

Comparando Algoritmos de Otimização Meta-heurística para Recriação de Imagens Estáticas

Aluno: Antônio Isaac Silva Lima
Departamento de Ciências da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
isaaclima@dcc.ufmg.br

Orientadora: Gisele L. Pappa
Departamento de Ciências da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
glpappa@dcc.ufmg.br

Abstract—Esse estudo investiga a eficiência de três algoritmos meta-heurísticos baseados em população—Algoritmo Genético (GA), Evolução Diferencial (DE) e Otimização por Enxame de Partículas (PSO)—na recriação de imagens estáticas. Os algoritmos foram implementados e testados em quatro pinturas famosas com resoluções variadas. Os resultados indicam que o DE consistentemente supera o GA e o PSO em termos de velocidade de convergência e qualidade da solução. O estudo destaca a robustez e eficiência do DE, sugerindo seu potencial para aplicações mais amplas em processamento de imagens e tarefas de otimização.

Index Terms—Meta-heurísticas, Algoritmo Genético, Evolução Diferencial, Otimização por Enxame de Partículas, Reconstrução de Imagens, Algoritmos de Otimização, Computação Evolucionária, Inteligência Computacional, Programação em Rust, Avaliação de Aptidão

I. INTRODUÇÃO

A maioria dos algoritmos de otimização meta-heurísticos são inspirados na natureza [1], e desde a década de 90, muitos desses algoritmos têm sido aplicados em diversas áreas como otimização, design, agendamento, planejamento e outras. Apesar da eficácia dessas ferramentas, nosso conhecimento sobre a eficiência e a convergência dessas meta-heurísticas ainda é insuficiente para compreender plenamente por que elas funcionam [2].

Além disso, considerando os teoremas de "Sem Almoço Grátis" [3], a performance média de qualquer algoritmo em todas as avaliações possíveis é exatamente a mesma. Isso implica que não existe um algoritmo meta-heurístico que seja superior em todas as situações. Portanto, somente faz sentido comparar os algoritmos meta-heurísticos sob a ótica de um conjunto específico de problemas de otimização ou de um problema particular.

Com isso em mente, esse estudo visa estudar como diferentes algoritmos de otimização meta-heurísticos se comparam em termos de eficiência na recriação de imagens estáticas, considerando suas velocidades de convergência, a qualidade da solução final e a eficiência computacional de cada algoritmo. Essa investigação é relevante tanto para avaliar as capacidades e limitações das abordagens computacionais na geração de arte quanto para aprofundar nosso entendimento sobre a performance desses algoritmos.

II. REFERENCIAL TEÓRICO

As meta-heurísticas podem ser consideradas como um método eficiente para produzir soluções por meio de tentativa e erro para problemas complexos em um tempo razoável. Porém não se há garantia de que as melhores soluções podem ser encontradas e nem que será possível saber se um algoritmo irá funcionar ou porque funcionou [1], ou seja, eles são não-determinísticos.

Os principais componentes de qualquer algoritmo meta-heurístico são a exploração e o aproveitamento (exploration and exploitation) [2]. Exploração refere-se à geração de uma gama diversificada de soluções para explorar amplamente o espaço de busca, enquanto aproveitamento foca a pesquisa em regiões locais do espaço de busca, com base na informação de que uma boa solução pode ser encontrada nessas regiões. Tendo isso em mente, a questão de balancear a busca global (exploração) com a busca local (aproveitamento) é de suma importância para as meta-heurísticas modernas.

Além disso, as meta-heurísticas podem ser divididas entre as que são baseadas em uma única solução e as que são baseadas em uma população de soluções. As meta-heurísticas baseadas em população são especialmente eficazes em evitar ficar presas em ótimos locais [4]. Ou seja, em problemas de otimização nos quais o espaço de busca é rugoso e complexo, é preferível o uso de algoritmos meta-heurísticos baseados em população devido à sua capacidade de explorar múltiplas regiões do espaço de busca simultaneamente.

