

Aplicação de Algoritmos Genéticos em otimização de leituras de antenas RFID

GUSTAVO SALIBI / HATOS ALBERT BARBOSA

Graduação

g174135@dac.unicamp.br / h261409@dac.unicamp.br

Resumo – Para as antenas RFID funcionarem de forma adequada, as configurações dos parâmetros dos módulos devem ser selecionadas de forma otimizada. Porém, isso é inviável de ser feito testando todas as possibilidades. Assim, utilizamos algoritmos genéticos buscando o resultado de configuração ideal. Diferentes parâmetros foram utilizados no algoritmo, produzindo diversas soluções. Os resultados obtidos solucionam de diferentes formas o problema e são, de modo geral, bem satisfatórios.

Palavras-chave – Inteligência Artificial, RFID, antenas, otimização, algoritmos genéticos, leitores, radio-frequência

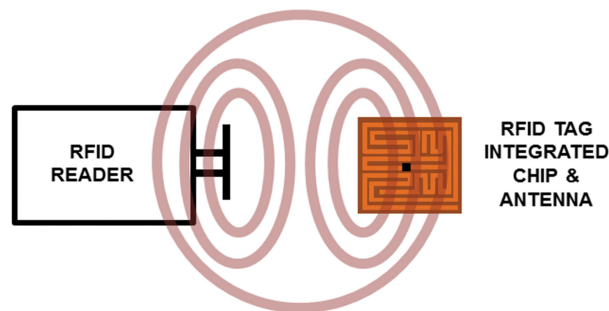


Figura 1. Acoplamento entre um leitor RFID e uma tag.

I. INTRODUÇÃO

A identificação por rádio-frequência, cujo primórdio histórico remete-se a meados de 1970, tem sido implementada no mercado de controle de acesso predial e residencial, cobranças de pedágios, controle de estoque e no ramo pecuário brasileiro. Está em vasta expansão nos últimos quinze anos, onde o acesso a tecnologia tornou-se facilitado pelo investimento de grandes indústrias do ramo no desenvolvimento de soluções específicas para as necessidades do país. A tecnologia RFID tem se provado eficiente pela sua rapidez, confiabilidade e abrangência de situações em que pode ser aplicada, pois elimina a verificação manual para identificação de objetos, sejam eles etiquetas em roupas, animais com etiquetas presas ao corpo para controle de vacinação e movimentação, carros que passam por portões (prédios ou pedágios), entre outras áreas.

A. A tecnologia RFID

O RFID é uma tecnologia wireless que utiliza a radio-frequência emitida por um interrogador, comumente chamado de *leitor*, que é recebida por um transpônder, comumente chamada de etiqueta (do inglês *tag*), sendo que este último, ao receber a rádio-frequência emitida pelo leitor, retornará - caso possível - os valores guardados em seus bancos de memória internos, sejam eles criptografados ou não.

Vale lembrar que, assim como todos os produtos emissores de ondas eletromagnéticas, existem regulamentações nacionais, continentais e mundiais que delimitam quais as faixas de operação autorizadas, a largura de banda, os canais, os padrões de mensagens e outros aspectos que são relevantes para a implementação de um sistema RFID. Porém, para o caso deste estudo, será tratado apenas o caráter mais básico da comunicação: a emissão das ondas de rádio do leitor (chamado de período de interrogação), o acoplamento da etiqueta ao receber este sinal e a resposta da etiqueta ao leitor, fornecendo seus dados.

B. Configuração de uma antena

Apesar da padronização da comunicação por rádio-frequência no mercado, ainda existem parâmetros que a antena de um leitor RFID pode alterar, de forma a conseguir comunicar-se com diferentes tipos de tags, ou obter melhores resultados de leitura, sejam estes definidos como melhor qualidade do sinal de resposta, maior confiabilidade de resposta ou velocidade de resposta.

