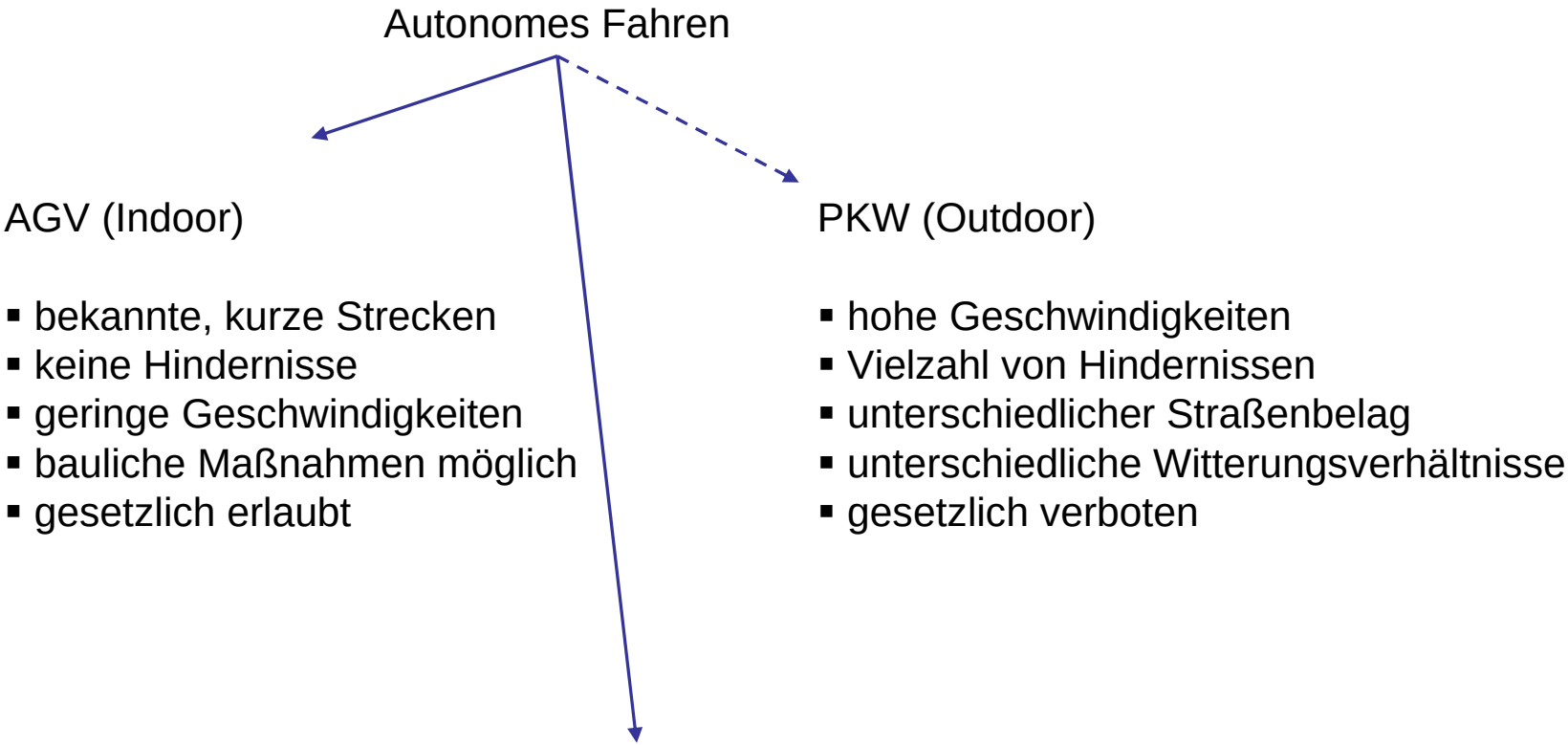
Conflict-free Real-time AGV Routing

Beschreibung eines Routing-Algorithmus mit Zeitfenstern


Autonomes Fahren
AGV (Indoor)
▪ bekannte, kurze Strecken
▪ keine Hindernisse
▪ geringe Geschwindigkeiten
▪ bauliche Maßnahmen möglich
▪ gesetzlich erlaubt
PKW (Outdoor)
▪ hohe Geschwindigkeiten
▪ Vielzahl von Hindernissen
▪ unterschiedlicher Straßenbelag
▪ unterschiedliche Witterungsverhältnisse
▪ gesetzlich verboten
▪ Busbahn
▪ elektrische Deichsel

Spurführung

- Leitdraht
- Transponder
- Magnet
- Scanner
- Bildverarbeitung
- landgestützte Satellitennavigation (DGPS)

Hinderniserkennung

- Radar
- Bildverarbeitung
- Laser

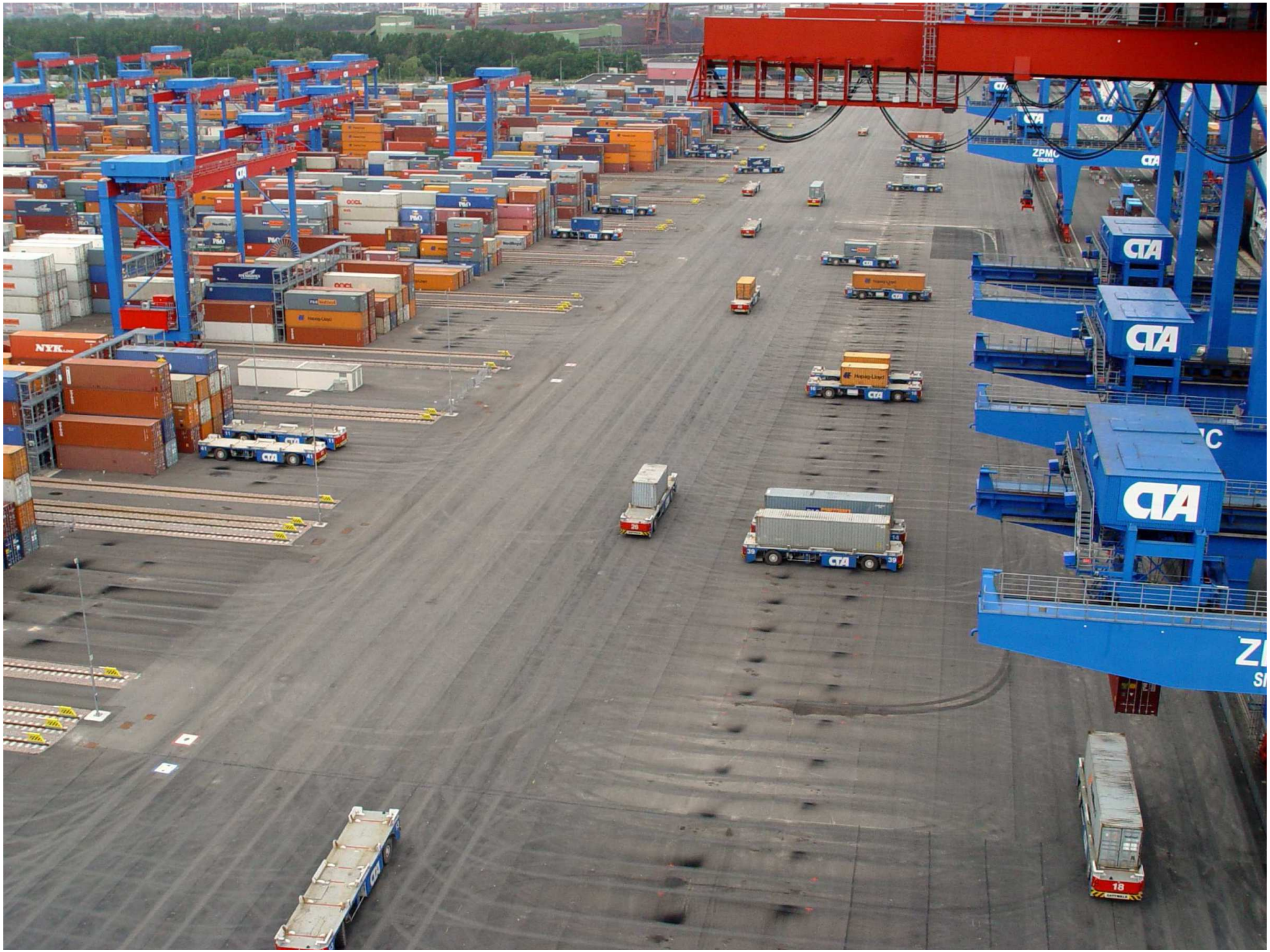
Kommunikation

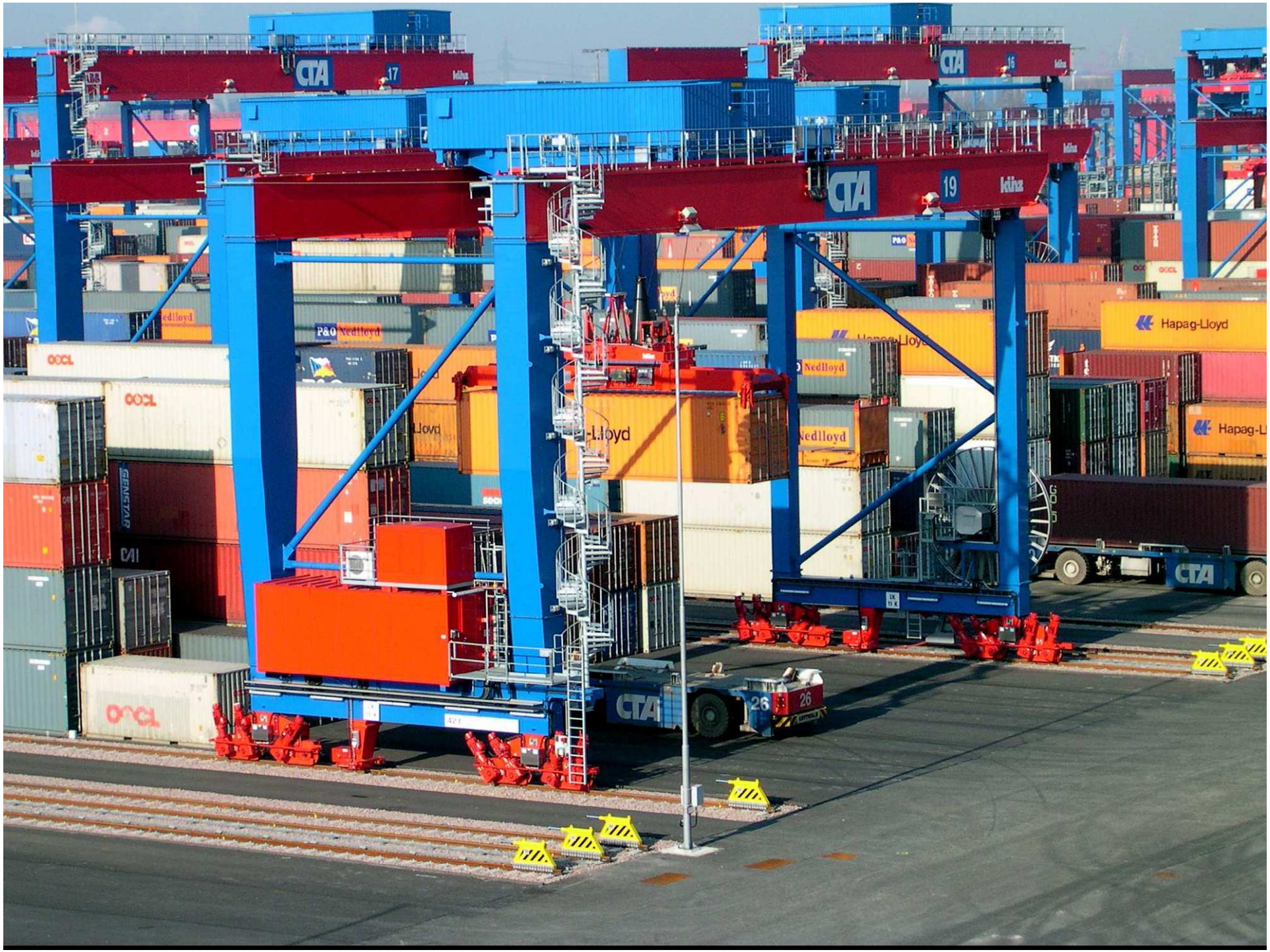
- Induktive Datenübertragung (100 kHz)
- Infrarot
- Datenfunk (2,54 GHz)
    - geringe Störempfindlichkeit
    - uneingeschränkte Reichweite
    - hohe Übertragungsgeschwindigkeit