Dentre as meta-heurísticas baseadas em população, os maiores grupos são os algoritmos evolucionários e os algoritmos baseados em inteligência de enxame. Os algoritmos evolucionários são inspirados em processos evolutivos observados na natureza, como seleção natural e mutação. Por outro lado, os algoritmos de inteligência de enxame são baseados no comportamento coletivo de animais na natureza, como pássaros ou vaga-lumes. Para este estudo, foram escolhidos algoritmos de ambos esses grupos.

III. METODOLOGIA

A seguir, serão delineadas as fases da abordagem metodológica empregada neste estudo com o intuito de atingir os objetivos estabelecidos. Esta seção detalha a implementação

dos algoritmos, a configuração dos experimentos e as métricas de avaliação utilizadas.

A. Implementação dos Algoritmos

Os três algoritmos escolhidos foram: GA (Algoritmo Genético), inspirado na teoria da evolução, utilizando operadores de seleção, cruzamento e mutação; DE (Evolução Diferencial), que utiliza operadores de recombinação e mutação específicos para otimizar problemas contínuos, sendo o único dos três que não é diretamente inspirado na natureza [5]; e, por fim, PSO (Otimização por Enxame de Partículas), baseado no comportamento coletivo de pássaros ou peixes.

Cada algoritmo possui configurações específicas que serão discutidas em III-B. Além dessas, eles também possuem configurações gerais compartilhadas entre os algoritmos: o número de indivíduos na população e o número de triângulos. O primeiro refere-se à quantidade de soluções que serão usadas para explorar o espaço de busca, enquanto o segundo refere-se à quantidade de triângulos semitransparentes que cada algoritmo pode usar para reconstruir a imagem estática alvo.

Cada indivíduo da população é formado por um número n de triângulos, e todas as operações realizadas resultam em mover os vértices dos triângulos ou modificar a cor dos triângulos. No entanto, não é permitido modificar o valor alfa das cores, ou seja, os triângulos permanecem semitransparentes na mesma medida. Além disso, não é possível adicionar ou remover triângulos durante a execução do algoritmo; a quantidade inicial de triângulos é fixa.

Para a geração de imagens a partir de triângulos, foi implementado o algoritmo de rasterização de triângulos utilizando interpolação baricêntrica [6] que manipula diretamente os pixels. Esta escolha foi crucial para garantir a eficiência do algoritmo, visto que outras soluções prontas se mostraram lentas demais para permitir a convergência das meta-heurísticas em tempo hábil.

A métrica usada para determinar a *fitness* (aptidão) dos algoritmos foi o MSE (Erro Quadrático Médio), calculado medindo a distância entre cada pixel da imagem original e os pixels da imagem gerada pelo algoritmo de rasterização de triângulos. Por isso, valores menores de *fitness* representam um resultado melhor.

Por fim, todas as meta-heurísticas foram implementadas em Rust, visando obter melhores resultados com os mesmos recursos computacionais. Rust foi escolhida por sua capacidade de alcançar velocidades comparáveis às linguagens de baixo nível, como C e C++ [7].

1) *Detalhes do PSO*: A fim de conseguir melhores resultados para o PSO, cuja versão comum não estava conseguindo lidar bem com o espaço de busca enrugado e complexo desse problema, foram feitas duas mudanças: a implementação de uma vizinhança que não é global e a hibridização do PSO com o DE.

A vizinhança escolhida inicialmente, a vizinhança global, na qual todas as partículas possuem informações de todas as partículas, não estava conseguindo convergir apropriadamente. Para melhorar os resultados, foi implementada a *Singly-Linked*

Ring Topology, que ao conectar cada partícula a apenas duas outras de forma assimétrica, reduz a velocidade da transferência de informações entre as partículas sobre o espaço de busca, o que aumenta a exploração [8].

E, para também auxiliar na convergência, foi feita uma hibridização em rodadas com o algoritmo DE. O que ocorre é uma transferência da população entre os dois algoritmos, na qual o DE é responsável pela população por 50 iterações e então o PSO fica responsável pela população por 150 iterações.

