



**FACULDADE DE ENGENHARIA**  
**DEPARTAMENTO DE ELECTROTECNIA**  
**LICENCIATURA EM ENGENHARIA INFORMÁTICA**  
**INTELIGÊNCIA ARTIFICIAL I**

**TPC 7**

**Estudante:**

Lino, Miro Pedro Tipaneque

**Docente:**

Ruben Manhiça

**Resumo de Leitura**

**4.2. Funções Heurísticas**

Funções heurísticas são utilizadas para guiar a busca em direção a soluções mais promissoras, estimando o custo ou a distância de um estado ao objectivo. Elas são centrais em algoritmos como a Busca Gulosa pela melhor escolha e Busca A\*.

Russell e Norvig explicam que heurísticas bem projetadas podem reduzir significativamente o espaço de busca. Heurísticas admissíveis (que nunca superestimam o custo até o objetivo) são especialmente importantes para garantir a optimalidade em A\*. Métodos como relaxamento de problemas e funções heurísticas derivadas de subproblemas são discutidos como estratégias para criar heurísticas eficientes.

### 4.3. Algoritmos de Busca Local e Problemas de Otimização

Busca local é uma abordagem para resolver problemas de otimização e satisfação de restrições onde o foco está em melhorar uma solução inicial, movendo-se incrementalmente em seu espaço de busca. Esses algoritmos, como *Busca de Subida de Encosta*, *Busca de Têmpera Simulada*, *Busca em Feixe Local* e *Algoritmos Genéticos*, são úteis em contextos onde a solução global pode ser encontrada através de otimizações locais. Ao contrário da busca tradicional, esses algoritmos não requerem a expansão de um espaço de busca estruturado, sendo úteis em problemas onde o espaço é grande e não completamente conhecido.

### 4.4. Busca Local em Espaço Contínuo

Esse tipo de busca adapta os métodos de otimização para problemas com variáveis contínuas, utilizando gradientes e outras técnicas matemáticas para ajustar as soluções. Russell e Norvig destacam o uso de métodos como *Gradiente Descendente*, onde se busca minimizar uma função de custo ajustando as variáveis continuamente. Essa abordagem é comum em problemas de ajuste fino de parâmetros, como em aprendizado de máquina e robótica, onde é necessário otimizar funções complexas e não lineares.

### 4.5. Agentes de Busca On-line e Ambientes Desconhecidos

Agentes de busca on-line atuam em ambientes desconhecidos, onde a exploração e aprendizado do ambiente são tão importantes quanto a busca. Ao contrário dos algoritmos de busca clássicos, que requerem um modelo completo do ambiente, os agentes on-line exploram enquanto buscam uma solução, atualizando seu entendimento do ambiente a cada passo.

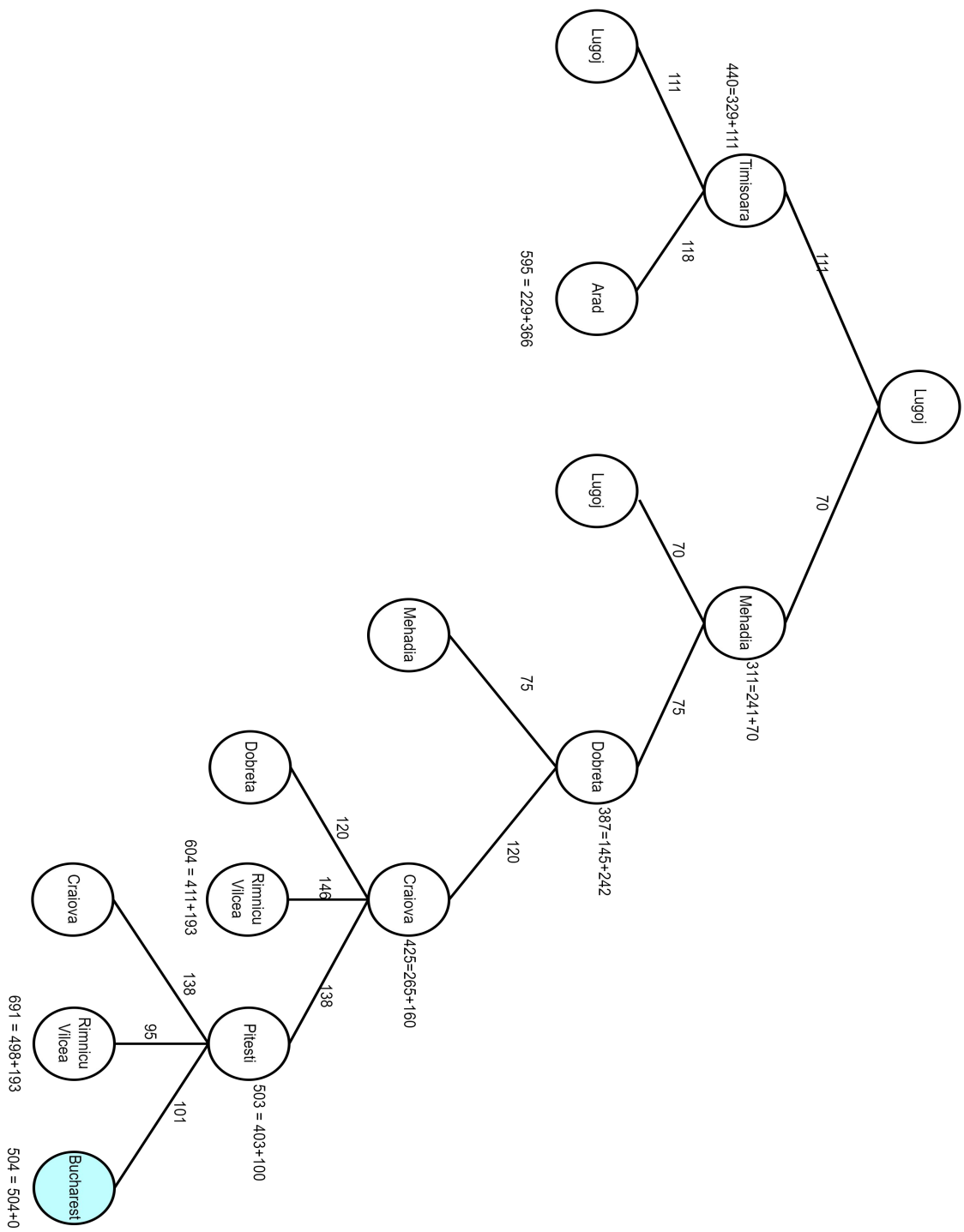
Técnicas como busca gulosa on-line e algoritmos que integram mapeamento e planejamento em tempo real são essenciais para navegar em ambientes dinâmicos e parcialmente observáveis, como exploração de terrenos desconhecidos por robôs.

