TRABALHO DE CONCLUSÃO DE CURSO

Desenvolvimento de um algoritmo genético com variação dinâmica de domínio de busca

Palton Lima Alves

Brasília, setembro de 2022

UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE FÍSICA

UNIVERSIDADE DE BRASÍLIA
INSTITUTO DE FÍSICA

TRABALHO DE CONCLUSÃO DE CURSO

Desenvolvimento de um algoritmo genético com variação dinâmica de domínio de busca

Palton Lima Alves

Trabalho de conclusão de curso submetido ao Instituto de Física

como requisito parcial para obtenção

do grau de Bacharel em Física

Banca Examinadora

Prof. Luiz Antonio Ribeiro Junior, IF/UnB
Orientador

Prof. Alexandre Cavalheiro Dias, IF/UnB
Membro Interno

Prof. Marcelo Lopes Pereira Júnior, FT/UnB
Membro Externo

FICHA CATALOGRÁFICA

LIMA ALVES, PALTON
Desenvolvimento de um algoritmo genético com variação dinâmica de domínio de busca [Distrito Federal] 2022.
xvi, 57 p., 210 x 297 mm (EFL/FT/UnB, Bacharel, Física, 2022).
Trabalho de conclusão de curso - Universidade de Brasília, INSTITUTO DE FÍSICA.

1. Algoritmo Genético	2. Métodos computacionais
3. Otimização	4. Python
I. EFL/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ALVES, P. L. (2022). *Desenvolvimento de um algoritmo genético com variação dinâmica de domínio de busca*. Trabalho de conclusão de curso, Instituto de Física, Universidade de Brasília, Brasília, DF, 57 p.

CESSÃO DE DIREITOS

AUTOR: Palton Lima Alves
TÍTULO: Desenvolvimento de um algoritmo genético com variação dinâmica de domínio de busca.
GRAU: Bacharel em Física ANO: 2022

Palton Lima Alves
Instituto de Física - IF
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

RESUMO

Em todas as áreas das ciências exatas e engenharias existem problemas cujas soluções analíticas são inviáveis de serem reproduzidas e as soluções computacionais mais diretas demandariam um grande tempo de execução. Os algoritmos genéticos surgiram na perspectiva de buscar soluções para problemas desse tipo, de forma a encontrar soluções otimizadas em intervalos de tempo praticáveis. Na perspectiva da seleção natural, os algoritmos genéticos buscam as melhores configurações da solução de um problema com base em uma lógica da predominância dos indivíduos mais aptos. Nesse sentido, o meio se traduz como o problema a ser resolvido e os indivíduos como as possíveis configurações da solução do problema. A execução do algoritmo genético baseado então em recriar gerações com variabilidade das características genéticas herdadas dos melhores indivíduos, que farão com que a população caminhe rumo a encontrar a solução mais otimizada para um determinado problema.

ABSTRACT

In all areas of exact sciences and engineering, there are problems whose analytical solutions are impossible to reproduce and the most direct computational solutions would require a large execution time. Genetic algorithms emerged from the perspective of seeking solutions to problems of this type, in order to find optimized solutions in practicable time intervals. From the perspective of natural selection, genetic algorithms seek the best configurations of the solution to a problem based on a logic of the predominance of the fittest individuals. In this sense, the environment translates as the problem to be solved and individuals as the possible configurations of the solution to the problem. The execution of the genetic algorithm based on recreating generations with variability of the genetic characteristics inherited from the best individuals, which will make the population walk towards finding the most optimized solution for a given problem.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Algoritmos genéticos (AG) são métodos computacionais pensados para otimizar soluções ou resolver problemas com muitos parâmetros e cujas soluções exatas, ou não existem, ou são computacionalmente inviáveis. O conceito do AG foi pensado inicialmente por John Holland por volta da década de 1960 (2). Tomando forte inspiração na teoria da evolução das espécies (3), Holland propôs então um modelo computacional para resolução de problemas baseado nos conceitos de seleção natural, variabilidade genética e mutação.

1.1 PRINCÍPIOS DO ALGORITMO GENÉTICO

Os algoritmos genéticos consistem em uma maneira otimizada de buscar de soluções de problemas que originalmente demandariam muito tempo computacional, com base na teoria de evolução das espécies. Segundo a teoria de Darwin, indivíduos mais bem adaptados ao meio deixarão descendentes com suas características genéticas. Com isso as novas gerações repetirão o mesmo processo, produzindo indivíduos cada vez mais bem adaptados ao meio.

Na busca de soluções para problemas reais, os algoritmos genéticos traduzem o contexto biológico para o computacional. A analogia entre os dois contextos diferentes pode ser vista na tabela a seguir:

Natureza	Algoritmos Genéticos
Cromossoma	Lista de bits
Gene	Característica do problema
Alelo	Bit
Genótipo	Estrutura
Fenótipo	Valor da estrutura quando submetida ao problema
Indivíduo	Solução
Geração	Ciclo

Tabela 1.1: Analogias entre o modelo genético biológico e o modelo de representação do algoritmo genético.(1)

A lógica de funcionamento do algoritmo genético é baseada em 7 etapas principais, são elas (4): O problema a ser otimizado, representação das soluções do problema, decodificação do cromossomo, avaliação, seleção, reprodução (*crossover*), mutação e elitização.

1.1.1 Problema a ser otimizado

O problema a ser otimizado se define como um problema cujas soluções exatas não são podem ser obtidas analiticamente ou o processo computacional para resolver o problema demande um

tempo de execução impraticável, em função do número de variáveis.

Alguns dos problemas que utilizam algoritmos genéticos são (1): Determinação de extremos de funções matemáticas, otimização combinatorial, o problema do caixeiro viajante, problema de otimização de rota de Veículos e otimização de layout de circuitos.

1.1.2 O problema do caixeiro viajante

O problema do caixeiro viajante é um problema de otimização muito conhecido nas áreas de ciência da computação e matemática. O problema consiste em basicamente determinar a melhor rota possível para se percorrer n pontos espalhados em posições diferentes, de maneira a minimizar a distância percorrida.

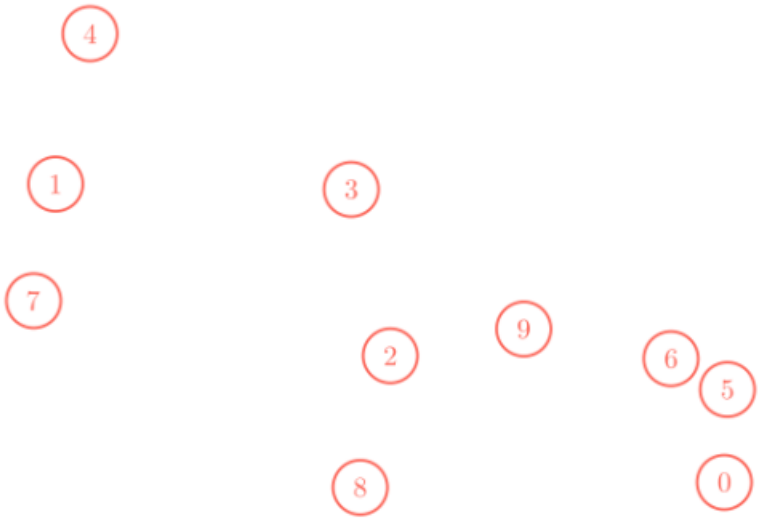

Figura 1.1: Problema do caixeiro viajante.

Dessa forma, em um caso onde se deseja passar por 9 pontos distintos, a maneira mais direta de se obter a solução seria testar todas as possíveis rotas e então selecionar a de menor distância.

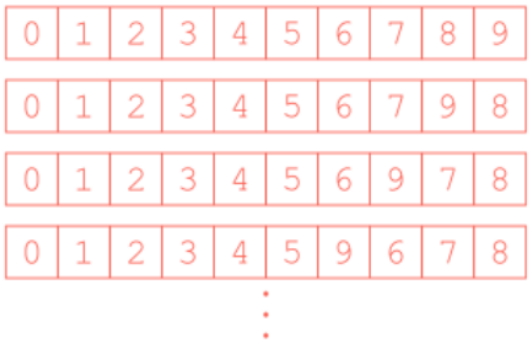

Figura 1.2: Ciclos de execução do algoritmo genético para o problema do caixeiro viajante.

A busca direta pela solução com essa abordagem demandaria testar todas as 362.880 diferentes

configurações até que se obtivesse a melhor solução possível para o problema,

Em um algoritmo genético a solução do problema do caixeiro viajante seria buscada testando inicialmente um conjunto aleatório de rotas, nesse contexto cada rota representa um indivíduo e o conjunto de todas as rotas seria a população. A partir dessas respostas aleatórias são selecionadas as que apresentam os menores trajetos e então um novo conjunto de possíveis soluções é gerado com base em combinações e pequenas alterações dessas melhores configurações.

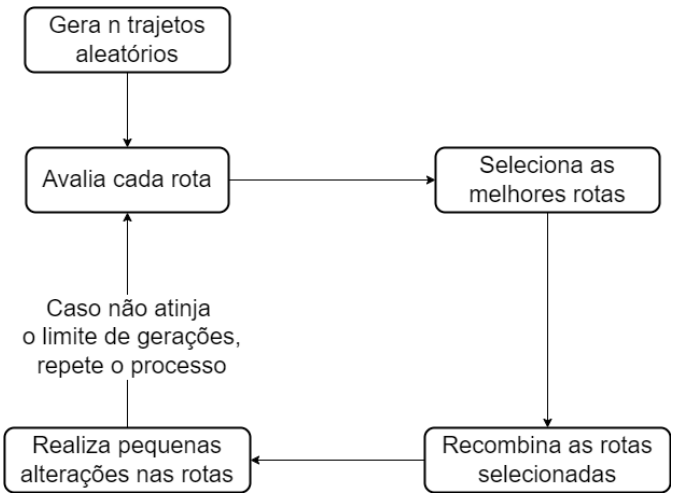

Figura 1.3: Diagrama de busca da solução do problema do caixeiro viajante.

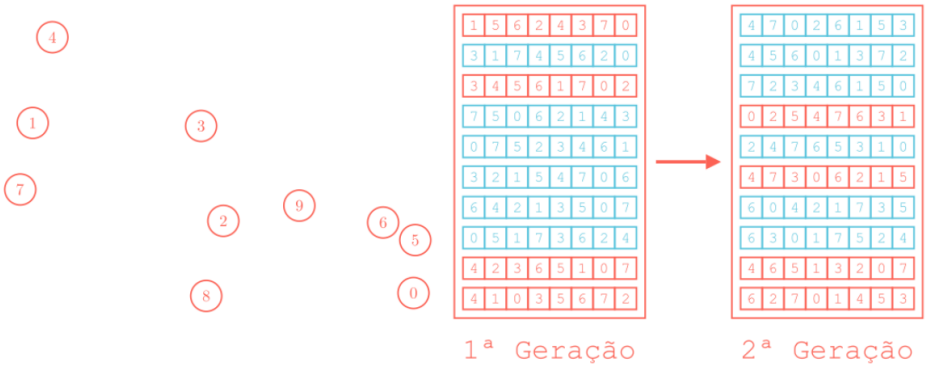

Figura 1.4: Busca de solução do problema do caixeiro viajante utilizando algoritmo genético.

1.1.3 Representação das soluções de problema

Considerando o caso de determinar o máximo de uma superfície. A solução do problema demandaria a existência de dois cromossomos diferentes, estes cromossomos representariam as características observáveis do indivíduos, que no problema em questão estão associados aos valores das coordenadas x e y.

Dessa forma, uma população de 8 indivíduos com dois cromossomos cada é então um conjunto 8 de pares de cromossomos binários, como é ilustrado na figura a seguir:

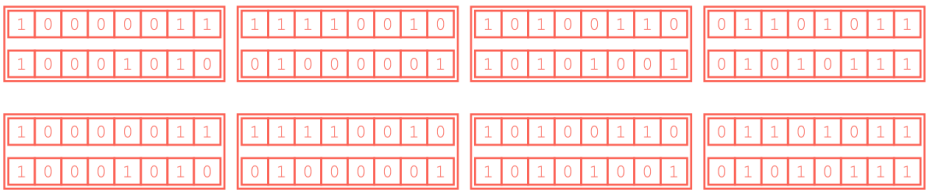

Figura 1.5: População de 8 indivíduos binários com 2 cromossomos cada.

1.1.4 Decodificação do Cromossomo

A decodificação do cromossomo é o processo pelo qual as configurações genéticas dos indivíduos são convertidas para valores reais.

No exemplo da procura de máximos de uma função gaussiana, os dois cromossomos de cada indivíduos são lidos como um número binário, convertidos para dois valores inteiros e em seguida escalados para uma determinada escala.

Considere o cromossomo ilustrado na figura a seguir:

Figura 1.6: Cromossomo de 16 bits.

Nesse caso, uma sequência de bits pode ser convertida para um número inteiro entre 0 e 2^n, onde n representa o número de bits, e em seguida convertido para uma escala desejada. A sequência de bits representaria então o número decimal 54420 e para que os cromossomos assumam valores entre -10 e 10, é necessário realizar a seguinte conversão:

$$X = \frac{54420}{2^{16}} \cdot 20 - 10 \approx 6,6 \tag{1.1}$$

Como valores convertidos são obtidos a partir de números inteiros, os valores dos cromossomos irão assumir valores discretos múltiplos de uma menor unidade, que é determinada pela seguinte relação:

$$\delta X = \frac{1}{2^n} \cdot f \tag{1.2}$$

Onde n é o número de bits em cada cromossomo e f é o fator de escala da decodificação.

A figura a seguir ilustra uma distribuição de pontos que representa a discretização do espaço de busca com base no número de bits escolhidos para a codificação de cada coordenada:

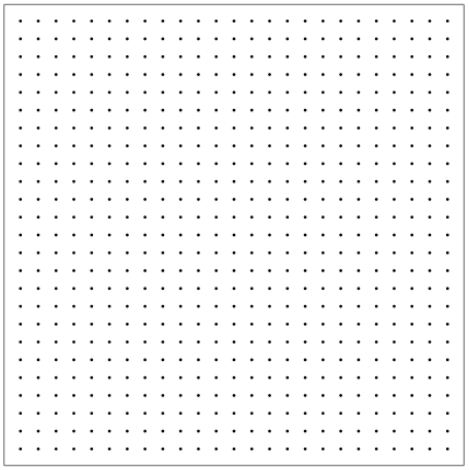

Figura 1.7: Discretização de um espaço de busca para codificação com 5 bits.

Para o caso da discretização do espaço da figura acima, são utilizados 5 bits para um intervalo entre -1 e 1, temos que a menor unidade de precisão é obtida da seguinte forma:

$$\delta X = \frac{1}{2^5} \cdot 2 \approx 0.06 \tag{1.3}$$

É importante ressaltar que o aspecto da precisão não se torna necessariamente uma desvantagem, já que ela restringe o campo de busca das soluções, o que consequentemente torna o algoritmo mais eficiente.

1.1.5 Avaliação

A avaliação dentro do algoritmo genético desempenha o papel do meio na seleção natural. Ela é um dos processos mais importantes em um algoritmo genético, pois é a responsável por avaliar quão bem as características genéticas de cada indivíduo se aproximam da solução do problema analisado.

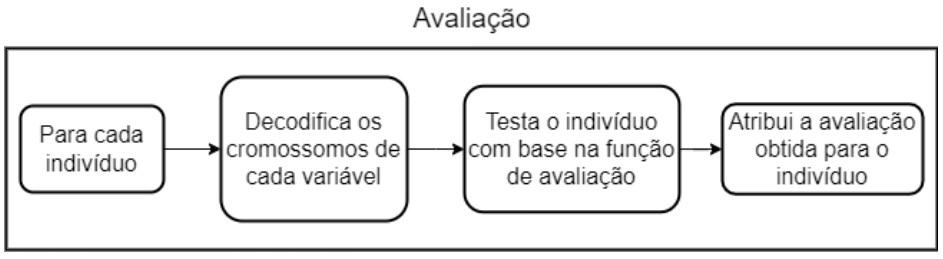

Figura 1.8: Diagrama de funcionamento do processo de avaliação.

No problema de determinar o máximo de uma superfície, a função de avaliação resgata os cromossomos dos indivíduos descodificados e pontua o indivíduo com base no quão próximo

do máximo da função esse indivíduo está. Considerando que a superfície analisada seja sempre positiva, o valor da avaliação poderia ser diretamente o valor da superfície no ponto $[x, y]$, como ilustra a figura a seguir:

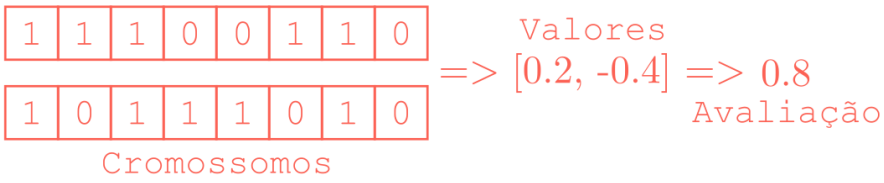

Figura 1.9: Representação da avaliação de dois cromossomos binários.

1.1.6 Seleção

A seleção é responsável por selecionar prioritariamente os melhores indivíduos para que possam dar origem a uma nova geração.

Um modelo de seleção eficiente deverá priorizar os indivíduos que possuem as melhores avaliações, no entanto não deve desprezar indivíduos com avaliações ruins. Uma vez que indivíduos mal avaliados podem possuir características genéticas que, em cruzamentos futuros, podem ser decisivas para a formação de indivíduos melhores(2).

Posto isso, um método eficiente de seleção de indivíduos e vastamente utilizado é o método baseado em uma roleta de avaliações(1). Esse método consiste em uma roleta, onde cada indivíduo possui uma fatia da roleta proporcional à sua avaliação. Conforme é possível visualizar na tabela a seguir:

Indivíduo	**Avaliação**	**Pedaço da roleta (%)**	**Fatia da roleta (º)**
1110010	27	8	11
1100001	39	11	23
0110001	79	23	46
0010010	97	28	92
0001010	99	29	185
Total	341	100	360

Tabela 1.2: Roleta de seleção de indivíduos.

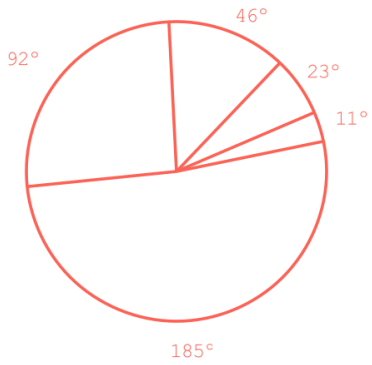

Figura 1.10: Roleta de sorteiro para a seleção de indivíduos.

Dessa forma, a cada seleção é sorteado uma posição na roleta e o indivíduo que representar o ângulo da fatia sorteada será selecionado para realizar a reprodução. Com isso, estatisticamente, os indivíduos mais bem avaliados possuirão maiores chances de serem selecionados sem que se despreze totalmente os indivíduos menos bem avaliados.

1.1.7 Reprodução (*Crossover*)

A reprodução (*crossover*) é o meio pelo qual os indivíduos selecionados geram novos indivíduos com configurações genéticas semelhantes. Nesse procedimento a recombinação dos cromossomos de dois indivíduos formam 2 novos indivíduos.

Considerando 2 indivíduos com 1 cromossomo cada, ilustrados na figura a sequir:

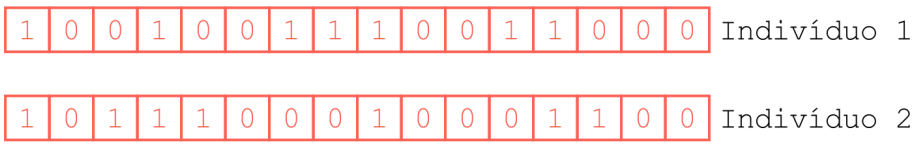

Figura 1.11: Dois indivíduos com 1 cromossomo cada.

Onde cada cromossomo representa um número entre -5 e 5.

O cruzamento separa um cromossomo em n pontos diferentes e então recombina partes dos códigos genéticos, gerando dois novos indivíduos. Como é possível visualizar na figura 1.12.

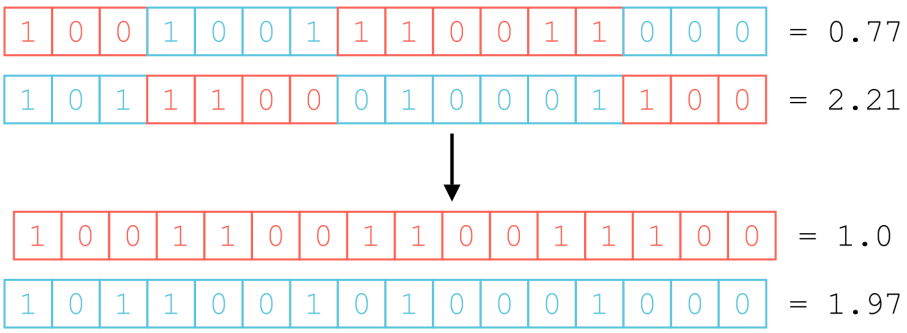

Figura 1.12: Cruzamento entre indivíduos com cromossomos binários em quatro partes.

