



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



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

**Proiectarea Algoritmilor**

**19. Algoritmii lui Johnson**

# Bibliografie

- [1] R. Sedgewick, K. Wayne - Algorithms and Data Structures Fall 2007 – Curs Princeton - <http://www.cs.princeton.edu/~rs/AlgsDS07/>
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*,

# Algoritmul lui Johnson

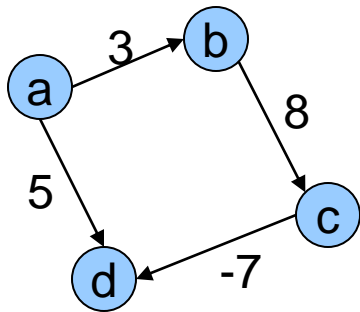
- Pentru grafuri rare.
- Folosește liste de adiacență.
- Bazat pe Dijkstra și Bellman-Ford.
- Complexitate:  $O(V^2 \log V + VE)$ 
  - mai bună decât Floyd-Warshall pentru grafuri rare.

# Idee algoritm Johnson

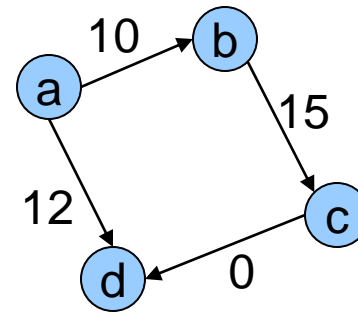
- Dacă graful are **numai muchii pozitive**:
  - se aplică **Dijkstra pentru fiecare nod**  $\Rightarrow$  cost  $V^2 \log V + VE$ .
- Altfel **se calculează costuri pozitive pentru fiecare muchie menținând proprietățile**:
  - $w_1(u,v) \geq 0, \forall (u,v) \in E$ ;
  - $p$  este drum minim utilizând  $w \Leftrightarrow p$  este drum minim utilizând  $w_1$ .

# Construcție $w_1$ (I)

- **Idee 1:** identificare muchia cu cel mai mic cost –  $c$ ; adunare la costul fiecărei muchii valoarea  $c$ ;



$$\text{cost}(a..b..d) < \text{cost}(a,d)$$



$$\text{cost}(a..b..d) > \text{cost}(a,d)$$

**Nu funcționează!!!!**

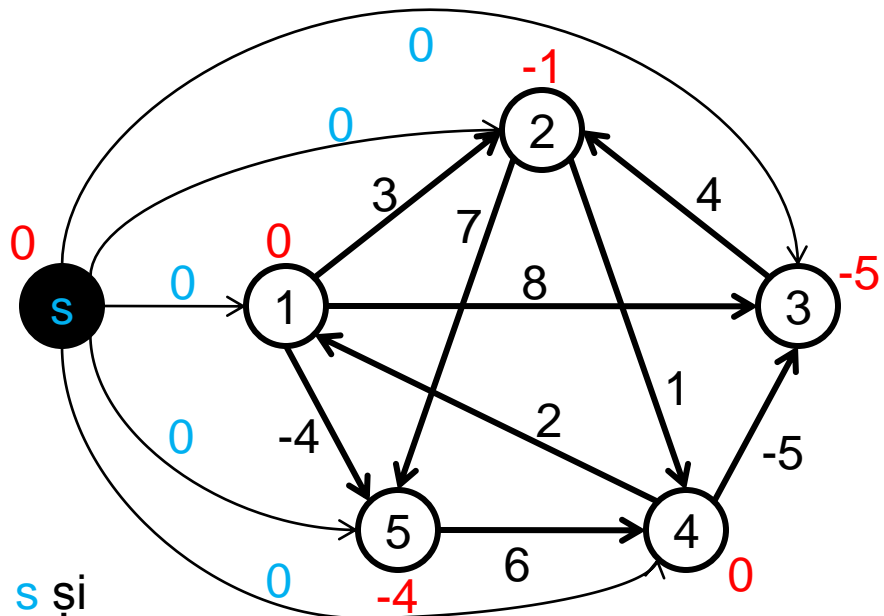
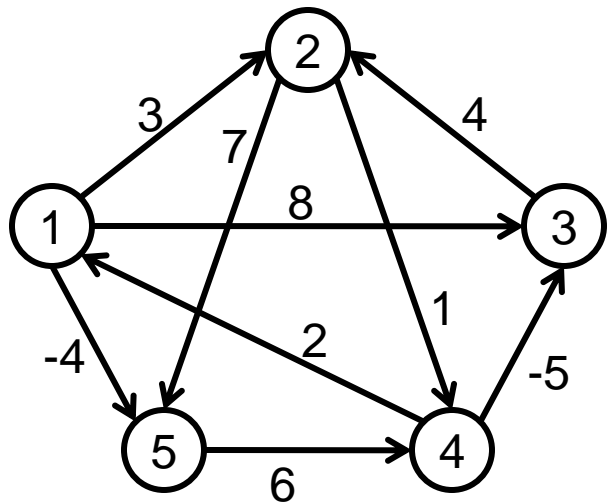
# Construcție $w_1$ (II)

