1. Basic Terminology, Notation and Results

Jørgen Bang-Jensen and Gregory Gutin

In this chapter we will provide most of the terminology and notation used in this book. Various examples, figures and results should help the reader to better understand the notions introduced in the chapter. We also prove some basic results on digraphs and provide some fundamental digraph results without proofs. Most of our terminology and notation is standard and agrees with [4]. Thus, some readers may proceed to other chapters after a quick look through this chapter (unfamiliar terminology and notation can be clarified later by consulting the indices supplied at the end of this book).

In Section 1.1 we provide basic terminology and notation on sets and matrices. Digraphs, directed pseudographs, subgraphs, weighted directed pseudographs, neighbourhoods, semi-degrees and other basic concepts of directed graph theory are introduced in Section 1.2. In Section 1.3, we introduce oriented and directed walks, trails, paths and cycles, and related subgraphs. Isomorphism and basic operations on digraphs are considered in Section 1.4. Basic notions and results on strong connectivity are considered in Section 1.5. Section 1.6 provides basic definitions on linkages in digraphs. Undirected graphs and orientations of undirected and directed graphs are considered in Section 1.7. In Section 1.8, we briefly discuss out-branchings or in-branchings. Section 1.9 is devoted to a brief discussion of some results on flows in networks. In the last three sections, we discuss algorithmic approaches and lower bounds for solving $\mathcal{NP}$-hard problems: exponential time algorithms and the Exponential Time Hypothesis, fixed-parameter tractable algorithms and W-complexity classes, and approximation algorithms.

1.1 Sets, Matrices, Vectors and Hypergraphs

For the sets of real numbers, rational numbers and integers we will use $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{Z}$, respectively. Also, let $\mathbb{Z}_+ = \{z \in \mathbb{Z} : z > 0\}$ and $\mathbb{Z}_0 = \{z \in \mathbb{Z} : z \geq 0\}$.

J. Bang-Jensen
Department of Mathematics and Computer Science,
University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark
e-mail: jbj@imada.sdu.dk

G. Gutin (✉)
Department of Computer Science,
Royal Holloway University, Egham Hill, TW20 0EX, Egham, England
e-mail: gutin@rhul.ac.uk

J. Bang-Jensen and G. Gutin (eds.), *Classes of Directed Graphs*, Springer Monographs in Mathematics, https://doi.org/10.1007/978-3-319-71840-8_1

The sets $\mathbb{R}_+$, $\mathbb{R}_0$, $\mathbb{Q}_+$ and $\mathbb{Q}_0$ are defined similarly. For a positive integer n, $[n]$ will denote the set $\{1, 2, \ldots, n\}$.

The main aim of this section is to establish some notation and terminology on finite sets used in this book. We assume that the reader is familiar with the following basic operations for a pair A, B of sets: the **intersection** $A \cap B$, the **union** $A \cup B$ (if $A \cap B = \emptyset$, then we will often write $A + B$ instead of $A \cup B$) and the **difference** $A \backslash B$ (often denoted by $A - B$). Sets A and B are **disjoint** if $A \cap B = \emptyset$.

Often we will not distinguish between a single element set (singleton) $\{x\}$ and the element x itself. For example, we may write $A \cup b$ or $A + b$ instead of $A \cup \{b\}$. The **Cartesian product** of sets $X_1, X_2, \ldots, X_p$ is defined as $X_1 \times X_2 \times \ldots \times X_p = \{(x_1, x_2, \ldots, x_p) : \ x_i \in X_i, \ 1 \leq i \leq p\}$.

For sets A, B, $A \subseteq B$ means that A is a subset of B; $A \subset B$ stands for $A \subseteq B$ and $A \neq B$. A set B is a **proper subset** of a set A if $B \subset A$ and $B \neq \emptyset$. A collection $S_1, S_2, \ldots, S_t$ of (not necessarily non-empty) subsets of a set S is a **subpartition** of S if $S_i \cap S_j = \emptyset$ for all $1 \leq i \neq j \leq t$. A subpartition $S_1, S_2, \ldots, S_t$ is a **partition** of S if $\cup_{i=1}^{t} S_i = S$. We will often use the name **family** for a collection of sets. A family $\mathcal{F} = \{X_1, X_2, \ldots, X_n\}$ of sets is **covered** by a set S if $S \cap X_i \neq \emptyset$ for every $i \in [n]$. We say that S is a **cover** of $\mathcal{F}$. For a finite set X, the number of elements in X (i.e., its **cardinality**) is denoted by $|X|$. We also say that X is an **$|X|$-element set** (or just an **$|X|$-set**). A set S satisfying a property $\mathcal{P}$ is a **maximum** (**maximal**, respectively) set with property $\mathcal{P}$ if there is no set S' satisfying $\mathcal{P}$ and $|S'| > |S|$ ($S \subset S'$, respectively). Similarly, one can define **minimum (minimal)** sets satisfying a property $\mathcal{P}$.

In this book, we will also use **multisets** which, unlike sets, are allowed to have repeated (multiple) elements. The **cardinality** $|S|$ of a multiset M is the total number of elements in S (including repetitions). Often, we will use the words 'family' and 'collection' instead of 'multiset'.

For an $m \times n$ matrix $S = [s_{ij}]$ the **transposed matrix** (of S) is the $n \times m$ matrix $S^T = [t_{kl}]$ such that $t_{ji} = s_{ij}$ for every $i \in [m]$ and $j \in [n]$. Unless otherwise specified, the vectors that we use are column-vectors. The operation of transposition is used to obtain row-vectors.

