

Aproximando a Mona Lisa com círculos utilizando Algoritmo Genético

AKARI ISHIKAWA (163282), TALES LELO (177312)

Instituto de Computação - Unicamp

Resumo – Neste projeto, exploramos o uso de algoritmo genético para o problema de se aproximar uma imagem utilizando círculos. Neste relatório, especificamos detalhes quanto à modelagem do problema, parâmetros e operadores de algoritmo genético. Na seção de experimentos, testamos e discutimos *trade-offs* e limitações dos parâmetros e propomos possíveis melhorias para o trabalho.

Palavras-chave – Algoritmo genético, Imagem aproximada, círculos

I. INTRODUÇÃO

A área dos **algoritmos evolutivos** inspira-se na biologia para propor algoritmos para solucionar problemas usando conceitos de evolução.

Neste projeto, empregamos **algoritmos genéticos** (AG), uma subclasse dos algoritmos evolutivos, para explorar o problema de **aproximar uma imagem com K círculos** (figura 1). Este problema pode ser avaliado como um tipo de vetorização da imagem, dado que, os círculos encontrados podem ser redimensionados conforme se necessite.

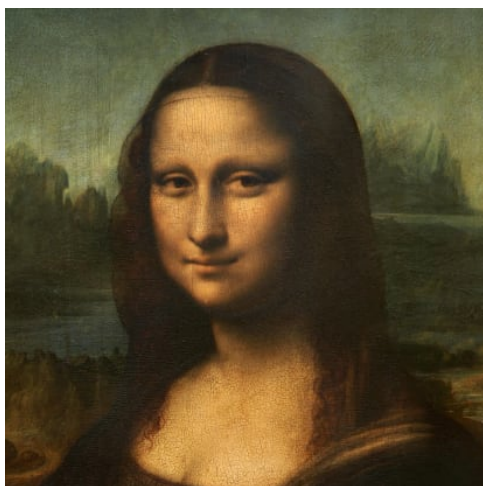


Figura 1. Imagem que buscamos aproximar com círculos usando algoritmo genético.

Uma das vantagens de se utilizar este método é a facilidade em paraleliza-lo, visto que, independente do problema, múltiplas soluções podem ser avaliadas simultaneamente e, ainda, das soluções parciais encontradas podem ser retirados fragmentos que combinados através da reprodução sexuada formam uma nova solução parcial melhor.

Nas seções seguintes, especificamos detalhes a respeito da modelagem do problema, os experimentos realizados e as observações feitas.

Toda a implementação é de código aberto e está disponível em <https://github.com/akariueda/mc906-lab2>.

II. MODELAGEM

Nesta seção, especificamos detalhes quanto à modelagem do problema para solucioná-lo com algoritmo genético.

Dado que em AG, cada solução é um indivíduo, nosso indivíduo é uma imagem de dimensões $N \times M$ com um conjunto de K círculos (genes).

Os genes que compõem cada indivíduo são os círculos. A seguir especificamos os atributos e a forma como eles são definidos quando o círculo é criado aleatoriamente:

- **Raio:** Um círculo é derivado da menor dimensão da imagem multiplicada por um valor aleatório dentro do intervalo $[0.01, 0.2]$ de acordo com uma distribuição uniforme. Isso garante que um círculo não tome uma área exagerada da imagem gerada e, também, que cada círculo seja grande o suficiente para influenciar no *fitness*.
- **Centro (x, y) :** Cada coordenada (x ou y) do centro do círculo é definida pela multiplicação de um valor aleatório entre 0 e 1 (distribuição uniforme) pela altura N , no caso de x , ou largura M , no caso de y .
- **Opacidade:** Definimos o α de opacidade com um valor fixo de 0.8 para permitir combinação de cores, mas reduzir uma dimensão no problema.
- **Cor:** A cor é definida com uma tupla com três valores entre 0 e 255 representando os três canais de cor: vermelho, verde e azul (RGB). Realizamos testes inicializando a cor aleatoriamente, porém para nosso problema a cor é um fator muito determinante para que um humano considere duas imagens semelhantes. Como estávamos tendo muitos problemas com convergência, inicializamos a cor de um novo círculo com a cor da imagem original no ponto (x, y) .

