Distributed Fringe Search with MPI

Christian Zeman & Lukas Mosimann

ETH Zürich

Design of Parallel and High-Performance Computing

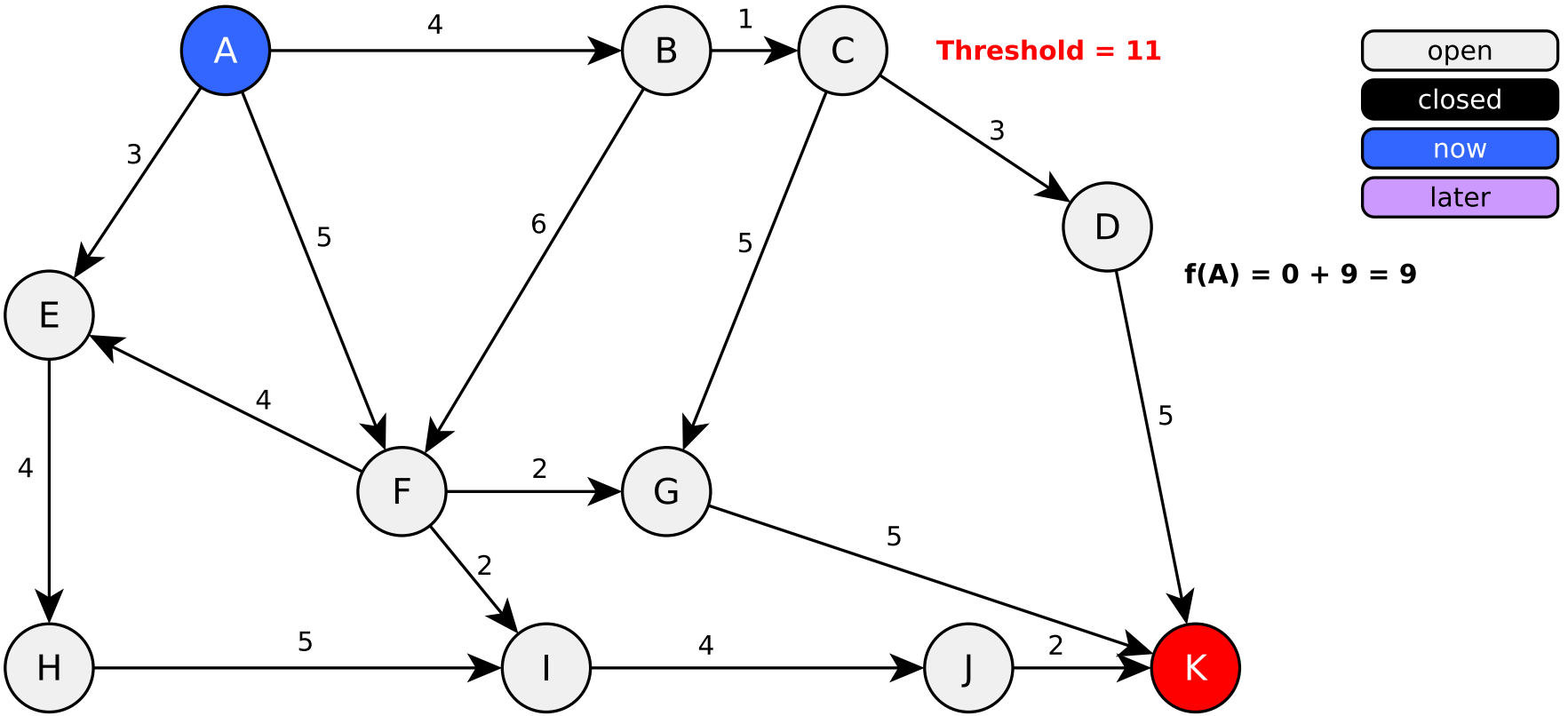
November 3, 2013

Overview

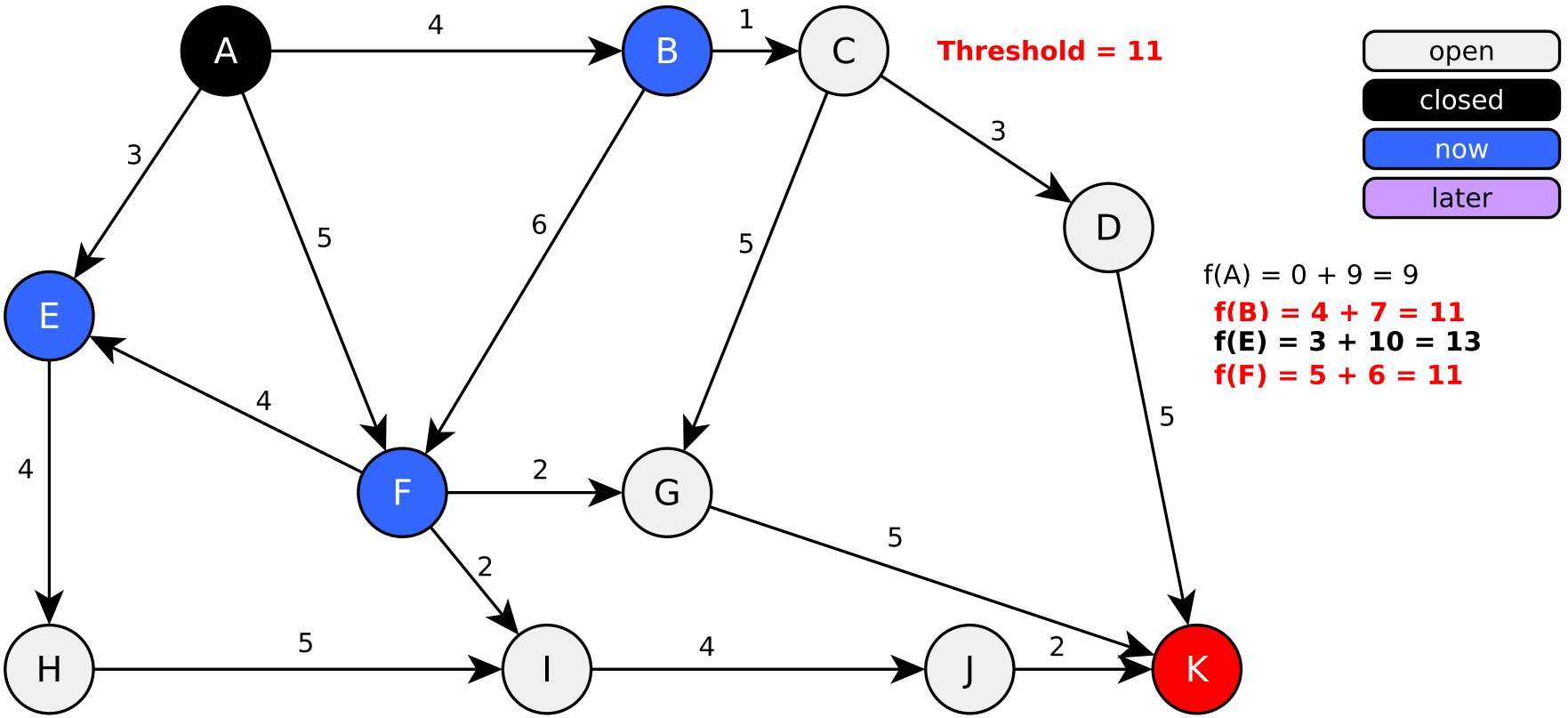
About Fringe Search

- Find a short path between two points
- Not optimal
- Similar to A*
- Uses threshold to determine the most promising nodes
- "Best-first" search with heuristic cost function
- Heuristic cost function h:
  - $h(x) \leq d(x, y) + h(y)$
  - e.g. Manhattan or Euclidean distance

