

Tarefa 10 de técnicas de otimização

Gabriel Ferreira Vieira (202106840041)

Resumo— Resolução referente a tarefa número 10 da disciplina de técnicas de otimização ministrada pelo professor Roberto Celio Lima de Oliveira, no curso de Engenharia da Computação na Universidade Federal do Pará. A resolução tratou de implementar o algoritmo genético seguindo os parâmetros definidos na atividade e realizar uma análise comparativa dos resultados obtidos de acordo com as diferentes implementações realizadas.

I. QUESTÃO

Orientações:

Implementar algoritmo genético (AG), com representação binária, para encontrar o máximo da função Schaffer's , conhecida como F6.

$$f_6(x, y) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1 + 0.001 \times (x^2 + y^2))^2}$$

Parâmetros definidos:

- Domínio das duas variáveis [-100, +100].
- Precisão de 5 casas decimais para as duas variáveis.
- Representação Binária para as duas variáveis da F6.
- Número de bits necessários para representar cada uma das variáveis: 25 bits.
- Taxa de Cruzamento igual a 0,85.
- Taxa de Mutação igual a 0,01.
- Tamanho da população igual a 200 indivíduos.
- Método de Seleção por meio da Roleta proporcional a medida de aptidão do indivíduo.
- Método de Cruzamento binário com um ponto de corte.
- Parar o AG em 500 gerações.

A. Implementação

O algoritmo genético foi implementado na linguagem de programação "python" seguindo as orientações definidas na atividade, o código produzido e os resultados obtidos estão disponíveis em [1]. A representação binária foi realizada utilizando strings. A primeira geração de indivíduos é gerada aleatoriamente bit a bit e são calculados os valores reais de x e y, o valor na função objetivo e tamanho da partição de cada indivíduo.

No cruzamento implementado, 2 progenitores são selecionados a partir do sorteio de uma variável aleatória de distribuição uniforme no intervalo fechado de 0 a 1, a partir do número resultante ou os pais se repetem na geração seguinte ou é realizado cruzamento por ponto de corte, isso ocorre de acordo com o valor de um número aleatório gerado de uma distribuição uniforme no intervalo fechado de 0 a 1 para cada cruzamento, sendo avaliado com o limiar definido pela taxa de cruzamento. Acima do limiar 0.85 ocorre repetição dos progenitores para a geração seguinte e, no intervalo oposto ocorre o cruzamento por ponto de corte. A implementação atua de forma que cada cruzamento dê origem a dois indivíduos

para a geração seguinte e, o número de cruzamentos preserva o número de indivíduos na população ao longo das gerações.

A mutação implementada é avaliada bit a bit de cada indivíduo em sua string binária logo após ser gerado no cruzamento. Sua ocorrência em cada bit é condicionada a um número aleatório, gerado de uma distribuição uniforme de intervalo fechado de 0 a 1, avaliado sub o limiar definido pela taxa de mutação: 0.85. Acima do limiar não corre mutação no bit avaliado, já no intervalo oposto ocorre inversão do bit. As taxas de mutação e cruzamento foram mantidas fixas ao longo de todas as gerações.

Após realizar a mutação de cada indivíduo e serem completadas as listas binárias de uma geração, foi calculada a conversão de x e y de cada indivíduo para a representação real. Isso foi realizado convertendo as strings de 0s e 1s para representação binária, depois para inteira e em seguida realizada a conversão de inteira para real com precisão de 5 casas decimais usando a equação a seguir, onde max e min são o limite superior e inferior do espaço de busca, 100 e -100, respectivamente. k é 25 para obter precisão de 5 casas decimais.

$$\begin{aligned} Real &= \frac{int \times (max - min)}{2^k - 1} + min \\ Real &= \frac{int \times (100 - (-100))}{2^{25} - 1} + (-100) \end{aligned}$$

A partir dos valores de x e y real foi calculado o valor de cada indivíduo na função objetivo z_i e, após calcular de toda a geração, foi calculada a parcela de cada um na roleta proporcionalmente, seguindo a equação a seguir. Onde a fatia da roleta de cada indivíduo é determinada pelo seu valor na função objetivo sobre o somatório do valor na função objetivo de todos os indivíduos da população na geração. Vale destacar que o resultante da função objetivo $f_6(x, y)$ é sempre um número não negativo, por isso é possível utilizar o somatório de seu resultado na população para atribuir as parcelas proporcionais da roleta de cada indivíduo.

