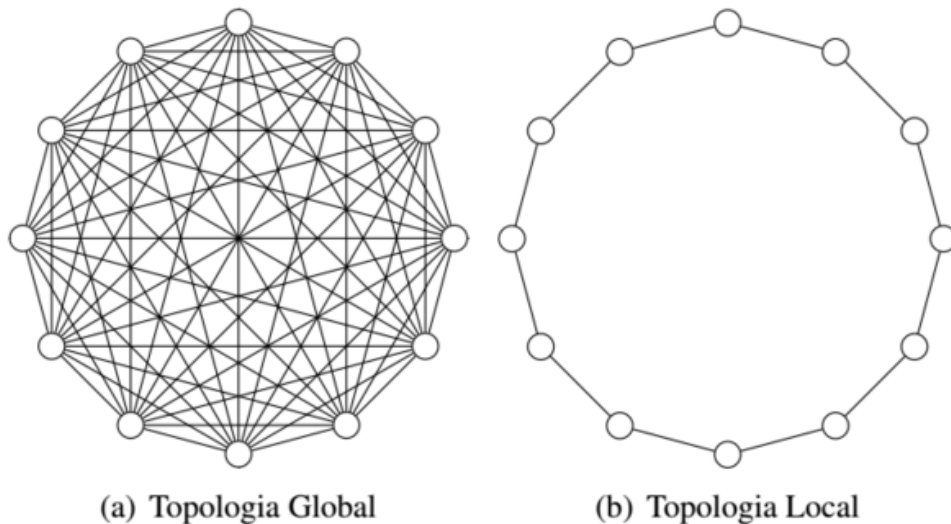
- Avaliar - Os indivíduos têm a capacidade de perceber o ambiente para estimar seu próprio comportamento;
- Comparar - Os indivíduos usam uns aos outros como referência comparativa;
- A imitação é o cerne da organização social humana e é importante para adquirir e manter habilidades psicológicas.

Como outros métodos de inteligência coletiva, o PSO é baseado em grupos de indivíduos que podem se comunicar entre si e interagir com o meio ambiente. Com base na natureza da autoavaliação, comparação e imitação, os indivíduos são capazes de lidar com muitas situações possíveis apresentadas a eles pelo ambiente. Portanto, o comportamento global será o resultado dessas interações.

3.6.2 Topologia

Existem dois métodos básicos para organizar a população. Essas formas são chamadas de topologias de algoritmo, que são: topologia global e topologia local, conforme mostrado na Figura 2a e 2b, respectivamente.

Figura 3 – 2 - Topologias do PSO



Observe que a versão mostrada usa a topologia global, porque todas as partículas são atualizadas da melhor posição global. Dependendo do problema, isso pode ser ruim. Se a função a ser otimizada tem múltiplos máximos e / ou mínimos, pode acontecer que uma partícula caia no mínimo / máximo local e afete todas as outras partículas que vão para lá. Desta forma, o algoritmo convergirá prematuramente, afetando seu desempenho. A topologia local ajuda a evitar esse problema, mas a velocidade de convergência pode ser mais lenta. Como as partículas apenas obtêm a melhor informação de posição de seus vizinhos, há pouco risco de convergirem para a mesma posição ao mesmo tempo.

3.6.3 Partículas

No algoritmo PSO, os indivíduos da população são representados por pontos chamados partículas, que voam no espaço de busca R^d , onde d é a dimensão do espaço. Mudanças nos atributos desses pontos resultarão em novos pontos no espaço, ou seja, eles correspondem

ao movimento no espaço. A ideia inspirada pelos sistemas cognitivos é que essas partículas tenderão a estar próximas umas das outras e influenciar umas às outras.

3.6.4 Princípios Socio-Métricos

A maioria dos algoritmos PSO usa dois princípios de medição social, que representam duas informações importantes no processo de tomada de decisão. O primeiro princípio (g_{Best}) conecta conceitualmente todos os membros da população uns aos outros. Como resultado, o comportamento de cada partícula é afetado pelo comportamento de todas as outras partículas. A segunda métrica (p_{Best}) cria uma vizinhança para cada indivíduo composta por ele mesmo e seus vizinhos mais próximos. Ambas as métricas são medidas pela função de avaliação ($f(p)$), que também é chamada de objetivo ou função de aptidão (fitness), que corresponde à otimização da solução do problema.

