

ENG1456 - Algoritmos Genéticos - Trabalho

1

Aluno: Matheus Carneiro Nogueira - 1810764

Professora: Karla Figueiredo

Sumário

1	Reproduzindo Resultados	1
2	GAP ideal	3
3	Taxas Crossover e Mutação	4
4	Tamanho da População	5
5	Convergência	6
6	Crossover	8
7	Normalização Linear	10
8	Gerais	11

Resumo

Este documento consiste no relatório do trabalho 1 do módulo de Algoritmos Genéticos da disciplina ENG1456 da PUC-Rio. O objetivo deste trabalho é estudar diferentes modelos de Algoritmos Genéticos para a tentativa de otimização da função $F6$ apresentada em sala. Foi utilizado o programa *ICADEMO* para gerar os modelos pedidos nos enunciados. Em todas as figuras do *ICADEMO* em que há mais de uma curva plotada, as configurações são referentes apenas à última curva. Foram consultados os materiais de aula, o livro [1] e outros materiais devidamente referenciados.

1 Reproduzindo Resultados

Enunciado: Variando os parâmetros, execute Algoritmos Genéticos de modo a obter resultados semelhantes aos apresentados no livro texto. Os parâmetros usados no livro se encontram na tabela abaixo. Compare as curvas referentes à média de 20 rodadas de cada GA. Incluir dois gráficos: um com GA1-1, GA2-1 e GA2-2 e outro com GA 2 -3 e GA2-4. Utilizar somente one-point-crossover.

GA	População	Total.Ind.	Crossover	Mutação	NormLinear	Elitismo	Stead-State
1-1	100	4000	65%	0.8%	NÃO	NÃO	NÃO
2-1	100	4000	65%	0.8%	Max=100/Min=1	NÃO	NÃO
2-2	100	4000	65%	0.8%	Max=100/Min=1	SIM	NÃO
2-3	100	4000	65%	0.8%	Max=100/Min=1	NÃO	C/Duplicados
2-4	100	4000	65%	0.8%	Max=100/Min=1	NÃO	S/Duplicados

Figura 1: Tabela com as especificações dos modelos

As duas imagens abaixo exibem os 5 GA's solicitados no enunciado. Em ambas as figuras também foi plotada a curva da busca aleatória para fins de comparação.

