The shortest path problem

Raimondas Čiegis

Matematinio modeliavimo katedra, email: rc@vgtu.lt

April 25 d., 2025

In this lecture, we will consider in detail the problem of finding the shortest path between the vertices of a graph. This is a very frequently solved logistics problem, and it is important to examine the main methods for solving it

First, we will present the most important definitions, many of which you have already studied in discrete mathematics lectures.

Graphs

Let us have a set of vertices $V = \left\{v_1, v_2, \ldots, v_N\right\}$ and a set of edges $E = \left\{e_1, e_2, \ldots, e_K\right\}$.

An edge is a pair of vertices $e_j = (v_{1j}, v_{2j})$.

Graph is denoted by $G = (V, E)$.

Graphs

Let us have a set of vertices $V = \{v_1, v_2, \dots, v_N\}$ and a set of edges $E = \{e_1, e_2, \dots, e_K\}$.

An edge is a pair of vertices $e_j = (v_{1j}, v_{2j})$.

Graph is denoted by $G = (V, E)$.

The simplest example of a graph is a map of a country's roads: cities and towns form a set of vertices, and roads form a set of edges.

If the edges $e_j = (v_{1j}, v_{2j})$ and $e_k = (v_{2j}, v_{1j})$ are different (the direction of the connection is also important), then they are called *directed*, and the graph consisting of such edges is called *directed* graph.

If the edges $e_j = (v_{1j}, v_{2j})$ and $e_k = (v_{2j}, v_{1j})$ are different (the direction of the connection is also important), then they are called *directed*, and the graph consisting of such edges is called *directed* graph.

On city roads, we also encounter a similar situation where only one-way traffic is allowed on the street.

Real numbers can be assigned to the edges of a graph that evaluate distance, time, weight and similar attributes.

Such a graph is called *weighted*.

We will denote the weight of the edge $e_j \in E$ by $w(e_j)$.

Real numbers can be assigned to the edges of a graph that evaluate distance, time, weight and similar attributes.

Such a graph is called *weighted*.

We will denote the weight of the edge $e_j \in E$ by $w(e_j)$.

The set of neighbors of vertex v is defined by

$$N(v) = \{\, u : \ \ u \in V,\ (u, v) \in E \text{ or } (v, u) \in E \,\}$$

and it is called the *neighborhood* of vertex v.

The degree of a vertex v is denoted by $\deg(v)$ and it is equal to the number of its neighbors.

Examples of some important cases of graphs are presented in this figure .

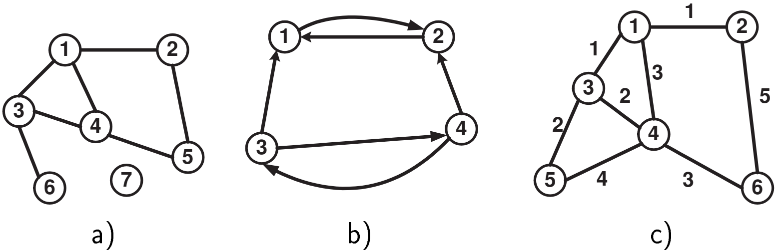

FIGURE: Examples of graphs: a) undirected graph, $|V| = 7$, $|E| = 7$, the degree of vertices v_1, v_3, v_4 is equal to 3, the degree of vertices v_2, v_5 is equal to 2, v_6 and end vertex, v_7 is an isolated vertex, b) directed graph, $|V| = 4$, $|E| = 6$, c) weighted graph, $|V| = 6$, $|E| = 8$.

A set of vertices

$$p = \left\{ v_{i_0},\ v_{i_1},\ \ldots,\ v_{i_k} \right\}$$

is called *a path* if all adjacent vertices are connected by edges, i.e.

$$(v_{i_j}, v_{i_{j+1}}) \in E, \quad j = 0, 1, \ldots, k-1 \,.$$

A set of vertices

$$p = \left\{ v_{i_0},\ v_{i_1},\ \ldots,\ v_{i_k} \right\}$$

is called *a path* if all adjacent vertices are connected by edges, i.e.

$$(v_{i_j}, v_{i_{j+1}}) \in E, \quad j = 0, 1, \ldots, k-1\,.$$

The graph is called *connected* if for each pair of vertices x and y of the graph, there is a path joining x and y.