A **hypergraph** is an ordered pair $H = (V, \mathcal{E})$ such that V is a set (of **vertices** of H) and $\mathcal{E}$ is a family of subsets of V (called **edges** of H). The **rank** of H is the cardinality of the largest edge of H. For example, $H_0 = (\{1, 2, 3, 4\}, \{\{1, 2, 3\}, \{2, 3\}, \{1, 2, 4\}\})$ is a hypergraph of rank three. The number of vertices in H is its **order**. We say that H is **2-colourable** if there is a function $f : \ V \rightarrow \{0, 1\}$ such that, for every edge $E \in \mathcal{E}$, there exist a pair of vertices $x, y \in E$ such that $f(x) \neq f(y)$. The function f is called a **2-colouring** of H. It is easy to verify that H_0 is 2-colourable. In particular, $f(1) = f(2) = 0, \ f(3) = f(4) = 1$ is a 2-colouring of H_0. A hypergraph is **uniform** if all its edges have the same cardinality. Thus an undirected graph is just a 2-uniform hypergraph.

1.2 Digraphs, Subgraphs, Neighbours, Degrees

A **directed graph** (or just **digraph**[1]) D consists of a non-empty finite set $V(D)$ of elements called **vertices** and a finite set $A(D)$ of ordered pairs of distinct vertices called **arcs**. We call $V(D)$ the **vertex set** and $A(D)$ the **arc set** of D. We will often write $D = (V, A)$, which means that V and A are the vertex set and arc set of D, respectively. The **order** (**size**) of D is the number of vertices (arcs) in D; the order of D will sometimes be denoted by $|D|$. For example, the digraph D in Figure 1.1 is of order and size 6; $V(D) = \{u, v, w, x, y, z\}$, $A(D) = \{(u, v), (u, w), (w, u), (z, u), (x, z), (y, z)\}$. Often the order (size, respectively) of the digraph under consideration is denoted by **n** (**m**, respectively).

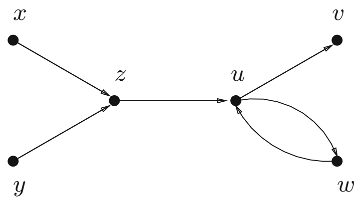


Figure 1.1 A digraph D.

For an arc (u, v) the first vertex u is its **tail** and the second vertex v is its **head**. We also say that the arc (u, v) **leaves** u and **enters** v. The head and tail of an arc are its **end-vertices**; we say that the end-vertices, are **adjacent**. If (u, v) is an arc, we also say that u **dominates** v (v **is dominated by** u) and denote it by $u \to v$. We say that a vertex u is **incident** to an arc a if u is the head or tail of a. We will often denote an arc (x, y) by xy.

For a pair X, Y of vertex sets of a digraph D, we define

$$(X, Y)_D = \{xy \in A(D) : \ x \in X, y \in Y\},$$

i.e., $(X, Y)_D$ is the set of arcs with tail in X and head in Y. For example, for the digraph H in Figure 1.2, $(\{u, v\}, \{w, z\})_H = \{uw\}$, $(\{w, z\}, \{u, v\})_H = \{wv\}$ and $(\{u, v\}, \{u, v\})_H = \{uv, vu\}$. For disjoint subsets X and Y of $V(D)$, $X \to Y$ means that every vertex of X dominates every vertex of Y. Also, $X {\mapsto} Y$ stands for $X \to Y$ and no vertex of Y dominates a vertex in X. For example, in the digraph D of Figure 1.1, $u \to \{v, w\}$ and $\{x, y\} {\mapsto} z$.

The above definition of a digraph implies that we allow a digraph to have arcs with the same end-vertices (for example, uv and vu in the digraph H in Figure 1.2), but we do not allow it to contain **parallel** (also called **multiple**) arcs, that is, pairs of arcs with the same tail and the same head, or

[1] If we know from the context that D is directed, D may be called a **graph**.

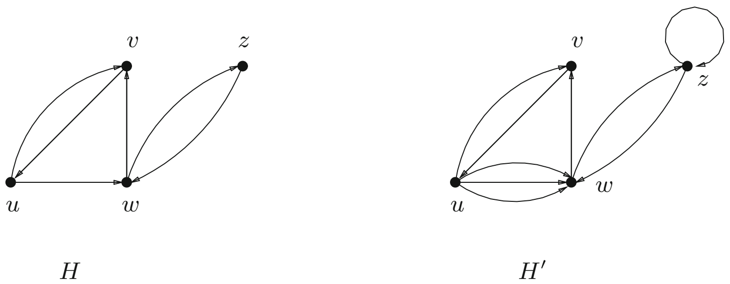


Figure 1.2 A digraph H and a directed pseudograph H'.

loops (i.e., arcs whose head and tail coincide). When parallel arcs and loops are admissible we speak of **directed pseudographs**; directed pseudographs without loops are **directed multigraphs**. In Figure 1.2 the directed pseudograph H' is obtained from H by appending a loop zz and two parallel arcs from u to w. Clearly, for a directed pseudograph D, $A(D)$ and $(X,Y)_D$ (for every pair X,Y of vertex sets of D) are multisets (parallel arcs provide repeated elements). We use the symbol $\mu_D(x,y)$ to denote the number of arcs from a vertex x to a vertex y in a directed pseudograph D. In particular, $\mu_D(x,y)=0$ means that there is no arc from x to y.

We will sometimes give terminology and notation for digraphs only, but we will provide necessary remarks on their extension to directed pseudographs, unless this is trivial.

Below, unless otherwise specified, $D=(V,A)$ is a directed pseudograph. For a vertex v in D, we use the following notation:

$$N_D^+(v) = \{u \in V - v:\ vu \in A\},\ N_D^-(v) = \{w \in V - v:\ wv \in A)\}.$$

The sets $N_D^+(v)$, $N_D^-(v)$ and $N_D(v) = N_D^+(v) \cup N_D^-(v)$ are called the **out-neighbourhood, in-neighbourhood** and **neighbourhood** of v. We call the vertices in $N_D^+(v)$, $N_D^-(v)$ and $N_D(v)$ the **out-neighbours**, **in-neighbours** and **neighbours** of v.

