Grafos - Caixeiro Viajante

Carolina Ribeiro Xavier

Maio de 2024

1 Problema do Caixeiro viajante

O problema do caixeiro viajante consiste na busca por um circuito que possua a menor distância, começando numa cidade qualquer, entre várias, e visitando todas cidades, cada uma precisamente uma vez, voltando então para a cidade de origem.

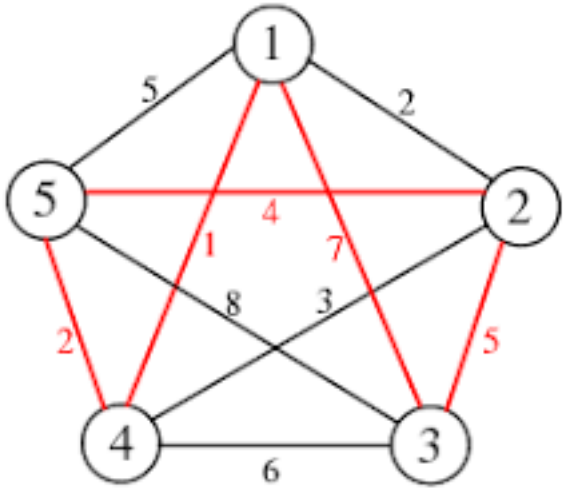

Figura 1: Grafo completo para aplicação do problema do caixeiro viajante

Dado um conjunto $C = \{c_1\ , ...,\ c_n\}$ de n cidades c_i e uma matriz de distâncias (ρ_{ij}), onde $\rho_{ij} = \rho(c_i,\ c_j)((i,\ j \in \{1,\ ...,\ n\}, \rho_{ij} = \rho_{ji}, \rho_{ii} = 0)$, a tarefa passa por encontrar a permutação $\pi \in S_n = \{s : \{1, ..., n\} \rightarrow$

$\{1, ..., n\}\}$ que faça com que a função objetivo (distância do circuito) $f : S_n \to \mathbb{R}$, onde:

$$f(\pi) = \sum_{i=1}^{n-1} \rho(\pi(i), \pi(i+1)) + \rho(\pi(n), \pi(1)), \tag{1}$$

seja mínima.

2 Questões de projeto

Como já foi dito, existem várias questões que podem mudar de acordo com o problema a ser resolvido, as mais importantes são as questões relativas à função objetivo escolhida e a representação, sendo que a segunda reflete na implementação de detalhes de todos os operadores genéticos. As principais questões são as listadas a seguir:

- Função objetivo;
- Representação - permutação das cidades;
- Estratégia de seleção; ✓
- **Cruzamento**;
- Mutação;
- Elitismo; ✓

Valores como tamanho da população e número máximo de gerações do AG também devem ser testados para verificar o melhor cenário para solução do problema que se quer resolver.

2.1 Função objetivo

A função objetivo do problema do caixeiro viajante será direta, basta fazer o cálculo da distância do percurso dado por 1.

2.2 Representação

A representação da solução do problema do caixeiro viajante será feita por um vetor de n posições, que contenha alguma permutação dos vértices da instância a ser otimizada (minimização).

2.3 Cruzamento

Para o cruzamento vamos mudar completamente o que vinhamos fazendo, por se tratar de um problema de permutação. Veremos um cruzamento baseado em ordem, o *ox-crossover*.

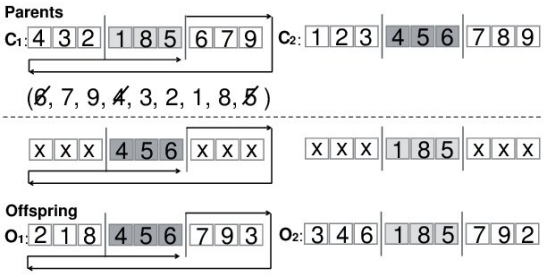

Figura 2: Cruzamento baseado em ordem

Neste operador, sortearemos dois pontos, os valores entre estes dois pontos serão preservados nos filhos na mesma posição que eles ocorrem nos pais. Como são dois filhos, gerados por dois pais, cada filho herda o trecho de um dos pais, o restante do filho será preenchido de acordo com a ordem que os índices ocorrem no segundo pai(o que passou o trecho sequencial para o outro filho).

Esse preenchimento é feito de forma a não ocorrer repetição no indivíduo, observe a Figura 2, o O1 está com o trecho preservado de C2 e as posições marcadas na segunda linha com x, serão preenchidas de acordo com C1. Partindo da posições subsequente ao segundo ponto sorteado, as posições recebem o índice de C1 se ele ainda não fizer parte de O1. Neste exemplo, ele não recebe o valor 6, recebe os valores 7 e 8, passa para o início do pai C1, não recebe o valor 4, recebe o valor 3, e passa a preencher o início de O1, recebendo os valores 2, 1 e 8.