B. Configuração dos Experimentos

Foram escolhidas quatro imagens de pinturas famosas para a realização dos testes, especificamente: Moça com Brinco de Pérola, O Grito, Mona Lisa e Noite Estrelada. Cada pintura foi redimensionada para resoluções específicas: 50x59 px, 100x127 px, 168x250 px e 500x396 px, respectivamente. A diferença de resoluções é importante pois quanto maior a resolução, maior é o espaço de busca que os algoritmos devem procurar e mais tempo é gasto na rasterização de triângulos para a criação das possíveis imagens de solução.

Para cada imagem e algoritmo, foram feitas 20 execuções. As execuções se encaixam em dois grupos: as de tempo fixado e as de alvo fixado. Nas execuções de tempo fixo, cada algoritmo teve 30 segundos para alcançar o melhor *fitness* possível. Nas execuções de alvo fixo, cada algoritmo recebeu um alvo de *fitness* correspondente com as resoluções das imagens. Os alvos de *fitness* foram de 500, 600, 700 e 1400 para as pinturas, partindo da de menor resolução até a de maior resolução, respectivamente.

1) *Parâmetros Gerais*: Para as três meta-heurísticas, foi definido um tamanho de população igual a 40 e, independentemente da resolução da imagem alvo, os algoritmos receberam a quantidade fixa de 200 triângulos.

Além do mais, cada meta-heurística possui parâmetros particulares que não são compartilhados, e todos os testes foram feitos com o mesmos parâmetros. A fim de determinar o valor ótimo de cada parâmetro, foi utilizado uma busca por força bruta dentro do intervalo onde os valores desses parâmetros são comumente esperados na literatura. Mas a busca pelos melhores valores dos parâmetros não é contemplada nesse estudo.

2) *Parâmetros do Algoritmo Genético*: Como o GA se baseia em seleção, cruzamento e mutação, a única configuração restante foi a definição da taxa de mutação, a qual foi definida como 0.0003, ou 0.03%. Esse valor pode parecer pequeno, mas como há uma quantidade muito grande de triângulos para cada indivíduo, a chance de ocorrer uma mutação em cada iteração é alta.

3) *Parâmetros da Evolução Diferencial*: Para o DE há dois parâmetros importantes: o fator de escala, que foi definido como 1.7 e a probabilidade de cruzamento, que foi definida como 0.03, ou 3%.

O fator de escala controla a amplitude das diferenças vetoriais utilizadas na mutação. Então valores altos aumentam a diversidade das soluções, aumentando a exploração. Já a probabilidade de cruzamento determina a taxa com que os

componentes do vetor mutado substituem os componentes do vetor-alvo. De forma semelhante ao fator de escala, valores altos para a probabilidade de cruzamento incentivam a exploração.

4) Parâmetros da Otimização por Enxame de Partículas:

O PSO possui quatro parâmetros: o peso de inércia, definido como 0.9, ou 90%; o coeficiente cognitivo, definido como 1.3; o coeficiente social, definido como 1.9; e o coeficiente de velocidade máxima, definido como 0.7, ou 70% do valor da altura da resolução da imagem alvo.

O peso de inércia controla a influência da velocidade anterior da partícula na atualização de sua velocidade atual. De forma semelhante, o coeficiente de velocidade máxima limita a velocidade máxima que uma partícula pode atingir. O coeficiente cognitivo representa a tendência da partícula de retornar à sua melhor posição conhecida e, por último, o coeficiente social representa a tendência da partícula de se mover em direção à melhor posição conhecida pelo enxame.

Como esse algoritmo é hibridizado com o DE, é importante mencionar que as partes onde a hibridização ocorrem recebem os mesmos parâmetros definidos em III-B3.

C. Métricas de Avaliação

Para os experimentos de alvo fixo, a métrica principal de avaliação é o tempo decorrido em segundos para atingir a meta. Já para os experimentos de tempo fixo, a métrica principal é a *fitness* atingida.

IV. RESULTADOS

Esta seção apresenta os resultados obtidos a partir da condução metódica dos processos descritos em III. Para a construção dos gráficos foi utilizada a mediana de cada tipo de execução.

A. Análise com Tempo Fixado

Pelos gráficos em 1 pode-se perceber que em todos, com exceção de 1d, o algoritmo DE conseguiu alcançar uma *fitness* melhor durante o mesmo tempo.

