

RELATÓRIO FINAL

EvoArtLab: uma Ferramenta Educativa de Código Aberto para Experimentar com Medidas Estéticas

Antônio Isaac Silva Lima
Orientadora: Gisele L. Pappa

Departamento de Ciências da Computação
Universidade Federal de Minas Gerais

19 de janeiro de 2025

Resumo

A estética computacional é um campo de pesquisa que busca replicar decisões estéticas humanas por meio de métodos computacionais, sendo amplamente utilizada na criação de artes evolucionárias. Este projeto teve como objetivo desenvolver uma ferramenta educacional que permite aos usuários explorar o impacto de diferentes medidas estéticas em imagens geradas por algoritmos evolucionários. Utilizando o algoritmo Evolução Diferencial (DE), que se destacou em etapas anteriores por sua eficiência e qualidade de resultados, juntamente com o algoritmo Non-dominated Sorting Genetic Algorithm II (NSGA-II), a fim de combinar múltiplas medidas estéticas, a ferramenta possibilita o uso de medidas estéticas e a análise de seus efeitos em tempo real. Usuários avançados também podem implementar novas medidas, ampliando o potencial da ferramenta. Ao final, foi entregue uma solução de código aberto que promove o entendimento de conceitos avançados em estética computacional e estimula o aprendizado interativo no campo das artes evolucionárias. A ferramenta está disponível publicamente em: <https://github.com/isaacs4m/evoartlab>.

Palavras-chave: Estética Computacional, Artes Evolucionárias, Algoritmo Evolução Diferencial, Ferramenta Educacional, Medidas Estéticas.

Conteúdo

1	Introdução	4
1.1	Objetivos Específicos	4
2	Referencial Teórico	5
3	Metodologia	7
3.1	Implementação do NSGA-II	7
3.2	Implementação de Medidas Estéticas Iniciais	8
3.2.1	Pixel Compare	8
3.2.2	Global Contrast Factor	9
3.2.3	Benford Law	9
3.3	Implementação de Sistema de Plug-In	9
4	Resultados	10
4.1	Distribuição	10
4.2	Medidas Estéticas	11
4.3	Saídas da Ferramenta	12
5	Exemplo de Uso	13
6	Conclusão	14

1 Introdução

Os algoritmos de otimização meta-heurísticos são inspirados na natureza [1] e desde a década de 90 muitos desses algoritmos são aplicados em quase todas as áreas de otimização, design, agendamento, planejamento e muitas outras. Um de seus usos é a criação de arte evolucionária, isto é, arte criada através da aplicação de computação evolucionária para a criação de imagens esteticamente agradáveis [2].

O campo de pesquisa de Estética Computacional tem como objetivo a investigação de "métodos computacionais que possam tomar decisões estéticas de forma semelhante aos humanos" [3]. Essas decisões são feitas por meio de medidas estéticas: funções que computam o valor estético de um objeto [4]. Dessa forma, essas medidas são a parte mais complexa na tentativa de desenvolver modelos das preferências estéticas de pessoas [5], também sendo importantes na criação de arte evolucionária sem supervisão.

A primeira parte do projeto [6] buscou testar diferentes algoritmos evolucionários e compará-los na eficiência e qualidade de geração de imagens a partir de imagens estáticas, isto é, ao alimentar os algoritmos com uma imagem específica, se comparava a imagem criada por eles com a imagem original. Ao fim dela, foi possível determinar que o algoritmo DE teve os melhores resultados tanto em relação à eficiência do algoritmo em si quanto ao resultado final.

1.1 Objetivos Específicos

Nessa segunda parte, o objetivo específico será usar os conhecimentos adquiridos com geração de imagens usando DE e NSGA-II para criar uma ferramenta educacional que auxiliará no entendimento de diferentes medidas estéticas, permitindo ao usuário usar e combinar diferentes medidas estéticas e observar o efeito delas nas imagens resultantes. Além disso, um usuário mais avançado também poderá adicionar novas medidas estéticas e observar seus efeitos nas artes geradas. Os objetivos gerais dessa etapa serão propor uma maneira prática e fácil de comparar e entender os impactos nas artes evolucionárias das escolhas de como definimos o que é estética para o computador.

