APLICAÇÃO DE ALGORITMOS DE OTIMIZAÇÃO EVOLUCIONÁRIA PARA TRADING DE CRIPTOMOEDAS

Victor Ferreira Teixeira da Silva
Instituto de Computação – Universidade Federal Fluminense
Niterói, RJ – Brasil
victorfts@id.uff.br

Luiz Satoru Ochi
Instituto de Computação – Universidade Federal Fluminense
Niterói, RJ – Brasil
satoru@ic.uff.br

Igor Machado Coelho
Instituto de Computação – Universidade Federal Fluminense
Niterói, RJ – Brasil
imcoelho@ic.uff.br

RESUMO

Nos mercados financeiros o uso de algoritmos combinados a modelos matemáticos tem se apresentado como um método popular para encontrar padrões e correlações no preço de ativos, permitindo a tomada de decisões sobre quando comprar e vender em uma corretora. Com o advento das criptomoedas como Bitcoin e Ethereum, novas oportunidades foram criadas para aplicar estes modelos, e assim desvendar novas correlações. Por outro lado, novos algoritmos de otimização para busca global chamados de meta-heurísticas tem sido propostos nas últimas décadas, explorando ideias de processos naturais, como a evolução por seleção natural, de forma a resolver rapidamente tais problemas desafiadores. Esse trabalho expande o uso de algoritmos evolutivos em mercados financeiros, proposto por Hirabayashi em 2009, especificamente para criptomoedas e ativos relacionados. Como experimentos, são validadas aplicações da meta-heurística BRKGA, sendo possível descrever um modelo de trade de modo simples, encontrando soluções que melhor se encaixam nas condições de mercado, utilizando dados reais da corretora Binance.

PALAVRAS CHAVE. Trading, Criptomoedas, BRKGA.

TÓPICOS DE INTERESSE: MH – Meta-heurísticas; GF – Gestão Financeira

ABSTRACT

In financial markets, the use of algorithms combined with mathematical models has been a popular method of finding patterns and correlations in asset prices, to make decisions to buy or sell assets on an exchange. With the advent of cryptocurrencies such as Bitcoin and Ethereum, new opportunities were created to apply such models and unveil new correlations.On the other hand, novel global search optimization algorithms called metaheuristics have been developed in past few decades, harnessing ideas from natural processes, such as evolution by natural selection, in order to

https://proceedings.science/p/157643?lang=pt-br

quickly solve these challenging problems. This work expands the use of evolutionary algorithms in financial markets proposed by Hirabayashi in 2009, specifically in cryptocurrency related markets. As experiments, applications of BRKGA are validated, being possible to describe a trading model in a simple manner, finding solutions that best fit the market conditions, using real data from the Binance exchange.

KEYWORDS. Trading, Cryptocurrencies, BRKGA.

TOPICS OF INTEREST: MH – Metaheuristics; GF – Finance Management

1. Introdução

O uso de algoritmos, em suas diversas formas, é cada vez mais popular para executar operações de modo automatizado nos mercados financeiros globalmente [Gsell, 2008]; com sua aplicação no segmento financeiro denominada *algorithmic trading*. Em especial, o mercado de câmbio *(Forex Market)*, analogamente ao mercado de criptoativos, opera todos os dias, 24 horas por dia, tornando o uso de *algorithmic trading* uma proposta atrativa. A operação fora do horário comercial, conhecida como *overnight*, revela diversas oportunidades ainda pouco exploradas, dado seu menor volume de negociações, bem como possíveis lançamentos de notícias e anúncios importantes durante horário noturno, acarretando em possibilidade de ganhos com operação dos ativos envolvidos [Basdekidou et al., 2017].

Com a evolução do mercado de criptoativos, ocorreu uma grande disponibilização de Interfaces de Programação de Aplicações (APIs), as quais possibilitaram a automação na execução de operações para compra e venda de ativos (operações que normalmente são feitas somente dentro de um terminal gráfico com interação humana). Nas corretoras de criptoativos, de modo geral, existem APIs que permitem executar operações de compra e venda, consulta de preços em tempo real, estando disponíveis para qualquer tipo de usuário independente da quantia monetária investida. Como exemplo, podemos citar a corretora Binance[1], possuindo um vasto catálogo de criptoativos em uma API robusta e bem documentada. Nesse sentido, existe um grande contraste na disponibilidade de APIs quando comparamos o mundo dos criptoativos ao mercado tradicional da bolsa de valores, sendo o acesso a APIs na bolsa mais restrito, tipicamente exclusivo a investidores institucionais e indivíduos com grandes quantias aportadas [Martins, 2021].

