

Algoritmos Genéticos e o Problema do Caixeiro Viajante

Gabriel Alencastro

Orientador: Mateus R. Leal

Agradecimentos

- Agradeço especialmente a dedicação do meu orientador em me ajudar a estruturar o trabalho e tirar todas as minhas dúvidas.
- A minha tutora Mayra por promover a experiência de realizar essa pesquisa e por me apoiar a todo momento.
- A professora de português Juliana por me ajudar a entender a estrutura e requisitos do trabalho de conclusão.
- E, por fim, aos meus amigos e familiares que me apoiaram ao longo dessa caminhada.

1 Resumo

Neste trabalho apresento uma introdução a conceitos de algoritmos e métodos heurísticos para resolver problemas com grande dificuldade de processamento como o problema da mochila binária. O foco do trabalho são os algoritmos genéticos, que se baseiam no conceito de seleção natural de C. Darwin para resolver problemas de busca e combinação. Ao decorrer do trabalho falarei sobre os processos que compõe o algoritmo genético como crossover e mutação. Por fim falarei sobre o processo de desenvolvimento de um programa de computador que resolve o problema do caixeiro viajante utilizando o método do algoritmo genético.

Sumário

1	Resumo	2
2	Introdução	5
3	Conceitos Básicos	6
3.1	Seleção Natural	6
3.2	Algoritmos	7
3.3	Heurística	8
4	O que é Algoritmo Genético?	9
4.1	Origem	10
5	Inicialização da População	11
5.1	Representação do Cromossomo	11
6	Avaliação e Seleção	12
6.1	Avaliação	12
6.2	Seleção	12
6.3	Métodos de Seleção	13
6.4	Elitismo	14
7	Operadores Genéticos	15
7.1	Crossover	15
7.2	Mutação	17
7.3	Finalização de um Ciclo e Critério de Parada	17
8	Problema do Caixeiro Viajante	19
8.1	Complexidade	19
8.2	Minha experiência prática	20
9	Conclusão	24

Lista de Figuras

1	Fluxograma algoritmo genético.	9
2	Exemplo Autômato celular.	10
3	Esquema de um cromossomo algoritmo genético.	11
4	Cruzamento de um ponto.	16
5	Cruzamento multiponto.	16
6	Cruzamento uniforme.	16
7	Tabuleiro do Jogo Icosiano.	19
8	Tabela de complexidade tsp.	20
9	Rota mapa do Brasil.	22
10	Diferentes distâncias.	23

2 Introdução

Neste trabalho, busco apresentar a estrutura básica dos algoritmos genéticos - suas formas, parâmetros e processos - e como cada um desses se inspira em conceitos da biologia para resolver diversos problemas. Tento trazer a beleza da natureza a partir de um ponto de vista tecnológico e como podemos achar soluções para problemas reais olhando para processos naturais. Busco também, por meio desse trabalho, introduzir como se dá o funcionamento dos computadores, ferramentas que estão presente a todo momento em nossas vidas.

Um dos meus principais objetivos com esse trabalho foi estruturar e desenvolver um programa de computador do início ao fim com algum grau de utilidade prática. Junto a essa vontade, sempre tive uma paixão enorme por programação, principalmente na parte de algoritmos e resolução de problemas por meios criativos, daí surgiu minha vontade de trabalhar com algoritmos genéticos.

Iniciarei o trabalho apresentando alguns conceitos básicos para a compreensão dos algoritmos genéticos, para, no capítulo seguinte, falar um pouco sobre o que eles são e sua origem. Em seguida, falarei um pouco sobre o problema do caixeiro viajante, um clássico problema facilmente resolvido por meio de um algoritmo genético. Por fim, nos capítulos seguintes, falarei um pouco sobre os processos que compõe os algoritmos genéticos e finalizarei o trabalho falando sobre a minha experiência prática.

3 Conceitos Básicos

Neste capítulo irei explicar alguns conceitos básicos para a compreensão do funcionamento dos algoritmos genéticos. Entre eles o conceito de seleção natural, que é a base dos algoritmos genéticos. Outro assunto tratado nesse capítulo é o próprio conceito de algoritmo, um termo muito utilizado nos tempos atuais e que nem todos conhecem seu real significado. Por fim falarei sobre heurística, uma classe de métodos no qual os algoritmos genéticos se encaixam.