Esses cruzamentos não irão tornar os descendentes necessariamente mais bem avaliados, mas farão com que as novas gerações sejam ao mesmo tempo próximas dos indivíduos pais e ligeiramente diferentes entre si.

1.1.8 Mutação

A mutação é o processo responsável por aumentar a variabilidade genética e assim permitir que indivíduos possuam novas características e possivelmente caminhem para uma melhor configuração.

Após a realização dos cruzamentos entre indivíduos selecionados, cada cromossomo de cada indivíduo possui uma probabilidade de f_{mut} de sofrer uma mutação. Caso o indivíduo seja sorteado para realizar mutação, o código varrerá todos os seus genes, tendo uma probabilidade i_{mut} de alterar o valor binário do gene para o seu oposto. Esse mecanismo permite controlar a frequência de mutação nos indivíduos e a intensidade de mutação em cada cromossomo.

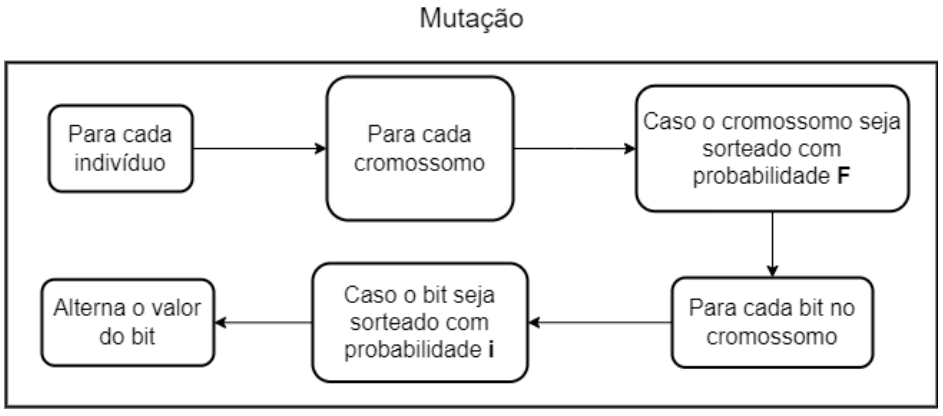

Figura 1.13: Diagrama de mutação dos indivíduos.

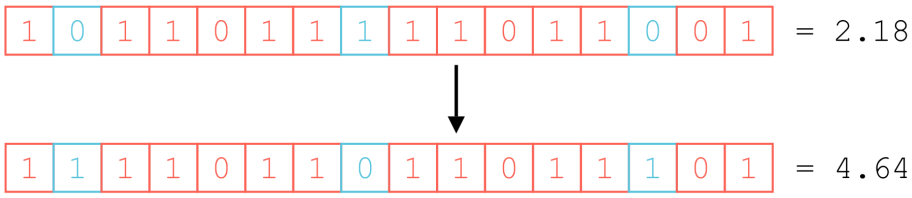

Figura 1.14: Ilustração dos métodos de mutação agindo sobre 1 indivíduos com 1 cromossomo.

1.1.9 A elitização

A estratégia de elitização consiste em selecionar uma determinada quantidade dos melhores indivíduos. Estes indivíduos permanecerão nas próximas gerações a fim de manter o material genético dos melhores indivíduos, não permitindo assim que a população perca uma posição de máximo local e sempre esteja apta a caminhar rumo a melhores configurações.

2 TESTES PRELIMINARES

A fim de testar eficiência do algoritmo genético, foram realizados testes com superfícies com 2 ou mais pontos de máximos. Estes testes possuem o objetivo de verificar a capacidade do algoritmo genético de encontrar um máximo global em meio a diversos máximos locais.

2.1 PRIMEIRO TESTE - SUPERFÍCIE COM INFINITOS MÁXIMOS LOCAIS

O teste inicial consistiu em determinar o máximo da superfície ilustrada na figura a seguir:

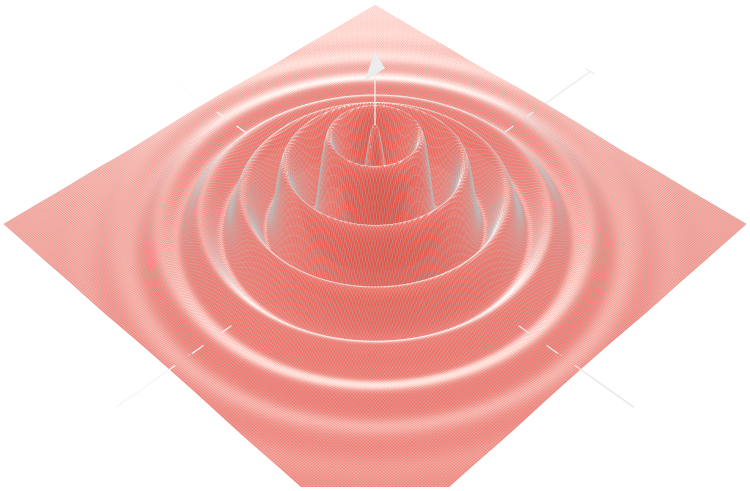

Figura 2.1: Superfície $cos^2(3\pi r)e^{-r^2}$.

Esta superfície apresenta um ponto de máximo global na origem e infinitos pontos de máximos locais, o que permite testar a capacidade do algoritmo genético de não estagnar em máximos locais.

Realizando um total de 100 execuções com o algoritmo genético configurado com 200 indivíduos em 50 gerações, foi obtida uma relação entre a média das avaliações de todos os indivíduos em função da geração. Como é possível visualizar na figura a seguir:

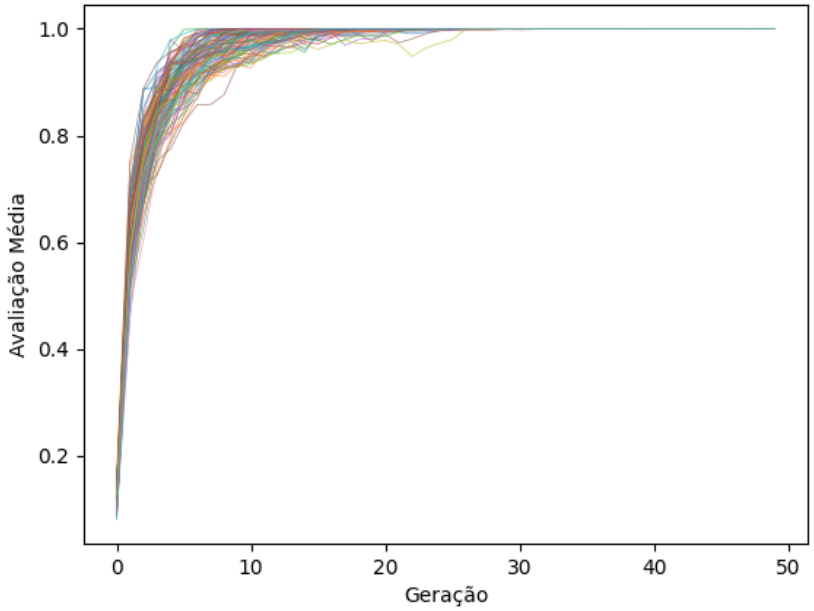

Figura 2.2: Gráfico da avaliação média de 100 execuções do algoritmo genético aplicado para a superfície $cos^2(3\pi r)e^{-r^2}$.

Nesta configuração, a disposição dos indivíduos de de todas as execuções realizadas podea ser visualizada na figura a seguir:

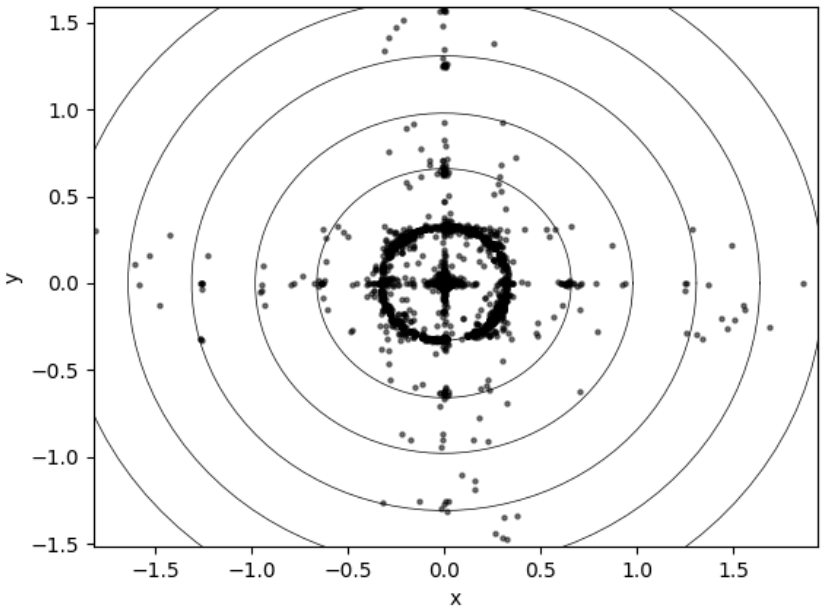

Figura 2.3: Distribuição de indivíduos do algoritmo genético aplicado à superfície $cos^2(3\pi r)e^{-r^2}$.

Nesta figura as curvas de nível ilustradas representam os pontos de máximo da função. loca-

lizados em $r = \frac{n}{3}$, para $n = 0, 1, 2, 3....$

É possível notar que ons indivíduos se acumularam na origem e se espalharam pelos outros máximos da superfície.

Nestas execuções os indivíduos apresentaram uma tendência a se acumularem sobre os eixos x e y de maneira independente, o que configura a distribuição dos indivíduos em formato de cruz centrada na origem. Este comportamento ocorre devido ao algoritmo pontuar bem indivíduos que possuam coordenadas próximas da origem individualmente. Uma alternativa para a otimização deste problema seria a adoção de um sistema de coordenadas mais apropriadas.

Originalmente a função de conversão decodifica os cromossomos de um indivíduo como valores de x e y. No entanto, como o problema possui simetria polar, adotar o sistema de coordenadas polares na decodificação dos cromossomos otimizaria o problema para apenas uma dimensão radial.

Com isso em mente, é possível observar que, em todas as 100 execuções, o algoritmo genético foi capaz de determinar os máximos da superfície na 30ª geração. Foi possível inferir também que uma otimização na função de avaliação que se beneficie as simetrias do problema poderia tanto diminuir o tempo de execução quanto produzir resultados mais consistentes.

2.2 SEGUNDO TESTE - SUPERFÍCIE COM MÁXIMOS LOCAIS PRÓXIMOS

Este teste tem como objetivo determinar os máximos da superfície ilustrada na figura a seguir:

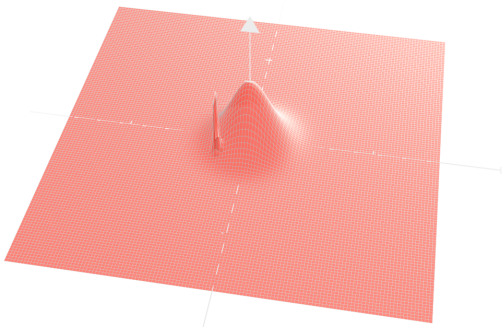

Figura 2.4: Superfície $0.8 * e^{-\frac{x^2+y^2}{0.3^2}} + 1 * e^{-\frac{(x-0.3)^2+(y-0.3)^2}{0.03^2}}$.

Realizando um total de 100 testes com o algoritmo genético configurado com 50 indivíduos em 100 gerações, foi obtida uma relação entre a média das avaliações de todos os indivíduos em função da geração. Como ilustra a figura a seguir:

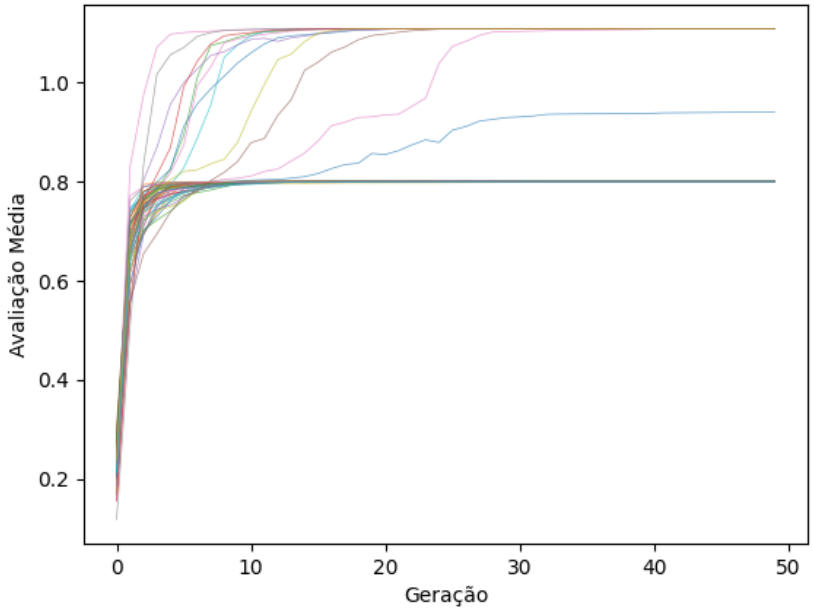

Figura 2.5: Evolução da avaliação dos indivíduos do algoritmo genético aplicado à superfície $0.8 * e^{-\frac{x^2+y^2}{0.3^2}} + 1 * e^{-\frac{(x-0.3)^2+(y-0.3)^2}{0.03^2}}$.

Neste caso é possível observar que nem todas as populações convergem para o ponto de máximo global. No entanto isso não significa que o algoritmo não foi capaz de encontrar o máximo global em todas as execuções. Como é possível observar na figura 2.7, uma fração substancial dos indivíduos tendem a se acumular na gaussiana localizada na origem, todavia os melhores indivíduos das gerações se concentram no máximo global localizado nas coordenadas $[0.3, 0.3]$.

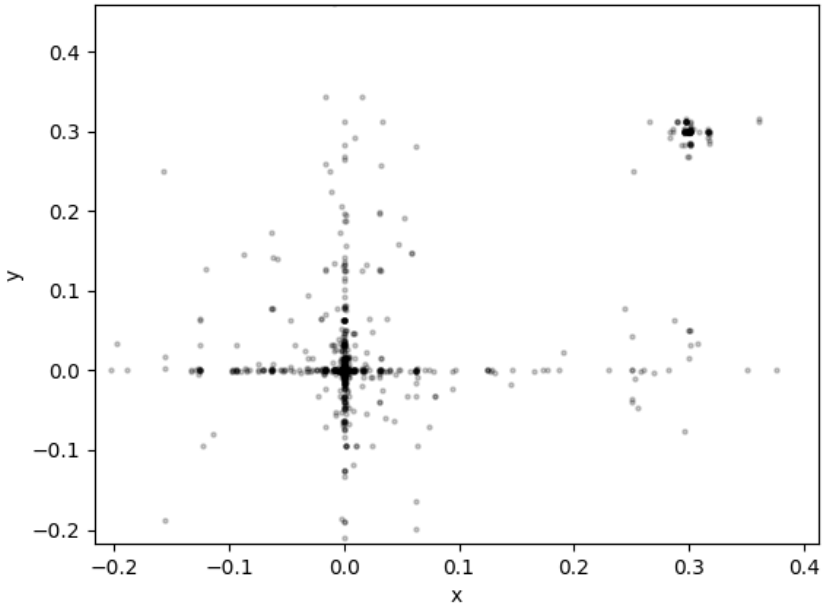

Figura 2.6: Distribuição de indivíduos do algoritmo genético aplicado à superfície $0.8 * e^{-\frac{x^2+y^2}{0.3^2}} + 1 * e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.09^2}}$.

2.3 A MELHOR CONFIGURAÇÃO

A melhor configuração de um algoritmo genético não está associada necessariamente ao tempo de execução, ou ao quão bem treinado um algoritmo genético está. A eficiência de um AG está em maior parte associada às otimizações realizadas para resolver um determinado problema. Essas otimizações muitas vezes estarão ligadas ao modelo de seleção dos melhores indivíduos, à estratégia de cruzamento, à mutação e à função de avaliação.

Dado isso, os próximos passos para a implementação do AG serão guiados com base em determinar as melhores configurações de população, cruzamento, mutação e também na busca de métodos que auxiliem a otimizar a busca em extensos domínios.

2.4 POR QUÊ UTILIZAR PYTHON?

A boa performance, rápida curva de aprendizagem, documentação e a facilidade de se tratar dados utilizando funções de bibliotecas disponíveis são diferenciais na escolha da linguagem Python. No entanto, o fato de a linguagem Python ser interpretada e possuir um sólido o ambiente de depuração por meio do ambiente de desenvolvimento *Visual Studio Code* é uma vantagem considerável no que se refere à experiência de desenvolvimento e produtividade.

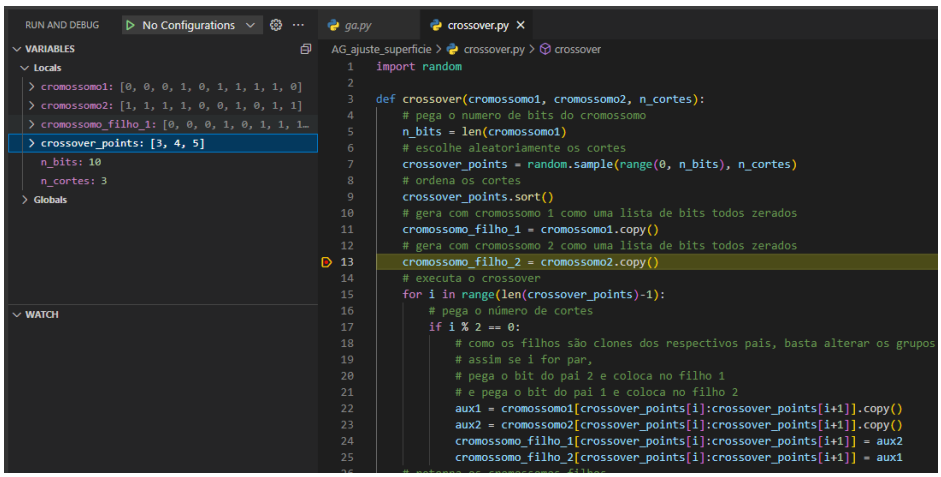

Figura 2.7: Ambiente de depuração de Python com o *Visual Studio Code*.

Com o ambiente de depuração é possível realizar pausa em processos em tempo real, com isso é possível monitorar variáveis, alterar valor de variáveis durante a execução do algoritmo, realizar parada do código com a ocorrência de erros.

Todas essas possibilidades na utilização do Python auxiliam no desenvolvimento do código, no rastreamento e correção de bugs e também principalmente na experiência execução de testes com o algoritmo.

3 AJUSTE DE FUNÇÕES COM ALGORITMO GENÉTICO

Dentre as principais utilizações de algoritmos genéticos está a aplicação para determinação de parâmetros de ajuste de curvas.

Dado um conjunto de dados, experimentais ou teóricos, determinar a melhor curva que se ajuste poderá envolver uma extensa busca pelos valores de n parâmetros. E com isso a complexidade desta busca é tão grande quanto maior for o número de parâmetros e quanto menos se sabe sobre quanto vale cada parâmetro.

Em um ajuste de uma curva onde são informados unicamente pontos no plano, como os ilustrados na figura a seguir:

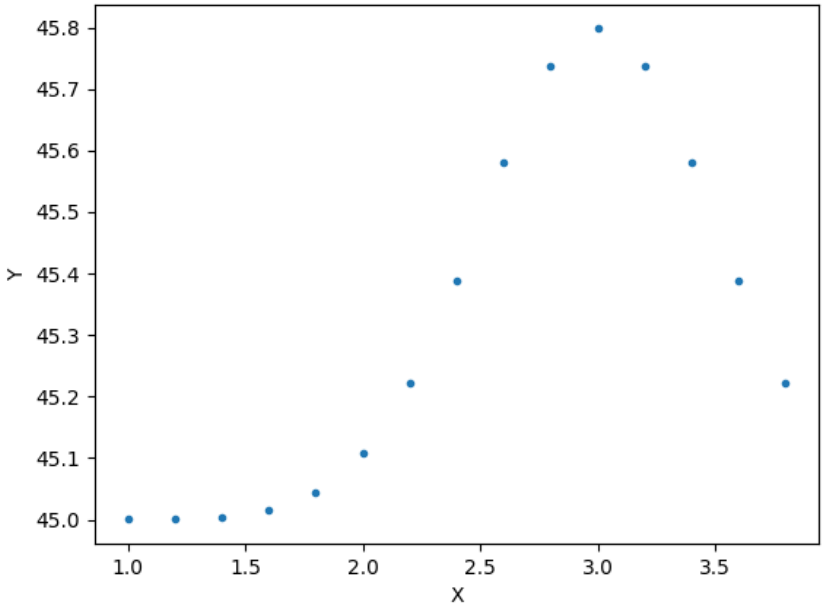

Figura 3.1: Amostra de dados para ajuste

Os dados sugerem claramente um comportamento gaussiano, onde a função de ajuste mais generalizada seria do tipo dada pela equação a seguir:

$$f(x) = a + \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.1}$$

Dessa forma, determinar a melhor curva de ajuste para esta distribuição de dados seria o equivalente a determinar os parâmetros a, σ e μ da função.