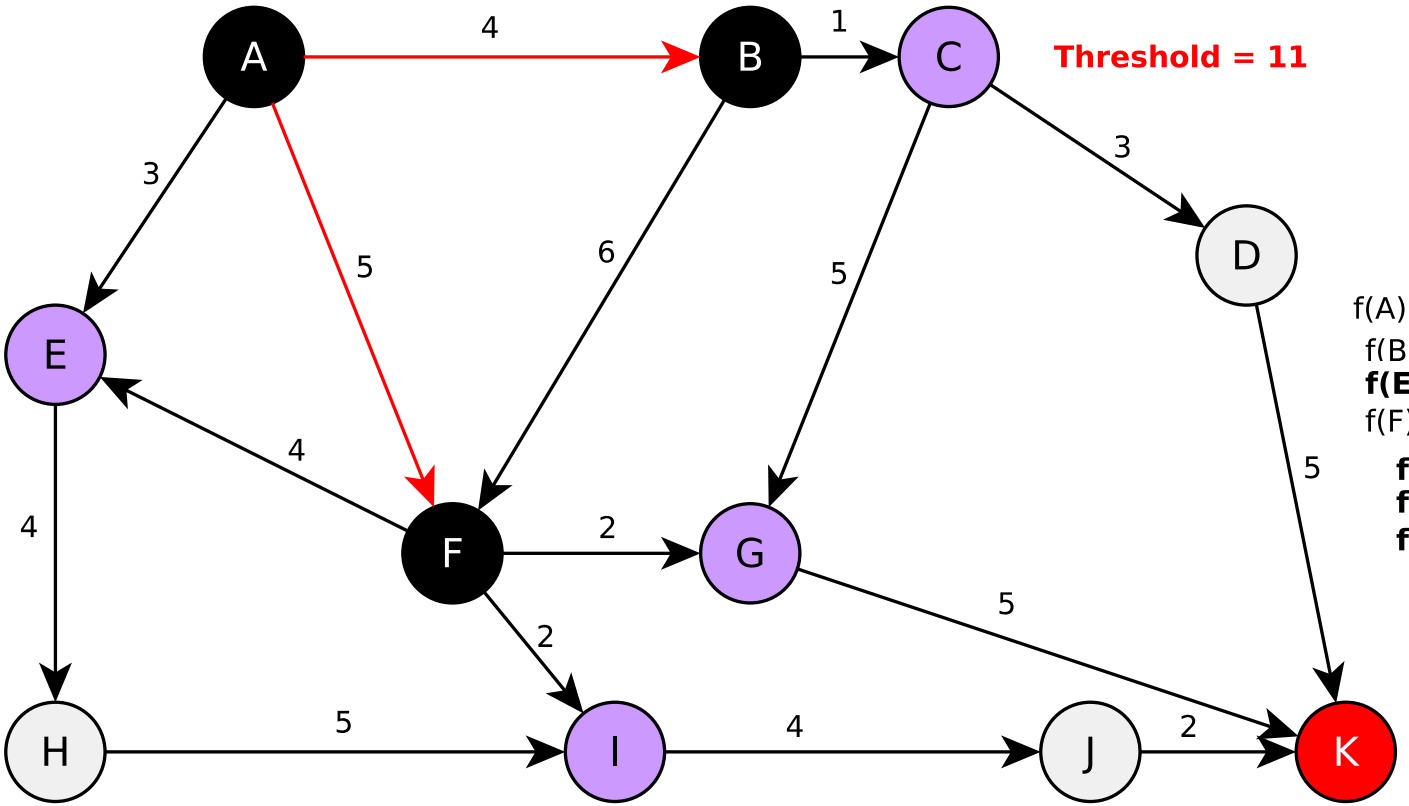
Example 1/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
3
5
6
5
4
5
4
2
2
5
5
4
2
Threshold = 11
f(A) = 0 + 9 = 9
open
closed
now
later