Observe que dessa forma, a maioria das arestas potencialmente boas nos pais que passaram por uma seleção, permanecem nos filhos, as alterações

permitem um busca em soluções próximas as soluções encontradas até agora.

2.4 Mutação

O operador de mutação será bastante simples. Para cada gene do cromossomo sortearemos uma valor $r \in [0, 1]$, se o $r \leq pm$, sortearemos a posição de outro gene e faremos a troca dos valores desses dois genes.

Pode ocorrer de haver mais de uma troca em um mesmo indivíduo, mas é importante que você faça uma troca de cada vez para garantir a manutenção de todos os valores da permutação.

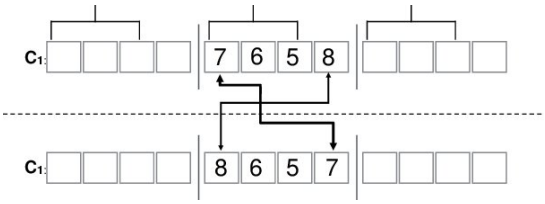

Figura 3: Mutação por troca de posição

3 Implementação

Agora vamos implementar um AG simples para a caixeiro viajante, que seguirá os passos do fluxograma da Figura 4.

Você deve definir a estrutura de dados que você irá armazenar os indivíduos e seus respectivos valores de *fitness*.

Defina os valores listados de acordo com o experimento fatorial que vimos na última tarefa e use a alguma instância encontrada **aqui** ou **aqui**.

As instâncias que melhor representam o problema no primeiro link são:

LAU15 - que possui o resultado da solução ótima e a matriz de distâncias (lau15_dist.txt)

SGB128 - que não traz a solução ótima, mas possui a matriz de distâncias (sgb128_dist.txt)

No segundo link você deve se atentar aos dados possuam problemas simétricos.

- Tamanho da população;

- Número máximo de gerações a serem executadas;
- Critério de seleção de pais;
- Taxa de cruzamento;
- Taxa de mutação;
- Elitismo.

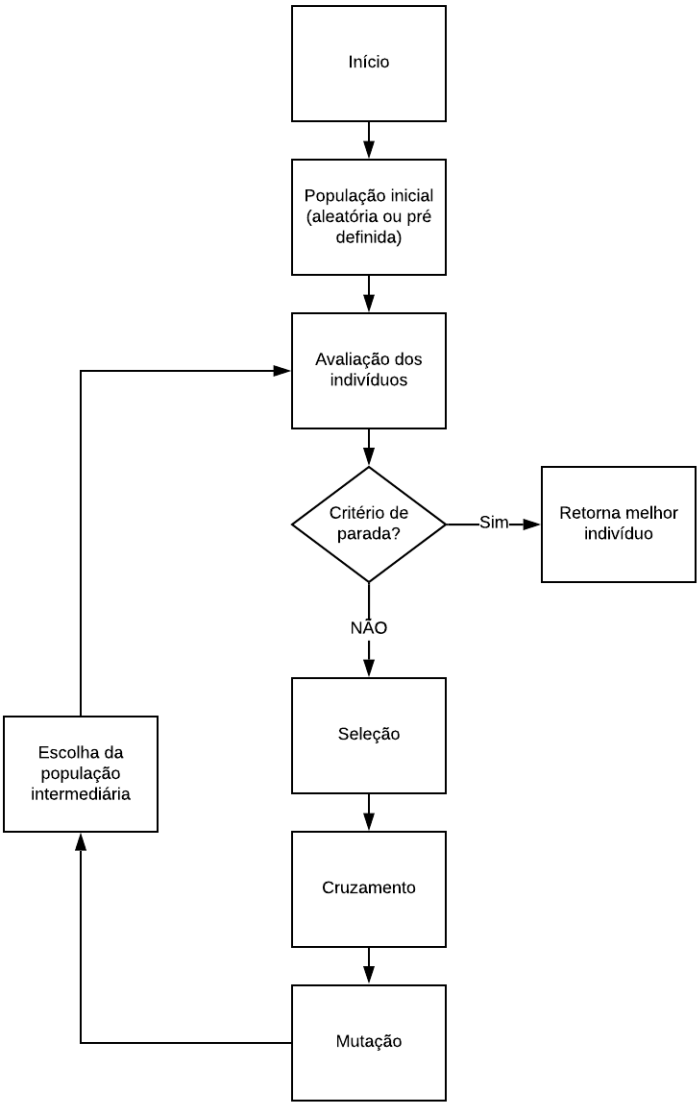

Figura 4: Fluxograma de um algoritmo genético básico