CE 191: Civil and Environmental Engineering Systems Analysis

LEC 07 : Dijkstra's Algorithm

Professor Scott Moura
Civil & Environmental Engineering
University of California, Berkeley

Fall 2014

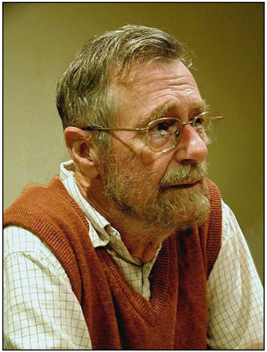
Problem Statement

Question

How do we find integer solutions to shortest-path algorithms?

One Answer

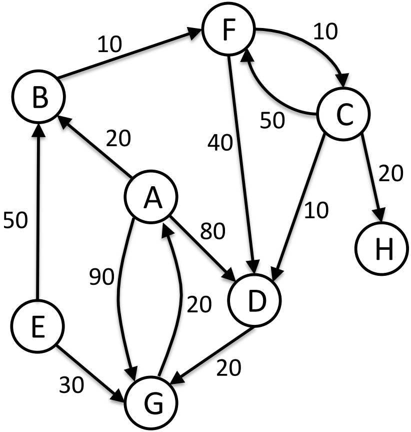
Dijkstra's Algorithm (Polynomial time)



Edsger Wybe Dijkstra, Computer Scientist, 1930 - 2002

Dijkstra’s Algorithm Example - 0

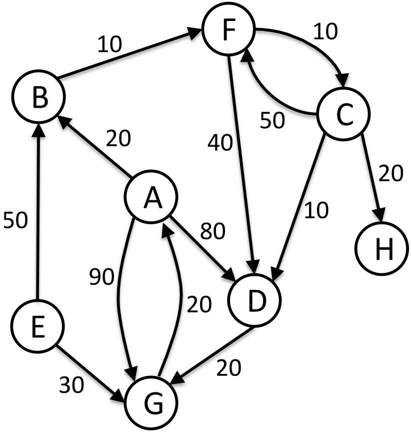
Def’n: A directed graph is a set of nodes connected by edges which have associated directions.



A $\rightarrow$	B	C	D	E	F	G	H
(1)							
(2)							
(2)							
(3)							
(4)							
(5)							
(6)							
(7)							
(8)							

Dijkstra’s Algorithm Example - 1

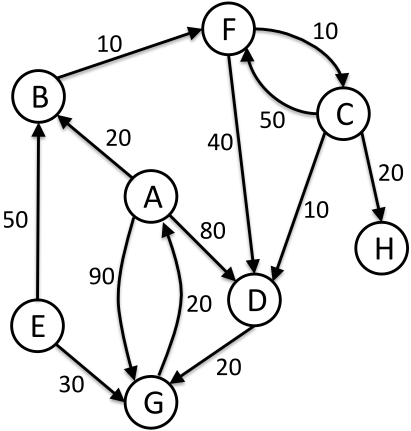
Step 1: Start from A. Assign cost to each node. Infinity for non-connected nodes. Next consider the node with “shortest path distance” from A, which is B.



A $\rightarrow$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2)							
(2)							
(3)							
(4)							
(5)							
(6)							
(7)							
(8)							

Dijkstra's Algorithm Example - 2

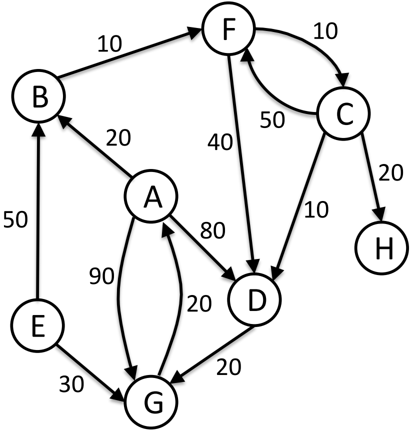
Step 2: Through B, compute cumulative cost to each connected node. Remaining costs are unchanged. Non-connected nodes are assigned ∞. Consider the remaining node with "shortest path distance" from A, which is F.



$A \rightarrow$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3)							
(4)							
(5)							
(6)							
(7)							
(8)							

Dijkstra’s Algorithm Example - 3

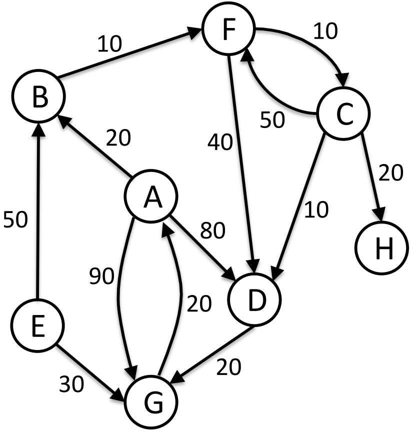
Step 3: Through F, compute cumulative cost to each connected node. Consider the remaining node with “shortest path distance” from A, which is C.