B. Análise com Alvo Fixado

Para o conjunto de gráficos em 2, é nítido que o algoritmo DE é capaz de superar os outros dois, conseguindo evitar uma tendência que é pode ser vista tanto no GA, quanto no PSO: de convergir muito rápido para um local ótimo e não explorar o suficiente para conseguir um resultado satisfatório. Esse comportamento pode ser melhor percebido em IV-C.

C. Análise da Distribuição de Fitness

No gráfico em 3, a primeira coisa que é perceptível é o maior tamanho do algoritmo de PSO; esse é um indicativo da capacidade exploratória do algoritmo, mas também revela que ele possui problemas de convergência, mesmo depois da hibridização dele com o DE.

Comparado com o PSO, o GA e DE conseguem achar explorar rapidamente o espaço de busca e chegar em soluções melhores de formas mais rápida.

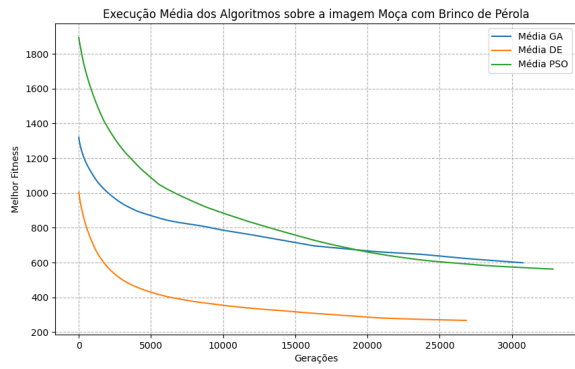
D. Exemplos de Imagens Geradas

Para a geração das imagens em 4, 5, 6 e 7 cada algoritmo recebeu 10 minutos de execução.

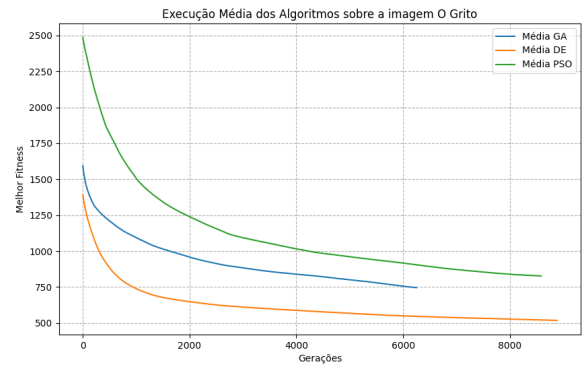
Para o tempo fornecido, é fácil de perceber que quanto maior a resolução, pior fica o resultado, porém o algoritmo consegue ter excelência em baixas resoluções, sendo artisticamente interessante. Além disso, é possível perceber outros padrões, como o GA geralmente utilizar mais cores e o PSO utilizar triângulos maiores. Esses artefatos observados nas imagens de exemplos são provavelmente advindos do processo de otimização de cada meta-heurística.



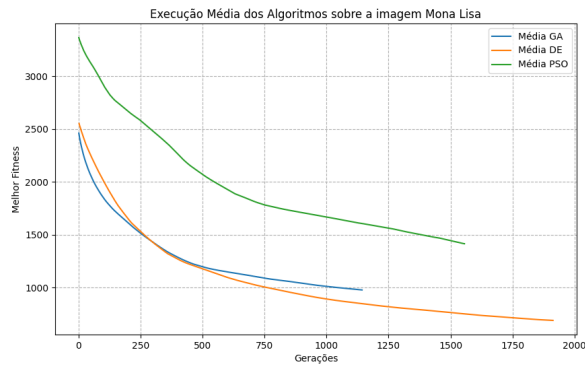
Fig. 4: Conjunto de Imagens com Resolução 50x59 px



