Graphs and Optimization

21

Graphs

Graph theory is an important tool for tackling complex problems in various scientific fields. The interdisciplinary nature of graph theory stems from the structure of graphs. Graph theory is applicable whenever a problem involves pairs of objects that are in relation to each other. Examples include road networks, electrical networks, and flow diagrams. In this chapter, fundamental concepts of graph theory are discussed.

21.1 Basic Concepts

A graph is a pair $G = (V, E)$ consisting of a non-empty set V and a set E of 2-element subsets of V . The elements of V are called vertices and the elements of E are called edges. An edge $e = \{u, v\}$ is also written as a word $e = uv$ (or $e = vu$). If an edge $e = uv$ exists, then u and v are incident with e, u and v are adjacent (or directly connected), and u and v are the endpoints of e.

In the following, only finite graphs are considered, i.e., graphs with a finite vertex set. The number of vertices of a graph G is called the order of G and the number of edges of G is called the size of G.

A graph can be represented by a diagram in which vertices are represented by points in the plane and edges by continuous line segments.

ex 21.1. The graph G with vertex set $V = \{v_1, ..., v_4\}$ and edge set $E = \{v_1v_3, v_2v_3, v_2v_4, v_3v_4\}$ is represented by the diagram in Fig. 21.1.

A graph $G = (V, E)$ has neither loops nor multiple edges. Loops are 1-element subsets of V , i.e., edges that are incident with only one vertex. Multiple edges are multisets of 2-element subsets of V , i.e., multiple connections between two vertices.

TODO

Fig. Fig. 21.1. A diagram of the graph in 21.1.

Let $G = (V, E)$ be a graph. The number of edges incident with a vertex $v \in V$ is called the degree of v and denoted by $d(v)$. If $d(v) = 0$, then v is isolated. If all vertices in G have the same degree k, then G is called k-regular.

Theorem 21.2. (Handshaking Lemma) For every graph $G = (V, E)$ it holds

$$\sum_{\{u \in V\}} d(v) = 2|E|$$

Proof. Each edge is counted twice in the sum on the left-hand side, once for each incident vertex. □

Corollary 21.3. In every graph, the number of vertices with odd degree is always even.

Proof. According to the handshaking lemma, the sum of all degrees is even. By subtracting the sum of all even degrees, an even number is obtained, which represents the sum of all odd degrees. Therefore, the sum of all odd degrees must have an even number of summands. □

Example 21.4. Can 333 telephones be connected such that each telephone is directly connected to three other telephones? The answer is no, because the sum of the degrees of the telephone network would be odd ($333 \cdot 3$).

The degree sequence of a graph G is the decreasingly sorted sequence of the degrees of all vertices in G. For example, the graph in Fig. 21.1 has the degree sequence $(3, 2, 2, 1)$. Conversely, not every decreasing sequence of natural numbers corresponds to a graph. For example, there is no graph with degree sequence $(5, 3, 2, 2, 2, 1)$, because the sum of the degrees is odd.

Subgraphs

Let $G = (V, E)$ be a graph. A subgraph of G is a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$. A subgraph G' of G can be viewed as the subgraph of G induced by its edge set E'. A subgraph G' of G is called the subgraph induced by its vertex set V'if every edge in G connecting two vertices in G' belongs to G'.

Example 21.5. In Fig. 21.2, a graph G along with two of its subgraphs G_1 and G_2 is shown. The subgraph G_2 is induced by the vertex set $\{b, c, d\}$, while the subgraph G_1 is not.

TODO

Fig. 21.2. A graph G with two subgraphs.

Isomorphisms

Let $G = (V, E)$ and $G' = (V', E')$ be graphs. A mapping $\phi : V \to V'$ is called an isomorphism from G to G' if ϕ is bijective and for all $u, v \in V$ it holds $uv \in E$ if and only if $\phi(u)\phi(v) \in E'$. Two graphs G and G' are called isomorphic if there exists an isomorphism from G to G'.

Example 21.6. The two graphs in Fig. 21.3 are isomorphic, because an isomorphism is given by $\phi(a) = 1$, $\phi(b) = 2$, $\phi(c) = 3$ and $\phi(d) = 4$.

Isomorphic graphs have the same number of vertices (order), the same number of edges (size), and the same degree sequence. In other words, two graphs are not isomorphic if they have different order, size, or degree sequence. There exist non-isomorphic graphs with the same degree sequence (Fig. 21.4).

TODO

Fig. 21.3. Two isomorphic graphs.

TODO

Fig. 21.4. Two non-isomorphic graphs with the same degree sequence.

