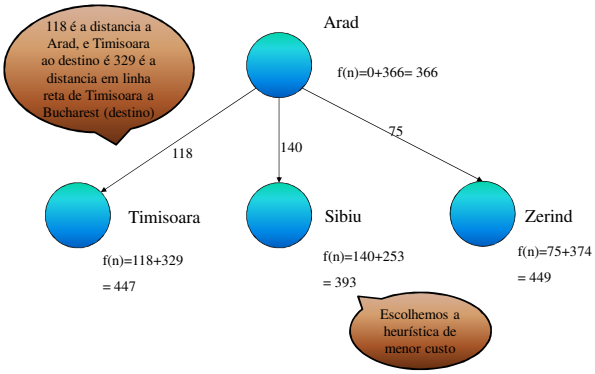
Heurísticas,
ou
"o que fazer quando
não se sabe o que fazer"

IA

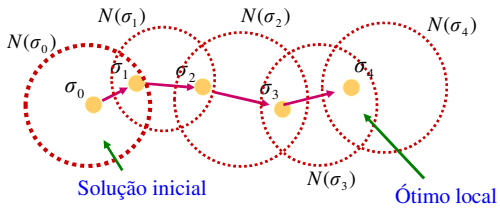
Heurística

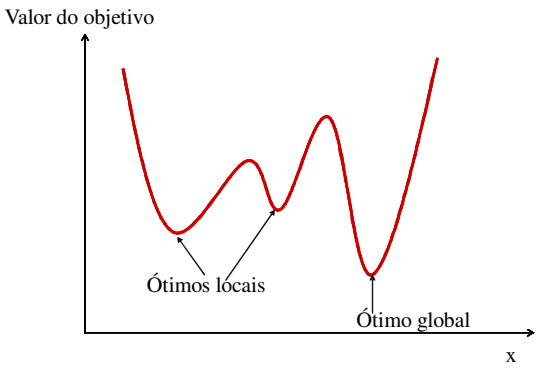
- "Heurística" deriva do grego *heuriskein*, Que significa "encontrar" ou "descobrir";
- Objetivos principais para o algoritmo:
  - Reduzir tempos de execução;
  - Encontrar boas soluções, geralmente a ótima.
- Uma heurística é um algoritmo atende pelo menos um desses objetivos;
- Não há garantia para obter a solução ideal (ou solução ótima), mas sim boas soluções;
- Heurística de construção:
  - Vai-se adicionando gradualmente componentes individuais para a solução;
  - Algoritmos "gulosos" (greedy): a cada passo, escolhe a melhor opção
    - Exemplo: algoritmo KNN aplicado ao problema do caixeiro viajante.

Usando A - Exemplo*



- Métodos de busca local
  - Explorando soluções "vizinhas", a passando de uma solução à outra que seja melhor;
  - Provável que estacione em ótimos locais;
  - Portanto, dependendo do ponto inicial, pode apresentar diferentes resultados;
  - Na busca cega, nenhuma informação coletada é utilizada durante a execução do algoritmo.





- Nas heurísticas podem ser incorporados mecanismos aleatórios de construção para "escapar" de ótimos locais;
- Razões para utilização de heurística:
  - Resolução de problemas complexos:
    - Grandes Problemas;
    - Problemas não-lineares;
    - Problemas combinatórios.
  - Ausência, ou ineficácia, de algoritmos ótimos;
  - Evitar ficar preso em ótimos locais.

Algoritmos de busca local

- ***Hill-Climbing***: Subida pela Encosta mais Íngrime (ou Busca Local Gulosa)
  - Só faz modificações que melhoram o estado atual.
- ***Simulated Annealing***: Têmpera Simulada
  - Pode fazer modificações que pioram o estado no momento, para possivelmente melhorá-lo no futuro.
- ***Local Beam Search***: Busca em feixe local
  - Mantém k estados em vez de um único.
- ***Algoritmos Genéticos (GA)***
  - É uma busca subida da encosta, estocástica, na qual uma grande população de estados é mantida e novos estados são gerados por mutação ou cruzamento.