A set of vertices

$$p = \left\{ v_{i_0},\ v_{i_1},\ \ldots,\ v_{i_k} \right\}$$

is called *a path* if all adjacent vertices are connected by edges, i.e.

$$(v_{i_j}, v_{i_{j+1}}) \in E, \quad j = 0, 1, \ldots, k-1 .$$

The graph is called *connected* if for each pair of vertices x and y of the graph, there is a path joining x and y.

For example, if we have a road map and the graph is connected, then it is possible to drive from any city or settlement to another one.

During spring floods, some settlements become inaccessible.

Let's consider a weighted graph. The length of a path p is defined as

$$W(p) = \sum_{j=0}^{k-1} w\left(v_{i_j}, v_{i_{j+1}}\right).$$

In the case where the weights of the edges of a graph are not specified, the path length is the number of edges on the path.

Let's consider a weighted graph. The length of a path p is defined as

$$W(p) = \sum_{j=0}^{k-1} w\left(v_{i_j}, v_{i_{j+1}}\right).$$

In the case where the weights of the edges of a graph are not specified, the path length is the number of edges on the path.

The shortest path connecting two vertices $a, b \in V$ of the graph G is the path

$$p = \left\{a, \; v_{i_1}, \; \dots, \; v_{i_k}, b\right\},$$

satisfying the condition

$$W(p) \leqslant W(p^{'}),$$

where $p^{'}$ is any other path connecting a and b.

Let's consider a weighted graph. The length of a path p is defined as

$$W(p) = \sum_{j=0}^{k-1} w\left(v_{i_j}, v_{i_{j+1}}\right).$$

In the case where the weights of the edges of a graph are not specified, the path length is the number of edges on the path.

The shortest path connecting two vertices $a, b \in V$ of the graph G is the path

$$p = \left\{a,\ v_{i_1},\ \dots,\ v_{i_k}, b\right\},$$

satisfying the condition

$$W(p) \leqslant W(p^{'}),$$

where $p^{'}$ is any other path connecting a and b.
If weights of all edges are positive numbers, then the shortest path always exist.

We will solve the problem of finding the shortest paths between one vertex s of a graph G and all other vertices $v \in V$ of this graph.

We will solve the problem of finding the shortest paths between one vertex s of a graph G and all other vertices $v \in V$ of this graph.

It is clear that the shortest path exists only to vertices that are reachable from s.

We will solve the problem of finding the shortest paths between one vertex s of a graph G and all other vertices $v \in V$ of this graph.

It is clear that the shortest path exists only to vertices that are reachable from s.

For graphs with different structure, special efficient shortest path finding algorithms are developed that best utilize information about the graph structure.

Now let's introduce some more notations.

Now let's introduce some more notations.

Let $d[v]$ denote the length of the best path already found from vertex s to vertex v (this onformation is dynamicaly updated during search algorithm).

Now let's introduce some more notations.

Let $d[v]$ denote the length of the best path already found from vertex s to vertex v (this onformation is dynamicaly updated during search algorithm).

Let $\pi[v]$ denote the vertex from which we reach the vertex v in the already known shortest path.

Now let's introduce some more notations.

Let $d[v]$ denote the length of the best path already found from vertex s to vertex v (this onformation is dynamicaly updated during search algorithm).

Let $\pi[v]$ denote the vertex from which we reach the vertex v in the already known shortest path.

Let $\delta(s, v)$ denote the length of the shortest path from vertex s to vertex v.

In graphs (as in real road maps), the triangle inequality is not necessary valid. It states that the length of the edge connecting two vertices a and b is not greater than the length of the path passing through one (or several) intermediate vertices.

In graphs (as in real road maps), the triangle inequality is not necessary valid. It states that the length of the edge connecting two vertices a and b is not greater than the length of the path passing through one (or several) intermediate vertices.

However, it is easy to prove two important properties of shortest paths, they make a basis for algorithms that are used to construct the shortest paths.