Painting with Evolutionary Algorithms:

the Effects of Brush Stroke Sparsity

By Karim Nasr, Okke van Eck and Daan van den Berg

Introduction

Introduction

Based off of Dijkzeul et al'. paper, 'Painting with Evolutionary Algorithms'.

- Uses plant propagation (PPA), simulated annealing (SA) and hill climbing algorithms (HC)
- SA > HC > PPA

Why?

- Evolutionary nature of computational art
- Algorithmic behavior

Objective

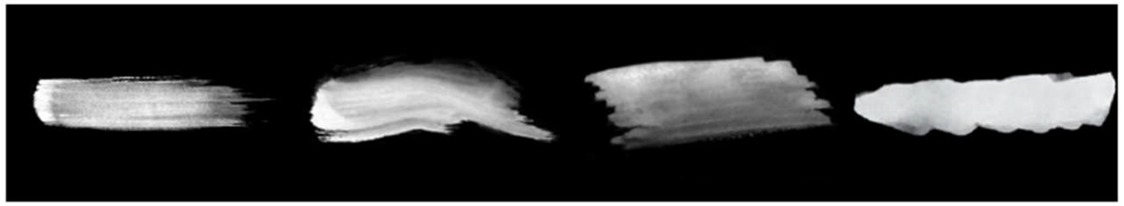
To compare the performances of the different algorithms by their performance in:

- Mean squared error (MSE) → $MSE = \frac{1}{N}\sum_{i=1}^{N}(I_1(i) - I_2(i))^2$
- Average brush stroke size
- Brush type frequency

Research question

The research question is as follows:

How does the choice of brush types impact the performance and runtime of evolutionary algorithms in generating paintings?



Brush type 1 Brush type 2 Brush type 3 Brush type 4

Methodology

Scope

Seven different paintings:

Algorithms

- Hill climber (HC) algorithm
- Simulated annealing (SA) algorithm →

$$P(accept) = e - (\frac{\triangle MSE}{temp})$$

$$temp = \frac{c}{log(i)}$$

- Tabu search (TS) algorithm
    - Makes use of a tabu list

Mean squared error

- Calculating the MSE of the newly generated canvas on every evaluation and comparing it to the previous one, only accepting a better error.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I_1(i) - I_2(i))^2$$

MSE = 2948.67

Experimental Setup

Overview

- Canvas specifications
  - 240×180 pixels
  - black background
  - initialize a random canvas
- Experiments with 3 and 4 brush types
- 1,000,000 evaluations per run (totalling 210,000,000 evaluations)
- 5 runs for every painting per algorithm (totalling 210 runs)
- 25 brush strokes
- Brush stroke mutation on every evaluation

3 brush types

4 brush types

Mutation

- Colour
- Shape
- Size
- Rotation
- Position
- Brush type
- Index

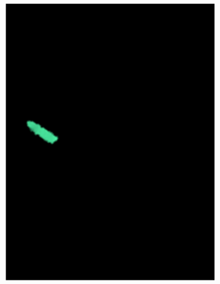
Parameters

- Simulated annealing uses the current evaluation number in its cooling function, the higher the evaluation number the lower the probability of accepting a worse solution.