Let G be a graph. An isomorphism from G to G is also called an automorphism of G.

Theorem 21.7. The set of all automorphisms of a graph forms a group under composition. The group of all automorphisms of G is called the automorphism group of G and denoted by $\mathrm{Aut}(G)$.

Example 21.8. The automorphism group of a square (Fig. 21.5) is the dihedral group D4 according to 15.30, consisting of four rotations id, (1234), (1234)2 = (13)(24), (1234)3 = (1432) and four reflections (12)(34), (14)(23), (13), (24).

21.2 Paths, Cycles and Connectivity

Let $G = (V, E)$ be a graph. A sequence $W = (v_0, ..., v_k)$ of vertices $v_i \in V$ is called a path in G if for all $i, 1 \leq i \leq k$, it holds $v_{i-1}v_i \in E$. The vertex v_0 is the starting vertex and the vertex v_k is the ending vertex of W. The length of W is n, the number of its edges. A path W is called simple if W does not contain any vertex more than once.

Example 21.9. The graph in Fig. 21.6 contains, for example, the simple paths (s, a, d, g, f, i, t) and (s, a, b, e, h, i, t) of length 6.

A cycle in G is a path in G in which the starting and ending vertices are identical. A cycle is called simple if it does not contain any vertex more than once (except for the starting and ending vertex). A back-and-forth traversal of an edge uv results in a simple cycle (u, v, u) of length 2.

Example 21.1. The graph in Fig. 21.6 contains, for example, the simple cycles (a, b, c, h, e, d, a) and (a, b, e, f, g, d, a) of length 6.

Connectivity

Let $G = (V, E)$ be a graph. Two vertices $u, v \in V$ are called connected in G, briefly $u \underset{G}{\equiv} v$, if u = v or there exists a path from u to v. If every pair of vertices in G is connected, then G is called connected.

Lemma 21.11. Let G = (V, E) be a graph. The connectivity $\underset{G}{\equiv}$ in G is an equivalence relation on V.

The equivalence classes of connectivity form, according to Theorem 5.5, a partition of V . The subgraphs spanned by the equivalence classes are called components of G. If G is connected, then there is only one component.

TODO

Fig. Fig. 21.6. A road network

Example 21.12. The graph in Fig. 21.7 consists of two components, spanned by the equivalence classes {a, b} and {c, d, e}.

Theorem 21.13. Let G = (V, E) be a connected graph and e ∈ E an edge lying on a simple cycle in G. The subgraph of G obtained by removing e is also connected.

Proof. Let G′ denote the subgraph of G obtained by removing e. Let $u, v \in V$. Since G is connected, there exists a path W in G from u to v. If the path does not use the edge e, then W is also a path in G'. Otherwise, W can be modified so that instead of the edge e, the rest of the cycle on which e lies according to assumption is used. □

Distances

Trees

Bipartite Graphs

21.3 Planar Graphs

21.4 Data Structures and Algorithms

Self-test Problems

22

Networks

22.1 Shortes Paths

Shortest Paths Between Any Two Vertices

Shortest Paths from One to All Others

22.2 Minimum Spanning Trees

Kruskal's Algorithm

A Criterion for Minimum Spanning Trees

22.3 Maximum Flows

Flows in Flow Networks

Cuts in Flow Networks

22.4 Theorems of Hall, König-Egerváry and Menger

The Theorem of Menger

The theorem of König and Egerváry

The Theorem of Hall

Self-test Problems

23

Combinatorial Optimization

23.1 Complexity Classes

Complexity of Decision Problems

23.1.1.1 Complexity of Optimization Problems

23.2 Backtracking Algorithms

Combinatorial Optimization

Backtracking Procedures

Pruning Subtrees

Bounding Functions

The Traveling Salesman Problem

23.3 Heuristic Algorithms

The Hill-Climbing Method

Simulated Annealing

Genetic Algorithms

23.4 Greedy Algorithms and Matroids

Greedy Algorithms

Matroids

23.5 Vertex Coloring

Self-test Problems

24

Linear Optimization

24.1 Examples

Knapsack Problem

Matching Problem

Network Problem

24.2 Linear Programs and Duality

Linear Programs

24.3 The Feasible Region and the Role of Vertices

24.4 Integer Programming

24.5 The Simplex Method

Disclaimer

This automated translation is provided “as is” without warranties of any kind. The translator and provider make no representations about the accuracy, completeness, or suitability of the translation for any purpose. Use of this material is at your own risk. The original work remains under copyright protection, and this translation is provided for non-commercial academic use only.