



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Quadern de treball: Cerca en profunditat iterativa¹

Albert Sanchis

DSIC

Departament de Sistemes
Informàtics i Computació

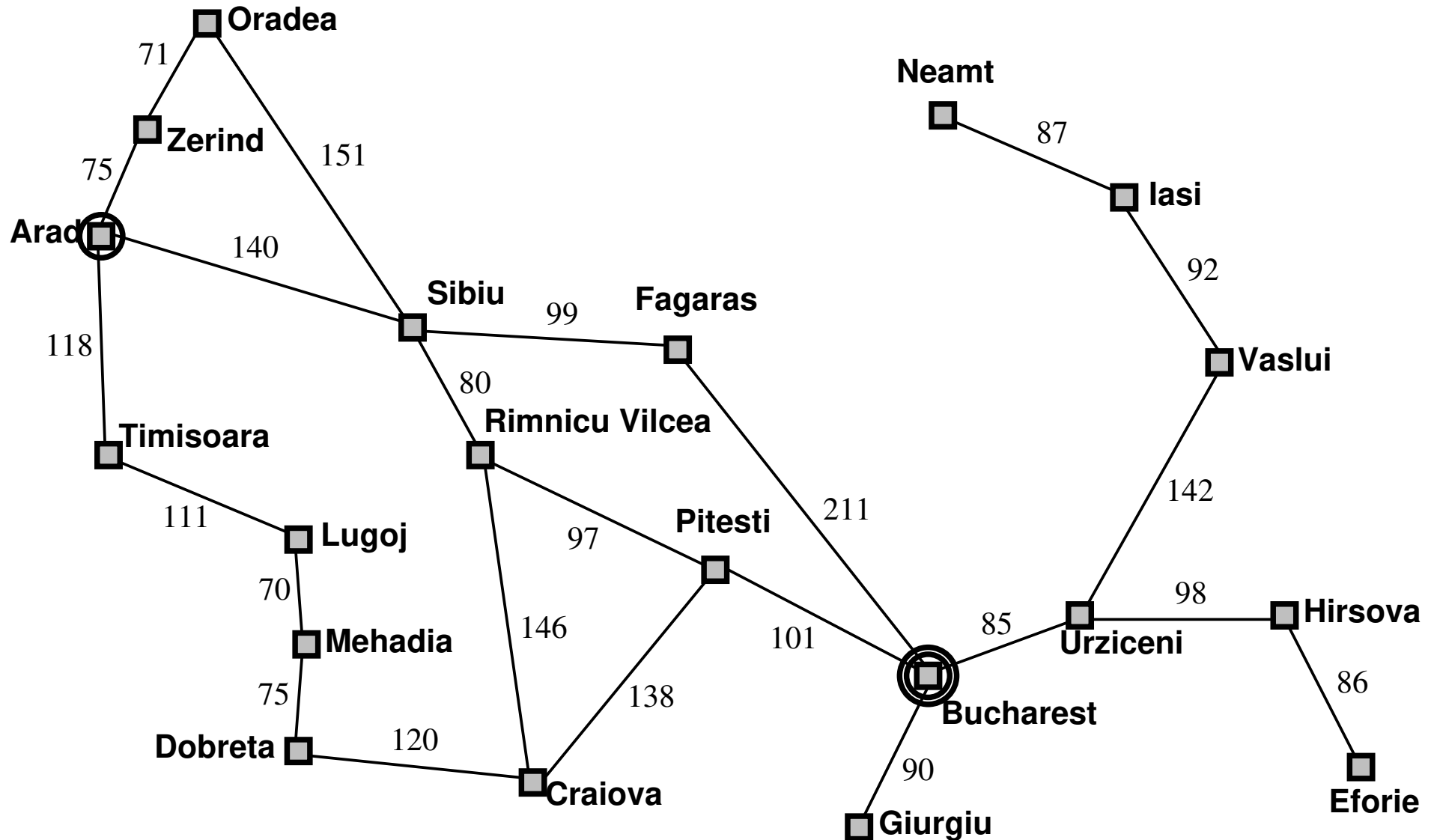
¹Per a una correcta visualització, es requereix l'Acrobat Reader v. 7.0 o superior

Objectius formatius

- ▶ Caracteritzar la cerca convencional en un graf d'estats.
- ▶ Descriure cerca en profunditat iterativa.
- ▶ Construir l'arbre de cerca en profundita iterativa.
- ▶ Aplicar cerca en profunditat iterativa a un problema clàssic.
- ▶ Analitzar la qualitat de cerca en profunditat iterativa.