2 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 possam 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) [7]. 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 [8]. 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.

O algoritmo DE, utilizado em foco para esse projeto, é uma meta-heurística baseada em população amplamente utilizada devido à sua simplicidade e eficiência. Ele é mais simples de codificar, implementar e usar do que muitos outros métodos de otimização disponíveis, como simulated annealing e algoritmos genéticos tradicionais [9]. Sua abordagem baseia-se na criação de novas soluções a partir de combinações lineares de membros da população, permitindo que a mutação seja adaptativa e mais adequada a problemas com variáveis contínuas. E estudos [9] demonstram que o DE supera consistentemente métodos clássicos em desempenho, apresentando resultados equivalentes ou superiores a outros algoritmos evolucionários em problemas de otimização de engenharia mecânica e design.

Nesse contexto, o NSGA-II é uma meta-heurística evolucionária proje-

tada para resolver problemas de otimização multiobjetivo, proporcionando um equilíbrio eficaz entre diversidade e convergência. Diferentemente de métodos tradicionais, que convertem problemas multiobjetivo em problemas de objetivo único, o NSGA-II é capaz de encontrar múltiplas soluções Pareto-ótimas em uma única execução, aproveitando sua abordagem baseada em populações [10]. Ele introduziu um procedimento rápido de ordenação não-dominada, um mecanismo elitista para preservar soluções de alta qualidade ao longo das gerações, e um operador de diversidade sem parâmetros para manter a distribuição uniforme no espaço objetivo [10].

3 Metodologia

Esse trabalho buscou desenvolver uma ferramenta educacional na linguagem Rust capaz de auxiliar no entendimento de diferentes medidas estéticas no contexto de artes evolucionárias, se utilizando de uma combinação dos algoritmos conhecidos DE e NSGA-II. A seguir estão delimitados os passos que foram seguidos para cumprir o objetivo do projeto.

3.1 Implementação do NSGA-II

Primeiramente, o projeto anterior [6] desenvolveu uma arquitetura robusta de criação de imagens a partir de combinação de polígonos semitransparentes utilizando somente o DE, logo, o primeiro passo da implementação da segunda fase do trabalho envolve a implementação do algoritmo NSGA-II, permitindo que mais de uma função de *fitness* seja utilizada ao mesmo tempo. Os objetivos gerais desse passo consistiam na implementação do algoritmo de forma que o custo computacional não se elevasse mesmo no caso no qual apenas uma medida estética era utilizada; além disso, se desejava manter a maior quantidade possível de elementos do algoritmo DE, que apresentou excelentes resultados na etapa anterior.

Para atender os requisitos planejados, foi criado o algoritmo híbrido em 1. A maior diferença do algoritmo padrão de NSGA-II é que no caso em que apenas uma medida estética é usada, o algoritmo não faz o cálculo de *crowding distance*, pois nesse caso não há uma necessidade elevada de se manter um alto nível de diversidade. Essa mudança permite que, no caso de apenas uma medida estética, o algoritmo NSGA-II tenha o mesmo nível de eficiência que o algoritmo DE sozinho teria.

Algorithm 1: Algoritmo Híbrido de DE e NSGA-II

Data: Parâmetros de configuração da evolução
Result: Fronte não-dominada de indivíduos
Inicialize população inicial de forma aleatória
while *Condição de parada não atendida* **do**
 Faça um clone da população
 for *Cada índice da população* **do**
 Escolha 3 indivíduos através da seleção por torneio
 Combine os 3 indivíduos para gerar um novo usando a técnica de DE
 Substitua o indivíduo criado posição do índice
 end
 Compare a fitness de cada indivíduo da antiga e nova população
 Mantenha somente os melhores indivíduos em cada índice das populações
 if *Apenas uma função fitness* **then**
 | Usar seleção por NSGA-II sem *crowding distance*
 end
 else
 | Usar seleção por NSGA-II com *crowding distance*
 end
end

3.2 Implementação de Medidas Estéticas Iniciais

Uma vez que os algoritmos evolucionários estavam devidamente implementados e tínhamos a opção de utilizar mais de uma medida estética simultaneamente, o próximo passo foi implementar diferentes medidas estéticas por padrão, para facilitar a demonstração de uso da ferramenta.