- **Idee 2:**  $w_1(u..v) = w(u..v) + h(u) - h(v)$ ;
- unde  $h : V \rightarrow \mathcal{R}$ ;
- se adaugă un nod  $s$ ;
- se unește  $s$  cu toate nodurile grafului prin muchii de cost 0;
- se aplică BF pe acest graf  $\Rightarrow h(v) = \delta(s,v)$ ;
- $\rightarrow w_1(u,v) = w(u,v) + h(u) - h(v)$ .

# Algoritm Johnson

- Johnson(G)
  - $G' = (V', E')$ ;
  - $V' = V \cup \{s\}$ ; // adăugăm nodul s
  - $E' = E \cup (s, u), \forall u \in V; w(s, u) = 0$ ; // și îl legăm de toate nodurile
  - Dacă BF( $G'$ ) e fals // aplic BF pe  $G'$ 
    - Eroare “ciclu negativ”
  - Altfel
    - Pentru fiecare  $v \in V$ 
      - $h(v) = \delta(s, v)$ ; // calculat prin BF
    - Pentru fiecare  $(u, v) \in E$ 
      - $w_1(u, v) = w(u, v) + h(u) - h(v)$  // calculez noile costuri pozitive
    - Pentru fiecare  $(u \in V)$ 
      - Dijkstra( $G, w_1, u$ ) // aplic Dijkstra pentru fiecare nod
      - Pentru fiecare  $(v \in V)$ 
        - $d(u, v) = \delta_1(u, v) + h(v) - h(u)$  // calculez costurile pe graful inițial

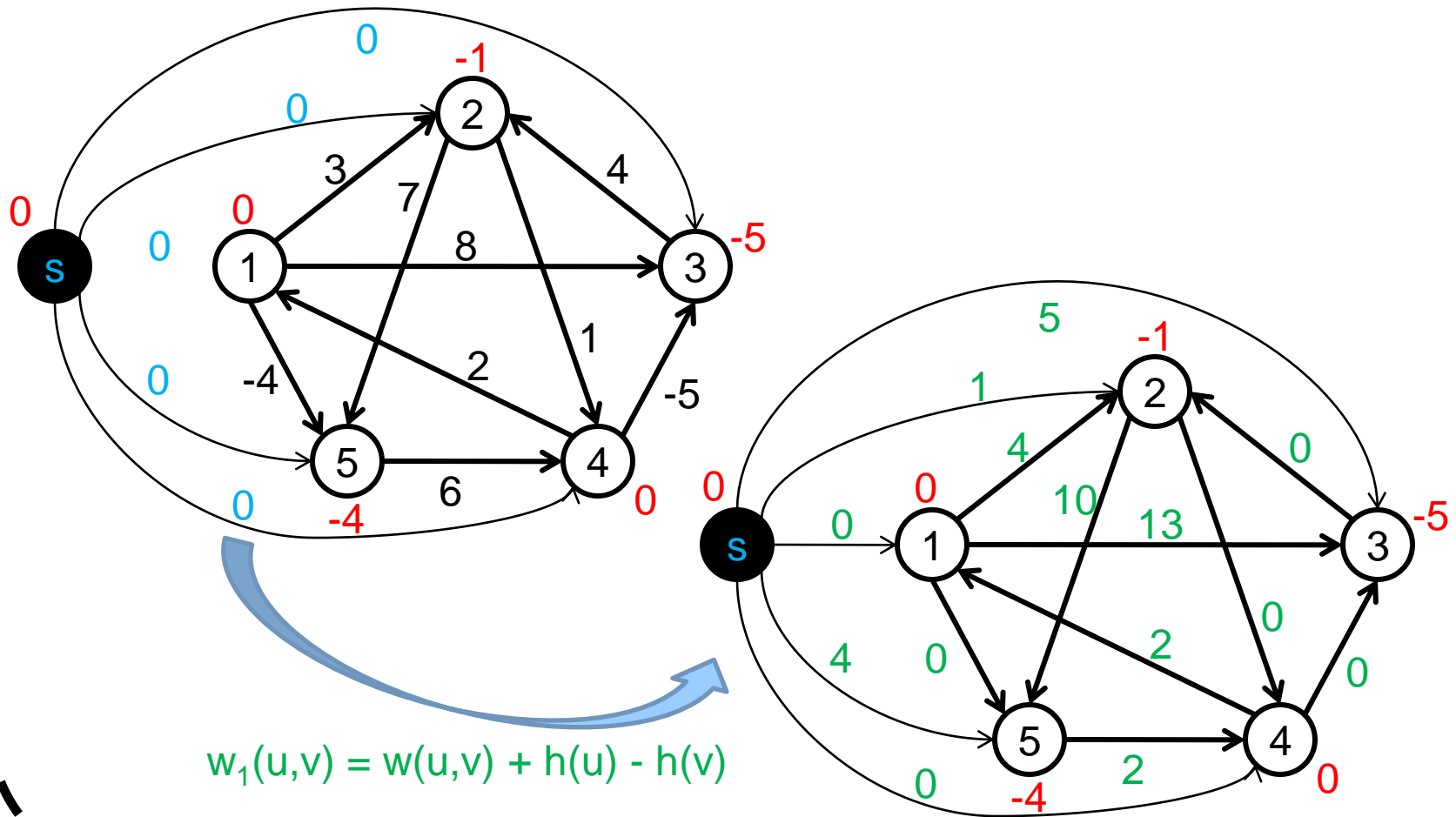
# Exemplu (I)