$$Fatia = \frac{z_i}{\sum_{n=1}^{Populacao} z_n}$$

Os dados de cada população (representação binária de x, representação binária de y, representação real de x, representação real de y, o valor da função objetivo, a respectiva partição na roleta de cada indivíduo) foram armazenados em dataframes pandas e, a pedido da atividade, foram organizadas as informações de cada população das gerações múltiplas de 10 em diferentes arquivos .csv. Na figura 1 é possível observar os dados da população da geração 1, na figura 1 é possível observar as informações da população da geração de número 50, pode-se perceber que os indivíduos estão começando a ficar bem parecidos entre si. Os arquivos .csv das gerações

múltiplas de 10 estão disponíveis em [1], bem como seus respectivos gráficos dos indivíduos dispostos no espaço de busca.

Fig. 1. .csv da primeira geração.

Fig. 2. .csv da geração 50.

As figuras de 3 a 8 mostram os indivíduos no espaço de busca ao longo das gerações 1, 10, 20, 30, 40 e 50. Pode ser observado o aumento da concentração espacial dos indivíduos, isso se deve a o ponto central ser o ponto de ótimo da função. A mutação binária implementada, por mutação aleatória de bit, permitiu dispersão considerável dos indivíduos no domínio definido, o restante das gerações manteve comportamento similar ao da 8, com a população próxima do ponto de máximo ($0, 0$), com valores de z variando.

O gráfico da evolução do melhor indivíduo ao longo das épocas pode ser visto na figura 9.

II. QUESTÃO

Orientações:

Utilizar o Algoritmo Genético (AG), com representação real, para encontrar o máximo da função Schaffer's , conhecida como F6, a seguir:

$$f_6(x, y) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1 + 0.001 \times (x^2 + y^2))^2}$$

Parâmetros definidos:

- Domínio das duas variáveis $[-100, +100]$.
 - Taxa de Cruzamento igual a 0,85.
 - Taxa de Mutação igual a 0,01.
 - Tamanho da população igual a 200 indivíduos.
 - Método de Seleção por meio da Roleta proporcional a medida de aptidão do indivíduo.
 - Método de Cruzamento Aritmético com $a = 0,35$.
 - Método de Mutação Randômica Não Uniforme, com rand = normal de média zero e variância a ser definida pelo aluno, optou-se por utilizar variância = 1.
 - Parar o AG em 500 gerações.

Geração 1

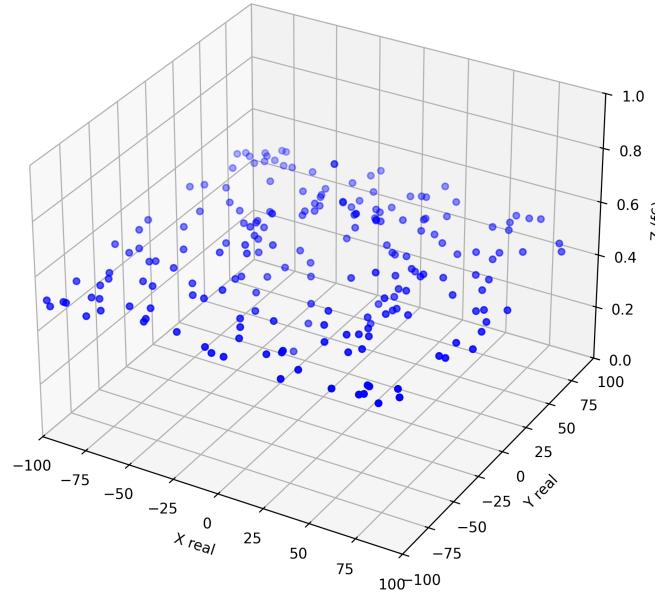


Fig. 3. Gráfico da geração 1 no espaço de busca.