In Figure 1.2, $N_H^+(u)=\{v,w\}$, $N_H^-(u)=\{v\}$, $N_H(u)=\{v,w\}$, $N_{H'}^+(w)=\{v,z\}$, $N_{H'}^-(w)=\{u,z\}$, $N_{H'}^+(z)=\{w\}$. For a set $W\subseteq V$, we let

$$N_D^+(W) = \bigcup_{w\in W} N_D^+(w) - W,\ N_D^-(W) = \bigcup_{w\in W} N_D^-(w) - W.$$

That is, $N_D^+(W)$ consists of those vertices from $V-W$ which are out-neighbours of at least one vertex from W. In Figure 1.2, $N_H^+(\{w,z\})=\{v\}$ and $N_H^-(\{w,z\})=\{u\}$.

Recursively, we can define the i**th out-neighbourhood** of a set W as follows: $N^{+i}(W)=N^+(N^{+(i-1)}(W))$ for $i\geq 2$. We will denote $N^{+2}(W)$ as $N^{++}(W)$. Similarly, we can define the i**th in-neighbourhood** of a set W.

The neighbourhoods above are sometimes called **open neighbourhoods**. **Closed neighbourhoods** are defined as follows: For a set $W \subseteq V$ and positive integer p, let $N^{+p}[W] = N^{+p}(W) \cup W$ and $N^{-p}[W] = N^{-p}(W) \cup W$.

A digraph is called an **oriented graph** if it has no pair of arcs of the form xy, yx. Seymour's Second Neighbourhood Conjecture is one of the most interesting open questions in digraph theory. It has the following simple formulation.

Conjecture 1.2.1 (Seymour's Second Neighbourhood Conjecture) *In every oriented graph D, there exists a vertex x such that $|N_D^+(x)| \leq |N_D^{++}(x)|$.*

The conjecture is discussed in detail in Chapter 2 including two proofs of the conjecture for tournaments. In addition, recently Gutin and Li [23] proved the conjecture for **quasi-transitive oriented graphs**; a digraph D is called **quasi-transitive** if whenever $x \rightarrow y$ and $y \rightarrow z$ $(x \neq z)$ we have that $x \rightarrow z$ or $z \rightarrow x$ (or both). Quasi-transitive digraphs and their generalizations are considered in Chapter 8.

For a set $W \subseteq V$, the **out-degree** of W (denoted by $d_D^+(W)$) is the number of arcs in D whose tails are in W and heads are in $V - W$, i.e., $d_D^+(W) = |(W, V - W)_D|$. The **in-degree** of W, $d_D^-(W) = |(V - W, W)_D|$. In particular, for a vertex v, the out-degree is the number of arcs, except for loops, with tail v. If D is a digraph (that is, it has no loops or multiple arcs), then the out-degree of a vertex equals the number of out-neighbours of this vertex. We call the out-degree and in-degree of a set of vertices W its **semi-degrees**. The **degree** of W is the sum of its semi-degrees, i.e., the number $d_D(W) = d_D^+(W) + d_D^-(W)$. For example, in Figure 1.2, $d_H^+(u) = 2, d_H^-(u) = 1, d_H(u) = 3$, $d_{H'}^+(w) = 2, d_{H'}^-(w) = 4, d_{H'}^+(z) = d_{H'}^-(z) = 1, d_H^+(\{u, v, w\}) = d_H^-(\{u, v, w\}) = 1$. Sometimes, it is useful to count loops in the semi-degrees: the **out-pseudodegree** of a vertex v of a directed pseudograph D is the number of arcs with tail v. Similarly, one can define the **in-pseudodegree** of a vertex. In Figure 1.2, both the in-pseudodegree and out-pseudodegree of z in H' are equal to 2.

The **minimum out-degree** (**minimum in-degree**) of D is

$$\delta^+(D) = \min\{d_D^+(x) : \ x \in V(D)\} \ \ (\delta^-(D) = \min\{d_D^-(x) : \ x \in V(D)\}).$$

The **minimum semi-degree** of D is

$$\delta^0(D) = \min\{\delta^+(D), \delta^-(D)\}.$$

Finally, the **minimum degree of D is**

$$\delta(D) = \min\{d^+(v) + d^-(v) : \ v \in V(D)\}.$$

Similarly, one can define the **maximum out-degree** of D, $\Delta^+(D)$, and the **maximum in-degree**, $\Delta^-(D)$. The **maximum semi-degree** of D is

$$\Delta^0(D) = \max\{\Delta^+(D), \Delta^-(D)\}.$$

We say that D is **regular** if $\delta^0(D) = \Delta^0(D)$. In this case, D is also called **$\boldsymbol{\delta^0(D)}$-regular**.

For degrees, semi-degrees and for other parameters and sets of digraphs, we will usually omit the subscript for the digraph when it is clear which digraph is meant.

Since the number of arcs in a directed multigraph equals the number of their tails (or their heads), we obtain the following very basic result. Recall that m denotes the number of arcs in the digraph under consideration.

Proposition 1.2.2 *For every directed multigraph D we have*

$$\textstyle\sum_{x\in V(D)} d^-(x) = \sum_{x\in V(D)} d^+(x) = m.$$

$\square$

Clearly, this proposition is valid for directed pseudographs if in-degrees and out-degrees are replaced by in-pseudodegrees and out-pseudodegrees.