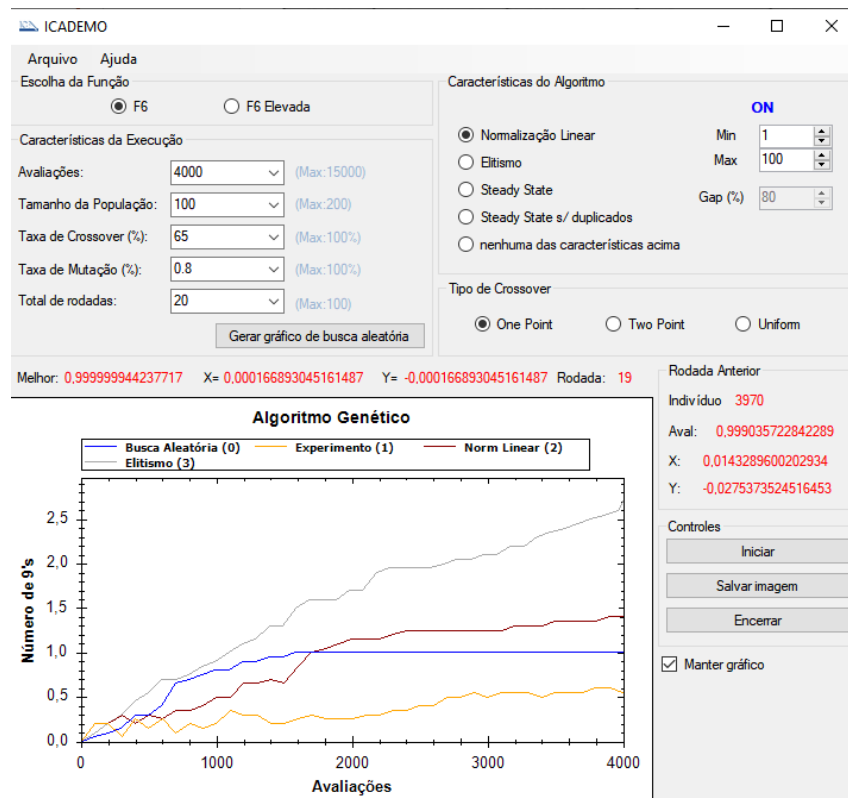


Figura 2: Experimento 1: GA=1-1, Norm. Linear: GA=2-1, Elitismo: GA=2-2

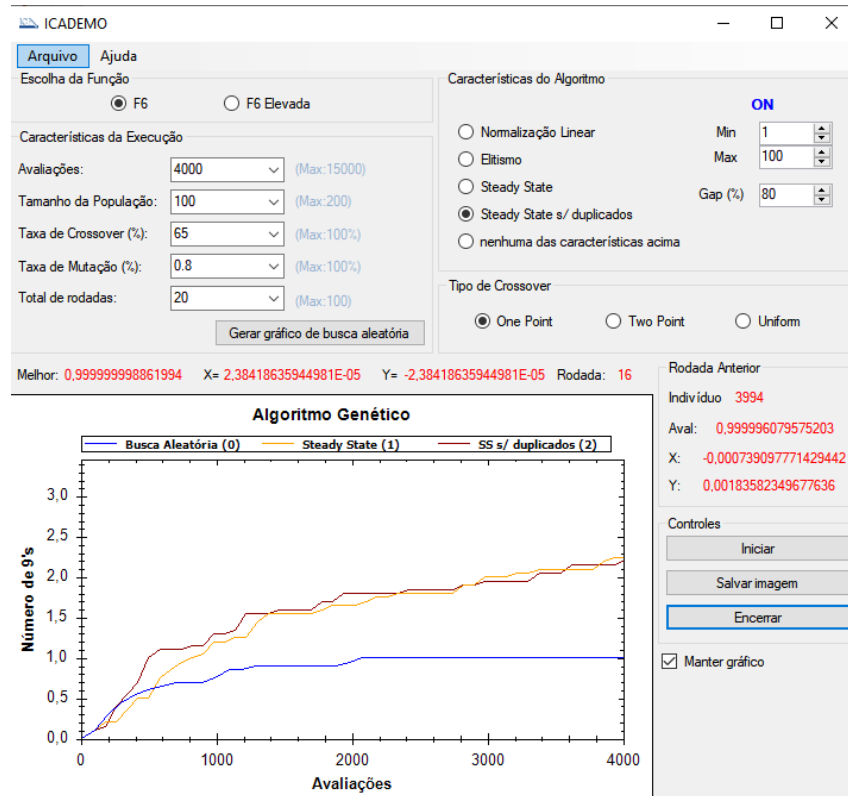


Figura 3: Steady State 1: GA=2-3, SS s/ duplicados: GA=2-4

Podemos, facilmente, perceber alguns detalhes importantes. Primeiramente, o GA 1-1 da figura 2 se mostrou pior que a busca aleatória. Isso não é surpresa, uma vez que esse modelo não usufrui de nenhum operador além do crossover e mutação. Ao analisar as demais curvas que utilizam esses recursos, é de se esperar que o resultado seja melhor que uma busca aleatória, senão não haveria motivo para utilizar um algoritmo genético.

Uma simples normalização linear já foi suficiente para que o GA 2-1 apresentasse melhor resultado que a busca aleatória.

A utilização do elitismo é importante por garantir que, durante a evolução, o melhor indivíduo de $t + 1$ seja sempre melhor ou igual ao melhor de t . Isso se percebe pelo fato da curva cinza da figura 2 ser monotônica, isto é, sempre crescente. Esse fato é suficiente para explicar o melhor desempenho desse modelo em relação aos anteriores.

Ambos os Steady States foram utilizados com GAP=80%, o que quer dizer que **COMPLETAR COM INTERPRETAÇÃO DO GAP E COM DIFERENÇA DE COM E SEM DUPLICADOS**

Comparando as duas figuras, podemos perceber que, dentre os 5 GA's apresentados e a busca aleatória, aquele que utiliza elitismo junto com normalização linear (GA2-2) apresenta o melhor resultado.