Um aspecto importante a ser considerado quando aplicamos algoritmos na prática é a explicabilidade das soluções obtidas. Em muitos problemas da atualidade, o uso de Redes Neurais tem sido considerado o estado da arte, embora os modelos obtidos possam sofrer de baixa transparência e levarem a resultados de difícil compreensão [Došilović et al., 2018]. Este trabalho explorará o uso de previsões geradas por indicadores do mercado combinados a operadores da Lógica Proposicional [Büning e Lettmann, 1999], permitindo maior compreensão dos fatores que mais influenciam no processo de decisão. Assim, são consideradas neste trabalho técnicas de otimização evolucionárias capazes de gerar códigos automatizados e legíveis por humanos, fruto de uma área de pesquisa conhecida como Programação Genética [Koza e Poli, 2005].

Técnicas de busca global para problemas de otimização, em especial as meta-heurísticas, tem crescido fortemente nas últimas décadas [Potvin e Gendreau, 2018].Tais técnicas se destacam pela capacidade de escapar de ótimos locais de baixa qualidade, ao mesmo tempo em que proporcionam soluções aproximadas rapidamente para problemas clássicos desafiadores tipicamente de classe NP-Difícil [Garey e Johnson, 1979], onde tipicamente não se conhecem soluções garantidamente exatas de tempo polinomial. Este trabalho propõe uma modelagem para *algorithmic*

[1] https://binance.com

https://proceedings.science/p/157643?lang=pt-br

trading utilizando a meta-heurística *Biased Random Keys Genetic Algorithm* (BRKGA), proposta por Gonçalves e Resende [2011], integrada a indicadores populares de análise técnica [Achelis, 2001], que buscam representar o comportamento dos ativos utilizando exclusivamente informações sobre o preço e volume. A metodologia utilizada para combinar tais indicadores é inspirada na modelagem proposta por Hirabayashi et al. [2009], combinando indicadores que serão utilizados para decisões, após serem otimizados por um algoritmo evolutivo.

Este artigo é organizado em cinco seções, incluindo esta introdução. Na Seção 2 é apresentado o referencial teórico sobre análise técnica no mercado de criptoativos e *backtesting*. Na Seção 3 é apresentada o algoritmo proposto inspirado na meta-heurística BRKGA, bem como a modelagem das soluções para o problema. A Seção 4 detalha experimentos computacionais para validar a proposta utilizando dados reais do mercado. Finalmente, a Seção 5 apresenta conclui o trabalho e discute perspectivas para trabalhos futuros.

2. Referencial Teórico

Esta seção apresenta, de forma breve, conceitos fundamentais de análise técnica, como: indicadores de análise técnica (Subseção 2.1); contratos futuros (Subseção 2.2); representação de preços com *candlesticks* (Subseção 2.3); bem como o conceito de *backtesting*, utilizado neste trabalho para validar algoritmos preditivos (Subseção 2.4). Finalmente, são destacadas as premissas utilizadas durante a otimização de dados históricos (Subseção 2.5).

2.1. Indicadores de Análise Técnica

Os indicadores de análise técnica fornecem um ferramental para quantificar informações sobre o real comportamento observado no mercado [Colby, 2003], produzindo uma base para decisões especulativas, embasadas nos precedentes de preços históricos dos ativos. Existem diversos indicadores fundamentais para análise técnica, dentre os quais destacamos: RSI, SMA, EMA e MACD.

2.1.1. RSI – *Relative Strength Index*

O RSI é um indicador que tem como objetivo medir a tendência no preço do ativo, evidenciando áreas onde o ativo está sobrevendido *(o preço é barato)* ou sobrecomprado *(o preço é caro)*. Como regra geral um valor de RSI menor do que 30 indica sobrevenda e um valor acima de 70 indica sobrecompra; esta informação pode revelar potenciais pontos de entrada na compra de ativos [Hirabayashi et al., 2009]. O RSI é calculado sobre n períodos anteriores, de acordo com a Equação (1).

$$RSI(\%) = \frac{|U|}{|U| + |D|} \tag{1}$$

U: Soma do valor absoluto dos preços de fechamento que apresentaram crescimento de preço nos últimos n períodos.

D: Soma do valor absoluto dos preços de fechamento que apresentaram queda de preço nos últimos n períodos.

2.1.2. SMA – *Simple Moving Average*

A média móvel simples SMA é um dos indicadores mais básicos em análise técnica, tendo objetivo de suavizar a curva de preços avaliando últimos n períodos, onde a curva representa o preço médio do ativo. A distância do preço em relação a curva SMA indica potenciais pontos de compra e venda, marcados através de intersecções com a curva [Colby, 2003].