A digraph H is a **subdigraph** (or just **subgraph**) of a digraph D if $V(H) \subseteq V(D)$, $A(H) \subseteq A(D)$ and every arc in $A(H)$ has both end-vertices in $V(H)$. If $V(H) = V(D)$, we say that H is a **spanning subgraph** (or a **factor**) of D. The digraph L with vertex set $\{u, v, w, z\}$ and arc set $\{uv, uw, wz\}$ is a spanning subgraph of H in Figure 1.2. If every arc of $A(D)$ with both end-vertices in $V(H)$ is in $A(H)$, we say that H is **induced** by $X = V(H)$ (we write $H = D[X]$ or $H = D\langle X\rangle$) and call H an **induced** subgraph of D. The digraph G with vertex set $\{u, v, w\}$ and arc set $\{uw, wv, vu\}$ is a subgraph of the digraph H in Figure 1.2; G is neither a spanning subgraph nor an induced subgraph of H. The digraph G along with the arc uv is an induced subgraph of H. For a subset $A' \subseteq A(D)$ the subgraph **arc-induced** by A' is the digraph $D[A'] = (V', A')$, where V' is the set of vertices in V which are incident with at least one arc from A'. For example, in Figure 1.2, $H[\{zw, uw\}]$ has vertex set $\{u, w, z\}$ and arc set $\{zw, uw\}$. If H is a subgraph of D, then we say that D is a **supergraph** of H.

It is trivial to extend the above definitions of subgraphs to directed pseudographs. To avoid lengthy terminology, we call the 'parts' of directed pseudographs just **subgraphs** (instead of, say, subpseudographs).

For vertex-disjoint subgraphs H, L of a digraph D, we will often use the shorthand notation $(H, L)_D$ and $H \to L$ instead of $(V(H), V(L))_D$ and $V(H) \to V(L)$, respectively. We may also drop the index D when the digraph is clear from the context.

A **weighted directed pseudograph** is a directed pseudograph D along with a mapping $c : A(D) \to \mathbb{R}$. Thus, a weighted directed pseudograph is a triple $D = (V(D), A(D), c)$. We will also consider **vertex-weighted directed pseudographs**, i.e., directed pseudographs D along with a mapping $c :\ V(D) \to \mathbb{R}$. (See Figure 1.3.) If a is an **element** (i.e., a vertex or an arc) of a weighted directed pseudograph $D = (V(D), A(D), c)$, then $c(a)$ is called the **weight** or the **cost** of a. An (unweighted) directed pseudograph can be viewed as a (vertex-)weighted directed pseudograph whose elements are all of weight 1. For a set B of arcs of a weighted directed pseudograph $D = (V, A, c)$, we define the weight of B as follows: $c(B) = \sum_{a \in B} c(a)$. Similarly, one can define the weight of a set of vertices in a vertex-weighted directed pseudograph. The **weight of a subgraph** H of a weighted (vertex-weighted) directed pseudograph D is the sum of the weights of the arcs in H (vertices in H). For example, in the weighted directed pseudograph D in Figure 1.3 the set of arcs $\{xy, yz, zx\}$ has weight 9.5 (here we have assumed that we used the arc zx of weight 1). In the directed pseudograph H in Figure 1.3 the subgraph $U = (\{u, x, z\}, \{xz, zu\})$ has weight 5.

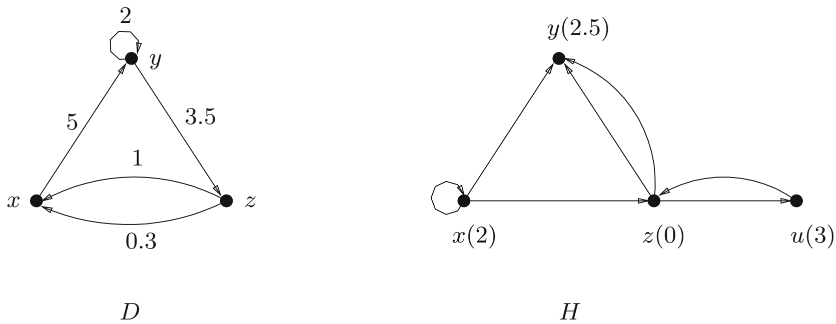


Figure 1.3 Weighted and vertex-weighted directed pseudographs (the vertex weights are given in brackets).

1.3 Walks, Trails, Paths, Cycles and Path-Cycle Subgraphs

In the following, D is always a directed pseudograph, unless otherwise specified. An **oriented walk** (or, just a **walk**) in D is an alternating sequence $W = x_1 a_1 x_2 a_2 x_3 \ldots x_{k-1} a_{k-1} x_k$ of vertices x_i and arcs a_j from D such that x_i and x_{i+1} are end-vertices of a_i for every $i \in [k-1]$. In particular, if x_i and x_{i+1} are the tail and head of a_i, respectively, for every $i \in [k-1]$, then W is a **directed walk (diwalk)**. When the fact that W is directed is known from the context, we will often say that W is a walk (this convention extends to every type of walk, i.e. trials, paths and cycles defined below). A walk W is **closed** if $x_1 = x_k$, and **open** otherwise. The set of vertices $\{x_i : i \in [k]\}$ is denoted by $V(W)$; the set of arcs $\{a_j : j \in [k-1]\}$ is denoted by $A(W)$. We say that W is a diwalk **from** x_1 **to** x_k or an $\boldsymbol{(x_1, x_k)}$**-diwalk**. If a diwalk W is open, then we say that the vertex x_1 is the **initial** vertex of W, the vertex x_k is the **terminal** vertex of W, and x_1 and x_k are **end-vertices** of W (the last term can be used for any oriented walk). The **length** of a walk is its number of arcs. Hence the walk W above has length $k-1$; we will say that W is a **$(k-1)$-walk**. A walk is **even** (**odd**) if its length is even (odd). When the arcs of W are defined from the context or simply unimportant, we will denote W by $x_1x_2\dots x_k$.

