



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content
pentru învățământul superior tehnic

Proiectarea Algoritmilor

29. Algoritmul Best First

Bibliografie

[1] C. Giumale – Introducere in Analiza Algoritmilor - cap. 7

[2] <http://www.gamasutra.com/features/19990212/pathdemo.zip>

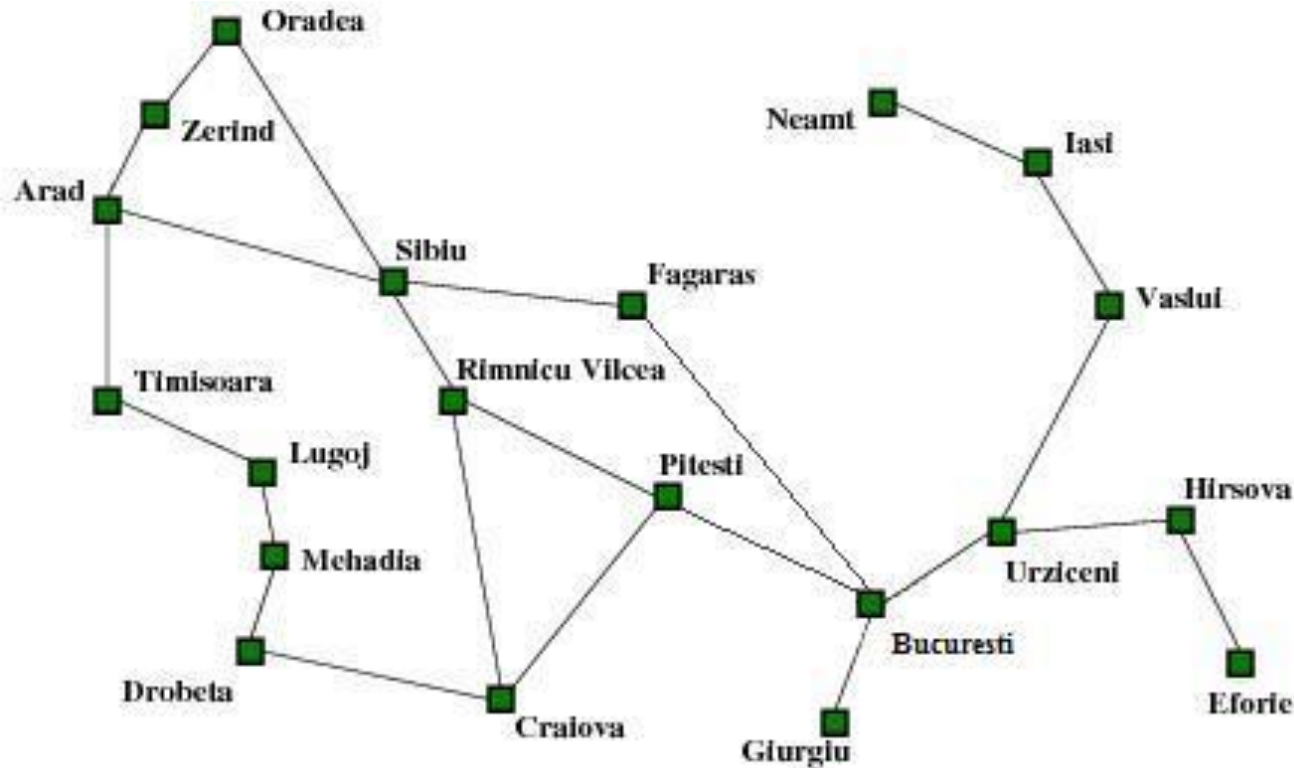
[3] <http://www.policyalmanac.org/games/aStarTutorial.htm>

[4] <http://www.ai.mit.edu/courses/6.034b/searchcomplex.pdf>

Detalii generale

- Păstrează toate nodurile de pe frontieră (**OPEN**), unii păstrând și nodurile expandate (**CLOSED**).
- Fiecare nod are un cost asociat $f(n) \geq 0$ care **estimează calitatea nodului** (distanța de la nodul respectiv până la un nod soluție).
- Cu cât **$f(n)$ este mai mic**, cu atât **nodul este mai bun**.

