Cambridge University Press
978-1-108-49260-7 — Topics in Algorithmic Graph Theory
Edited by Lowell W. Beineke , Martin Charles Golumbic , Robin J. Wilson
Excerpt
More Information

Preliminaries

LOWELL W. BEINEKE, MARTIN CHARLES GOLUMBIC and ROBIN J. WILSON

1. Graph theory

This section presents the basic definitions, terminology and notation of graph theory, along with some fundamental results. Further information can be found in the many standard books on the subject – for example, Bondy and Murty [1], Chartrand, Lesniak and Zhang [2], Golumbic [4], Gross and Yellen [5] or West [7], or, for a simpler treatment, Even [3], Marcus [6] or Wilson [8].

Graphs

A *graph* G is a pair of sets (V, E), where V is a finite non-empty set of elements called *vertices*, and E is a finite set of elements called *edges*, each of which has two associated vertices. The sets V and E are the *vertex-set* and *edge-set* of G, and are sometimes denoted by $V(G)$ and $E(G)$. The number of vertices in G is called the *order* of G and is usually denoted by n (but sometimes by $|G|$ or $|V(G)|$); the number of edges is denoted by m. A graph with only one vertex and no edges is called *trivial*.

An edge whose vertices coincide is a *loop*, and if two edges have the same pair of associated vertices, they are called *multiple edges*. In this book, unless otherwise specified, graphs are assumed to have no loops or multiple edges; that is, they are taken to be *simple*. Hence, an edge e can be considered as its associated pair of vertices, $e = \{v, w\}$, usually shortened to vw. An example of a graph of order 5 is shown in Fig. 1(a).

The *complement* $\overline{G}$ of a graph G has the same vertices as G, but two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G. Figure 1(b) shows the complement of the graph in Fig. 1(a).

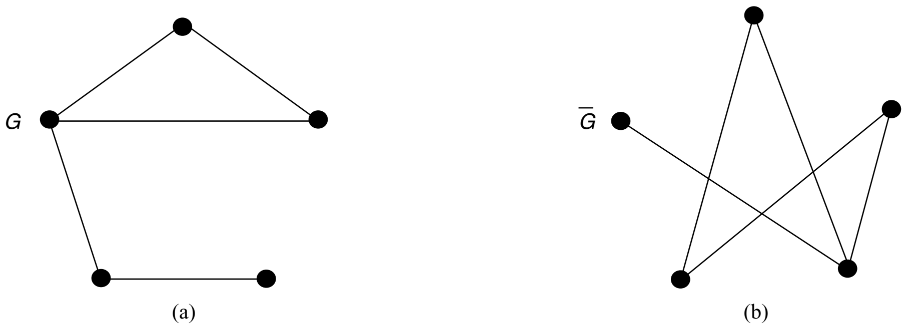

Fig. 1. A graph and its complement

Adjacency and degrees

The vertices of an edge are its *endpoints* or *ends*, and the edge is said to *join* these vertices. An endpoint of an edge and the edge are *incident* with each other. Two vertices that are joined by an edge are called *neighbours* and are said to be *adjacent*; if v and w are adjacent vertices, we sometimes write $v \sim w$, and if they are not adjacent we write $v \nsim w$. Two edges are *adjacent* if they have a vertex in common.

The set $N(v)$ of *neighbours* of a vertex v is called its *neighbourhood*. If $X \subset V$, then $N(X)$ denotes the set of vertices not in X that are adjacent to some vertex of X. The *closed neighbourhood* of a vertex v is defined as $N[v] = N(v) \cup \{v\}$. Two vertices v and w are *true twins* if $N[v] = N[w]$ and are *false twins* if $N(v) = N(w)$.

The *degree* $\deg v$, or $d(v)$, of a vertex v is the number of its neighbours; in a non-simple graph, it is the number of occurrences of the vertex as an endpoint of an edge, with loops counted twice. A vertex of degree 0 is an *isolated vertex* and one of degree 1 is a *pendant vertex*. A graph is *regular* if all of its vertices have the same degree, and is *k-regular* if that degree is k; a 3-regular graph is sometimes called *cubic*. The maximum degree in a graph G is denoted by $\Delta(G)$ or just Δ, and the minimum degree by $\delta(G)$ or δ. The *degree sequence* of a graph is the non-increasing sequence of its vertex degrees, for example, $[3, 2, 2, 2, 1]$ in both Fig. 1(a) and Fig. 1(b), although they are not the same graph. Determining whether a given sequence of numbers is the degree sequence of a simple graph can be done using an algorithm by Havel and Hakimi or a characterization theorem of Erdős and Gallai.