3.1 FUNÇÃO DE AVALIAÇÃO

A função de avaliação no algoritmo genético é o ponto de conexão do algoritmo com o problema a ser otimizado. Sendo assim para que se possa obter soluções para o problema é necessário que se estabeleça critérios de pontuação para os indivíduos do algoritmo genético que condizentes com a solução que se deseja obter.

Na busca de parâmetros de ajuste de curvas, cada indivíduo da população, munido dos parâmetros da curva, é avaliado a com base em quão próximo a curva gerada pelas suas configurações está dos pontos da distribuição informada.

Dessa forma, a avaliação de um indivíduo pode ser determinada com base na média da distância mínima quadrática entre cada ponto da distribuição de pontos $[x_0, y_0]$ com cada ponto da curva gerada pelo indivíduo da população $[x_0, f(x_0)]$. Onde $f(x_0)$ é o ponto da curva gerada pela função de ajuste com os parâmetros dados pela a partir das configurações do indivíduo em $x = x_0$. Com isso a função de avaliação é calculada com base na seguinte equação:

$$A = \frac{\sum_{i=0}^{n}(y_0 - f(x_0))^2}{n}$$

À vista disso, a bem adaptação do indivíduo ao problema está associada a quanto menor é a distância quadrática dos seus valores em relação aos pontos do da curva previamente informados.

3.1.1 Implementação da função de avaliação para ajuste de curva

A função de avaliação para ajuste de curvas com algoritmo genético foi implementada com base no seguinte pseudo código:

```
1  Define funcao de ajuste
2  def funcao_de_ajuste(parametro_a, Parametro_b, Parametro_c, x):
3      retorna valor da funcao no ponto x
4
5  Para cada individuo da populacao:
6      # Resgata os as coordenadas x e y dos pontos de ajuste
7      x = [...]
8      y = [...]
9      #Inicia a soma quadratica como0
10     soma_quadratica = 0
11     Para cada ponto x_0 em x:
12         parametro_a = individuo[0]
13         parametro_b = individuo[1]
14         parametro_c = individuo[2]
15         ponto_ajustado = funcao_de_ajuste(param_a, Param_b, Param_c, x_0)
16         soma_quadratica += ((ponto_ajustado - y[i])**2)
17     quadratic_mean = mean((sqrt(quadratic_sum)))
```

```
18      avaliacao = quadratic_mean
```

3.2 SOLUÇÕES DO AJUSTE DE CURVA

A solução do ajuste de curva é determinada pelos valores dos n parâmetros de ajuste. As soluções, no entanto, podem ser visualizadas graficamente, como é ilustrado na figura a seguir:

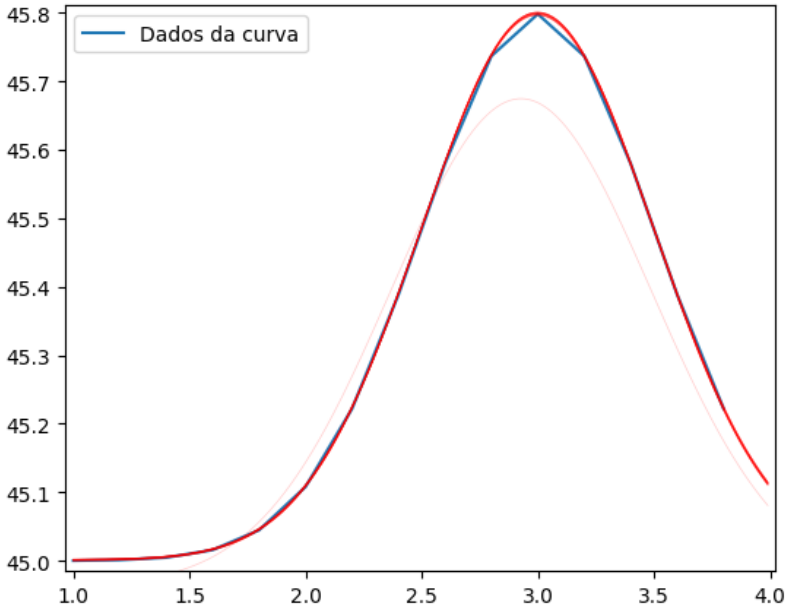

Figura 3.2: Curvas com os melhores indivíduos para ajuste da função gaussiana.

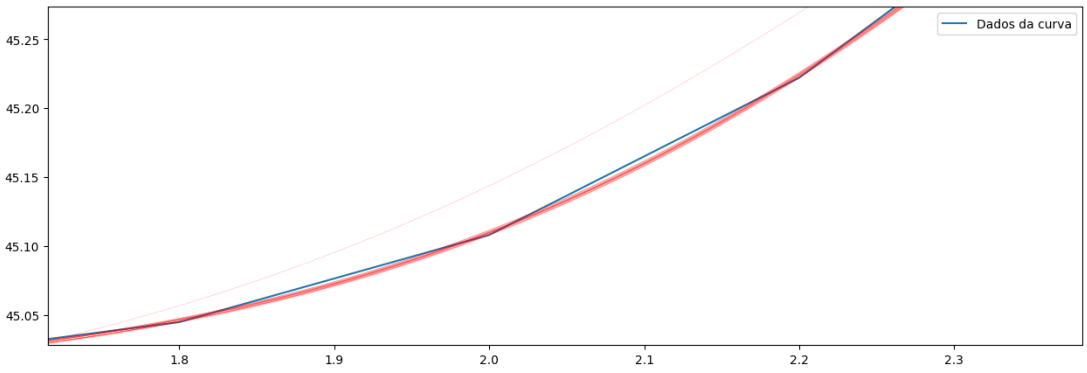

Figura 3.3: Zoom da curva das soluções para o ajuste da função gaussiana.

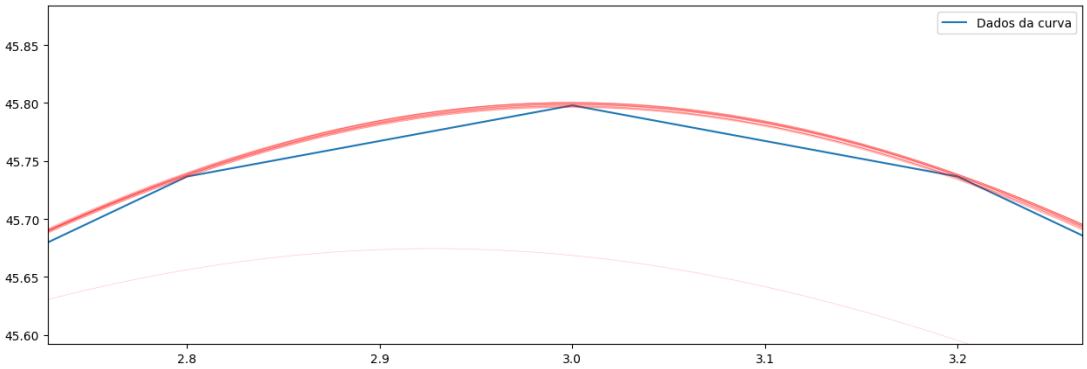

Figura 3.4: Zoom do ponto máximo das soluções para o ajuste da função gaussiana.

4 CONFIGURAÇÃO DO AG

4.1 CRUZAMENTO DE INDIVÍDUOS

A escolha do modelo de codificação binária foi motivada principalmente pela possibilidade de realização do cruzamentos com a maior variabilidade genética possível, juntamente com a possibilidade de realizar mutações arbitrárias nos cromossomos dos indivíduos.

A variabilidade genética no cruzamento do algoritmo genético está mais precisamente associada a quão arbitrárias as configurações de indivíduos filhos estarão entre as configurações dos indivíduos pais. Como está ilustrado na figura a seguir:

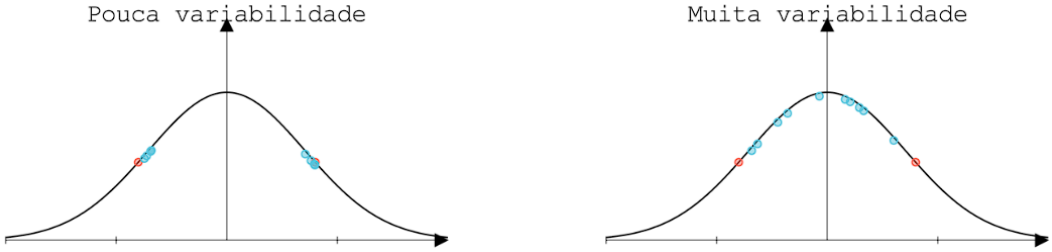

Figura 4.1: Relação entre variabilidade genética no cruzamento com a disposição dos indivíduos sobre o problema. Nesta representação os indivíduos pais são indicados pelos pontos vermelhos e os indivíduos filhos pelos pontos azuis.

No processo de cruzamento esta variabilidade genética é controlada por meio do número de divisões realizadas em um par de cromossomos no momento da recombinação de genes. Com é possível visualizar na ilustração da figura a seguir:

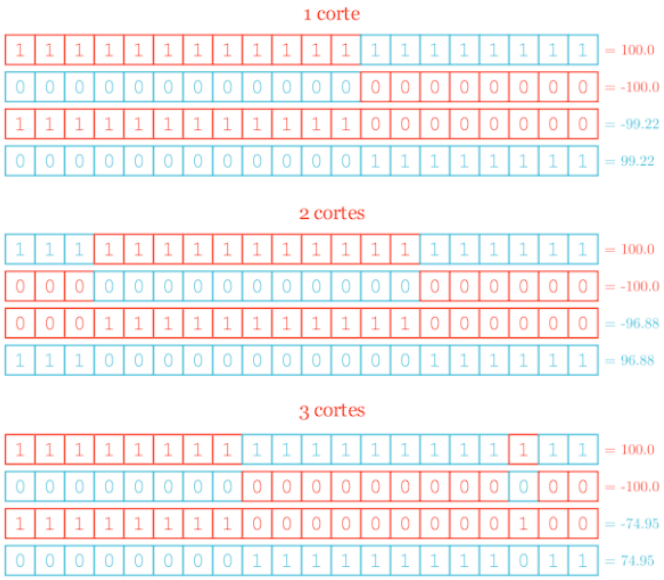

Figura 4.2: Cruzamento entre cromossomos com diferentes números de corte.

No intuito de buscar a melhor configuração para o número de cortes dos cromossomos para o cruzamento, realizou-se testes cruzando 50000 indivíduos com valores entre -100 e 100 para números de cortes que vão de 1 até 19, obtendo uma relação entre do desvio padrão dos indivíduos filhos e o número de cortes no cruzamento, como é possível visualizar na figura a seguir:

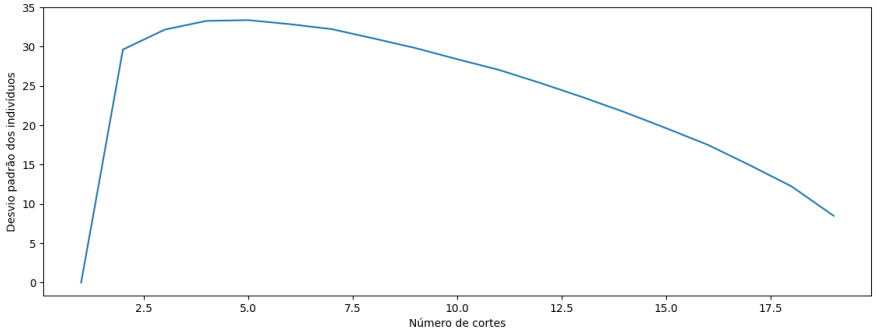

Figura 4.3: Cruzamento entre cromossomos com diferentes números de corte.

Com isso, para esta configuração, é possível notar que a variabilidade genética assume seu melhor valor quando o número de cortes está próximo de 5 cortes, onde o desvio padrão dos cruzamentos apresenta maior valor.

Realizando testes com o mesmo número de indivíduos, mas para codificações que variam de 5 bits a 30 bits, foi possível obter a relação entre o número de bits utilizados na codificação com o melhor número de cortes. Ilustrado na figura a seguir:

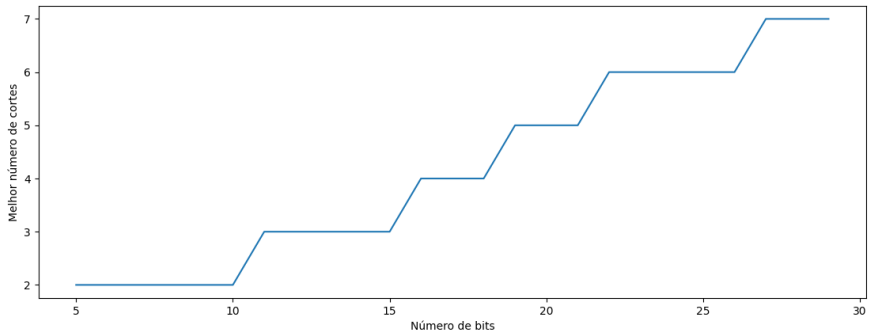

Figura 4.4: Relação entre o número de bits para codificação e melhor número de cortes para cruzamento.

A partir desta relação é possível inferir que a melhor variabilidade genética no cruzamento poderá ser obtida dividindo os cromossomos em média em $n/4$ partes, onde n representa o número de bits na codificação dos cromossomos.

5 OTIMIZAÇÃO DE TEMPO DE EXECUÇÃO

Por mais que métodos eurísticos sejam processos eficientes para buscas otimizadas em largos domínios e em n dimensões, a busca por uma solução otimizada em um tempo praticável[1] pressupõe sempre um domínio de busca delimitado.

A delimitação do domínio de cada variável não é apenas uma resposta para otimizar o tempo de busca do algoritmo rumo à solução otimizada. Com base no modelo de codificação genética escolhido para o algoritmo em questão, o tamanho do domínio de busca, juntamente com o número de bits escolhidos para representar cada dimensão, determinam com qual precisão os a solução será encontrada (equação 1.3).

Tomando como exemplo o a busca do máximo de uma superfície dada pela equação:

$$f(x,y) = e^{-\frac{(x-50000)^2+(y-50000)^2}{0.03^2}} \tag{5.1}$$

Em um domínio $x \in [-100000, 100000]$ e $y \in [-100000, 100000]$. Mesmo que em apenas duas dimensões, seria como procurar um curto pulso pulso em um vasto oceano.

Tais dificuldades são comumente encontradas em algoritmos genéticos de ajuste de curvas onde os parâmetros podem ser tão arbitrários quanto possível.

Uma solução seria elevar o número de indivíduos para que possam varrer o domínio mais rapidamente de maneira a viabilizar o tempo de execução, no entanto, o número de indivíduos de uma população tem impacto, tanto na utilização de memória da máquina, quanto indiretamente no tempo de execução do algoritmo.

Logo, incrementar o número de indivíduos de uma população com o intuito de varrer melhor o domínio é uma estratégia que é paga com tempo de execução.

A tabela a seguir mostra a relação entre a menor divisão dos valores de uma coordenada de um indivíduo com a largura do domínio, utilizando 16 bits para a codificação:

[1] Com "tempo praticável"diz-se um intervalo de tempo que seja viável à execução do programa. Visto que um programa que garantiria a solução otimizada, mas que demandaria meses ou anos de execução não seria prático de ser utilizado.

Alcance do domínio	Precisão em cada indivíduo
20000	3,05E-01
10000	1,53E-01
5000	7,63E-02
2500	3,81E-02
1250	1,91E-02
625	9,54E-03
312,5	4,77E-03
156,2	2,38E-03
78,1	1,19E-03
39	5,96E-04
19,6	2,98E-04
9,8	1,49E-04
4,8	7,45E-05
2,4	3,73E-05

Tabela 5.1: Relação entre o alcance do domínio de busca e a menor divisão de busca para uma codificação cromossomial de 16 bits.

Neste ponto, mesmo que os dados sugiram indiretamente que a solução para se obter maior precisão para o valor de cada parâmetro de busca seja elevar o número de bits de codificação, aumentar o número de bits significa também elevar o tamanho do campo de busca para o algoritmo genético. Uma vez que aumenta o número de combinações possíveis para a busca do algoritmo.

Dessa forma, para o caso em questão, elevar o número de bits com a intenção de se manter o domínio largo e conseguir a precisão que se conseguiria com um domínio 1000 vezes menor implicaria em aumentar espaço de busca em cerca de

$$2^{10} = 1024$$

vezes.

Em 5 dimensões seria o equivalente então a elevar o espaço de busca a um fator de

$$(2^{10})^5 = 2^{50}$$

.

Dessa forma, a busca por otimização da precisão dos parâmetros não deveria se basear somente em elevar o número de bits da codificação dos parâmetros. Visto que o crescimento exponencial do espaço de busca comprometeria o tempo de execução com e/ou a memória da CPU, se o problema fosse respondido com a elevação da população.

5.1 CONVERGÊNCIA DO DOMÍNIO

Exposto o problema, uma estratégia abordada para a otimização do domínio de busca do algoritmo genético é a convergência do domínio.

O método de convergência do domínio consiste em reduzir o domínio de busca de cada variável do algoritmo genético a cada conjunto de n gerações. Para isso o algoritmo genético reduz o domínio de busca de cada parâmetro com base nos valores de uma fração dos melhores indivíduos.

Tomando como exemplo a busca de máximos da superfície da figura a seguir:

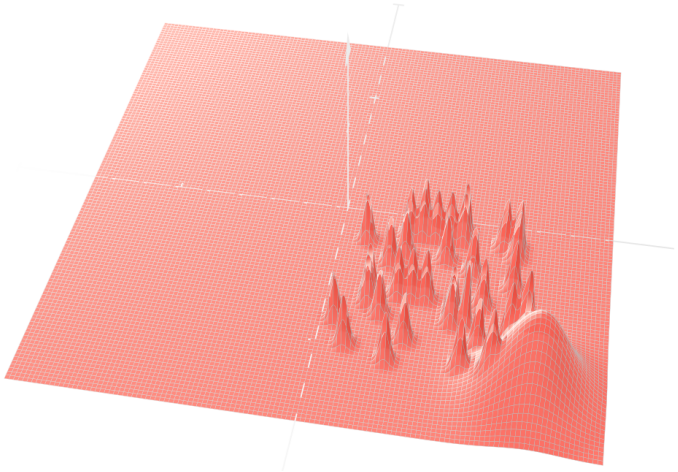

Figura 5.1: Superfície genérica com máximos locais e global localizados em um único quadrante do domínio.

Neste caso os máximos locais e global da superfície se encontram em apenas um dos 4 quadrantes. Para o caso em questão o algoritmo perceberá esta distribuição logo nas primeiras gerações, no entanto se o algoritmo genético fosse capaz de limitar o domínio de busca para apenas o quarto quadrante, as busca pelas soluções não só convergiria mais rápido, quanto se tornaria mais precisa.

Em um contexto de aplicação diferente, como no caso de busca de parâmetros para ajustes de curvas com algoritmo genético, esta estratégia se mostraria eficaz, uma vez que viabilizaria a exclusão de domínios dos parâmetros onde a função a ser ajustada possuiria comportamentos assintóticos indesejados.

Em um ajuste de curva como a da equação a seguir:

$$f(x) = ae^{-b(x)^2} \tag{5.2}$$

A função de avaliação do algoritmo genético avaliará o quão perto a função gerada pela configuração do indivíduo se aproxima, ponto a ponto, da função procurada.

Na hipótese de que sejam informados domínios arbitrários para os parâmetros a e b, a função divergirá exponencialmente da solução para valores de b que façam a função divergir para $+\infty$ ou convergir para zero.

Nesta circunstância o método de convergência do domínio, ao excluir os domínios onde a função de avaliação diverge, não só proporcionaria maior precisão dos valores ao diminuir o domínio, como faria com que o algoritmo genético fizesse um melhor uso do espaço de busca.

Em casos como este, o domínio informado para o parâmetro b não poderia ser arbitrário, visto que é explicito que a função de avaliação divergiria para valores negativos ou valores que fariam com que a função exponencial divergisse para $+\infty$. No entanto, pensando em problemas de otimização que poderiam vir a possuir 10 parâmetros de ajuste e que não são necessariamente parâmetros de ajuste de uma função, é possível que não seja claro em quais domínios de quais parâmetros a função de avaliação poderia divergir.

É importante ressaltar que em contextos onde o algoritmo genético careceria de longas execuções para encontrar soluções razoáveis, o algoritmo inevitavelmente faria com que o domínio convergisse para intervalos tão curtos que fariam com que não só a precisão fosse desnecessariamente alta, mas também com que toda a população convergisse para um ponto de máximo local e nunca deixasse este ponto.