2 GAP ideal

Enunciado: Para os GAs que utilizam steady-state, determine o GAP (número de indivíduos substituídos a cada ciclo) ideal. Para isso, use um incremento de 5 indivíduos a cada tentativa, começando com um GAP=5. Não entregue os gráficos referentes aos testes de GAP.

O melhor resultado para cada GAP é exibido nas tabelas as seguir.

GAP	5	10	15	20	25	30	35	40
Num de 9's								
GAP	45	50	55	60	65	70	75	80
Num de 9's								
GAP	85	90	95	100				
Num de 9's								

Tabela 1: Tabela com melhor resultado para cada GAP Steady State com duplicados

GAP	5	10	15	20	25	30	35	40
Num de 9's								
GAP	45	50	55	60	65	70	75	80
Num de 9's								
GAP	85	90	95	100				
Num de 9's								

Tabela 2: Tabela com melhor resultado para cada GAP Steady State sem duplicados

Como explicado na seção anterior, GAP significa a porcentagem da população que será trocada. Logo, um GAP de 0% significaria que nenhuma parcela da população seria trocada, ou seja, não haveria Steady State.

COMPLETAR INTERPRETAÇÃO

3 Taxas Crossover e Mutação

Enunciado: Verifique o que acontece quando se roda o GA2-1 20 vezes com taxa de crossover muito baixa (pouca recombinação em torno de 10%) e alta taxa de mutação (muitas mudanças aleatórias em torno de 80%). Imprima o resultado (um gráfico), compare com o resultado do GA2-1 obtido no item 1 e explique brevemente o que acontece.

A figura 4 abaixo exibe o GA com as taxas pedidas.

Sabemos que a altas taxas de mutação aumentam a aleatoriedade do processo de evolução e que baixas taxas de crossover tornam a busca pela seleção ótima mais lenta, uma vez que há menor combinação dos progenitores.

Ao analisar a figura 4 podemos perceber uma alta volatilidade na qualidade da solução, isto é, grande variação entre picos e vales. Esse fato é resultado da alta taxa de mutação, uma vez que ela atrapalha a evolução normal do algoritmo ao adicionar aleatoriedade. A baixa taxa de mutação, por sua vez, é percebida de forma mais sutil pela falta de caráter crescente da curva. Como são feitos poucos cruzamentos, utiliza-se pouco os bons progenitores para gerar bons filhos, o que diminui a qualidade da evolução da solução.