3.6.5 Função Fitness

Cada partícula na população do PSO tem um valor de aptidão(fitness). Uma partícula se move no espaço de solução em relação à sua posição anterior, onde encontra o melhor valor de adequação p_{best} , e à posição anterior do vizinho, onde encontrou o melhor valor de adequação g_{best} . Neste trabalho, o Índice de Sharpe é usado como uma única função objetivo. Isso é definido como:

$$IS = (R_i - R_f)/(\sigma i) \quad (3.6)$$

Onde:

- IS = Índice de Sharpe
- R_i = Retorno do Ativo
- R_f = Retorno Livre de Risco (Risk-Free)
- i = Risco do Ativo (a letra grega sigma representa volatilidade)

onde f_p é o valor de adequação da partícula p . A cada etapa, a melhor posição pessoal de uma partícula e o melhor vizinho no enxame são atualizados se uma melhoria em qualquer um dos melhores valores de aptidão for observada.

3.6.6 Equação do Algoritmo

O PSO busca a melhor solução mudando a trajetória dos indivíduos em sua população. Para fazer isso, o algoritmo atualiza a velocidade e a posição de cada partícula. Considere uma população com K partículas. As variáveis v_k e x_k são a velocidade e a posição

da partícula k , respectivamente. A velocidade e a posição de cada partícula devem ser alteradas de acordo com as equações 1 e 2 respectivamente.

Equação Da Velocidade

$$v_{k+1} = (p_{Best} - p_{Atual}) * x_1 + (g_{Best} - p_{Atual}) * x_2 \quad (3.7)$$

Equação Da Posição

$$x_{k+1} = x_k + v_k \quad (3.8)$$

As variáveis v_{k+1} e x_{k+1} serão, respectivamente, a velocidade e a posição atualizada da partícula de acordo com

- p_{best_k} : melhor posição conhecida da partícula
- g_{best} : melhor posição conhecida dentre todas as partículas
- c_1 e c_2 : constantes de aceleração referentes ao melhor individual e global, respectivamente.
- r_1 e r_2 : números aleatórios extraídos do intervalo $[0,1]$

De acordo com a questão discutida, o valor da constante é selecionado com base na experiência. De acordo com os bons valores de (EBERHART; SHI; KENNEDY, 2001), as opções são: c_1 e c_2 são iguais a 2,05 e w é igual a 0,5, mas nada impede a tentativa de outros métodos. As variáveis aleatórias r_1 e r_2 devem ser extraídas da distribuição uniforme e atualizadas com cada cálculo de velocidade geral. Obviamente, a melhor posição pode ser obtida por meio da função de fitness, seja ela individual ou global.

3.6.7 Pseudocódigo do PSO

O algoritmo PSO segue os passos apresentados a seguir:

1. Inicialize a população de K indivíduos com dimensão D . Se o espaço de busca for conhecido, esta inicialização deve ser unificada. Caso contrário, comece aleatoriamente.
2. Estipule os valores das constantes.
3. Verifique se o critério de parada foi alcançado (o critério pode ser um valor predeterminado ou número de iterações). Se sim, complete o algoritmo.
4. Sorteie os números aleatórios para r_1 e r_2

5. Determine a melhor localização global e local.
6. Altere a velocidade das partículas.
7. Altere as posições das partículas.
8. Volte para a etapa 3

4 Modelagem

O trabalho desenvolvido tinha por objetivo realizar uma análise de dois algoritmos da computação natural no problema de otimização de portfólio. Posto isto, e com as definições já introduzidas dos dois algoritmos, neste momento iremos preparar os dados adquiridos e modelar o problema de otimização de portfólio. Todas as análises e constantes utilizadas após a modelagem estarão descritas no capítulo 5

4.0.1 Preparação dos dados

Tanto o PSO e GA foram implementados em python e os dados usados foram baixados do Kaggle. Esse dataset consiste de 5 anos de dados históricos de todas as ações listadas pelo SP 500 de 08/02/2013 – 07/02/2018. Inicialmente esse dataset era composto por 505 ações listadas. Entretanto, as ações que ingressaram no índice após o início do período de 5 anos escolhido não tiveram registros disponíveis para todos os pregões analisados, os registros dessas ações foram excluídos para garantir que o número de registros de todas essas ações no conjunto de dados seja o mesmo. O conjunto de dados final usado para implementação contém registros de 470 ações, e cada ação tem dados de preços de fechamento para todos os 1.259 dias de negociação em um período de 5 anos.