Uma solução para este obstáculo seria então determinar uma largura mínima de domínio para cada parâmetro. Dessa forma o algoritmo não só deixaria de estagnar com o domínio ao longo de algumas gerações, mas permitiria que o domínio de cada parâmetro se deslocasse para valores de domínio que não foram informados, mas que possivelmente viabilizariam avaliações ainda melhores.

5.1.1 Implementação da convergência do domínio

O método de convergência do domínio foi implementado com base no seguinte pseudocódigo:

```
1  Define os dominios iniciais de cada parametro
2  Define a largura minima de cada dominio
3  define a largura maxima de cada dominio
4  Define o numero de geracoes ate que o dominio seja convergido
5  Inicializa uma populacao de indivduos aleatorios
6  Para cada geracao:
7      Avalia os individuos
8      Ordena os individuos com base em suas avaliacoes
9      Seleciona os individuos Elite
10     Realiza o crossover
11     Realiza a mutacao
12     A cada convergencia do dominio:
13         Seleciona o menor e maior valor de cada coordenada dentre X%
14         dos melhores indivduos da populacao
```

```
15          Se (maior valor - menor valor < largura minima):
16              valor medio = (maior valor + menor valor)/2
17              menor valor = valor medio + largura minima / 2
18              maior valor = valor medio - largura minima / 2
19              Ajusta valores dominio para que nao se reduza a um dominio
20              desnecessariamente curto
21          Se (maior valor - menor valor > largura maxima):
22              valor m d i o = (maior valor + menor valor)/2
23              menor valor = valor medio + largura maxima / 2
24              maior valor = valor medio - largura maxima / 2
25              Ajusta valores dominio para que nao se extenda a um dominio
26              grande demais
27          Redefine o dominio de busca de cada variavel para o menor e
28          maior valor encontrado do respectivo parametro
29          Reconfigura os cromossomos para que possam equivaler aos mesmos
30          valores anteriores, mas para o novo dom n i o
```

5.1.2 Testes de busca

5.1.2.1 Ajuste de uma curva gaussiana

A fim de testar a eficiência do algoritmo genético utilizando o método de convergência do domínio em relação ao algoritmo genético onde o domínio permanece estático. Gerou-se uma distribuição de pontos com base na função de ajuste dada pela equação 3.1 com os seguintes parâmetros:

$$a = 45$$

$$\sigma = 0.5$$

$$\mu = 3$$

Os testes consistiram em determinar os 3 parâmetros de ajuste em um total de 350 gerações com 1000 indivíduos, convergindo o domínio da população a cada 60 gerações.

A decisão de se realizar testes com base em parâmetros conhecidos foi motivada pela facilidade de se acompanhar a evolução dos indivíduos para as soluções conhecidas. Dessa forma é possível testar mais precisamente os parâmetros que levam o algoritmo a obter convergir melhor para soluções.

Realizando primeiramente os testes onde se manteve o domínio de cada parâmetro estático:

$$a = [0, 1000]$$

$$\sigma = [0, 1000]$$

$$\mu = [0, 1000]$$

Ao plotar a evolução dos 20 melhores indivíduos da população ao longo das gerações obteve-se os gráficos ilustrados na figura a seguir:

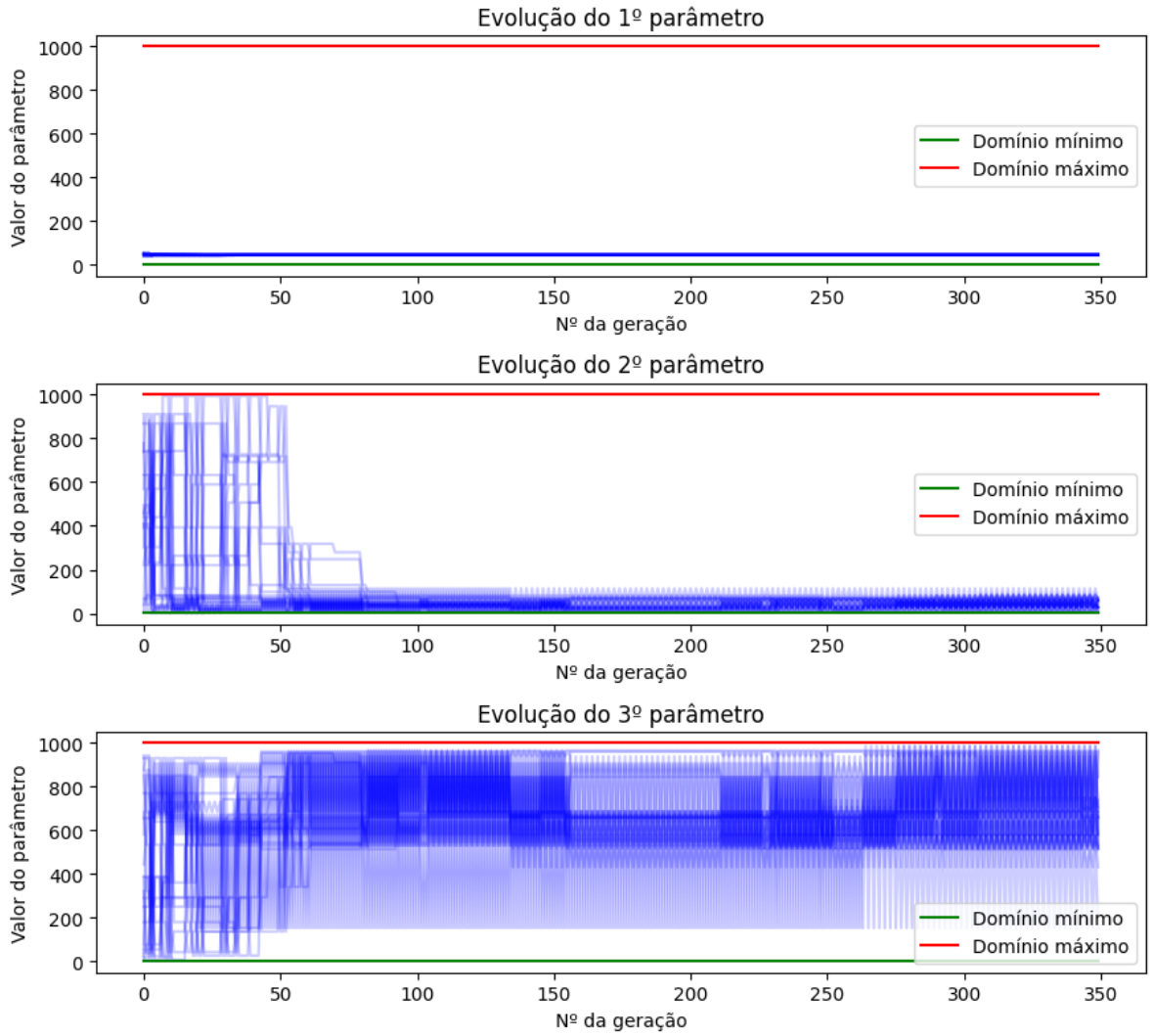

Figura 5.2: Gráfico da evolução de cada parâmetro da população ao longo de 350 gerações mantendo o domínio de busca estático.

Com base no gráfico da figura acima, é possível notar como o primeiro parâmetro converge rapidamente para o seu melhor valor. Isso se deve ao fato de que uma gaussiana cuja diferença entre o ponto mais alto e o ponto mais baixo seja inferior a uma unidade, vista em uma escala de 0 a 1000, se aproxima a uma reta horizontal independentemente dos demais parâmetros. Dessa forma o algoritmo basicamente pontua os indivíduos com base unicamente no primeiro parâmetro, até que ele seja ajustado, quando os outros parâmetros passarem a influenciar a função de avaliação.

Os demais parâmetros, no entanto, apresentam grande flutuação em relação aos pontos da solução.

Após o fim da execução obteve-se a seguinte solução para os parâmetros de ajuste da curva:

$$a = 47.06$$

$$\sigma = 4.92$$

$$\mu = 509.8$$

Executando o algoritmo genético com as mesmas configurações e utilizando o método de convergência do domínio foi possível gerar os gráficos da evolução dos parâmetros como mostra a figura a seguir:

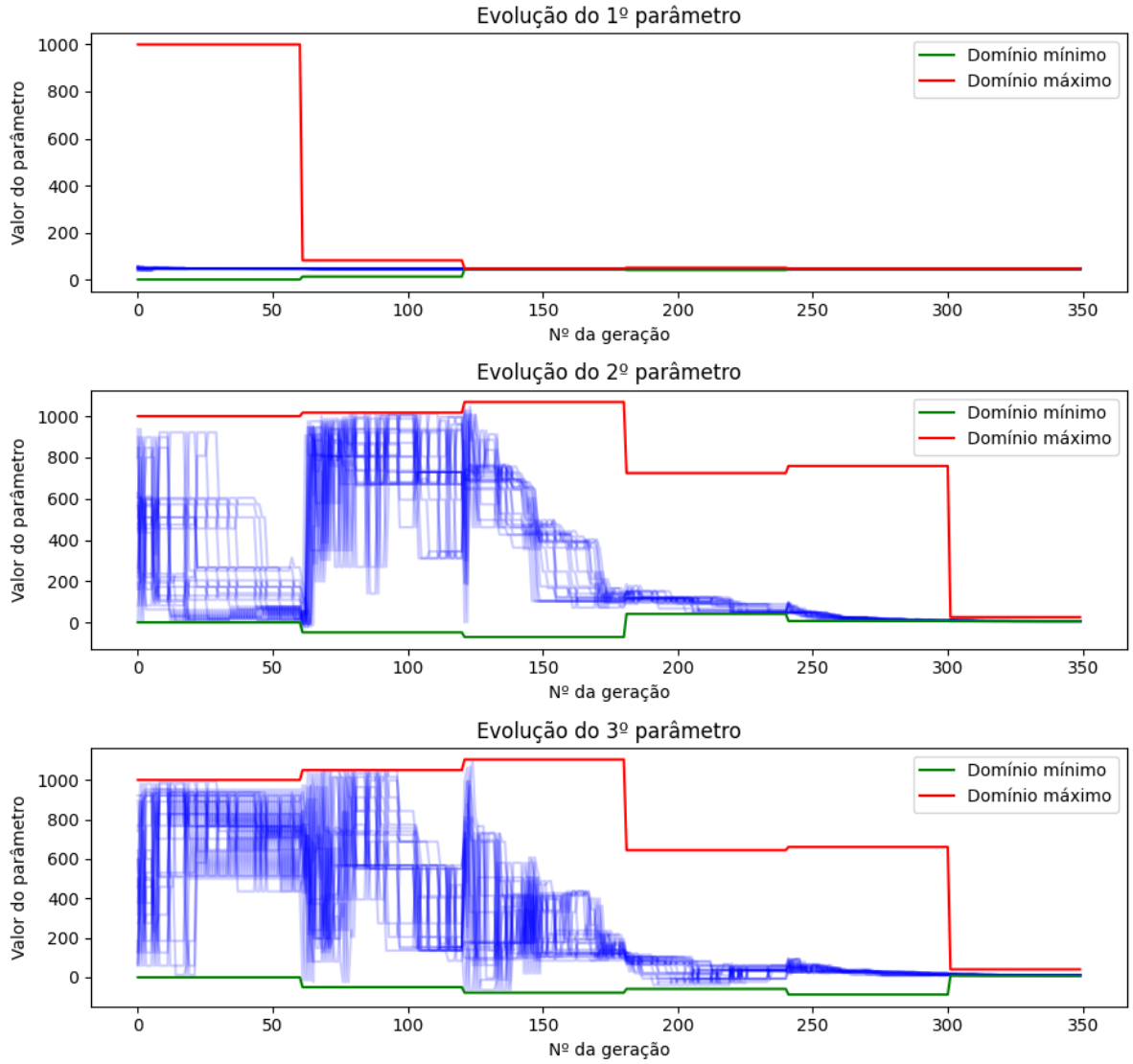

Figura 5.3: Evolução dos domínios e dos parâmetros ao longo das gerações de execução do algoritmo genético

Analisando o gráfico é possível notar como a evolução dos três parâmetros se assemelha ao teste anterior até a 60º geração. No entanto, a partir desta geração o algoritmo nota que o primeiro

parâmetro converge para valores em torno do ponto onde o parâmetro vale 50 unidades e então restringe bruscamente o domínio de busca.

Para os demais parâmetros o algoritmo aumenta o domínio de busca tendo em vista que as melhores soluções parecem buscar valores menores que 0 e maiores que 1000.

Na 120º geração é possível notar que o domínio do primeiro parâmetro converge novamente e a partir deste ponto, ao longo das próximas 60 gerações, é possível notar como os parâmetros avançam para as soluções e em seguida, após sucessivas convergências do domínio, tanto o domínio quanto as melhores soluções convergem precisamente para a solução previamente conhecida.

Para esta execução foram encontrados os seguintes parâmetros:

$$a = 45,011$$

$$\sigma = 0,488$$

$$\mu = 3,156$$

Isso mostra o quão preciso e eficaz este método de otimização do algoritmo genético pode ser.

Este problema acaba por salientar um aspecto importante da busca por parâmetros de ajuste com algoritmo genético, este aspecto é a dependência da convergência entre parâmetros. Evidentemente a busca por parâmetros de ajuste deve levar em conta todos os parâmetros ao mesmo tempo, no entanto, certos parâmetros só irão começar a convergir para a solução com a condição que um parâmetro tenha convergido antes.

Este tipo de comportamento do ajuste de função acontece ocorre em casos onde certos parâmetros atuam como responsáveis por realizar o ajuste fino em uma curva, enquanto que outros parâmetros agem como a base para o ajuste da curva.

Retomando o exemplo da gaussiana. Se torna inviável procurar o parâmetro de ajuste da largura do pico da gaussiana quando se não possui a informação da posição em que a gaussiana está transladada no eixo y.

Em vista disso, o algoritmo de convergência do domínio permite que o domínio dos parâmetros que são inicialmente mais relevantes para a função de avaliação sejam fixados para que então seja possível realizar a buscas mais precisas nos outros parâmetros. Diferentemente do que acontece quando se permite que o algoritmo varre todo o domínio de forma arbitrária.

Reajustando o domínio do primeiro parâmetro para:

$$a \in [500, 1500]$$

Onde o valor do parâmetro se está localizado em $a = 45$. Ao longo de 300 gerações, foi possível obter o gráfico da evolução dos indivíduos e do domínio de busca do parâmetro:

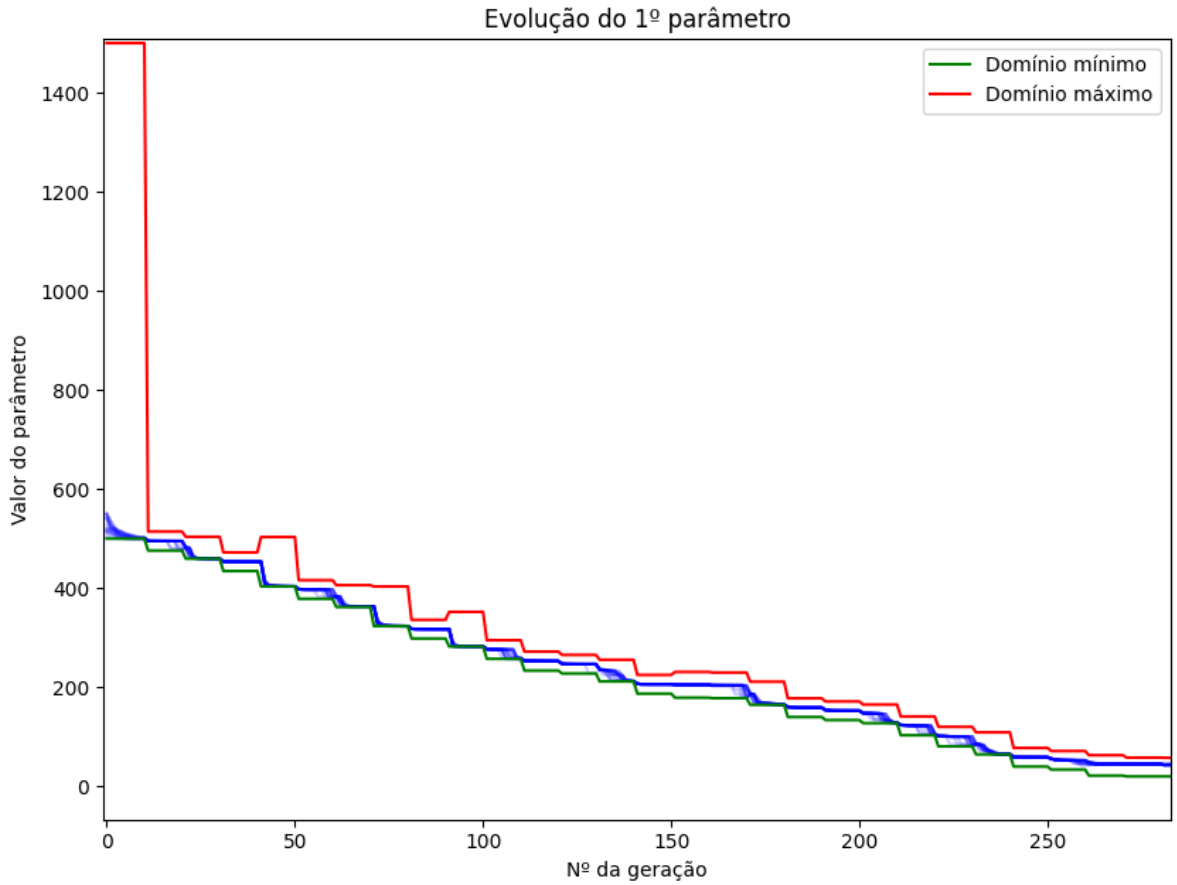

Figura 5.4: Evolução do domínio de busca para o parâmetro a do ajuste da curva dada pela equação 3.1.

5.1.2.2 Ajuste de uma curva

Considerando o problema de se determinar o ajuste do conjunto de dados da figura a seguir:

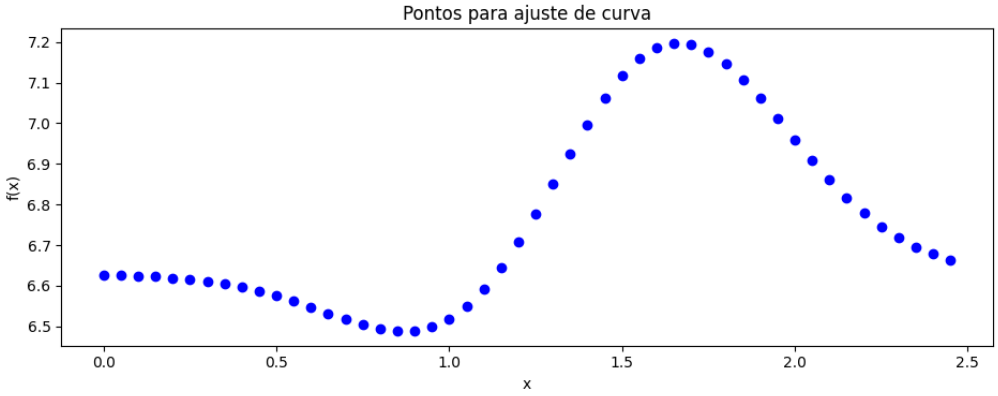

Figura 5.5: Pontos de ajuste de curva.

Com base na curva dada pela equação:

$$f(x) = A + B(x - C)e^{-D(x-B)^2} \tag{5.3}$$

O problema se resume por determinar os quatro parâmetros da função A, B, C e D que caracterizam a curva.

Utilizando o método de convergência do domínio em uma execução com 300 indivíduos, em 350 gerações e convergindo o domínio de busca de cada parâmetro a cada 10 gerações, foi possível traçar a evolução de cada parâmetro ao longo das gerações, como é possível visualizar nas figuras a seguir:

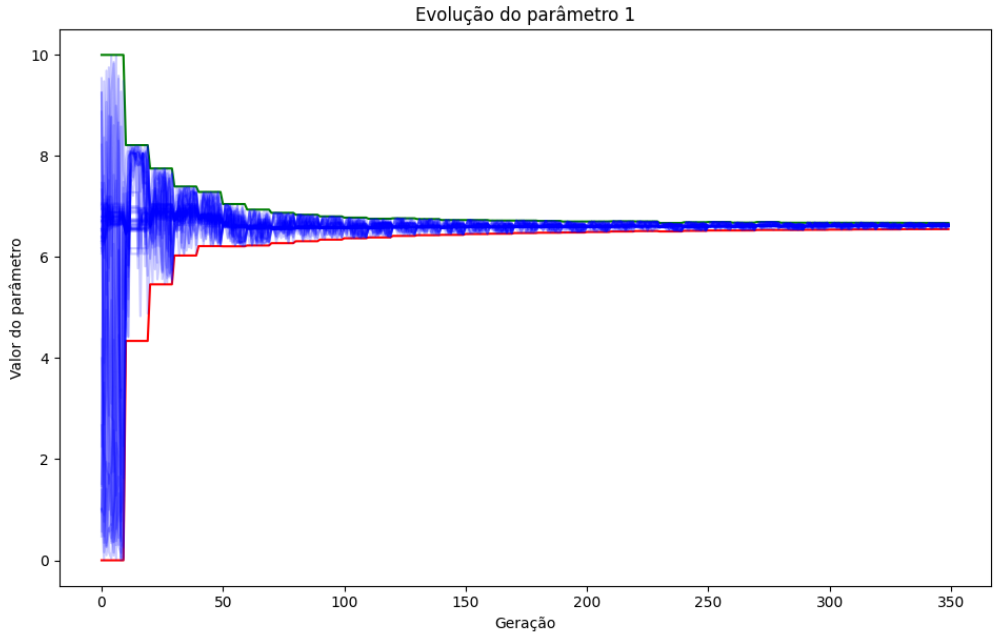

Figura 5.6: Evolução do parâmetro A para o ajuste da curva.