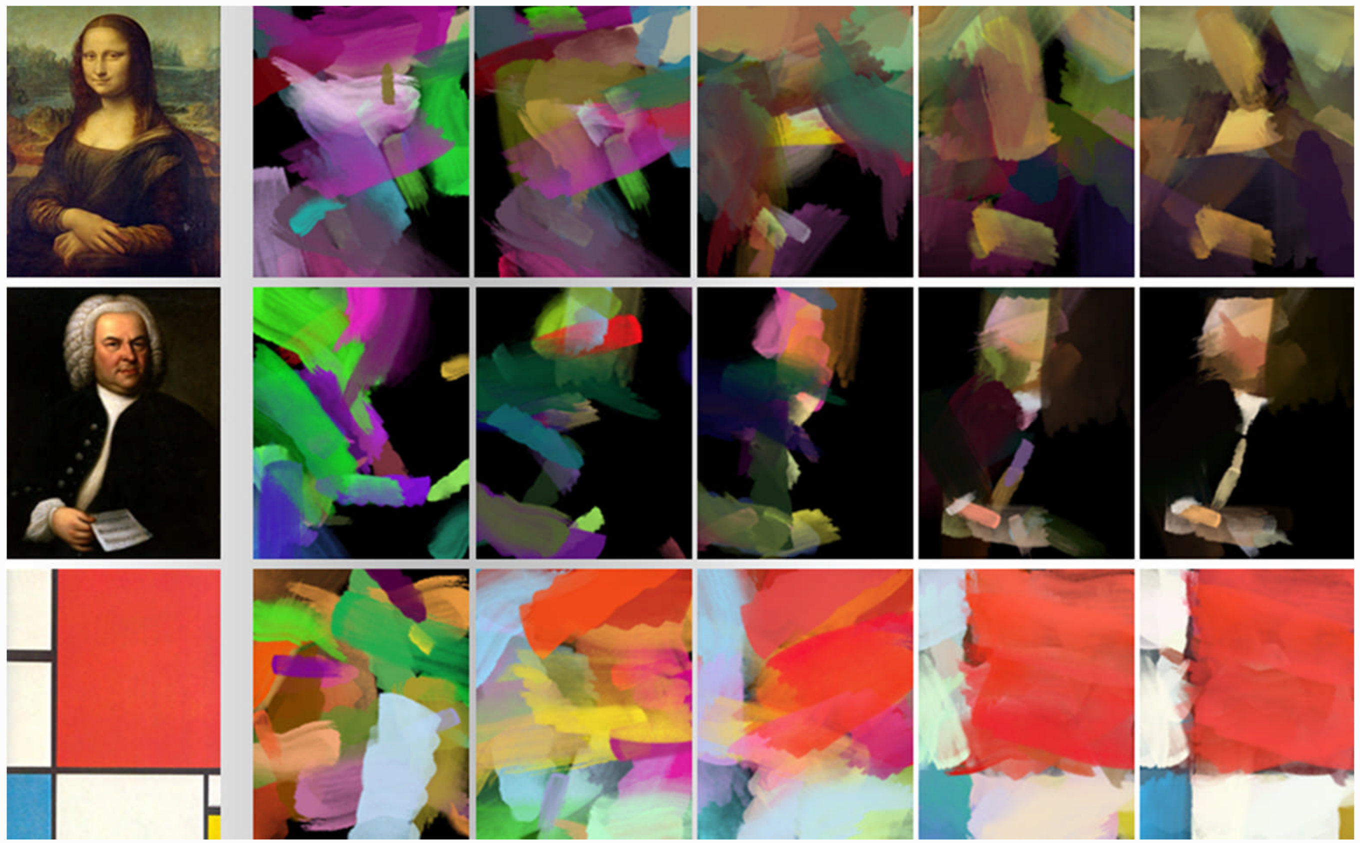
- Tabu search uses a tabu list of size 50.

- Brush color using RGB color codes.

- Brush size between 0.1 and 0.7.



Example:

