Chapter 4
Shortest path problems

Graph theory on September 25, 2023

Huynh Tuong Nguyen, Vo Dang Khoa
Faculty of Information Technology
Industrial University of Ho Chi Minh City
{htnguyen,khoavo}@iuh.edu.vn

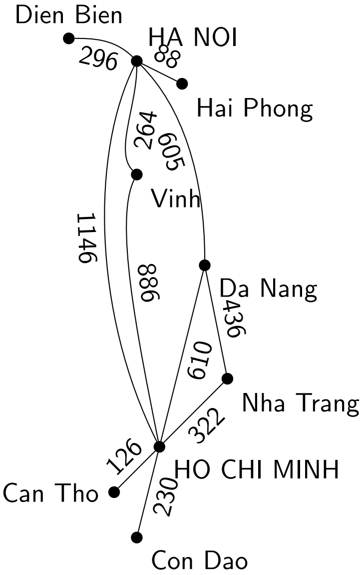
INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

Course outcomes

	Course learning outcomes
CLO.1	Understanding of the basic concepts of graphs
	Special types of graph, computer based graph representation, isomorphism, planar graph, connectivity in graph, graph traversal.
CLO.2	Describe definition of path and circuit
	Identify the existence of Euler path & circuit Identify the existence of Hamilton path & circuit
CLO.3	Compute minimum spanning tree in a (weighted) graph
	Use algorithms: Prim, Kruskal
CLO.4	Determine shortest path in a weighted graph
	Use algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall
CLO.5	Solve maximum flow problem
	Use Ford-Fulkerson's algorithm

Contents

Weighted Graphs



Problem

The problem is also sometimes called the single-pair shortest path problem, to distinguish it from the following generalizations:

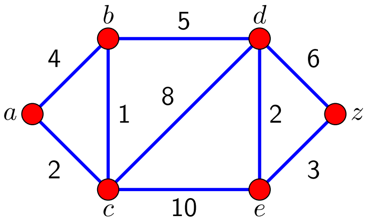
- The single-source shortest path problem, in which we have to find shortest paths from a source vertex v to all other vertices in the graph.
- The single-destination shortest path problem, in which we have to find shortest paths from all vertices in the graph to a single destination vertex v. This can be reduced to the single-source shortest path problem by reversing the edges in the graph.
- The all-pairs shortest path problem, in which we have to find shortest paths between every pair of vertices v, v' in the graph.

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices.

Dijkstra's Algorithm

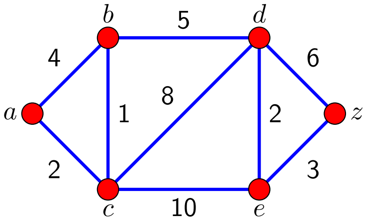
```
procedure Dijkstra(G,a)
// Initialization Step
  forall vertices v
     Label[v] := ∞
     Prev[v] := -1
  endfor
  Label(a) := 0 // a is the source node
  S := ∅

// Iteration Step
  while z ∉ S
     u := a vertex not in S with minimal Label
     S := S ∪ {u}
     forall vertices v not in S
       if (Label[u] + Wt(u,v)) < Label(v)
         then begin
                Label[v] := Label[u] + Wt(u,v)
                Pred[v] := u
              end
  endwhile
```

Example



S	a	b	c	d	e	z
$\emptyset$	**0**	∞	∞	∞	∞	∞
a	0	4	**2**	∞	∞	∞
c	0	**3**	2	10	12	∞
b	0	3	2	**8**	12	∞
d	0	3	2	8	**10**	14
e	0	3	2	8	10	**13**

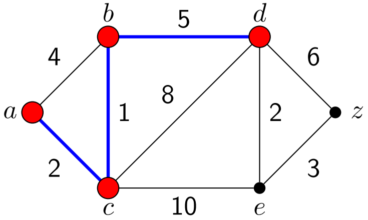
Example



S	a	b	c	d	e	z
$\emptyset$	$\underline{\mathbf{0}}$	∞	∞	∞	∞	∞
a	0	4	$2\ \underline{\mathbf{2}}$	∞	∞	∞
c	0	$3\ \underline{\mathbf{3}}$	2	10	12	∞
b	0	3	2	$8\ \underline{\mathbf{8}}$	12	∞
d	0	3	2	8	$10\ \underline{\mathbf{10}}$	14
e	0	3	2	8	10	$13\ \underline{\mathbf{13}}$

Back tracking procedure

How to determine shortest path from a to d according to Dijkstra's algorithm?