4.0.2 Modelagem do Problema

O problema de seleção de portfólio foi assim resolvido em um espaço de busca de 470 dimensões, no qual PSO e GA tentaram encontrar a combinação ideal de pesos representando o capital alocado para todas as ações na carteira. Cada ponto no espaço de busca representa uma combinação possível de pesos e, portanto, um portfólio possível. Os pesos que representam uma determinada carteira e os dados de retorno diário de todas as ações são usados para calcular o retorno médio, o desvio padrão e, portanto, o índice de ações da carteira. A aplicabilidade da combinação de pesos (representada pela posição da partícula no PSO ou pela posição de um único indivíduo no AG) é avaliada pelo índice de Sharpe do portfólio resultante.

Considerando a fórmula do índice de Sharpe, a taxa de retorno de um título sem risco (R_f) exigida no cálculo do índice foi considerada como sendo a taxa de juros anual do Tesouro dos EUA de 5 anos, que era de 2,57% em 02/07/2018 (CNBC, 2020). A combinação de pesos que maximizou a proporção de Sharpe do portfólio foi considerada a solução mais adequada ou ideal.

4.0.3 Código Fonte

O código construído nesta pesquisa encontra-se em:

<https://github.com/Nagatoh/Tcc-Algoritmos-Computacao-Natural>

5 Análise dos Algoritmos

Este capítulo apresenta os dados coletados através do uso dos dois algoritmos sugeridos neste trabalho: algoritmo de otimização de enxame de partículas (PSO) e o algoritmo genético (GA). Para chegar no resultado descrito na conclusão deste trabalho, foram analisados para cada algoritmo os melhores parâmetros usando o método de tentativa e erro.

5.0.1 Analisando Resultados do PSO

Os melhores resultados para PSO foram alcançados quando o tamanho do enxame foi definido como 500, o número de iterações por tentativa foi definido como 400, formas variantes de tempo de w , c_1 e c_2 foram usadas de modo que $w(0)$ foi definido como 0,9, $w(n)$ foi definido como 0,4, c_1 (min) e c_2 (min) foram definidos como 0,5 e c_1 (max) e c_2 (max) foram definidos como 2,5. A melhor solução alcançada forneceu a proporção de Sharpe do portfólio de 2,237 (consulte a Figura 4). Boa convergência de partículas em torno do valor g_{best} foi observada usando as formas variantes no tempo de w , c_1 e c_2 (consulte a Figura 5). O número de iterações para o resultado ideal foi observado em 250. Resultados ideais alcançados pelo PSO:

- Retorno médio ótimo do portfólio - 38,38
- Risco de portfólio ideal - 16,00
- Razão de Sharpe maximizada - 2,24

5.0.2 Analisando Resultados do GA

Os parâmetros de ajuste do GA são tamanho da população (número de indivíduos na população), número de iterações (gerações) por tentativa, probabilidade de cruzamento pc e probabilidade de mutação pm . O número de genes em cada indivíduo foi fixado em 470, pois o portfólio de 470 ações era desejado. Na implementação do GA, a convergência da população para o indivíduo mais apto foi observada desde o início do processo de ajuste dos parâmetros. Isso foi entendido como devido à implementação de um elitismo simples no algoritmo que garantiu que gerações sucessivas fossem mais adequadas do que as anteriores. No entanto, a população não convergiu para um valor final de aptidão. Os melhores valores para o tamanho da população e o número de iterações por tentativa permaneceram elusivos, pois cada aumento sucessivo em seus valores produziria melhores resultados. Com o tamanho da população definido em 500, o número de iterações por teste definido em 2.000, pc definido em 0,85 e pm definido em 0,025, o indivíduo mais apto representando pesos ideais maximizou a proporção de Sharpe para 1,62 (ver Figura 6).

