

TRABALHO DE INTELIGÊNCIA ARTIFICIAL

Prof. Bonfim Amaro Júnior | Semestre: 2022.1

OBJETIVO: Complementar os estudos a partir de implementação computacional ou apresentação de temas pertinentes ao conteúdo da disciplina. Dessa forma, a equipe escolhe, pelo menos, uma opção para realizar a entrega/apresentação ao professor e a turma.

Equipe composta por 4 alunos, no máximo.

Data Limite para entrega (Implementação): 14/07/2022

Data das Apresentações: 12/07/2022 e 14/07/2022

Opção 1 – Implementação: Algoritmo Genético projetado para problema de permutação.

O problema de permutação a ser estudado é o caixeiro viajante com (origem e destino) variáveis. O problema do caixeiro viajante é de natureza combinatória e aparece em diversas aplicações práticas. Em sua forma mais simples (simétrica), o caixeiro viajante deve visitar cada cidade em um dado território somente uma vez e depois retornar à cidade de origem. Dado o custo da viagem entre cada par de cidades, a pergunta que desejamos responder é: qual o itinerário (caminho no grafo subjacente) que resulta no custo mínimo total? A figura 1 traz um esboço desse processo de otimização.

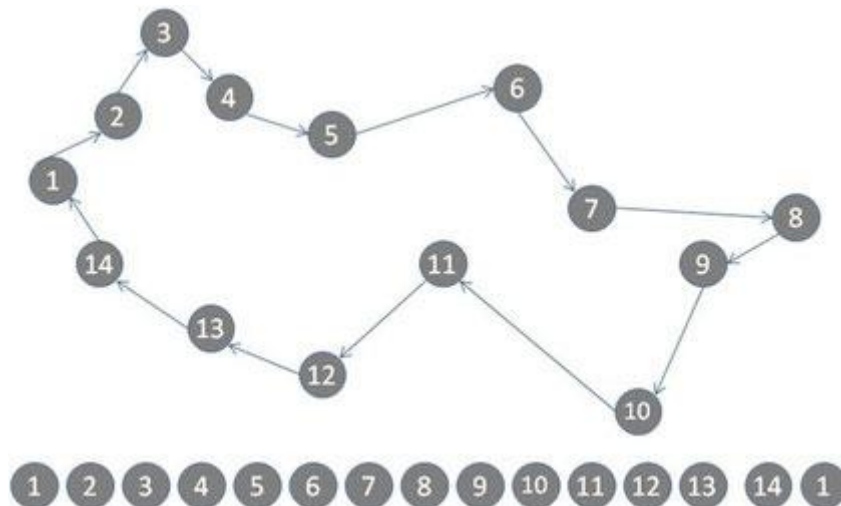


Figura 1 – Definição de uma solução para o caixeiro viajante com 14 cidades.

Cada solução/indivíduo corresponderá a uma possível rota. Assim, as soluções deverão satisfazer duas restrições:

- (i) a de cada cidade a ser visitada apareça em alguma posição da rota;
- (ii) a de que uma mesma cidade não apareça mais de uma vez no itinerário.

A função de Fitness (aptidão) será computada a partir soma total dos custos de movimentação entre cada par de cidades da rota, devendo ser calculado também o custo entre a última e a primeira cidade da sequência. Como a versão desse problema é simétrica, o custo de ir de uma cidade A para uma cidade B é o mesmo de ir de B para A.

O estudo será realizado sobre três instâncias distintas, cada uma com um número diferente de cidades e mapeadas nos arquivos tspcit30.dat, tspcit100.dat, tspcit101.dat.