A **trail** is a walk in which all arcs are distinct. Sometimes, we identify a trail W with the *directed pseudograph* $(V(W), A(W))$, which is a *subgraph* of D. If the vertices of the diwalk W are distinct, W is a **directed path (dipath)**. If the vertices $x_1, x_2, \dots, x_{k-1}$ are distinct, $k \geq 3$ and $x_1 = x_k$, W is a **directed cycle (dicycle)**. Note that a loop is a directed cycle of length 1 and a pair of opposite arcs forms a directed cycle of length 2. A digraph is **acyclic** if it has no dicycle. An ordering $v_1, v_2, \dots, v_n$ of the vertices of a digraph D is called an **acyclic ordering** if for every arc $v_iv_j \in A(D)$, we have $i < j$. The following proposition is well-known and not hard to prove (see Chapter 3).

Proposition 1.3.1 *Every acyclic digraph has an acyclic ordering of its vertices.*

Since paths and cycles are special cases of walks, the **length** of a path and a cycle is already defined. The same remark is valid for other parameters and notions, e.g., an **(x,y)-path**. A directed path P is an **$[x,y]$-path** if P is a path between x and y, e.g., P is either an (x,y)-path or a (y,x)-path. A **longest** (**shortest**) (x,y)-dipath in a digraph D is a (x,y)-dipath of maximum (minimum) length in D. The **distance** $\mathrm{dist}(x,y)$ from a vertex x to a vertex y is the length of a shortest (x,y)-dipath. If in a digraph D there is a dipath from every vertex to every other vertex (i.e., D is strongly connected, see Section 1.5), then the **diameter** of D is the maximum of the distances $\mathrm{dist}(x,y)$ over all vertices x and y in D. If D is not strongly connected, the diameter of D is ∞. An (x,y)-dipath P is a **minimal** (x,y)**-dipath** if it is the only (x,y)-dipath in $D[V(P)]$.

When W is a cycle and x is a vertex of W, we say that W is a cycle **through** x. The **girth** $g(D)$ of D is the length of a shortest dicycle in D. If D does not have a cycle, we define $g(D) = \infty$. A digraph D is **vertex-k-cyclic** (**arc-k-cyclic**, respectively) if every vertex (arc, respectively) of D is contained in a directed k-cycle. A digraph D is **pancyclic** if it has a k-cycle for every $k \in \{3, 4, \dots, n\}$; D is **vertex-pancyclic** (**arc-pancyclic**, respectively) if D is vertex-k-cyclic (arc-k-cyclic, respectively) for every $k \in \{3, 4, \dots, n\}$.

For subsets X, Y of $V(D)$, a directed (x,y)-path P is an **(X,Y)-path** if $x \in X$, $y \in Y$ and $V(P) \cap (X \cup Y) = \{x, y\}$. Note that if $X \cap Y \neq \emptyset$, then a vertex $x \in X \cap Y$ forms an (X,Y)-path by itself. Sometimes we will talk about an (H, H')-path when H and H' are subgraphs of D. By this we mean a $(V(H), V(H'))$-path in D.

For a cycle $C = x_1x_2\dots x_px_1$, the subscripts are considered modulo p, i.e., $x_s = x_i$ for every s and i such that $i \equiv s \bmod p$. As pointed out above (for trails), we will often view paths and cycles as subgraphs. We can also consider paths and cycles as digraphs themselves. Let $\vec{\boldsymbol{P}}_{\boldsymbol{n}}$ ($\vec{\boldsymbol{C}}_{\boldsymbol{n}}$) denote a dipath (a dicycle) with n vertices, i.e., $\vec{P}_n = ([n], \{(1,2), (2,3), \dots, (n-1,n)\})$ and $\vec{C}_n = \vec{P}_n + (n,1)$.

A directed walk (path, cycle) W is a **Hamilton** (or **Hamiltonian**) walk (path, cycle) if $V(W) = V(D)$. A digraph D is **Hamiltonian** (**traceable**) if D contains a Hamilton dicycle (Hamilton dipath). A directed trail W is an **Euler** (or **Eulerian**) trail if W is closed, $V(W) = V(D)$ and $A(W) = A(D)$; a directed multigraph D is **Eulerian** if it has an Euler trail.

To illustrate these definitions, consider Figure 1.4.

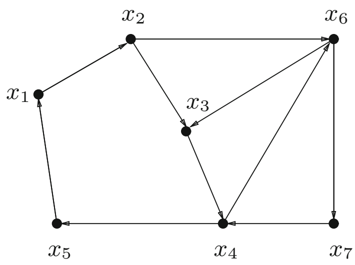


Figure 1.4 A directed graph H.

The walk $x_1x_2x_6x_3x_4x_6x_7x_4x_5x_1$ is a Hamiltonian diwalk in D. The path $x_5x_1x_2x_3x_4x_6x_7$ is a Hamiltonian dipath in D. The path $x_1x_2x_3x_4x_6$ is an (x_1, x_6)-path and $x_2x_3x_4x_6x_3$ is an (x_2, x_3)-trail. The cycle $x_1x_2x_3x_4x_5x_1$ is a 5-cycle in D. The girth of D is 3 and the longest dicycle in D has length 6.

Let $W = x_1x_2\dots x_k$, $Q = y_1y_2\dots y_t$ be a pair of walks in a digraph D. The walks W and Q are **disjoint** if $V(W) \cap V(Q) = \emptyset$ and **arc-disjoint** if $A(W) \cap A(Q) = \emptyset$. If W and Q are open walks, they are called **internally disjoint** if $\{x_2, x_3, \dots, x_{k-1}\} \cap V(Q) = \emptyset$ and $V(W) \cap \{y_2, y_3, \dots, y_{t-1}\} = \emptyset$.