Kontrolle

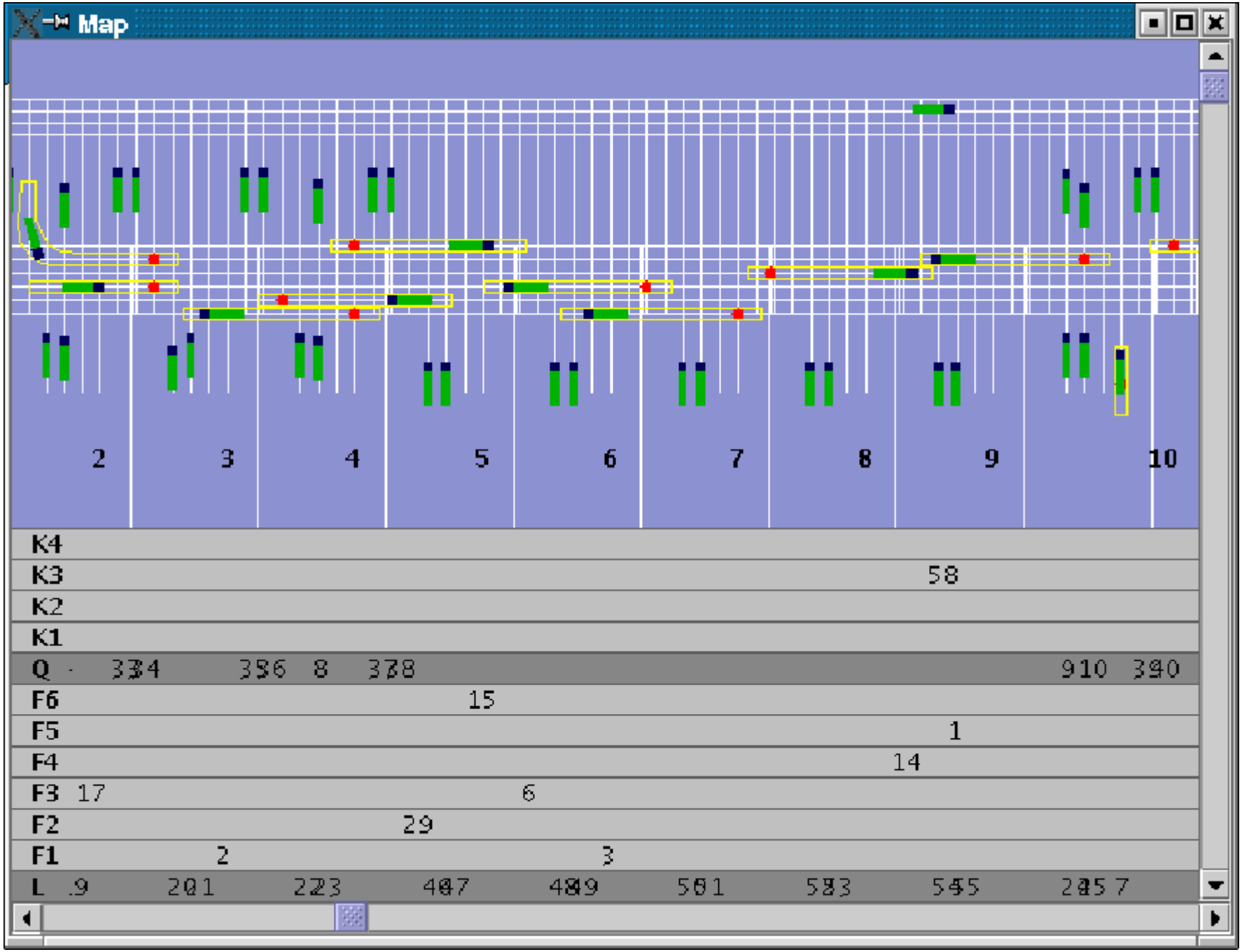
- Routing
- Kollisionsverhütung


P&O Nedlloyd


CTA
ZPMC
NYK
OOCL
P&O
Hapag-Lloyd


CTA
17
19
26
Hapag-Lloyd
Nedlloyd
P&O Nedlloyd
OOCL
GENSTAR


HYDAC
CTA

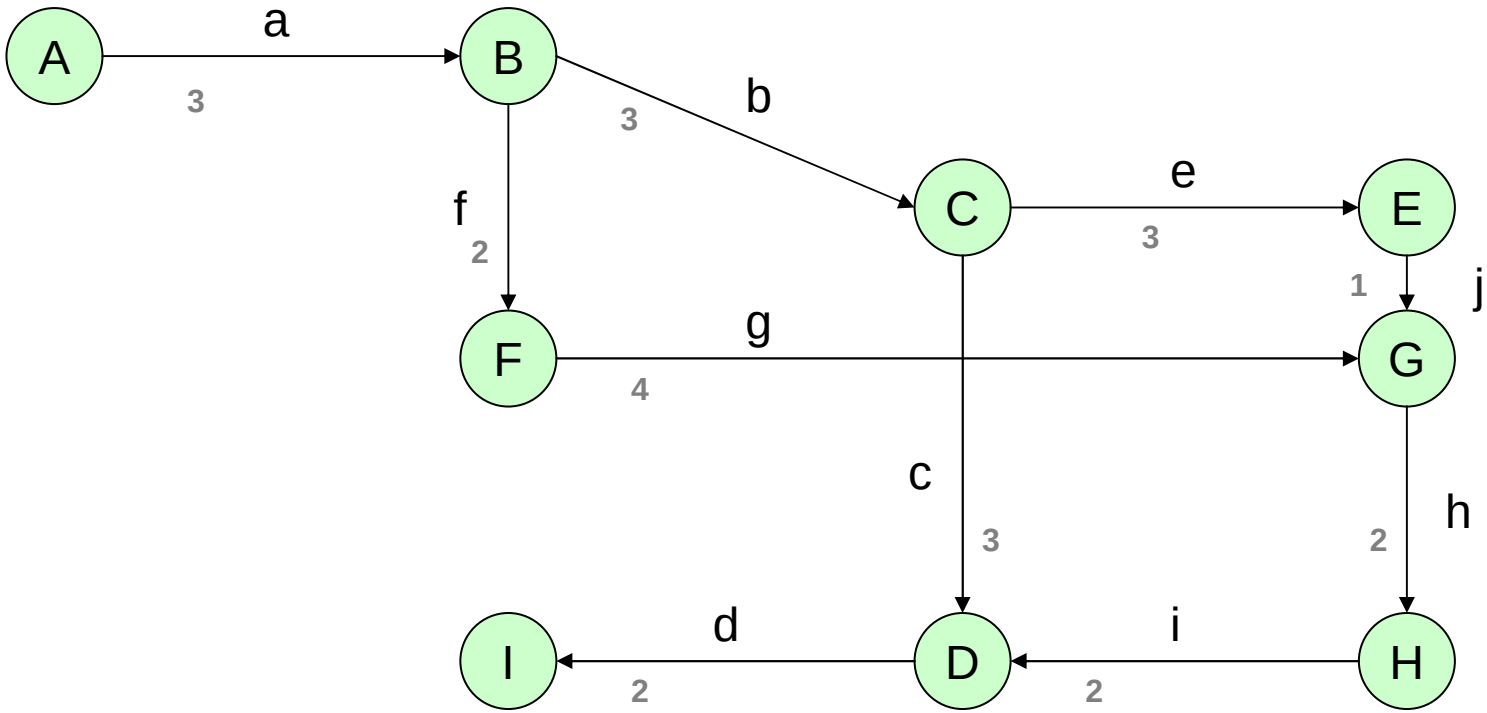

Map
2
3
4
5
6
7
8
9
10
K4
K3
58
K2
K1
Q
910
F6
15
F5
1
F4
14
F3
17
6
F2
29
F1
2
3
L

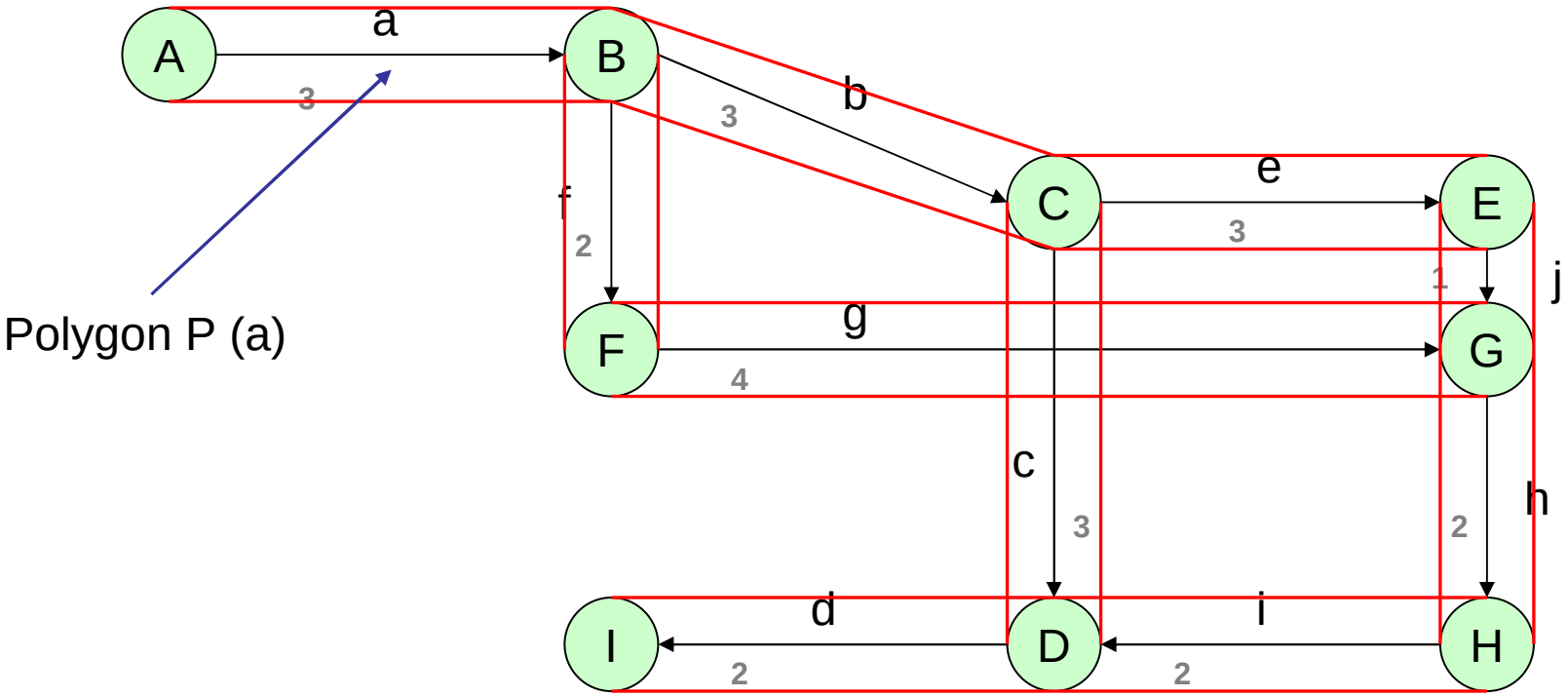
Das Modell

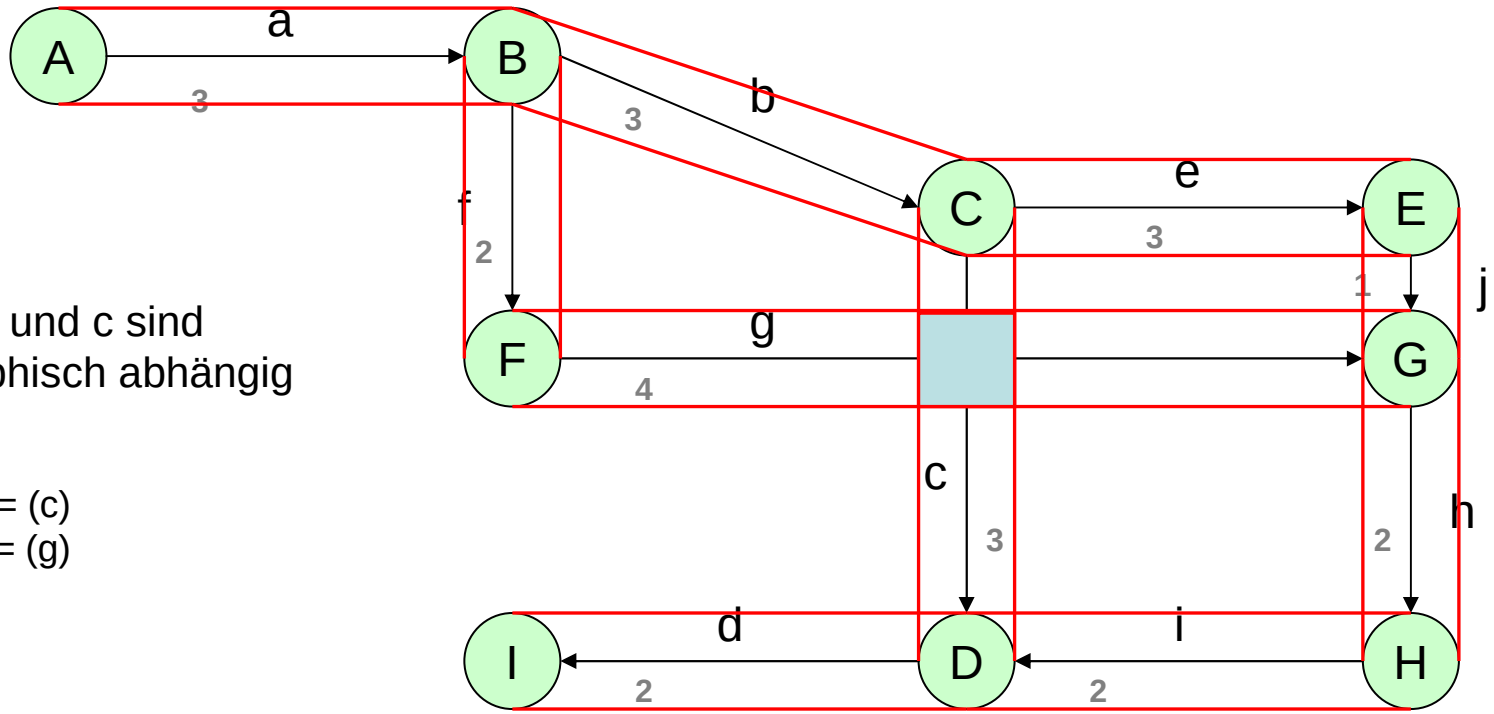
$G = (V, A)$	gerichteter Graph, der die Wege des Transportsystems repräsentiert
$T(\alpha)$	Fahrtzeit für Kante α
$\sigma = r_1, \ldots, r_n$	Sequenz von Anfragen

Gesucht ist eine Route von Startpunkt s_j zum Zielknoten t_j zum Zeitpunkt θ_j

$r_j = (s_j, t_j, \theta_j)$	Online-Anfrage eines AGV

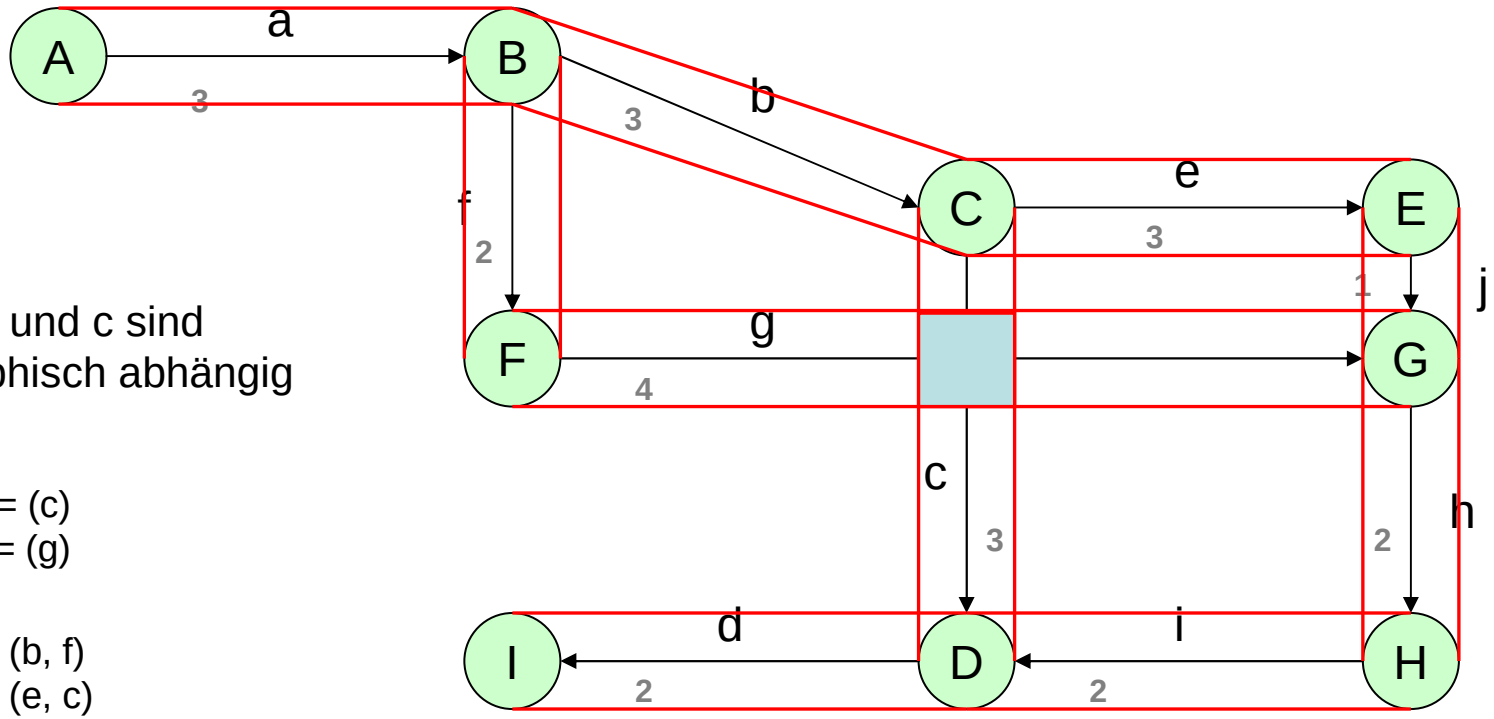

A
a
3
B
b
3
f
2
C
e
3
E
1
j
F
g
4
G
c
3