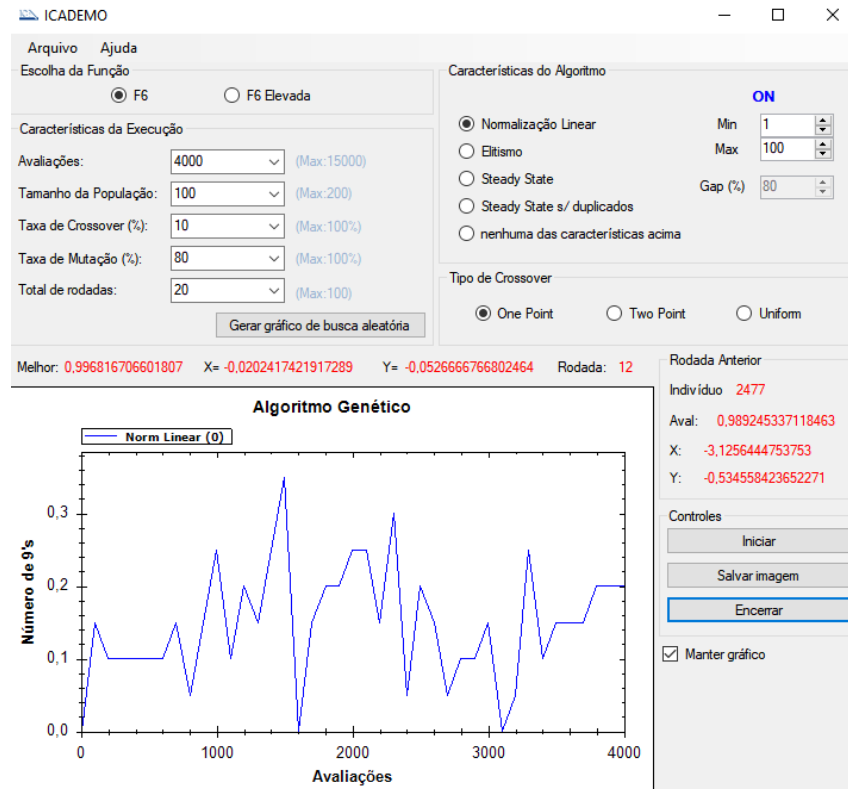


Figura 4: GA2-1 com taxas de crossover 10% e mutação 80%

4 Tamanho da População

Enunciado: Analise o efeito do tamanho da população, obtendo as curvas de desempenho do GA2-2 (20 rodadas) para vários tamanhos de população (ex: 20, 50, 100, 150) e sempre com o mesmo número de gerações (total de indivíduos variável). Imprima as curvas para e tire conclusões sobre o efeito do tamanho da população no desempenho do algoritmo genético.

Antes de partirmos para a avaliação dos diferentes GA's, devemos lembrar como definir o tamanho da população em relação do número de gerações e do número total de indivíduos (avaliações).

Esses parâmetros relacionam-se de acordo com a seguinte expressão:

$$avaliacoes = num_geracoes \times tamanho_populacao$$

A partir dessa relação e dos valores solicitados pelo enunciado para os tamanhos de população, mantendo o número de gerações igual a 40, chegamos aos seguintes valores:

tam_pop	20	50	100	150
avaliações	800	2000	4000	6000

A figura abaixo exhibe os resultados para esses valores de população.

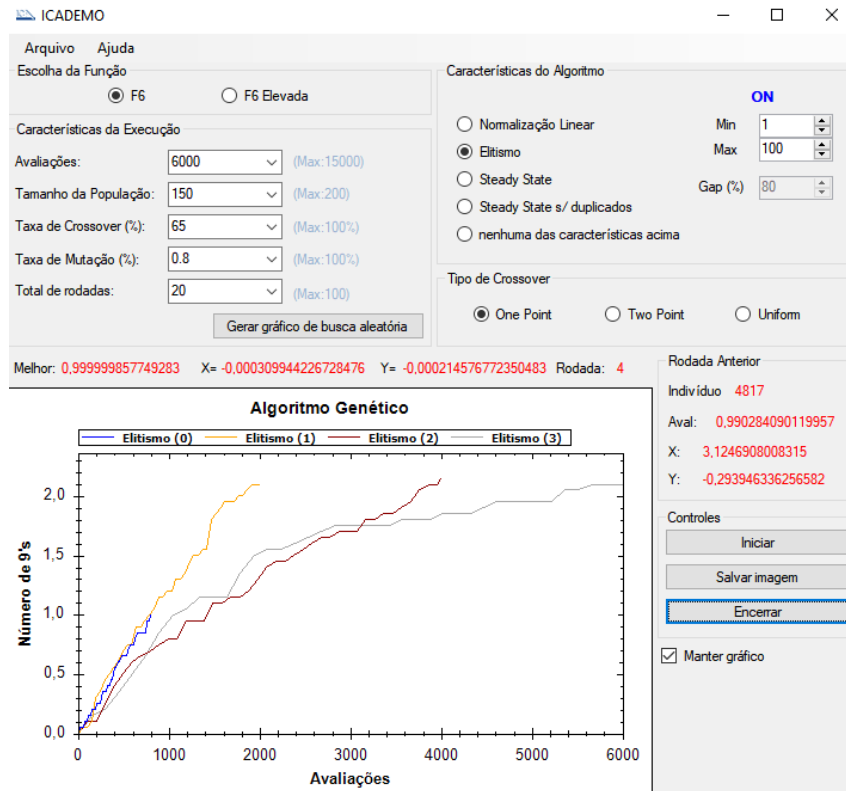


Figura 5: Elitismo (0) = tam_pop=20; Elitismo (1) = tam_pop=50; Elitismo (2) = tam_pop=100; Elitismo (3) = tam_pop=150;

Podemos, a partir da análise da imagem, concluir que uma população de 20 indivíduos com 40 gerações é pouco para a evolução do algoritmo. O que nos indicará esse fato é a curva amarela da figura 5, que representa a população de 50, ser praticamente a mesma da curva azul, população de 20, até o número de avaliações existentes na curva azul. Isso nos indica que, se houver mais do que 20 indivíduos por população, o algoritmo seria capaz de evoluir mais em direção à solução ótima.