We will use the following notation for a path or a cycle $W = x_1x_2\dots x_k$ (recall that $x_1 = x_k$ if W is a cycle):

$$W[x_i, x_j] = x_ix_{i+1}\dots x_j.$$

It is easy to see that $W[x_i, x_j]$ is a path for $x_i \neq x_j$; we call it the **subpath** of W from x_i to x_j. If $1 < i \leq k$, then the **predecessor** of x_i on W is the vertex x_{i-1}. If $1 \leq i < k$, then the **successor** of x_i on W is the vertex x_{i+1}.

Proposition 1.3.2 *Let D be a digraph and let x, y be a pair of distinct vertices in D. If D has an (x, y)-diwalk W, then D contains an (x, y)-dipath P*

such that $A(P) \subseteq A(W)$. If D has a closed (x,x)-diwalk W, then D contains a dicycle C through x such that $A(C) \subseteq A(W)$.

Proof: Consider a diwalk P from x to y of minimum length among all (x,y)-diwalks whose arcs belong to $A(W)$. We show that P is a path. Let $P = x_1x_2\dots x_k$, where $x = x_1$ and $y = x_k$. If $x_i = x_j$ for some $1 \leq i < j \leq k$, then the walk $P[x_1, x_i]P[x_{j+1}, x_k]$ is shorter than P; a contradiction. Thus, all vertices of P are distinct, so P is a dipath with $A(P) \subseteq A(W)$.

Let $W = z_1z_2\dots z_k$ be a diwalk from $x = z_1$ to itself ($x = z_k$). Since D has no loop, $z_{k-1} \neq z_k$. Let $y_1y_2\dots y_t$ be a shortest diwalk from $y_1 = z_1$ to $y_t = z_{k-1}$. We have proved above that $y_1y_2\dots y_t$ is a dipath. Thus, $y_1y_2\dots y_ty_1$ is a dicycle through $y_1 = x$. □

An **oriented** graph is a digraph with no cycle of length two. A **tournament** is an oriented graph where every pair of distinct vertices are adjacent. In other words, a digraph T with vertex set $\{v_i : i \in [n]\}$ is a tournament if exactly one of the arcs v_iv_j and v_jv_i is in T for every $i \neq j \in [n]$. In Figure 1.5, one can see a pair of tournaments. It is easy to see that each of them contains a Hamilton dipath. Actually, this is not a coincidence due to the following theorem of Rédei [32].

Theorem 1.3.3 (Redei's theorem) *Every tournament T is traceable.*

Proof: Let $x_1, \dots, x_n$ be an ordering of the vertices of T such that the number of **forward** arcs, i.e. arcs of the form x_ix_j $(i < j)$, is maximal. Observe that $x_i \rightarrow x_{i+1}$ for each $i \in [n-1]$. Indeed, if we had $x_{i+1} \rightarrow x_i$ for some i, we could swap vertices x_i and x_{i+1} in the ordering and obtain one more forward arc, a contradiction. Thus, $x_1 \dots x_n$ is a Hamiltonian dipath. □

In fact, Rédei proved a stronger result: every tournament contains an odd number of Hamiltonian dipaths (see Theorem 2.6.1).

Figure 1.5 Tournaments.

A **directed *q*-path-cycle subgraph** $\mathcal{F}$ of a digraph D is a collection of q dipaths $P_1, \dots, P_q$ and t dicycles $C_1, \dots, C_t$ such that all of $P_1, \dots, P_q, C_1, \dots, C_t$ are pairwise disjoint (possibly, $q = 0$ or $t = 0$). We will denote $\mathcal{F}$ by $\mathcal{F} = P_1 \cup \dots \cup P_q \cup C_1 \cup \dots \cup C_t$ (the paths always being listed first). Quite often, we will consider **directed *q*-path-cycle factors**, i.e., spanning directed q-path-cycle subgraphs. If $t = 0$, $\mathcal{F}$ is a **directed q-path subgraph** and it is a **directed q-path factor** (or just a **directed path factor**) if it is spanning. If $q = 0$, we say that $\mathcal{F}$ is a **directed t-cycle subgraph** (or just a **directed cycle subgraph**) and it is a **directed t-cycle factor** (or just a **directed cycle factor**) if it is spanning. In Figure 1.6, $abc \cup defd$ is a directed 1-path-cycle factor, and $abcea \cup dfd$ is a directed cycle factor (or, more precisely, a directed 2-cycle factor).

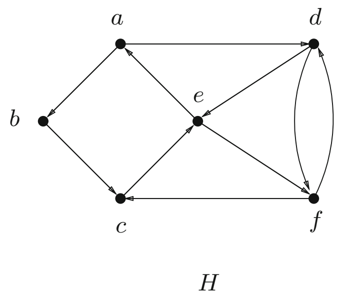


Figure 1.6 A digraph H.

A **multipartite tournament** is a digraph obtained from a complete multipartite undirected graph by replacing every edge by an arc with the same end-vertices. The following extension of Redei's theorem (Theorem 1.3.3) to multipartite tournaments was proved by Gutin [22].

Theorem 1.3.4 *A multipartite tournament has a Hamilton dipath if and only if it contains a 1-path-cycle factor.*

Chapter 7 is devoted to multipartite tournaments and their generalization, semicomplete multipartite digraphs.