3.2.1 Pixel Compare

Essa medida estética básica avalia a similaridade entre duas imagens, comparando diretamente os valores de pixel. Ela é útil como métrica de referência devido à sua simplicidade e sua capacidade de capturar mudanças detalhadas entre imagens. O erro entre as imagens é calculado usando *Mean Square Error* (MSE), uma medida de erro usada na etapa anterior do trabalho.

3.2.2 Global Contrast Factor

O *Global Contrast Factor* (GCF) é uma medida estética descrita em [11]. Ele calcula o contraste global de uma imagem, ou seja, as diferenças de luminância ou brilho em diversas resoluções. Imagens com poucas diferenças de luminância apresentam baixo contraste e são consideradas "monótonas" [11], com menor valor estético.

3.2.3 Benford Law

A Lei de Benford, ou Lei do Primeiro Dígito, é uma medida estética baseada na observação de que conjuntos de números obtidos da vida real (e não criados artificialmente) seguem uma distribuição específica e não uniforme [12]. Por exemplo, o dígito inicial aparece em cerca de um terço das vezes, enquanto o segundo dígito aparece em 17,6% dos casos [12]. Para avaliar imagens, aplicamos a Lei de Benford sobre a distribuição de brilho dos pixels. A implementação seguiu os parâmetros descritos em [11].

3.3 Implementação de Sistema de Plug-In

Para possibilitar que usuários avançados com conhecimento básico de Rust possam experimentar com suas próprias medidas estéticas, buscamos que a integração de código extra fosse feita de forma simples e sem complicações extras. Atualmente, para se criar uma nova medida estética, basta criar um arquivo dedicado para a nova medida e colocá-lo dentro do projeto. Porém, devido à natureza compilada da linguagem Rust, a implementação escolhida para realizar a integração com o resto do projeto foi ter o usuário manualmente importar a medida estética criada no arquivo principal do projeto.

4 Resultados

Esta seção apresenta os resultados obtidos a partir da condução dos processos descritos na metodologia.

4.1 Distribuição

Atualmente, o projeto compilado e seu código-fonte estão disponíveis na plataforma de controle de versões GitHub. Além do código, a plataforma também traz uma página inicial chamada *README*, exibida em 1, com informações a respeito do projeto e como ele deve ser utilizado.

O *README* contém uma descrição do projeto, uma seção com explicações de como instalá-lo, um guia de utilização do EvoArtLab através da GUI e um exemplo de uso real da aplicação.

