



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Quadern de treball: Recursive Best First Search (RBFS)¹

Albert Sanchis

DSIC

Departament de Sistemes
Informàtics i Computació

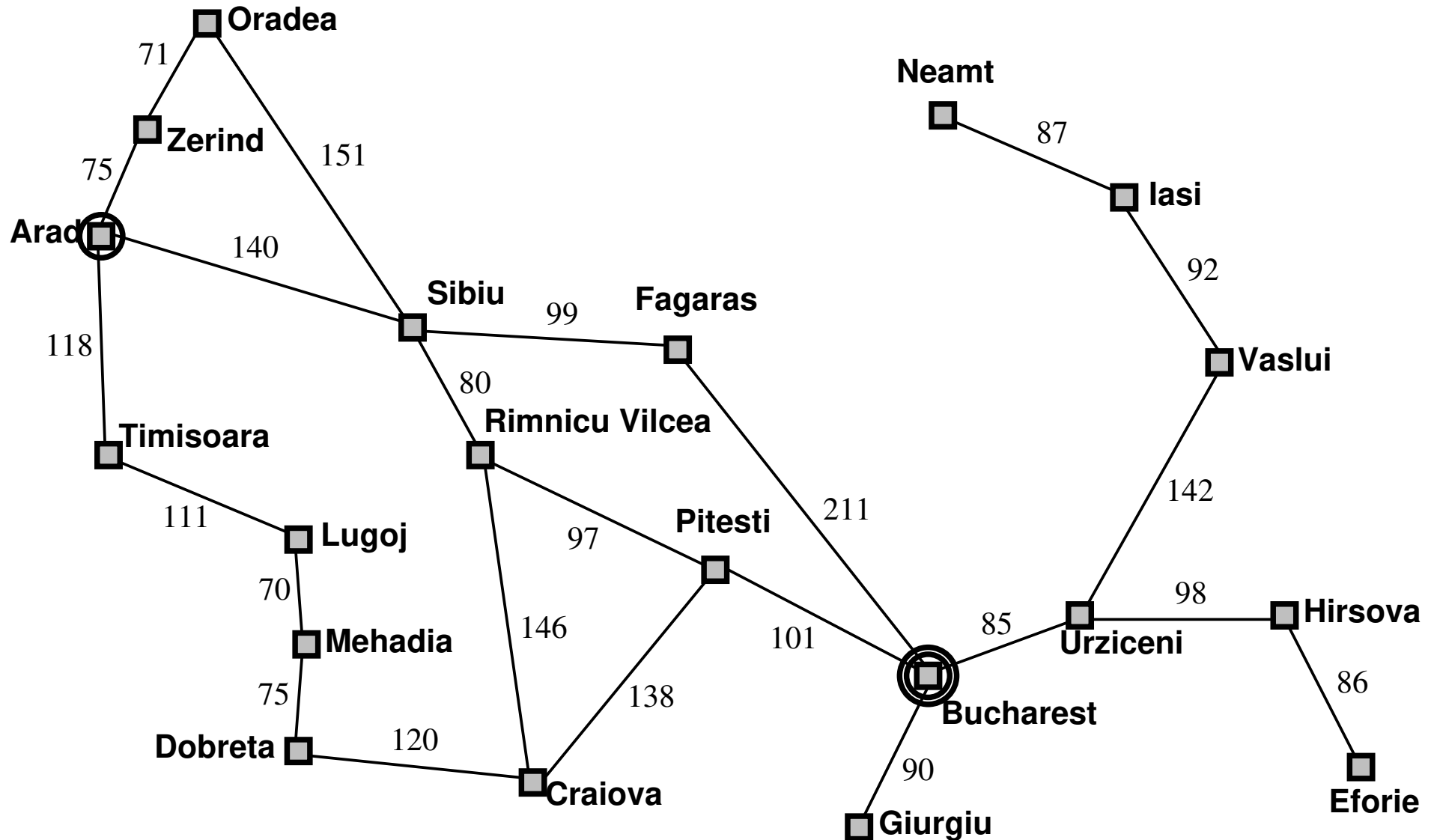
¹Per a una correcta visualització, es requereix l'Acrobat Reader v. 7.0 o superior

Objectius formatius

- ▶ Caracteritzar la cerca convencional en un graf d'estats.
- ▶ Descriure cerca *Recursive Best First Search (RBFS)*.
- ▶ Construir l'arbre de cerca *RBFS*.
- ▶ Aplicar cerca *RBFS* a un problema clàssic.
- ▶ Analitzar la qualitat de cerca *RBFS*.