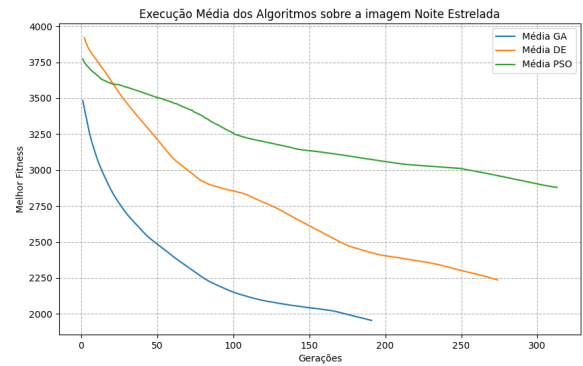
(a) Moça com Brinco de Pérola (50x59 px)



(b) O Grito (100x127 px)

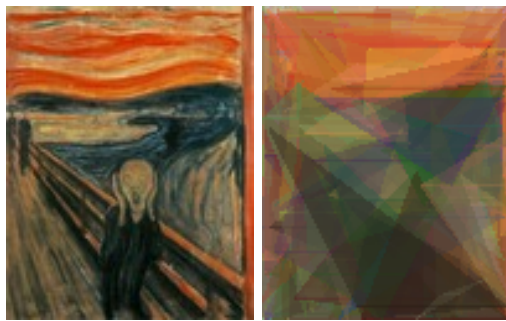


(c) Mona Lisa (168x250 px)



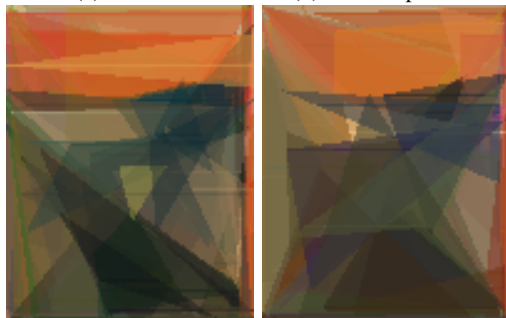
(d) Noite Estrelada (500x396 px)