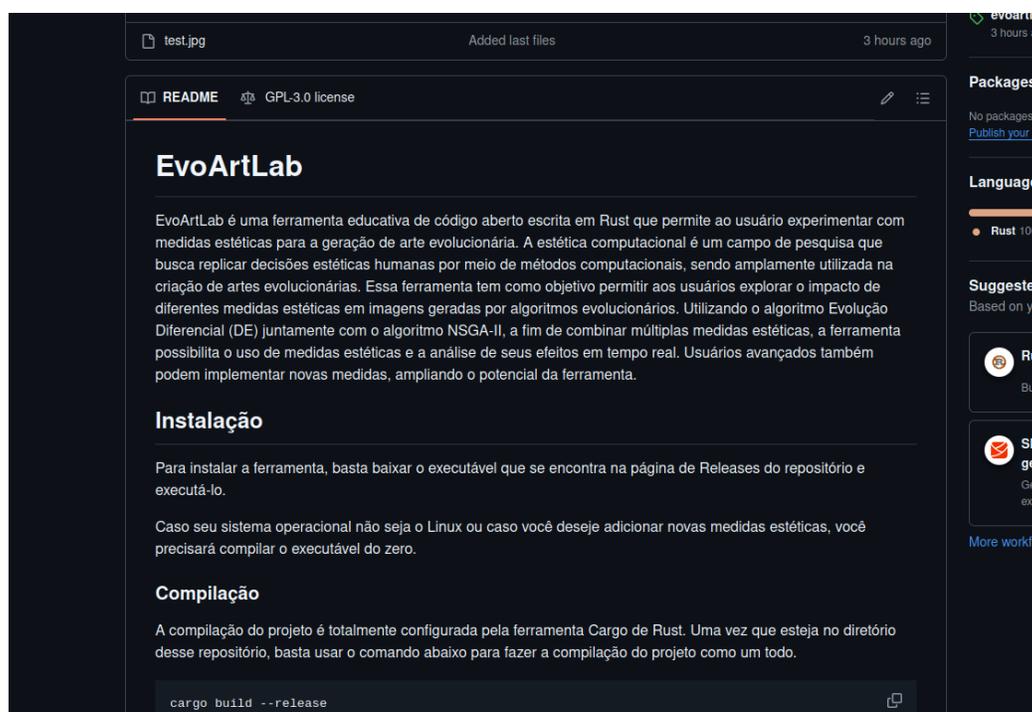


Figura 1: Página README encontrada no repositório do GitHub da ferramenta.

4.2 Medidas Estéticas

Cada medida estética foi capaz de gerar imagens distintas o suficiente para se poder observar o impacto que as diferentes implementações de funções fitness pudessem ser observadas. É possível observar o resultado das imagens gerada por cada medida estética em 2. Ademais, é importante notar que apenas a medida estética de Pixel Compare depende de uma imagem base para funcionar.



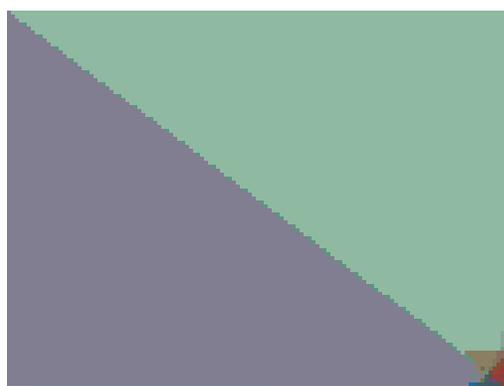
(a) Imagem base para o Pixel Compare.



(b) Medida estética de Pixel Compare.



(c) Medida estética de GCF.



(d) Medida estética de Benford Law.

Figura 2: Resultado de uso das diferentes medidas estéticas.

4.3 Saídas da Ferramenta

A principal saída do programa é um conjunto de imagens representando o fronte não-dominado gerado pelo algoritmo NSGA-II. Quando apenas uma medida estética é usada, o fronte é composto de apenas uma imagem, porém quando múltiplas medidas são utilizadas, o fronte terá dezenas de imagens representando os melhores indivíduos gerados pelo processo de arte evolucionária.

5 Exemplo de Uso

Para demonstrar a utilização da ferramenta, propomos nesta seção a geração de imagens utilizando uma combinação das medidas estéticas Pixel Compare e GCF. Os resultados em 3 demonstraram que a integração de ambas as medidas permitiu a criação de imagens visualmente equilibradas, combinando alta similaridade estrutural com um contraste visual mais dinâmico, destacando o potencial da ferramenta em manipular múltiplos critérios estéticos de forma simultânea e interativa.



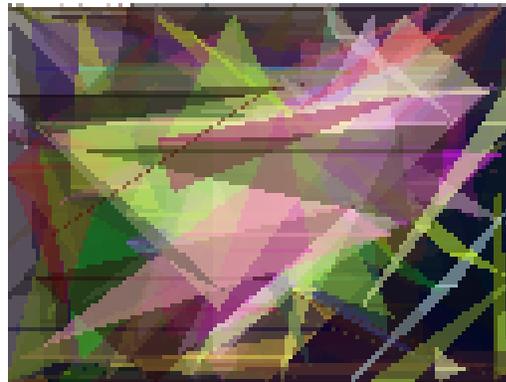
(a) Imagem mais próxima do Pixel Compare.