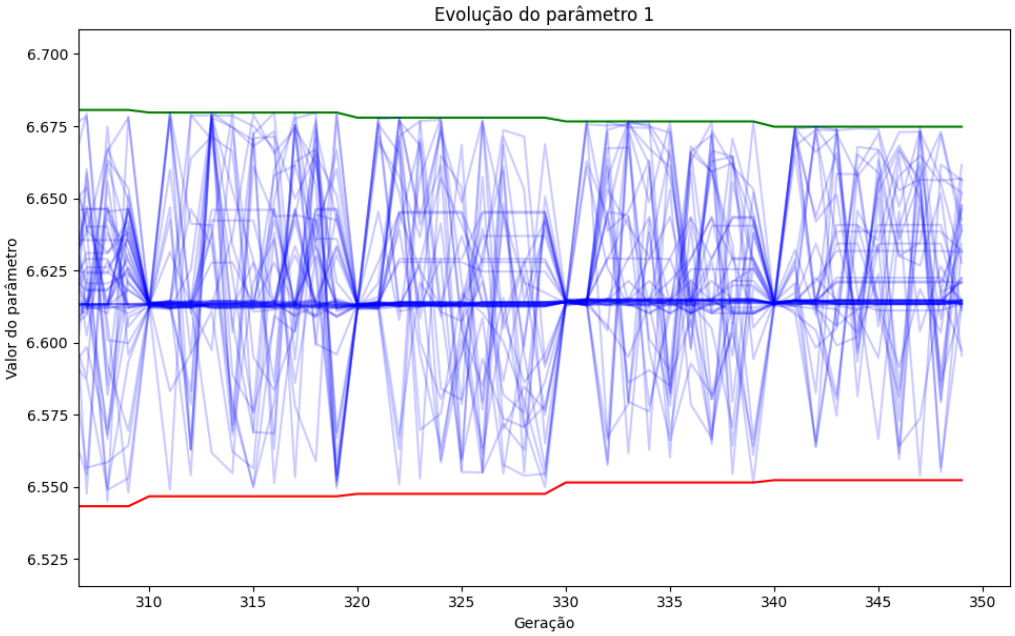

Figura 5.7: Evolução do parâmetro A para o ajuste da curva nas últimas gerações da execução do algoritmo.

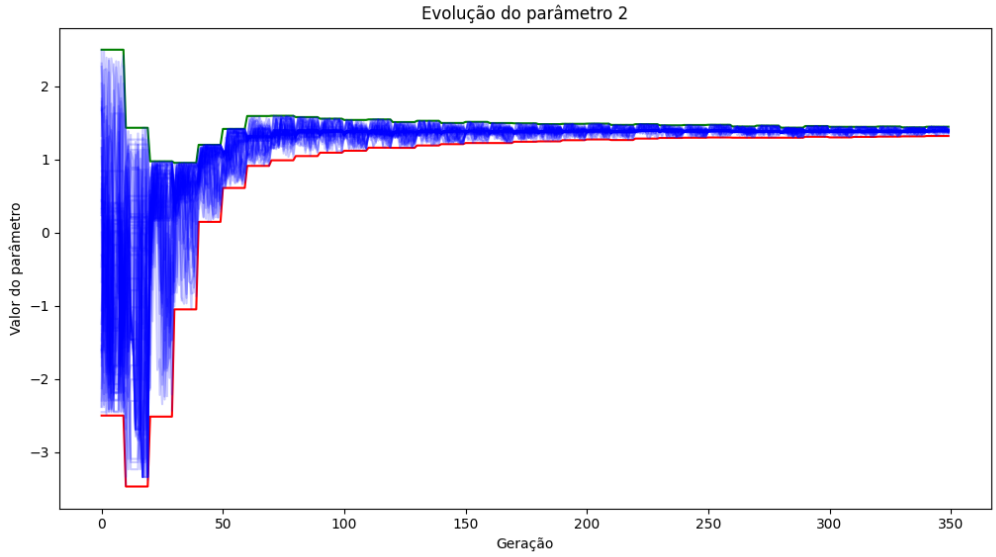

Figura 5.8: Evolução do parâmetro B para o ajuste da curva.

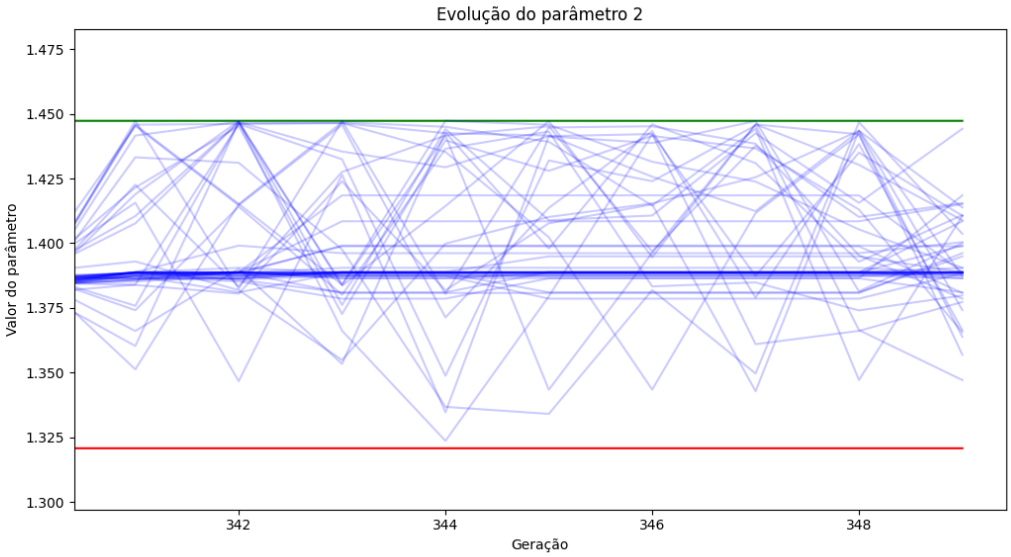

Figura 5.9: Evolução do parâmetro B para o ajuste da curva nas últimas gerações da execução do algoritmo.

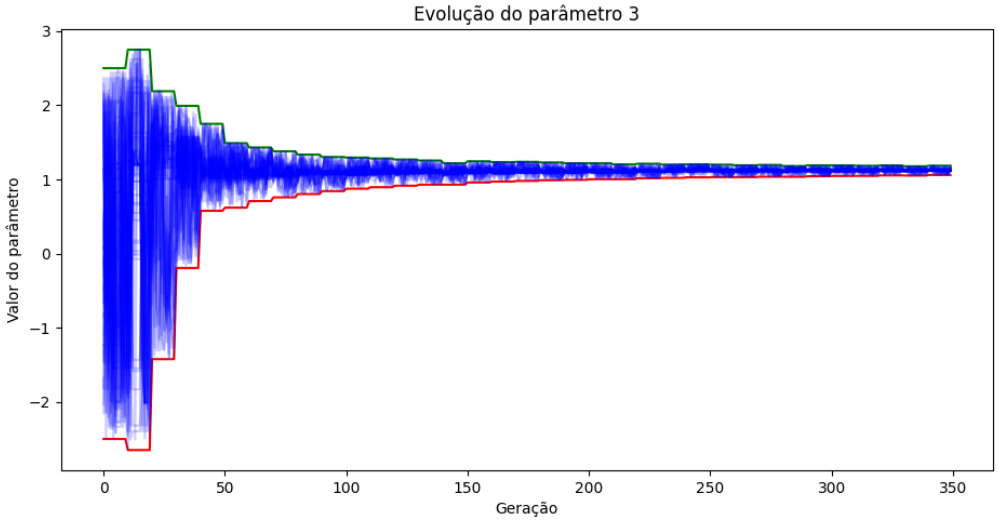

Figura 5.10: Evolução do parâmetro C para o ajuste da curva.

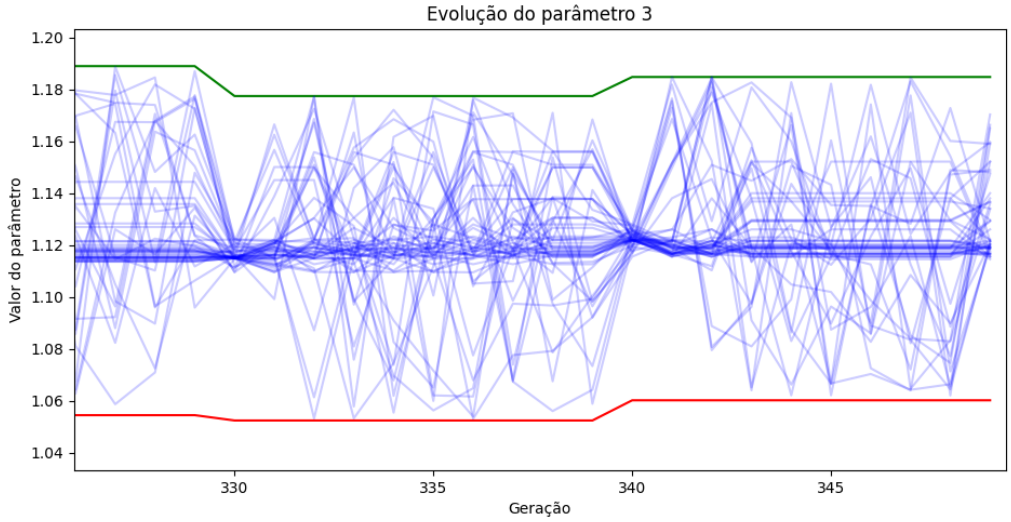

Figura 5.11: Evolução do parâmetro C para o ajuste da curva nas últimas gerações da execução do algoritmo.

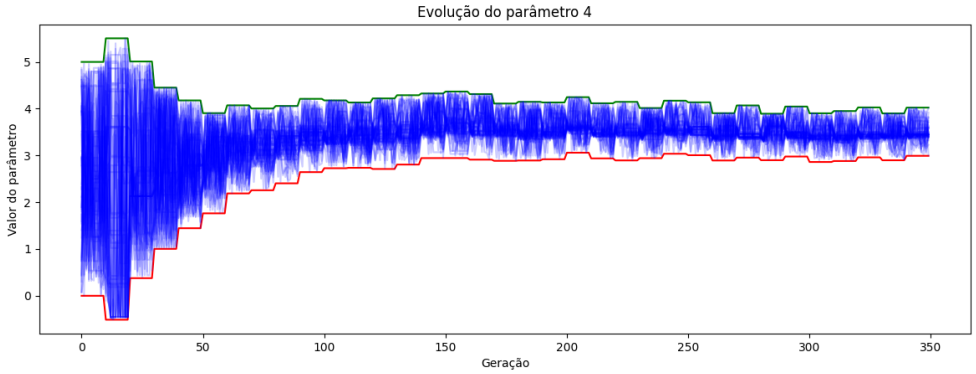

Figura 5.12: Evolução do parâmetro D para o ajuste da curva.

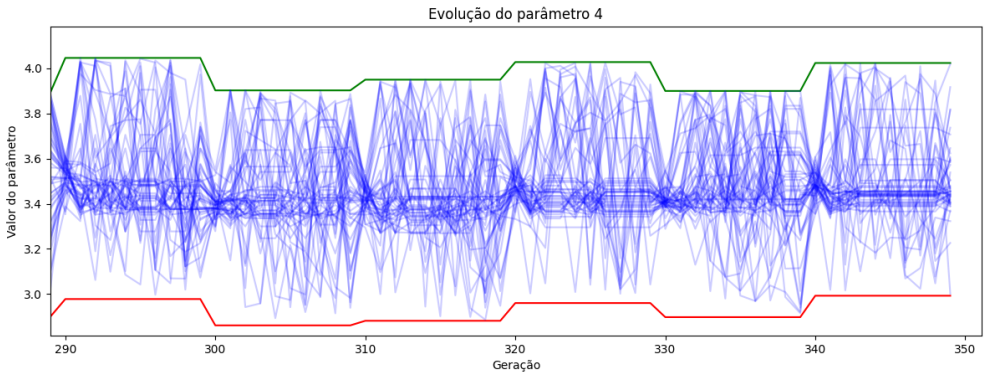

Figura 5.13: Evolução do parâmetro D para o ajuste da curva nas últimas gerações da execução do algoritmo.

Resgatando a evolução da avaliação dos indivíduos ao longo das gerações, foi possível obter o gráfico da figura a seguir:

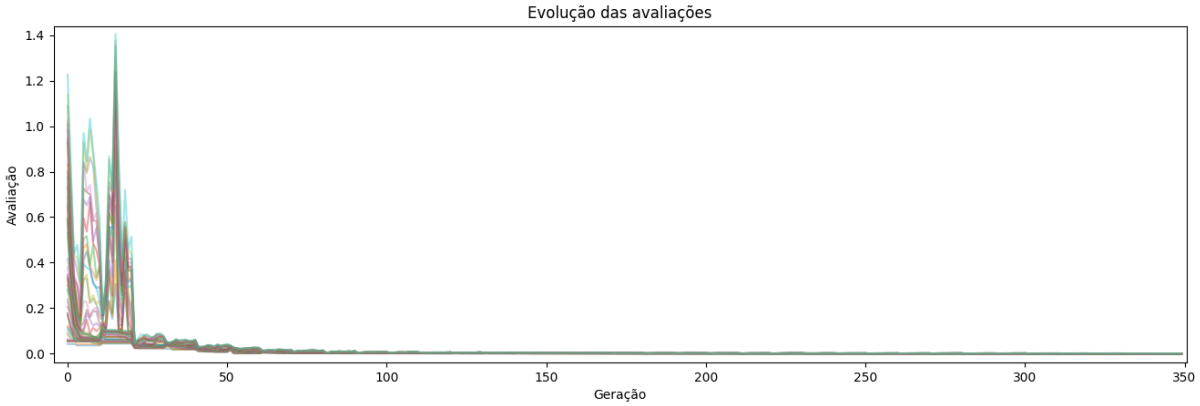

Figura 5.14: Evolução da avaliação dos indivíduos.

Com base nos gráficos da evolução dos parâmetros e na avaliação da população, é possível notar a eficiência do algoritmo em minimizar a função de avaliação quando enquanto converge o domínio de busca de cada parâmetro individualmente.

Observando mais precisamente as primeiras gerações da evolução do parâmetro A na figura a seguir:

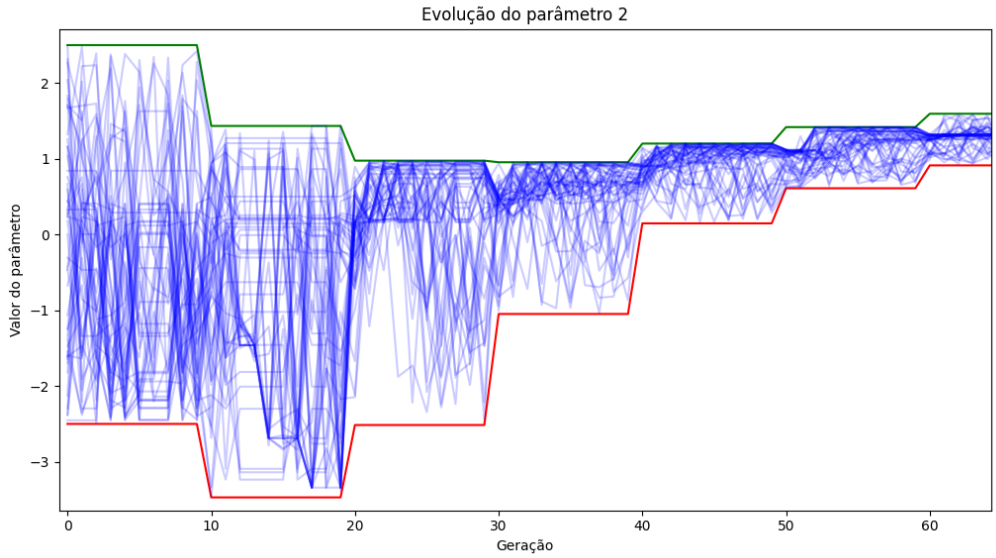

Figura 5.15: Evolução do parâmetro B para o ajuste da curva nas primeiras gerações da execução do algoritmo.

É possível observar que o domínio inicialmente possui uma tendência de se deslocar para valores negativos, quando na 20ª geração o domínio de todos os parâmetros é convergido mais uma vez e então o fluxo de convergência do domínio deste parâmetro toma um rumo diferente, caminhando mais precisamente para a solução a cada vez que o domínio é reduzido. Como é possível visualizar na avaliação das primeiras gerações:

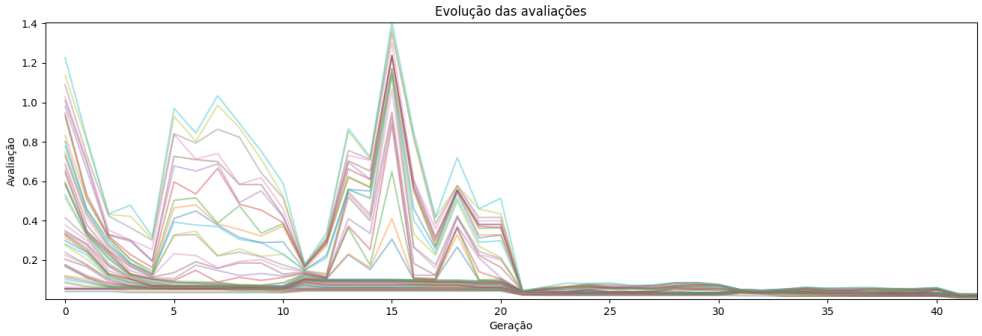

Figura 5.16: Evolução da avaliação dos indivíduos nas primeiras gerações.

O método de convergência do domínio possui uma grande vantagem de gerar uma alta precisão na busca de valores do domínio.

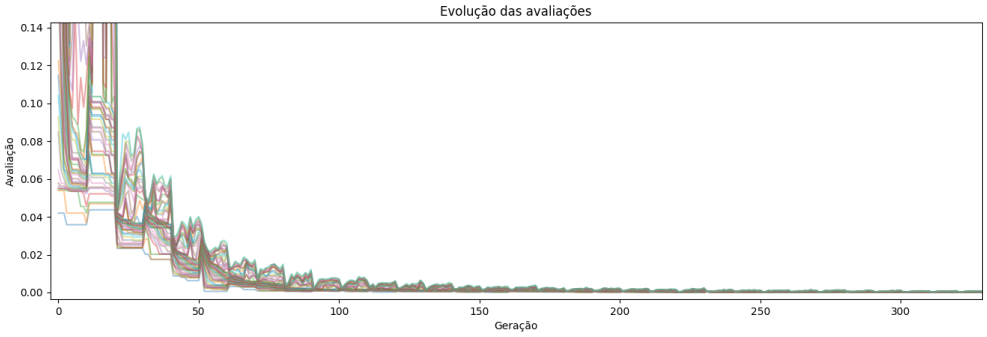

Figura 5.17: Evolução da avaliação dos indivíduos.

No gráfico da figura acima é possível observar como a evolução das avaliações possuem um comportamento periódico de queda que acontece a cada 10 gerações. Este comportamento é marcado justamente pela redução do domínio de busca de cada parâmetro em torno dos valores dos melhores indivíduos.

Plotando as curvas com os parâmetros dos melhores indivíduos, foi possível obter o gráfico das curvas obtidas na figura a seguir:

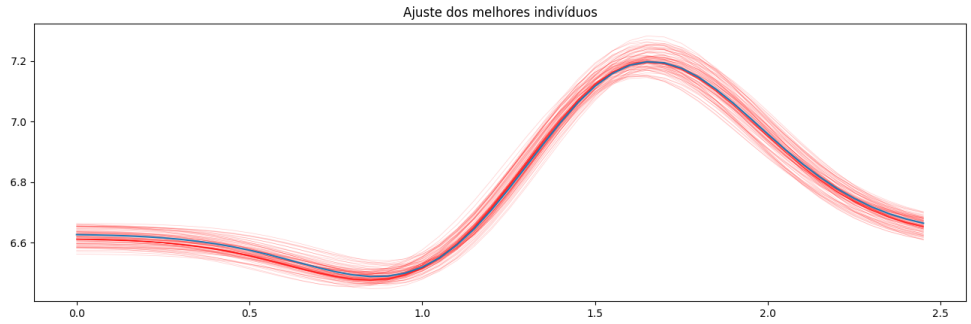

Figura 5.18: Curvas obtidas com base nos parâmetros dos 200 melhores indivíduos da última geração.