3.1 Seleção Natural

Podemos definir, de forma simples, o termo seleção natural como o mecanismo evolutivo no qual sobrevivem e se reproduzem aqueles indivíduos que melhor se adaptam ao ambiente em que vivem. Ou seja, aqueles indivíduos de uma espécie que conseguirem se alimentar, fugir de presas, se abrigar, achar um parceiro para a reprodução e assim por diante, serão aqueles que passarão sua informação genética para frente. Esse conceito foi proposto por Charles R. Darwin em seu *On the Origin of Species* [1] onde defendeu a ideia de que seres vivos se adaptam ao ambiente em que vivem ao longo de gerações dando origem a novas espécies. Podemos também analisar a seleção natural a partir do seguinte argumento :

1. **SE** há organismos que se reproduzem e...
2. **SE** os descendentes herdam as características de seus progenitores e...
3. **SE** há variação nas características e...
4. **SE** o ambiente não suporta todos os membros de uma população em crescimento,
5. **ENTÃO** aqueles membros da população com características menos adaptadas (de acordo com o ambiente) morrerão e...
6. **ENTÃO** aqueles membros com características mais adaptadas (de acordo com o ambiente) prosperarão.

3.2 Algoritmos

Um algoritmo pode ser definido basicamente como uma sequência de instruções (passos) para alcançar um determinado resultado ou completar uma tarefa específica. O desenvolvimento de um algoritmo depende do comprimento de algumas regras básicas. A primeira é que as instruções devem ser diretas e não podem apresentar ambiguidade, ou seja, devem ser claras, sem brechas para interpretação. A segunda regra diz que os passos do algoritmo devem estar organizados em uma sequência, dispostos cada passo de uma vez. A terceira e última regra que define um algoritmo é que a quantidade de passos a serem seguidos deve ser finita, ou seja, deve haver um passo final.

Um exemplo bem simples que pode ser usado para entendermos o que é um algoritmo é o cálculo da área de um triângulo, que se define basicamente em: medir a base (b) do triângulo, medir sua altura (h) e por fim multiplicar a base pela altura e dividir por dois: $b \times h/2$. Não necessariamente algoritmos tem relação com a matemática. Eles estão presentes em qualquer tarefa que realizamos no nosso dia a dia. Como por exemplo ao pegar um ônibus. Observe a baixo a sequência de passos que representa o algoritmo para pegar um ônibus:

1. Chegar no ponto.
2. Enquanto o ônibus não passa: espere.
3. Quando o ônibus chegar: suba no ônibus.
4. Pegue o dinheiro e pague o cobrador.
5. Passe pela roleta.
6. Enquanto não chegar ao seu ponto de destino: espere.
7. Quando chegar seu ponto de destino: desça do ônibus.

3.3 Heurística

O termo “heurística” se origina no grego e quer dizer “eu encontro” ou “eu acho”. Na área da computação um algoritmo se classifica como heurístico quando ele utiliza de técnicas não exatas, que não trarão - necessariamente, a melhor solução para o problema em questão. Algoritmos heurísticos abrem mão da exatidão e precisão dos métodos clássicos para obter mais velocidade em sua execução. Um bom exemplo para compreender a utilidade dos métodos heurísticos é o problema da mochila binária. Neste clássico problema de otimização é dado um conjunto de itens, cada um com seu respectivo valor de utilidade e peso, e com eles se deseja obter o maior valor agregado de utilidade sem que ultrapasse um valor x de peso. Podemos observar que a quantidade de itens determinará a dificuldade de resolução do problema. Por exemplo, se tivermos apenas 3 itens não haverá dificuldade para resolver o problema pois existem apenas 8 possibilidades de composição para a mochila. Já se analisarmos um caso onde o número de itens é igual a 30 teremos mais de um milhão de possíveis composições, o que torna a resolução por força bruta (testar todas as possibilidades) inviável. É aí que surgem os métodos heurísticos, que tentarão, de diversas formas, achar um caminho que não necessariamente analise todas as possibilidades, chegando a soluções boas o suficiente para o uso prático. Métodos heurísticos utilizam de uma estrutura de iteração, ou seja, repetem um processo diversas vezes para tentar melhorar a solução proposta. A cada iteração é esperada uma solução mais próxima do máximo global (melhor solução possível), no entanto muitos algoritmos chegarão em soluções chamadas máximos locais (soluções boas mas não necessariamente a melhor possível). No caso do algoritmo genético, a iteração é chamada de geração, e tem o mesmo princípio das gerações de indivíduos na vida real, cada geração será gerada de acordo com a anterior e assim por diante.