Geração 10

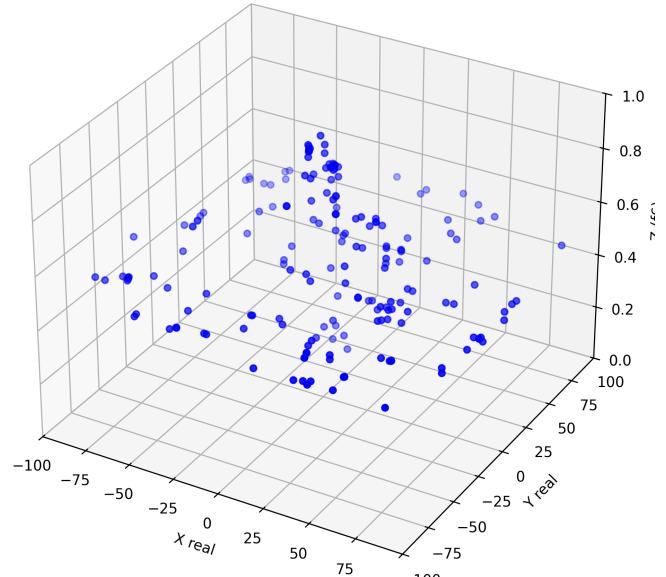


Fig. 4. Gráfico da geração 10 no espaço de busca.

A. Implementação

O algoritmo genético foi implementado na linguagem de programação "python" seguindo as orientações definidas na atividade, o código produzido e os resultados obtidos estão disponíveis em [1]. A primeira geração de indivíduos foi gerada aleatoriamente na representação de string binária, foram calculados os valores reais de x e y, o valor na função objetivo e tamanho da partição de cada indivíduo. No cruzamento implementado, 2 progenitores são selecionados a partir do

Geração 20

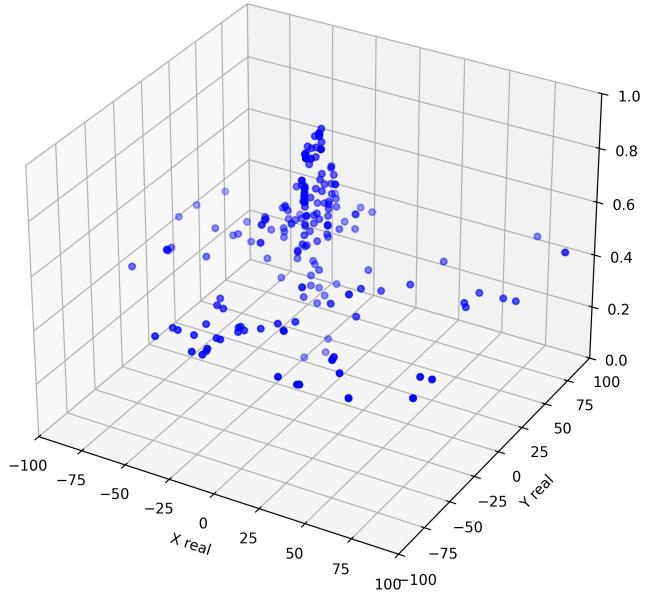


Fig. 5. Gráfico da geração 20 no espaço de busca.

Geração 40

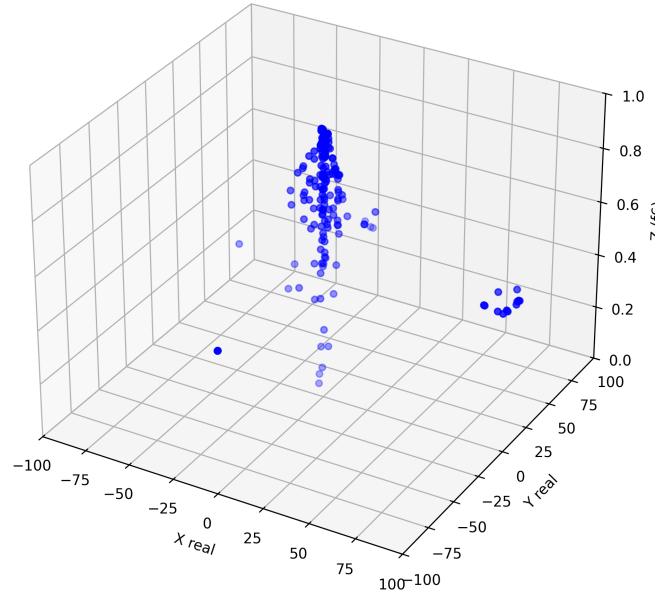


Fig. 7. Gráfico da geração 40 no espaço de busca.