Isomorphisms, automorphisms and homomorphisms

An *isomorphism* between two graphs G and H is a bijection between their vertex-sets that preserves both adjacency and non-adjacency. The graphs G and H are *isomorphic*, written $G \cong H$, if there exists an isomorphism between them.

An *automorphism* of a graph G is an isomorphism of G with itself. The set of all automorphisms of a graph G forms a group, called the *automorphism group* of G and denoted by Aut(G).

A *homomorphism* of a graph G to a graph H is a mapping of the vertex-set of G to the vertex-set of H that preserves adjacency (but not necessarily non-adjacency). The graph G is *homomorphic* to H if there exists such a homomorphism. Graph homomorphisms are the subject of Chapter 13.

Walks, paths and cycles

A *walk* in a graph is a sequence of vertices and edges $v_0, e_1, v_1, \ldots, e_k, v_k$, in which each edge e_i joins the vertices v_{i-1} and v_i. This walk is said to *go from* v_0 *to* v_k or to *connect* v_0 *and* v_k, and is called a v_0–v_k walk. It is frequently shortened to $v_0 v_1 \cdots v_k$, for a simple graph. A walk is *closed* if the first and last vertices are the same. Some important types of walk are the following:

- a *path* is a walk in which no vertex is repeated;
- a *cycle* is a non-trivial closed walk in which no vertex is repeated, except the first and last;
- a *trail* is a walk in which no edge is repeated;
- a *circuit* is a non-trivial closed trail.

Connectedness and distance

A graph is *connected* if it has a path connecting each pair of vertices, and *disconnected* otherwise. A (*connected*) *component* of a graph is a maximal connected subgraph.

The number of occurrences of edges in a walk is called its *length*, and in a connected graph, the *distance* $d(v, w)$ from v to w is the length of a shortest v–w path. It is easy to check that distance satisfies the properties of a metric. The *diameter* of a connected graph G is the greatest distance between any pair of vertices in G. If G has a cycle, the *girth* of G is the length of a shortest cycle.

A connected graph is *Eulerian* if it has a closed trail containing all of its edges; such a trail is an *Eulerian trail*. The following statements are equivalent for a connected graph G:

- G is Eulerian;
- every vertex of G has even degree;
- the edge-set of G can be partitioned into cycles.

A graph of order n is *Hamiltonian* if it has a cycle containing all of its vertices, and is *pancyclic* if it has a cycle of every length from 3 to n. It is *traceable* if it has a path containing all of its vertices. No ‘good’ characterizations of these properties are known.

Bipartite graphs and trees

If the set of vertices of a graph G can be partitioned into two non-empty subsets so that no edge joins two vertices in the same subset, then G is *bipartite*. The two subsets are called *partite sets* and, if they have orders r and s, G is an $r \times s$ *bipartite graph*. (For convenience, the trivial graph is also called bipartite.) Bipartite graphs are characterized by having no cycles of odd length.

Among the bipartite graphs are *trees*, those connected graphs with no cycles. Any graph without cycles is a *forest*; thus, each component of a forest is a tree. Trees have been characterized in many ways, some of which we give here. For a graph G of order n, the following statements are equivalent:

- G is a tree;
- G is connected and has no cycles;
- G is connected and has $n - 1$ edges;
- G has no cycles and has $n - 1$ edges;
- G has exactly one path between any two vertices.

The set of trees can also be defined inductively: a single vertex is a tree; and for $n \geq 1$, the trees with $n + 1$ vertices are those graphs obtainable from some tree with n vertices by adding a new vertex adjacent to precisely one of its vertices.