Problema: La ruta més curta entre dos punts

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



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

Problema: La ruta més curta entre dos punts

Distàncies en línia recta a Bucharest

	Bucharest		Bucharest
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

L'algorisme RBFS (main) [2]

```
RBFS( $G, s', f$ )           //  $G$  graf ponderat,  $s'$  inici,  $f$  func. aval.  
   $P = InitStack(s')$        // Inicialitza Path amb el node arrel  
   $b = \infty$                // Límit inicial  
   $F_{s'} = f_{s'}$           // El valor emmagatzemat és inicialitzat al valor  $f$   
   $(F_r, r) = \mathbf{BT}(G, P, F_{s'}, f, b)$  // Torna v. emmagatzemat i est. obj.  
  if  $r \neq \text{NULL}$ : return  $P$  // Si solució, torna Path a l'objectiu
```

L'algorisme RBFS (backtracking) [2]

```
BT( $G, P, F_s, f, b$ )           //  $G$  graf,  $P$  Path, Valor emmagatzemat  $F_s$ ,  $f$ ,  $b$  límit  
     $s = \text{Top}(P)$                 // Path: extraure cim de la pila  
    if  $\text{Goal}(s)$ : return ( $f_s, s$ )           // Solució trobada!  
     $O = \text{InitQueue}()$            // Open: cua de prioritat per a nodes fill  
    for all  $(s, n) \in \text{Adjacents}(G, s)$  and  $n \notin P$ : // Generant fills  $n$  no en Path  
        if  $f_s < F_s$ :  $F_n = \max(f_n, F_s)$  // Si  $s$  visitat, el fill hereta el v. emmagatzemat  
        else:  $F_n = f_n$                 // En altre cas, el v. emmagatzemat és  $f$   
         $\text{Push}(O, n, F_n)$  // Fills ordenats en cua de prioritat per v. emmagatzemats  
    if  $\text{EmptyQueue}(O)$ : return ( $\infty, \text{NULL}$ ) // No fills, límit =  $\infty$   
    while True:  
         $(n, F_n) = \text{Top}(O)$  // Millor fill en funció del valor emmagatzemat  $F$   
        if  $F_n > b$ : return ( $F_n, \text{NULL}$ ) // S'excedeix el límit, backtracking  
         $(n', F_{n'}) = \text{Top2}(O)$  // 2-millor F o si no existeix, llavors  $F_{n'} = \infty$   
         $\text{Push}(P, n)$  // Afegir fill al Path explorat  
         $(F_n, r) = \text{BT}(G, P, F_n, f, \min(b, F_{n'}))$  // Recursió amb possible nou límit  
        if  $r \neq \text{NULL}$ : return ( $F_n, r$ ) // Si solució, fí recursió sense actualització  
         $\text{Update}(O, n, F_n)$  // Actualitzar node  $n$  en  $O$   
         $\text{Pop}(P)$  // Descartar últim fill del Path
```

- ▶ **Qüestió 1:** Construeix l'arbre de cerca resultant d'aplicar l'algorisme **RBFS** al problema de cerca d'una ruta més curta des d'Arad a Bucarest.

- ▶ **Qüestió 2:** L'algorisme troba solució? Si la resposta es “Sí”:
 - ▷ Quina ha sigut la solució trobada?
 - ▷ Quin és el cost d'aquesta solució?
 - ▷ Es tracta de la solució òptima?

Referències

- [1] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, third edition, 2010.
- [2] Richard E. Korf. Linear-space best-first search. *Artificial Intelligence*, 62(1):41–78, 1993.