h
2
I
d
2
D
i
2
H


A
B
C
E
F
G
D
H
I
a
b
e
f
g
j
c
h
d
i
3
3
2
3
1
4
3
2
2
2
Polygon P (a)



Kante g und c sind geographisch abhängig

confl (g) = (c)
confl (c) = (g)



Kante g und c sind geographisch abhängig

confl (g) = (c)
confl (c) = (g)

Out (a) = (b, f)
Out (b) = (e, c)
Out (c) = (d)
Out (d) = nil
Out (e) = (j)
Out (f) = (g)
Out (g) = (h)
Out (h) = (i)
Out (i) = (d)
Out (j) = (h)

Das Problem

„Kürzeste Wege Problem mit Zeitfenstern“

c_α	Kosten für Kante α
F_α	Menge der Zeitfenster für Kante α

Der Algorithmus

$(\alpha_L, d_L, I_L, pred_L)$	Label L

Jedes Label L repräsentiert einen Pfad vom Startknoten s zum Ende von α_L

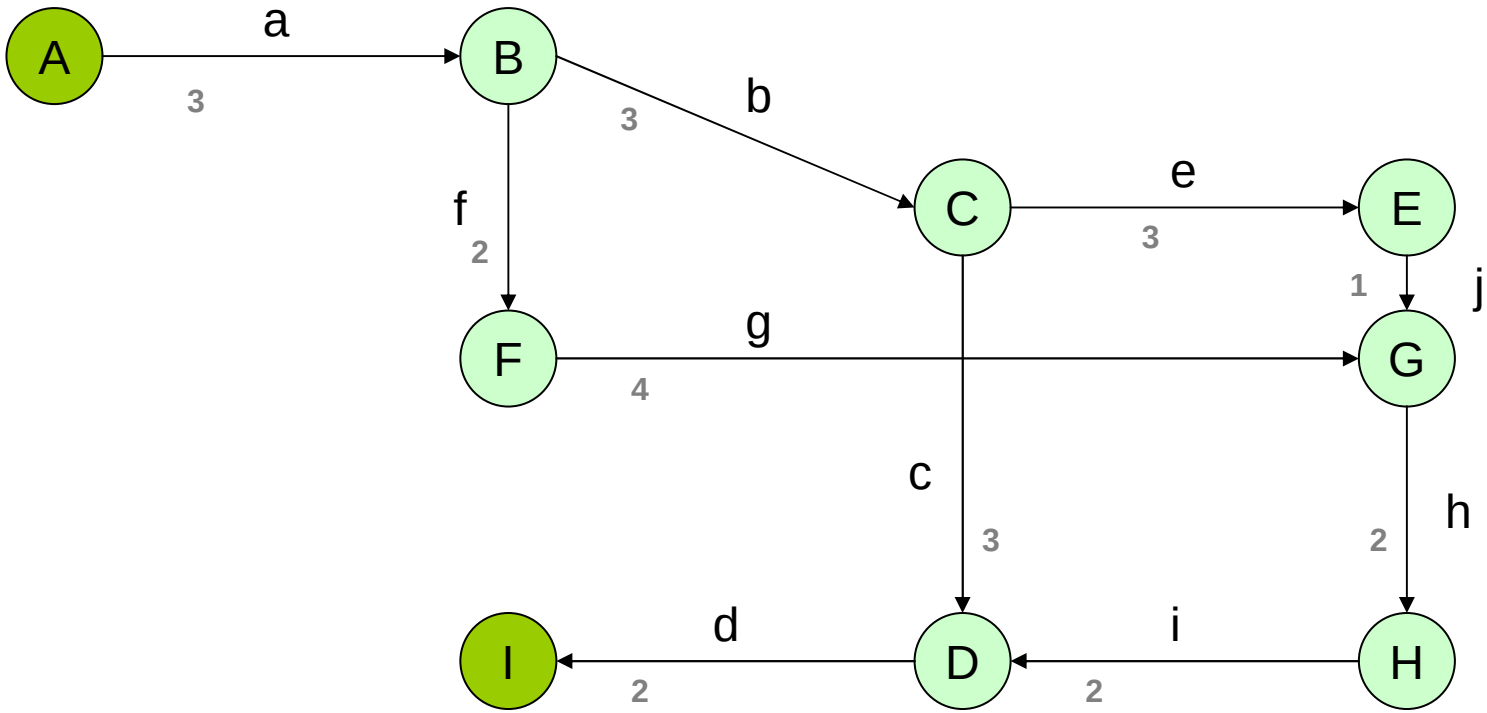
α_L	Label für Kante α
d_L	Distanzwert (Reise- und Wartezeiten)
I_L	Labelinterval $I_L = (A_L, B_L)$
$pred_L$	Vorgänger von α_L