This definition has a natural extension to higher dimensions. The *k-dimensional trees*, or *k-trees* for short, are defined as follows: the complete graph on k vertices is a k-tree, and for $n \geq k$, the k-trees with $n + 1$ vertices are those graphs obtainable from some k-tree with n vertices by adding a new vertex adjacent to k mutually adjacent vertices in the k-tree. Figure 2 shows a tree and a 2-tree. An important concept in the study of graph minors (introduced later) is the *tree-width* of a graph G, the minimum dimension of any k-tree that contains G as a subgraph.

Fig. 2. A tree and a 2-tree

Special graphs

We now introduce some individual types of graph:

- the *complete graph* K_n has n vertices, each adjacent to all the others; a complete graph is often called a *clique*;

- the *null graph* $\overline{K}_n$ has n vertices and no edges;
- the *path graph* P_n consists of the vertices and edges of a path of length $n-1$;
- the *cycle graph* C_n consists of the vertices and edges of a cycle of length n; for $k \geq 4$, the graph C_k is often called a *chordless cycle* or a *hole* and $\overline{C_k}$ is an *antihole*;
- the *complete bipartite graph* $K_{r,s}$ is the $r \times s$ bipartite graph in which each vertex is adjacent to all of the vertices in the other partite set;
- the *complete k-partite graph* $K_{r_1, r_2, \ldots, r_k}$ has its vertices in k sets with orders $r_1, r_2, \ldots, r_k$, and every vertex is adjacent to all of the vertices in the other sets; if the k sets all have order r, the graph is denoted by $K_{k(r)}$.

Examples of these graphs are given in Fig. 3.

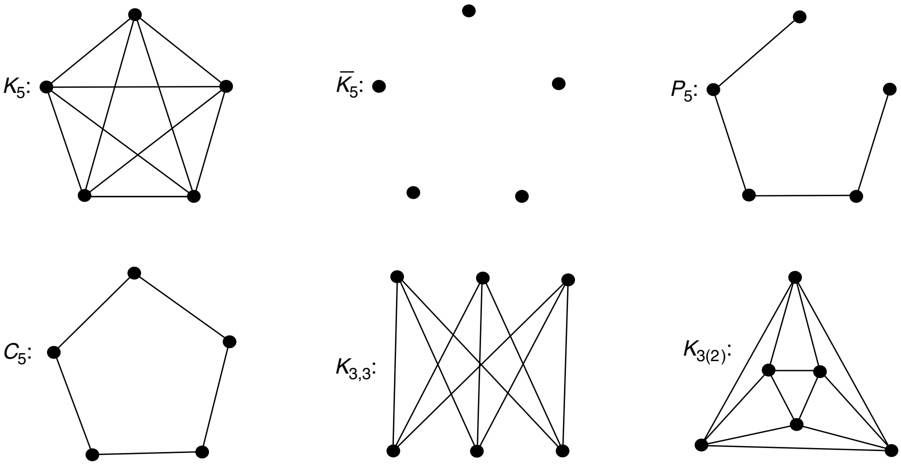

Fig. 3. Examples of special graphs

Operations on graphs

Let G and H be graphs with disjoint vertex-sets $V(G) = \{v_1, v_2, \ldots, v_r\}$ and $V(H) = \{w_1, w_2, \ldots, w_s\}$.

- The *union* $G \cup H$ has vertex-set $V(G) \cup V(H)$ and edge-set $E(G) \cup E(H)$. The union of k graphs isomorphic to G is denoted by kG.
- The *join* $G + H$ is obtained from $G \cup H$ by adding an edge from each vertex in G to each vertex in H.
- The *Cartesian product* $G \times H$ (or $G \,\Box\, H$) has vertex-set $V(G) \times V(H)$, with (v_i, w_j) adjacent to (v_h, w_k) if either v_i is adjacent to v_h in G and $w_j = w_k$, or $v_i = v_h$ and w_j is adjacent to w_k in H; in less formal terms, $G \times H$ can be obtained by taking n copies of H and joining corresponding vertices in different copies whenever there is an edge in G.

- The *lexicographic product* (or *composition*) $G[H]$ also has vertex-set $V(G) \times V(H)$, but with (v_i, w_j) adjacent to (v_h, w_k) if either v_i is adjacent to v_h in G or $v_i = v_h$ and w_j is adjacent to w_k in H.

Examples of these binary operations are given in Fig. 4.