2.1.3. EMA – *Exponential Moving Average*

A média móvel exponencial EMA é uma variação da média móvel simples, onde valores mais recentes tem uma contribuição exponencialmente maior para o valor atual da curva, reagindo assim mais rapidamente às mudanças de preço. A Equação (2) apresenta o EMA para n períodos anteriores com parâmetro α.

$$EMA_n = \frac{P_n + \alpha \cdot P_{n-1} + \alpha^2 \cdot P_{n-2} + \ldots}{1 + \alpha + \alpha^2 + \ldots} \tag{2}$$

2.1.4. MACD – *Moving Average Convergence Divergence*

O MACD tem como objetivo seguir tendências no preço ao acompanhar duas médias móveis exponenciais EMA. Seu cálculo é realizado subtraindo uma curva de período longo, por exemplo $n = 26$, ao de uma curva de período curto, comumente $n = 12$ períodos. Após o cálculo do MACD, uma nova EMA de $n = 9$ períodos é calculada, sendo esta utilizada para sinais de compra e venda quando existe intersecção entre o valor absoluto do MACD e a média móvel [Colby, 2003].

2.2. Contratos Futuros

Contratos futuros são um tipo de instrumento financeiro que permite *acordar o preço* para uma compra ou venda de um determinado ativo que *será executada no futuro*, permitindo ganhar tanto com a valorização do ativo como também na queda. Deste modo, se um contrato futuro fosse adquirido para compra de 1 Bitcoin no valor de 40.000 USD, e posteriormente este mesmo Bitcoin se valoriza-se para 45.000 USD a diferença de 5000 USD será o lucro do detentor do contrato, o detentor do contrato possui o direito a compra de 1 Bitcoin por um preço menor (40.000 USD) do que atualmente é praticado no mercado (45.000 USD) [Fassas et al., 2020]. Este processo normalmente é amplificado pelo uso de alavancagem fornecido pelas corretoras, permitindo *traders* operarem com uma quantia maior do que possuem disponível em conta. Por exemplo, uma operação que normalmente seria de 1 Bitcoin sem alavancagem pode ser elevada para 20 Bitcoins *(com uma alavancagem de 20x)*, multiplicando consideravelmente os lucros e também as perdas.

2.3. Representando preços com *candlesticks*

Uma das estruturas mais comuns utilizadas para representar dados sobre o preço de ativos financeiros são os *candlesticks* [Nison, 1994]. Os *candlesticks* contém informações sobre: preço de abertura, alta, baixa, fechamento e volume de negociação. Esta estrutura também é conhecida como OHCLV (***O**pen*, ***H**igh*, ***L**ow*, ***C**lose* e ***V**olume*). A Figura 1 apresenta um exemplo de *candlestick*.

Cada *candlestick* representa um período de tempo definido, variando desde alguns minutos ou até mesmo dias, permitindo a análise de ativos com diferentes panoramas com relação ao movimento de seu preço.

2.4. Simulação com dados históricos

A simulação com dados históricos, conhecida como *backtesting*, é uma das técnicas mais comuns para se avaliar algoritmos preditivos [Hirabayashi et al., 2009]. Na prática, é um passo crucial na busca de estratégias de mercado eficazes. Através da simulação, é possível isolar modelos de decisão que são consistentemente lucrativos em comparação ao histórico real. O *backtest* consiste em utilizar os dados históricos para avaliar o desempenho da estratégia e indicadores utilizados, sendo uma ferramenta útil na prática, dado que o comportamento do mercado não varia drasticamente ao longo do tempo [Colby, 2003].

Críticos afirmam que os padrões de comportamento do mercado futuro não correspondem aos do passado [Colby, 2003]. No contexto de automatização de tarefas para *overnight*, uma equivalência perfeita não é necessária. Se um mercado é uma manifestação de psicologia humana, e

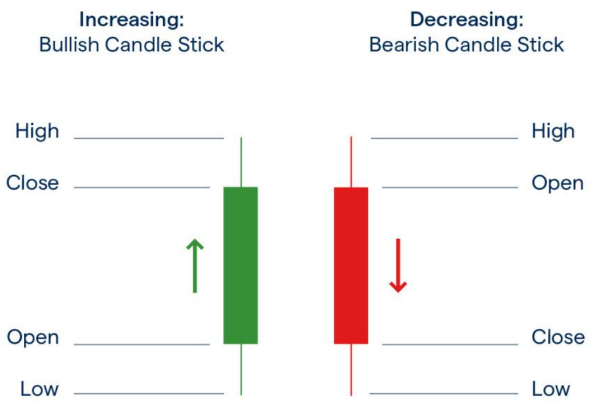


Figura 1: Estrutura de um *candlestick*. [Fonte: investopedia]

este comportamento tem algum padrão ou características repetitivas, então testes históricos podem auxiliar a encontrar esses padrões. De forma geral, o *backtest* nos permite avaliar o desempenho do modelo utilizando dados históricos, todavia sua implementação pode ocorrer de diferentes formas, alterando significativamente os resultados.