Subida de Encosta pela Trilha mais íngreme

- Considera todos os movimento já feitos a partir do estado corrente selecionando o melhor como próximo estado;
- Algoritmo:
  1. Avalie o estado inicial. Se ele também for um estado meta, retorne-o e encerre, caso contrário, continue com o estado inicial como estado corrente;
  2. Repita o processo até encontrar uma solução ou até que uma iteração completa não produza nenhuma mudança no estado corrente:

- a) atribua a *sucessor* um estado;
- b) para cada operador que se aplique ao estado corrente faça o seguinte:
  - aplique o operador e gere um estado novo;
  - avalie o novo estado. Se for um estado-meta, retorne-o e encerre. Senão, compare-o a *sucessor*, se for melhor, então atribua-o a *sucessor*, caso contrário deixe o sucessor como está.

- Hill Climbing - Exemplos:
  - Cubos coloridos neste método, precisamos considerar todas as alterações do estado inicial e escolher a melhor;
  - Neste problema isto é difícil pois temos muitos movimentos possíveis (explosão combinatória);
  - Sempre há perdas e ganhos entre o tempo exigido para escolher um movimento e o número de movimentos necessários para chegar a uma solução;

- Os métodos vistos até então podem não encontrar uma solução, qualquer um dos algoritmos pode terminar, não porque o estado-meta foi encontrado, mas porque chegou-se a um estado que não há nenhum estado melhor que ele, isto ocorre quando um programa tiver alcançado o *máximo local*, um *platô* ou uma *cordilheira*;
  - *máximo local:* estado melhor que todos os seus vizinhos, mas não melhor que estados mais distantes;
  - *platô:* área dentro do espaço de busca onde o grupo de estados vizinhos tem o mesmo valor, não sendo possível determinar a melhor direção através de comparações locais;
  - *cume:*um tipo de máximo local. Área do espaço de busca mais alta que seus vizinhos que não permite atravessar a cordilheira num único movimento;

- modos para lidar com estes problemas:
  - Volte a um nó anterior e siga uma heurística diferente (máximos locais);
  - De um salto grande em alguma direção para tentar chegar a uma nova seção do espaço de busca (platôs);
  - Aplique duas ou mais regras, move-se em várias direções ao mesmo tempo (cordilheiras);
- Apesar de todos estas estratégias a subida de encosta não é muito eficiente

Têmpera Simulada

- Variação da subida de encosta;
- No início do processo podem ser feitos movimentos descendentes (morro abaixo) explorando suficientemente o espaço do problema, para que a solução final não fique presa a um máximo local, platô ou cordilheira;
- Utilizada em processos físicos de recozimento, onde substâncias físicas como os metais são fundidas (altos níveis de energia) e depois gradualmente resfriados até alcançar um estado sólido;

- Objetivo: produzir um estado final com um mínimo de energia (estado-meta);
- A função-objetivo (função heurística) é o nível de energia;
- A velocidade com que o sistema é resfriado é chamado de *cronograma de têmpera;*
- Algoritmo:
  1. Avalie o estado inicial. Se for também um estado-meta, então retorne-o e encerre. Caso contrário, continue com o estado inicial como estado corrente;
  2. Inicialize *melhor-até-o-momento* como igual ao estado corrente;

3. Inicialize T de acordo com o cronograma de têmpera;
4. Repita o processo até ser encontrada uma solução ou até não haver mais nenhum operador que possa ser aplicado no estado corrente:

   a) selecione um operador que ainda não tenha sido aplicado ao estado corrente e aplique-o para produzir um estado novo;

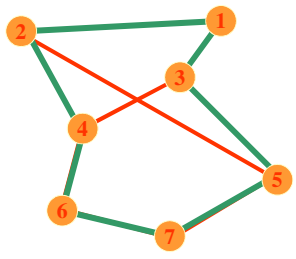
   b) avalie o estado novo. Calcule
   - ΔE = valor do estado corrente - valor do estado novo (mudança de energia)
     - se o estado novo for um estado meta, retorne-o e encerre;
     - se não for um estado meta, mas for melhore que o estado corrente, então torne-o o estado corrente com a probabilidade p´ (definido aleatoriamente um número entre [0,1] se for menor que p´ então o movimento é aceito, senão , não faz nada;

   c) retorne melhor-até-o-momento como resposta;