4 O que é Algoritmo Genético?

Algoritmo genético é um modelo computacional heurístico inspirado na teoria da seleção natural de Charles Darwin que se propõe a resolver problemas de otimização, busca, entre outros. No algoritmo genético um conjunto (população) de possíveis soluções para o problema é gerado aleatoriamente. Cada item (indivíduo) desse conjunto será representado por uma sequência de valores (cromossomo). Em seguida cada indivíduo da população passa por uma avaliação que classifica o quão bem-adaptado ao problema ele está. O próximo passo é selecionar alguns indivíduos de acordo com a avaliação feita. A partir dos indivíduos selecionados são gerados novos indivíduos por meio do cruzamento dos cromossomos. E, por fim, cada novo indivíduo passa por um processo de mutação, que altera levemente o seu material genético. Assim é gerada uma nova população que passa por todos os processos anteriores até satisfazer o critério de parada do algoritmo, como mostra a figura 1. Chegando a uma solução viável para o problema em questão.



Figura 1: Fluxograma algoritmo genético.

4.1 Origem

O algoritmo genético foi desenvolvido pelo cientista americano John Henry Holland e seus alunos da universidade de Michigan através do estudo de autômatos celulares (figura 2) - algoritmo que representa a evolução de uma colônia de células em um grid ao longo de gerações. Holland desejava descobrir o desenvolvimento de uma colônia ao passar das gerações e assim surgiram as primeiras ideias do que viria a ser o algoritmo genético. Holland publicou em 1975 seu livro *Adaptation in Natural and Artificial Systems* [2] onde trazia pela primeira vez os conceitos do algoritmo genético. No entanto a invenção apenas foi ganhar popularidade com a publicação do livro *Genetic algorithms in search, optimization and machine learning* [3] de David E. Goldberg, um dos alunos de Holland. Nesse livro Goldberg trouxe algumas aplicações práticas para o algoritmo genético e como utiliza-lo em problemas de busca e otimização sendo até hoje um dos livros mais importantes da área de machine learning (aprendizado de máquina).

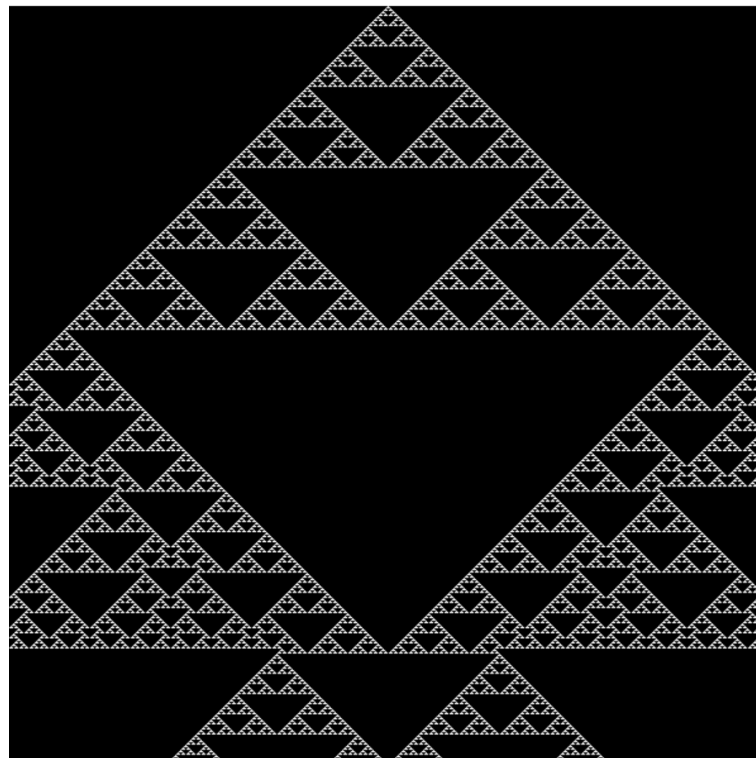


Figura 2: Exemplo de Autômato celular.

5 Inicialização da População

O primeiro passo para a realização de um algoritmo genético é a inicialização da população. Essa etapa é feita com a geração de uma lista de cromossomos, com valores aleatórios, que representam possíveis soluções do problema a ser resolvido. Essa lista pode variar de tamanho de acordo com a necessidade do problema. Problemas com um espaço amostral pequeno necessitam, conseqüentemente, de uma população menor.

5.1 Representação do Cromossomo

O cromossomo no algoritmo genético, assim como na biologia, é responsável por carregar as informações genéticas de um indivíduo. Geralmente o cromossomo será representado por uma sequência de bits ou por um vetor de números reais (dependendo do problema a ser resolvido). As informações nele contidas serão os parâmetros específicos do problema que influenciarão diretamente no desempenho daquela possível solução. Vejamos uma estrutura de cromossomo para o problema da mochila binária:

No problema da mochila binária temos uma lista de n itens que podem ou não ser levados para uma viagem, cada item terá dois valores agregados a ele: O primeiro valor será o seu peso em quilogramas; Já o segundo será o seu valor de utilidade. O cromossomo deverá conter a informação de quais itens estarão presentes na mochila naquela possível solução. Sendo assim temos que achar alguma forma de representar, com um tamanho fixo, essa informação. A estrutura mais comum para esse tipo de problema é criar um vetor binário, onde cada bit (gene) estará ligado a um item; o valor 1 representa a presença do item na possível solução, já o valor 0 dirá que o item não compõe a solução. Essa estrutura facilitará os próximos processos como avaliação e operações genéticas. A imagem a seguir demonstra a representação do cromossomo no problema da mochila binária:



Figura 3: Esquema de um cromossomo algoritmo genético.