2.5. Premissas de otimização

Este trabalho toma como referência o processo de otimização de indicadores proposto por Hirabayashi et al. [2009], do qual destacamos as premissas também contidas na presente proposta:

- **Avaliação pessimista dos preços:** ao utilizar candlesticks para representação de preços existe uma incerteza inerente quanto a real ordem do menor *(low)* e maior preço *(high)*, desta forma o backtest sempre prioriza o resultado mais prejudicial, sendo ele ativar o *stoploss* ou liquidação e não disparar o *takeprofit* casos estes estiverem presentes em um mesmo candlestick.
- **Escorregamento de preços (*slippage*)**: quando operamos no mercado real, o preço visto na tela e o preço executado raramente são os mesmos, com o objetivo de melhor reproduzir este cenário, uma porcentagem slippage foi incluída onde todos os resultados ao executar ou fechar uma operação se movem uma porcentagem *contra o lucro* da operação.
- **Taxas:** a grande maioria das corretoras trabalham com algum tipo de taxa baseada no volume operado, as taxas podem ter grande impacto nas operações, logo toda movimentação tem uma taxa atribuída a mesma.

3. Algoritmo Proposto

A otimização via técnicas heurísticas é tipicamente empregada no contexto de problemas de grande grau de dificuldade, com tempo limitado na tomada de decisão, mesmo que as solução sejam sub-ótimas. Meta-heurísticas são arcabouços heurísticos flexíveis e independentes de problema, geralmente organizadas no formato de metáforas [Sörensen, 2015]. Para o sucesso na aplicação destes métodos, é necessário compreender seus processos e ajustar componentes internos para o problema em questão, como o *Simulated Annealing* [Kirkpatrick et al., 1983], cuja metáfora se baseia no processo de recozimento de metais; e também a família dos algoritmos evolucionários, algoritmos genéticos e programação genética, com analogias no processo de evolução por seleção natural Koza e Poli [2005].

O Algoritmo Genético de Chaves Aleatórias Enviesadas, do inglês *Biased Random Keys Genetic Algorithm* – BRKGA, é um algoritmo da família de algoritmos evolutivos, apresentando conceitos similares ao algoritmo genético clássico [Gonçalves e Resende, 2011]. Porém, o BRKGA pré-define componentes de geração de novos indivíduos, mutação, além de oferecer uma nova estrutura padronizada para representar os indivíduos *solução*, resultando em uma implementação mais objetiva com menos parâmetros configuráveis. Em troca, ele exige a definição de uma função de decodificação para chaves no intervalo $[0, 1]$.

O Algoritmo 1 apresenta o BRKGA. A execução do BRKGA ocorre da seguinte maneira: a população inicial é gerada aleatoriamente até a quantia máxima de indivíduos N, após isto o *fitness* de cada individuo é calculado e a população é ordenada por esta informação; neste passo já possuímos um conjunto inicial de indivíduos (soluções) com desempenho conhecido, isto define a geração inicial. O próximo passo é aprimorar as soluções já existentes para uma próxima geração, sendo dividida em: *fração bot*, *fração top*, *fração crossover*. De acordo com a *fração bot* temos γN novos indivíduos aleatórios gerados (chamados de mutantes) para compor a próxima geração. Seguindo o parâmetro *fração top*, os melhores βN indivíduos são copiados para a próxima geração, agora é necessário gerar $(1 - \beta - \gamma)N$ novos indivíduos baseados nas soluções já existentes para completar a população, sendo este processo é chamado de reprodução ou *crossover*. O *crossover* utiliza genes de pares de indivíduos (pais) para gerar novas soluções (filhos) com preferência pelos melhores pais (por isso surge o termo *biased* ou *enviesado*) até que a população alcance seu limite de indivíduos N, ao fim desta etapa o fitness é calculado e a população ordenada. Com isto executamos todo o processo de geração e aprimoramento de soluções contemplado pelo BRKGA, onde essas etapas (chamadas *gerações*) ocorrem um número pré determinado de vezes $iter_{max}$ (outros critérios de parada podem ser utilizados, como tempo de processamento, qualidade de solução, etc).