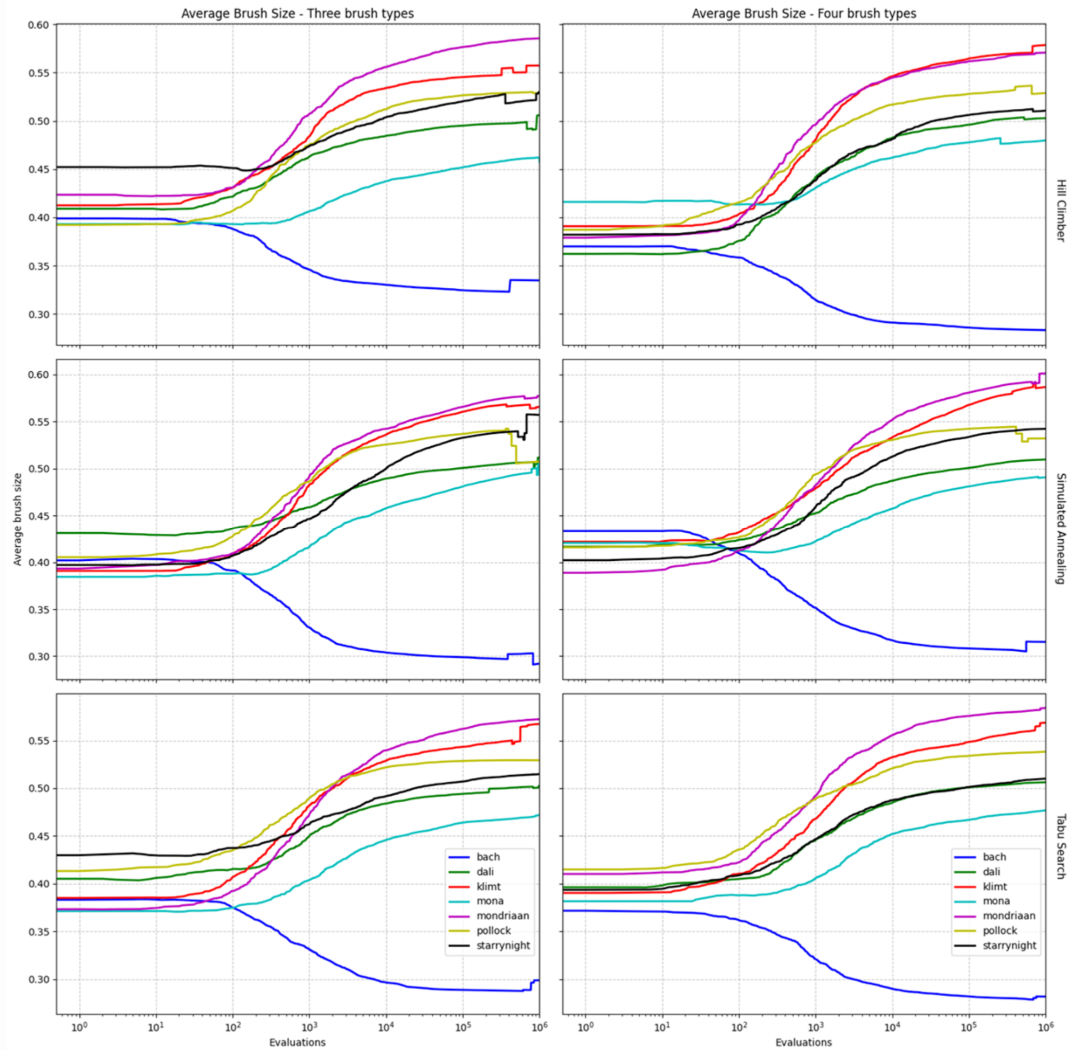


Results

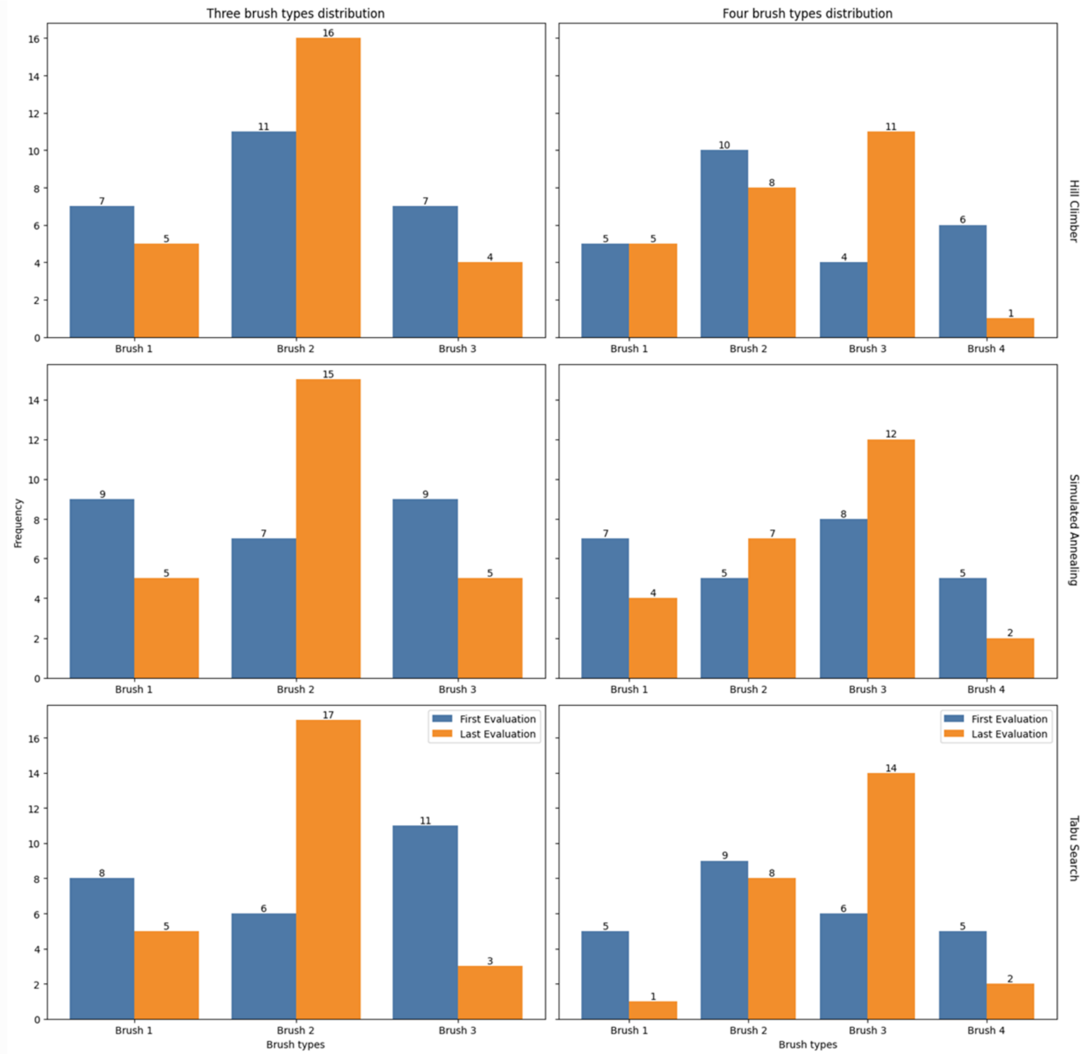
Metrics

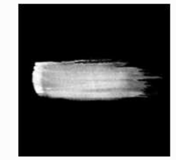
- Average brush size
  - Average size change over iterations
  - Notable pattern: General increase except for Bach portrait
- Brush type frequency
  - Expected baseline:
    - 33.3% for 3-type experiment
    - 25% for 4-type experiment
  - Actual findings:
    - 44-48% preference for type 2 in 3-type experiment
    - 40-44% preference for type 3 in 4-type experiment

Average brush size

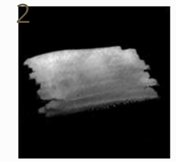


Brush type frequency




Brush type 1


Brush type 2


Brush type 3


Brush type 4

Conclusion

The ‘Bach Anomaly’ and its significance

- The only painting showing decreasing brush stroke sizes.
- Contradicts the pattern seen in all other paintings.
- Potential explanations:
  - Role of the black background becoming more influential in sparse settings.
  - Possible trade-off between detail preservation and background coverage.
  - Questions about whether this is a general pattern for portraits with dark backgrounds.

Discussion

- Similar performance across all three algorithms (HC, SA, Tabu).
- Shift in brush type preference when adding a fourth type.
- Technical implications:
    - State space complexity (10^284 possible states)
    - Question of whether current parameters are optimal
    - Possibility that sparse conditions create different optimization landscapes

