

Inteligência Artificial

Trabalho AV3 - Busca/Otimização Meta-heurística

Professor: Prof. Msc. Paulo Cirillo Souza Barbosa

2.2) Problema da caixeiro-viajante.

O problema do caixeiro viajante, é um problema de logística, em que há uma lista de cidades a serem visitadas e uma rota a ser desempenhada. Este problema é baseado na seguinte pergunta: Sejam cidades a serem visitadas e suas distâncias em pares, qual é a menor rota para visitar todas as cidades e retornar para a cidade origem? Este é um problema clássico de busca/otimização combinatória, e sua complexidade depende da quantidade de cidades existentes na lista. Pode-se pensar que é possível encontrar a solução do problema ao listar todas as possíveis combinações e compará-las umas com as outras. Contudo, pode não ser viável resolver este problema desta maneira, tendo em vista que com apenas 20 cidades, o número de rotas possíveis é 20!. Assim, cabe a utilização de um algoritmo genético para solucionar tal problema.

Para o presente trabalho, considere que o contexto em que se encaixa o problema do caixeiro viajante é o seguinte: Um drone precisa realizar acesso a pontos em um espaço tridimensional e retornar à sua origem. Os pontos que o drone deve acessar, são gerados de maneira aleatória, e para fins didáticos um exemplo desses posicionamentos é exibido na Figura 1.

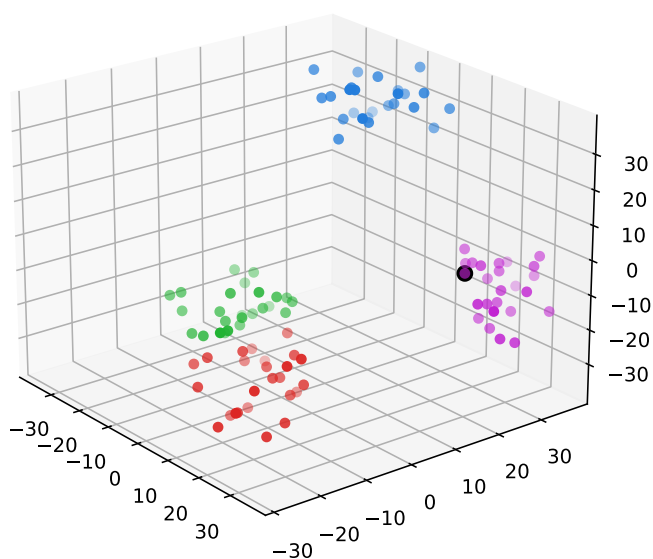


Figura 1: Problema das 8-rainhas.

Para o algoritmo genético, considere que cada indivíduo da população, são rotas que podem ser seguidas. Assim, cada estrutura cromossômica do problema é composta por k genes representando cada um dos pontos a serem visitados. Pode-se exemplificar o problema a ser resolvido, utilizando um exemplo em duas dimensões como se segue:

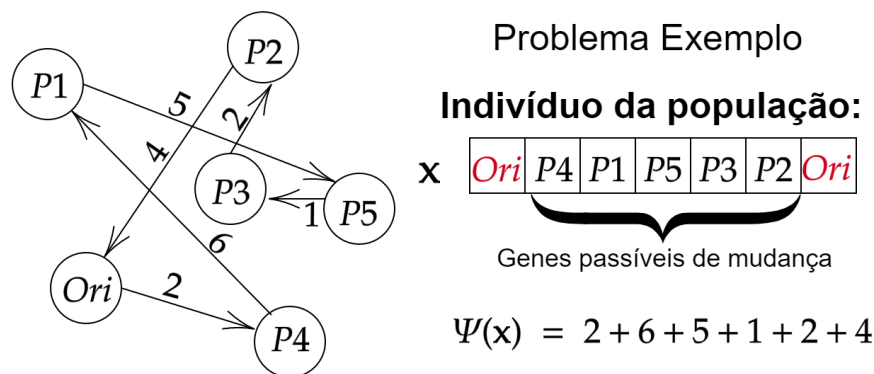


Figura 2: Problema do caixeiro viajante.

Assim, solicita-se que acesse os pontos disponibilizados no AVA (CaixeiroGrupos.csv) e faça com que seja aplicado um GA para encontrar uma trajetória subótima.

1. Faça a definição de quantos pontos devem ser gerados por região. Escolha um valor $30 < N_{\text{pontos}} < 60$.
2. Faça a definição da quantidade N de indivíduos em uma população e quantidade máxima de gerações.
3. Projete o operador de **seleção**, baseado no método do torneio.
4. Na etapa de **recombinação**, como este trata-se de um problema de combinatória, não pode haver pontos repetidos na sequência cromossômica. Desta maneira, pede-se que desenvolva uma variação do operador de recombinação de dois pontos. Assim, cada seção selecionada de modo aleatório deve ser propagada nos filhos e em seguida, a sequência genética do filho deve ser completada com os demais pontos sem repetição.
5. Na prole gerada, deve-se aplicar a mutação com probabilidade de 1%. Para o problema do caixeiro viajante, deve-se aplicar uma mutação que faz a troca de um gene por outro de uma mesma sequência cromossômica.
6. O algoritmo deve parar quando atingir o máximo número de gerações ou quando a função custo atingir seu valor ótimo aceitável (de acordo com a regra descrita no slide 31/61).
7. Faça uma análise se de qual é a moda de gerações para obter uma solução aceitável. Além disso, analise se é necessário incluir um operador de elitismo em um grupo N_e de indivíduos.

5) Relatório.

Além das implementações, o presente trabalho deve ser entregue em modelo de relatório. Este deve possuir as características descritas nos slides de apresentação do curso. Desta maneira, deve possuir:

1. Título.
2. Resumo.
3. Introdução.
4. Metodologia.
5. Resultados.
6. Conclusões.
7. Referências.

O modelo para trabalho pode ser encontrado neste [LINK](#)

6) Observações.

- Obs1: O envio das implementações é **obrigatório**. Caso a equipe não realize esta entrega, será atribuído nota **zero** para os respectivos alunos.
- Obs2: A data estipulada para entrega do trabalho, também é um critério avaliativo. Assim, caso haja atraso na entrega do trabalho, será aplicada: **de 00:15h até 24h: penalidade de 20% ; 24:15h até 48h: penalidade de 40% ; acima de 48h: penalização máxima (100%)**.
- Ob3: Os trabalhos e implementações serão enviadas a um software anti-plágio. Qualquer caracterização de plágio ocasionará em nota zero para ambas equipes.
- Obs4: Para o presente trabalho, não será permitido o uso de bibliotecas que tenham as implementações prontas dos métodos/modelos de qualquer algoritmo requisitado no trabalho. Caso sua equipe faça o uso de tais bibliotecas, serão descontadas as pontuações proporcionais.