Por outro lado, para as populações de 50, 100 e 150, o resultado final da solução é muito similar, ficando em torno de 2.1 noventa. Isso nos revela que aumentar o número de indivíduos da população mantendo a mesma quantidade de gerações não infere, necessariamente, na melhora da solução. Podemos justificar isso da seguinte maneira: mais indivíduos por população aumenta o paralelismo da busca, mas, em dado momento, o algoritmo já possui indivíduos suficientes buscando a solução ótima e precisaria, apenas, de mais tempo (gerações) para encontrá-la.

5 Convergência

Enunciado: Repita o GA2-1 e o GA2-2 (20 rodadas cada) modificando apenas o total de indivíduos criados para o 10000. Imprima as curvas em dois gráficos separados, um para o GA2-1 e outro para o GA2-2, e verifique se é vantajoso todo esse esforço computacional, em outras palavras, determine o número de indivíduos para o qual cada algoritmo converge.

A figura abaixo exibe os resultados dos GA's solicitados no enunciado.

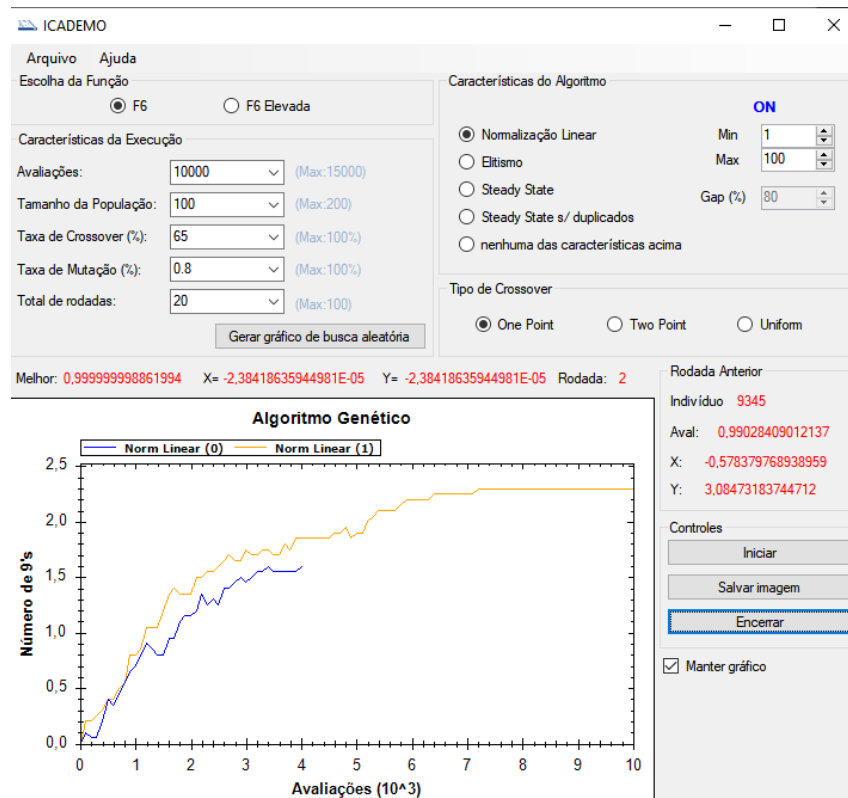


Figura 6: GA 2-1 com avaliações de 4000 (azul) e 10000 (amarelo)