S	a	b	c	d	e	z
$\emptyset$	**0**	∞	∞	∞	∞	∞
a	**0**	4	**2**	∞	∞	∞
c	0	**3**	**2**	10	12	∞
b	0	**3**	2	**8**	12	∞
d	0	3	2	**8**	10	14
e	0	3	2	**8**	10	13

Dijkstra's Algorithm

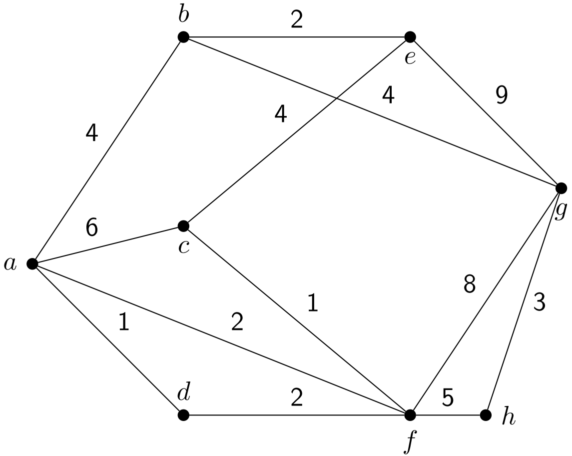
Property

Applicable for any G, any length $\ell(v_i) \geq 0$, $\forall i$; one-to-all; complexity $O(|V|^2)$.

Exercise

Example

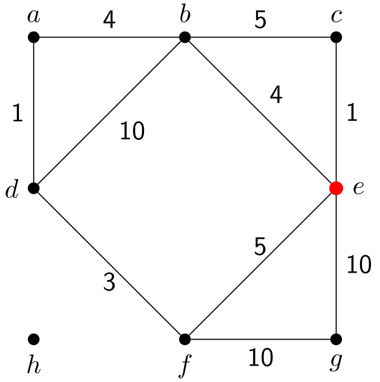
Find the shortest path from b to other vertices using Dijkstra's algorithm.



Exercise

Example

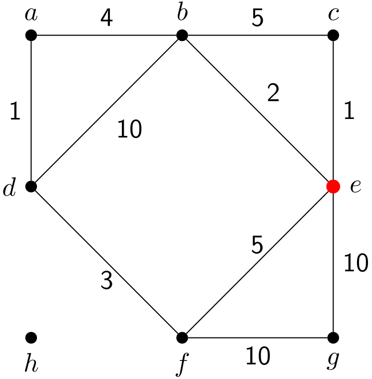
Find the shortest path from e to other vertices using Dijkstra's algorithm.



Exercise

Example

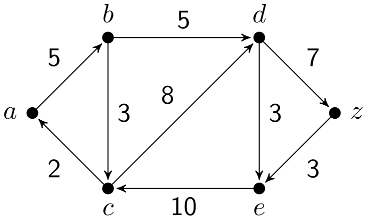
Find the shortest path from e to other vertices using Dijkstra's algorithm.



Exercise

Example

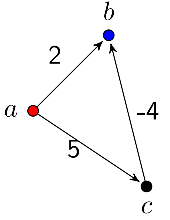
Find the shortest path from a to other vertices using Dijkstra's algorithm.

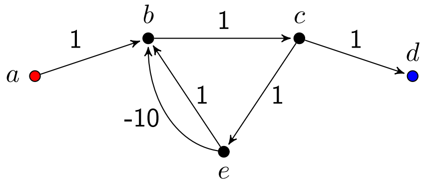


Dijkstra's Algorithm Flaw

Can Dijkstra's Algorithm be used on...

- ...digraph?
  - Yes!
- ...negative weighted graph?
  - No! Why?





Bellman-Ford Algorithm

```
procedure BellmanFord(G,a)
// Initialization Step
  forall vertices v
     Label[v] := ∞
     Prev[v]  := -1
  Label(a) := 0 // a is the source node
// Iteration Step
  for i from 1 to size(vertices)-1
    forall vertices v
      if (Label[u] + Wt(u,v)) < Label[v]
        then
          Label[v] := Label[u] + Wt(u,v)
          Prev[v] := u
// Check circuit of negative weight
     forall vertices v
       if (Label[u] + Wt(u,v)) < Label(v)
         error "Contains circuit of negative weight"
```

Property

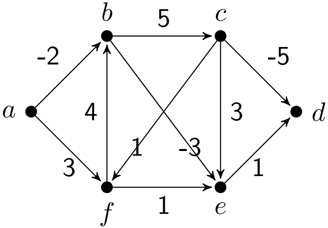
any G, any weighted; one-to-all; detect whether there exists a circle of negative weight; complexity $O(|V| \times |E|)$.