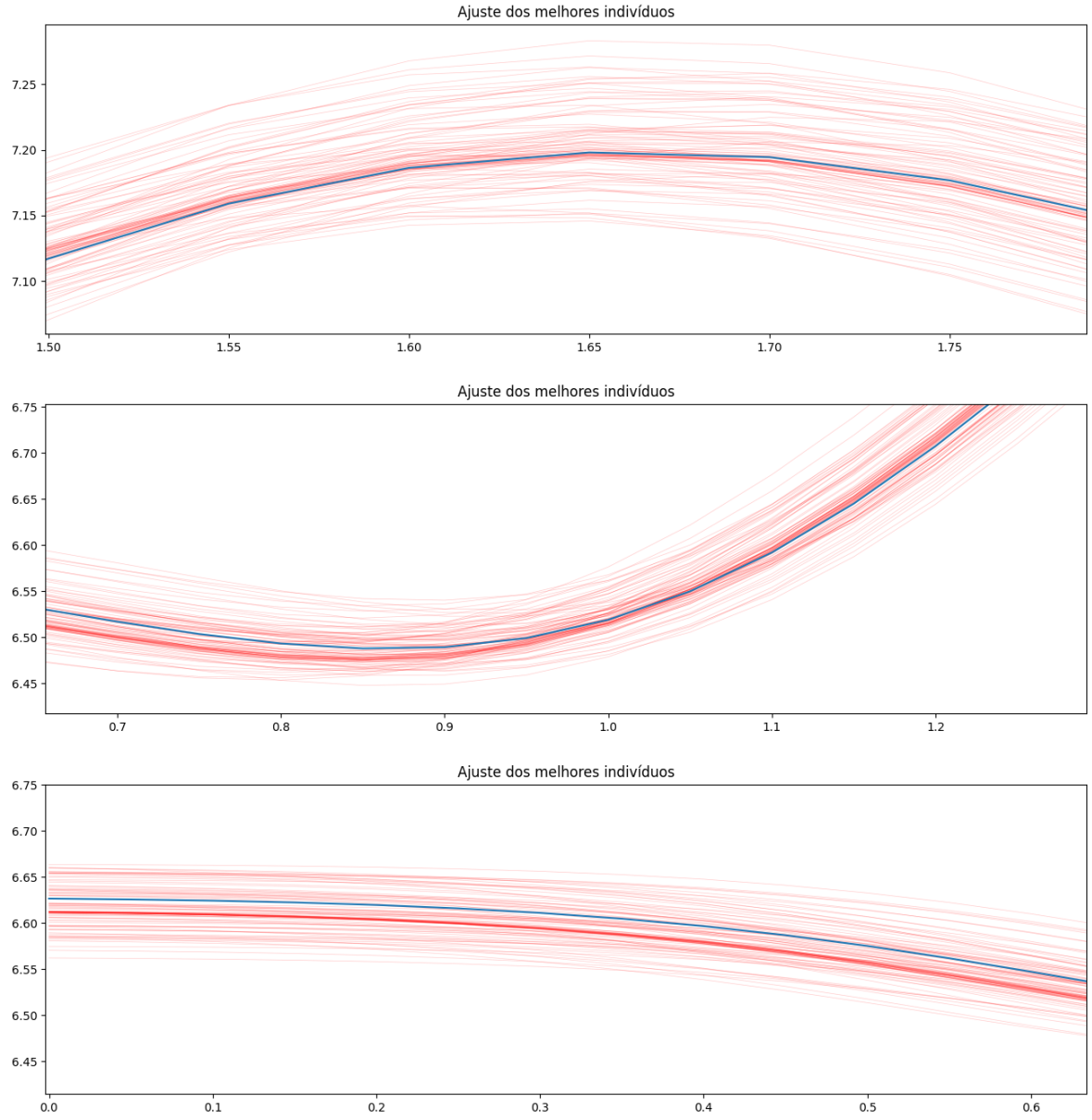

Figura 5.19: Curvas obtidas com base nos parâmetros dos 200 melhores indivíduos da última geração.

Com base no gráfico da figura acima é possível notar como os 200 melhores indivíduos conseguiram se ajustar consideravelmente bem bem ao comportamento da curva.

Realizando uma execução do algoritmo com as exatas mesmas configurações, porém sem utilizar o método de convergência foi possível obter os gráficos de evolução dos parâmetros ilustrados na figura a seguir:

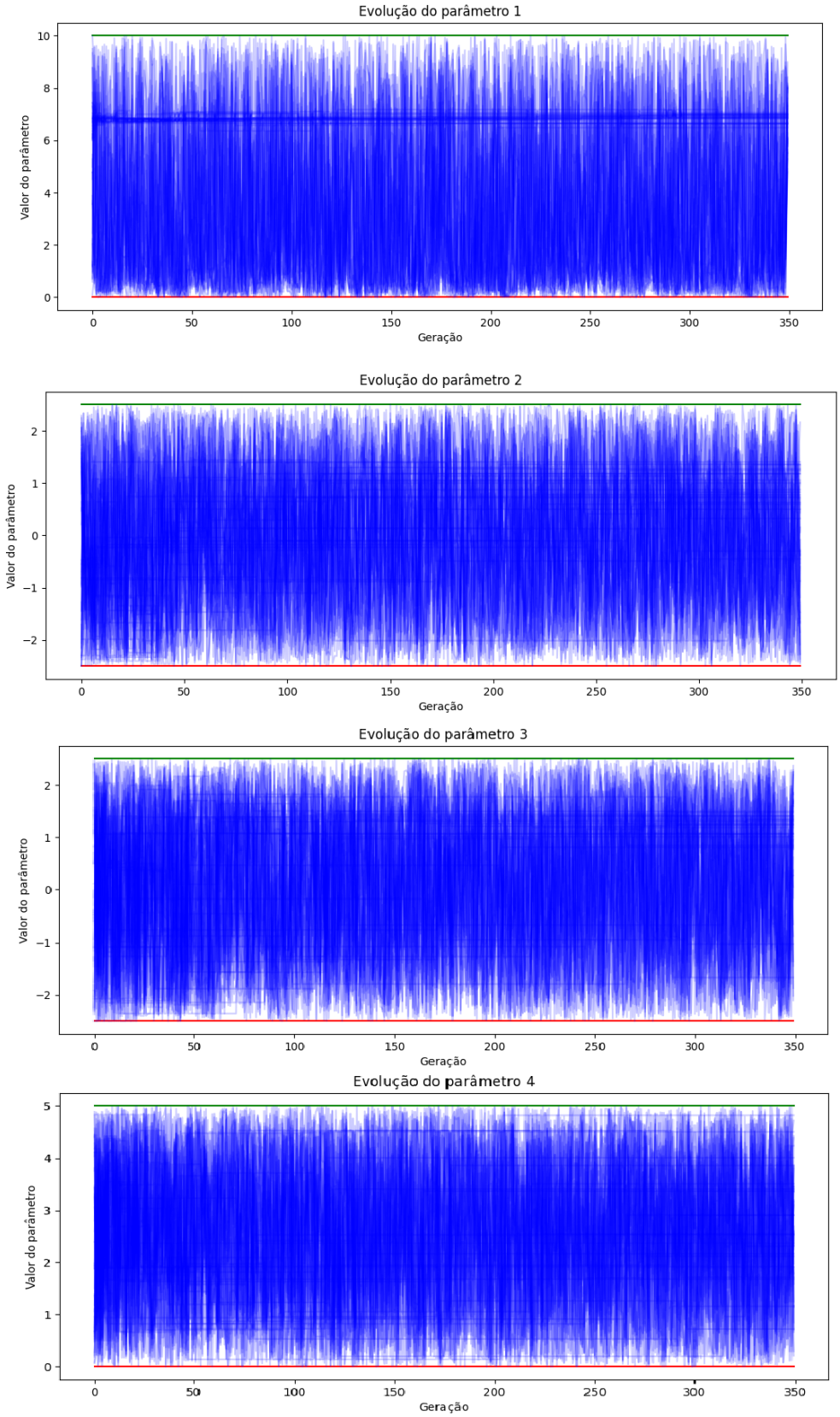

Figura 5.20: Evolução dos 4 parâmetros de ajuste da curva ao longo das 350 gerações

Cujas avaliações são traçadas na figura a seguir:

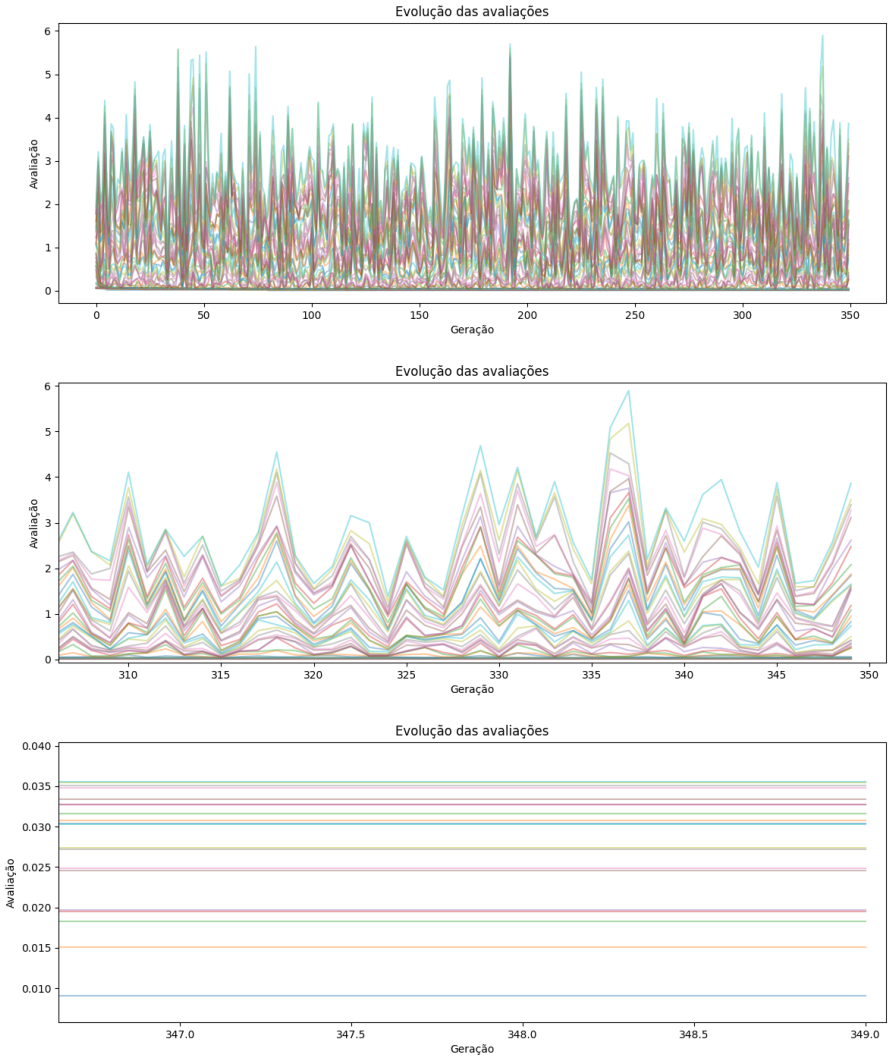

Figura 5.21: Evolução das avaliações dos indivíduos ao longo das gerações sem a utilização do método de convergência.

E com base nos parâmetros dos melhores indivíduos da última geração, foi possível traçar as curvas de ajuste para o algoritmo:

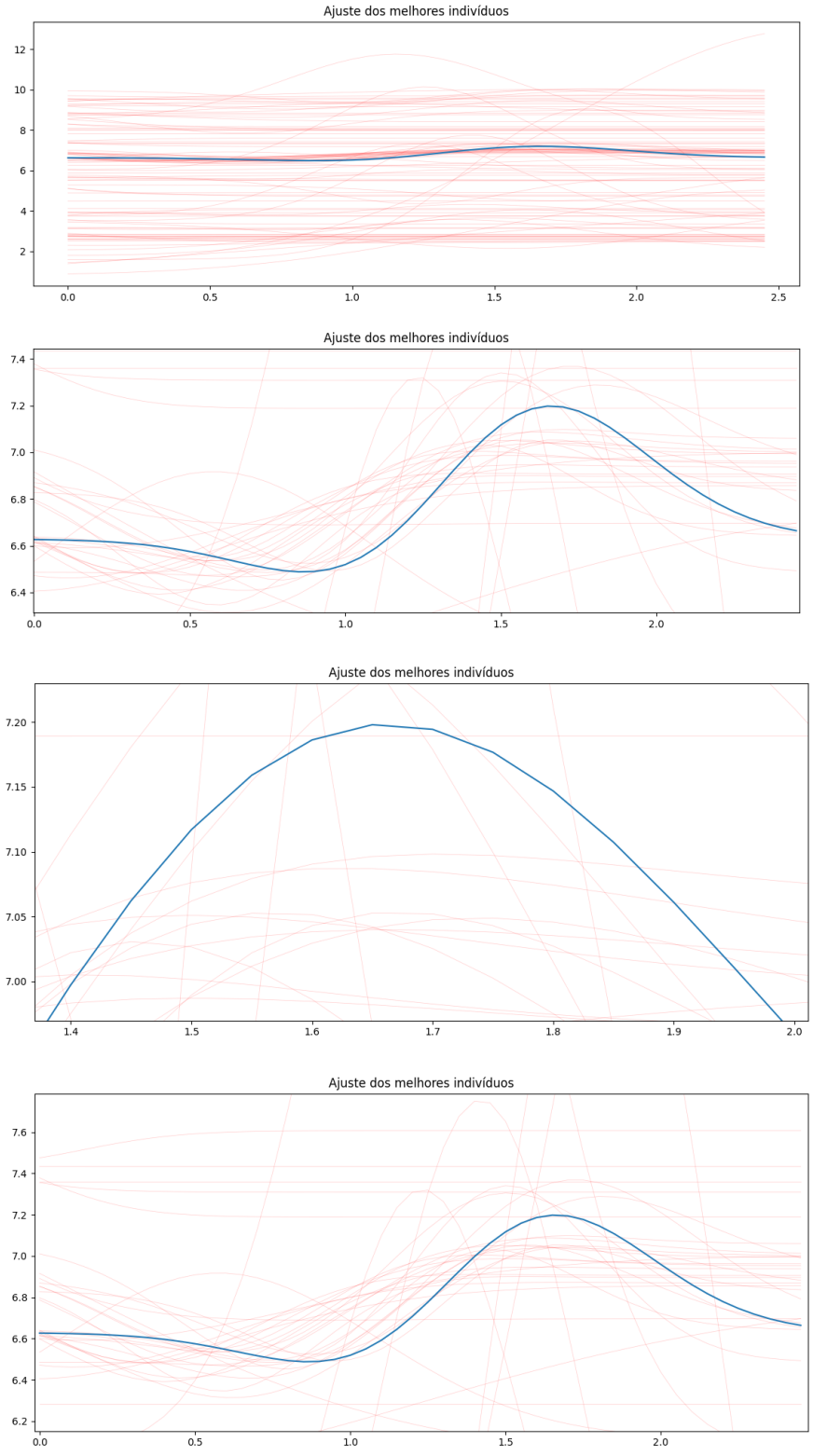

Figura 5.22: Curvas obtidas com base nos parâmetros dos 200 melhores indivíduos da última geração sem o método de convergência do domínio.

Na execução do algoritmo genético sem a utilização do método, com base nos gráficos de

evolução dos parâmetros, é possível observar que a população permanece dispersa sob o domínio de busca ao longo de toda a execução e consequentemente a avaliação dos indivíduos também permanece caótica ao longo de toda a execução, mesmo que os indivíduos elite permaneçam nas melhores configurações, conservando as melhores avaliações.

Na execução do algoritmo com o método de convergência o mesmo comportamento se repete, no entanto ao reduzir o domínio de busca os indivíduos ficam dispersos em domínios mais próximos das melhores configurações e por consequência exploram os domínios que se apresentam mais significantes. Sendo assim o método permite que o algoritmo concentre todos os seus indivíduos em uma busca mais precisa das soluções ao longo das gerações.

5.1.2.3 Busca de solução fora do domínio

O método de convergência funciona deslocando o domínio para a região onde os melhores indivíduos se concentram, dessa forma é possível que o método busque soluções para os parâmetros do algoritmo em até mesmo pontos que estejam foram do domínio de busca informado.

Executando o algoritmo para a determinação do ponto máximo da superfície da figura 2.4, onde o ponto máximo está localizado nas coordenadas $[0.3, 0.3]$, e informando o domínio de busca como sendo:

$$x \in [2, 5]$$

$$y \in [-5, -2]$$

Executando o algoritmo utilizando o método ao longo de 350 gerações, obteve-se os seguintes gráficos da evolução das soluções:

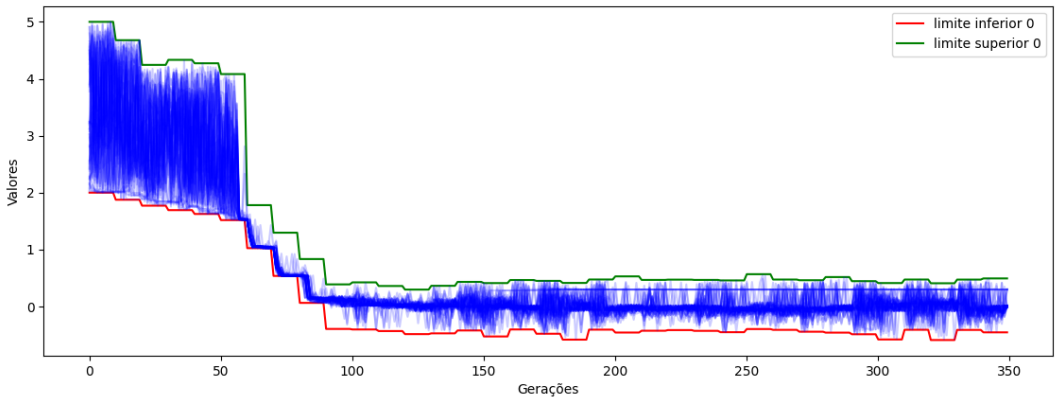

Figura 5.23: Evolução dos indivíduos que representam a coordenada x ao longo gerações da execução do algoritmo.

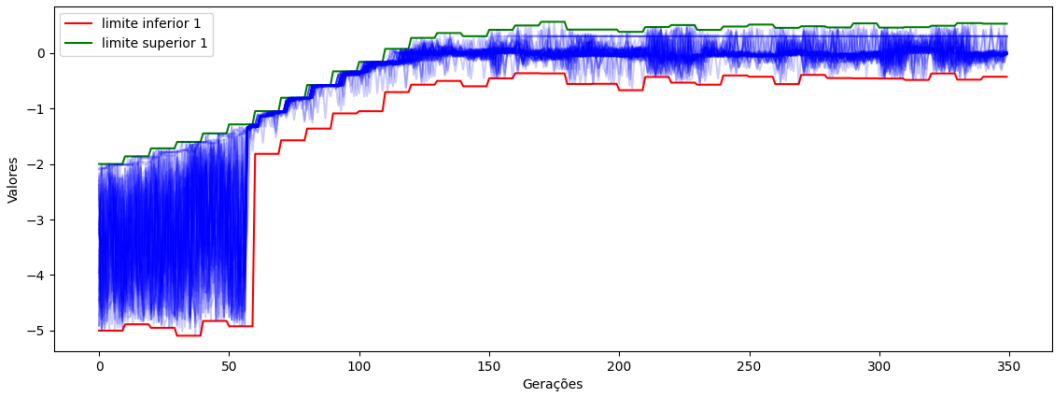

Figura 5.24: Evolução dos indivíduos que representam a coordenada y ao longo gerações da execução do algoritmo.

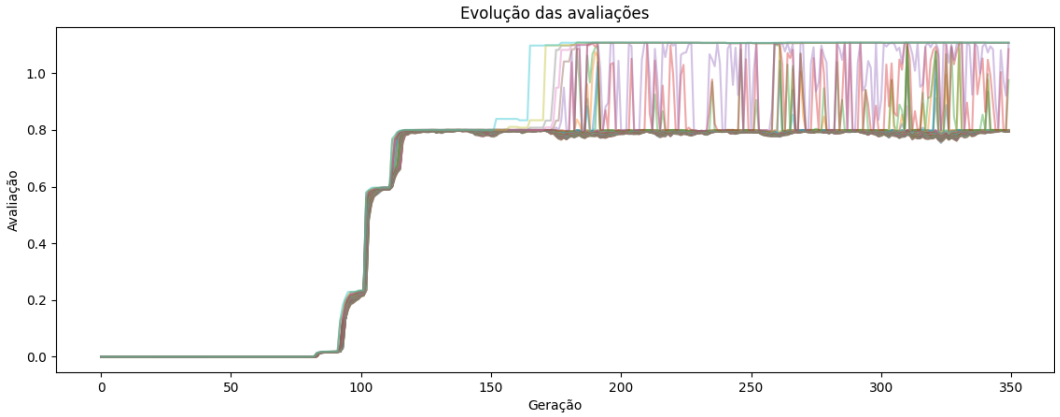

Figura 5.25: Evolução da avaliação dos indivíduos ao longo gerações da execução do algoritmo.