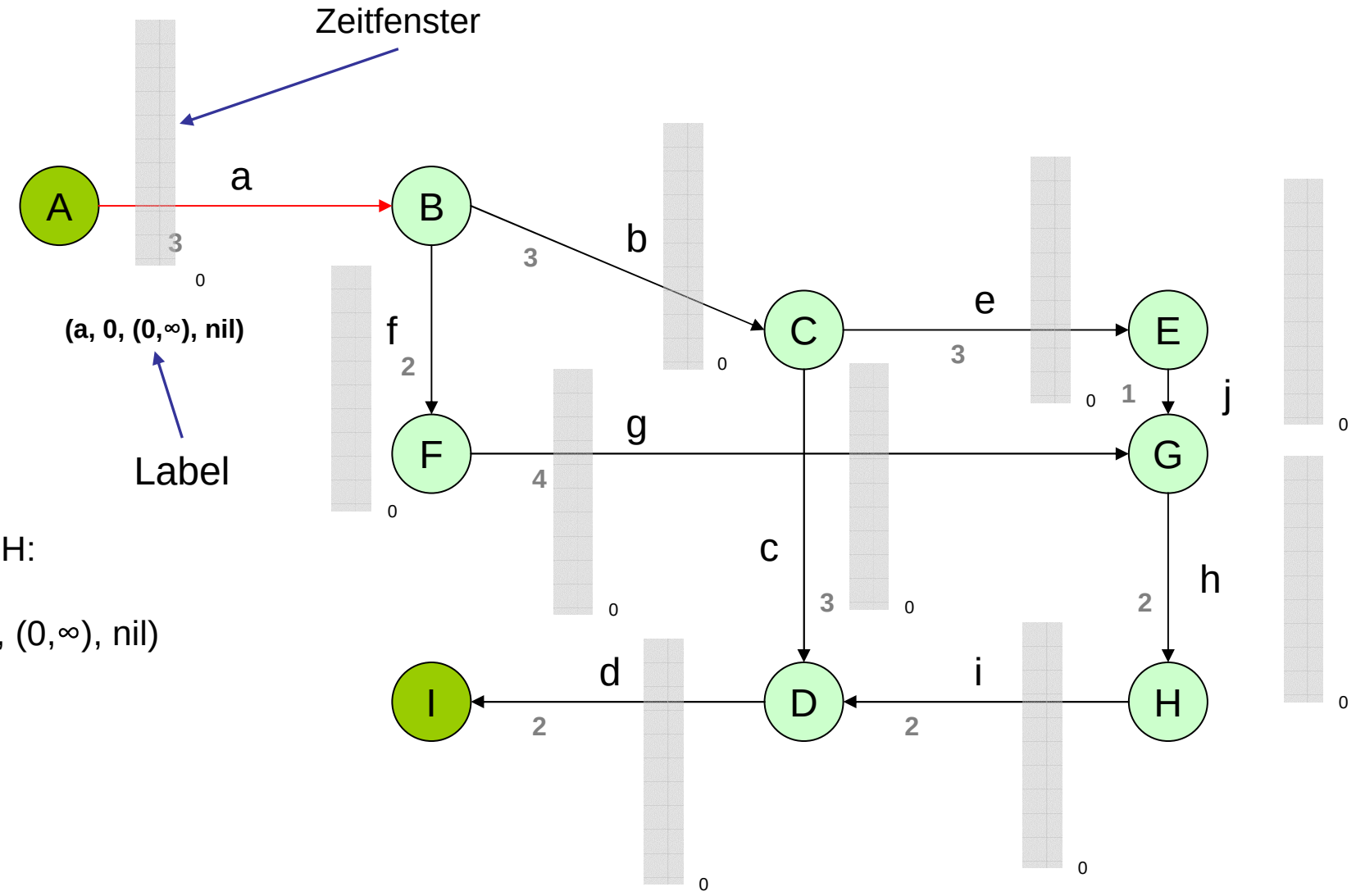
Definition: Ein Label L dominiert Label L‘, genau dann wenn…

$$d_L \leq d_{L'} \text{ und } I_{L'} \subseteq I_L$$

```
* Initialisierung.
* Schleife
    Beginn
            * Label mit kleinstem d auswählen.
            * Wenn keine Label vorhanden,
                    dann Stop.
            * Wenn Ziel erreicht,
                    dann Stop.
            * Für jedes Zeitfenster
                    * Label expandieren.
                    * Dominanztest.
     Ende.
* Zeitfenster aktualisieren.
```

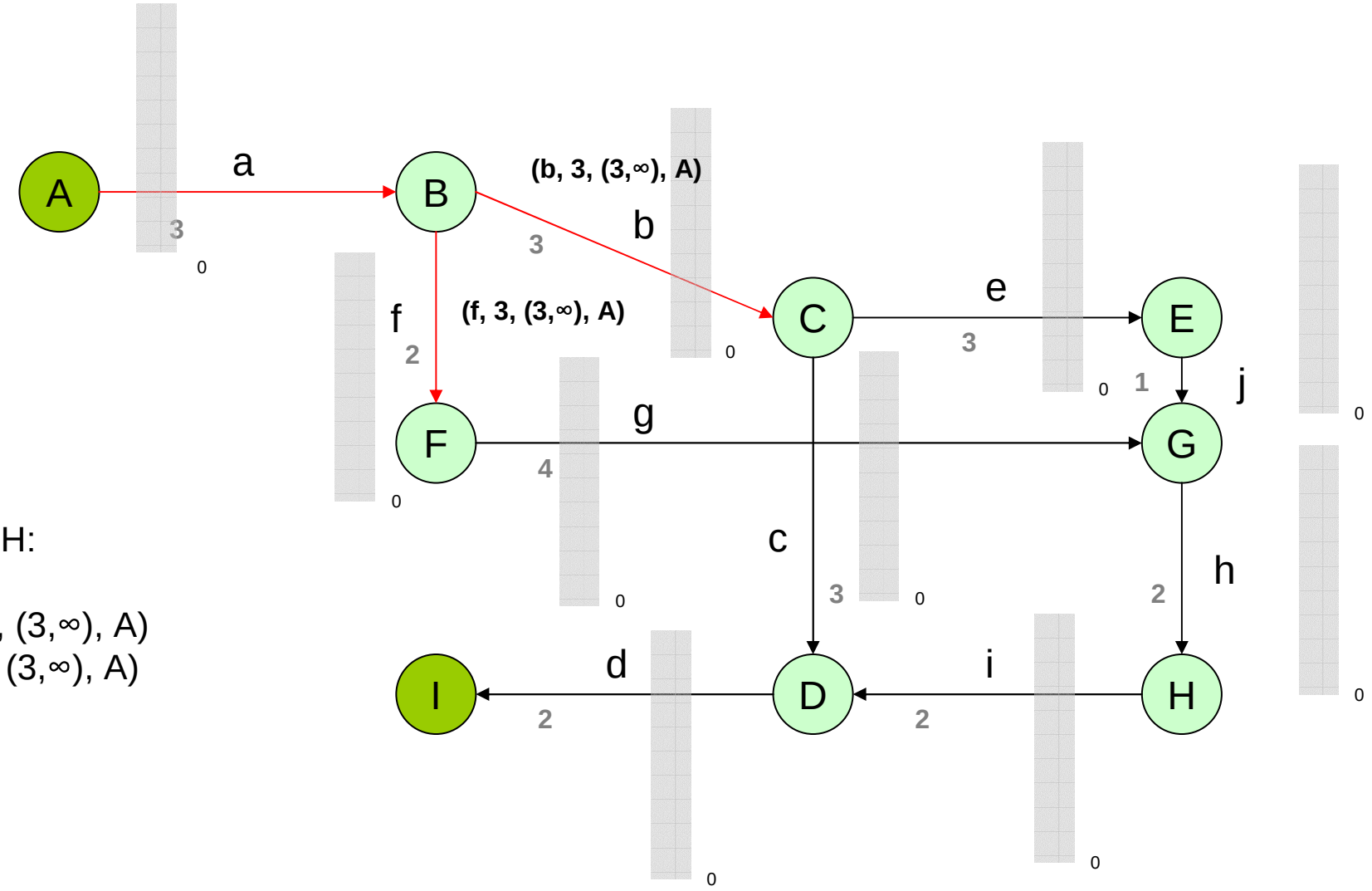
1. Anfrage: r = (A,I,0)





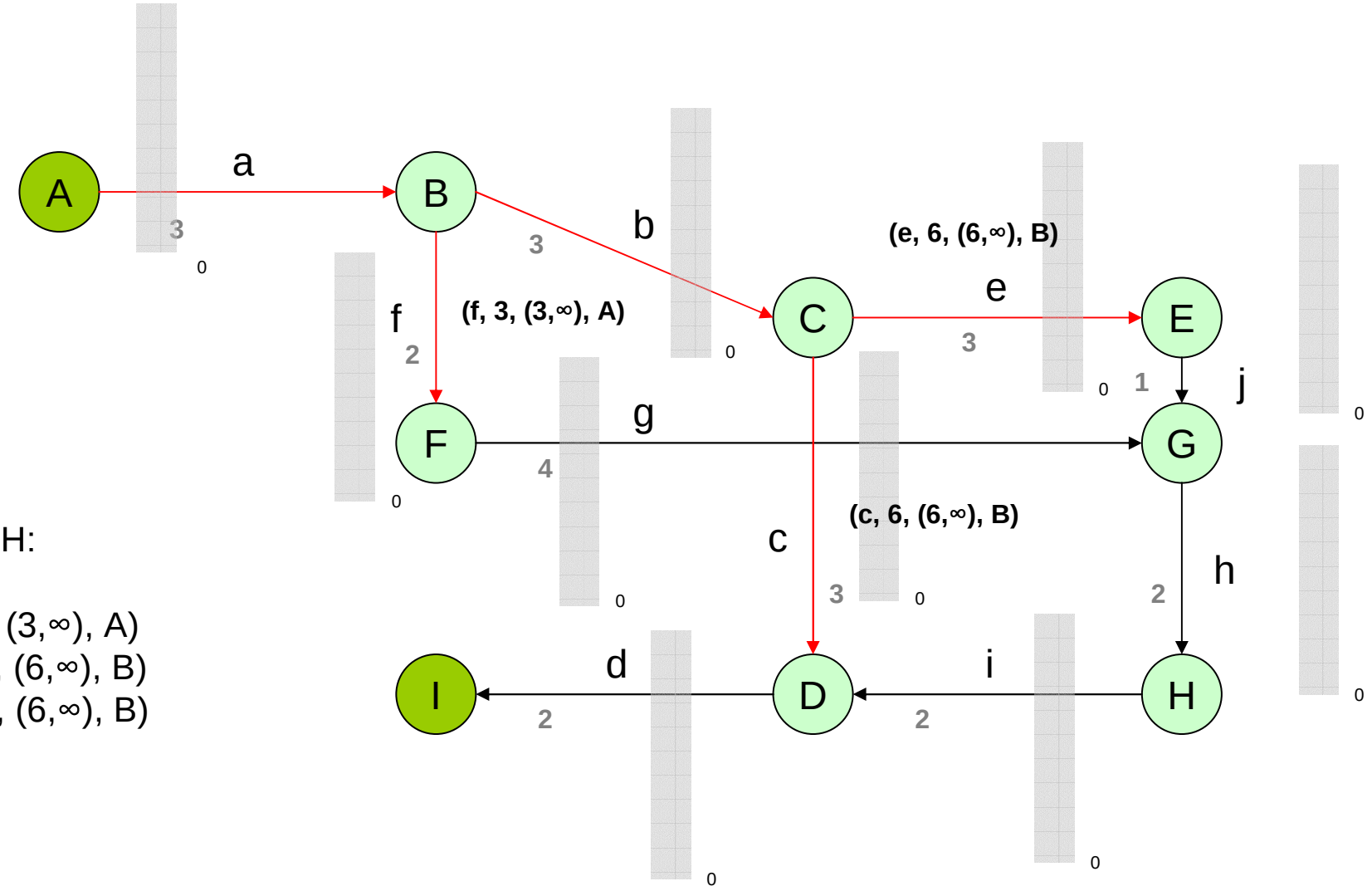
Queue H:

1. (a, 0, (0,∞), nil)
2.
3.
4.
5.