Prezentarea problemei



- Trebuie să ajungem în București din diverse puncte ale țării pe ruta cea mai scurtă.

Explorare lacomă

- Explorare_lacomă (StInit, f, test_sol)

- nod = constr_nod(StInit); // starea inițială
- $\pi(\text{nod}) = \text{null}$;
- OPEN = {nod};

Inițializări

- **Cât timp** (OPEN $\neq \emptyset$) // mai am noduri de prelucrat

- nod = selecție_nod (OPEN); // $f(\text{nod}) = \min \{f(n) \mid n \in \text{OPEN}\}$

- **Dacă** (test_sol(nod)) **Întoarce** nod;

Soluția

- OPEN = OPEN \setminus {nod}; // nodul nu e soluție, trebuie expandat

- succs = expand(nod); // expandare nod

- **Pentru fiecare** (succ \in succs)

- { OPEN = OPEN \cup {succ}; $\pi(\text{succ}) = \text{nod}$; } // actualizare succesori

Continui căutarea

- **Întoarce** insucces; **Insucces**

Optimalitate?

Completitudine?



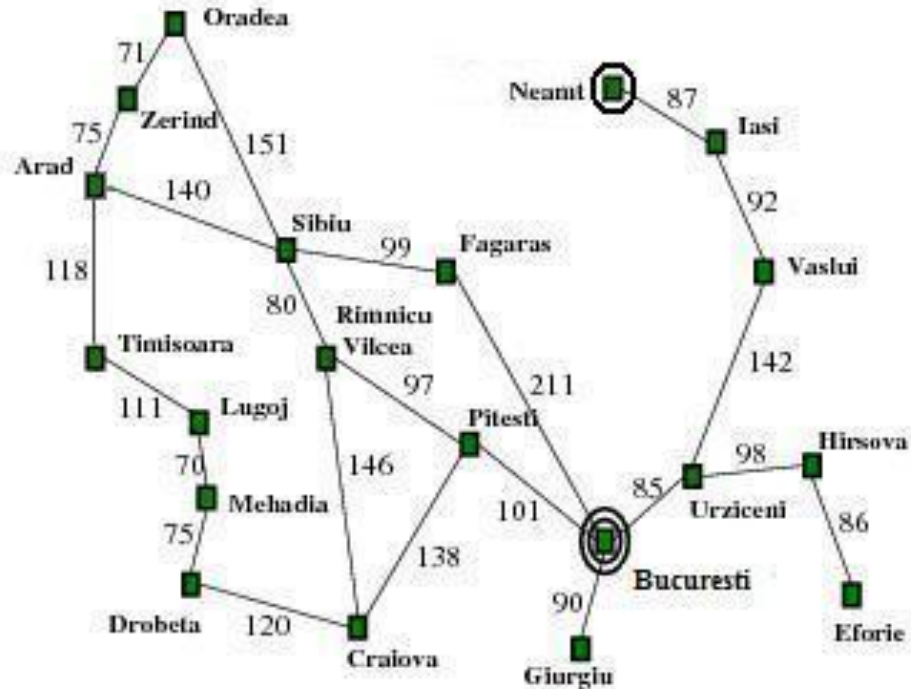
Problema?