C. RSSI

Em um sistema RFID, ao enviar um sinal de interrogação para uma tag próxima, que responde sincronicamente, o leitor analisa a resposta e reporta os dados da tag junto com o RSSI – Received Signal Strength Indicator (Indicador de Força do Sinal Recebido) –, ou seja, a medida de potência recebida de volta da tag. Quando um leitor reporta o valor do RSSI de uma tag, ele está efetivamente reportando o nível de potência com que

a tag respondeu seu sinal inicial. Essa potência é tipicamente reportada em decibéis por miliwatt, ou dBm. Decibéis é a unidade de comparação que relaciona duas propriedades físicas, em escala logarítmica. No caso do RFID, a propriedade medida é a mudança de potência com referência a um único miliwatt. Para referência ao leitor, o valor de RSSI pode variar entre -30 a -85 dbm. Pode-se, assim, perceber que os valores de RSSI são utilizados para se ter uma ideia de o quão bem uma tag está respondendo em sua zona de leitura. Se um leitor está bem configurado para a leitura de uma tag, o valor de RSSI pode melhorar consideravelmente, se comparado a uma leitura realizada para um leitor mal otimizado para aquele tipo de tag ou ambiente.

D. Otimização de leitura

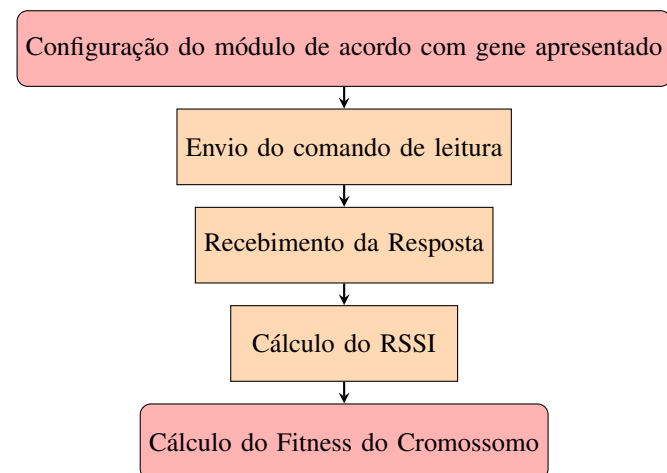
Como visto, existem casos em que mesmo uma tag estando posicionada no campo de leitura teórico (lóbulo) de uma antena, as configurações desta última podem não permitir que a tag seja encontrada pelo leitor RFID e, conseqüentemente, não haverá comunicação adequada. Como um exemplo de cenário, pode-se pensar no caso de um leitor posicionado em uma praça de pedágio.

II. MODELAGEM DO PROBLEMA

Visando a melhor comunicação possível entre leitor e tag, busca-se a configuração ideal dos parâmetros do leitor. Para este trabalho, foi utilizado um módulo RFID IM11, amplamente adotado pela indústria para UHF (Ultra High Frequencies), desenvolvido pela empresa Honeywell (anteriormente chamada Intermec). Nele, o alcance do campo de leitura da antena pode chegar a alguns metros. Esse módulo é muito utilizado em controle de estoque e praças de pedágio no Brasil.

Utilizando comandos BRI, é possível configurar os atributos da antena de acordo com a especificação que o cromossomo carrega em seus genes, em que cada gene representa uma configuração, e o alelo deste gene representa o valor a ser inserido na configuração para realizar a leitura e obter a resposta da tag junto do RSSI medido.

Assim, a seguinte sequência de ações ilustra o que deve ser implementado pelo algoritmo desenvolvido



A. Representação dos parâmetros

Segundo BRI [1], o módulo RFID IM11 possui os seguintes 31 parâmetros configuráveis:

- 1) ANTENNAS
- 2) ANTTIMEOUT
- 3) ANTTIMEOUT
- 4) ATA
- 5) BAUD
- 6) BTPWROFF
- 7) CHKSUM
- 8) DENSEREADERMODE
- 9) ECHO
- 10) EPCC1G2PARAMETERS
- 11) FIELDSTRENGTH
- 12) IDREPORT
- 13) IDTIMEOUT
- 14) IDTRIES
- 15) INITIALQ
- 16) INITTRIES
- 17) LBTCHANNEL
- 18) LBTSCANENABLE
- 19) LOCKTRIES
- 20) NOTAGRPT
- 21) RDTRIES
- 22) RPTTIMEOUT
- 23) SCHEDULEOPT
- 24) SELTRIES
- 25) SESSION
- 26) TAGTYPE
- 27) TIMEOUTMODE
- 28) TTY
- 29) UNSELTRIES
- 30) XONXOFF
- 31) WRTRIES

Visando uma maior clareza na implementação e estudo das soluções, esses parâmetros foram abstraídos em um vetor de inteiros com 31 posições. Cada posição assume um valor entre 0 e 100, representando o percentual ativado do parâmetro em questão.