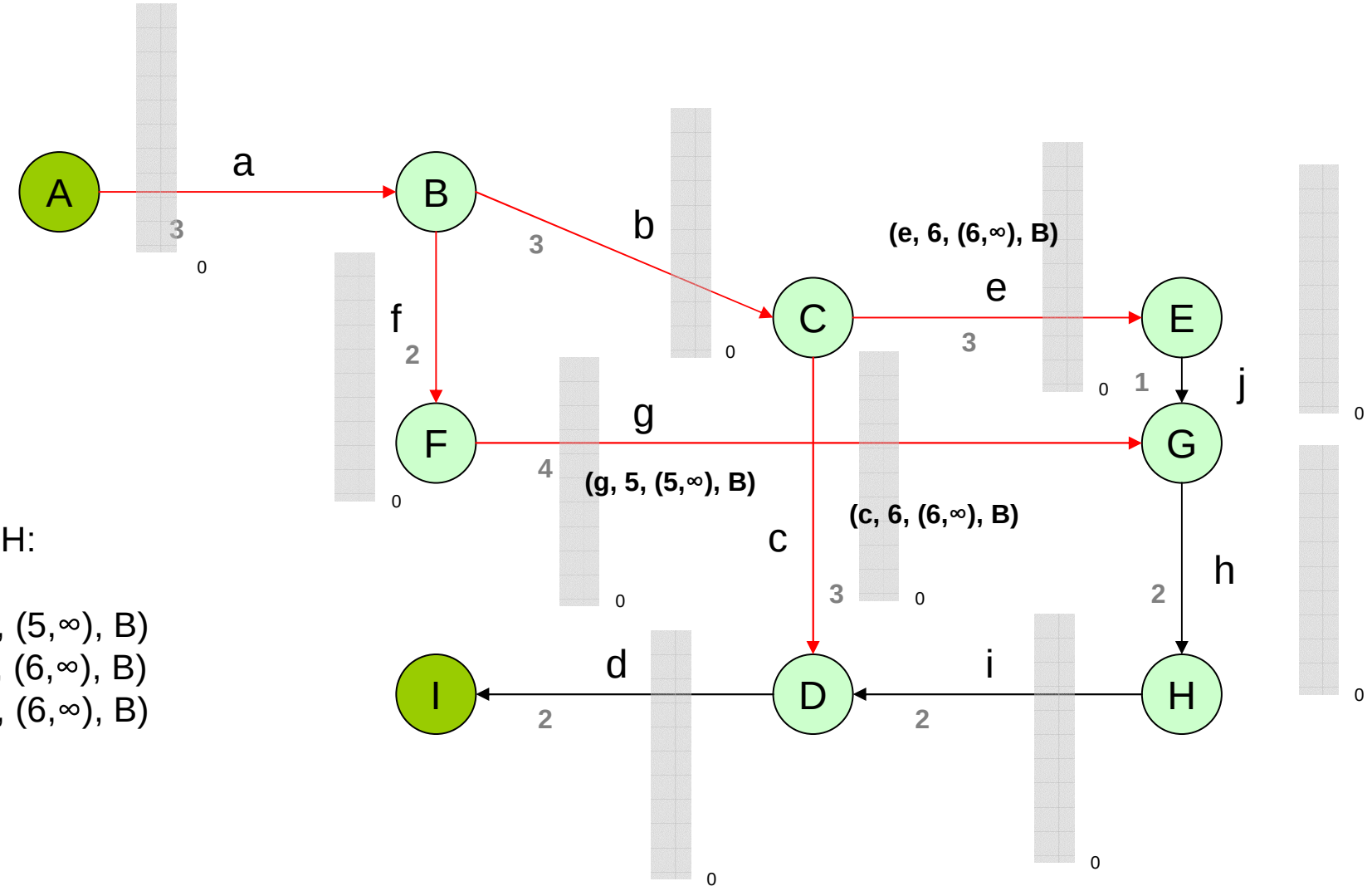
Queue H:

1. (b, 3, (3,∞), A)
2. (f, 3, (3,∞), A)
3.
4.
5.



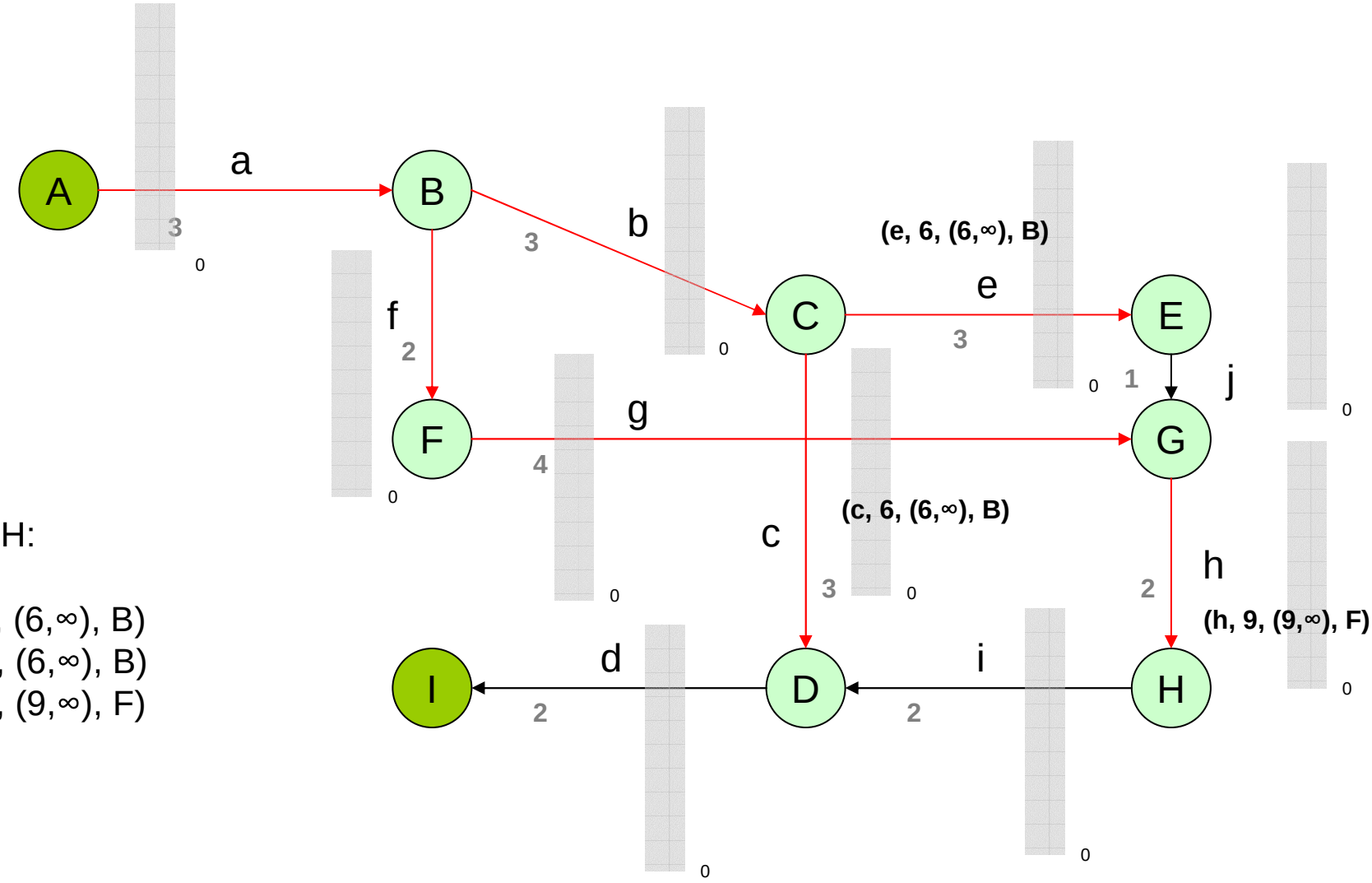
Queue H:

1. (f, 3, (3,∞), A)
2. (c, 6, (6,∞), B)
3. (e, 6, (6,∞), B)
4.
5.



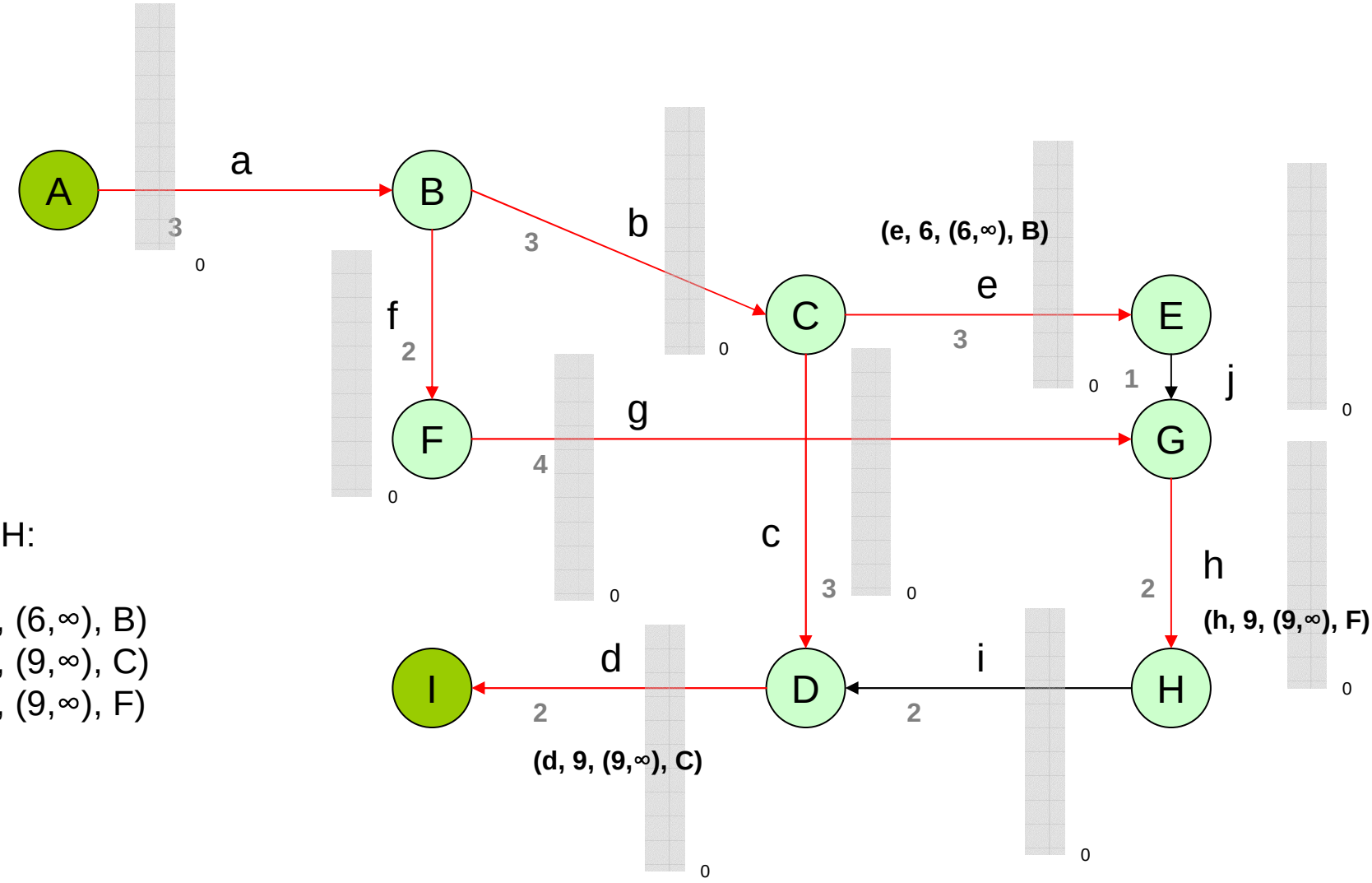
Queue H:

1. (g, 5, (5,∞), B)
2. (c, 6, (6,∞), B)
3. (e, 6, (6,∞), B)
4.
5.