A população, de acordo com a teoria de algoritmos genéticos, é um conjunto de indivíduos em que, à cada geração, aplicamos alguns *operadores genéticos* para produzir a próxima geração. A próxima geração idealmente deveria conter soluções (imagens) melhores para o problema buscado.

Neste projeto, usamos os operadores de seleção, *crossover* e mutação. As especificidades quanto às técnicas e parâmetros são detalhadas na próxima seção.

A função de *fitness* utilizada em todos os experimentos deste projeto foi o Índice de Similaridade Estrutural (SSIM), com a implementação proposta pelo módulo de Python *Scikit-image*. O SSIM possui valores de $[-1, 1]$ sendo 1 similaridade entre duas imagens idênticas.

III. EXPERIMENTOS E DISCUSSÃO

Dado a grande gama de parâmetros envolvidos numa solução com algoritmo genético, neste projeto experimentamos variações em cada um dos parâmetros para observar o efeito causado. Fizemos experimentos variando os parâmetros de tamanho da população, critério de parada, técnica de seleção, técnica e taxa de *crossover*, técnica de mutação, método de substituição e taxa de mutação.

A. Baseline

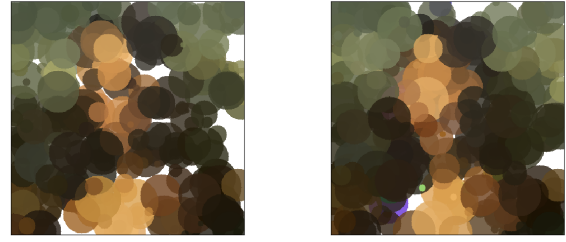
Para notar a diferença causada na variação dos parâmetros, precisávamos de um *baseline* para as comparações. Escolhemos técnicas e valores baseados em poucos experimentos que fizemos enquanto explorávamos a técnica. A baseline possui a seguinte configuração:

- **Tamanho da População:** por limitações computacionais, não podíamos realizar testes com populações com muitos indivíduos. Decidimos então por fazer um teste base com 10 indivíduos.
- **Critério de Parada:** Colocamos um critério de parada de 90 gerações para coletar um resultado inicial.
- **Técnica de Seleção:** os 90% melhores indivíduos de uma geração são escolhidos para se reproduzirem.
- **Técnica de *crossover*:** Dado 2 indivíduos selecionados, cada gene i tem chance de vir de um dos pais com probabilidade $p = 0.5$, mesclando ordenadamente seus genes (*uniform crossover*) para gerar um novo indivíduo que terá G genes (círculos) sendo G o menor valor entre a quantidade de círculos de seus pais.
- **Técnica de Mutação:** Mutação uniforme. A cada gene (círculo) escolhido para mutação, ele é completamente substituído por outro círculo criado aleatoriamente.
- **Taxa de mutação:** 50% dos indivíduos com a menor fitness tem 3% dos seus genes (círculos) modificados após a operação de *crossover*.
- **Método de Substituição:** os melhores indivíduos dentre esta geração e seus filhos vão para a próxima

B. Modificando o tamanho da população

Fizemos um breve experimento modificando o tamanho da população de 10 para 20. Mantivemos todos demais parâmetros intactos. Na figura 3 é possível ver o melhor indivíduo da geração 0 (iniciada aleatoriamente) e da geração 90.

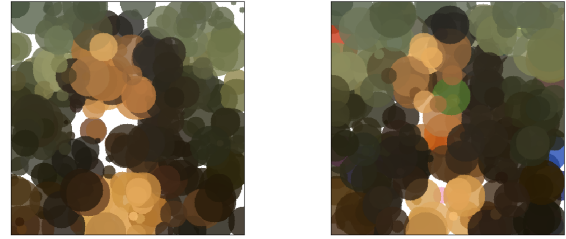
Notamos que o experimento com 10 indivíduos tomou 6,43 segundos para executar 90 gerações, enquanto que o experimento com 20 indivíduos demorou 10,62 segundos, um aumento de 65% no tempo de processamento.