Com base nos gráficos das figuras acima, é possível observar como o método de otimização é capaz de percorrer até mesmo domínios não informados em busca de melhores soluções.

Com isso é possível notar que este método aplicado ao algoritmo não só viabiliza uma busca mais precisa das soluções, quanto permite que o algoritmo busque por soluções fora do domínio informado.

O método no entanto possui a grande desvantagem de não ser capaz de realizar buscas em domínios muito largos sem que o domínio seja convirja para um domínio estreito de um ponto máximo local e chegando a este ponto a população dificilmente deixará o ponto de máximo local.

5.2 DIVISÃO E CONQUISTA

Uma estratégia otimização implementada junto ao algoritmo genético é a técnica chamada "divisão e conquista".

Este método consiste em inicialmente subdividir um problema em n partes menores, esses subproblemas são resolvidos de forma independente e por fim as n soluções corroboram para a

construção de uma única solução ou a geração de um novos subproblemas de maneira recursiva.

A estrutura de funcionamento deste método pode ser visualizada na no diagrama da figura a seguir:

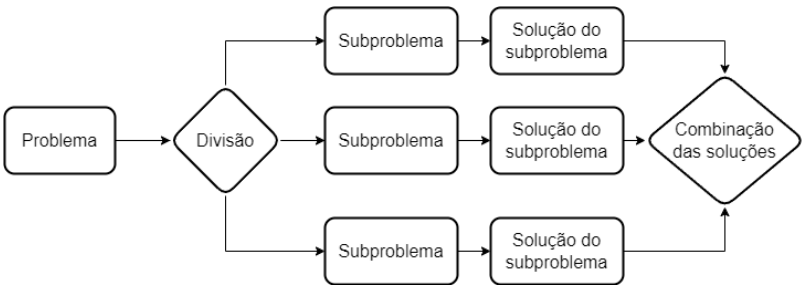

Figura 5.26: Diagrama de funcionamento do método de divisão e conquista.

As busca por soluções com algoritmo genético podem se tornar tão complexas quanto maior o domínio de busca e o número de parâmetros a serem buscados. Em vista disso, o método quando aplicado ao algoritmo genético busca subdividir o domínio de busca das soluções em n partes e resolve-las separadamente.

O método de divisão e conquista ser ampliado para n dimensões, dessa forma se cada dimensão for repartida em 2 subdimensões, o problema se divide em 2^n fragmentos, com execuções independentes em um domínio de busca 2^n vezes menor que o original.

Com isso o número de subdomínios gerados na divisão do domínio é dado pela seguinte relação:

$$N = d^n \tag{5.4}$$

Onde N representa o número de subdomínios, d o número de divisões realizadas em cada domínio e n o número de dimensões do problema.

Este método viabiliza a busca por soluções de maneira mais dispersa e mais independente das condições iniciais da população do algoritmo genético, uma vez que explora cada subdivisão do domínio mais precisamente, evitando comportamentos que levam a população a convergir rapidamente para uma primeira solução e permitindo que cada parte do domínio seja testada por mais gerações e indivíduos.

5.2.1 Implementação do método

A implementação do método de divisão e conquista aplicado ao algoritmo genético pode ser realizada com base no seguinte pseudo-código:

```
1 Define um dominio de busca inicial para d dimensoes
2 dominio = [[-100,100],[-100,100],[-100,100]]
```

```
Subdivide cada dimensao em n partes, gerando n**d subdomnios
subdominios = divide_dominios(dominio,4)
Para cada subdominio
for sudmonio in subdominios:
    Empilha uma thread com a execucao do algoritmo para o subdominio
Une  threads ao fim de todas execucoes
Coleta as melhores avaliacoes
```

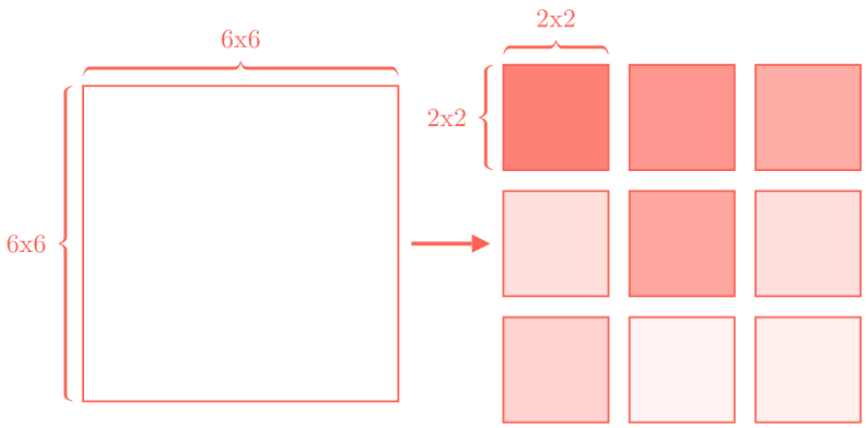

Figura 5.27: Diagrama de funcionamento do método de divisão e conquista aplicado ao algoritmo genético.Neste diagrama a intensidade da cor de preenchimento dos domínios representa a avaliação do melhor indivíduo.

Com isso o método consegue obter as melhores avaliações com execuções em cada subdomínio. Dessa forma é possível então concentrar as execuções no domínio mais que se mostrar mais pertinente ou seguir com o processo de divisão e conquista, subdividindo novamente o novo domínio quantas vezes se mostrar necessário.

O processo recorrente de divisão e conquista pode ser facilmente implementado com base no seguinte pseudo-código:

```
Define um dominio de busca inicial para d dimensoes
dominio = [[-100,100],[-100,100],[-100,100]]
Para n divisoes:
for i in range(n)
    Subdivide cada dimensao em n partes, gerando n**d subdomnios
    subdominios = divide_dominios(dominio,4)
    Para cada subdominio
    for sudmonio in subdominios:
        Empilha uma thread com a execucao do algoritmo para o subdominio
    Une as threads ao fim de todas execucoes e coleta a melhor avaliacao de cada
    Com base nas avaliacoes, define o melhor subdominio como o dominio
    dominio = melhor_subdominio
    Reinicia o processo
```

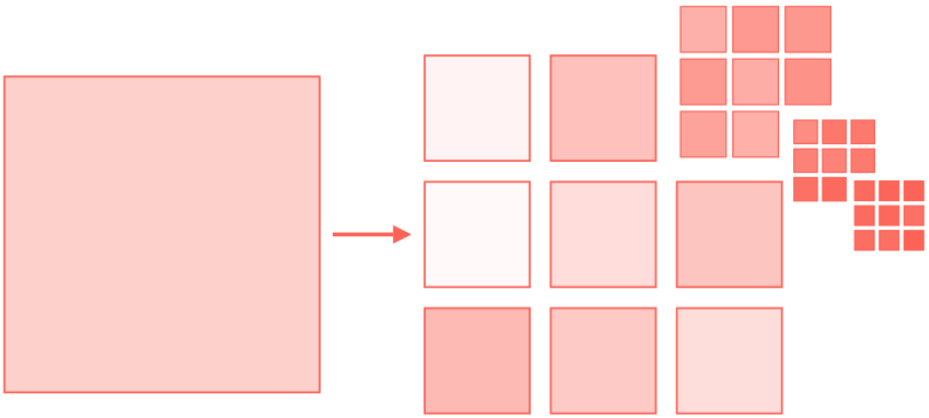

Figura 5.28: Diagrama de funcionamento do método recursivo de divisão e conquista aplicado ao algoritmo genético. Neste diagrama a intensidade da cor de preenchimento dos domínios representa a avaliação do melhor indivíduo.

O funcionamento do método aplicado ao algoritmo genético também pode ser descrito pelo diagrama da figura a seguir:

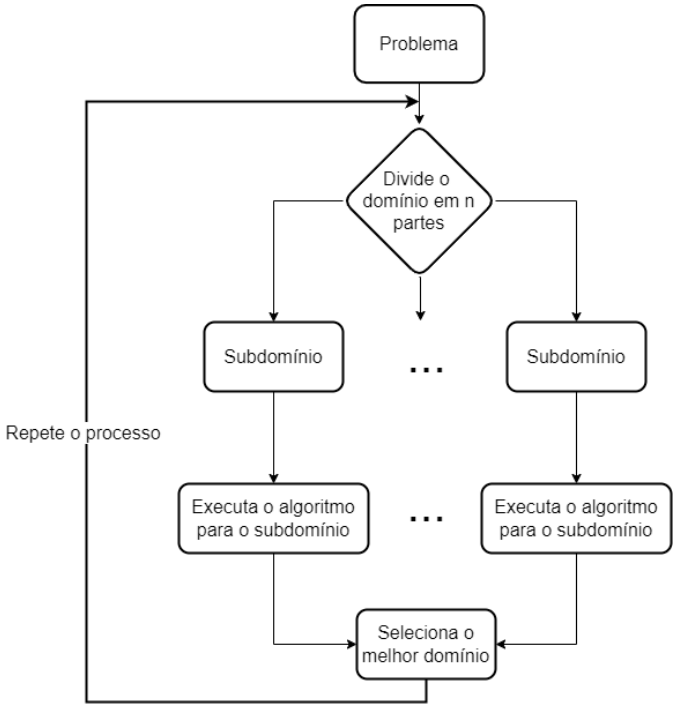

Figura 5.29: Diagrama de funcionamento do método de divisão e conquista recorrente aplicado ao algoritmo genético.

Com isso o domínio poderá ser subdividido em n vezes com base nos subdomínios que forem melhor avaliados.

No entanto a utilização da recorrência no método de divisão e conquista pode ser melhor aproveitado se utilizado com o intuito de subdividir não apenas o melhor domínio, mas uma

combinação dos domínios mais bem avaliados.

Dessa forma uma implementação mais sólida do método pode ser desenvolvida com base no seguinte pseudocódigo:

```
Define um dominio de busca inicial para d dimensoes
dominio = [[-100,100],[-100,100],[-100,100]]
Inicia um loop perpetuo
while True:
    Subdivide cada dimensao em n partes, gerando n**d subdom nios
    subdominios = divide_dominios(dominio,4)
    Para cada subdominio
    for sudmonio in subdominios:
        Empilha uma thread com a execucao do algoritmo para o subdominio
    Une as threads ao fim de todas execucoes e coleta a melhor avaliacao de cada
    Adiciona as avaliacoes com os respectivos dominios em um arquivo de saida
    Le o arquivo de saida, resgatando os melhores dominios
    Gera um dominio a partir da combinacao dos dominios selecionados
    dominio = dominio_combinado
    Reinicia o processo
```

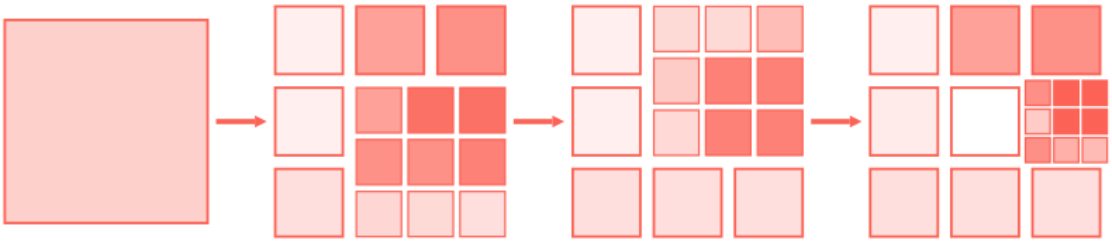

Figura 5.30: Diagrama de funcionamento do método recursivo de divisão e conquista aplicado ao algoritmo genético. Neste diagrama a intensidade da cor de preenchimento dos domínios representa a avaliação do melhor indivíduo.

O funcionamento do método aplicado ao algoritmo genético também pode ser descrito pelo diagrama da figura a seguir:

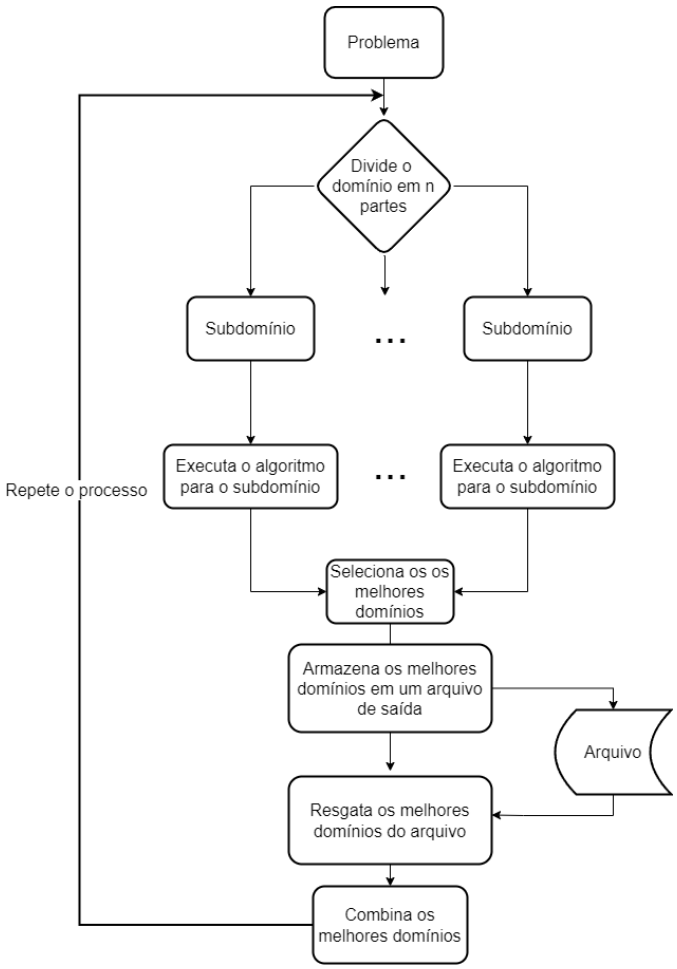

Figura 5.31: Diagrama de funcionamento do método de divisão e conquista aplicado ao algoritmo genético.

Dessa forma o domínio de busca irá sempre se concentrar no perímetro que cerca as melhores soluções. Este processo, assim como todos os métodos aplicados juntamente com algoritmo genético, jamais garantirá que a solução será encontrada, no entanto é uma ferramenta eficaz na exclusão de domínios de busca que não favorecem a solução, pois não permite que o domínio convirja mais que o necessário e também pode ser utilizado para longas buscas, pois possui a vantagem de tanto subdividir o domínio como viabilizar que o algoritmo seja reexecutado indefinidas vezes com diferentes populações iniciais, sem que seja necessário o gerenciamento das execuções por parte do usuário.

5.2.2 Testes de otimização

A fim de realizar testes consistentes da otimização da busca baseada no método de divisão e conquista, utilizou-se uma superfície que consiste em 62 gaussianas localizadas em posições aleatórias cujo ponto máximo está localizado em uma estreita gaussiana no ponto $(1.02753, 1.5793)$. A superfície é expressa pela seguinte equação:

$$f(x,y) = \sum_{i=0}^{62} a_i e^{-\frac{(x-b_i)^2+(y-c_i)^2}{d_i}} \tag{5.5}$$

E pode ser visualizada na figura a seguir:

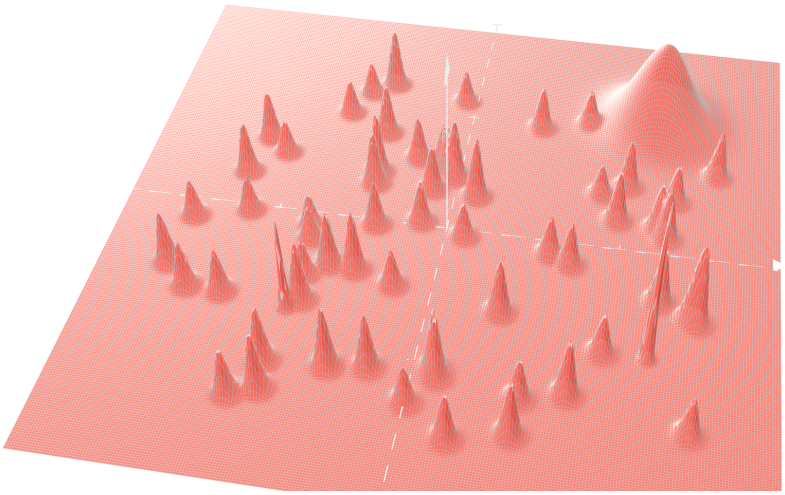

Figura 5.32: Superfície de teste para execução do método de divisão e conquista.

Aplicando o método de divisão e conquista onde inicialmente se subdivide cada dimensão em 3 partes, o domínio se divide em 9 subdomínios. A partir disto o problema sucede em determinar o ponto de máximo de cada domínio, sendo cada um 9 vezes menor que o domínio anterior. Como é possível visualizar na figura a seguir:

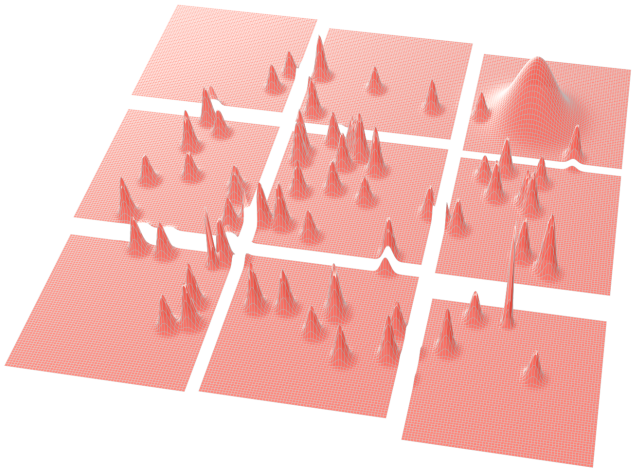

Figura 5.33: Diagrama de funcionamento do método de divisão e conquista aplicado ao algoritmo genético.

Após o fim de cada execução, caso a melhor solução dentre todas encontradas seja suficiente o programa se encerra, do contrário o domínio das melhores soluções são combinados e então

subdivididos, dando início a um novo conjunto de execuções. Este processo se repete indefinidamente até que uma solução suficientemente boa seja encontrada ou até que o programa seja interrompido.

Com base na execução do algoritmo para cada subdivisão do domínio da superfície em questão com 200 indivíduos em 30 gerações, foi possível traçar a evolução dos melhores indivíduos, ilustrada na figura a seguir:

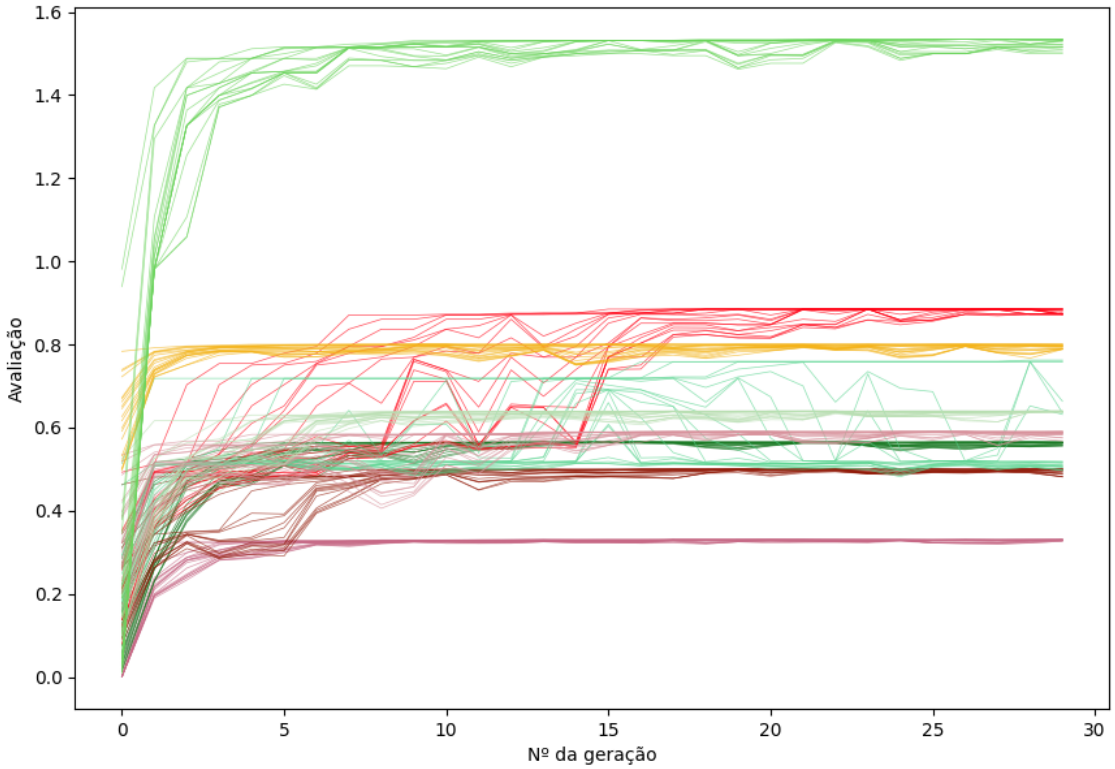

Figura 5.34: Evolução da avaliação dos melhores indivíduos nas subdivisões do domínio da superfície.