Geração 30

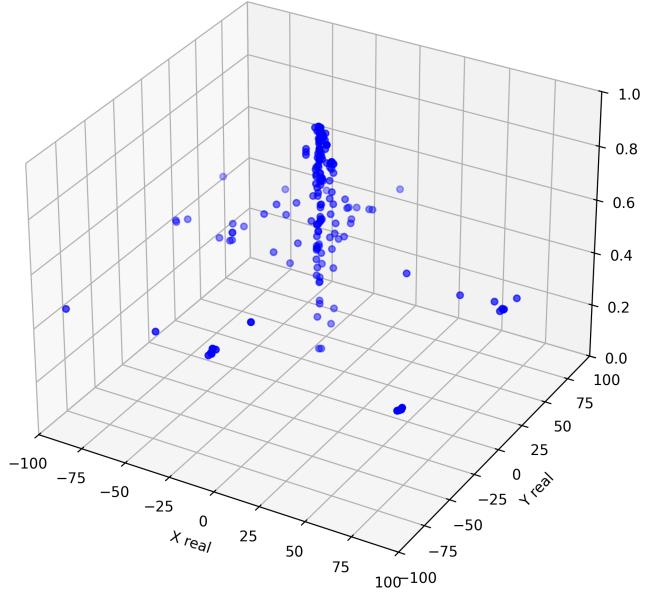


Fig. 6. Gráfico da geração 30 no espaço de busca.

Geração 50

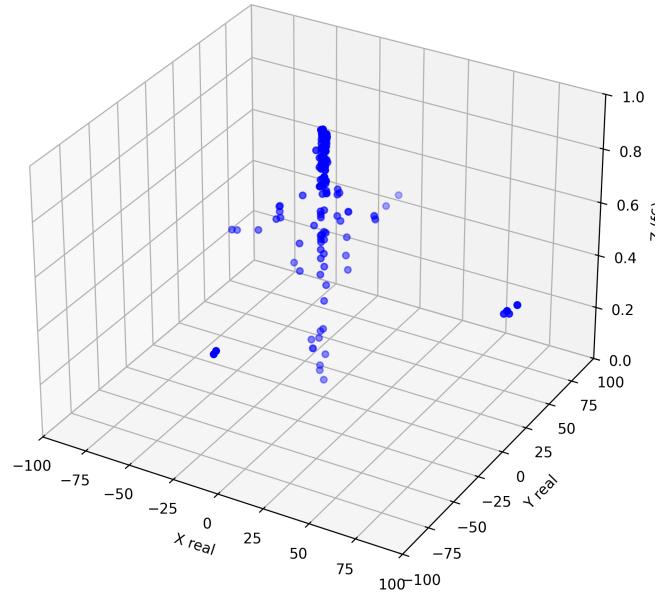


Fig. 8. Gráfico da geração 50 no espaço de busca.

sorteio de uma variável aleatória de distribuição uniforme no intervalo fechado de 0 a 1, a partir do número resultante ou os pais se repetem na geração seguinte ou é realizado cruzamento aritmético com $a = 0.35$, seguindo a equação a seguir. Essas equações foram realizadas para determinar x e y de maneira isolada.

$$\begin{aligned} Real_{filho1} &= a \times Real_{pai1} + (1 - a) \times Real_{pai2} \\ Real_{filho2} &= a \times Real_{pai2} + (1 - a) \times Real_{pai1} \end{aligned}$$

O cruzamento é determinado a partir do sorteio de um número aleatório gerado de uma distribuição uniforme no intervalo fechado de 0 a 1 para cada cruzamento, sendo avaliado com o limiar definido pela taxa de cruzamento. Acima do limiar 0.85 ocorre repetição dos progenitores para a geração seguinte e, no intervalo oposto ocorre o cruzamento aritmético. A implementação atua de forma que cada cruzamento dê origem a dois indivíduos para a geração seguinte e, o número de cruzamentos preserva o número de indivíduos na população