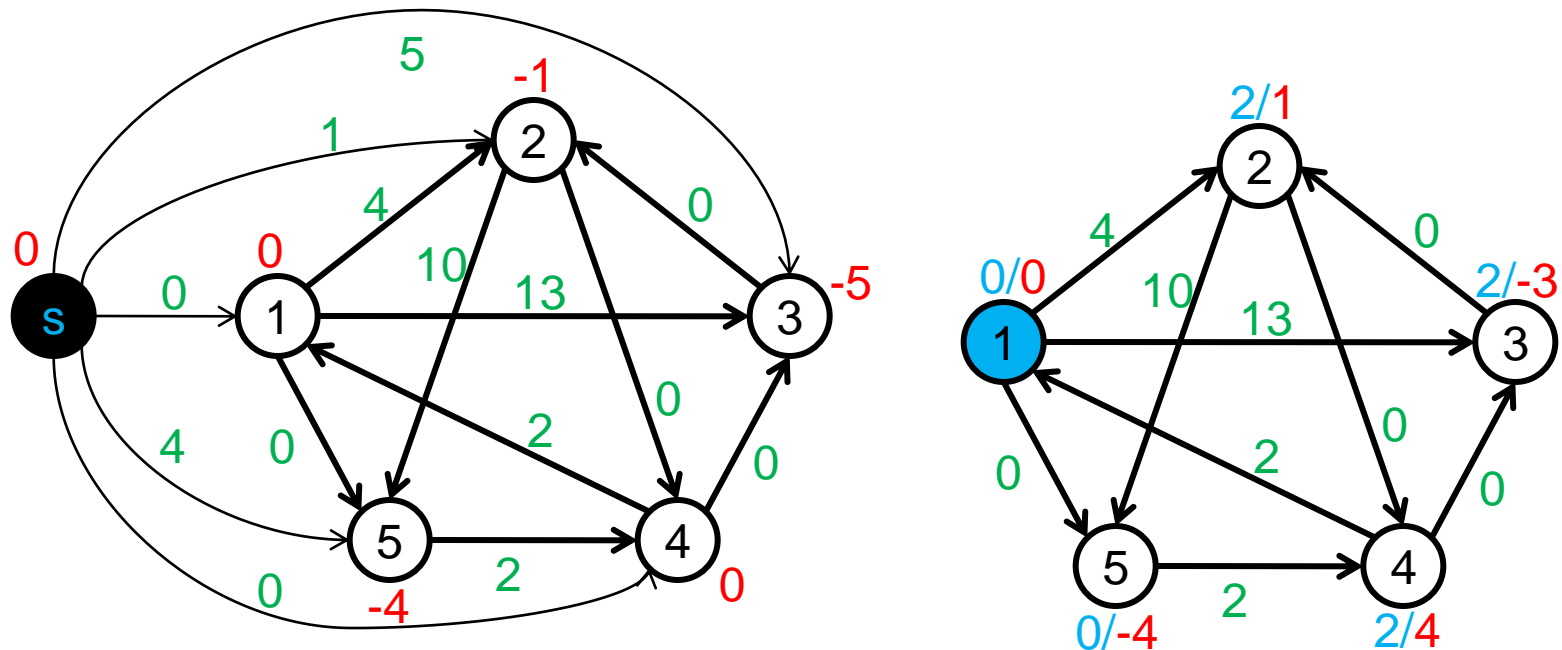
Adaug  $s$  și  
aplic **BF** pe  
noul graf.