Example

Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	-2a	∞	∞	∞	3a
2	0	-2a	3b	∞	-5b	3a
3	0	-2a	3b	-4e	-5b	3a
4	0	-2a	3b	-4e	-5b	3a

Stop since Step 4 = Step 3.



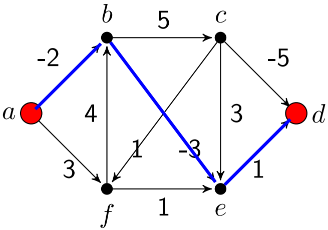
Backtracking procedure

Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	**-2a**	∞	∞	∞	3a
2	0	**-2a**	3b	∞	**-5b**	3a
3	0	-2a	3b	**-4e**	**-5b**	3a
4	0	-2a	3b	**-4e**	-5b	3a

Stop since Step 4 = Step 3.

How to find shortest path from a to d? $a \to b \to e \to d$

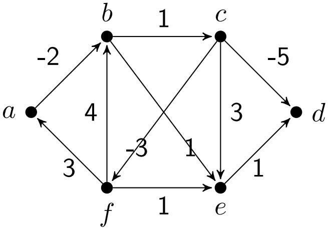


Example

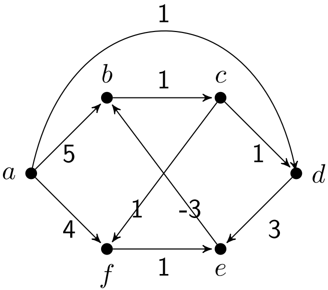
Example

Step	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	-2a	∞	∞	∞	∞
2	0	-2a	-1b	∞	-1b	∞
3	0	-2a	-1b	-6c	-1b	-4c
4	-1f	-2a	-1b	-6c	-3f	-4c
5	-1f	-3a	-1b	-6c	-3f	-4c
6	-1f	-3a	-2b	-6c	-3f	-4c
7	-1f	-3a	-2b	-7c	-3f	-5c

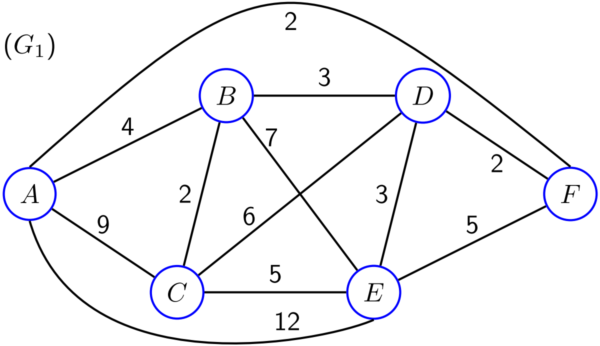
There exists a circle of negative weight since Step 6 $\neq$ Step 5.



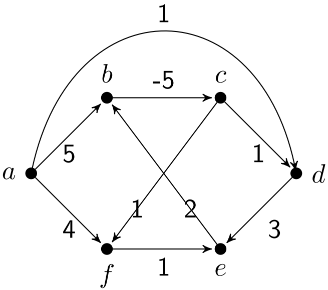
Exercise



Exercise

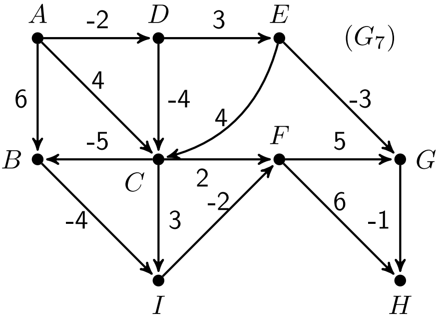


Exercise



INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

Exercise



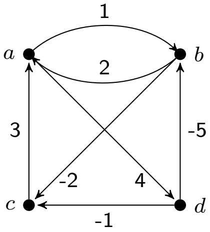
Floyd-Warshall Algorithm [1962]

```
procedure FloydWarshall ()
   for k := 1 to n
      for i := 1 to n
         for j := 1 to n
            path[i,j] = min (path[i,j],
                             path[i,k]+path[k,j]);
```

Property

any G, any weighted; all-to-all; this is an software algorithm; complexity $O(|V|^3)$.