Figura 4 – Variação da melhor proporção de Sharpe com iterações para PSO

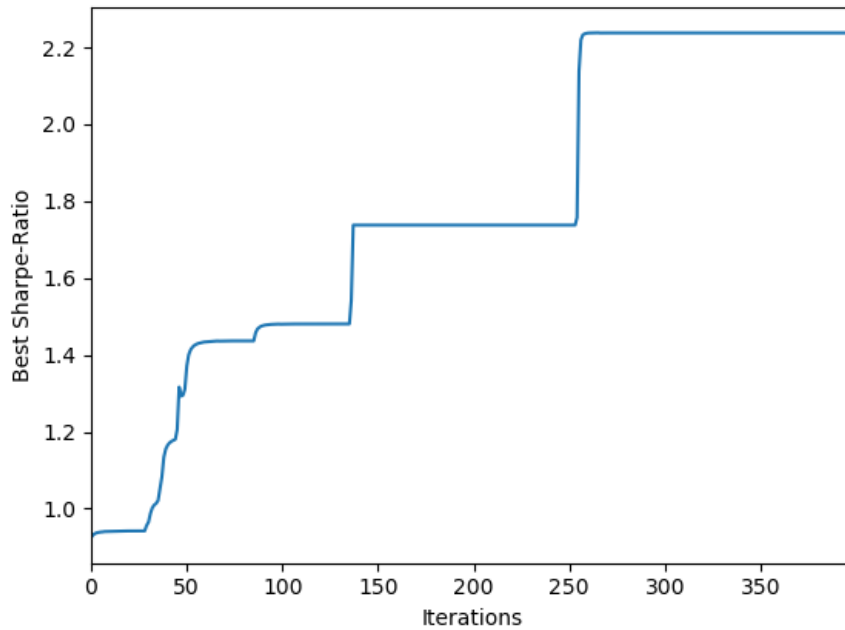
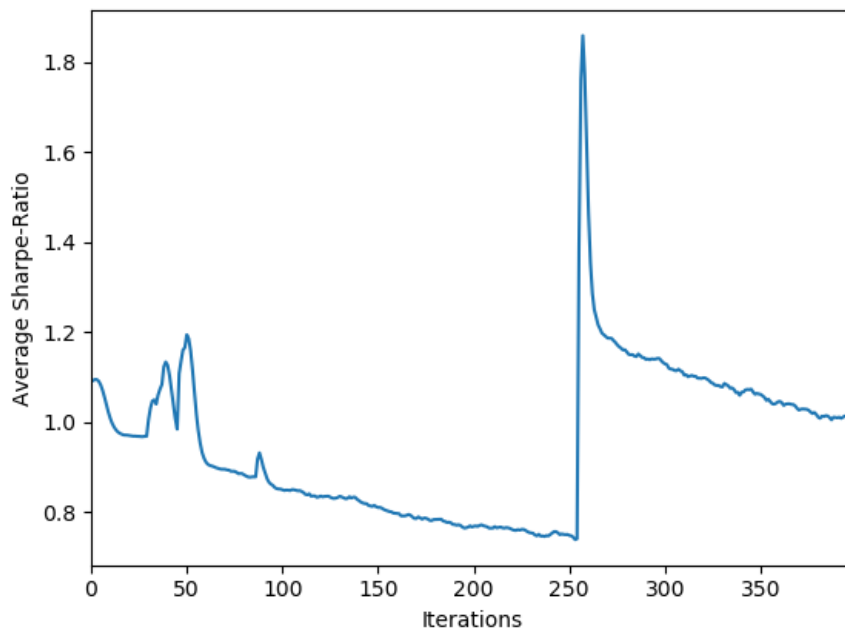


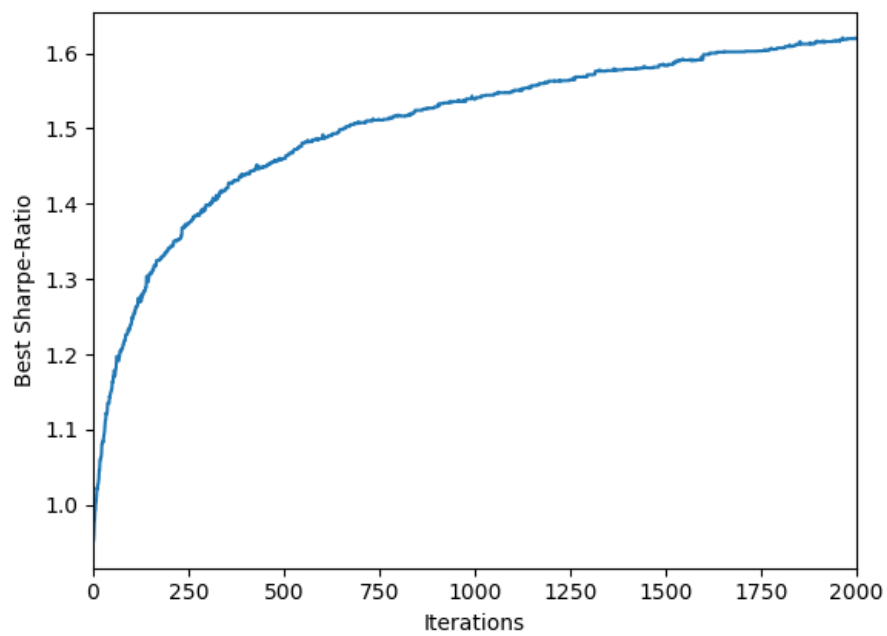
Figura 5 – Variação da proporção média de Sharpe com iterações para PSO



