



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Quadern de treball: Cerca en cost uniforme Algorisme de Dijkstra¹

Albert Sanchis

DSIC

Departament de Sistemes
Informàtics i Computació

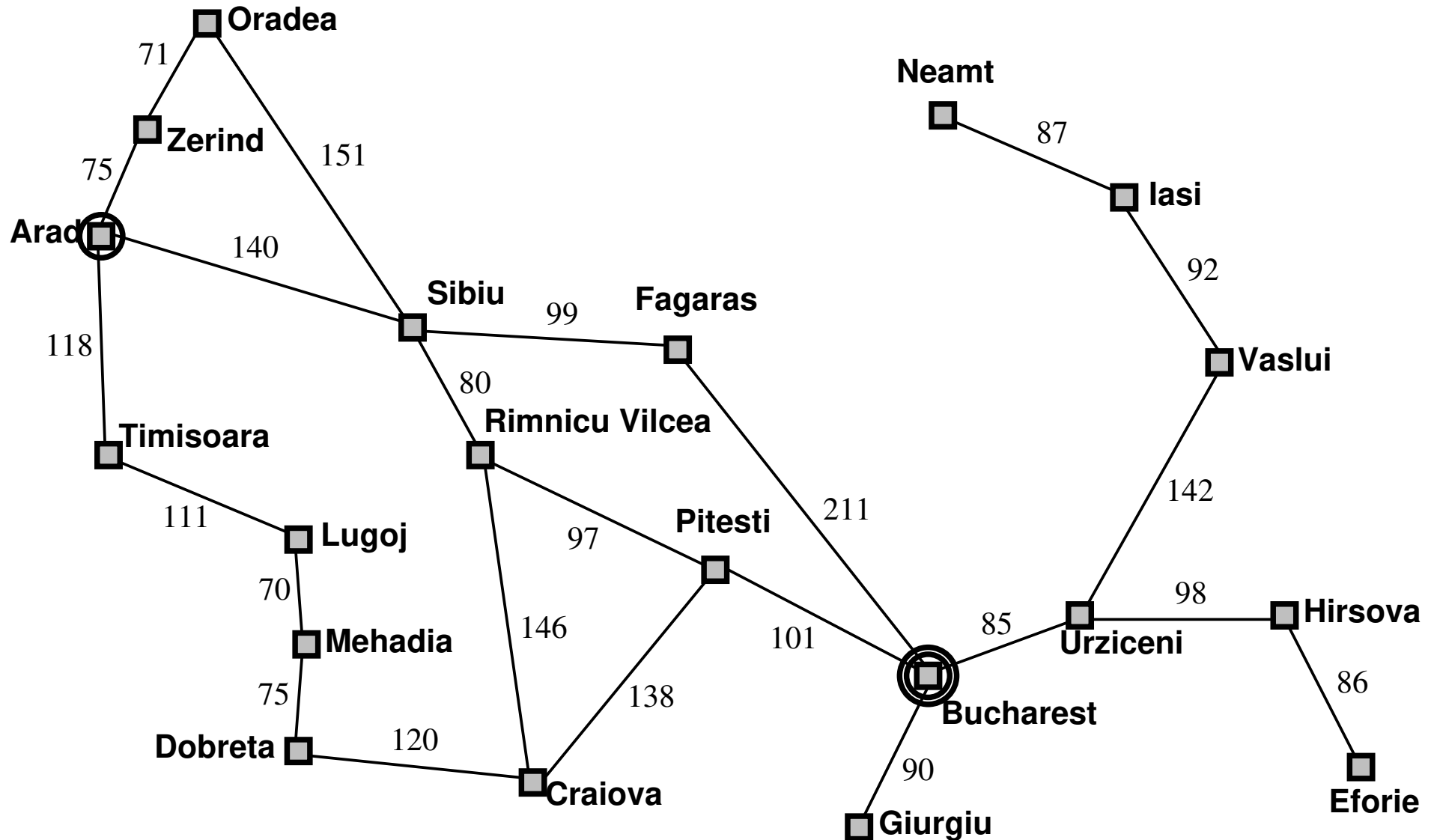
¹Per a una correcta visualització, es requereix l'Acrobat Reader v. 7.0 o superior

Objectius formatius

- ▶ Caracteritzar la cerca convencional en un graf d'estats.
- ▶ Descriure cerca en cost uniforme (o algorisme de Dijkstra).
- ▶ Construir l'arbre de cerca en cost uniforme.
- ▶ Aplicar cerca en cost uniforme a un problema clàssic.
- ▶ Analitzar la qualitat de cerca en cost uniforme.