Problema do Caixeiro Viajante

- Problema do caixeiro viajante ou "Traveling Salesman Problem (TSP)" :
- Encontrar um caminho por um determinado conjunto de cidades de modo que:
  - Cada cidade deve ser visitada apenas uma vez;
  - A distância total percorrida deve ser minimizada;
  - Ou seja, o caixeiro viajante:
    - Sai de casa
    - Visita n cidades
    - Retorna à sua casa
    - Deve visitar cada cidade uma única vez
    - Objetivo: Encontrar o caminho *mais curto*

- O problema do caixeiro viajante



Percorrer todas as cidades, passando uma vez em cada uma, terminando na cidade de partida e com a mínima distância total percorrida

Número de combinações: n!

7! = 5040 possíveis rotas

Rota obtida pelo algoritmo guloso

Melhor rota

Tempo estimado de solução do problema

Função de complexidade	Tamanho da Instância do Problema					
	10	20	30	40	50	60
n	0,00001 segundos	0,00002 segundos	0,00003 segundos	0,00004 segundos	0,00005 segundos	0,00006 segundos
n^2	0,0001 segundos	0,0004 segundos	0,0009 segundos	0,0016 segundos	0,0025 segundos	0,0036 segundos
n^3	0,001 segundos	0,008 segundos	0,027 segundos	0,064 segundos	0,125 segundos	0,216 segundos
n^5	0,1 segundos	3,2 segundos	24,3 segundos	1,7 minutos	5,2 minutos	13,0 minutos
2^n	0,001 segundos	1,0 segundos	17,9 segundos	12,7 dias	35,7 anos	366 séculos
3^n	0,059 segundos	58 minutos	6,5 anos	3855 séculos	2×10^8 séculos	$1,3 \times 10^{13}$ séculos

Fonte: Garey & Johnson

Computador mais rápido resolve?

Maior instância que um computador resolve em 1 hora			
Função de complexidade	Computador Atual	Computador 100x mais rápido	Computador 1000x mais rápido
n	N	100 N	1000 N
n^2	M	10 M	31,6 M
n^3	Z	4,64 Z	10 Z
n^5	W	2,5 W	3,98 W
2^n	X	X + 6,64	X + 9,97
3^n	Y	Y + 4,19	Y + 6,29

Fonte: Garey & Johnson

Metaheurísticas

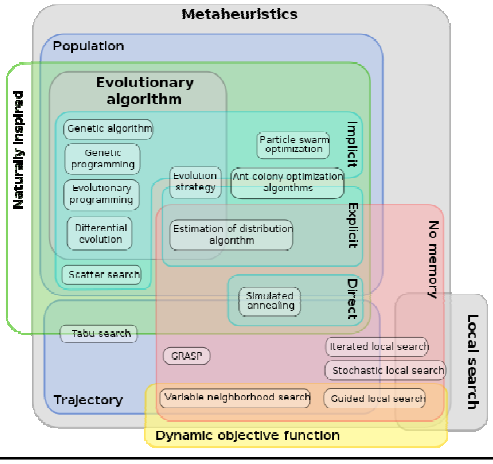
- Heurística é qualquer método desenvolvido para resolver um determinado tipo de problema;
- As metaheurísticas são definidas como heurísticas de uso mais geral ou uma "heurísticas das heurísticas";
- Exemplo de heurísticas:
  - Algoritmos genéticos ;
  - Simulated Annealing;

Metaheurísticas

- O sufixo meta significa a mais alta, ou um nível superior;
- São de uso geral;
- Simples de implementar;
- Possuem mecanismos que possibilitam escapar de soluções de baixa qualidade (ótimo local);
- Baixo tempo computacional;
- São robustas (instâncias de tamanho variado)

Metaheurísticas

- Produzem ótimos globais, ou ótimos locais, de qualidade superior aos ótimos produzidos por uma heurística;
- Iterativamente melhorar um conjunto de soluções
- Pouco conhecimento do problema;
- Precisa poder distinguir boas soluções;
- Geralmente encontra boas soluções possivelmente não o ótimo;
- Adaptáveis: parâmetros ajustáveis



IA