$f(\text{nod}) = \text{distanța de la nodul curent până la nodul nod}$

$f(\text{Iași}) = 87 \rightarrow \text{Iași}$

$f(\text{Neamț}) = 87$

$f(\text{Vaslui}) = 92 \rightarrow \text{Neamț}$



- Drumul Neamț-București? \rightarrow **nu se termină algoritmul!**
- \rightarrow Explorarea lacomă **nu e completă** \rightarrow trebuie să se rețină teritoriul deja parcurs ca să se **evite ciclurile!**

Explorare tentativă completă BF* (BEST FIRST) (1)

- BF*(Stlnit, f, test sol)

- `nod = constr_nod(Stlnit); // starea inițială`
- `$\pi(\text{nod}) = \text{null};$`
- `OPEN = {nod}; // noduri explorate dar neexpandate`
- `CLOSED = \emptyset ; // noduri expandate` **Inițializări**

- **Cât timp** (`OPEN $\neq \emptyset$`)

- `nod = selecție nod (OPEN); // $f(\text{nod}) = \min \{f(n) \mid n \in \text{OPEN}\}$`

- **Dacă** (`test_sol(nod)`) **Întoarce** nod;

Soluția

- `OPEN = OPEN \ {nod};`
- `CLOSED = CLOSED U {nod};`
- `succs = expand(nod);`

Continuarea căutării

Explorare tentativă completă BF* (BEST FIRST) (2)

- **Pentru fiecare** ($\text{succ} \in \text{SUCCS}$)

- **Dacă** ($\text{succ} \notin \text{CLOSED} \cup \text{OPEN}$) **atunci**
 - $\{ \text{OPEN} = \text{OPEN} \cup \{\text{succ}\}; \pi(\text{succ}) = \text{nod}; \}$

Nod nou

- **Altfel**

- $\text{succ}' = \text{apariția lui succ în CLOSED} \cup \text{OPEN}$
- **Dacă** ($f(\text{succ}) < f(\text{succ}')$) // am găsit o cale mai bună către succ și
// redeschidem nodul

$\pi(\text{succ}') = \text{nod};$ // actualizez părintele

$f(\text{succ}') = f(\text{succ});$ // și costul nodului

Actualizări

Reprelucrare

Dacă ($\text{succ}' \in \text{CLOSED}$) // dacă era considerat expandat, îl redeschid

$\{ \text{CLOSED} = \text{CLOSED} \setminus \{\text{succ}'\}; \text{OPEN} = \text{OPEN} \cup \{\text{succ}'\}; \}$

- **Întoarce** insucces;
Insucces

Optimalitate?

Completitudine?

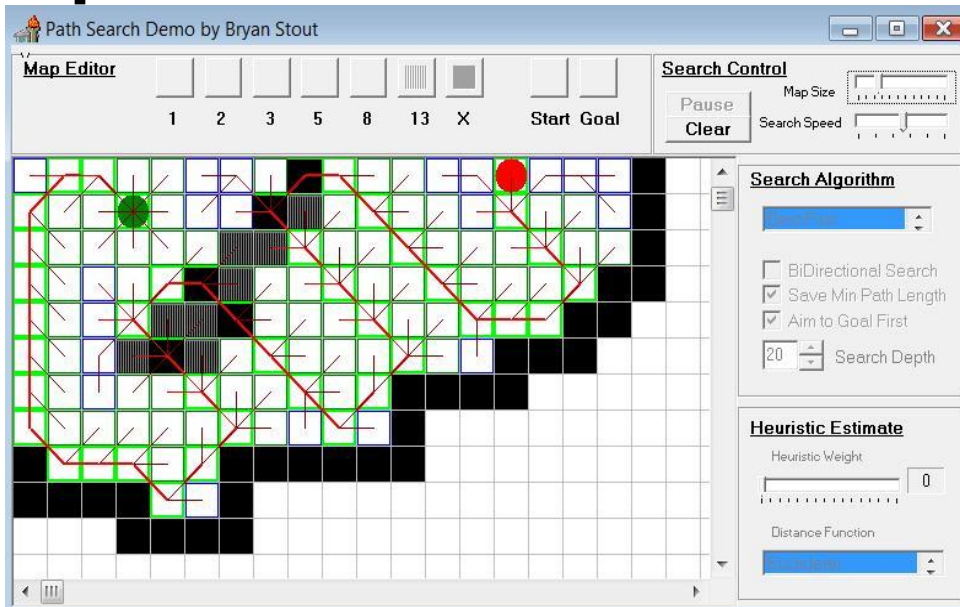
Complexitate?