(b) Imagem equilibrada, com maior influência de Pixel Compare.



(c) Imagem equilibrada, com maior influência de GCF.



(d) Imagem mais próxima de Global Contrast Factor.

Figura 3: Resultado de uso da combinação da medida Pixel Compare com Contrast.

6 Conclusão

Neste trabalho, buscamos desenvolver uma ferramenta educacional capaz de auxiliar no entendimento de diferentes medidas estéticas no contexto de artes evolucionárias. Aproveitamos os conhecimentos adquiridos em etapas anteriores, utilizando os algoritmos DE e NSGA-II para permitir que usuários explorem e combinem medidas estéticas, observando os efeitos dessas combinações nas imagens resultantes. Além disso, a ferramenta foi projetada para atender tanto iniciantes quanto usuários avançados, oferecendo a possibilidade de incluir novas medidas estéticas e avaliar seus impactos. Através do processo de desenvolvimento e validação, conseguimos entregar uma solução prática, acessível e aberta, que promove o aprendizado interativo e facilita a compreensão de como definições computacionais de estética influenciam a geração de artes evolucionárias.

Em trabalhos futuros, a ferramenta pode ser ampliada com a inclusão de novas medidas estéticas, suporte a outros algoritmos evolucionários, e melhorias na interface para proporcionar uma experiência mais intuitiva para os usuários.

Referências

- [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)*, pp. 1–5, June 2019.
- [2] E. den Heijer and A. Eiben, “Investigating aesthetic measures for unsupervised evolutionary art,” *Swarm and evolutionary computation*, vol. 16, pp. 52–68, 2014.
- [3] F. Hoenig, “Defining computational aesthetics,” in *Proceedings of the First Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics’05, (Goslar, DEU), p. 13–18, Eurographics Association, 2005.
- [4] E. den Heijer and A. E. Eiben, “Using aesthetic measures to evolve art,” in *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2010.
- [5] C. G. Johnson, J. McCormack, I. Santos, and J. Romero, “Understanding aesthetics and fitness measures in evolutionary art systems,” *Complexity*, vol. 2019, no. 1, p. 3495962, 2019.
- [6] A. I. S. Lima, “Comparando algoritmos de otimização meta-heurística para recriação de imagens estáticas,” *POC I*, 2024.
- [7] X.-S. Yang, “Metaheuristic optimization: Algorithm analysis and open problems,” in *Experimental Algorithms* (P. M. Pardalos and S. Rebenack, eds.), (Berlin, Heidelberg), pp. 21–32, Springer Berlin Heidelberg, 2011.
- [8] 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.
- [9] D. G. Mayer, B. P. Kinghorn, and A. A. Archer, “Differential evolution – an easy and efficient evolutionary algorithm for model optimisation,” *Agric. Syst.*, vol. 83, pp. 315–328, Mar. 2005.
- [10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, Apr. 2002.

- [11] E. den Heijer and A. E. Eiben, “Evolving art using multiple aesthetic measures,” in *Applications of Evolutionary Computation* (C. Di Chio, A. Brabazon, G. A. Di Caro, R. Drechsler, M. Farooq, J. Grahl, G. Greenfield, C. Prins, J. Romero, G. Squillero, E. Tarantino, A. G. B. Tettamanzi, N. Urquhart, and A. Ş. Uyar, eds.), (Berlin, Heidelberg), pp. 234–243, Springer Berlin Heidelberg, 2011.
- [12] A. Berger and T. P. Hill, “Benford’s law strikes back: No simple explanation in sight for mathematical gem,” *Math. Intelligencer*, vol. 33, pp. 85–91, Mar. 2011.