Algoritmo 1: BRKGA

Entrada: Máximo Iterações $iter_{max}$, Fitness Alvo f_{max}, Número de indivíduos N, taxa de mutantes γ, taxa de elite β
Saída: Melhor indivíduo s^*

$P \leftarrow GeraAleatorio(N)$;
$CalculaFitness(P)$;
$OrdenarPorFitness(P)$;
$iter \leftarrow 0$;
enquanto $MaiorFitness(P) < f_{max}$ **ou** $iter < iter_{max}$ **faça**
  $P' \leftarrow GeraAleatorio(\gamma \cdot N)$;
  $P' \leftarrow P' \cup ConjuntoElite(P, \beta \cdot N)$;
  $Novos \leftarrow$ Reprodução usando elites e não elites de P ;
  $CalculaFitness(Novos)$;
  $P' \leftarrow P' \cup Novos$;
  $OrdenarPorFitness(P')$;
  $P \leftarrow P'$;
  $iter \leftarrow iter + 1$;
fim
$s^* \leftarrow MelhorIndividuo(P)$;
return s^*;

3.1. Estrutura de chaves aleatórias do indivíduo

Indivíduos no BRKGA são representados como um array de k chaves aleatórias (valores no intervalo $[0, 1]$). Ao utilizar esta representação há uma facilidade em modelar soluções para diversos problemas, uma vez que cada *random-key* pode representar uma infinidade de valores e não apenas 0 ou 1.

3.2. População inicial

A população inicial é sempre composta completamente por mutantes (indivíduos com *random-keys* completamente aleatórias).

3.3. Parâmetros configuráveis

Agora temos apenas 4 parâmetros configuráveis para customizar o comportamento do algoritmo, são estes:

- **Fração Top:** dada pela porcentagem β dos indivíduos considerados de elite *(indivíduos de maior fitness)*.
- **Fração Bot:** dada pela porcentagem γ dos indivíduos que são mutantes *(mutantes são indivíduos com random-keys completamente aleatórias)*.
- **População máxima:** quantidade máxima N de indivíduos que podem estar presentes na população.
- **Taxa de elitismo:** probabilidade α no qual uma chave aleatória de um indivíduo elite será escolhida no processo de *reprodução*.

3.4. Seleção de parentes

Para gerar novos indivíduos por meio de de reprodução é necessário selecionar 2 indivíduos que terão suas *random-key* combinadas para gerar um filho, para estes indivíduos envolvidos na reprodução é necessário que um deles seja **elite** e outro **não-elite**, ambos selecionados aleatoriamente. A combinação de elite e não-elite permite que as características que estão gerando boas soluções sejam propagadas de maneira direcionada para novas soluções geradas.

3.5. Crossover

Após selecionar um parente **elite** e outro **não-elite**, para cada gene, será escolhido um valor do parente elite ou do não elite baseado na probabilidade definida pela taxa de elitismo de modo, tudo sendo avaliado aleatoriamente.

3.6. Representação e Decodificação de Soluções para Previsão em Mercado

Um dos pontos chaves ao trabalhar com BRKGA é como decodificar o array de chaves aleatórias utilizado internamente para um modelo de decisão (de previsão de mercado), sendo o modelo responsável por produzir indicações para abertura e fechamento de posições nos contratos futuros.

A composição do vetor dentro do BRKGA é representa com valores entre 0 e 1, sendo este vetor traduzido em 35 campos. Ambas as regras para abertura de posições *Long* ou *Short* tem parâmetros análogos internamente, porém seus valores são distintos para encapsular seu diferente objetivo. Internamente os parâmetros que são otimizados nestas regras são ilustrados na Figura 2, com um parâmetro adicional descrevendo o fator de alavancagem utilizado (limitado a 60 vezes). Como exemplo prático de um indivíduo decodificado, podemos considerar[2]: $ATR_{entrada} = 6,2$; $ATR_{saida} = 7,3$; $ATR_{periodo} = 15$; $ATR_{periodo} = 30$; $RSI_{min} = 40$; $RSI_{max} = 70$; $RSI_{periodo} = 15$; $MACD_{periodo} = 20$; $MACD_{periodo} = 30$; $MACD_{periodo} = 22$; $EMA_{periodo} = 40$; $EMA_{\%\ diferença} = 2,5\%$; $SMA_{periodo} = 120$; $SMA_{\%\ diferença} = 1,0\%$; $STOPLOSS = 25\%$; $TAKEPROFIT = 50\%$.

[2]Os períodos são medidos em *candlesticks*, não em tempo absoluto.



Figura 2: Composição das regras de long e short

3.7. Função Objetivo e Abertura de Posições

A função de objetivo busca maximizar os ganhos ao longo do tempo durante o *backtest*, onde são efetuadas operações de abertura de posição de acordo com regras pré-estabelecidas. A abertura de posições é definida por uma expressão lógica simples. Em seu trabalho, Hirabayashi et al. [2009] torna os operadores lógicos em parâmetros mapeáveis em seu algoritmo genético, de forma a garantir maior explicabilidade e simplificar o entendimento de como as ativações estavam ocorrendo. Assim, a regra de ativação do Código 1 foi adotada.