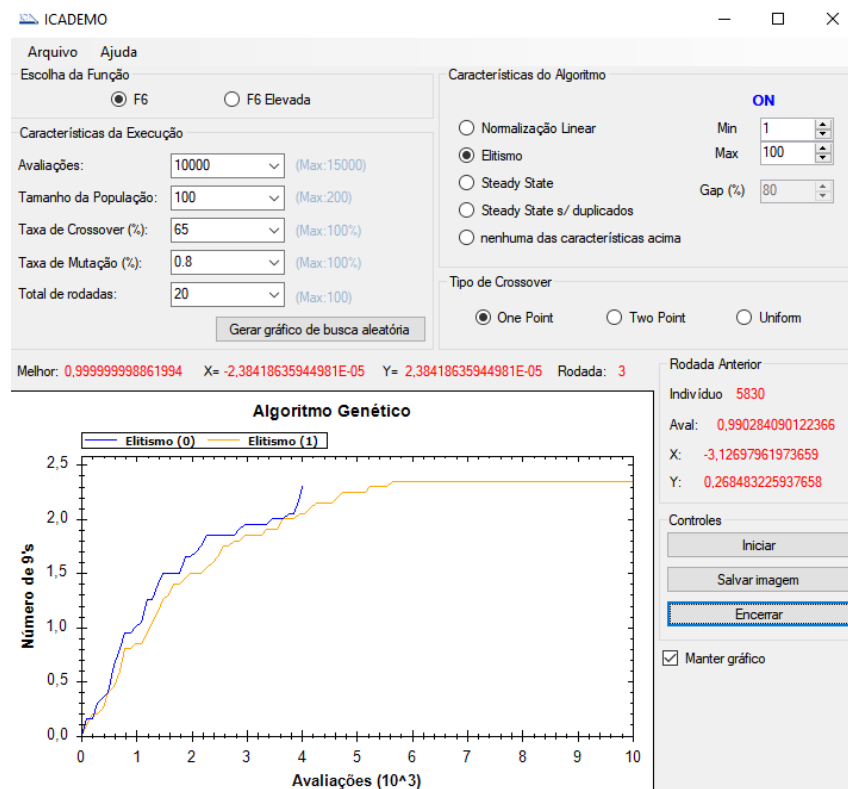


Figura 7: GA 2-2 com avaliações de 4000 (azul) e 10000 (amarelo)

A análise de convergência dos GA's fornece conclusões distintas para o caso com e sem

elitismo. Começamos pelo GA 2-1, com normalização linear e sem elitismo.

Nesse caso, percebe-se pela figura 6 que não há uma convergência para o mesmo valor de avaliação, isto é, o algoritmo com mais avaliações (curva amarela) apresentou um desempenho melhor que o com menos (curva azul). Isso pode ser explicado pelo fato de, em elitismo, não garantirmos que a curva de avaliação será sempre crescente. Sendo assim, precisamos de mais tempo, ou seja, mais avaliações e gerações, para chegar a um resultado mais próximo do ótimo, o que é evidenciado pela diferença entre as curvas.

Para o GA 2-2, com elitismo e normalização linear, a conclusão é outra. Com a utilização do elitismo, percebemos convergência para o mesmo valor de número de iterações. Note que a curva amarela, após alcançar o valor mais alto da curva azul (2.6), permanece praticamente constante. Com esse operador, geramos uma curva monotônica e convergimos para um valor bom mais rapidamente. Isso nos mostra que aumentar a população de um GA com elitismo não gera melhores resultados, uma vez que esse operador já é suficiente para produzir uma solução boa. Aumentar o esforço computacional, nesse caso, é desnecessário e, dito isso, não recomendável.

6 Crossover

Enunciado: Compare o efeito dos 3 tipos de crossover disponíveis na ferramenta, executando o GA2-1 (s/ elitismo) e o GA2-2 (c/elitismo) com apenas 2500 indivíduos (20 rodadas) para cada tipo de crossover, usando taxa de crossover 80%. Imprima as curvas em dois gráficos separados, um para o GA2-1 e outro para o GA2-2, e tire conclusões a respeito da característica conservadora/destrutiva de cada crossover.

As figuras abaixo exibem os gráficos para os 3 tipos de crossover para os GA's 2-1 e 2-2.

COMENTAR SOBRE CROSSOVER!!!