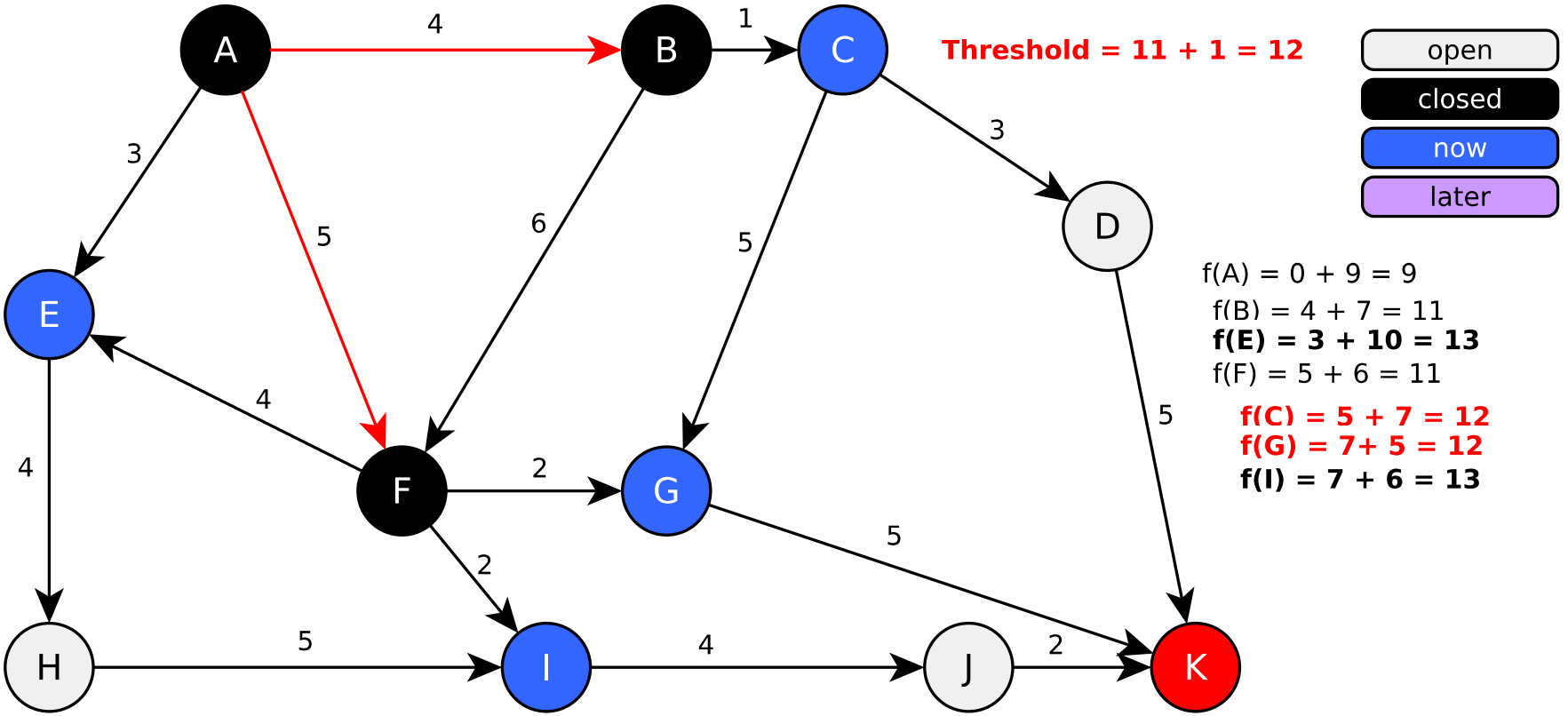
Example 2/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
3
5
6
5
4
4
2
5
2
5
4
2
Threshold = 11
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11