Esses não foram os resultados ideais, pois melhorias adicionais nos resultados poderiam ser alcançadas nas gerações futuras. No entanto, o aumento no número de gerações aumentou o tempo de CPU e os resultados não parecem melhorar significativamente. Os resultados alcançados pelo GA até 2.000 iterações por tentativa são:

- Retorno médio ideal do portfólio - 22,48
- Risco de portfólio ideal - 12,28
- Razão de Sharpe maximizada - 1,62

Figura 6 – Variação da melhor proporção de Sharpe com iterações para GA



6 Conclusão

Neste capítulo concluí-se sobre o cumprimento dos objetivos propostos, comentando as capacidades dos componentes desenvolvidos e também as suas limitações. No final são dadas algumas sugestões para trabalhos futuros.

6.0.1 Conclusões

O objetivo principal da pesquisa apresentada no capítulo 1 era realizar uma análise de dois algoritmos da computação natural no problema de otimização de portfólio, e comparar os resultados para obter a solução mais satisfatória para esse problema.

Embora os parâmetros do PSO fossem difíceis de ajustar para atingir a convergência do enxame, ele alcançou resultados ideais em 250 iterações por teste. Por outro lado, o AG não pôde ser ajustado para atingir a convergência da população em direção ao indivíduo mais apto, mesmo após 2.000 iterações por tentativa. Os resultados obtidos pelo GA após 2.000 iterações não chegaram nem perto dos resultados obtidos pelo PSO após 250 iterações. O tamanho da população em ambos os casos foi ajustado para 500 e as dimensões do espaço de busca foram fixadas em 470. O portfólio ideal encontrado pelo PSO teve proporção de compartilhamento de 2,24 com retorno do portfólio em 38,38% e risco do portfólio em 16,00%. Por outro lado, o GA poderia maximizar a proporção de Sharpe do portfólio para apenas 1,62 após 2.000 iterações.

6.0.2 Trabalhos Futuros

Em termos de trabalho futuro, existem muitas adaptações que podem ser desenvolvidas. Seria interessante adaptar melhor o modelo para situações reais, implementar uma forma de escolher dinamicamente um número fixo de ativos listados no índice S&P 500 em vez de todas as ações do índice em uma única carteira. Isso exigirá restrições sobre o número de ações na carteira no problema de otimização. Muitas outras restrições práticas também podem ser introduzidas, como restrições ao investimento e custos de transação. Além disso, as carteiras de investimento da vida real podem conter ativos não patrimoniais, como títulos de renda fixa, o que exigirá outros conjuntos de dados e funções de aplicabilidade relacionadas para implementar algoritmos de otimização de portfólio.

Referências

CNBC.

BACANIN, N.; TUBA, M. Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint. *The Scientific World Journal*, Hindawi, v. 2014, 2014.

BALLARD, D. H. *An introduction to natural computation*. [S.l.]: MIT press, 1999.

CHAHARSOOGHI, S. K.; KERMANI, A. H. M. An effective ant colony optimization algorithm (aco) for multi-objective resource allocation problem (morap). *Applied mathematics and computation*, Elsevier, v. 200, n. 1, p. 167–177, 2008.

CNBC. *www.cnb.com*. 2020. Last accessed 15 June 2020. Disponível em: <<https://www.cnb.com/quotes/?symbol=US5Y>>.

DARWIN, C. *The origin of species*. 6th. [S.l.]: John Murray, London, 1859. v. 570.

DARWIN, C. *The autobiography of Charles Darwin, 1809-1882: With original omissions restored*. [S.l.]: WW Norton & Company, 1993.

DARWIN, C. *A origem das espécies*. [S.l.]: Lelo & Irmão, 2009.

DENG, G.-F.; LIN, W.-T. Ant colony optimization for markowitz mean-variance portfolio model. In: SPRINGER. *International Conference on Swarm, Evolutionary, and Memetic Computing*. [S.l.], 2010. p. 238–245.