6 Avaliação e Seleção

O primeiro passo após a estruturação do problema e a inicialização da população é colocar os indivíduos à prova, simulando a adaptação dos mesmos ao ambiente, ou seja, testar o quão bom eles são para o problema a ser resolvido e decidir quais são aptos a se reproduzir e passar seus genes para as próximas gerações. Neste capítulo falarei sobre como são feitos esses dois processos no algoritmo genético.

6.1 Avaliação

A avaliação é a etapa responsável por classificar os indivíduos de acordo com a sua eficiência em resolver o problema em questão. Neste processo devemos atribuir um valor objetivo para quantificar a adaptação do indivíduo. Utiliza-se uma função chamada fitness function (função objetivo) que tem como entrada o cromossomo do indivíduo e retorna um valor real, que pode ser interpretado como índice de adaptação daquela solução. Quanto maior o retorno da função, melhor adaptado o indivíduo está. A função objetivo é para o algoritmo genético o que o ambiente é para os seres vivos. Então se pensarmos em um exemplo real, como o desenvolvimento de pescoços mais longos das girafas para alcançar folhas mais altas, a função objetivo classificaria as girafas de acordo com a distância entre a altura de suas bocas e a altura média das folhas, recompensando assim girafas que nascem com pescoços de tamanho próximo à altura das folhas e penalizando aquelas que tem um pescoço curto ou longo demais. Escolher corretamente a função objetivo é essencial para o funcionamento do algoritmo genético. Com uma função objetivo que não representa a otimização do problema, o algoritmo genético não atinge soluções desejadas. Se voltarmos ao exemplo das girafas e considerarmos uma função objetivo que apenas recompensa o indivíduo com maior pescoço, sem considerar que pescoços muito longos ultrapassam a altura das folhas, obteremos indivíduos com pescoços longos demais sendo ineficientes para a alimentação.

6.2 Seleção

A seleção é o processo responsável por escolher indivíduos aptos para a reprodução e é essencial para que uma população se torne mais adaptada ao problema ao longo das gerações. A ideia principal por trás da seleção é escolher os indivíduos que melhor se adaptaram ao problema dentro daquela população para depois fazer o cruzamento do seu material genético

(através do processo de crossover - trataremos desse conceito no próximo capítulo). Na natureza, a seleção dos indivíduos é feita de diversas maneiras. Numa primeira abordagem, podemos perceber que indivíduos que se adaptam muito mal ao ambiente não serão capazes sequer de sobreviver. Se voltarmos ao caso das girafas, percebe-se que indivíduos com pescoço muito curto não são capazes de se alimentar, por isso não sobreviverão. Já em um segundo momento podemos observar que girafas com pescoço curto e que, conseqüentemente, têm uma alimentação restrita conseguirão sobreviver mas, no entanto, terão dificuldade para achar um par para se reproduzir e passar seus genes para a próxima geração. No algoritmo genético tentamos reproduzir tais comportamentos, e assim como na natureza, existe mais de uma abordagem para o critério de seleção.

6.3 Métodos de Seleção

No algoritmo genético existem diversas maneiras de fazer a seleção dos indivíduos. Citarei a seguir algumas das mais utilizadas:

Seleção por Rank (rank selection): neste método de seleção os indivíduos são ordenados de acordo com sua pontuação e em seguida é atribuída a cada um uma probabilidade de ser selecionado correspondente ao seu rank.

Seleção por Roleta (roulette wheel selection): neste método cada indivíduo assume uma fatia de uma roleta correspondente a sua porção do fitness acumulado da população, ou seja, a soma de cada fatia será correspondente a soma do fitness de todos os indivíduos da população somados. E por fim a roleta é girada e o indivíduo correspondente a fatia selecionada é escolhido.

Seleção por Torneio (tournament selection): na seleção por torneio grupos aleatórios são montados e o indivíduo mais apto de cada grupo é selecionado.