Código 1: Função Objetivo

```
macd.signal > macd_target_value &&
rsi > rsi_higher_bound && rsi < rsi_lower_bound &&
pd_EMA_from_close >= EMA_min_percent_diff_from_price||
pd_sma_from_EMA <= sma_max_percentage_diff_from_EMA
```

As regras são formadas por cinco expressões conectadas por operadores **e** e **ou**. Quando a expressão completa é avaliada como verdadeira, uma nova posição será aberta. A operação *não é aberta* com uma avaliação verdadeira, nos seguintes casos:

1. Já existe uma outra operação aberta no momento.

2. Ambas as regras para abertura de posições *Long* e *Short* indicam a abertura de uma posição ao mesmo tempo, (isto denota uma ambiguidade em avaliar a direção do preço para maximizar o lucro), na implementação realizada por Hirabayashi et al. [2009] todos as soluções que apresentam estas ambiguidades são descartadas têm sua execução cancelada e seu fitness anulado.

4. Experimentos Computacionais

O código fonte completo da aplicação está disponível no GitHub[3], podendo ser utilizado e aprimorado de forma livre pela comunidade de otimização e de criptomoedas. A ferramenta proposta **Trade Optimizer** foi escrita na linguagem Rust, com objetivo de simplificar o desenvolvimento sem sacrificar demasiadamente o desempenho (em comparação a outras linguagens compiladas como Fortran, C e C++), sendo naturalmente desejável um baixo tempo de execução. Após a busca por soluções utilizando a heurística BRKGA (durante uma fase de treino), os valores encontrados são decodificados em parâmetros internos dos indicadores de análise técnica, sendo estes: **RSI, MACD, EMA, SMA** e **ATR**, além de outros parâmetros de controle, estas informações são utilizadas para determinar a entrada em posições nos *contratos futuros* de criptoativos.

Os indicadores listados anteriormente foram selecionados por serem alguns dos mais populares no contexto de análise técnica, além de também contemplarem diferentes interpretações sobre o preço, uma hipótese a qual assumimos é que diferentes interpretações tornam a solução

[3]Link: `https://github.com/victorl2/trade-optimizer`

https://proceedings.science/p/157643?lang=pt-br

mais robusta, por fim selecionados apenas estes indicadores para não dificultar análises sobre as soluções geradas, as quais podem trazer novas hipóteses sobre possíveis ajustes na modelagem.

A abordagem de experimentação focou em explorar um ambiente prático, completamente compatível com os sistemas e corretoras do mercado, e pronto para uso por usuários da ferramenta. Para auxiliar a coleta de dados sobre o preço dos criptoativos, um extrator foi criado em Python, buscando os dados de preço no formato candlestick, oriundos da corretora Binance. Atualmente a Binance é uma das maiores corretoras por volume de operações, apresentando mais de 200 ativos disponíveis em sua plataforma.

4.1. Configurações de *hardware*

Os experimentos foram executados em um computador com as seguintes especificações:

- Processador Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 2592 Mhz, 6 Core(s), 12 Núcleos lógicos.
- 16GB de memoria a 3200Mhz de frequência.

4.2. Dados utilizados

Os dados utilizados para execução do *backtest* foram extraídos da Binance, utilizando o próprio extrator da ferramenta proposta, onde analisamos dois principais *datasets*:

- o par ETH/USDT com *candlesticks* no período em 5 minutos.
- o par ETH/USDT com *candlesticks* no período em 30 minutos.

Os dados são referentes aos últimos *dois anos* de preços (2020 e 2021).

4.3. Estrutura do dataset

O dataset possui a estrutura necessária para analisar os preços utilizando *candlesticks*, sendo este o formato fornecido pelas corretoras (veja Tabela 1).

Tabela 1: Estrutura do dataset

OpenTime	Open	High	Low	Close	Volume
1644098400000	3019,37	3035,82	3018,71	3026,99	16551,706
1644100200000	3026,99	3026,99	3014,32	3018,58	8724,268
1644102000000	3018,57	3021,84	3012,28	3015,30	10576,693
1644103800000	3015,29	3022,38	3007,04	3011,59	17777,919

Na Tabela 1, observamos exemplos de tempo de abertura $OpenTime$ (em *timestamp*), bem como valores de $Open$, $High$, Low, $Close$ e $Volume$, cotados em dólar.

4.4. Parâmetros BRKGA

Todos os experimentos realizados utilizaram as seguintes configurações para execução do BRKGA:

- **taxa de mutantes** γ**:** 10%.
- **taxa de elites** β**:** 25%.
- **percentual de elitismo** α**:** 70%.
- **tamanho máximo da população** N**:** 30000 indivíduos
- **número máximo de iterações** $iter_{max}$**:** 1000 iterações

4.5. Parâmetros *backtest*

Os parâmetros utilizados internamente pelo motor de backtest foram:

- **slippage:** 0,5%.
- **taxa de execução:** 0,01%.
- **divisões no dataset:** 12.