O módulo é capaz de receber comandos de leitura através de comunicação TCP/IP ou serial, sendo que possui uma linguagem única de comunicação chamada de BRI (Basic Reader Interface). O guia das instruções de comandos desta linguagem pode ser encontrado anexo a este documento ou através da página oficial do produto no site da Honeywell [?].

B. Cromossomo e gene

Cada parâmetro – posição em um vetor – é representado como um gene. Cada vetor de parâmetros é representado como um cromossomo.

C. Representação do sinal recebido

Para simular o sinal recebido pelo leitor, foram utilizados 31 valores gerados aleatoriamente para representar cada um dos parâmetros que seriam considerados ideais. Não existe fórmula para se chegar a tais valores. Após isso, os valores alcançados

pelo algoritmo genético implementado são comparados com os valores ideais para produzir uma medida de fitness.

D. Avaliação dos resultados

Para se avaliar o quão boa é uma solução, ela é comparada com os valores ideais gerados anteriormente, produzindo um número entre 0 e 1, representando o percentual de semelhança entre a solução atual e a ideal.

III. TRABALHO PROPOSTO

Buscamos alcançar o resultado ótimo para cada um dos parâmetros disponíveis, de forma a obter a melhor resposta possível da antena simulada. Para isso, utilizamos algoritmos genéticos testando variações das seguintes características:

A. Tamanho da população

Para os testes, foram utilizadas populações de tamanho variando entre 30 e 1.500.

B. Critérios de parada

Consideramos o limite de 100.000 gerações como parada ou ao atingir o resultado ótimo, onde todos os parâmetros são ideais.

C. Técnicas de seleção

- **Elite Selection:** Seleciona os cromossomos com melhor valor de fitness.
- **Roulette Wheel Selection:** O primeiro passo é calcular fitness população através da soma do valor de todos os indivíduos. Depois disso, a probabilidade de seleção é calculada para cada indivíduo. Em seguida, um vetor é construído contendo as probabilidades cumulativas dos indivíduos. Portanto, n números aleatórios são gerados no intervalo de 0 a "soma total". Para cada número aleatório, é procurado um elemento do vetor que pode ter um valor mais alto. Portanto, os indivíduos são selecionados de acordo com suas probabilidades de seleção.
- **Tournament Selection:** Envolve a execução de vários "torneios" entre alguns indivíduos escolhidos aleatoriamente pela população. O vencedor de cada torneio (aquele com melhor fitness) é selecionado para crossover.

D. Técnicas de crossover

- **Uniform Crossover:** Usa uma taxa de mistura fixa entre dois cromossomos pais. Neste trabalho, a taxa é de 50%, ou seja, cada pai contribui com a mesma quantidade de genes.
- **Three Parent Crossover:** Usa três cromossomos pais escolhidos aleatoriamente. Cada gene do primeiro pai é verificado com o do segundo, e, caso sejam iguais, o gene é passado ao filho. Se diferentes, o gene do terceiro é passado.
- **One Point Crossover:** Um único gene de cada cromossomo pai é selecionado para ser mantido. Todos os outros são trocados entre eles.

E. Taxa de crossover

Para este trabalho, utilizamos taxas variando entre 25% e 75%.

F. Técnicas de mutação

- **Uniform Mutation:** Muda o valor de um gene escolhido aleatoriamente para um valor aleatório dentro dos limites válidos.
- **Reverse Sequence Mutation:** Tomamos uma sequência S limitada por duas posições i e j escolhidas aleatoriamente, de modo que $i \neq j$. A ordem dos genes dentro dessa sequência será invertida.
- **Twors Mutation:** Troca a posição de dois genes escolhidos aleatoriamente.