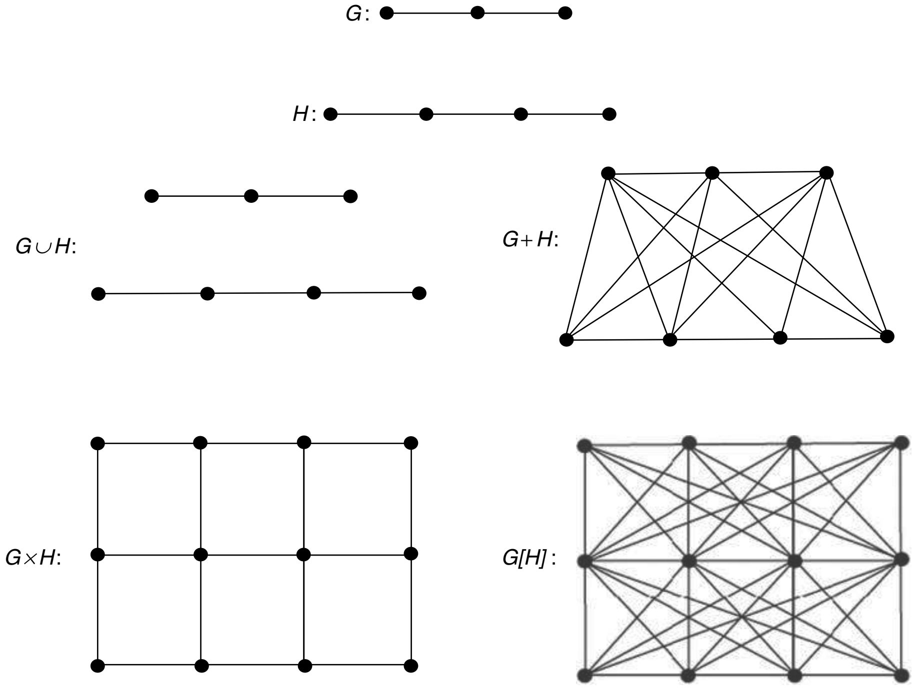

Fig. 4. Binary operations on graphs

Subgraphs and minors

If G and H are graphs with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, then H is a *subgraph* of G, and is a *spanning subgraph* if $V(H) = V(G)$. The subgraph $\langle S \rangle$ (or $G[S]$) *induced* by a non-empty set of S of vertices of G is the subgraph H whose vertex-set is S and whose edge-set consists of those edges of G that join two vertices in S. A subgraph H of G is called an *induced subgraph* if $H = \langle V(H) \rangle$. In Fig. 5, H_1 is a spanning subgraph of G, and H_2 is an induced subgraph.

A graph G is called *H-free* if it contains no induced subgraph isomorphic to the graph H. For example, a forest is $\{C_k : k \geq 3\}$-free, a *claw-free* graph has no induced $K_{1,3}$ and a *triangle-free* graph has no induced K_3. Similarly, for a set of graphs $\mathcal{H}$, we say that G is *$\mathcal{H}$-free* if it is *H-free* for each graph $H \in \mathcal{H}$. For example, the class of *threshold graphs* (introduced later) has a forbidden subgraph characterization as the $\{P_4, C_4, 2K_2\}$-free graphs.

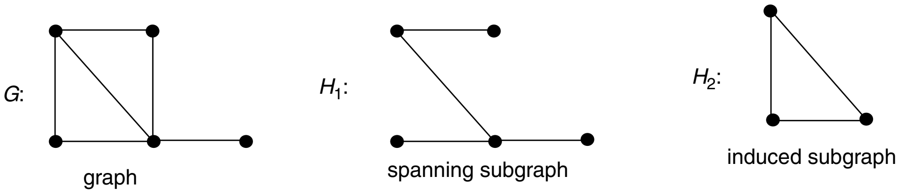

Fig. 5. Spanning and induced subgraphs of the graph G

The *deletion of a vertex v* from a graph G results in the subgraph obtained by removing v and all of its incident edges; it is denoted by $G - v$ and is the subgraph induced by $V - \{v\}$. More generally, if S is any set of vertices in G, then $G - S$ is the graph obtained from G by deleting all of the vertices in S and their incident edges – that is, $G - S = \langle V(G) - S \rangle$.