$A \to$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4)							
(5)							
(6)							
(7)							
(8)							

Dijkstra’s Algorithm Example - 4

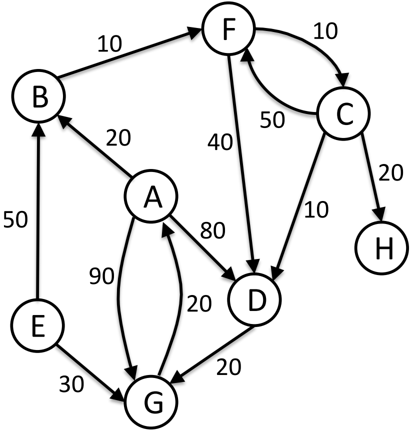
Step 4: Through C, compute cumulative cost to each connected node. Consider the remaining node with “shortest path distance” from A, which is D.



A →	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4) C	20	40	50	∞	30	90	60
(5)							
(6)							
(7)							
(8)							

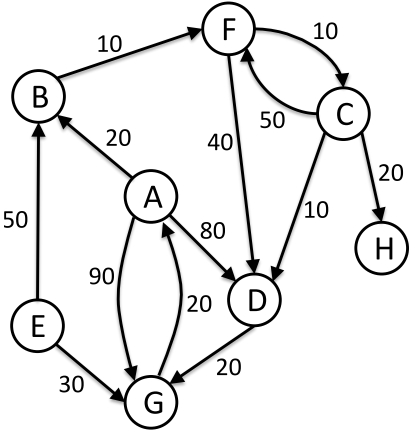
Dijkstra's Algorithm Example - 5

Step 5: Through D, compute cumulative cost to each connected node. Consider the remaining node with "shortest path distance" from A, which is H.



$A \rightarrow$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4) C	20	40	50	∞	30	90	60
(5) D	20	40	50	∞	30	70	60
(6)							
(7)							
(8)							

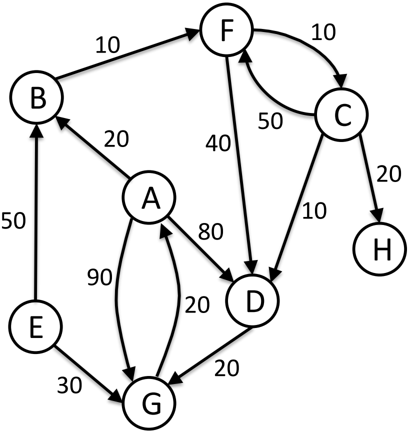
Step 6: Through H, compute cumulative cost to each connected node. *Note no nodes connect from H*. Consider the remaining node with “shortest path distance” from A, which is G.



$A \rightarrow$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4) C	20	40	50	∞	30	90	60
(5) D	20	40	50	∞	30	70	60
(6) H	20	40	50	∞	30	70	60
(7)							
(8)							

Dijkstra's Algorithm Example - 7

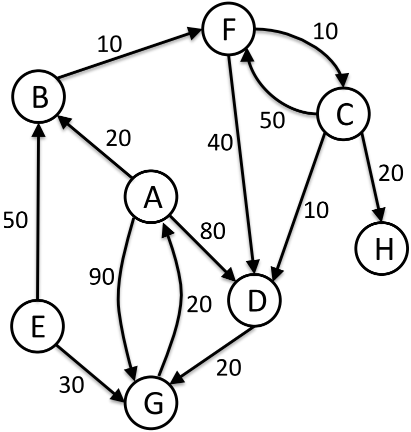
Step 7: Through G, compute cumulative cost to each connected node. Consider the remaining node with "shortest path distance" from A. *Note only E is left, which is unreachable - cost is* ∞*.*



A $\rightarrow$	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4) C	20	40	50	∞	30	90	60
(5) D	20	40	50	∞	30	70	60
(6) H	20	40	50	∞	30	70	60
(7) G	20	40	50	∞	30	70	60
(8)							

Dijkstra’s Algorithm Example - Final Result

Result: Shortest path and distance from A



A →	B	C	D	E	F	G	H
(1) A	20	∞	80	∞	∞	90	∞
(2) B	20	∞	80	∞	30	90	∞
(3) F	20	40	70	∞	30	90	∞
(4) C	20	40	50	∞	30	90	60
(5) D	20	40	50	∞	30	70	60
(6) H	20	40	50	∞	30	70	60
(7) G	20	40	50	∞	30	70	60
(8) E	20	40	50	∞	30	70	60

Summary of Dijkstra's Algorithm

1. Pick initial node (A). Shortest-path to (A) is zero.
2. Assign ∞ to non-connected nodes, path length to connected nodes.
3. Consider unfinished node with shortest-path length from (A), denoted $(\cdot)$.
4. Remove $(\cdot)$ from unfinished set. If unfinished set is empty - done.
5. Compute cumulative cost to each connected node through $(\cdot)$, ∞ otherwise.
6. Go back to Step 3.

Example `dijkstra.m` code on bCourses

Interesting Applications

- Maps.
- Robot navigation.
- Texture mapping.
- Typesetting in TeX.
- Urban traffic planning.
- Network routing protocols.
- Optimal trace routing in PCBs.
- Subroutine in advanced algorithms.
- Telemarketer operating scheduling.
- Routing of telecommunications messages.
- Approximating piecewise linear functions.
- Exploiting arbitrage opportunities in currency exchange.
- Optimal truck routing through given traffic congestion pattern.

Additional Reading

Chapter 1 of Eric V. Denardo, “Dynamic Programming: Models and Applications,” Dover Publications 2003.