G. Taxa de mutação

Para este trabalho, utilizamos taxas variando entre 0% e 80%.

IV. MATERIAIS E MÉTODOS

Para alcançarmos o resultado, escrevemos um programa na linguagem C#, escolhida por ser conhecida pelos membros do grupo e por disponibilizar uma biblioteca de código livre que abrange de forma muito satisfatória os requisitos necessários para a execução do trabalho planejado. A biblioteca de Algoritmos Genéticos em questão é a GeneticSharp [2].

A. Como executar o programa

A forma mais eficiente de executar e visualizar os arquivos do programa é utilizando o Visual Studio Community, disponibilizado gratuitamente [3]. Para isso, basta abrir o arquivo "GeneticRFSetup.sln" e clicar no botão "executar" no canto superior esquerdo.

B. Utilizando o programa

O programa abrirá um menu de opções em modo texto, contendo os seguintes campos que devem ser respondidos:

- **Tamanho máximo da população**
- **Número de gerações (0 para sem limites)**
- **Técnica de seleção:**
 - 1 - Elite Selection
 - 2 - Roulette Wheel Selection
 - 3 - Tournament Selection
- **Técnica de crossover:**
 - 1 - Uniform Crossover
 - 2 - Three Parent Crossover
 - 3 - One Point Crossover
- **Probabilidade de crossover (entre 0 e 100)**
- **Técnica de mutação:**
 - 1 - Uniform Mutation
 - 2 - Reverse Sequence Mutation
 - 3 - Twors Mutation
- **Imprimir todos os genes dos melhores cromossomos:**
 - 1 - Sim
 - 2 - Não

- Imprimir todos os cromossomos com os genes:
 - 1 - Sim
 - 2 - Não

C. Saída do programa

Após selecionadas todas as opções, o programa executará e produzirá uma saída na tela contendo o resultado de acordo com o que foi configurado. Imprimindo sempre o melhor fitness encontrado em cada geração e também as opções marcadas como "sim" para impressão. Além disso, é imprimido o tempo de execução do programa.

D. Entendendo o código

Além dos arquivos correspondentes à biblioteca utilizada, se encontram no programa os seguintes arquivos:

- Program.cs: Arquivo principal, contendo a impressão do menu, a impressão das saídas e o fluxo de chamada das funções necessárias.
- RFFitness.cs: Calcula e retorna o fitness de um dado cromossomo.
- RFIDAntena.cs: Simula a antena. Para isso, produz automaticamente o valor ideal de cada parâmetro, gerando assim um cromossomo ideal, que é mantido para retornar um valor, através de comparação, de quão bom é o sinal RSSI recebido.
- RFSetupChromosome.cs: produz novos genes aleatoriamente para a criação de um novo cromossomo.

E. Configurações escolhidas para o trabalho

Para comparar diferentes resultados, testamos várias configurações iniciais e escolhemos a que obteve melhor resultado (alcançou o valor ótimo no menor tempo). Essa configuração serviu de base para compararmos os resultados com outras configurações feitas. Assim, um parâmetro foi variado por vez, permitindo a análise de seu impacto isolado no resultado final.

V. RESULTADOS E DISCUSSÃO

Os histogramas a seguir apresentam o número da geração no eixo x e o valor do fitness obtido até então no eixo y. Em todos os testes, o número máximo de gerações é 100.000, para o caso do resultado ideal não ser alcançado até então.

A. Teste 1 - Base

Como citado na seção anterior, a configuração inicial foi alcançada após várias tentativas de parâmetros iniciais. Ela serve como base para comparação com os outros testes, onde apenas um parâmetro é alterado por vez.