Problema: La ruta més curta entre dos punts

Cerca d'una ruta més curta des d'Arad a Bucarest [1]:



$\text{Accions}(\text{Arad}) = \{\text{Anar}(\text{Sibiu}), \text{Anar}(\text{Timisoara}), \text{Anar}(\text{Zerind})\}.$

Cost uniforme o algorisme de Dijkstra [1, 2, 3]

```
UCS( $G, s'$ )           // Uniform-cost search;  $G$  graf ponderat,  $s'$  start  
   $O = \text{IniCua}(s', g_{s'} \triangleq 0)$            // Open: cua de prioritat  $g$   
   $C = \emptyset$                                // Closed: nodes explorats  
  mentre no  $\text{CuaBuida}(O)$ :                 // 1r el millor:  $s = \arg \min_{n \in O} g_n$   
     $s = \text{Desencua}(O)$                        // desempats a favor d'objectius  
    si  $\text{Objectiu}(s)$  retorna  $s$                 // solució trobada!  
     $C = C \cup \{s\}$                            //  $s$  explorat  
    per a tota  $(s, n) \in \text{Adjacents}(G, s)$ :    // generació:  $n$  fill d' $s$   
       $x = g_s + w(s, n)$                        // cost del camí d' $s'$  a  $n$  passant per  $s$   
      si  $n \notin C \cup O$ :  $\text{Encua}(O, n, g_n \triangleq x)$   
      si no si  $n \in O$  i  $x < g_n$ :  $\text{Modcua}(O, n, g_n \triangleq x)$   
  retorna NULL                               // cap solució trobada
```

- **Qüestió 1:** Fes una traça de l'algorisme **Cost Uniforme** aplicat al problema de cerca d'una ruta més curta des d'Arad a Bucarest.

O	C	s
{Arad (c=0)}	{}	—
{Zerind (c=75), Timisoara (c=118), Sibiu (c=140)}	{Arad (c=0)}	Arad (c=0)
{Timisoara (c=118), Sibiu (c=140), Oradea (c=146)}	{Arad (c=0), Zerind (c=75)}	Zerind (c=75)
{Sibiu (c=140), Oradea (c=146), Lugoj (c=229)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118)}	Timisoara (c=118)
{Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140)}	Sibiu (c=140)
{Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146)}	Oradea (c=146)
{Lugoj (c=229), Fagaras (c=239), Pitesti (c=317), Craiova (c=366)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220)}	Rimnicu (c=220)
{Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229)}	Lugoj (c=229)
{Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Bucharest (c=450)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239)}	Fagaras (c=239)

O	C	s
{Pitesti (c=317), Craiova (c=366), Dobreta (c=374), Bucharest (c=450)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299)}	Mehadia (c=299)
{Craiova (c=366), Dobreta (c=374), Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317)}	Pitesti (c=317)
{Dobreta (c=374), Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366)}	Craiova (c=366)
{Bucharest (c=418)}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Dobreta (c=374)}	Dobreta (c=374)
{}	{Arad (c=0), Zerind (c=75), Timisoara (c=118), Sibiu (c=140), Oradea (c=146), Rimnicu (c=220), Lugoj (c=229), Fagaras (c=239), Mehadia (c=299), Pitesti (c=317), Craiova (c=366), Dobreta (c=374)}	Bucharest (c=418)

- **Qüestió 2:** Construeix l'arbre de cerca resultant d'aplicar l'algorisme **Cost Uniforme** al problema de cerca d'una ruta més curta des d'Arad a Bucarest.

- ▶ **Qüestió 3:** L'algorisme troba solució? **Sí**
- ▶ **Qüestió 4:** Si la resposta es “Sí”:
 - ▷ Quina ha sigut la solució trobada? ***El camí solució trobat ha sigut: Arad, Sibiu, Rimnicu, Pitesti, Bucharest***
 - ▷ Quin és el cost d'aquesta solució? **418**
 - ▷ Es tracta de la solució òptima? **Sí**
 - ▷ Quin tipus de solució troba l'algorisme cost uniforme? ***Solucions òptimes si els costos de les accions són positius***

Referències

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.
- [2] E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1959.
- [3] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2018.