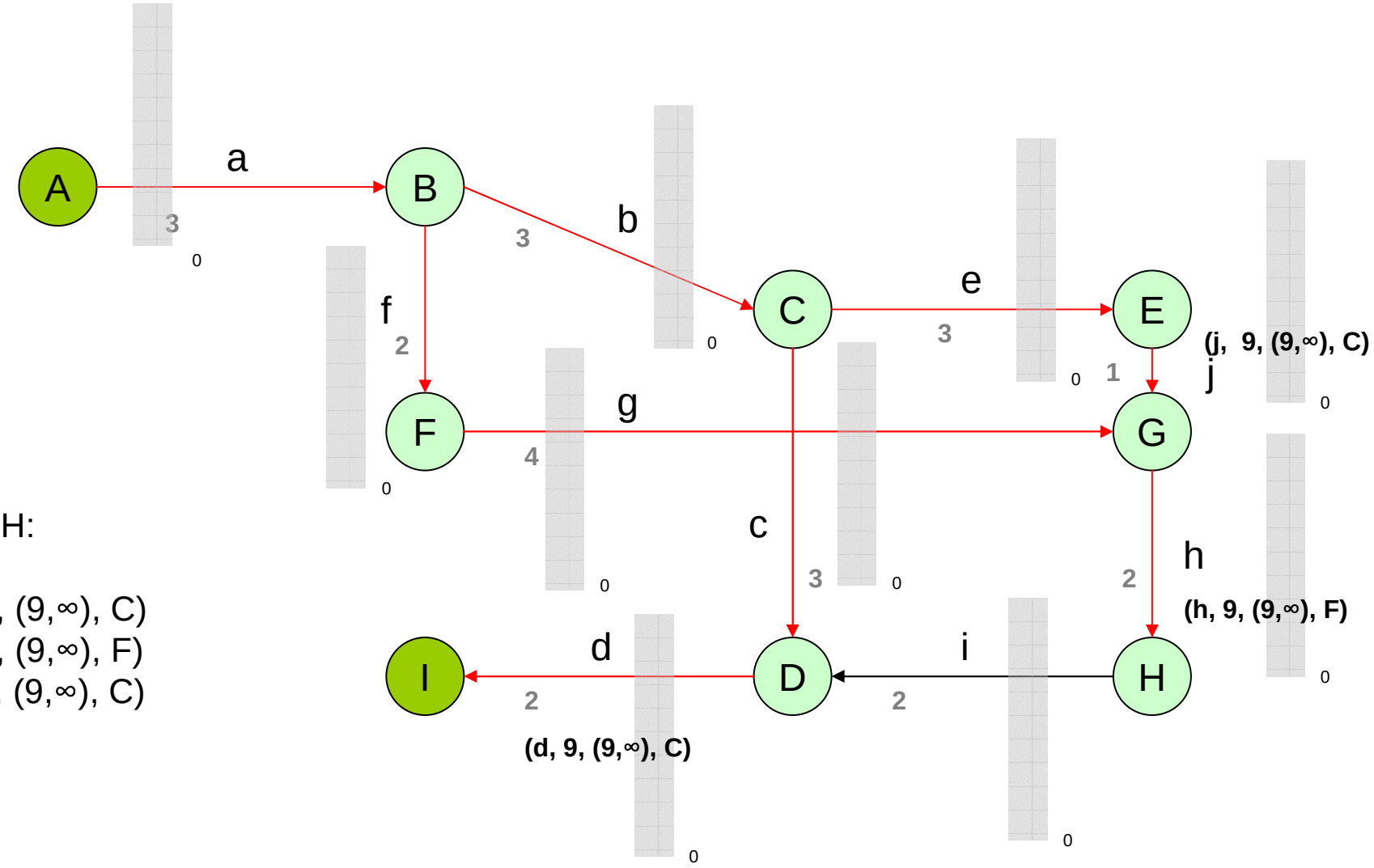
Queue H:

1. (c, 6, (6,∞), B)
2. (e, 6, (6,∞), B)
3. (h, 9, (9,∞), F)
4.
5.



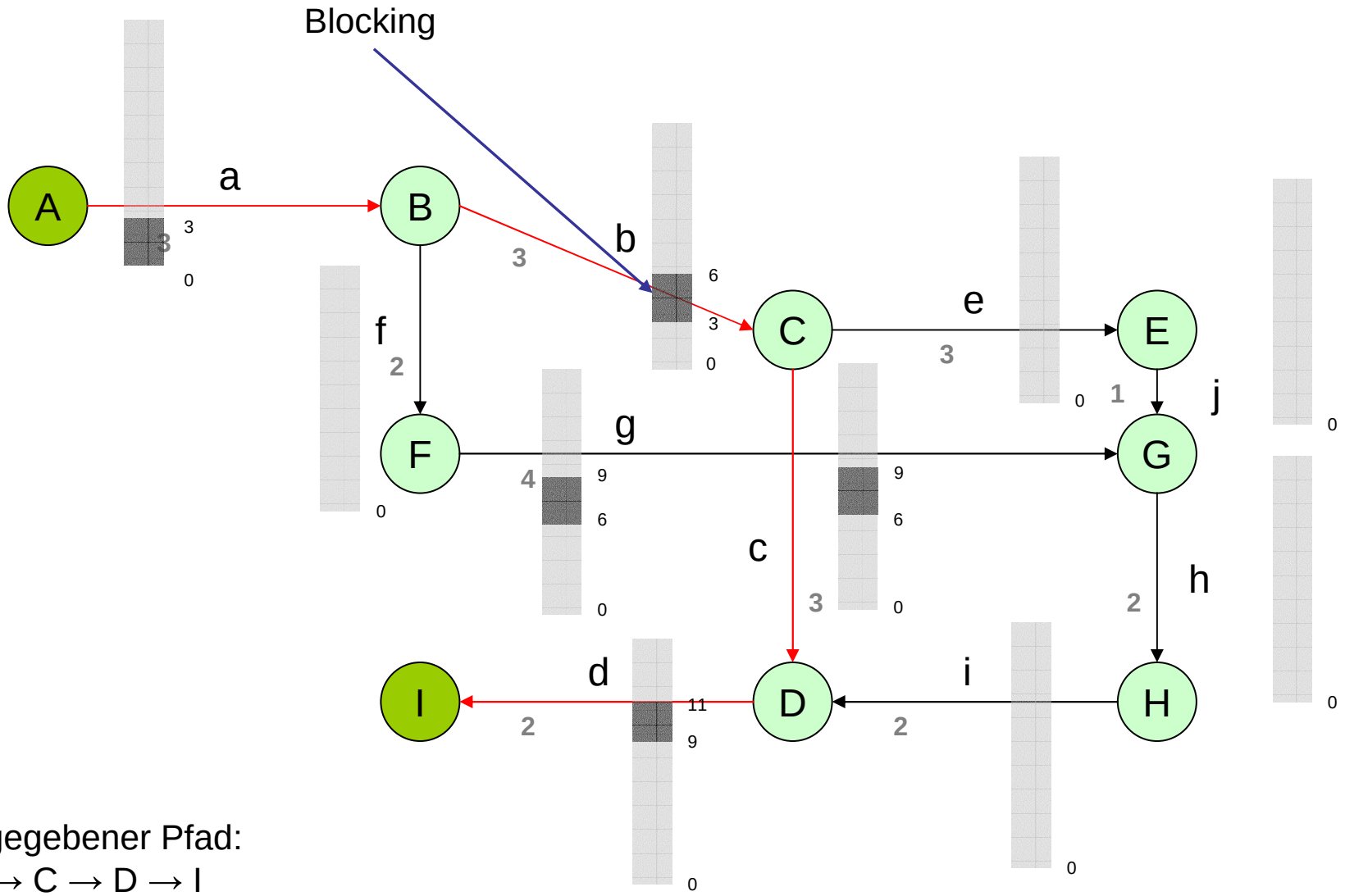
Queue H:

1. (e, 6, (6,∞), B)
2. (d, 9, (9,∞), C)
3. (h, 9, (9,∞), F)
4.
5.



Queue H:

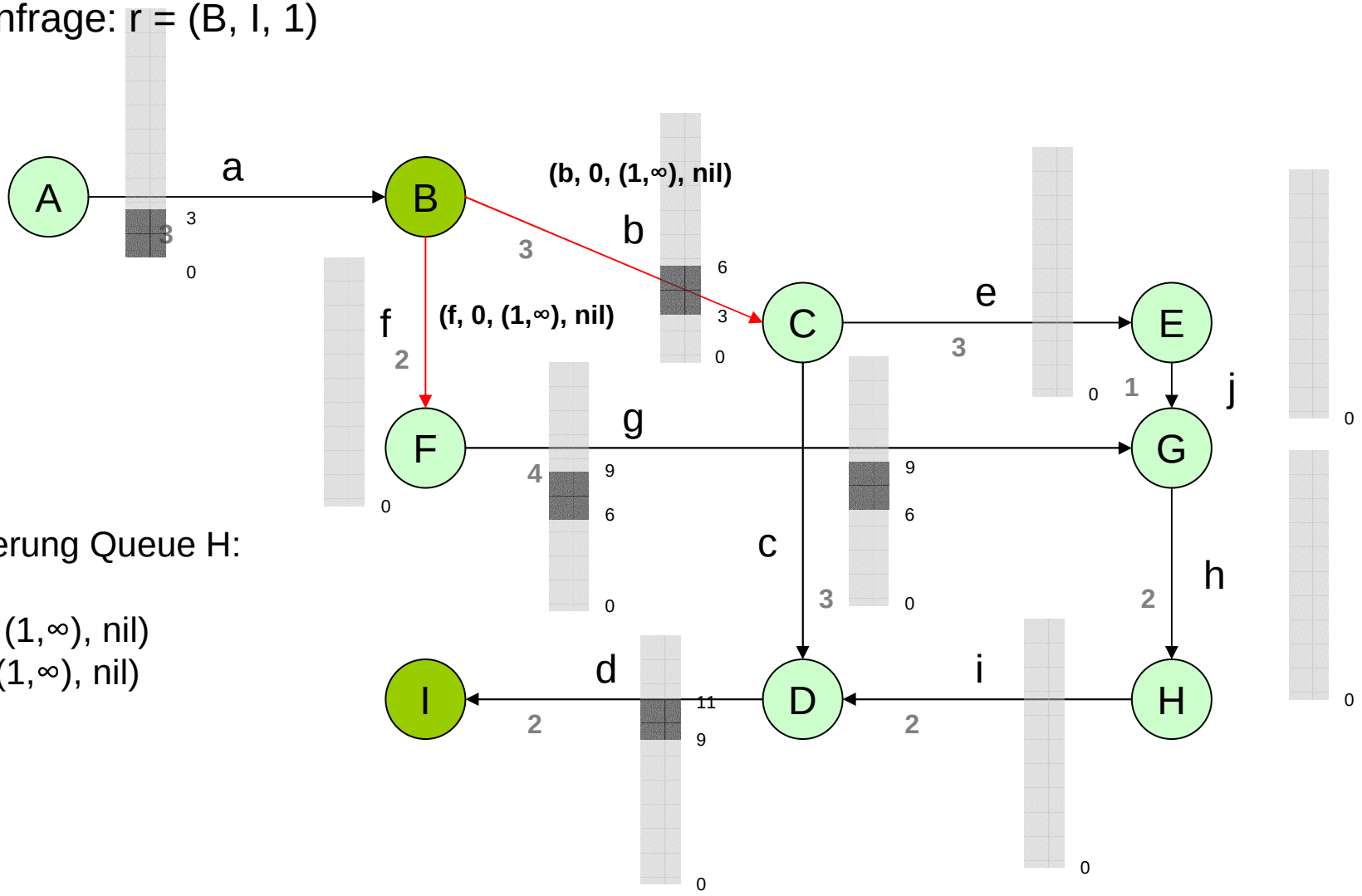
1. (d, 9, (9,∞), C)
2. (h, 9, (9,∞), F)
3. (j, 9, (9,∞), C)
4.
5.