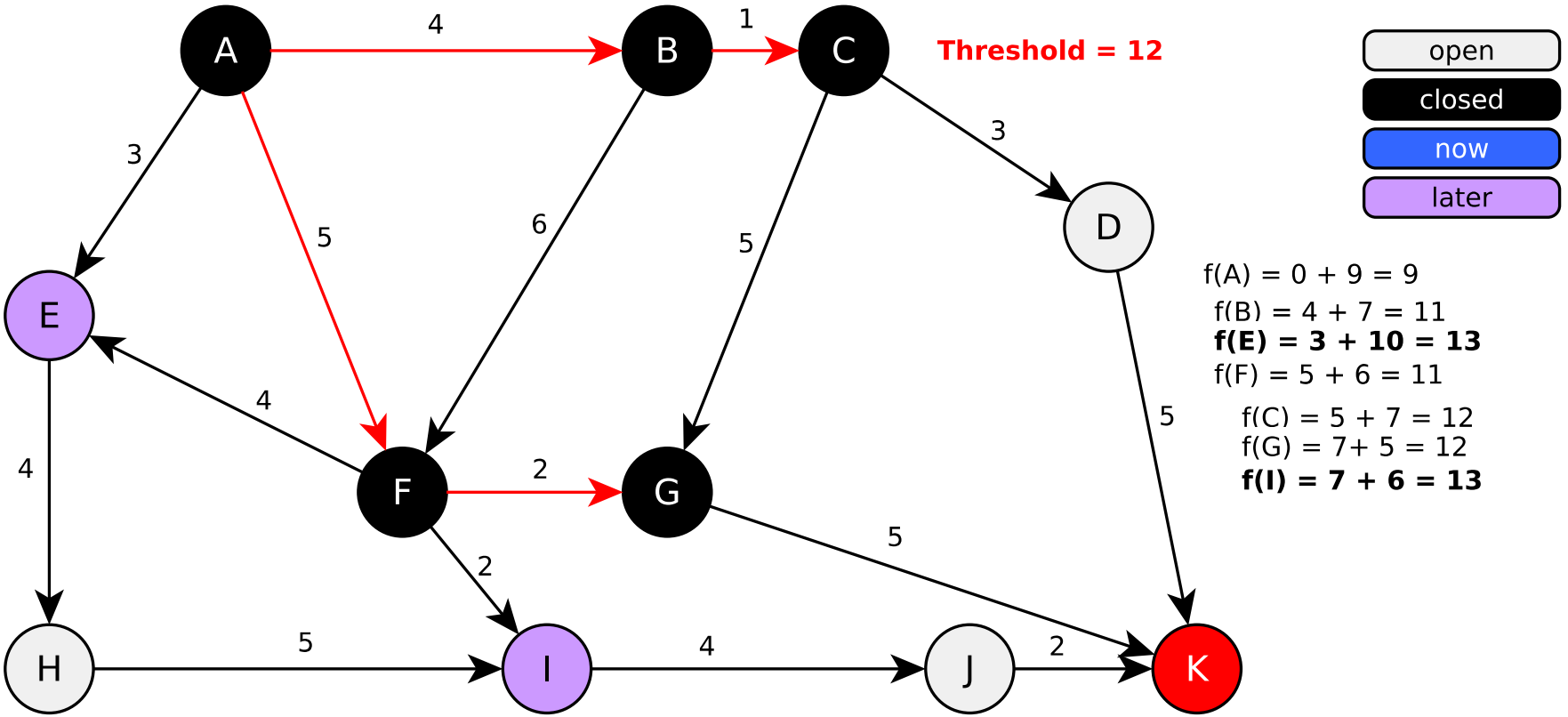
Example 3/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
5
6
5
3
4
4
2
5
2
5
4
2
Threshold = 11
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11
f(C) = 5 + 7 = 12
f(G) = 7+ 5 = 12
f(I) = 7 + 6 = 13