Fig. 1: Análise com Tempo Fixado



(a) O Grito

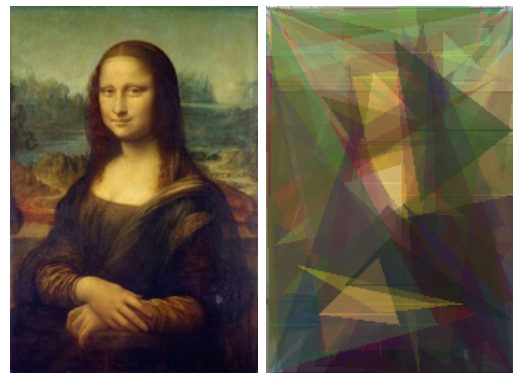
(b) Gerada por GA



(c) Gerada por DE

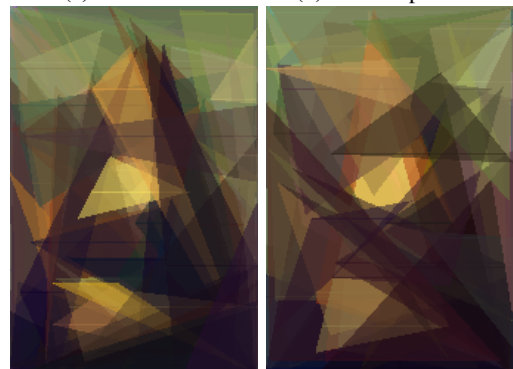
(d) Gerada por PSO

Fig. 5: Conjunto de Imagens com Resolução 100x127 px



(a) Mona Lisa

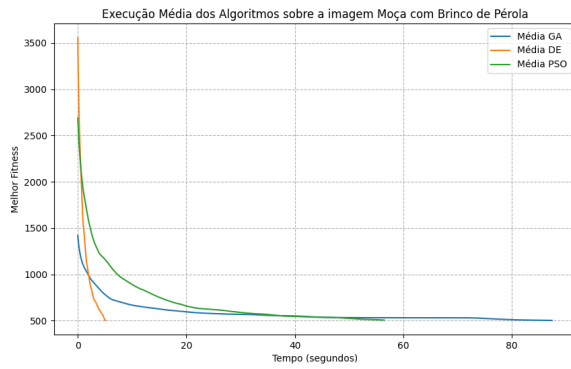
(b) Gerada por GA



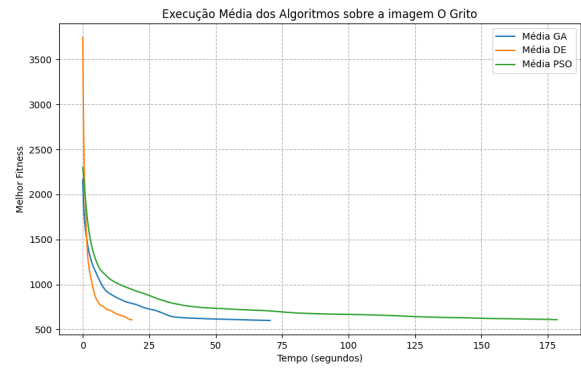
(c) Gerada por DE

(d) Gerada por PSO

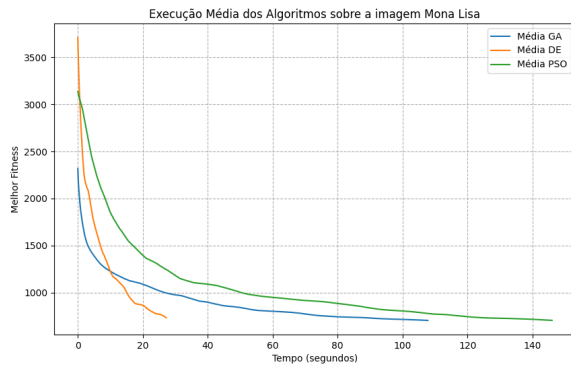
Fig. 6: Conjunto de Imagens com Resolução 168x250 px



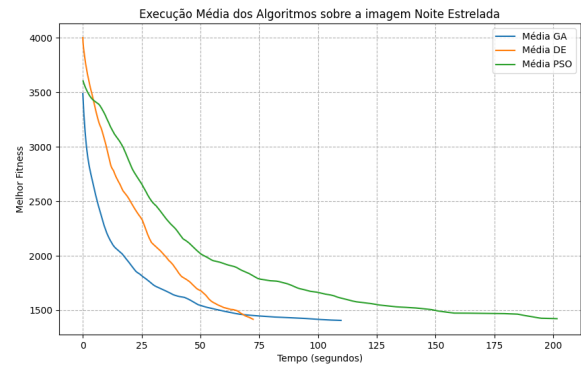
(a) Moça com Brinco de Pérola (50x59 px)



(b) O Grito (100x127 px)

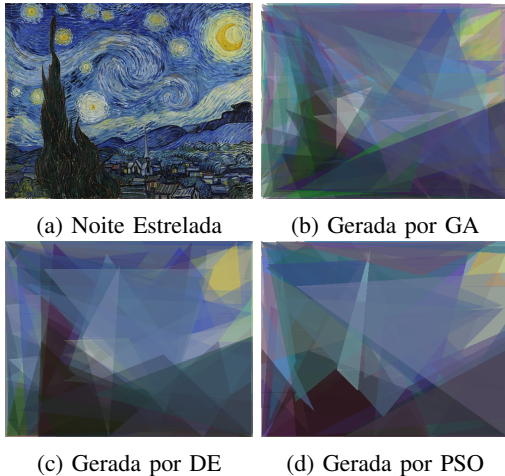


(c) Mona Lisa (168x250 px)



(d) Noite Estrelada (500x396 px)

Fig. 2: Análise com Alvo Fixado



(a) Noite Estrelada

(b) Gerada por GA

(c) Gerada por DE

(d) Gerada por PSO

Fig. 7: Conjunto de Imagens com Resolução 500x396 px

V. CONCLUSÃO

A partir dos resultados obtidos é possível afirmar que o algoritmo de otimização meta-heurística mais apropriado para recriação de imagens estáticas é o DE. Ele é capaz de atingir os mesmos resultados dos outros dois algoritmos em uma fração do tempo.

Levando em consideração a qualidade do produto final, é possível elevar a posição do GA pela variedade de cores que geralmente aparece como artefato de suas mutações, mas esse atributo não é capaz de substituir a velocidade e eficiência que o DE consegue garantir.