Atingiu o resultado ideal em 21s:38ms.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

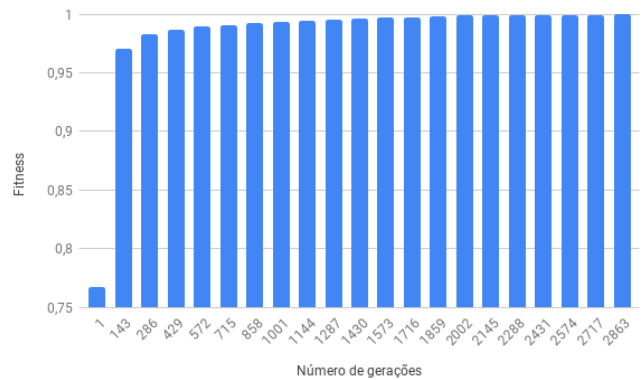


Figura 2. Teste 1.

B. Teste 2 - População = 1500

O número de gerações necessárias para se atingir o resultado ideal foi consideravelmente menor, entretanto, o tempo necessário para processar cada uma delas foi bem maior. Assim, ao final do programa, o tempo total de execução foi mais do que o dobro.

Atingiu o resultado ideal em 53s:181ms.

Configuração dos parâmetros:

- Tamanho máximo da população: 1500
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

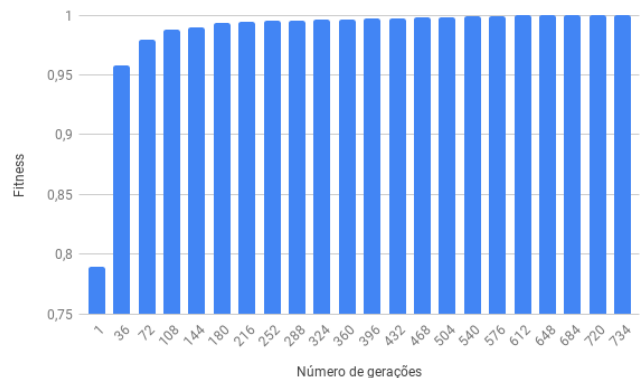


Figura 3. Teste 2.

C. Teste 3 - População = 30

Com uma população tão baixa, o valor de fitness convergiu dentro do limite de gerações antes de chegar no resultado ótimo, porém com um resultado satisfatório. Além disso, o valor de fitness até mesmo regrediu ao final do teste.

Não atingiu o resultado ideal dentro de 100.000 gerações.

Configuração dos parâmetros:

- Tamanho máximo da população: 30
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

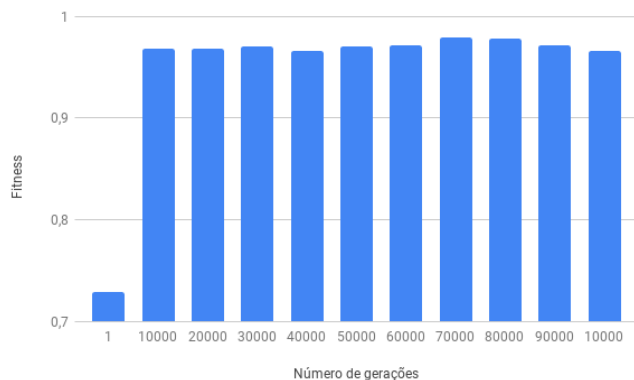


Figura 4. Teste 3.

D. Teste 4 - Técnica de seleção = Roulette

Essa técnica de seleção apresentou um resultado inferior para o nosso problema, fazendo com que o valor não convergisse para o ideal dentro do limite de gerações. Não atingiu o resultado ideal dentro de 100.000 gerações.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Roulette Wheel Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

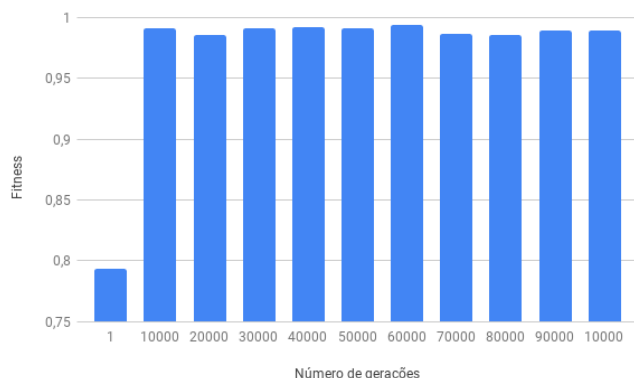


Figura 5. Teste 4.