A class (or family) $\mathcal{F}$ of graphs is called *hereditary* if it is closed under vertex deletion. For example, the class of bipartite graphs is hereditary, but the class of connected graphs is not.

The *deletion of an edge e* removes it from the graph without deleting its associated vertices, resulting in the subgraph $G - e$. Similarly, for any set X of edges, $G - X$ is the graph obtained from G by deleting all the edges in X.

If the edge e joins vertices v and w, then the *subdivision* of e replaces e by a new vertex u and two new edges vu and uw. Two graphs are *homeomorphic* if there is some graph from which each can be obtained by a sequence of subdivisions. The *contraction* of e replaces its vertices v and w by a new vertex u and edges uz for every vertex z adjacent to either v or w in G. The operations of subdivision and contraction are illustrated in Fig. 6.

If H can be obtained from G by a sequence of edge-contractions and the removal of isolated vertices, then G is *contractible* to H. A *minor* of G is any graph that can be obtained from G by a sequence of edge-deletions and edge-contractions, along with deletions of isolated vertices. Note that if G has a subgraph homeomorphic to H, then H is a minor of G.

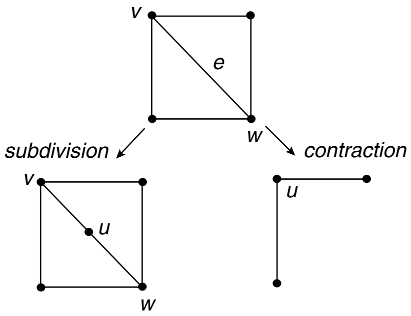

Fig. 6. The operations of subdivision and contraction

2. Connectivity

In this section, we give the primary definitions and some of the basic results on connectivity, including several versions of the most important one of all, Menger's theorem.

Vertex-connectivity

A vertex v in a graph G is a *cut-vertex* if $G - v$ has more components than G. For a connected graph, this is equivalent to saying that $G-v$ is disconnected, and that there exist vertices u and w, different from v, for which v is on every u–w path.

A non-trivial graph is *non-separable* if it is connected and has no cut-vertices. Note that under this definition the graph K_2 is non-separable. There are many characterizations of the other non-separable graphs, as the following statements are all equivalent for a connected graph G with at least three vertices:

- G is non-separable;
- every two vertices of G are on a cycle;
- every vertex and edge of G are on a cycle;
- every two edges of G are on a cycle;
- for any three vertices u, v and w in G, there is a v–w path that contains u;
- for any three vertices u, v and w in G, there is a v–w path that does not contain u;
- for any two vertices v and w and any edge e in G, there is a v–w path that contains e.

A *block* in a graph is a maximal non-separable subgraph. Each edge of a graph lies in exactly one block, and a vertex that is in more than one block is a cut-vertex. An *end-block* is a block with only one cut-vertex; every connected separable graph has at least two end-blocks. The graph in Fig. 7 illustrates these concepts.

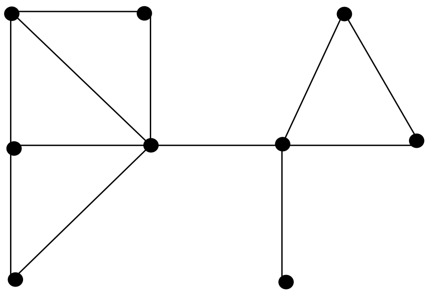

Fig. 7. A graph with 4 blocks, 3 end-blocks and 2 cut-vertices

The basic idea of non-separability has a natural generalization: a graph G is *k-connected* if the removal of fewer than k vertices always leaves a non-trivial connected graph. The main result on graph connectivity is Menger's theorem, first published in 1927. It has many equivalent forms, and the first that we give here is the global vertex version due to H. Whitney. Paths joining the same pair of vertices are called *internally disjoint* if they have no other vertices in common.

Whitney's theorem (Global vertex version) *A graph is k-connected if and only if every pair of vertices are joined by k internally disjoint paths.*

The *connectivity* $\kappa(G)$ of a graph G is the largest non-negative integer k for which G is k-connected; for example, the connectivity of the complete graph K_n is $n-1$, and a graph has connectivity 0 if and only if it is trivial or disconnected.