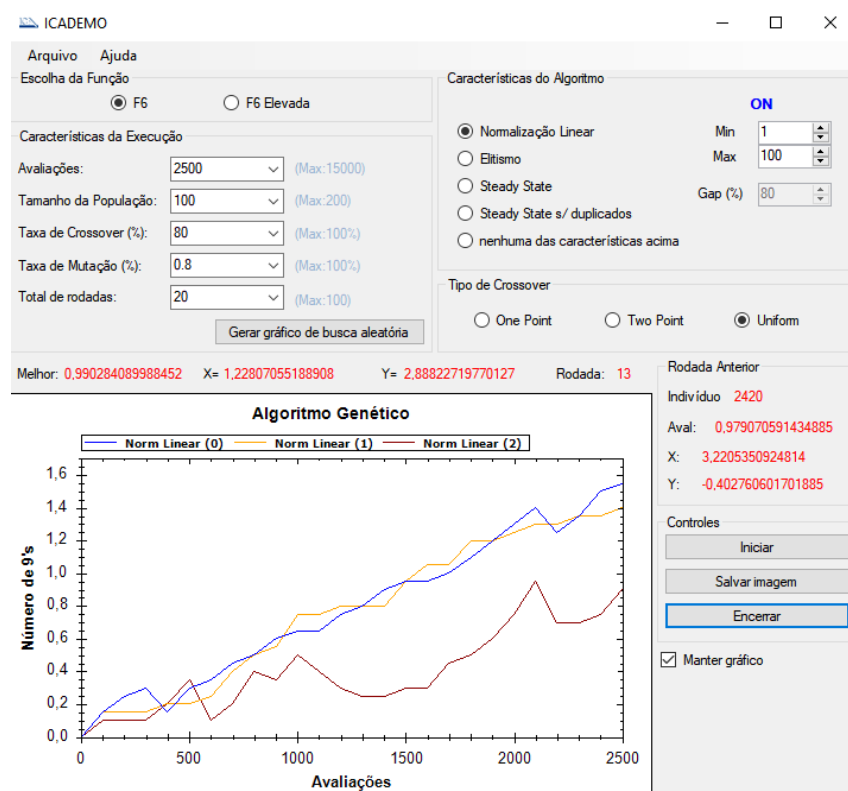


Figura 8: GA 2-1; Norm Linear (0) = crossover de um ponto ; Norm Linear (1) = crossover de dois pontos ; Norm Linear (2) = crossover uniforme

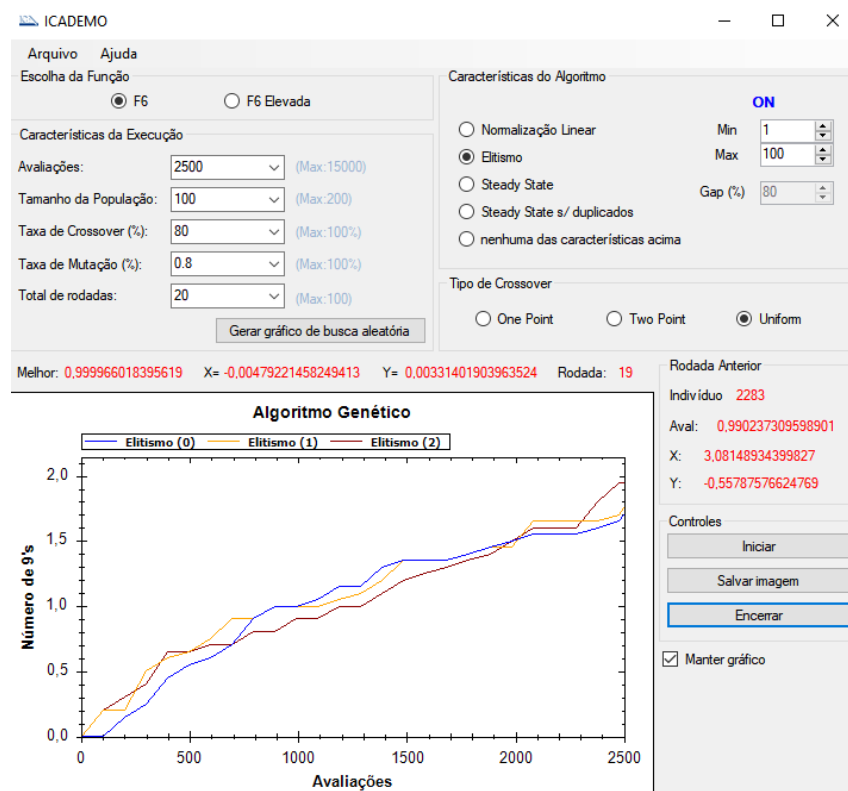


Figura 9: GA 2-2; Elitismo (0) = crossover de um ponto ; Elitismo (1) = crossover de dois pontos ; Elitismo (2) = crossover uniforme

7 Normalização Linear

Enunciado: Repita o GA2-3COM $\text{gap} = 75$ para vários valores de máximo. Verifique o que acontece quando o valor de máximo aumenta e diminui (avale para os valores 10, 50, 100, 200, 300). Imprima as curvas em apenas um gráfico e tire breves conclusões.