(a) Geração 0

(b) Geração 90

Figura 2. Comparação entre a geração 0 e geração 90 do experimento baseline.



(a) Geração 0

(b) Geração 90

Figura 3. Comparação entre a geração 0 e geração 90 do experimento com 20 indivíduos na população.

Comparando a fitness dos dois experimentos na figura 4, notamos que o experimento com 20 indivíduos convergiu com menos gerações, sem muito prejuízo ao custo computacional.

Decidimos então para os experimentos seguintes usar população de tamanho 20.

C. Modificando a técnica de mutação

Nos experimentos anteriores, utilizamos a técnica de mutação uniforme. Ou seja, quando um gene era escolhido para sofrer mutação, ele era completamente substituído por um novo círculo criado aleatoriamente. Testamos nesse experimento uma técnica diferente: para cada atributo atr_i do círculo a ser mutado, somamos à ele o mesmo atributo atr_j de um círculo criado aleatoriamente regulado por um parâmetro de mutação interna (*inner_mr*) de acordo com a equação:

$$atr_i = atr_i * (1 - inner_mr) + (atr_j * inner_mr) \quad (1)$$

Experimentamos com $inner_mr = 0.1$, ou seja, somamos 10% dos atributos do novo círculo ao círculo atual.

Comparando a geração 90 usando a nova mutação com a mutação anterior, encontramos um valor de *fitness* de 0.613 para a nova técnica e 0.569 para a mutação uniforme, sem prejuízos ao tempo de processamento.

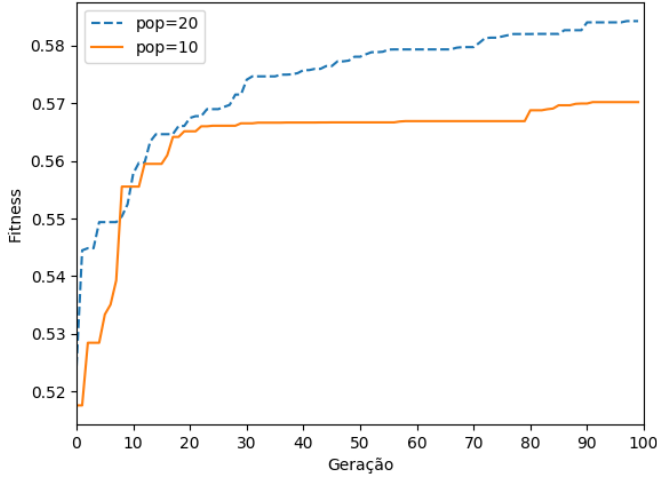


Figura 4. Imagem que buscamos aproximar com círculos usando algoritmo genético.

Hipotetizamos que a mutação por soma traz melhores resultados por modificar os genes o suficiente para gerar variabilidade sem descartar as boas soluções já encontradas.

Observando a melhor solução encontrada na geração 90 com a nova mutação (figura 5), notamos uma melhora promissora que nos levou, então, a manter a nova técnica de mutação para os próximos experimentos.

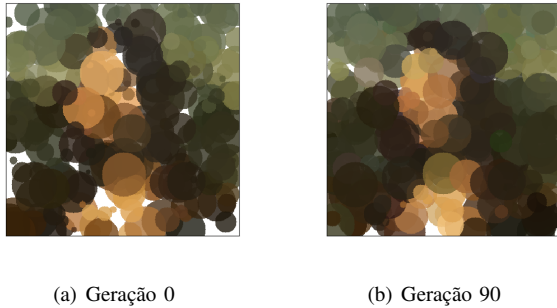


Figura 5. Comparação entre a geração 0 e geração 90 do experimento com a técnica de mutação por soma.

