



Inteligência Artificial

Trabalho 1, AV2 - Resolução de problemas por GAs

Professor: Prof. Msc. Paulo Cirillo Souza Barbosa

Discussão inicial sobre o trabalho.

O presente trabalho se trata da resolução de dois problemas utilizando um algoritmo genético a ser desenvolvido pela sua equipe. O primeiro, trata-se de um problema de domínio contínuo, em que deseja-se encontrar o mínimo de uma função. O segundo, trata-se de um problema de domínio discreto, baseado no problema do caixeiro viajante.

1) Problema de domínio contínuo.

Pede-se na primeira etapa, que utilize o algoritmo genético para encontrar o mínimo da função de Rastrigin. Essa clássica função tem características de ser não convexa, multimodal (com muitos pontos de mínimos locais) e multivariada. A função de Rastrigin é descrita como:

$$f(\mathbf{x}) = A \cdot p + \sum_{i=1}^p (x_i^2 - A \cdot \cos(2 \cdot \pi \cdot x_i))$$

Em que $A = 10$ é uma constante e p é a quantidade de dimensões do problema a ser resolvido. Assim, considerando que para todas as p variáveis, tem-se como limite de restrição, $[-10, 10]$ e que $p = 20$, utilize um algoritmo genético para resolver o problema de minimização da função. Nesse processo, faça as seguintes definições:

- Uma representação cromossômica através de uma sequência de bits.
- A função aptidão deve ser $\Psi(x) = f(x) + 1$.
- Utilize o método de seleção pela roleta.
- Faça a escolha do percentual de recombinação.
- Escolha a maneira de recombinar dois parentes.
- Mutações fora de ordem serão aplicados nos indivíduos da prole gerada.
- Escolha a estratégia associada ao critério de convergência do algoritmo genético.

É interessante que hajam 100 chamadas ao algoritmo genético. Ao final destas execuções (chamada de rodadas), construa uma tabela com as seguintes informações:

- O menor valor de aptidão obtido para todas as rodadas.
- O maior valor de aptidão obtido para todas as rodadas.
- A média de valor de aptidão obtido para todas as rodadas.
- O desvio-padrão de valor de aptidão obtido para todas as rodadas.

2) Problema de domínio discreto.

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, estão disponibilizados no AVA, e podem ser visualizados na Figura 1. Nesta figura, é possível verificar que há um ponto em destaque, o qual deve ser tratado como o ponto de origem em que o drone deve sair e que não pode ser modificado. No arquivo disponibilizado no AVA (CaixeiroSimples.csv), existem 101 linhas com tais pontos, sendo a linha 1 composta pela coordenada deste ponto (0,0,0) de partida.

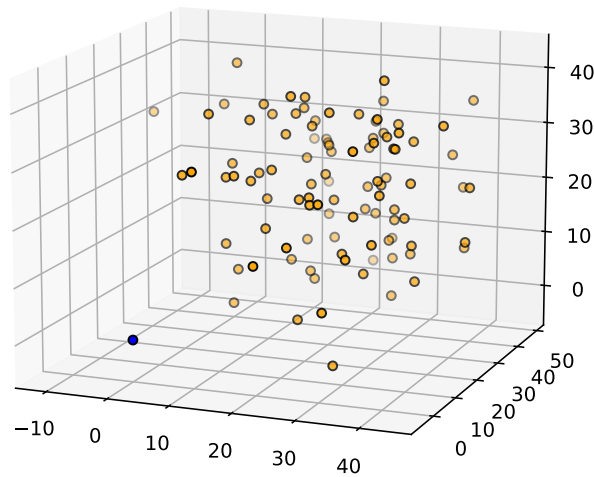


Figura 1: Problema das 8-rainhas.

Para o algoritmo genético, considere que cada indivíduo da população, são rotas que podem ser seguidas. Assim, cada indivíduo, possui uma sequência cromossômica do problema é composta por 100 genes representando cada um dos pontos a serem visitados. Pode-se exemplificar o problema a ser resolvido, bem como o projeto de sua função aptidão, ao exemplificá-lo com um problema de 5 pontos a serem visitados, com exceção da origem. Tal exemplo é destacado pela Figura 2.

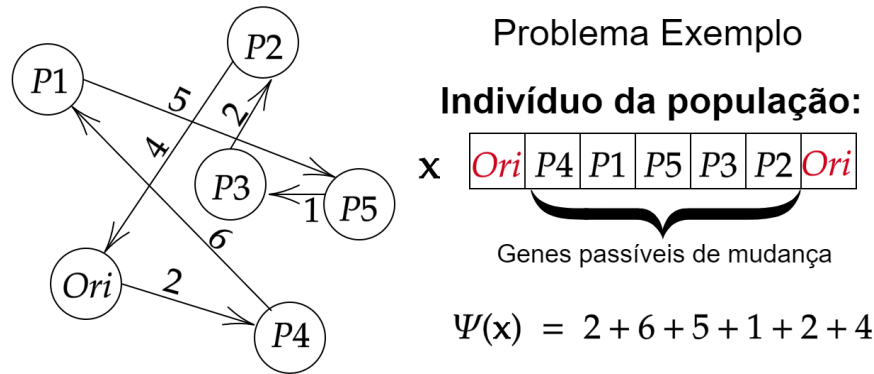


Figura 2: Problema do caixeiro viajante.

