

## **1 – Quais são os métodos de construção de algoritmos heurísticos?**

- **Método construtivo** -> processo iterativo que inicia com uma solução vazia e adiciona um novo elemento a cada iteração até obter uma solução.
- **Busca local (melhorativo)** -> inicia em uma solução (podendo ser obtido a partir de outra heurística) e caminha sobre as soluções vizinhas.
- **Meta-heurística** -> estrutura genérica de heurística que pode ser adaptada para diversos problemas.
- **Métodos de decomposição** -> dividir o problema em subproblemas menores de modo que a resolução de todos os subproblemas possam compor uma solução para o problema maior.
- **Relaxação** -> modificar o modelo de tal forma que ele fique mais fácil de resolver.

## **2 – Defina heurística. Dê pelo menos 2 exemplos.**

A palavra heurística significa descobrir, é um método para resolver problemas de forma rápida, baseado em informações, intuições e experiência. É um conjunto de regras e métodos que conduzem a descoberta, a invenção e a resolução de problemas complexos quando as soluções exatas são difíceis ou impossíveis de encontrar. Alguns exemplos de heurísticas são:

- Heurística de vizinhança mais próxima.
- Heurística de divisão e conquista.
- Heurística de busca em largura.
- Heurística de busca em profundidade.

## **3 – Cite exemplos de algoritmos heurísticos e meta-heurísticos.**

- Algoritmo guloso.
- Busca Local.
- Algoritmos Genéticos.
- Busca Tabu
- Simulated Annealing.
- Ant System.
- Ant Colony System.

## **4 – Os algoritmos heurísticos garantem uma solução ótima? E os meta-heurísticos? Cite exemplos, caso haja algum que garanta.**

A maioria dos algoritmos heurísticos e meta-heurísticos não são capazes de garantir uma solução ótima para problemas de otimização, porém existem alguns algoritmos heurísticos que garantem uma solução ótima como: algoritmo A\* com heurística admissível, algoritmo de Prim e Kruskal para árvore expandida de custo mínimo.

## **5 – Quais as características da busca Tabu? Para que serve a lista Tabu? Quais são os parâmetros que podem ser usados para entrada e saída da lista Tabu?**

Algumas características da Busca Tabu:

- Exploração Intensiva: explora o espaço de soluções de forma intensiva, evitando soluções já visitadas buscando encontrar melhores.
- Estratégia de diversificação e intensificação: explora novas soluções enquanto evita soluções visitadas.
- Adaptação: a lista Tabu é atualizada ao longo do processo de busca, permitindo que a busca se adapte as mudanças no espaço de soluções.
- Uso de estruturas de memória adaptativas.
- Possui muitos parâmetros, muitas estruturas e possibilidades de condução de busca.
- Traz soluções de boa qualidade.
- Depende um pouco da solução inicial, pois possui mecanismos eficientes que permite escapar de ótimos locais.

A busca Tabu é uma técnica de otimização que utiliza uma lista tabu, que é uma estrutura de memória básica que serve para proibir movimentos que levariam a soluções já visitadas e que não possuem soluções melhores ou trazem riscos de ficarem em loop, durante um tempo tabu e segue a busca tentando caminhos promissores. Os parâmetros de entrada são: tamanho da lista tabu, critério de entrada (se deve ou não entrar na lista), critério de saída (para remover da lista).

## 6 – Quais os elementos que compõe os algoritmos genéticos?

- **População:** conjunto de soluções candidas que evoluem ao longo do tempo, cada solução é representada por um cromossomoconjunto k de estados gerados aleatoriamente.
- **Indivíduo/cromossomo:** como uma estado/solução é chamada.
- **Função de aptidão (fitness):** responsável por avaliar cada estado ou indivíduo, a fim de selecionar aqueles que possuem melhores características.
- **Seleção dos indivíduos para reprodução:** os melhores indivíduos (maior aptidão) são selecionados para gerar filhos.
- **Crossover:** reprodução dos indivíduos selecionados.
- **Mutação:** indivíduos gerados podem sofrer mutações com pequena probabilidade, para ajudar a escapar de um ótimo local.
- **Adição de filhos a nova população:** podem ser simples (nova geração substitui a antiga) ou elitista (nova geração se mistura com a antiga).
- **Geração:** criada a partir da seleção, crossover, da geração anterior.

## 7 – Quais são os passos para a criação do algoritmo genético?

quantas gerações?

Definir representação do cromossomo/indivíduo, definir função fitness, inicializar a população (aleatória), selecionar os mais aptos para reprodução, realizar crossover dando inicio a uma nova geração, verificar se houve mutações, adicionar filhos a nova população de acordo com um critério de substituição e definir um critério de parada.