For non-adjacent vertices v and w in a graph G, a *v–w separating set* is a set of vertices whose removal leaves v and w in different components, and the *v–w connectivity* $\kappa(v,w)$ is the minimum order of a v–w separating set.

Menger's theorem (Local vertex version) *If v and w are non-adjacent vertices in a graph G, then the maximum number of internally disjoint v–w paths is* $\kappa(v,w)$.

Edge-connectivity

There is an analogous body of material that involves edges rather than vertices, and because of the similarities, we treat it in less detail.

An edge e is a *cut-edge* (or *bridge*) of a graph G if $G-e$ has more components than G. (In contrast to the situation with vertices, the removal of an edge cannot increase the number of components by more than 1.) An edge e is a cut-edge if and only if there exist vertices v and w for which e is on every v–w path. The cut-edges in a graph are also characterized by the property of not lying on a cycle; thus, a graph is a forest if and only if every edge is a cut-edge. Graphs having no cut-edges can be characterized in a variety of ways similar to those having no cut-vertices – that is, non-separable graphs. The concepts corresponding to cycles and paths for vertices are circuits and trails for edges.

Moving beyond cut-edges, we have the following definitions. A graph G is *l-edge-connected* if the removal of fewer than l edges always leaves a connected graph. Here is a third version of Menger's theorem.

Menger's theorem (Global edge version) *A graph is l-edge-connected if and only if each pair of its vertices are joined by l edge-disjoint paths.*

The *edge-connectivity* $\lambda(G)$ of a graph G is the largest non-negative integer l for which G is l-edge-connected. Obviously, $\lambda(G)$ cannot exceed the minimum degree of a vertex of G; furthermore, it is at least as large as the connectivity – that is,

$$\kappa(G) \le \lambda(G) \le \delta(G).$$

For non-adjacent vertices v and w in a graph G, a *v–w cutset* is a set of edges whose removal leaves v and w in different components, and the *v–w edge-connectivity* $\lambda(v,w)$ is the minimum number of edges in a v–w cutset.

Menger's theorem (Local edge version) *If v and w are vertices in a graph G, then the maximum number of edge-disjoint v–w paths is* $\lambda(v, w)$.

Along with the four undirected versions of Menger's theorem, there are corresponding directed versions (with directed paths and strong connectivity) and also weighted versions.

3. Optimization problems on graphs

In this section, we present some classical graph problems that have become fundamental in graph theory and have motivated the development of many graph algorithms.

Independent sets and cliques

A set of vertices of a graph G is an *independent set* (or *stable set*) if no two vertices are adjacent. An independent set of G is called *maximal* if it is not contained in a larger independent set, and *maximum* if its cardinality is largest possible. The *independence number* (or *stability number*) $\alpha(G)$ is the size of the largest independent set.

A set of vertices of G is *complete* if all pairs of vertices are adjacent. A complete set is a *clique* if it is a maximal complete set, and it is a *maximum clique* if its cardinality is largest possible. The *clique number* $\omega(G)$ is the size of a largest complete set.

An independent set in a graph is *strong* if it intersects every maximal clique. A *strong clique* is defined analogously. These concepts are related to others in graph theory, including perfect matchings, well-covered graphs and perfect graphs, as well as in other areas of mathematics. Chapter 10 gives an introduction to strong cliques and strong independent sets.

Colourings

A *colouring* of a graph G is an assignment of a colour to each vertex of G so that adjacent vertices always have different colours, and G is *k-colourable* if it has a colouring with k colours. The *chromatic number* $\chi(G)$ is the smallest value of k for which G has a k-colouring. It is easy to see that a graph is 2-colourable if and only if it is bipartite, but there is no 'good' way to determine which graphs are k-colourable for $k \geq 3$.

The complete graph K_n of order n has chromatic number n. Thus, $\omega(G) \leq \chi(G)$, for every graph G – that is, its clique number is a lower bound on its chromatic number. Brooks's theorem provides one of the best-known upper bounds on the chromatic number of a graph.

Brooks's theorem If G is a graph with maximum degree Δ that is neither an odd cycle nor a complete graph, then $\chi(G) \leq \Delta$.

Brooks's theorem also provides a greedy heuristic colouring algorithm. Graph algorithms form the topic of Chapter 1.