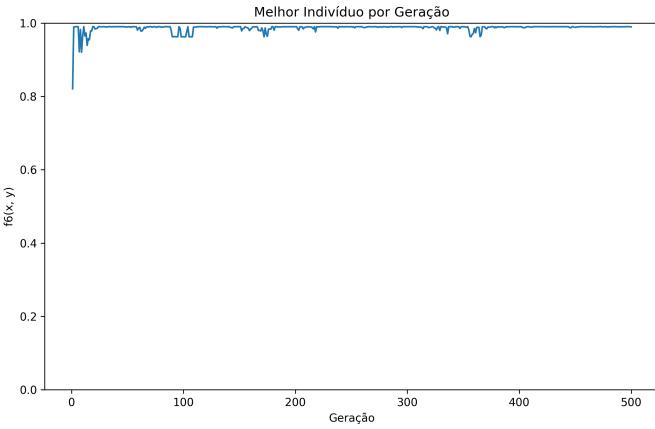


Fig. 9. Gráfico do valor de $f_6(x, y)$ do melhor indivíduo ao longo das 500 gerações.

ao longo das gerações.

A mutação implementada foi a randômica não uniforme, a distribuição utilizada foi uma distribuição normal de média 0 e variância igual a 1. A ocorrência da mutação está condicionada a avaliação do valor de uma variável aleatória, de distribuição uniforme de num intervalo fechado de 0 a 1, sob o limiar definido pela taxa de mutação. Acima de 0.01 não ocorre mutação e abaixo ocorre a randômica não uniforme. Na implementação, a ocorrência da mutação foi avaliada para cada variável de cada indivíduo de cada geração logo após o indivíduo ser gerado pelo cruzamento. Na mutação randômica não uniforme, o número sorteado pela distribuição definida é somado ao valor real que representa a característica, isso poderia ocasionar a saída das variáveis para fora do domínio determinado então, logo após a mutação, foi avaliada se alguma variável havia ultrapassado o domínio, se estivesse acima de 100 seria ajustada para 100, se estivesse abaixo de -100 seria ajustada para -100. As taxas de mutação e cruzamento foram mantidas fixas ao longo de todas as gerações.

A partir dos valores de x e y real foi calculado o valor de cada indivíduo na função objetivo z_i e, após calcular de toda a geração, foi calculada a parcela de cada um na roleta proporcionalmente, seguindo a equação a seguir. Onde a fatia da roleta de cada indivíduo é determinada pelo seu valor na função objetivo sobre o somatório do valor na função objetivo de todos os indivíduos da população na geração. Vale destacar que o resultante da função objetivo $f_6(x, y)$ é sempre um número não negativo, por isso é possível utilizar o somatório de seu resultado na população para atribuir as parcelas proporcionais da roleta de cada indivíduo.

$$Fatia = \frac{z_i}{\sum_{n=1}^{Populacao} z_n}$$

A pedido a pedido da atividade, foram plotados os gráficos das gerações múltiplas de 10 no espaço de busca, todos os gráficos estão disponíveis em [1], mas as figuras de 10 a 15 mostram a disposição da população das gerações 1, 10, 20, 30, 40 e 50 no espaço de busca.

O gráfico da evolução do melhor indivíduo ao longo das 500 épocas pode ser visto na figura 16.