Example



Shortest path from b to d
(5_3 from $L^{(4)}$):
$bd = bc + cd$
($5_3 = -2_0 + 7_1$ from $L^{(3)}$)
$cd = ca + ad$
($7_1 = 3_0 + 4_0$ from $L^{(1)}$)
$\Rightarrow bd = bc + ca + ad$

$$L^{(0)} = \begin{pmatrix} 0_0 & 1_0 & \infty_0 & 4_0 \\ 2_0 & 0_0 & -2_0 & \infty_0 \\ 3_0 & \infty_0 & 0_0 & \infty_0 \\ \infty_0 & -5_0 & -1_0 & 0_0 \end{pmatrix}$$

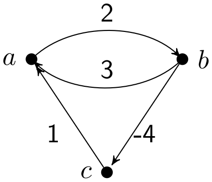
$$L^{(1)} = \begin{pmatrix} 0_0 & 1_0 & \infty_0 & 4_0 \\ 2_0 & 0_0 & -2_0 & 6_1 \\ 3_0 & 4_1 & 0_0 & 7_1 \\ \infty_0 & -5_0 & -1_0 & 0_0 \end{pmatrix}$$

$$L^{(2)} = \begin{pmatrix} 0_0 & 1_0 & -1_2 & 4_0 \\ 2_0 & 0_0 & -2_0 & 6_1 \\ 3_0 & 4_1 & 0_0 & 7_1 \\ -3_2 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

$$L^{(3)} = \begin{pmatrix} 0_0 & 1_0 & -1_2 & 4_0 \\ 1_3 & 0_0 & -2_0 & 5_3 \\ 3 & 4_1 & 0_0 & 7_1 \\ -4_3 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

$$L^{(4)} = \begin{pmatrix} 0_0 & -1_4 & -3_4 & 4_0 \\ 1_3 & 0_0 & -2_0 & 5_3 \\ 3_0 & 2_4 & 0_0 & 7_1 \\ -4_3 & -5_0 & -7_2 & 0_0 \end{pmatrix}$$

Example

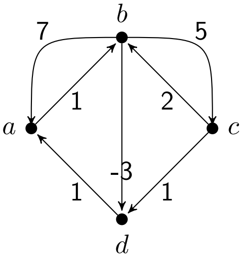


$$L^{(0)} = \begin{pmatrix} 0_0 & 2_0 & \infty_0 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & \infty_0 & 0_0 \end{pmatrix} \quad L^{(1)} = \begin{pmatrix} 0_0 & 2_0 & \infty_0 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & 3_1 & 0_0 \end{pmatrix}$$

$$L^{(2)} = \begin{pmatrix} 0_0 & 2_0 & -2_2 \\ 3_0 & 0_0 & -4_0 \\ 1_0 & 3_1 & -1_2 \end{pmatrix}$$

STOP, there exists a circuit of negative length.

Exercise



Ford's algorithm

$\pi(1) = 0$

For each $j \in V$ do

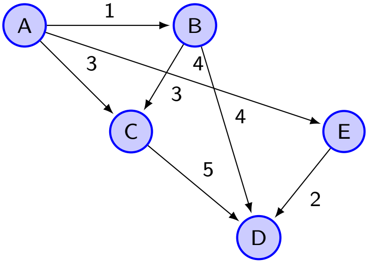
$\pi(j) = \min_{i \in \rho_j^{-1}(\pi(i)+\ell[i,j])}$

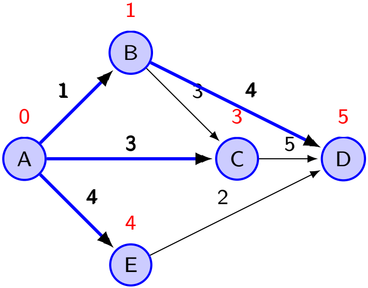
End

Property

G without circle, positive length; one-to-all; rank table definition; complexity $O(|V|)$.