Zurückgegebener Pfad:
A → B → C → D → I

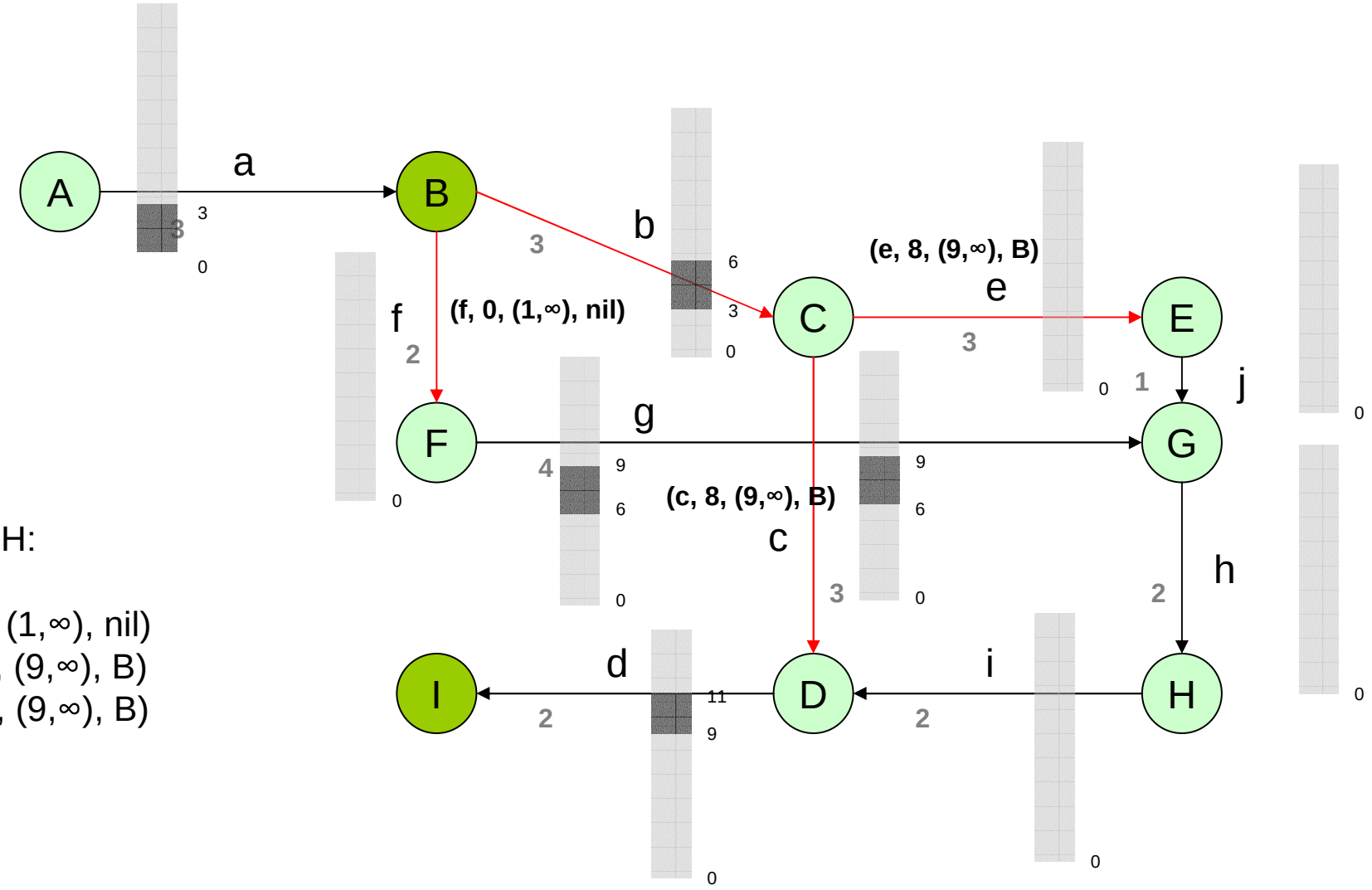
Fahrtzeit: 11 Zeiteinheiten
davon Wartezeit: 0 Zeiteinheiten

2. Anfrage: r = (B, I, 1)



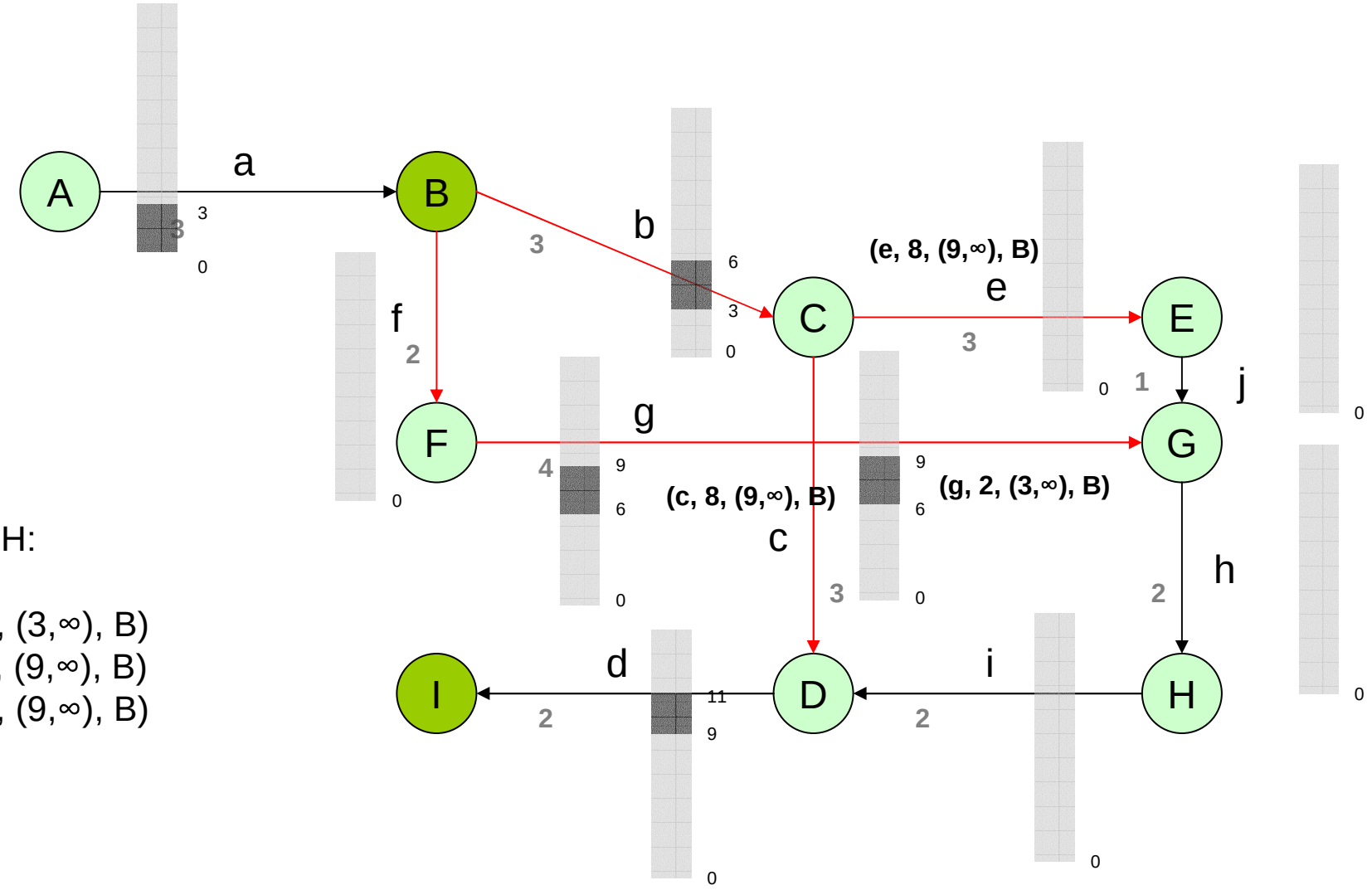
Initialisierung Queue H:

1. (b, 0, (1,∞), nil)
2. (f, 0, (1,∞), nil)
3.
4.
5.



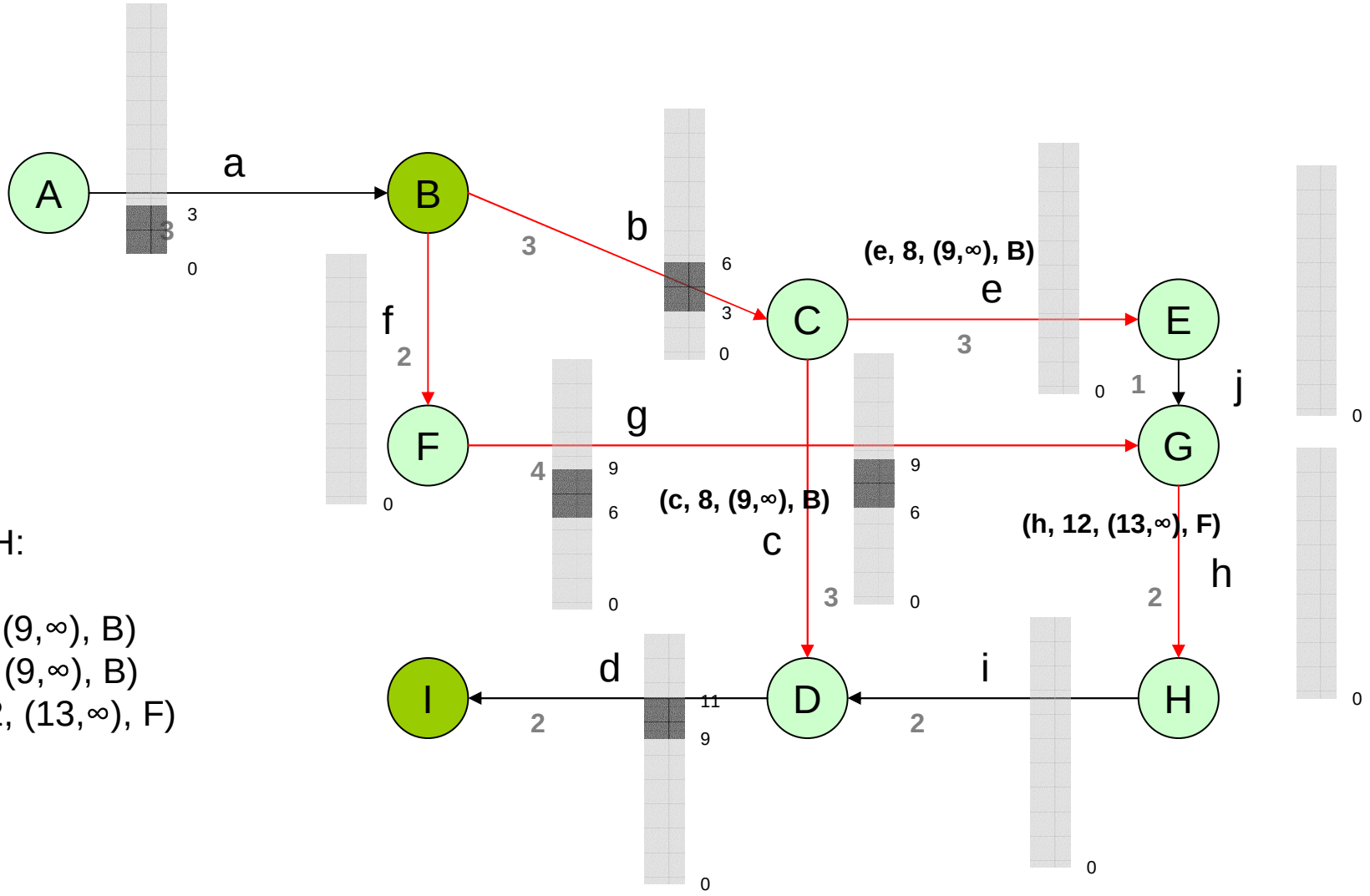
Queue H:

1. (f, 0, (1,∞), nil)
2. (c, 8, (9,∞), B)
3. (e, 8, (9,∞), B)
4.
5.