Sabemos que a normalização linear trabalha com aptidão A_i definida como se segue:

$$A_i = \min + \frac{\max - \min}{\text{pop_size} - 1} \times (i - 1)$$

Dito isso, quanto maior o valor de *máximo*, maior o valor dado à aptidão do indivíduo, o que, por sua vez, aumenta a pressão seletiva sobre os melhores indivíduos. É de se esperar, portanto, que tanto valores pequenos demais quanto grandes demais não se mostrem interessantes para fins de otimização, pois pouca pressão fará com que indivíduos ruins se reproduzam e pressão demais fará com que poucos indivíduos sejam selecionados para reprodução. A figura 10 exibe os resultados para valores de máximo de 10, 50, 100, 200, 300.

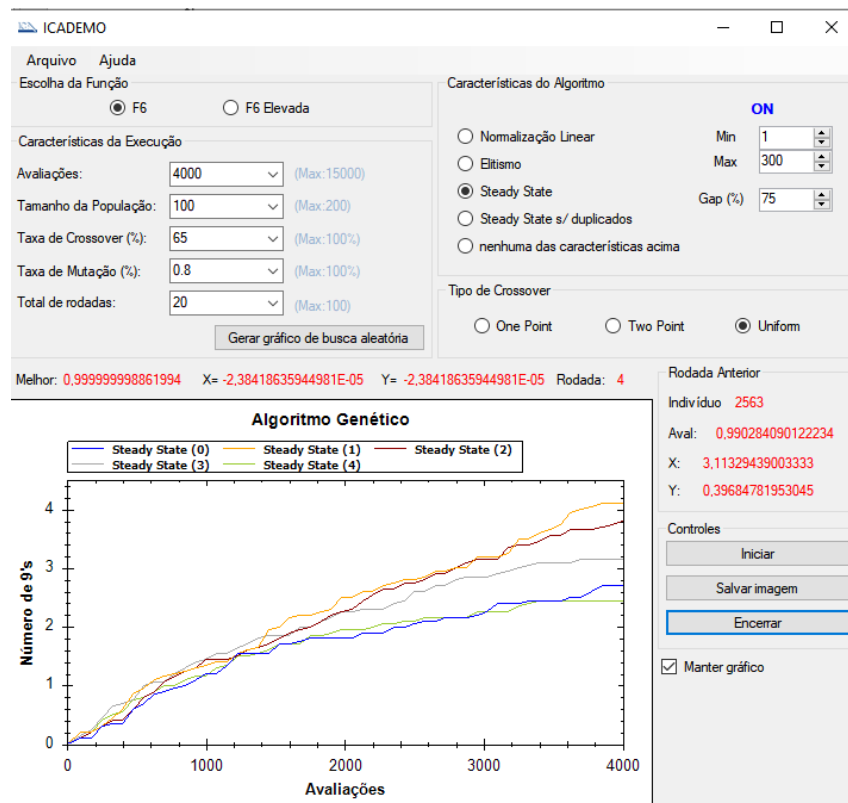


Figura 10: Steady State (0) = max 10; Steady State (1) = max 50; Steady State (2) = max 100; Steady State (3) = max 200; Steady State (4) = max 300;

Pela análise da figura, percebemos algo similar ao previsto. Os valores 10 e 300 para máximo, que são o menor e o maior de nossa lista de valores, apresentaram os piores resultados. O motivo é, justamente, o que foi comentado acima: pressão demais ou pressão de menos não é vantajoso para a evolução da solução, uma vez que o primeiro gera filhos inadequados e o último exerce pressão demais. Seguindo esse raciocínio, é

trivial entender porque os melhores valores de máximo, de acordo com a figura, foram os valores 50 e 100.

8 Gerais

Enunciado: Fazendo variações nos parâmetros e técnicas disponíveis no GADEMO, estude livremente o efeito de cada umdestes no desempenho de algoritmos genéticos. Destaque e explique uma importante constatação.

Referências

- [1] L. Davis. *Handbook of Genetic Algorithms*. VNR Computer Library VNR Computer Library. Van Nostrand Reinhold, 1991.