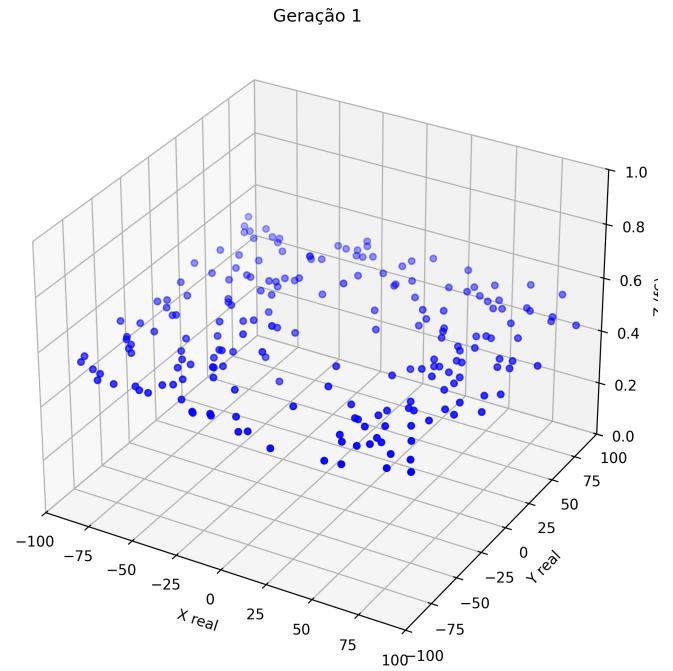


Fig. 10. Gráfico da geração 1 no espaço de busca.

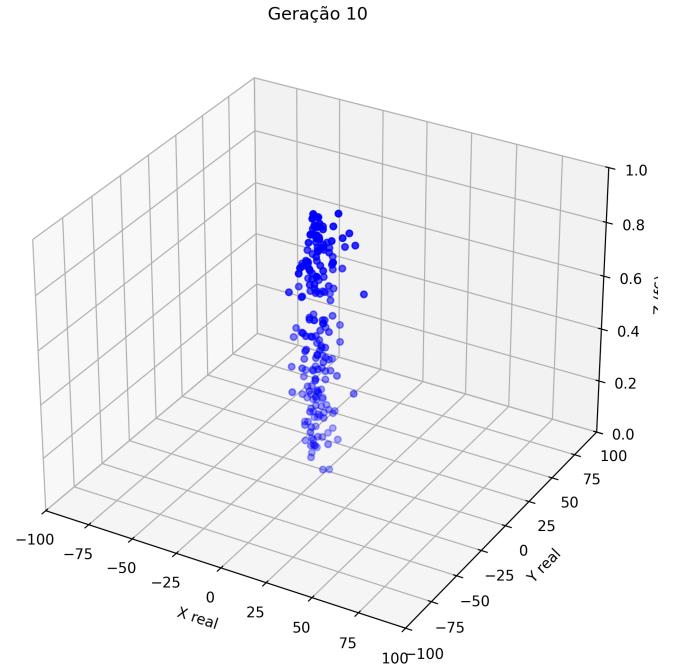


Fig. 11. Gráfico da geração 10 no espaço de busca.

III. QUESTÃO

Ao analisar os resultados obtidos pelo algoritmo genético nas duas implementações realizadas, onde as mudanças de implementação envolveram apenas do método de cruzamento e de mutação, foi observada maior dificuldade de convergência do algoritmo da questão 1 em comparação com a questão 2. A convergência do algoritmo está associada a quanto rápido a população dos indivíduos se aproximam pelo cruzamento e o quanto rápido divergem, nesse sentido se a convergência gerada

Geração 20

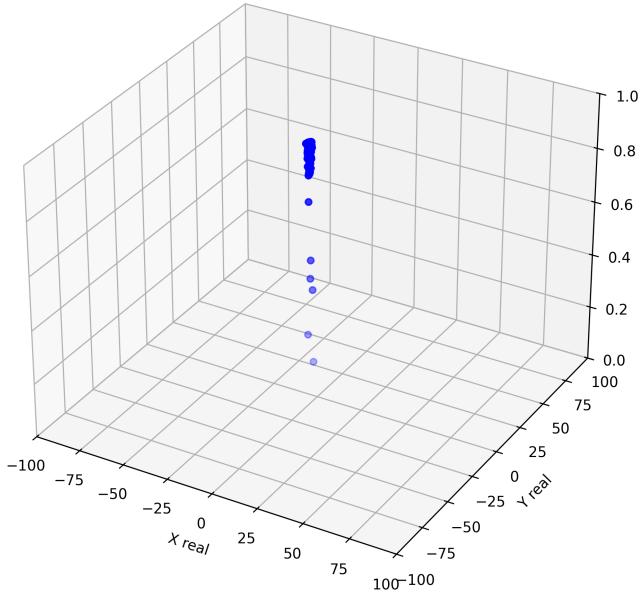


Fig. 12. Gráfico da geração 20 no espaço de busca.