Problema: La ruta més curta entre dos punts

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



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

Cerca en profunditat iterativa [2]

PI(G, s) // *Profunditat Iterativa*

per a $m = 0, 1, 2, \dots$: **si** ($r = \text{DFS}(G, s, m)$) $\neq \text{NULL}$: **retorna** r

DFS(G, s', m) // *Depth-first search* amb profunditat màxima m

$O = \text{IniPila}(s')$ // *Open: frontera-pila* de la cerca

mentre no $\text{PilaBuida}(O)$:

$s = \text{Desapila}(O)$ // selecció *LIFO (Last in, first out)*

si $\text{Objectiu}(s)$ **retorna** s // solució trobada!

si $\text{Profunditat}(s) < m$: // no a profunditat màxima

per a tota $(s, n) \in \text{Adjacents}(G, s)$: // generació: n fill d' s

$\text{Apila}(O, n)$ // afegim n a la pila

retorna NULL // cap solució trobada

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

Amb $m = 0$

O	S
{Arad (p=0)}	—
{}	Arad (p=0)

Amb $m = 1$

O	S
{Arad (p=0)}	—
{Sibiu (p=1), Timisoara (p=1), Zerind (p=1)}	Arad (p=0)
{Timisoara (p=1), Zerind (p=1)}	Sibiu (p=1)
{Zerind (p=1)}	Timisoara (p=1)
{}	Zerind (p=1)

Amb $m = 2$

O	S
{Arad (p=0)}	—
{Sibiu (p=1), Timisoara (p=1), Zerind (p=1)}	Arad (p=0)
{Arad (p=2), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Sibiu (p=1)
{Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Arad (p=2)
{Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Fagaras (p=2)
{Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Oradea (p=2)
{Timisoara (p=1), Zerind (p=1)}	Rimnicu (p=2)
{Arad (p=2), Lugoj (p=2), Zerind (p=1)}	Timisoara (p=1)
{Lugoj (p=2), Zerind (p=1)}	Arad (p=2)
{Zerind (p=1)}	Lugoj (p=2)
{Arad (p=2), Oradea (p=2)}	Zerind (p=1)
{Oradea (p=2)}	Arad (p=2)
{}	Oradea (p=2)

Amb $m = 3$

O	s
{Arad (p=0)}	—
{Sibiu (p=1), Timisoara (p=1), Zerind (p=1)}	Arad (p=0)
{Arad (p=2), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Sibiu (p=1)
{Sibiu (p=3), Timisoara (p=3), Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Arad (p=2)
{Timisoara (p=3), Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Sibiu (p=3)
{Zerind (p=3), Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Timisoara (p=3)
{Fagaras (p=2), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Zerind (p=3)
{Bucharest (p=3), Sibiu (p=3), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Fagaras (p=2)
{Sibiu (p=3), Oradea (p=2), Rimnicu(p=2), Timisoara (p=1), Zerind (p=1)}	Bucharest (p=3)

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

Amb $m = 0$

Amb $m = 1$

Amb $m = 2$

Amb $m = 3$

- ▶ **Qüestió 3:** L'algorisme troba solució? **Sí**
- ▶ **Qüestió 4:** Si la resposta es "Sí":
 - ▷ Quantes iteracions emprant l'algorisme DFS ha necessitat fins a trobar la solució? **Quatre**
 - ▷ De què depèn el nombre d'iteracions que necessita l'algorisme per a trobar la solució? **De la profunditat a la qual es trobe la solució més curta en nombre d'accions**
 - ▷ Quina ha sigut la solució trobada? **El camí solució trobat ha sigut: Arad, Sibiu, Fagaras, Bucharest**
 - ▷ Quin és el cost d'aquesta solució? **450**
 - ▷ Es tracta de la solució òptima? **No, perquè hi ha una altra solució amb un menor cost de 418: Arad, Sibiu, Rimnicu, Pitesti, Bucharest**
 - ▷ Quin tipus de solució troba l'algorisme profunditat iterativa? **Aplicant un recorregut en profunditat troba la solució que es troba a menys profunditat en l'arbre de cerca**

Referències

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.
- [2] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 1985.