ex: Best-first cu diverse euristici



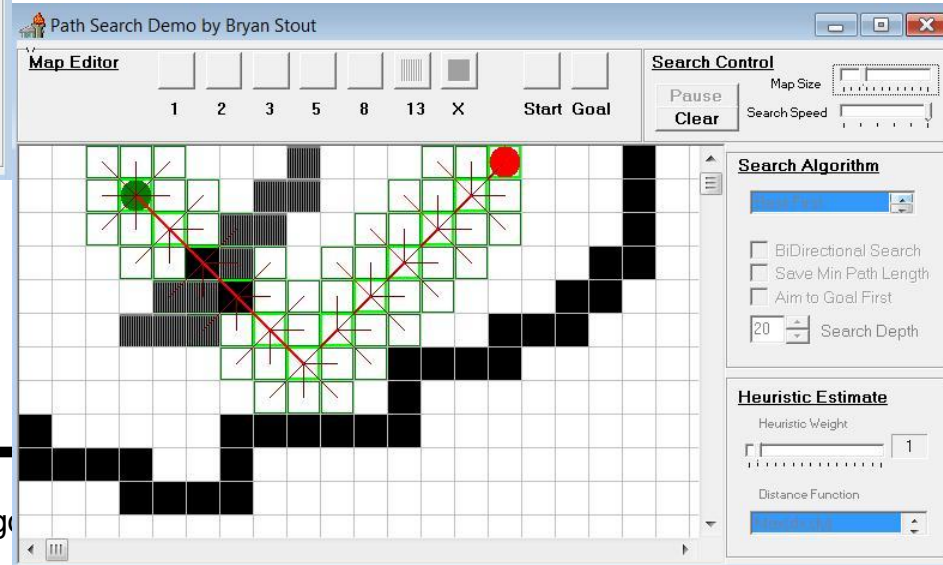
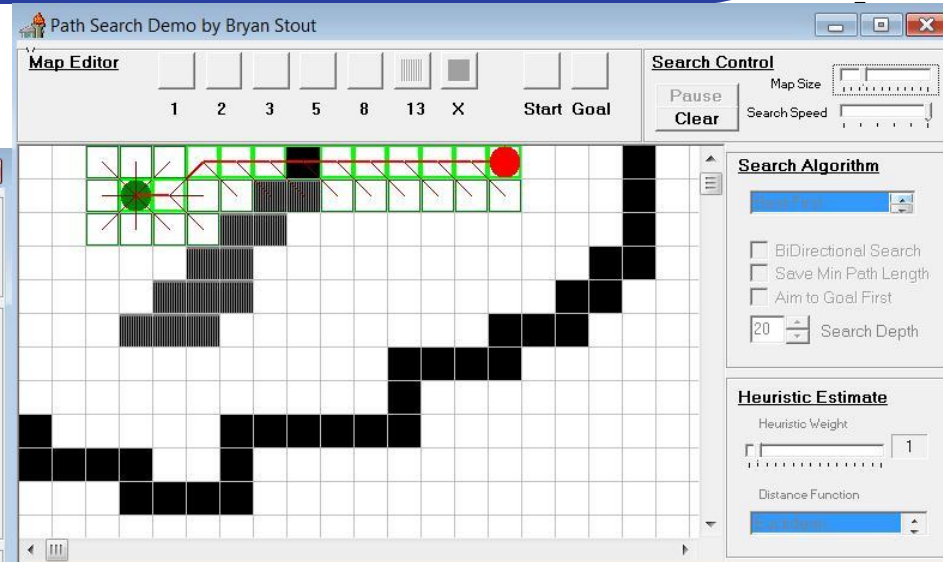
Exemple rulare BF* cu diferite euristici