## Resolução de Exercícios Práticos

4.1

Cidade inicial: Lugoj

Cidade objetivo: Bucharest



Solução: Lugoj, Mehadia, Dobreta, Craiova, Pitesti, Bucharest

Custo acumulado: 504

4.2 O algoritmo de caminho heurístico é uma busca pela melhor escolha na qual a função objectivo é  $f(n) = (2 - w)g(n) + wh(n)$ .

a) Para que valores de  $w$  mais garantia de ser óptimo?

### Resolução

Temos que evitar que o coeficiente de  $g(n)$  seja maior que zero:  $2 - w > 0 \rightarrow w < 2$

Resposta:  $w < 2$

b) Que espécie de busca ele executa quando  $w = 0$ ?

### Resolução

Quando  $w = 0$  teremos  $f(n) = (2 - 0)g(n) + 0 \cdot h(n) \leftrightarrow f(n) = 2g(n)$  que é uma busca em custo uniforme.

c) E quando  $w = 1$

### Resolução

Quando  $w = 1$  teremos  $f(n) = (2 - 1)g(n) + 1 \cdot h(n) \leftrightarrow f(n) = g(n) + h(n)$  que é  $A^*$

d) E quando  $w = 2$

### Resolução

Quando  $w = 2$  teremos  $f(n) = (2 - 2)g(n) + 2h(n) \leftrightarrow f(n) = 2h(n)$  que é a busca gulosa pela melhor escolha

4.3 Prove cada uma das afirmações a seguir

a) A busca em extensão é um caso especial de busca em custo uniforme

Prova:

A busca de custo uniforme é mais geral porque pode lidar com custos de arestas diferentes. No entanto, quando todos os custos são iguais, ela se reduz ao comportamento da busca em extensão.

b) A busca em extensão, a busca em profundidade e a busca de custo uniforme são casos especiais da busca pela melhor escolha.

Prova:

**Busca em extensão:** expande nós camada a camada, explorando todos os nós a uma determinada profundidade antes de explorar nós em profundidades maiores. Isso pode ser visto como um caso especial da busca pela melhor escolha se definirmos  $f(n) = d(n)$ , onde  $d(n)$  é a profundidade do nó.

**Busca em profundidade:** explora cada caminho completamente antes de retornar e explorar caminhos alternativos. Esse comportamento pode ser simulado em uma busca pela melhor escolha definindo  $f(n) = -d(n)$ , onde  $d(n)$  é a profundidade do nó.

**Busca de custo uniforme:** expande o nó com o menor custo acumulado  $g(n)$  a partir do nó inicial até  $n$ . Isso é um caso especial da busca pela melhor escolha ao definir  $f(n) = g(n)$ .

c) A busca de custo uniforme é um caso especial da busca  $A^*$

Prova:

Se a heurística de  $A^*$  for ZERO, o algoritmo se transforma em busca de custo uniforme.

4.5 Vimos na página 95 que a heurística de distancia em linha recta leva a busca gulosa pela melhor escolha a se perder no problema de ir de Iasi ate Fagaras. Porem, a heurística é perfeita no problema oposto: ir de Fagaras ate Iasi. Existem problemas para os quais a heurística é falha em ambos os sentidos?

**Resolução:**

Sim, existem.

Na busca gulosa, a heurística  $h(n)$  é usada para guiar a exploração, sempre expandindo o nó que parece mais promissor com base no valor mais baixo de  $h(n)$ . Se a heurística não é confiável, a busca pode se desviar e perder soluções óptimas.

## 4.11

Caso especial	Algoritmo resultante
Busca em feixe local com $k = 1$	Subida de encosta
Busca em feixe local com $k = \infty$	Busca em largura
Têmpera simulada com $T = 0$	Subida de encosta
Algoritmo genético com $N = 1$	Subida de encosta