Nas iterações futuras desse projeto, pretende-se adicionar mais algoritmos de otimização meta-heurísticas a fins de comparação e também a execução desses algoritmos por mais tempo sobre um número maior de imagens. Ademais, propõe-se a estudar como essa família de algoritmos pode ser utilizada sem a presença de uma imagem estática e somente guiada pelo *feedback* do usuário.

REFERENCES

- [1] W. Wong and C. I. Ming, "A review on metaheuristic algorithms: Recent trends, benchmarking and applications," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, June 2019, pp. 1–5.
- [2] X.-S. Yang, "Metaheuristic optimization: Algorithm analysis and open problems," in *Experimental Algorithms*, P. M. Pardalos and S. Rebennack, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 21–32.
- [3] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

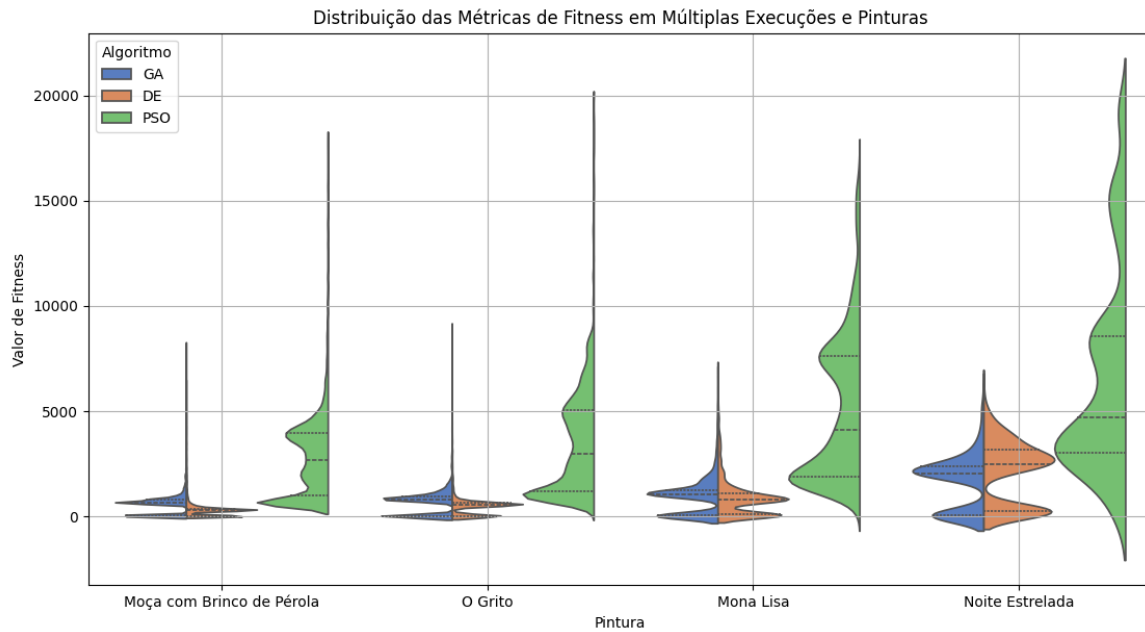


Fig. 3: Análise da Distribuição de Fitness

- [4] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017. [Online]. Available: <https://doi.org/10.1007/s10489-017-0903-6>
- [5] L. Camps Echevarría, O. Llanes Santiago, H. F. d. Campos Velho, and A. J. d. Silva Neto, *Metaheuristics for Optimization Problems*. Cham: Springer International Publishing, 2019, pp. 43–83. [Online]. Available: https://doi.org/10.1007/978-3-319-89978-7_3
- [6] Y. L. Tao, "Triangle rasterizer based on the barycentric arithmetic," *Applied Mechanics and Materials*, vol. 519-520, pp. 697–702, 2014.
- [7] N. Ivanov, "Is rust c++-fast? benchmarking system languages on everyday routines," 2022. [Online]. Available: <https://arxiv.org/abs/2209.09127>
- [8] A. E. Muñoz Zavala, "A comparison study of pso neighborhoods," in *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, O. Schütze, C. A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, and P. Legrand, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 251–265.