Neste gráfico, onde cada cor representa a execução em cada domínio, se torna claro a capacidade do algoritmo em mapear máximos locais de todo o domínio ao analisar o problema em pequenas frações.

A execução do método segue com a seleção do melhor domínio e então mais uma vez a subdivisão, vista na figura a seguir:

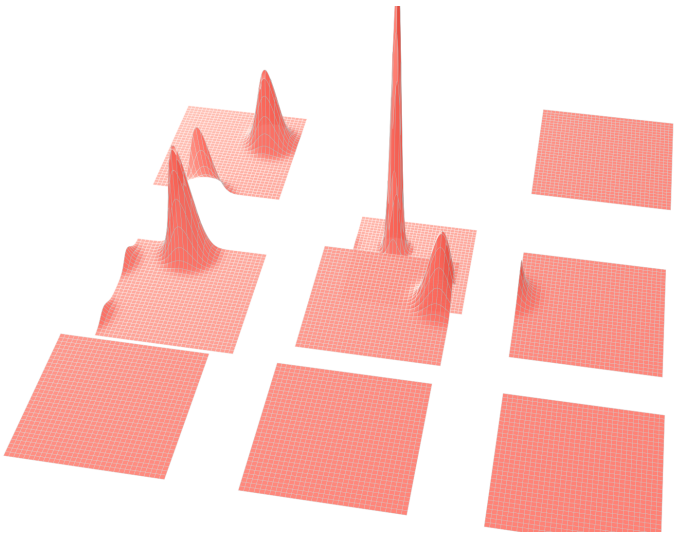

Figura 5.35: Diagrama de funcionamento do método de divisão e conquista aplicado ao algoritmo genético.

Dessa forma realizando a execução nos novos subdomínios se obtêm a evolução dos melhores indivíduos em cada dimensão, ilustrada na figura a seguir:

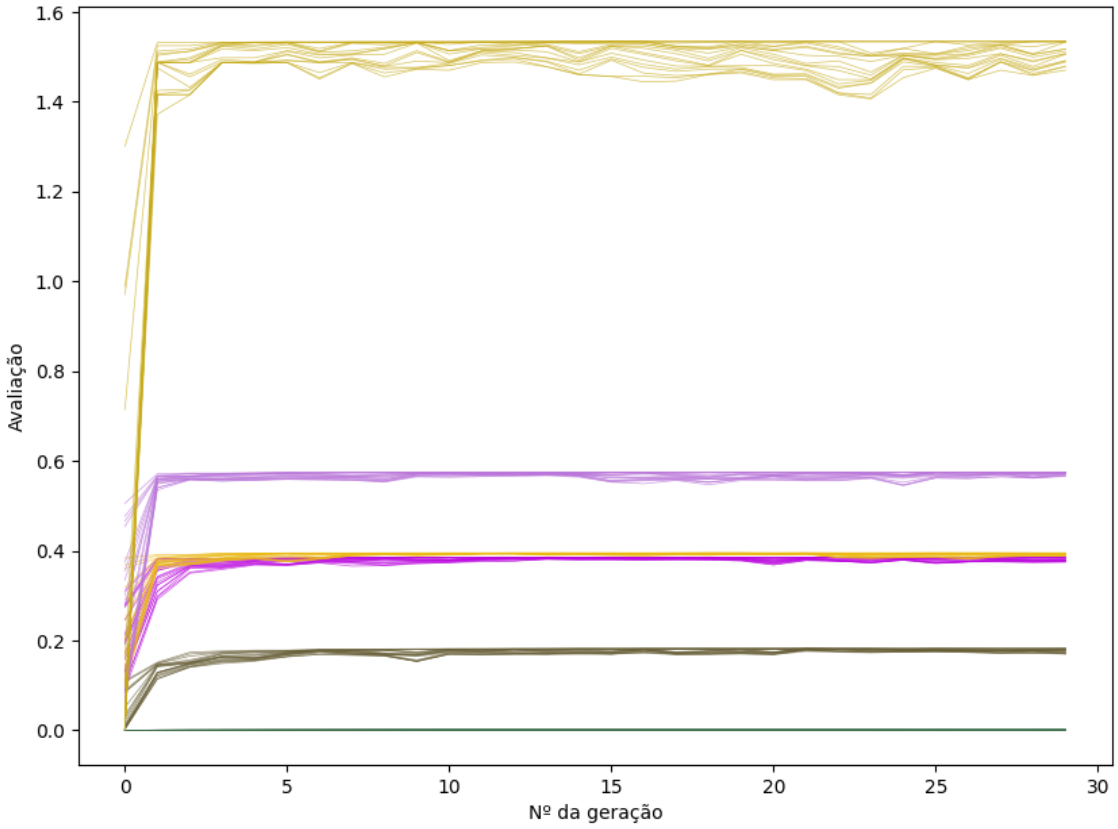

Figura 5.36: Evolução da avaliação dos melhores indivíduos nas subdivisões do domínio da superfície.

Diferentemente do comportamento da evolução dos indivíduos em uma busca realizada por 9 execuções do algoritmo em todo o domínio e com as mesmas configurações. Como é possível visualizar no gráfico da figura a seguir:

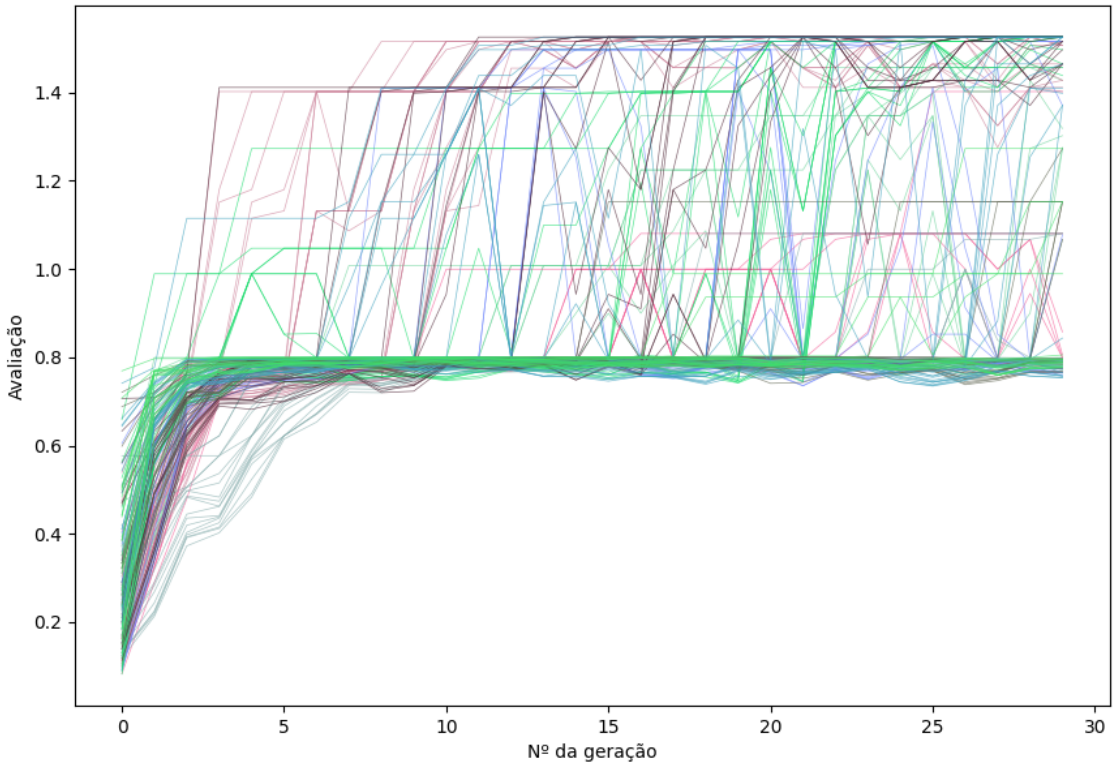

Figura 5.37: Evolução da avaliação dos melhores indivíduos em 9 execuções do algoritmo em todo o domínio.

Utilizando o problema de ajuste da função gaussiana dada pela equação 3.1, ao subdividir cada uma das 3 dimensões em 5 partes o problema se resume a realizar uma busca em 125 domínios que são 125 vezes menor que o original. Utilizando-se 2000 indivíduos em 50 gerações, foi possível traçar a evolução da busca em cada subdomínio, como é possível visualizar na figura a seguir:

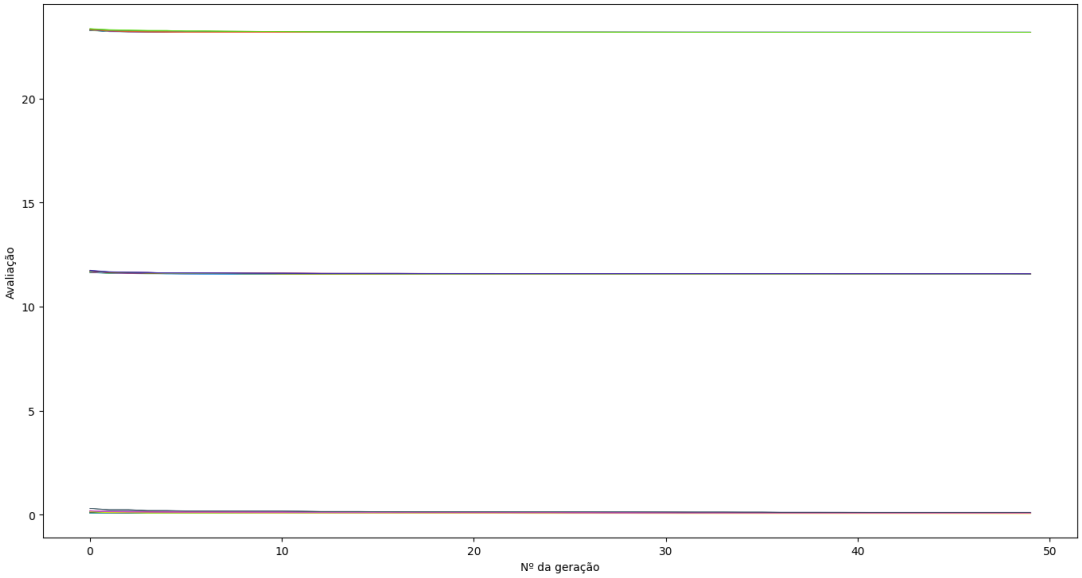

Figura 5.38: Evolução da avaliação dos melhores indivíduos nas subdivisões do domínio de busca para ajuste da função 3.1.

Ao se tratar do ajuste de uma função, o valor da avaliação representa o quão próximo cada indivíduo está da configuração desejada, sendo assim a qualidade de um indivíduo está associada ao quanto menor for o valor da avaliação. No caso ilustrado é possível verificar que as avaliações se agrupam em posições equidistantes. Isso se deve ao fato de que cada dimensão foi dividida em 3 partes e dessa forma o parâmetro que determina a posição vertical da gaussiana, ao predominar sobre a avaliações, acaba por limitar a avaliação de certos domínios.

Aproximando o gráfico das avaliações para os domínios onde onde os parâmetros são mais bem avaliados, é possível visualizar mais precisamente a evolução d avaliação em razão dos dois últimos parâmetros de ajuste:

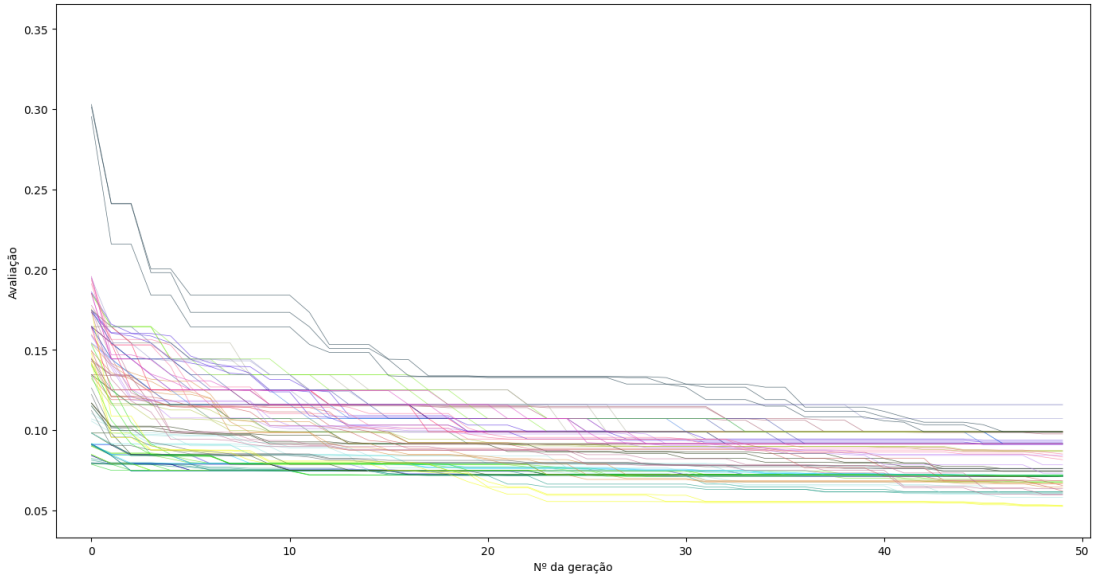

Figura 5.39: Aproximação para os melhores domínios de busca para o ajuste da função 3.1.

Com base nos testes realizados, é possível notar que a utilização deste método fornece uma convergência para as soluções de forma mais consistente do que se comparado com a execução do algoritmo em todo o domínio informado. É possível constatar também como o método de divisão e conquista é capaz de evitar com que o algoritmo permaneça estagnado em máximos locais, enquanto que mapeia mais precisamente os máximos locais e o global.

5.2.3 Utilização de threads e multiprocessamento

Dentre as vantagens apresentadas pelo método, está a possibilidade de se fazer uso de threads e multiprocessamento em uma única execução do algoritmo.

O método de divisão e conquista possibilita que um mesmo programa realize execuções do algoritmo genético para cada subdomínio em paralelo, enquanto analisa quais domínio são mais significativos para a função de avaliação.

Esse processo paraleliza o processamento da execução dos algoritmos em cada subdomínio, enquanto que analisa e seleciona os melhores subdomínios de busca para a soluções e então realiza novas subdivisões e execuções com base nos melhores subdomínios. Utilizando-se de computadores com múltiplos núcleos é possível otimizar ainda mais esta execução, dividindo a execução das threads por vários núcleos.

6 CONCLUSÕES FINAIS

Com base nos estudos e testes desenvolvidos neste trabalho, o maior desafio encontrado em buscar soluções otimizadas com algoritmo genético se concentra na determinação de um domínio de busca adequado.

Nos testes iniciais em que o algoritmo foi executado com um domínio de busca onde as soluções estavam bem definidas, o algoritmo conseguiu determinar os pontos de máximo com poucos indivíduos em poucas gerações. Ao realizar testes aplicados em problemas com amplos domínios de busca, o algoritmo apresentou constantes comportamentos de estagnação e que mesmo após muitas execuções não se mostrava eficiente.

Com base nesse problema enfrentado, foram desenvolvidos métodos que buscavam otimizar prioritariamente o domínio de busca.

O primeiro método buscou mover progressivamente o domínio de busca de cada variável do algoritmo dinamicamente com base nas melhores configurações dos indivíduos a cada conjunto de gerações. Com isso o algoritmo se mostrou capaz não só de reduzir o domínio de busca das soluções, como também de deslocar toda a população para domínios desconhecidos pelo algoritmo mas que se aparentavam promissores.

O método de convergência do domínio se mostrou altamente eficaz no que se refere a precisão no ajuste de dados. Como o algoritmo tende a reduzir o domínio de busca, os resultados se tornam gradativamente mais precisos e a população de indivíduos, que passa a explorar um domínio cada vez menor, faz um uso melhor do espaço de busca.

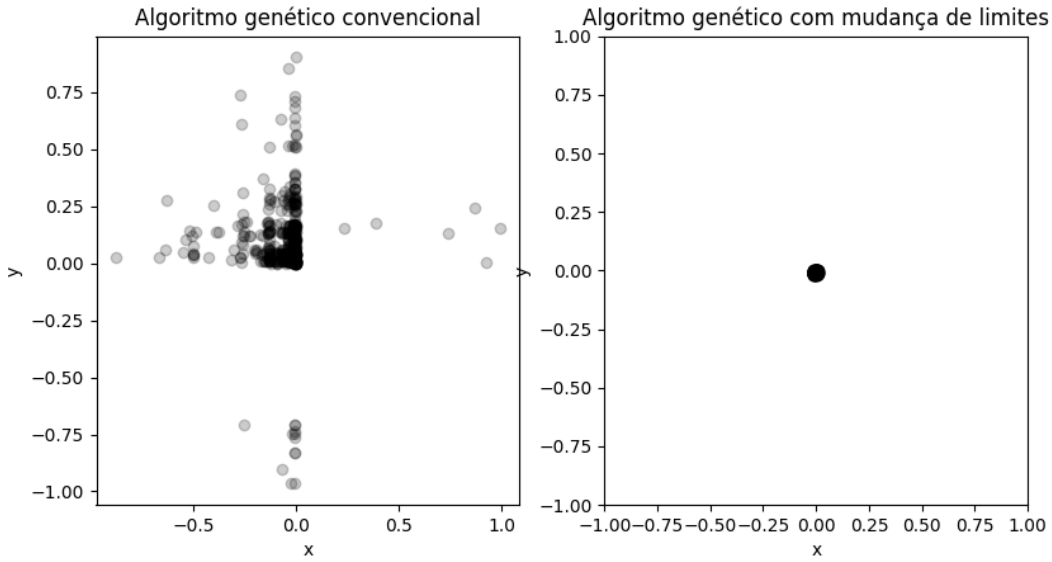

Figura 6.1: Comparação entre uma execução convencional do algoritmo e uma execução do algoritmo com o método de convergência do domínio aplicados para determinação do máximo de uma superfície gaussiana centrada na origem.

Em relação aos resultados obtidos nos testes que buscaram ajustar a curva da equação 5.3 a partir nos pontos da figura 5.5, ficou claro como o fator de precisão na busca é um aspecto fundamental no ajuste de curvas e dessa forma o método de convergência do domínio se mostrou altamente eficiente.

O método de divisão e conquista foi aplicado também com o intuito de otimizar a busca pelo domínio, mas com o foco de explorar grandes domínios de forma a determinar um o melhor espaço de busca para a execução do algoritmo, ao excluir os domínios onde as soluções não são bem avaliadas.

Este método procurou subdividir o domínio de busca em partes menores e então executar o algoritmo de maneira recursiva, limitando e subdividindo o domínio de busca a cada ciclo de execução.

Com base nos testes realizados, foi possível observar que o algoritmo foi capaz de mapear de maneira consistente os pontos de máximos locais e determinar o ponto de máximo global sem grandes dificuldades (fig. 5.36 e 5.34).

A implementação do método apresentado com base no diagrama da figura 5.31 não foi desenvolvida com o intuito se se obter maior precisão na busca, mas sim de excluir domínios que não favorecem a avaliação dos indivíduos, otimizando assim a busca pelas soluções em domínios que são realmente relevantes.

O método de divisão e conquista, no entanto, enfrenta dificuldades na otimização do domínio de problemas que envolvem muitas variáveis. Como o método subdivide cada parâmetro do algoritmo em n partes, gerando n^d subdomínios para execuções independentes (onde d representa o

número de dimensões do problema), em um problema que envolveria 10 parâmetros, o algoritmo, na configuração mais simples, subdividiria o domínio de busca em $2^{10} = 1024$ subdomínios. Ou seja, o algoritmo configuraria uma execução que seria impraticavelmente custosa em relação a memória, processamento e tempo de execução.

Com base nisso, o método de divisão e conquista seria mais bem aplicado quando submetido a problemas de otimização de amplos domínios e com poucas dimensões, como a busca de máximos de superfícies.

Por fim, com base em todas as implementações e testes realizados neste trabalho, os métodos desenvolvidos apresentaram resultados satisfatórios para os contextos que foram submetidos, e em aplicações reais ambos os métodos poderiam ser combinados com o algoritmo genético para uma busca mais otimizada em largos domínios.

REFERÊNCIAS BIBLIOGRÁFICAS

1 ALGORITMOS GENÉTICOS: PRINCÍPIOS E APLICAÇÕES. <http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf>. Acessado em: 30/04/2022.

2 LINDEN, R. *Algoritmos Genéticos*. [S.l.]: Ciência Moderna, 2012.

3 GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley Professional, 1989.

4 Computação Evolucionária. <http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/MQ-lista1-icademo.pdf>. Acessado em: 30/04/2022.