Seleção Uniforme (uniform selection): neste método a probabilidade de um indivíduo ser escolhido está relacionada apenas ao número de indivíduos na população, ou seja, a chance de todos os indivíduos serem selecionados é igual

6.4 Elitismo

Quando estamos utilizando um algoritmo genético para resolver algum problema é comum que, ao longo das gerações, se perca boas soluções, pois os indivíduos sempre são substituídos por novos indivíduos que não necessariamente terão um fitness igual ou maior a seus descendentes. Sendo assim, no intuito de não perder informações "genéticas" importantes é utilizado um método chamado elitismo que consiste, basicamente, em preservar os melhores indivíduos da população assim preservando as melhores soluções, trazendo estabilidade para o algoritmo e garantindo que ele não irá regredir. A quantidade de indivíduos que serão preservados é dada por uma variável chamada taxa de elitismo que varia de 0 a 1 e que será multiplicada pela quantidade total de indivíduos obtendo o número de soluções preservados. é importante se atentar ao fato de que uma taxa de elitismo muito alta ocasionará na estagnação da população pois quanto maior a taxa, menor será o número de novas soluções sendo geradas. Uma boa taxa de elitismo varia entre 0.05 e 0.2.

7 Operadores Genéticos

Após a seleção dos indivíduos aptos, resta apenas realizar o processo da geração de novas possíveis soluções. Com elas formaremos uma nova população, acabando assim um ciclo do algoritmo. Nestes dois processos (crossover e mutação) são aplicados diferentes métodos, que se inspiram em conceitos da genética, para gerar novos indivíduos (por isso “operadores genéticos”). Os diferentes métodos (tanto de crossover quanto de mutação) descritos a seguir foram retirados do artigo *Algoritmos Genéticos: uma Introdução* [5].

7.1 Crossover

Crossover é o processo responsável por fazer a recombinação do material genético entre dois indivíduos para a criação de descendentes. O termo crossover vem da biologia e é etapa da meiose (divisão celular) na qual ocorre a troca de material genético entre dois cromossomos homólogos. Os cromossomos são partidos em alguns pedaços e são efetuadas algumas trocas de partes equivalentes gerando novas combinações de genes. Para a biologia esse processo garante uma variedade genética, pois com apenas um par de cromossomos são possíveis várias combinações. No algoritmo genético o processo é bastante semelhante. Considerando que o cromossomo é um vetor de somente uma dimensão, devemos apenas decidir de qual pai o gene no índice i será atribuído ao filho. Para isso podemos utilizar uma máscara binária de tamanho igual ao do cromossomo, onde 1 dirá que o pai 1 cederá seu gene ao filho e 0 que o gene cedido será do pai 2. Resta apenas decidir como será a distribuição de zeros e uns da máscara, e para isso existem diversos métodos. Citarei alguns a seguir:

Cruzamento de um ponto: neste método a máscara é feita escolhendo aleatoriamente um valor i compreendido entre 0 e o tamanho q do cromossomo dos pais. Todos os itens com índice menor que i assumirão o valor 1, já aqueles que estão compreendidos entre i e q assumirão o valor 0, como mostra a imagem a baixo.

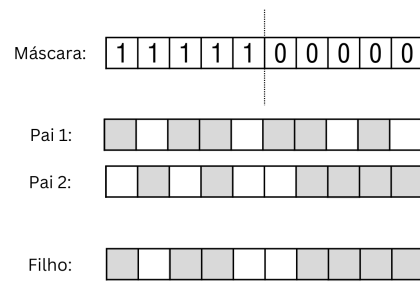


Figura 4: Cruzamento de um ponto.

Cruzamento multiponto: o cruzamento multiponto se assemelha bastante ao cruzamento de um ponto a única diferença é que ao invés de escolhermos apenas um valor aleatório i , escolhemos n valores, e assim como no cruzamento de um ponto, os pedaços entre um valor i e outro alternarão entre 0 e 1 na formação da máscara. A imagem a seguir exemplifica esse método.

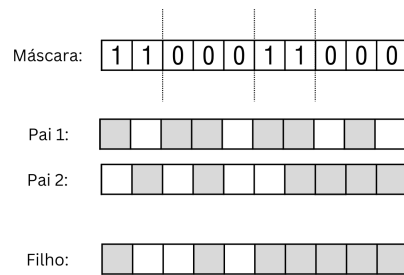


Figura 5: Cruzamento multiponto.

Cruzamento uniforme: no cruzamento uniforme, para cada posição i da máscara é escolhido aleatoriamente o valor a ser utilizado (0 ou 1). A seguinte imagem exemplifica o método de cruzamento uniforme.