# Exemplu (II)



# Exemplu (III)



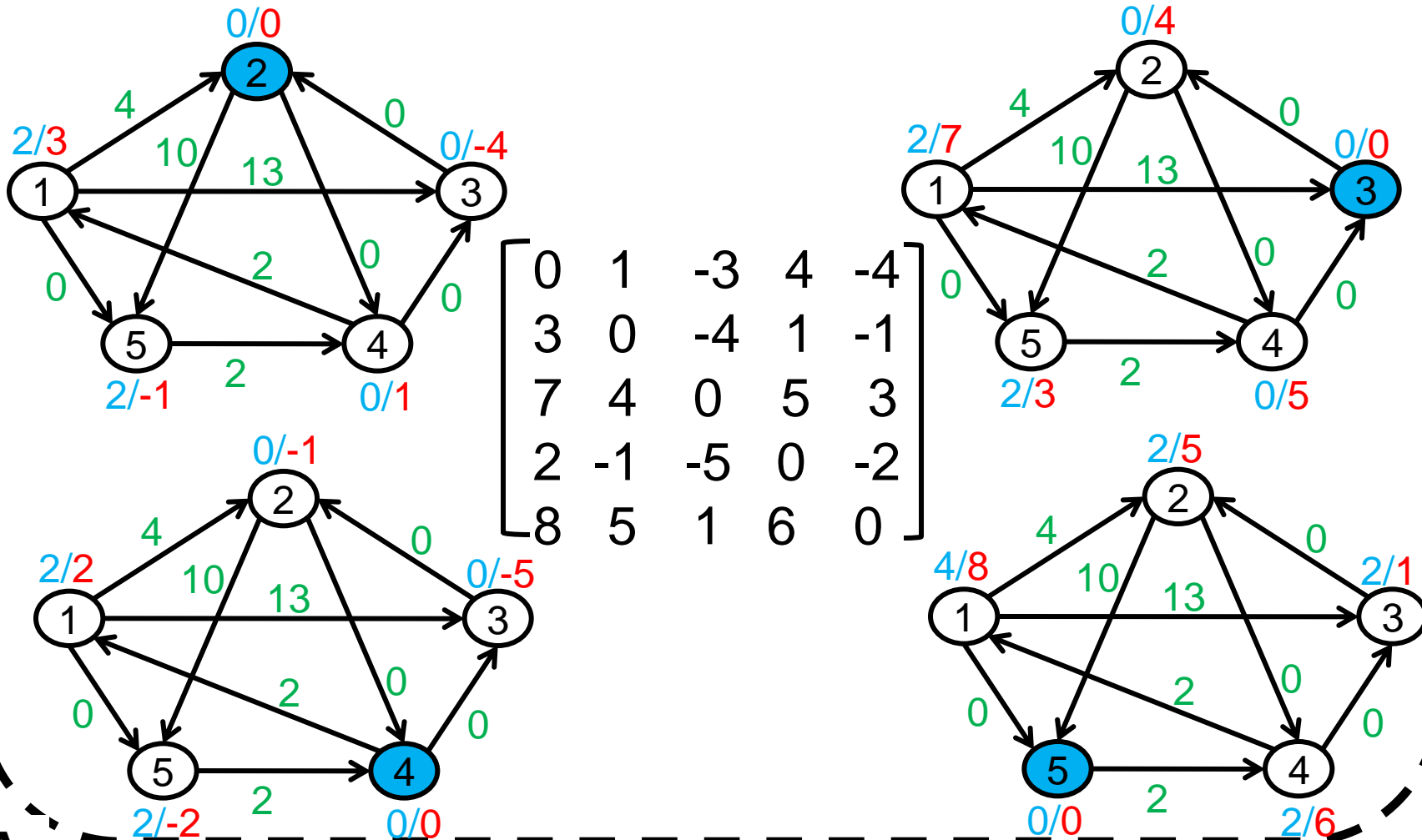
Eliminăm s

Aplicăm Dijkstra din fiecare nod ( $\delta_1(u,v)$ ).

Refacem distanțele:

$$d(u,v) = \delta_1(u,v) + h(v) - h(u)$$

# Exemplu (IV)



# Concluzii Floyd-Warshall & Johnson

- Algoritmi ce găsesc drumurile minime între oricare 2 noduri din graf.
- Funcționează pe grafuri cu muchii ce au costuri negative (dar care nu au cicluri de cost negativ).
- Floyd-Warshall e optim pentru grafuri dese.
- Johnson e mai bun pentru grafuri rare.