D. Modificando a técnica e taxa de crossover

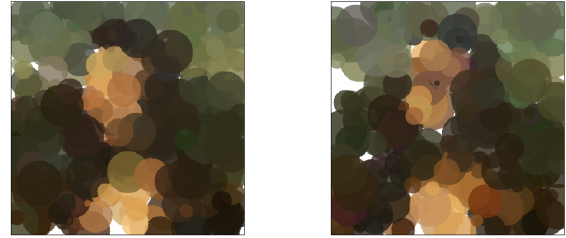
Nos experimentos anteriores usávamos a técnica de *crossover* uniforme que, dado dois indivíduos de uma geração selecionados para reprodução, criava um novo filho tal que cada gene i tem chance de vir de um dos pais com probabilidade $p = 0.5$.

Neste experimento, propomos o crossover com uma taxa $K = 1$ que pega os genes de um pai ou de outro a partir de um corte. A cada cruzamento de um casal, escolhemos um gene aleatório i de 0 a n do novo filho. Para a reprodução,

para os genes $j = 0$ até $j = i$, o filho recebia os genes do primeiro pai, e de $j = i + 1$ até $j = n$ recebia os genes do segundo pai.

Para o novo experimento, o melhor indivíduo encontrado após 90 gerações apresentou *fitness* com valor 0.591 enquanto que o experimento com cruzamento aleatório apresentou valor 0.613.

Na figura 6, podemos comparar as melhores soluções com a reprodução aleatória (à esquerda) com a nova taxa que se assemelha à um corte (à direita).



(a) Crossover aleatório

(b) Crossover com corte

Figura 6. Comparação entre os melhores indivíduos dos experimentos com a crossover com probabilidade $p = 0.5$ (à esquerda) e com um corte (à direita), ambos na 90ª geração.

Apesar de subjetivo, para nós autores, a imagem parece ter apresentado uma queda de qualidade, reforçado pela redução da pontuação. Sendo assim, decidimos prosseguir com a técnica de crossover de genes aleatórios, como estávamos executando anteriormente.

E. Modificando a taxa de mutação

Antes de prosseguirmos com mais experimentos, gostaríamos de oferecer ao leitor um lembrete com os parâmetros que iremos utilizar nesta seção, já adaptada com as modificações efetuadas após a discussão dos resultados anteriores:

- Tamanho da População: 20
- Critério de Parada: 90 gerações
- Técnica de Seleção: os 90% melhores indivíduos podem ser selecionados para cruzarem entre si
- Técnica de Crossover: dado dois pais, os genes deles são selecionados aleatoriamente para formar os genes do filho
- Técnica de Mutação: técnica baseada em soma de atributos de um novo círculo aleatório
- Método de Substituição: os melhores indivíduos dentre esta geração e seus filhos vão para a próxima
- Taxa de Mutação: 3% dos genes dos indivíduos mutados são modificados.

Neste experimento, propomos aumentar a taxa de mutação dos indivíduos. Ou seja, dado um indivíduo a sofrer mutação, agora 20% dos seus círculos sofrerão mutação ao invés de apenas 3% como vínhamos fazendo. Esperávamos, com essa modificação, gerar mais variabilidade durante a evolução.

No entanto, após a 30ª geração, a fitness ficou “presa” em um valor por volta de 0.575. Se compararmos com a taxa de 3% de mutação na figura 7 (linha tracejada), notamos que a melhoria na solução de 20% (linha sólida) se estabilizou significativamente mais cedo.