Observações:

1. Os arquivos tspcit*.dat são fornecidos com a seguinte configuração: na primeira linha é apresentado o número de cidades, e nas linhas seguintes, as coordenadas (x,y) de cada cidade. A ordem de apresentação das coordenadas das cidades nos arquivos tspcit*.dat é arbitrária. Percorrendo as cidades nesta ordem, a distância resultante é maior que a distância mínima possível.
2. O objetivo é encontrar uma combinação de tal modo que, uma vez percorrendo as cidades nesta ordem, a distância resultante seja mínima.
3. Não esquecer de contabilizar também o caminho de retorno da última cidade para a primeira.
4. Com a ordenação atual, a distância para percorrer as cidades, na sequência, apresenta os seguintes valores:
 - a. 100 cidades: 109952,5821;
 - b. 101 cidades: 213670,6846;
 - c. 30 cidades: 123865,3550;
5. As distâncias “ótimas conhecidas” para os três casos são dadas a seguir:
 - a. 100 cidades: 21285,4432;
 - b. 101 cidades: 65455,8441;
 - c. 30 cidades: 48872,4026;
6. Estas soluções ótimas devem servir apenas para indicar a quão próxima do ótimo se encontra uma solução fornecida pelo seu algoritmo. De forma alguma estes valores ótimos devem ser utilizados para obter o fitness de um candidato a solução ou direcionar de algum modo o processo de evolução. Ou seja, estes valores ótimos não podem estar presentes no código de seu programa.
7. Não existe nenhuma relação direta entre os três problemas. Por exemplo, as 30 cidades presentes em ncit30.dat não representam um subconjunto das 100 cidades de ncit100.dat.

Ademais, os seguintes parâmetros serão os mesmos para todas as execuções do seu AG.

P = 300 (Tamanho da População); TxCross = 0.80; TxMutacao = 0.1.

Critério de Parada: parar a execução, caso o valor do *fitness* do melhor indivíduo não melhore nas últimas 50 gerações. **Operador de seleção** será baseado em torneios de tamanho 2. **Aplicar Elitismo para geração populacional.**

A equipe deverá executar 10 vezes para cada instância. Os critérios de desempenho a serem analisados são:

- (i) Melhor rota encontrada para cada execução x instância
- (ii) Valor médio e desvio-padrão do *fitness* das melhores soluções finais para cada execução x instância

- (iii) Número médio de gerações para cada execução x instância
- (iv) Tempo de simulação para cada execução x instância

Opção 2 – Implementação/Apresentação: Mundo Wumpus usando Base de Conhecimento

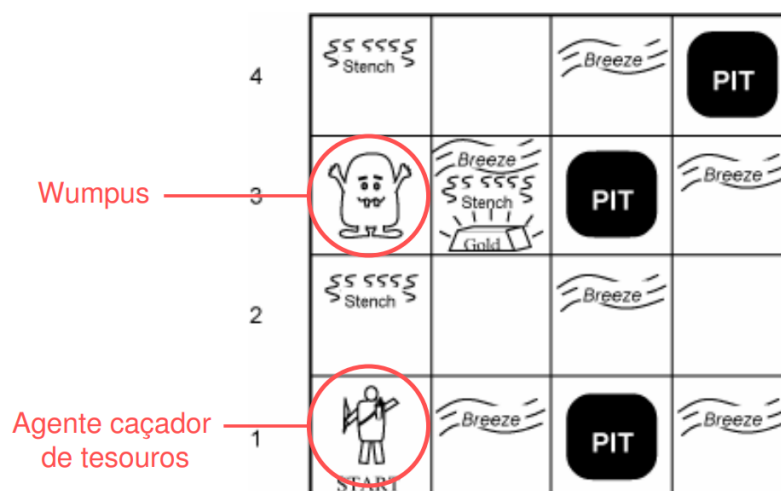
O agente *Wumpus World* é um exemplo de agente baseado no conhecimento que representa representação, raciocínio e planejamento do Conhecimento. O agente baseado em conhecimento vincula o conhecimento geral às percepções atuais para inferir caracteres ocultos do estado atual antes de selecionar ações. Sua necessidade é vital em ambientes parcialmente observáveis.

Declaração do problema:

O mundo Wumpus é uma caverna com 16 quartos (4×4). Cada sala está conectada a outras por meio de passarelas (nenhuma sala está conectada diagonalmente). O agente baseado em conhecimento começa na Sala [1, 1]. A caverna tem alguns poços, um tesouro e uma besta chamada Wumpus. O Wumpus não consegue se mover, mas come aquele que entra em seu quarto. Se o agente entrar no poço, ele ficará preso lá. O objetivo do agente é pegar o tesouro e sair da caverna. O agente é recompensado, quando as condições do objetivo são atendidas. O agente é penalizado, quando cai em uma cova ou é comido pelo Wumpus.