Geração 40

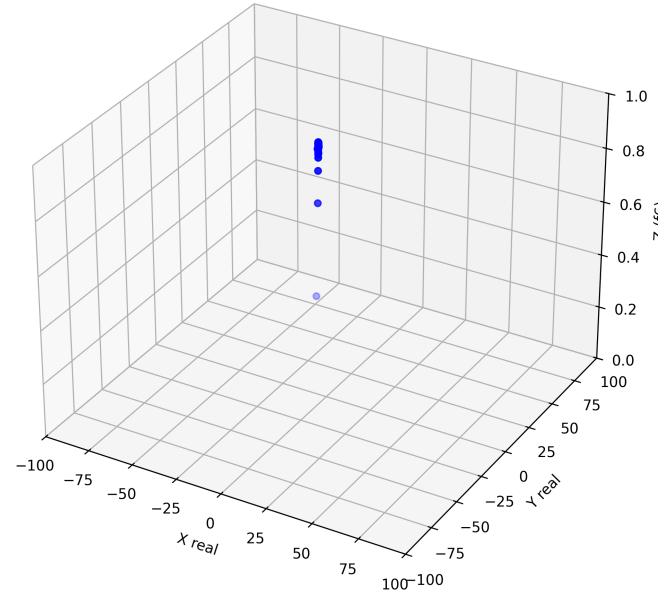


Fig. 14. Gráfico da geração 40 no espaço de busca.

Geração 30

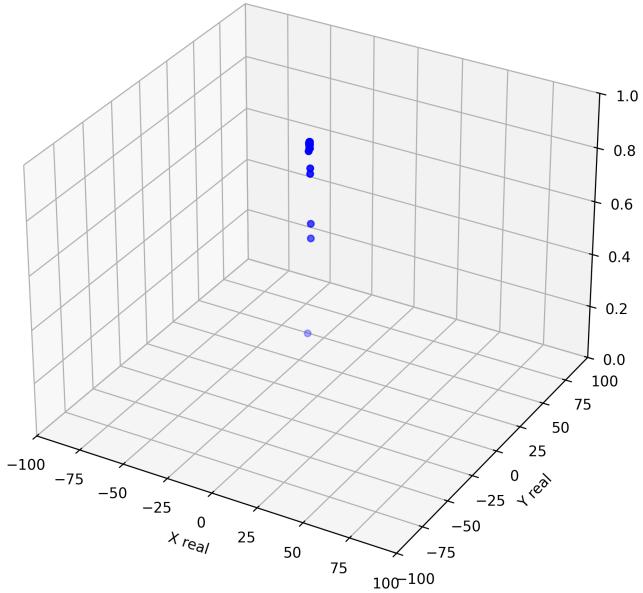


Fig. 13. Gráfico da geração 30 no espaço de busca.

Geração 50

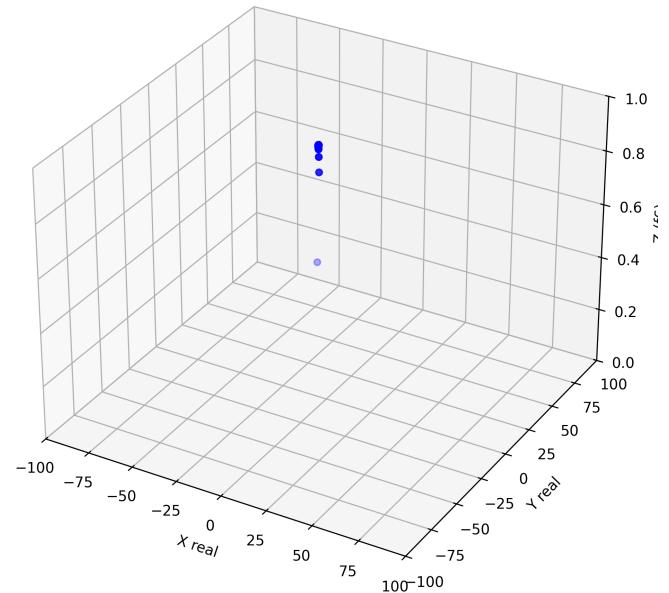


Fig. 15. Gráfico da geração 50 no espaço de busca.