## 8 – Quais as formas de seleção de população utilizadas em algoritmos genéticos e como elas funcionam?

Os tipos, servem para selecionar os melhores indivíduos (mais aptos) para gerar filhos, podendo ser:

- **Proporcional a aptidão (roleta)** -> Indivíduos com maior aptidão tem maior probabilidade de serem selecionados.
- **Torneio** -> Seleciona n (tipicamente 2) indivíduos aleatoriamente da população e o melhor é selecionado.
- **Ranking (os n mais adaptados)** -> Seleciona os n indivíduos mais adaptados.

## 9 – Explique o funcionamento do Simulated Annealing. Qual o parâmetro mais importante deste algoritmo.

O Simulated Annealing é um algoritmo de busca metaheurística que busca a solução ótima para um problema de otimização. A partir de uma alta temperatura deve ser permitido aceitar soluções ruins em alguns momentos, pois estamos longe do ótimo local, quando a temperatura começa a diminuir a possibilidade de alterações ruins vão reduzindo, pois estamos próximo do ótimo global.

O algoritmo começa com uma solução inicial qualquer, que é iterativamente modificada por pequenas mudanças aleatórias. O parâmetro mais importante do algoritmo é a **taxa de resfriamento**, que determina a velocidade com que a temperatura diminui, temperatura inicial, a função de transição de temperatura e o método de perturbação da solução.

## 10 – Quais são os parâmetros dos algoritmos Ant System?

- **Numero de formigas:** quantidades de formigas utilizadas no algoritmo para realizar as buscas no espaço de soluções (numero maior melhor capacidade de exploração, numero menor pode tornar o algoritmo lento).
- **Numero de iterações.**
- **Valor alfa e beta:** para o controle do algoritmos.
- **Intensidade do feromônio:** representa a quantidade de feromonios depositada pelas formigas a cada iteração (quanto maior pode aumentar a velocidade de convergencia e deixando mais propenso a ficar preso em minimos locais).
- **Taxa de evaporação do feromônio:** define a rapidez que o feromônio evapora (mais baixo ajuda a manter a diversidade no espaço de soluções, mas pode aumentar o tempo de execução).
- **Estratégia de atualização de feromonios.**
- **Tamanho da vizinhança:** define o tamanho da vizinhança que cada formiga explora em busca de soluções.

## 11 – Como funciona um algoritmo baseado em Ant System?

O algoritmo de Ant System simula colonias de formigas (artificiais) como forma de otimização, as formigas são usadas para buscar soluções em um espaço de soluções, cada formiga segue uma trilha de feromonios deixada por outras formigas e deposita o seu próprio feromônio, as formigas são atraídas pelas trilhas com niveis mais altos de feromonios, sendo influenciadas a seguir caminhos que já foram exploradas por outras formigas.

O algoritmo começa com uma população de formigas, distribuidas aleatoriamente no espaço de busca, cada formiga segue um caminho específico (determinado pela

combinação de heurísticas/feromonios). As formigas seguem uma regra simples para escolher seu próximo movimento (baseado na distancia e a taxa de feromonio). A medida que as formigas se movem pelo espaço de busca, elas deixam uma trilha de feromonios em seu caminho, atualizando a cada iteração com base na qualidade das soluções encontradas. O algoritmo segue até que um critério de parada seja atingido gerando a solução final

## **12 – Como é feito o cálculo de progressão no algoritmo A\*?**

O cálculo de progressão no algoritmo A\* é feito através do uso de duas funções, a função de custo de caminho  $g(n)$  e a função de heurística  $h(n)$ . A função  $g(n)$  calcula o custo atual do caminho percorrido do nó inicial até o nó atual  $n$ , enquanto a função  $h(n)$  estima o custo mínimo restante do nó  $n$  até o nó objetivo.

O cálculo de progressão é feito a partir da combinação dessas duas funções. O algoritmo A\* seleciona o próximo nó a ser expandido com base em uma medida de progressão, que é dada pela soma da função  $g(n)$  e da função  $h(n)$  para o nó atual. Essa soma é chamada de  $f(n)$ , que representa o custo total estimado para chegar ao nó objetivo passando pelo nó atual.

## **13 – O algoritmo A\* é ótimo? É eficiente? É completo?**

13 – O algoritmo A\* é ótimo e eficiente, pois a primeira solução que encontra já é a ótima, expande menos nós possíveis para encontrar a solução ótima, não é completo pois não expande todas as árvores de possibilidades.

## **14 – Avalie a seguinte afirmativa em verdadeira ou falsa e justifique sua resposta:**

**“Para um dado problema há uma única heurística que traz o melhor resultado possível, o principal objetivo é encontrá-la e depois disso testar os algoritmos com diversas configurações diferentes para se obter o melhor resultado possível”.**

Falso, para se chegar ao melhor resultado, basta testar todos os resultados possíveis, não existe encontrar a melhor heurística, escolho apenas uma heurística e testo as soluções para ela. Existem problemas que posso usar 5 heurísticas, e para alguns resultados  $x$  é melhor que  $y$ , mas não podemos generalizar.

Quando selecionamos uma heurística devemos defendê-la, o principal objetivo é escolher uma heurística e testar para vários resultados diferentes.