Alguns elementos ajudam o agente a explorar a caverna, como:

- Os quartos adjacentes do Wumpus são fedorentos.
- O agente recebe uma flecha que pode usar para matar o wumpus quando o enfrentar (Wumpus grita quando é morto).
- As salas adjacentes da sala com fossos estão cheias de brisa.
- A sala do tesouro está sempre brilhante.



A descrição PEAS ajuda no agrupamento dos agentes. Descrição PEAS para o problema do Wumpus World:

Medidas de desempenho:

- Agente obtém o ouro e retorna em segurança = +1000 pontos
- Agente morre = -1000 pontos
- Cada movimento do agente = -1 ponto
- Agente usa a seta = -10 pontos

Ambiente:

- Uma caverna com 16 (4×4) quartos
- Salas adjacentes (não diagonalmente) ao Wumpus estão fedendo
- Salas adjacentes (não diagonalmente) ao poço são arejadas
- A sala com os brilhos de ouro
- Posição inicial do agente - Sala [1, 1] e voltado para o lado direito
- A localização de Wumpus, ouro e 3 poços pode ser em qualquer lugar, exceto na Sala [1, 1] .

Atuadores:

- Dispositivos que permitem ao agente realizar as seguintes ações no ambiente.
- Siga em frente
- Vire à direita
- Vire à esquerda
- Atirar
- Pegar
- Liberar

Sensores:

- Dispositivos que ajudam o agente a detectar o seguinte do ambiente.
- Brisa
- Fedor
- Brilho
- Grito (quando o Wumpus é morto)
- Bump (quando o agente bate em uma parede)

Caracterização do mundo Wumpus:

- Parcialmente observável: conhece apenas as percepções locais
- Determinístico: o resultado é precisamente especificado
- Sequencial: nível subsequente de ações realizadas
- Estático: Wumpus, os poços estão imóveis
- Discreto: ambiente discreto
- Agente único: O agente baseado em conhecimento é o único agente, enquanto o wumpus é considerado o recurso do ambiente.

Diante desse contexto, o trabalho consiste na implementação do problema do mundo wumpus usando o conceito de agente de conhecimento usando a ideia de base de conhecimento ou ainda, explicação de algum framework que simule o problema do *Wumpus World*.

Por exemplo:

- WUMPUS WORLD SIMULATOR (<https://eecs.wsu.edu/~holder/courses/AI/wumpus/>) (C++ e/ou Python)
- <https://github.com/katikireddy622/wumpus-world-java> (Usando A* e Probabilidade (em Java))
- <https://github.com/abdulzakrt/WumpusWorld-CSharp> (C# interface e Prolog para base de conhecimento (Consultas).
- (<https://github.com/luka1199/wumpus-world>) modelo de interface para implementação da base de conhecimento. Detalhe. A implementação pode ser realizada com a troca da base de conhecimento por uma estratégia evolutiva para aprendizado.

Opção 3 – Apresentação: Temas pertinentes ao Conteúdo.

As equipes que optarem por essa opção deverão apresentar seminário de 25-30 minutos sobre cada um dos temas a seguir:

- A inteligência artificial no metaverso. (Tema 1)
- A combinação entre tecnologia e a humanização no contato com o cliente com a inteligência artificial. (Tema 2)
- Combate às *Fake News* com IA. (Tema 3)
- Redes Neurais Convolucionais. (Tema 4)
- Aplicação De *Data Science* Como Ferramenta De Apoio A Tomada De Decisão. (Tema 5)
- Aplicação dos algoritmos genéticos no aprendizado de redes neurais artificiais (NeuroEvolução). (Tema 6)
- Uso de Aprendizado de Máquina para Detecção de Fraudes em empresas. (Tema 7)
- Uso da inteligência artificial na educação. (Tema 8)

As equipes devem ser definidas e informadas pelo sigaa para que eu possa realizar o sorteio dos temas no dia (14/06/2022).