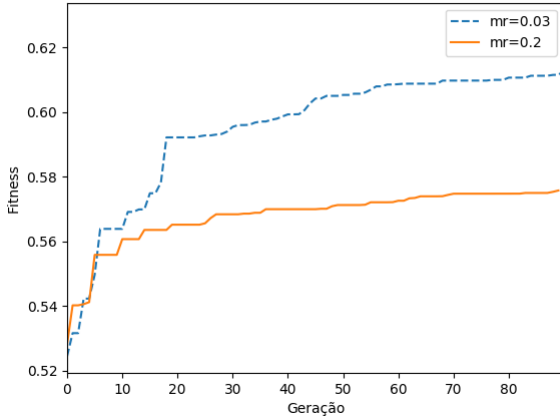


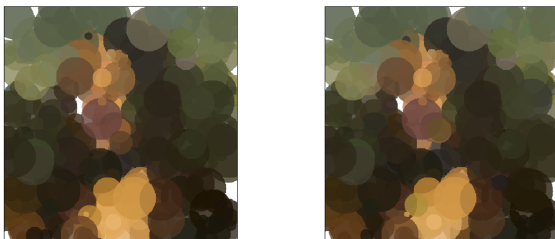
Figura 7. Comparação das curvas de fitness para uma taxa de mutação de 3% (linha tracejada) com a taxa de 20% (linha sólida).

Hipotetizamos que a taxa escolhida tenha sido alta demais e estaria causando o descarte de boas soluções e dificultando a convergência do algoritmo. Para trabalhos futuros, iremos procurar por outros valores de taxa de mutação que tragam melhores resultados.

F. Modificando o critério de parada

Nesta seção experimentamos com o critério de parada que anteriormente era apenas definido pelo número de gerações.

Dado que no experimento anterior cada geração demorava cerca de 9,2 segundos para ser completada, colocamos o critério de 1000 gerações ou 15 minutos sem atualização da melhor solução encontrada. A execução cumpriu o critério após a 362ª geração após ficar 15 minutos sem melhorias em sua fitness de 0.618.



(a) Geração 90

(b) Geração 362

Figura 8. Comparação entre a melhor solução encontrada com 90 gerações (à esquerda) e com 362 gerações (à direita).

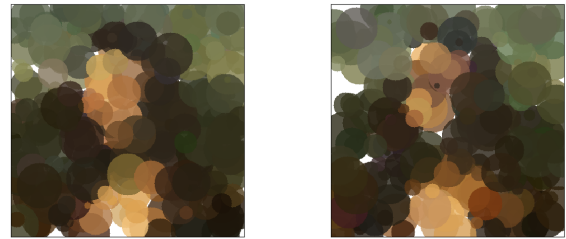
No entanto, no mesmo experimento, a 90ª geração alcançou fitness de 0.612 em 4 vezes menos tempo. A figura 8 nos mostra a comparação entre os resultados obtidos com 90 (à esquerda) e 362 gerações (à direita), nos dando um resultado que sugere que podemos realizar experimentos com menos gerações, economizando custo computacional, sem grandes prejuízos à solução obtida. Prosseguimos então a realizar testes com apenas 90 gerações.

G. Modificando o método de seleção

Até então, vínhamos usando a técnica de seleção por torneio. Ou seja, dada a taxa de seleção de 90%, a cada geração, os 90% melhores indivíduos eram selecionados para realizarem cruzamento.

Nesta seção, implementamos o método da roleta: A cada seleção de indivíduo para reprodução, todos os indivíduos possuem uma probabilidade de serem selecionados. No entanto, indivíduos com fitness maiores possuem também maiores chances de serem escolhidos.

O melhor indivíduo gerado após 90 gerações selecionadas por roleta pode ser visto na figura 9b.



(a) Seleção por torneio

(b) Seleção por roleta

Figura 9. Comparação entre os melhores indivíduos dos experimentos com seleção por torneio (à esquerda) e seleção por roleta (à direita).

A fitness da seleção por roleta porém não demonstrou melhora em relação ao método por torneio (figura 10). Decidimos então por prosseguir com o método de seleção por torneio.

H. Modificando o método de substituição

É possível também modificar o método de substituição. Atualmente, a cada geração com N indivíduos, geramos N filhos. Para escolher a próxima geração, pais e filhos competem entre si e os N indivíduos com maior fitness irão fazer parte da próxima geração. Em outras palavras, com a técnica que estamos utilizando agora é possível que um indivíduo criado na primeira geração permaneça até o fim.