As divisões no dataset se referem a períodos distintos que subdividem o dataset, sendo necessários aproximadamente 100 *candlesticks* para inicializar os indicadores internamente. Essas divisões melhor representam a execução real e nos ajuda a utilizar melhor uma parcela dos dados somente para fins de validação.

4.6. Tempos de execução

O tempo médio para executar o *backtest* para um indivíduo foi de **0,012 segundos**; onde a execução completa de uma geração do BRKGA foi de **15,21 segundos**, em média (tempo total aproximado da execução de 4 horas).

4.7. Análise de Resultados

Antes de analisar os resultados é importante perceber algumas diferenças entre os conjuntos de validação e treinamento: o conjunto de treinamento utiliza 10 de um total de 12 partes do *dataset*. Assim, a validação utiliza duas partes, e como a quantia de operações e quais operações podem ser realizadas em cada um dos subconjuntos não é igual, elas servem apenas como um guia para entender qual seria o comportamento do modelo de forma mais próxima ao cenário real (no qual os dados do futuro seriam dinâmicos e incertos).

É importante ressaltar que os dados utilizados para validação não são exclusivamente os mais recentes: uma das parcelas utiliza dados no tempo médio do dataset, permitindo que o treinamento interprete mais corretamente as informações recentes.

A métrica utilizada para representação do fitness é o lucro total na execução do backtest, onde o lucro não inclui o saldo inicial que é definido como 10.000 USD. O cálculo do fitness é dado pela soma dos ganhos como operações subtraído de todos os custos e perdas. Para as operações realizadas no backtest, um valor fixo de 1,5% do saldo disponível no momento é utilizado para cada operação.

Consideramos o par ETH/USDT com *candlesticks* de 5 minutos. Após executar o BRKGA com os parâmetros descritos anteriormente obtivemos os seguintes valores sobre o fitness dos indivíduos (Figura 3).

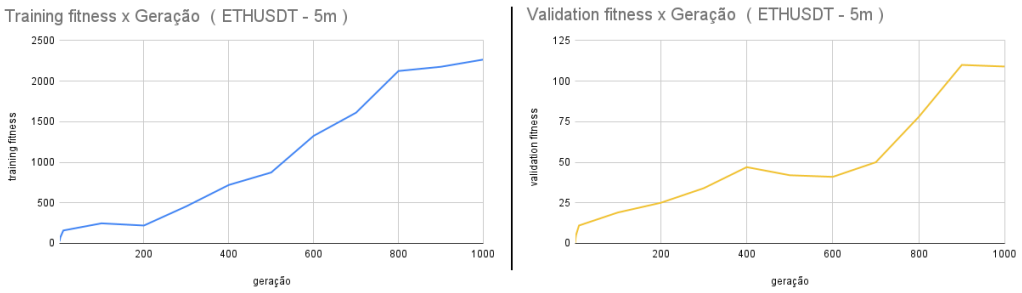


Figura 3: Execução BRKGA no dataset com período de 5 minutos [Fonte: O autor]

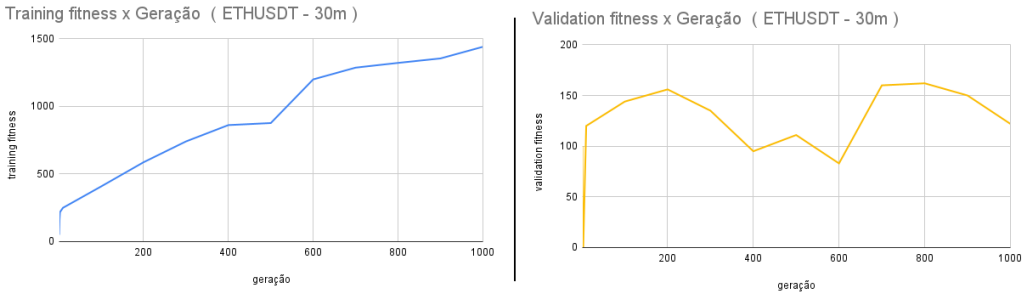


Figura 4: Execução no dataset de treinamento com período de 30 minutos [Fonte: O autor]

Consideramos agora o par ETH/USDT com *candlesticks* de 30 minutos. Analogamente ao executado anteriormente, temos os resultados com candlesticks de 30 minutos (Figura 4).