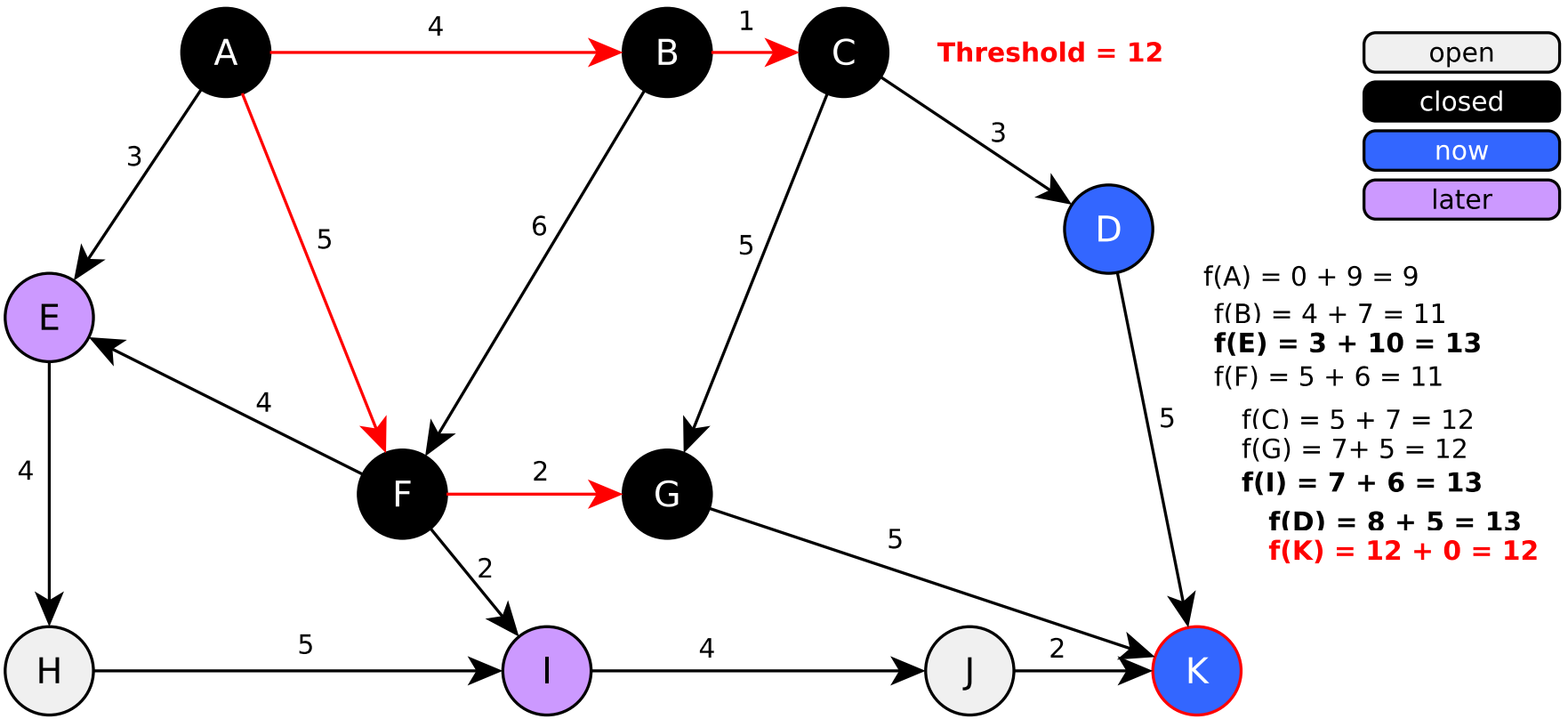
Example 4/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
5
6
5
3
4
4
2
5
2
5
4
2
Threshold = 11 + 1 = 12
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11
f(C) = 5 + 7 = 12
f(G) = 7+ 5 = 12
f(I) = 7 + 6 = 13