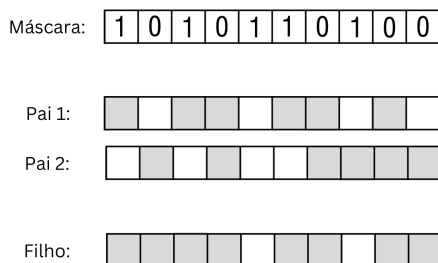


Figura 6: Cruzamento uniforme.

7.2 Mutação

Na biologia mutação diz respeito às mudanças causadas ao material genético de um indivíduo. É a mutação que garante boa parte da variabilidade genética e a adaptação de uma espécie ao ambiente, por meio do surgimento de novas características. Sem a existência da mutação, os seres vivos teriam muita dificuldade em evoluir pois ficariam presos às mesmas estruturas genéticas, dependendo apenas do DNA de seus ancestrais. No algoritmo genético, a mutação emprega similar papel. É graças a ela que o algoritmo sai de máximos locais (soluções que parecem boas, mas que no entanto não são boas o suficiente). Outro papel da mutação é acelerar o processo de evolução, visto que graças a ela surgem soluções levemente diferentes às demais, fazendo com que sejam abordadas diversas alternativas de soluções. O processo de mutação no algoritmo genético é feito após a geração de um novo indivíduo. Dada uma probabilidade x (taxa de mutação) é decidido se o indivíduo em questão passará ou não pela mutação (no caso afirmativo o indivíduo se expõe ao processo). Existe mais de uma maneira de realizar a mutação. Listarei 3 formas comumente usadas.

Mutação flip: na mutação flip cada gene a ser mutado recebe um novo valor aleatório.

Mutação swap: na mutação swap são selecionados pares de genes que terão sua posição trocada entre si.

Mutação creep: na mutação creep um valor é adicionado ao gene em questão.

7.3 Finalização de um Ciclo e Critério de Parada

Após serem gerados os indivíduos que farão parte da nova geração a partir dos métodos neste capítulo descritos, um novo ciclo começa no algoritmo. Assim como na natureza, uma nova geração foi criada e essa agora será colocada a prova, reiniciando o ciclo. Podemos lembrar que no segundo capítulo, definimos algoritmo como uma sequência finita de instruções e que, no entanto, estamos considerando que os algoritmos genéticos tem em sua estrutura um ciclo infinito. É considerando essa questão que se definem critérios para a parada do algoritmo. Abordarei três diferentes maneiras de definir o fim de um algoritmo genético. O primeiro e mais simples é determinando um número n de gerações, e quando esse número for atingindo o algoritmo é interrompido. O segundo critério é determinar uma meta de fitness a ser atingida pela população, e assim que algum indivíduo alcançar esse valor o algoritmo

para. Por fim, o terceiro critério se baseia em analisar a taxa de variação da média de fitness da população, quando a variação for zero ou próxima de zero por um determinado período, o algoritmo chega ao fim. Após chegarmos ao fim do algoritmo, resta ordenar os indivíduos em ordem decrescente de acordo com o valor de fitness e selecionar o primeiro. O valor do cromossomo desse indivíduo será a melhor solução encontrada pelo algoritmo para o seu problema.

8 Problema do Caixeiro Viajante

O problema do caixeiro viajante ou travelling salesman problem (tsp) é um clássico problema de otimização combinatória que surgiu no século XIX, desenvolvido pelo matemático irlandês Sir William Rowan Hamilton e o matemático britânico Thomas Penyngton Kirkman. O problema surgiu basicamente a partir do jogo "Icosiano" - um puzzle desenvolvido por Hamilton que consistia, basicamente, em encontrar um ciclo hamiltoniano (sequência de pontos ligados, sendo que cada ponto só apareça apenas uma vez no ciclo e que esse ciclo termine e comece no mesmo ponto).



Figura 7: Tabuleiro do Jogo Icosiano.
Figura retirada da referência [4].

O problema do caixeiro viajante pode ser definido da seguinte maneira: dada uma lista de n cidades e as respectivas distâncias entre cada par de cidade, qual rota terá a menor distância total considerando um percurso que saia de uma cidade x e passe por todas as outras n cidades apenas uma vez, voltando a cidade de origem? A imagem a seguir ilustra o problema do caixeiro viajante.