BF* - Euristică Distanța Euclidiană (11 pași, cost 23) →



BF* fără euristici ↑

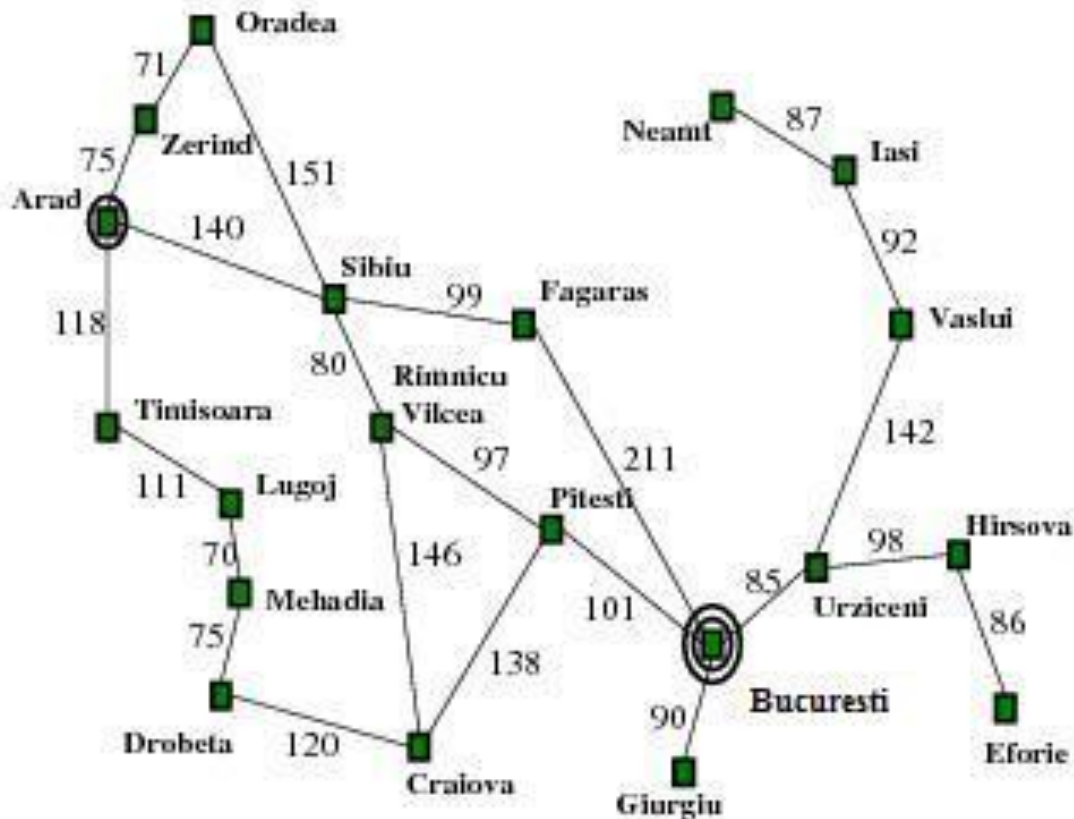
BF* - Euristică maximul distanței pe x și pe y (11 pași, cost 35) →



BF* - completitudine, optimalitate și complexitate

- Păstrează întreg teritoriul explorat:
 - OPEN – nodurile de pe frontieră;
 - CLOSED – nodurile expandate (unele noduri pot fi redeschise) → se evită ciclurile.
- Algoritmul **este complet dar nu este optim**
→ optimalitatea depinde de euristica f !
- **Complexitate: $O(b^{d+1})$**

Aplicație BF*



Distanța în linie dreaptă
pană la București

Arad	366
Craiova	160
Drobeta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- Drumul optim Arad-București ($f(\text{nod}) = \text{distanța în linie dreaptă până la București}$)