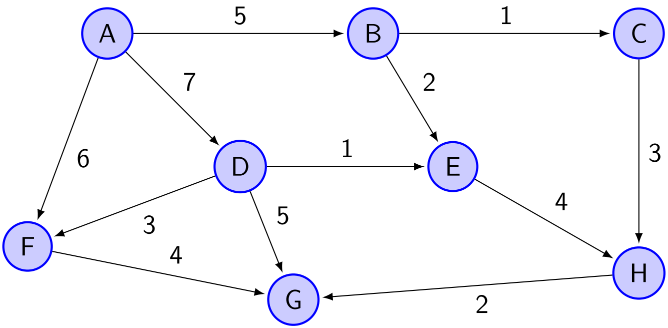
Example





i	Γ_i^{-1}	rank(i)
A	-	0
B	A	1
C	A, B	2
D	B, C , E	3
E	A	1

Exercise



Application

Problem

A young professor in Vinh is invited to teach some years in Ho Chi Minh university of technology. He decides to represent the diverse operations of his transfer by a graph and, in this purpose, establishes the list of following operations:

- A: Find a house in Ho Chi Minh city.
- B: Choose a removal man and sign a contract of move
- C: Make pack his furniture by the removal man
- D: Make transport his furniture towards Ho Chi Minh city
- E: Find an accommodation to HCM (from Vinh)
- F: Transport his family to HCM
- G: Move into his new accommodation
- H: Register the children to their new school
- I: Look for a temporary work for his wife
- J: Fit out the new accommodation and pay this arrangement with the first treatment of his wife
- K: Find a small bar to celebrate in family the success of the move and express the enjoyment to live in a good accommodation arrangement

INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY

Application

Considering constraint of posteriority following: $A < F$; $B < C$; $C < D \wedge F$; $D < G$; $E < F$; $F < G \wedge H \wedge I$; $G < K$; $H < K$; $I < J$; $J < K$.

Approximated task processing times :

A	B	C	D	E	F	G	H	I	J	K
10	2	3	4	7	3	5	1	3	8	2

Question

- Determine the minimal duration needed to completed all tasks.

Question

How to determine a shortest path from u to v in graph G which traverses at most $\leq$ a given constant number of intermediate vertices.

Other shortest path problems

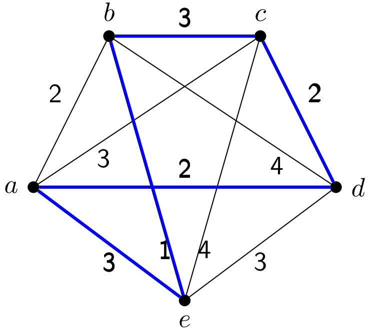
- multicriteria shortest path problem
  - linear combination
  - ε-constraint approach
  - lexico-graphical order
- k shortest paths problem
  - allowing loop
  - loopless
- multi-point shortest path
  - TSP, VRP

Traveling Salesman Problem (TSP)

Problem

- Given a set of n customers located in n cities and distances for each pair of cities, the problem involves finding a round-trip with the minimum traveling cost.
- The vehicle must visit each customer exactly once and return to its point of origin also called depot.
- The objective function is the total cost of the tour.
- $\mathcal{NP}$-complete: all known techniques for obtaining an exact solution require an exponentially increasing number of steps (computing resources) as the problems become larger.
- **TSP is one of the most intensely studied problems in computational mathematics, yet no effective solution method.**

Traveling Salesman Problem



- The total number of possible Hamilton circuit is $(n-1)!/2$.
- For example, if there are 25 customers to visit, the total number of solutions is $24!/2 = 3.1 \times 10^{23}$.
- If the depot is located at node 1, then the optimal tour is $1-5-2-3-4-1$ with total cost equal to 11.

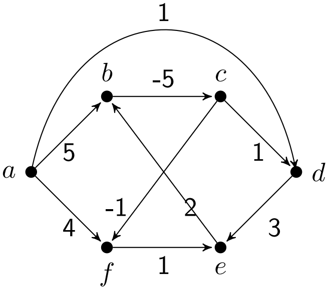
Vehicle Routing Problem (VRP)

Problem

- The vehicle routing problem involves finding a set of trips, one for each vehicle, to deliver known quantities of goods to a set of customers.
- The objective is to minimize the travel costs of all trips combined.
- There may be upper bounds on the total load of each vehicle and the total duration of its trip.
- The most basic Vehicle Routing Problem (VRP) is the single-depot capacitate VRP.

Exercise

Determine a shortest path from a to other vertices in the following graph.



INDUSTRIAL UNIVERSITY OF HOCHIMINH CITY