8.1 Complexidade

Para entendermos a complexidade do problema do caixeiro viajante podemos pensar na maneira mais primitiva de resolvê-lo: analisando todas as rotas possíveis e comparando-as

umas às outras para encontrar qual resulta em uma distância total menor. Assim encontraremos com certeza a melhor solução possível. Dessa maneira podemos ver que a quantidade de soluções a serem analisadas apenas cresce de acordo com o número de cidades. Para entendermos quanto a quantidade de rotas possíveis cresce de acordo com o número de cidades devemos pensar como as rotas são montadas. Considerando um trajeto de 4 cidades podemos perceber que, ao montar uma rota, a primeira e a última cidade sempre serão fixas, pois o trajeto começa e termina na mesma cidade; já na segunda posição haverão apenas três possibilidades, visto que a primeira cidade não pode se repetir; por fim, na terceira posição, se seguirmos o mesmo raciocínio, restarão apenas duas possibilidades de cidades. Considerando também que toda a rota terá uma rota equivalente, porém invertida podemos, por fim, chegar ao seguinte número de possíveis rotas: $(1 \times 3 \times 2 \times 1)/2 = 3$. Se considerarmos n cidades podemos concluir que o total de rotas possíveis é igual a $(n - 1)!/2$. A tabela a seguir mostra o tempo que um computador que realiza 1 bilhão de operações por segunda leva para calcular todas as rotas de um problema com n cidades.

n	$(n-1)!/2$	tempo
5	12	insignificante
10	181 mil	0.002 sec
15	43 bilhões	10 min
20	6×10^{16}	2 mil anos
25	3.1×10^{23}	14 bilhões de anos
30	4.4×10^{30}	2.5×10^{18} anos

Figura 8: Tabela de complexidade problema do caixeiro viajante.

Podemos perceber que o crescimento do custo de processamento torna inviável o cálculo de todas as rotas aproximadamente a partir de $n = 20$. Ou seja, não podemos depender de métodos que analisem todas as possíveis rotas. Precisamos de um método não determinístico que consiga achar uma solução viável sem necessariamente analisar todas as possibilidades.

8.2 Minha experiência prática

Na tentativa de colocar em prática o conhecimento adquirido no desenvolvimento desse trabalho, busquei encontrar algum problema real que pudesse ser resolvido com algoritmos genéticos. De início considerei algumas opções. Entre elas o desenvolvimento de uma tabela

de horários para as aulas do ensino médio da minha escola, no entanto, quando fui colocar em prática o projeto percebi que o problema era muito complexo para o meu nível de conhecimento e acabei decidindo tentar outra coisa. Nesse momento, logo após desistir da tabela de horários, eu já havia desenvolvido alguns programas de computador para resolver problemas clássicos como o da mochila binária e o do caixeiro viajante. Sendo assim decidi escolher um deles, no caso o do caixeiro viajante, para resolver algum problema prático. De início me veio a ideia de achar uma rota para a van escolar, que cada dia - de acordo com os alunos que iriam para a aula - poderia tomar uma rota mais vantajosa. Porém, após conversar com o responsável pelo transporte percebi que não seria muito útil pois ele já sabia quais mudanças fazer na rota quando algum aluno faltasse. Novamente voltei a pensar em algum problema para resolver. Certo dia eu estava consumindo algum conteúdo sobre algoritmo genético na internet e me deparei com um mapa dos Estados Unidos com uma rota trassada e logo me veio a ideia de fazer um algoritmo genético que calculasse a melhor rota para passar em todas as capitais do Brasil. Esse problema não era o mais prático possível, porém era algo simples de realizar, usava elementos da vida real e era um exemplo bastante ilustrativo. Com o projeto já decidido, me deparei com uma questão: eu poderia considerar as distâncias em linha reta entre uma cidade e outra, o que representaria uma viagem de avião ou poderia considerar a distância por estradas, ou seja, o percurso feito com um carro ou ônibus por exemplo. Por fim acabei fazendo as duas versões, visto que cada uma tem suas particularidades.

Decidi, para deixar o programa mais visual, pegar uma imagem do mapa do Brasil e trassar nela linhas entre as capitais representando as rotas do melhor indivíduo de cada geração, assim ficaria mais fácil de visualizar o desenvolvimento das soluções. Observe uma rota qualquer gerada pelo programa que eu fiz na imagem a seguir.