The **path covering number** $\mathrm{pc}(D)$ of D is the minimum positive integer k such that D contains a k-path factor. In particular, $\mathrm{pc}(D) = 1$ if and only if D is traceable. The **path-cycle covering number** $\mathrm{pcc}(D)$ of D is the minimum positive integer k such that D contains a k-path-cycle factor. Clearly, $\mathrm{pcc}(D) \leq \mathrm{pc}(D)$. The following simple yet helpful assertion on the path covering number is not hard to show and so it is left without a proof.

Proposition 1.3.5 *Let D be a digraph, and let k be a positive integer. Then the following statements are equivalent:*

1. $\mathrm{pc}(D) = k$.
2. *There are $k - 1$ (new) arcs $e_1, \ldots, e_{k-1}$ such that $D + \{e_1, \ldots, e_{k-1}\}$ is traceable, but there is no set of $k - 2$ arcs with this property.*
3. *$k - 1$ is the minimum integer s such that addition of s new vertices to D together with all possible arcs between $V(D)$ and these new vertices results in a traceable digraph.* □

1.4 Isomorphism and Basic Operations on Digraphs

Suppose $D = (V, A)$ is a directed multigraph. A directed multigraph obtained from D by **deleting multiple arcs** is a digraph $H = (V, A')$ where $xy \in A'$ if and only if $\mu_D(x, y) \geq 1$. Let xy be an arc of D. By **reversing the arc** xy, we mean that we replace the arc xy by the arc yx. That is, in the resulting directed multigraph D' we have $\mu_{D'}(x, y) = \mu_D(x, y) - 1$ and $\mu_{D'}(y, x) = \mu_D(y, x) + 1$.

A pair of (unweighted) directed pseudographs D and H are **isomorphic** (denoted by $D \cong H$) if there exists a bijection $\phi : V(D) \to V(H)$ such that $\mu_D(x, y) = \mu_H(\phi(x), \phi(y))$ for every ordered pair x, y of vertices in D. The mapping ϕ is an **isomorphism**. Quite often, we will not distinguish between isomorphic digraphs or directed pseudographs. For example, we may say that there is only one digraph on a single vertex and there are exactly three digraphs with two vertices. Also, there is only one digraph of order 2 and size 2, but there are two directed multigraphs and six directed pseudographs of order and size 2. For a set Ψ of directed pseudographs, we say that a directed pseudograph D **belongs** to Ψ or is a **member** of Ψ (denoted $D \in \Psi$) if D is isomorphic to a directed pseudograph in Ψ. Since we usually do not distinguish between isomorphic directed pseudographs, we will often write $D = H$ instead of $D \cong H$ for isomorphic D and H.

In case we do want to distinguish between isomorphic digraphs, we speak of **labelled digraphs**. In this case, a pair of digraphs D and H is indistinguishable if and only if they completely coincide (i.e., $V(D) = V(H)$ and $A(D) = A(H)$). In particular, there are four labeled digraphs with vertex set $\{1, 2\}$. Indeed, the labeled digraphs $(\{1, 2\}, \{(1, 2)\})$ and $(\{1, 2\}, \{(2, 1)\})$ are distinct, even though they are isomorphic.

The **converse** of a directed multigraph D is the directed multigraph H which one obtains from D by reversing all arcs. It is easy to verify, using only the definitions of isomorphism and converse, that a pair of directed multigraphs are isomorphic if and only if their converses are isomorphic. To obtain subdigraphs, we use the following operations of **deletion**. For a directed multigraph D and a set $B \subseteq A(D)$, the directed multigraph $D - B$ (sometimes denoted by $D \setminus B$) is the spanning subgraph of D with arc set $A(D) \setminus B$. If $X \subseteq V(D)$, the directed multigraph $D - X$ (sometimes denoted by $D \setminus X$) is the subgraph induced by $V(D) \setminus X$, i.e., $D - X = D\langle V(D) - X\rangle$. For a subgraph H of D, we define $D - H = D - V(H)$. Since we do not distinguish between a single element set $\{x\}$ and the element x itself, we will often write $D - x$ rather than $D - \{x\}$. If H is a non-induced subgraph of a digraph D and $xy \in A(D) - A(H)$ with $x, y \in V(H)$, we can construct another subgraph H' of D by adding the arc xy of H; $H' = H + xy$.

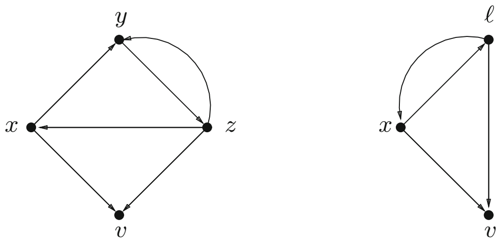
Let G be a subgraph of a directed multigraph D. The **contraction** of G in D is a directed multigraph D/G with $V(D/G) = \{g\} \cup (V(D) - V(G))$, where g is a 'new' vertex not in D, and $\mu_{D/G}(x, y) = \mu_D(x, y)$, and for all distinct vertices $x, y \in V(D) - V(G)$ we have

$$\mu_{D/G}(x,g) = \sum_{v \in V(G)} \mu_D(x,v),\ \mu_{D/G}(g,y) = \sum_{v \in V(G)} \mu_D(v,y).$$

(Note that there is no loop in D/G.) Let $G_1, G_2, \ldots, G_t$ be vertex-disjoint subgraphs of D. Then

$$D/\{G_1, G_2, \ldots, G_t\} = (\ldots((D/G_1)/G_2)\ldots)/G_t.$$

Clearly, the resulting directed multigraph $D/\{G_1, G_2, \ldots, G_t\}$ does not depend on the order of $G_1, G_2, \ldots, G_t$. Contraction can be defined for sets of vertices, rather than subgraphs. It suffices to view a set of vertices X as a subgraph with vertex set X and no arcs. Figure 1.7 depicts a digraph H and the contraction H/L, where L is the subgraph of H induced by the vertices y and z.