Neste experimento, propusemos testar o método de substituição total: a cada geração todos os N indivíduos são substituídos pelos N filhos gerados. Uma amostra da evolução pode ser vista na figura 11.

No entanto, o método não se mostrou promissor. Assim como pode ser notado na figura 11, a modificação apresentou quedas de performance em relação ao que tínhamos com

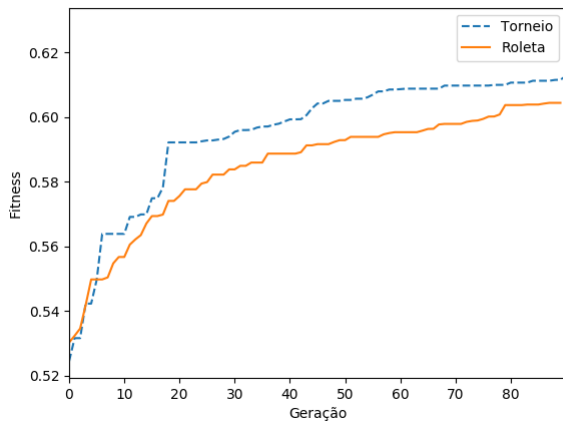
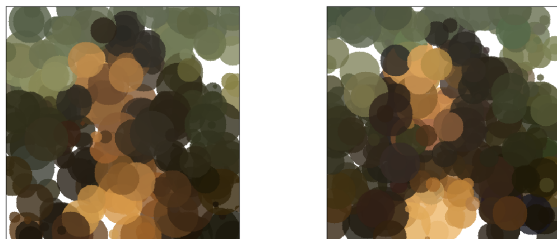


Figura 10. Comparação das curvas de fitness para a seleção por torneio (linha tracejada) e seleção por roleta (linha sólida).



(a) Geração 0

(b) Geração 90

Figura 11. Comparação entre os melhores indivíduos da geração 0 e geração 90 substituindo todos os indivíduos a cada geração.

substituição parcial. A geração 90 apresentou fitness 0.561, uma perda de desempenho significativa perto dos 0.613 do método anterior.

IV. CONCLUSÃO

Neste projeto exploramos o problema de aproximar uma imagem com um conjunto de círculos utilizando algoritmo genético. O nosso melhor resultado foi obtido com 20 indivíduos na população, critério de parada de 90 gerações, seleção por torneio dos 90% melhores indivíduos, crossover por seleção dos genes dos pais com probabilidade $p = 0.5$, técnica de mutação baseada em soma de atributos de um novo círculo, criação da nova geração selecionando os melhores indivíduos entre os pais e filhos e 3% de taxa de mutação. Os métodos de seleção por roleta e substituição total se mostraram ineficazes. No entanto, modificar a taxa de mutação, o tamanho da população e o critério de parada pode ser promissor dado uma escolha mais refinada dos parâmetros.

Dentre as principais limitações de nosso trabalho, podemos citar o grande viés adicionado por inicializar as cores dos círculos com a cor original da imagem. Outro ponto a ser

considerado é o alto custo computacional necessário para calcular a *fitness* dado que é necessário desenhar todos os N círculos numa imagem para então calcular a função de distância da imagem original. É importante enfatizar também que cada experimento apresentado neste projeto foi realizado apenas uma vez. Idealmente, deveríamos executá-los N vezes e apresentar a média dos resultados obtidos. A apresentação de dados cientificamente mais confiáveis e reproduzíveis pode ser explorada num trabalho futuro.

Outros experimentos interessantes para uma continuação desse trabalho pode ser testar outras funções de *fitness*, principalmente uma que penalizasse mais severamente cores distantes da imagem original. Poderíamos também aproximar a imagem por polígonos, como proposto por Johansson [1], que deve trazer uma aproximação mais precisa e com mais similaridade.

REFERÊNCIAS

- [1] R. Johansson, "Genetic programming: Evolution of mona lisa," <https://rogerjohansson.blog/2008/12/07/genetic-programming-evolution-of-mona-lisa/>. 5