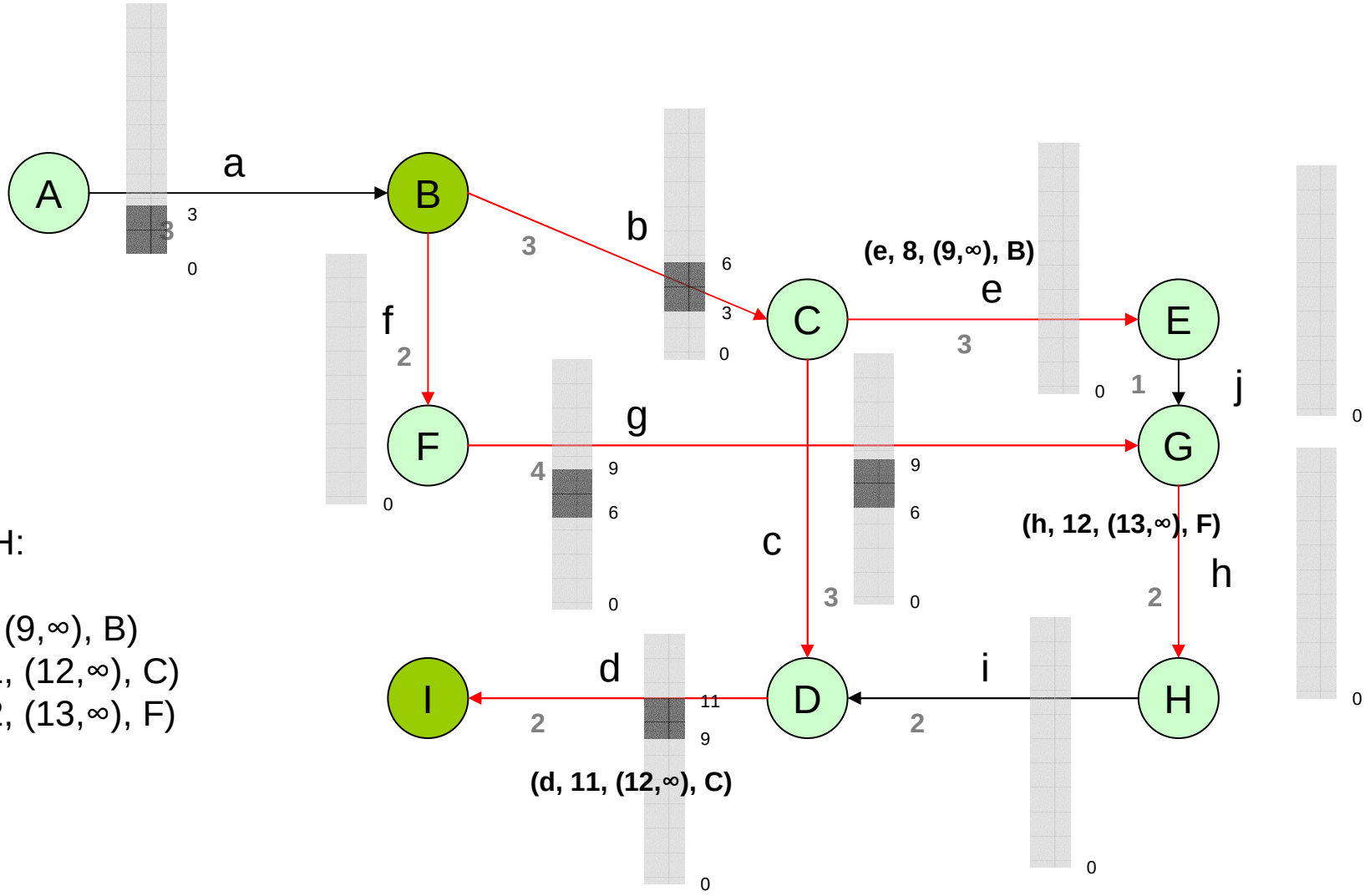
Queue H:

1. (g, 2, (3,∞), B)
2. (c, 8, (9,∞), B)
3. (e, 8, (9,∞), B)
4.
5.



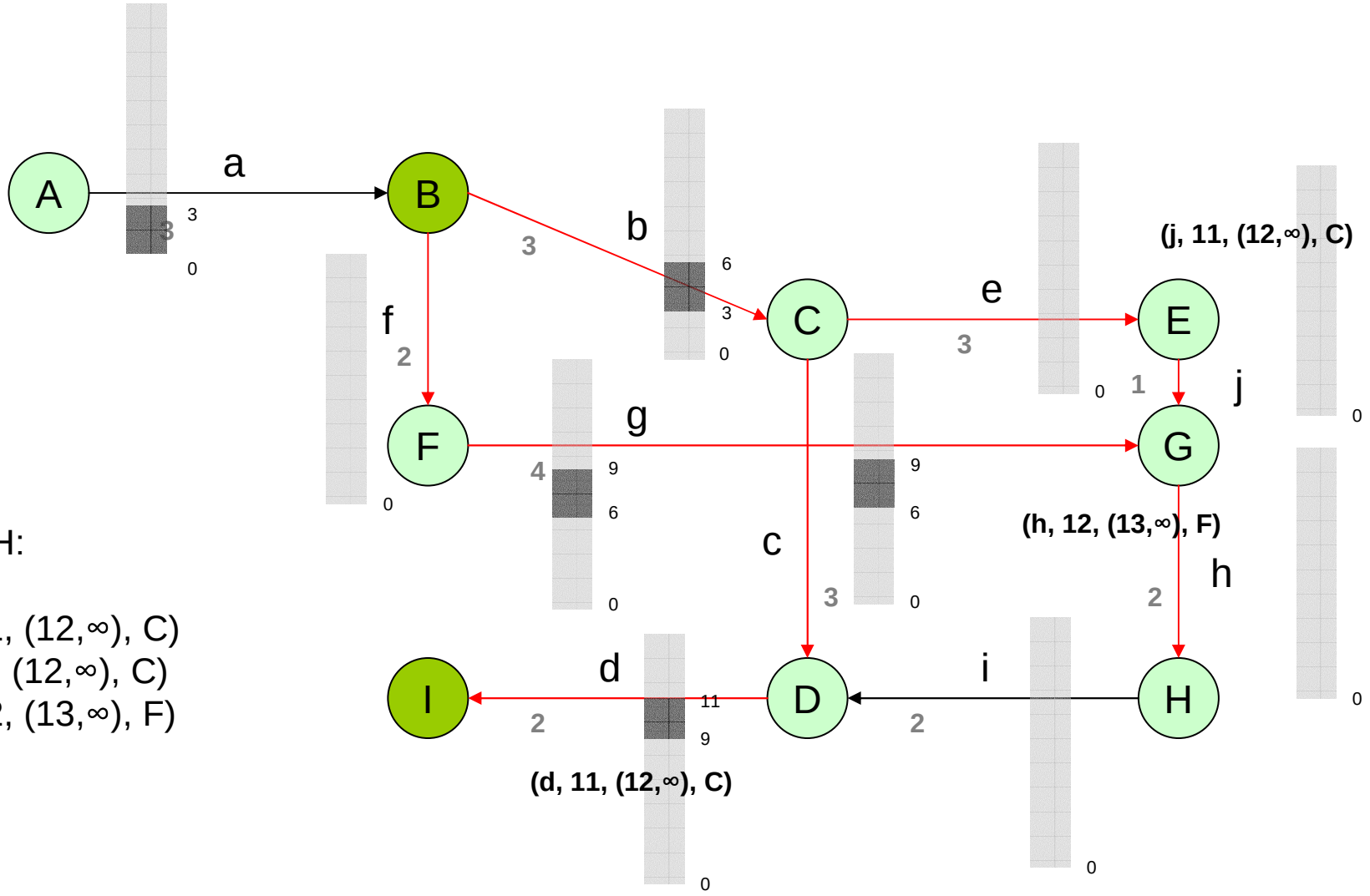
Queue H:

1. (c, 8, (9,∞), B)
2. (e, 8, (9,∞), B)
3. (h, 12, (13,∞), F)
4.
5.



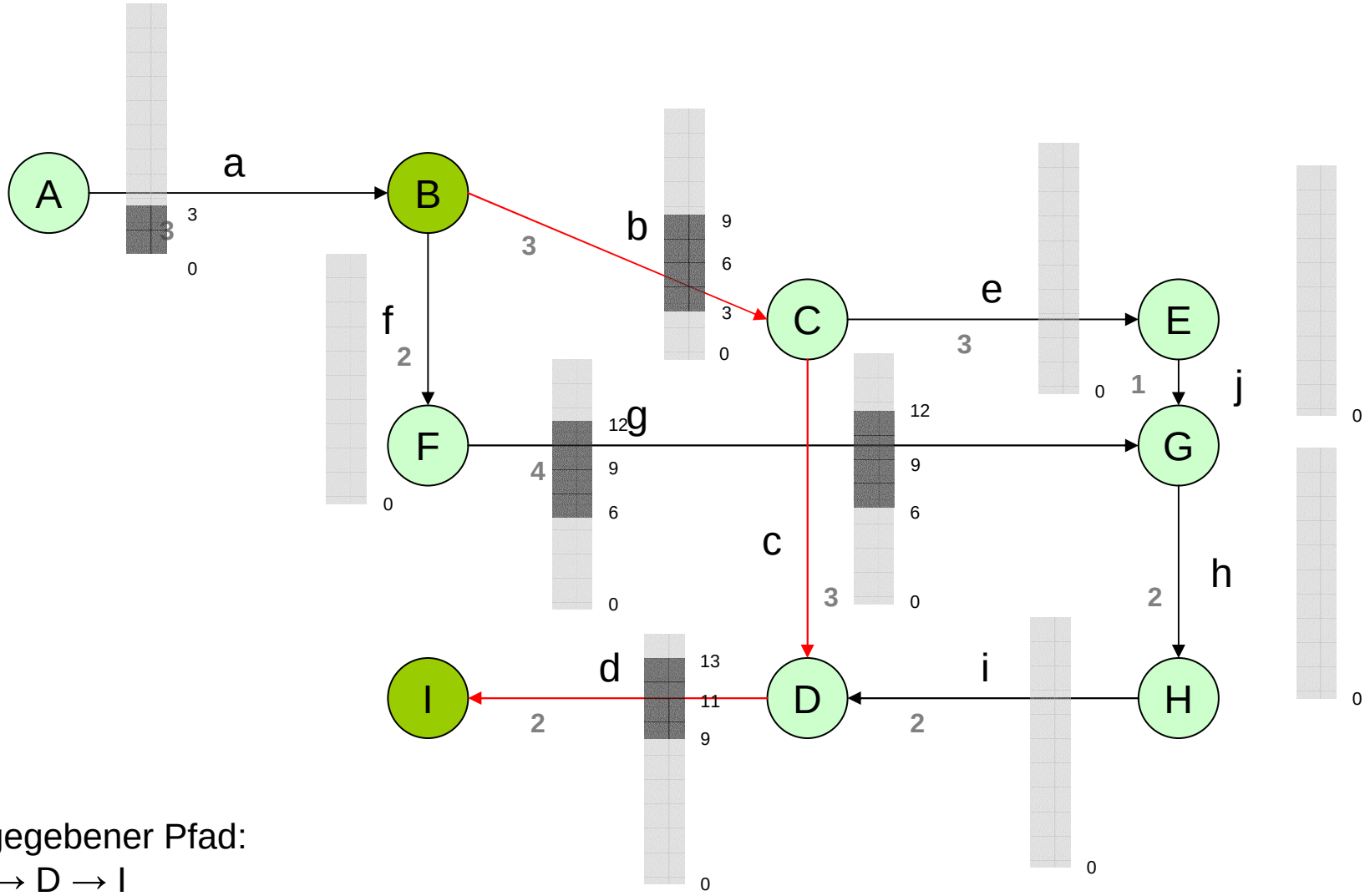
Queue H:

1. (e, 8, (9,∞), B)
2. (d, 11, (12,∞), C)
3. (h, 12, (13,∞), F)
4.
5.



Queue H:

1. (d, 11, (12,∞), C)
2. (j, 11, (12,∞), C)
3. (h, 12, (13,∞), F)
4.
5.



Zurückgegebener Pfad:
B → C → D → I

Fahrtzeit: 13 Zeiteinheiten
davon Wartezeit: 5 Zeiteinheiten