Observamos nas figuras um comportamento diferente do visto com as candlesticks de 5 e 30 minutos para o mesmo ativo. Analisando dados internos da aplicação, o número de liquidações e fechamentos aparenta ser maior com o período mais curto (5 minutos), o que vai de encontro com as premissas de um *backtest* pessimista, com relação a análise de resultados. Assim, durante os treinos o *backtest* de 5 minutos consegue maiores ganhos (acima de 2000), o que não se reflete na etapa de validação (conforme discutido anteriormente). As oscilações indicam momentos de ganhos e perdas, o que acontece mais acentuadamente no *backtest* de 30 minutos, em ambos casos gerando ganhos positivos superiores a 10% ao final do período de validação.

Observamos que o trabalho de Hirabayashi et al. [2009] não compartilha o código-fonte, bem como parâmetros cruciais para a reimplementação do algoritmo e comparação com a abordagem proposta. Além disso, o presente trabalho utiliza uma integração com dados da Binance e com foco em situações reais, não havendo concorrentes com código disponível e escopo similar para uma comparação direta. Ressaltamos que a calibração do algoritmo se deu de forma empírica com base em parâmetros típicos da literatura, buscando uma convergência com margem de lucro nos testes, e tempo computacional aceitável de treino e validação (horizonte considerado em 4 horas).

5. Conclusões e Trabalhos Futuros

Neste trabalho, abordamos a aplicação de BRKGA para encontrar parâmetros satisfatórios para uma modelagem de *algoritmic trading*, aplicada a contratos futuros de criptoativos, com foco em produzir resultados de fácil interpretação. A implementação apresentou uma prova de conceito que demonstrou capacidade de gerar retornos superiores a 10% ao ano, com dados reais. Não foi realizado refinamento profundo dos parâmetros utilizados para a estratégia, o que permite espaço para muitas melhorias, além de evidenciar a possibilidade de explorar uma infinidade de outros indicadores combinados de novas maneiras. Na abordagem de contratos futuros é possível denominar os aportes e ganhos em criptomoeda caso desejado, deste modo é possível garantir um desempenho implícito por possuir o ativo (conhecida como *buy and hold*), além de uma lucratividade superior operando o ativo.

Observando os resultados, modelagens com maior nível de refinamento prévio no modelo de decisão possivelmente resultariam em maior retorno. Um ponto de atenção, todavia, é a incerteza inerente em utilizar *candlesticks* com período maior, ficando evidente a piora nos resultados quando comparamos a execução no mesmo dataset com período de 5 e 30 minutos. É provável que, com maiores reduções de período, o desempenho será beneficiado, pois reduzir a incerteza irá minimizar o efeito pessimista imposto pelo *backtester*. Também é interessante, em trabalhos futuros, explorar

outras métricas que mensurem a variação do valor do ativo durante o tempo de validação, de forma a minimizar variações bruscas de valor durante a execução.

Referências

Achelis, S. B. (2001). *Technical Analysis from A to Z*. McGraw Hill New York.

Basdekidou, V. A. et al. (2017). The overnight return temporal market anomaly. *International Journal of Economics and Finance*, 9(3):1–10. URL https://pdfs.semanticscholar.org/702c/38db68f547d25ab85be1ce7c2f66edffff7e.pdf.

Büning, H. K. e Lettmann, T. (1999). *Propositional logic: deduction and algorithms*, volume 48. Cambridge University Press.

Colby, R. W. (2003). *The encyclopedia of technical market indicators*. McGraw-Hill.

Došilović, F. K., Brčić, M., e Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, p. 0210–0215. IEEE.

Fassas, A. P., Papadamou, S., e Koulis, A. (2020). Price discovery in bitcoin futures. *Research in International Business and Finance*, 52:101116.

Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability*, volume 174. freeman San Francisco.

Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.

Gsell, M. (2008). Assessing the impact of algorithmic trading on markets: A simulation approach. URL https://www.econstor.eu/bitstream/10419/43250/1/606199632.pdf.

Hirabayashi, A., Aranha, C., e Iba, H. (2009). Optimization of the trading rule in foreign exchange using genetic algorithm. p. 1529–1536. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.516.1317&rep=rep1&type=pdf.

Kirkpatrick, S., Gelatt Jr, C. D., e Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Koza, J. R. e Poli, R. (2005). Genetic programming. In *Search methodologies*, p. 127–164. Springer.

Martins, C. A. G. (2021). Bolsa de valores: plataformas computacionais e robôs. URL https://repositorio.ufgd.edu.br/jspui/handle/prefix/4635. Trabalho de Conclusão de Curso – Universidade Federal da Grande Dourados.

Nison, S. (1994). *Beyond candlesticks: New Japanese charting techniques revealed*, volume 56. John Wiley & Sons.

Potvin, J.-Y. e Gendreau, M. (2018). *Handbook of Metaheuristics*. Springer.

Sörensen, K. (2015). Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18.