E. Teste 5 - Técnica de crossover = Three Parent

O número de gerações necessárias para se atingir o resultado ideal foi ligeiramente menor, entretanto o tempo necessário para rodar o programa foi consideravelmente maior.

Atingiu o resultado ideal em 53s:181ms.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Three Parent Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

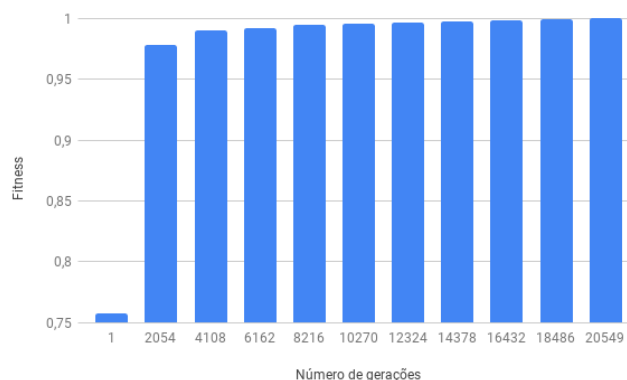


Figura 6. Teste 5.

F. Teste 6 - Probabilidade de crossover = 25

A menor taxa de cruzamento fez com que o número de gerações necessárias para se atingir o resultado ideal aumentasse em quase três vezes. Porém, o processamento necessário em cada geração foi consideravelmente menor, fazendo com que o tempo total fosse o menor entre todos os testes.

Atingiu o resultado ideal em 17s:227ms.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 25
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 10

G. Teste 7 - Técnica de mutação = Reverse Sequence

Essa não parece ser uma técnica muito adequada para o nosso problema, dado que ela troca posições dos genes e a ordem as posições da nossa solução é fixa. Assim, o desempenho foi prejudicado.

Não atingiu o resultado ideal dentro de 100.000 gerações.

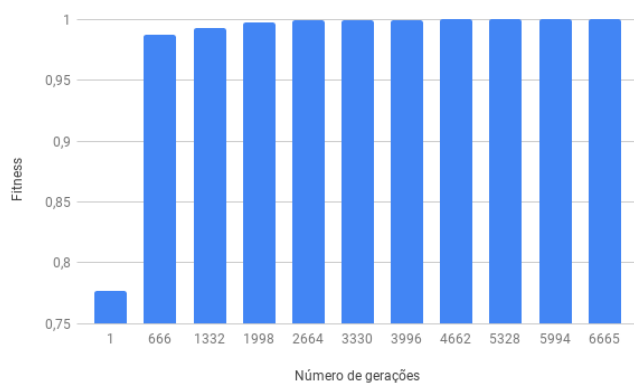


Figura 7. Teste 6.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Reverse Sequence Mutation
- Probabilidade de mutação: 10

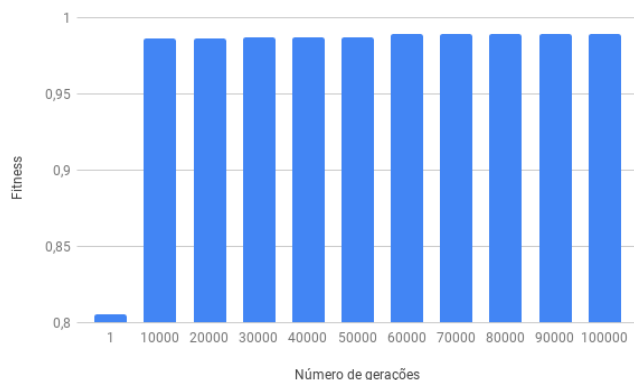


Figura 8. Teste 7.

H. Teste 8 - Probabilidade de mutação = 80%

Com uma taxa de mutação tão alta, houve, depois de certo ponto, uma parada na geração de indivíduos mais aptos que os anteriores, devido a características positivas não serem tão preservadas. Com isso, o valor se estagnou bem abaixo dos outros testes.