pelo cruzamento for mais rápida que a divergência gerada pela mutação o algoritmo converge mais rapidamente. A respeito disso, é possível apontar que:

- O cruzamento por ponto de corte não concentra tão rapidamente a população em torno dos indivíduos mais aptos quanto o cruzamento aritmético, implementado na questão dois.
- A respeito da mutação, o bit flip da primeira questão possui muito mais chance de dispersar a população no

espaço de busca com valores maiores que a mutação com distribuição normal implementada. Foi possível observar que as gerações da questão 1 sempre apresentaram alguns indivíduos mais dispersos no espaço de busca ao longo das gerações.

- A taxa de mutação na questão 1 era avaliada por bit, enquanto na segunda era por indivíduo, assim na primeira um indivíduo tinha muito mais probabilidade de sofrer uma mutação, pois na segunda questão a probabilidade

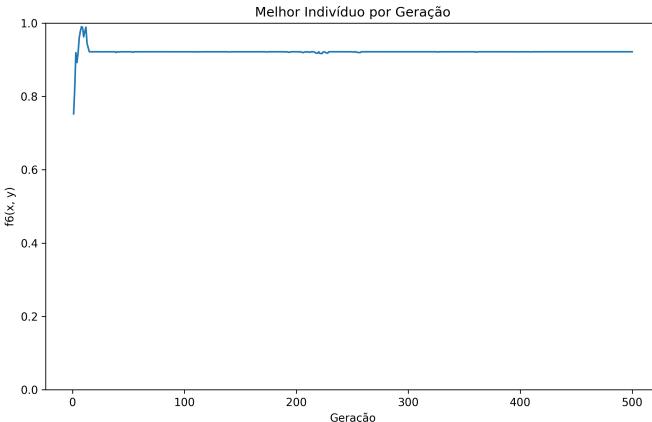


Fig. 16. Gráfico do valor de $f_6(x, y)$ do melhor indivíduo ao longo das 500 gerações.

de haver mutação em um indivíduo era a taxa de mutação: 0.01, enquanto na primeira a probabilidade pode ser modelada como na equação a seguir, que diz que a probabilidade de haver alguma mutação é 1 menos a probabilidade de não haver nenhuma em nenhum bit consecutivamente. Na equação N_{bits} é o número de bits que representam o indivíduo (50 no caso), e a $Taxa_{mutacao}$ é a taxa de mutação (0.01). É possível perceber o impacto da probabilidade do indivíduo sofrer uma mutação na convergência do algoritmo. No primeiro houve mais dispersão ao longo das gerações que o da segunda.

$$P_{mutacao} = 1 - (1 - Taxa_{mutacao})^{N_{bits}}$$

$$P_{mutacao} = 1 - (1 - 0.01)^{50} \approx 0.395$$

- A mutação seguindo a distribuição normal, de média 0 e variância igual a 1, teve menor capacidade de dispersão no espaço de busca que a mutação por bit flip, mas permitiu alta dispersão nas proximidades do ponto encontrado de máximo.

Observando o gráfico do melhor indivíduo ao longo das gerações das duas questões pôde-se perceber que a convergência no ponto de máximo na questão 1 correu em 1 e, na questão 2, ocorreu por volta de 0.92. Isso ocorreu pois o algoritmo genético da questão 2 ficou preso em um máximo local, talvez em outra tentativa o algoritmo desse sorte de convergir no máximo global a pesar de tê-lo encontrado em uma das gerações. Ambas implementações do algoritmo genético chegaram bem próximo ao ponto máximo da função objetivo em pelo menos uma de suas gerações, como pode ser observado nas figuras 9 e 16, que são os gráficos do melhor indivíduo ao longo das gerações da questão 1 e 2, respectivamente.

REFERÊNCIAS

- [1] <https://github.com/FerrariusF/Algoritmo-Genetico>