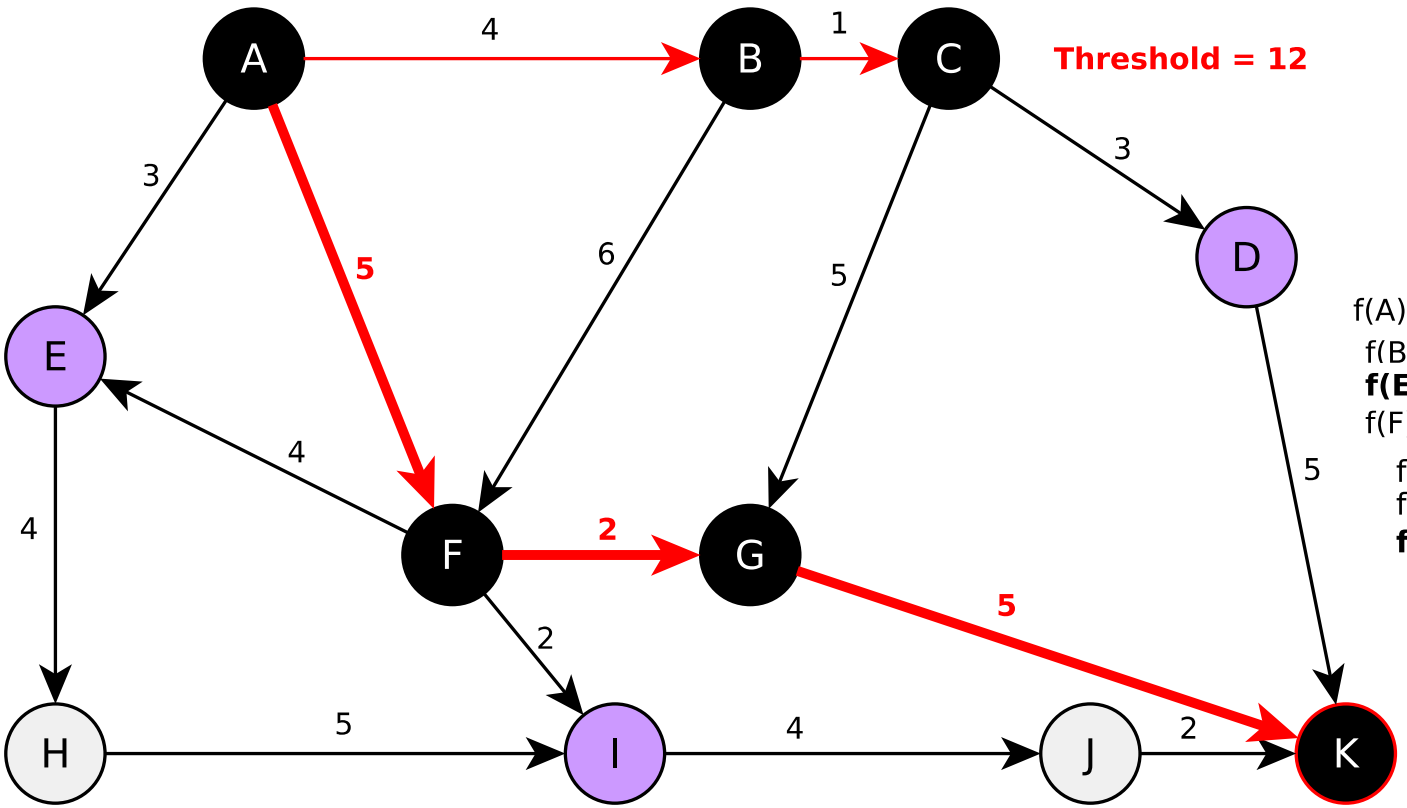
Example 5/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
5
6
5
3
4
4
2
5
2
5
4
2
Threshold = 12
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11
f(C) = 5 + 7 = 12
f(G) = 7+ 5 = 12
f(I) = 7 + 6 = 13

Example 6/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
3
5
6
5
4
4
2
2
5
5
4
2
Threshold = 12
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11
f(C) = 5 + 7 = 12
f(G) = 7+ 5 = 12
f(I) = 7 + 6 = 13
f(D) = 8 + 5 = 13
f(K) = 12 + 0 = 12

Example 7/7


A
B
C
D
E
F
G
H
I
J
K
4
1
3
3
5
6
5
4
5
4
2
2
5
5
4
2
Threshold = 12
open
closed
now
later
f(A) = 0 + 9 = 9
f(B) = 4 + 7 = 11
f(E) = 3 + 10 = 13
f(F) = 5 + 6 = 11
f(C) = 5 + 7 = 12
f(G) = 7+ 5 = 12
f(I) = 7 + 6 = 13
f(D) = 8 + 5 = 13
f(K) = 12 + 0 = 12

Advantages / Disadvantages

- Advantages:
  - "Best-first" search with no sorting
  - Fast
  - Configurable (relaxation function for threshold)
- Disadvantages:
  - Not optimal
  - Bad configuration may leed to bad path
  - Worst case not better then A*

Implementation and Evaluation

- Implementation will be done with MPI
- Evaluation in terms of
  - runtime
  - length of path compared to optimal path
- Different configurations for threshold relaxation

References

Sandy Brand and Rafael Bidarra (2012)

Multi-core scalable and efficient pathfinding with Parallel Ripple Search

Computer Animation and Virtual Worlds, Volume 23, Issue 2 2012, pp 73 – 85.

Sandy Brand (2009)

Efficient obstacle avoidance using autonomously generated navigation meshes

Master Thesis (Delft University of Technology)

The End