Não atingiu o resultado ideal dentro de 100.000 gerações.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation

- Probabilidade de mutação: 80

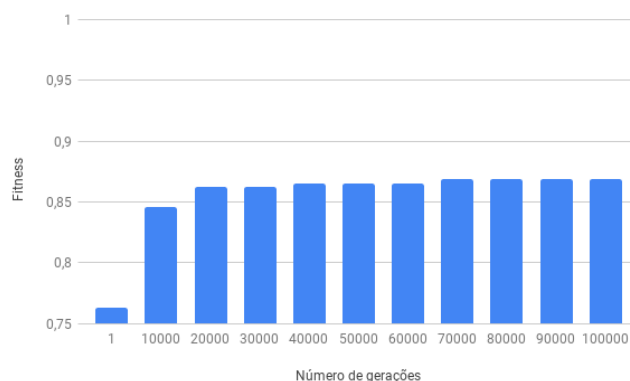


Figura 9. Teste 8.

I. Teste 9 - Probabilidade de mutação = 0%

Com uma taxa de mutação zerada, depois de um certo tempo, a população se tornou homogênea. Isso fez com que os crossovers não surtiram efeito de melhora no fitness, já que os cromossomos deixaram de possuir variabilidade genética. Não atingiu o resultado ideal dentro de 100.000 gerações.

Configuração dos parâmetros:

- Tamanho máximo da população: 150
- Técnica de seleção: Elite Selection
- Técnica de crossover: Uniform Crossover
- Probabilidade de crossover: 75
- Técnica de mutação: Uniform Mutation
- Probabilidade de mutação: 0

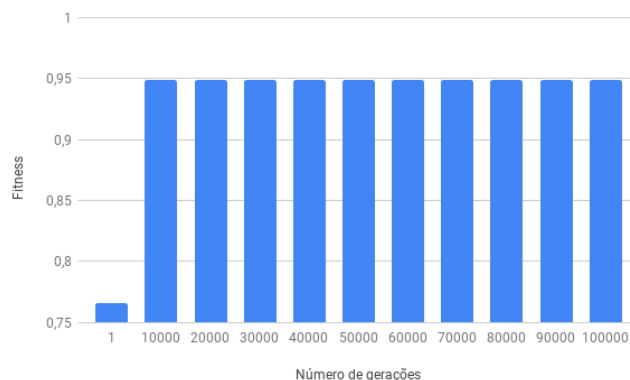


Figura 10. Teste 9.

VI. CONCLUSÕES

Dada a natureza do problema, cujo resultado ideal é impossível de ser previsto, seria necessário, na busca por força bruta, um algoritmo com ordem de complexidade $O(101^n)$, com $n = 31$, nesse caso. Isso seria absolutamente inviável.

de ser rodado. Utilizando algoritmos genéticos, nós conseguimos o resultado ideal em apenas 734 gerações, em um dos testes. Caso utilizássemos o algoritmo de força bruta, seriam necessários, no pior caso, 10^{62} passagens.

Como podemos verificar facilmente nos resultados obtidos, todos os testes, mesmo os piores, produziram resultados bem satisfatórios, chegando ao resultado ideal ou bem próximo disso.

Como uma das limitações de nossas análises, está o tempo de execução computado em um notebook de uso pessoal, sem um ambiente preparado para evitar influências de outros processos no tempo total.

Além disso, o normalizarmos os valores de cada parâmetro entre 0 e 1, com 101 possibilidades em cada parâmetro, adicionamos maior necessidade de processamento, já que os parâmetros reais da antena teriam intervalos menores. Para os nossos testes locais, isso não foi problema, entretanto, caso testado em uma antena real, o tempo de resposta da antena poderia deixar o algoritmo muito mais lento se a modelagem não fosse adaptada.

REFERÊNCIAS

- [1] I. T. Corporation, *Basic Reader Interface*, 3rd ed., Intermec, Intermec Technologies Corporation, dec 2013. 2
- [2] <https://github.com/giacomelli/GeneticSharp>. 3
- [3] <https://visualstudio.microsoft.com/pt-br/vs/community>. 3