Assim, de posse do algoritmo base presente na seção de anexos, faça o que se pede:

1. Faça a definição da quantidade N de indivíduos em uma população e quantidade máxima de gerações.
2. Projete o operador de **seleção**, baseado no método do torneio.
3. 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.
4. Na prole gerada, deve-se aplicar a mutação com probabilidade de 1%. Deve-se bolar uma maneira interessante de aplicar tal operador, pois, não há a possibilidade de visitar duas vezes um mesmo ponto.
5. O algoritmo deve parar quando atingir o máximo número de gerações ou quando a função custo atingir seu valor ótimo aceitável (descritas no slide).
6. Faça o uso do elitismo, com uma quantidade N_e de elites presentes a cada nova geração.

É interessante que hajam 100 chamadas ao algoritmo genético. Ao final destas execuções (chamada de rodadas), construa uma tabela com as seguintes informações:

- O menor valor de aptidão obtido para todas as rodadas.
- O maior valor de aptidão obtido para todas as rodadas.
- A média de valor de aptidão obtido para todas as rodadas.
- O desvio-padrão de valor de aptidão obtido para todas as rodadas.

Como última tarefa, pede-se para resolver uma extensão do problema anterior. No AVA, há um documento chamado CaixeiroGrupos.csv. Neste, há um total de 161 pontos a serem visitados pelo drone. No entanto, há grupos de pontos específicos em que o drone deve visitar e voltar ao ponto de origem. A extensão do problema se dá ao restringir que o drone apenas poderá ir para um grupo de pontos, somente quando visitar todos os pontos do grupo atual. No documento, existem 161 linhas e 4 colunas, no qual as três primeiras colunas representam as coordenadas X,Y,Z do espaço cartesiano, e a quarta coluna representa a qual grupo aquela coordenada pertence. Existem 5 grupos diferentes de pontos, em que:

- 0, representa o ponto de origem.

- 1, representa o grupo 1.
- 2, representa o grupo 2.
- 3, representa o grupo 3.
- 4, representa o grupo 4.

A Figura 3 apresenta um espalhamento de tais pontos com seus respectivos grupos.

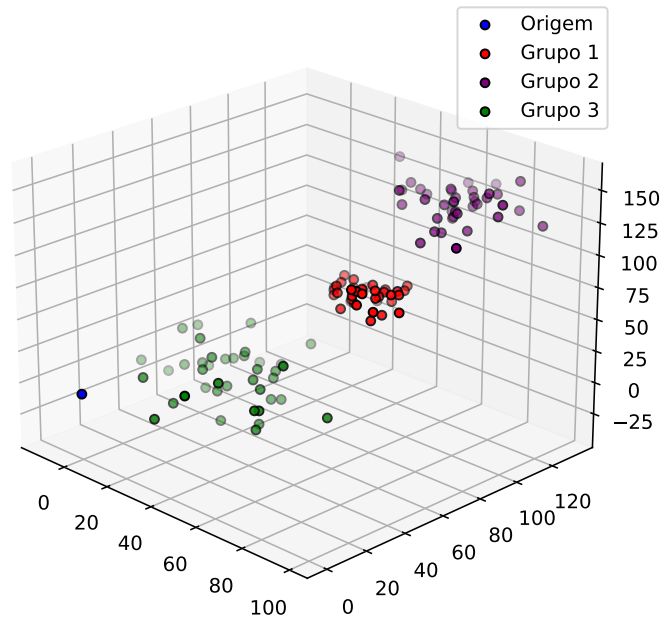


Figura 3: Problema do caixeiro viajante estendido.

Para a solução do problema estendido, pede-se que defina cada uma das características que formam o algoritmo genético. Descreva no relatório sobre as suas escolhas e resultados obtidos. Da mesma forma como nos problemas anteriores, deve-se compor os resultados na forma:

- O menor valor de aptidão obtido para todas as rodadas.
- O maior valor de aptidão obtido para todas as rodadas.
- A média de valor de aptidão obtido para todas as rodadas.
- O desvio-padrão de valor de aptidão obtido para todas as rodadas.

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. Fundamentação Teórica (Revisão Bibliográfica).
5. Metodologia.

6. Resultados.

7. Conclusões.

8. 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%)**.
- Obs3: 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 modelos Perceptron de qualquer algoritmo requisitado no trabalho. Caso sua equipe faça o uso de tais bibliotecas, serão descontadas as pontuações proporcionais.