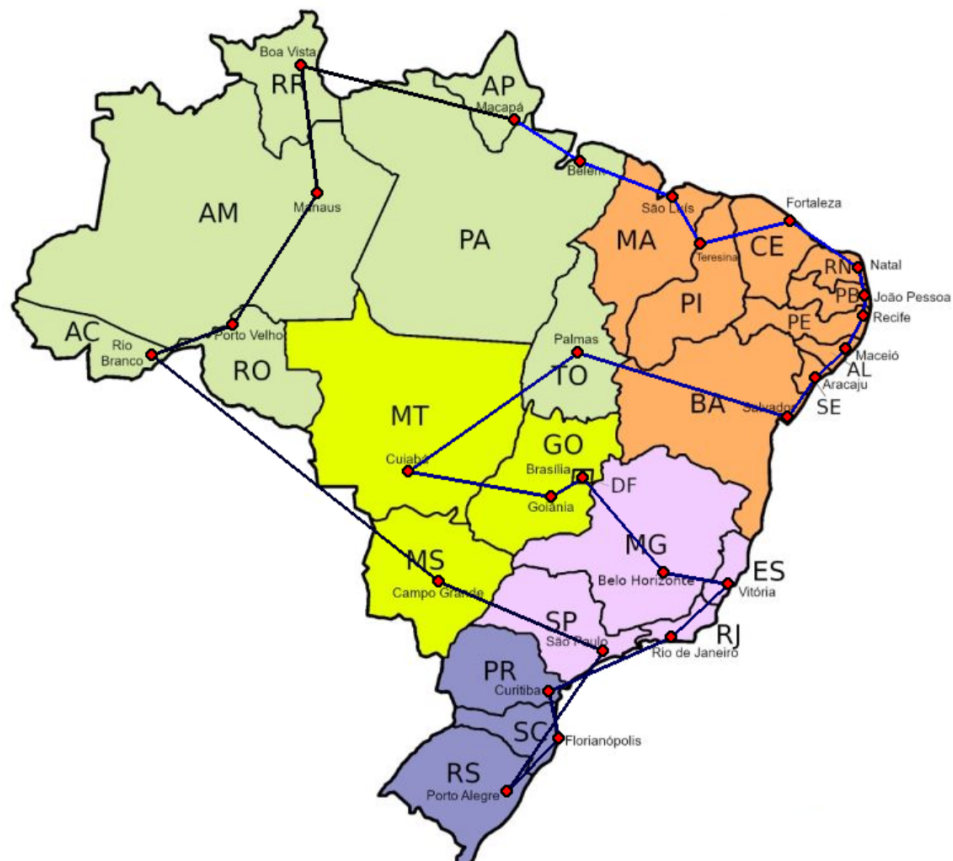


Figura 9: Exemplo de rota mapa do Brasil.

Podemos perceber que as linhas entre uma cidade e outra são apenas ilustrativas, pois seria muito mais difícil representar as rotas por rodovias ou então a rotas real dos aviões. Por isso podemos ver que ocorre de uma linha atravessar o oceano como na ligação entre Porto Alegre e São Paulo. Quando consideramos a distância por estradas podemos perceber que visualmente a rota nem sempre parecerá a melhor opção. Isso se da pelo fato de que as rotas por estrada nem sempre vão se assemelhar a linha trassada no mapa, como mostra a seguinte imagem.

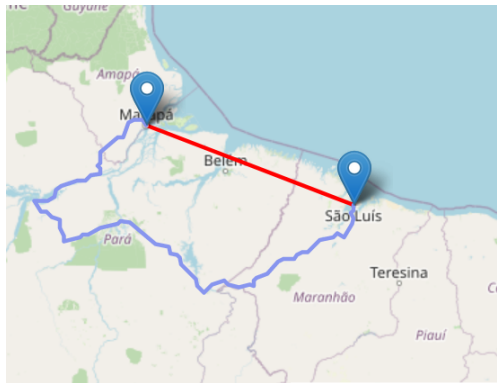


Figura 10: Diferença entre distâncias.

9 Conclusão

Pude perceber, após o desenvolvimento do meu trabalho, que os algoritmos genéticos são úteis em diversas situações mas que, no entanto, não serão uma boa opção para qualquer problema de busca ou otimização, eles são apenas uma alternativa a ser considerada. Percebi também os algoritmos genéticos não buscam a melhor solução, buscam apenas uma solução viável. Porém o fato que mais me intrigou é que podemos relacionar áreas de conhecimento para observar o mundo de outra maneira. Aprendi também a estruturar um projeto de programação do início ao fim e a me organizar para cumprir com as demandas e expectativas. Pretendo, a partir de agora, me aprofundar ainda mais nos estudos de algoritmos e de programação. Além de realizar outros projetos envolvendo algoritmos genéticos para problemas mais complexos e de outras origens.

Referências

- [1] Charles Darwin. *On the origin of species*, 1859. Routledge, 2004.
- [2] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [3] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addison wesley*, 1989(102):36, 1989.
- [4] E F Robertson J J O'Connor. Biography of william rowan hamilton. [urlhttps://mathshistory.st-andrews.ac.uk/Biographies/Hamilton/](https://mathshistory.st-andrews.ac.uk/Biographies/Hamilton/), 1998.
- [5] Diogo C Lucas. Algoritmos genéticos: uma introdução. *Apostila referente a disciplina de Inteligencia Computacional, Universidade Federal do Rio Grande do Sul, Brasil*, 2002.