DOERNER, K. F.; GUTJAHR, W. J.; HARTL, R. F.; STRAUSS, C.; STUMMER, C. Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, Elsevier, v. 171, n. 3, p. 830–841, 2006.

DORIGO, M.; BIRATTARI, M. Swarm intelligence. *Scholarpedia*, v. 2, n. 9, p. 1462, 2007. Revision #138640.

EBERHART, R. C.; SHI, Y.; KENNEDY, J. *Swarm intelligence*. [S.l.]: Elsevier, 2001.

FABOZZI, F. J.; KOLM, P. N.; PACHAMANOVA, D. A.; FOCARDI, S. M. *Robust portfolio optimization and management*. [S.l.]: John Wiley & Sons, 2007.

FORTUNA, E. *Mercado financeiro: produtos e serviços*. [S.l.]: Qualitymark Editora Ltda, 2008.

GOETZMANN, W. N.; BROWN, S. J.; GRUBER, M. J.; ELTON, E. J. Modern portfolio theory and investment analysis. *John Wiley & Sons*, v. 237, 2014.

GOLDBERG, D. E. Genetic algorithms in search. *Optimization, and Machine Learning*, Addison Wesley Publishing Co. Inc., 1989.

HAQIQI, K. F.; KAZEMI, T. Ant colony optimization approach to portfolio optimization. In: *Proceedings of the 3rd International Conference on Financial Theory and Engineering*. [S.l.: s.n.], 2012. p. 292–296.

- HAUGEN, R. A. *The new finance: the case against efficient markets*. [S.l.]: Prentice Hall, 1995.
- JOHN, H. Holland. genetic algorithms. *Scientific american*, v. 267, n. 1, p. 44–50, 1992.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. *Proceedings of ICNN'95-International Conference on Neural Networks*. [S.l.], 1995. v. 4, p. 1942–1948.
- KUO, R.; HONG, C. Integration of genetic algorithm and particle swarm optimization for investment portfolio optimization. *Applied mathematics & information sciences*, Natural Sciences Publishing Corp, v. 7, n. 6, p. 2397, 2013.
- LACERDA, E. G. de; CARVALHO, A. D. Introdução aos algoritmos genéticos. *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*, v. 1, p. 99–148, 1999.
- LAURENT, G. *Jean-Baptiste Lamarck, 1744-1829*. [S.l.]: Comité des travaux historiques et scientifiques-CTHS, 1997. v. 119.
- MANKIW, G. Principles of economics, (international student edition). *Thomson South-Western*, 2004.
- MIRANDA, M. N. de. *Algoritmos Genéticos: Fundamentos e Aplicações*. 2007.
- MIRJALILI, S.; DONG, J. S.; LEWIS, A. Ant colony optimizer: theory, literature review, and application in auv path planning. In: *Nature-Inspired Optimizers*. [S.l.]: Springer, 2020. p. 7–21.
- MITCHELL, T. Introduction to machine learning. *Machine Learning*, v. 7, p. 2–5, 1997.
- NETO, A. A. *Mercado financeiro*. [S.l.]: São Paulo: Atlas, 2001.
- NETO, A. A. *Finanças corporativas e valor*. [S.l.]: Atlas, 2003.
- NIGAM, A.; AGARWAL, Y. K. Ant colony optimization for index fund problem. *Journal of Applied Operational Research*, v. 5, n. 3, p. 96–104, 2013.
- NOBELPRIZE.ORG. *Harry M. Markowitz – Prize Lecture*. *NobelPrize.org*. 2020. Last accessed 15 June 2020. Disponível em: <<https://www.nobelprize.org/prizes/economic-sciences/1990/markowitz/lecture/>>.
- RUDOLPH, G. Convergence analysis of canonical genetic algorithms. *IEEE transactions on neural networks*, IEEE, v. 5, n. 1, p. 96–101, 1994.
- SECCHI, A. R. 2004. Last accessed 15 June 2020. Disponível em: <http://w3techs.com/technologies/overview/content_language/all>.
- SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A survey. *computer*, IEEE, v. 27, n. 6, p. 17–26, 1994.
- WRIGHT, A. H. Genetic algorithms for real parameter optimization. In: *Foundations of genetic algorithms*. [S.l.]: Elsevier, 1991. v. 1, p. 205–218.
- YAO, X. Evolutionary artificial neural networks. *International journal of neural systems*, World Scientific, v. 4, n. 03, p. 203–222, 1993.