Future work

- Incorporate a genetic algorithm.
- Adding more brush types.
- Enable background color mutations.
- Fine-tuning parameters for the existing algorithms.
- Testing a larger number of evaluations.
- Investigating other artistic styles.

Takeaways

- Deeper understanding of evolutionary optimization algorithms.
- Importance of background in sparse compositions.
- The importance of parameter tuning.
- The impact that one feature could have on the performance of the algorithms.

References

Dijkzeul, D., Brouwer, N., Pijning, I., Koppenhol, L., & van den Berg, D. (2022). *Painting with Evolutionary Algorithms.*

Eiben, A., & Smith, J. (2015). *Introduction to Evolutionary Computing.* Natural Computing Series.

Nasr, K. (2024, January 10). *Source code for this project*. Retrieved from GitHub: https://github.com/KarimN7/painting-generation

Opara, A. (2020, June 10). A genetic algorithm toy project for drawing. Retrieved November 2023, from Github: https://github.com/anopara/genetic-drawing

Chopard, B., & Tomassini, M. (2018). Simulated Annealing. In *An Introduction to Metaheuristics for Optimization* (pp. 59-79). Natural Computing Series.

Chopard, B., & Tomassini, M. (2018). Tabu Search. In *An Introduction to Metaheuristics for Optimization* (pp. 43-56). Natural Computing Series.

Selman, B., & Gomes, C. (2006). Hill-climbing Search. In Encyclopedia of Cognitive Science (pp. 333-335).

Thanks for listening

Questions?