H $\qquad\qquad$ $T = H/L,\ L = H[\{y,z\}]$

Figure 1.7 Contraction.

We will often use the following variation of the operation of contraction. This operation is called **path-contraction** and is defined as follows. Let P be a directed (x,y)-path in a directed multigraph $D = (V,A)$. Then $D//P$ stands for the directed multigraph with vertex set $V(D//P) = V \cup \{z\} - V(P)$, where $z \notin V$, and $\mu_{D//P}(u,v) = \mu_D(u,v)$, $\mu_{D//P}(u,z) = \mu_D(u,x)$, $\mu_{D//P}(z,v) = \mu_D(y,v)$ for all distinct $u,v \in V - V(P)$. In other words, $D//P$ is obtained from D by deleting all vertices of P and adding a new vertex z such that every arc with head x (tail y) and tail (head) in $V - V(P)$ becomes an arc with head (tail) z and the same tail (head). Observe that a path-contraction in a digraph results in a digraph (no parallel arcs arise). We will often consider path-contractions of paths of length one, i.e., arcs e. Clearly, a directed multigraph D has a directed k-cycle ($k \geq 3$) through an arc e if and only if $D//e$ has a cycle through z. Observe that the obvious analogue of path-contraction for undirected multigraphs does not have this nice property, which is of use in this section. The difference between (ordinary) contraction (which is also called **set-contraction**) and path-contraction is reflected in Figure 1.8.

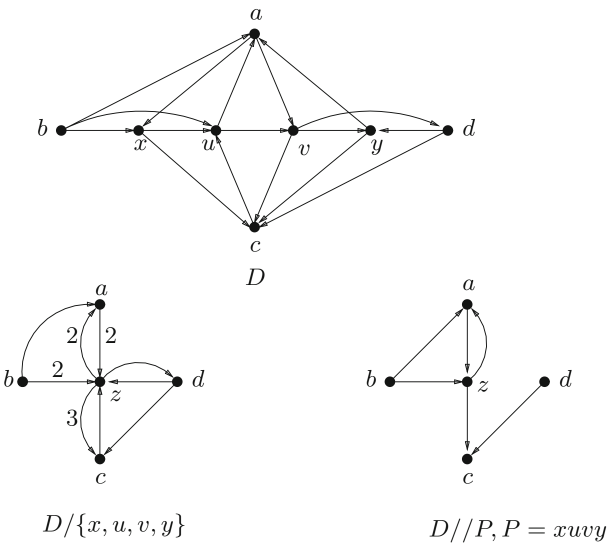


Figure 1.8 The two different kinds of contraction, set-contraction and path-contraction. The integers 2 and 3 indicate the number of corresponding parallel arcs.

As for set-contraction, for vertex-disjoint paths $P_1, P_2, \ldots, P_t$ in D, the path-contraction $D//\{P_1, \ldots, P_t\}$ is defined as the directed multigraph $(\ldots((D//P_1)//P_2)\ldots)//P_t$; clearly, the result does not depend on the order of $P_1, P_2, \ldots, P_t$.

To construct 'bigger' digraphs from 'smaller' ones, we will often use the following operation called **composition**. Let D be a digraph with vertex set $\{v_i : i \in [n]\}$, and let $G_1, G_2, \ldots, G_n$ be digraphs which are pairwise vertex-disjoint. The composition $D[G_1, G_2, \ldots, G_n]$ is the digraph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and arc set $(\cup_{i=1}^n A(G_i)) \cup \{g_i g_j : g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in A(D)\}$. Figure 1.9 shows the composition $T[G_x, G_l, G_v]$, where G_x consists of a pair of vertices and an arc between them, G_l has a single vertex, G_v consists of a pair of vertices and the pair of mutually opposite arcs between them, and the digraph T is from Figure 1.7.

If $D = H[S_1, \ldots, S_h]$ and none of the digraphs $S_1, \ldots S_h$ has an arc, then D is an **extension** of H. This notion is also used for classes of digraphs. Hence an **extended tournament** is any digraph $D = T[S_1, \ldots, S_t]$ that can be obtained from a tournament T by substituting each vertex i of T by an independent set S_i. Distinct vertices x, y are **similar** if x, y have the same in- and out-neighbours in D. For every $i \in [h]$, the vertices of S_i are similar in D.

Chapter 10 is devoted to digraph products. Here we will consider just one such product. The **Cartesian product** of a family of digraphs $D_1, D_2, \ldots, D_n$,

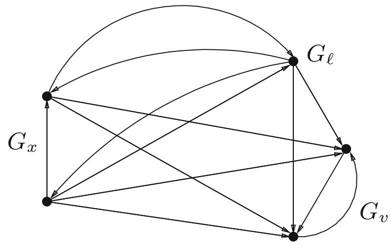


Figure 1.9 $T[G_x, G_\ell, G_v]$.

denoted by $D_1 \Box D_2 \Box \dots \Box D_n$ or $\Box_{i=1}^{n} D_i$, where $n \geq 2$, is the digraph D having

$$\begin{aligned} V(D) &= V(D_1) \times V(D_2) \times \dots \times V(D_n) \\ &= \{(w_1, w_2, \dots, w_n) : \; w_i \in V(D_i), i \in [n]\} \end{aligned}$$

and a vertex $(u_1, u_2, \dots, u_n)$ dominates a vertex $(v_1, v_2, \dots, v_n)$ of D if and only if there exists an $r \in [n]$ such that $u_r v_r \in A(D_r)$ and $u_i = v_i$ for all $i \in [n] \setminus \{r\}$. (See Figure 1.10.)

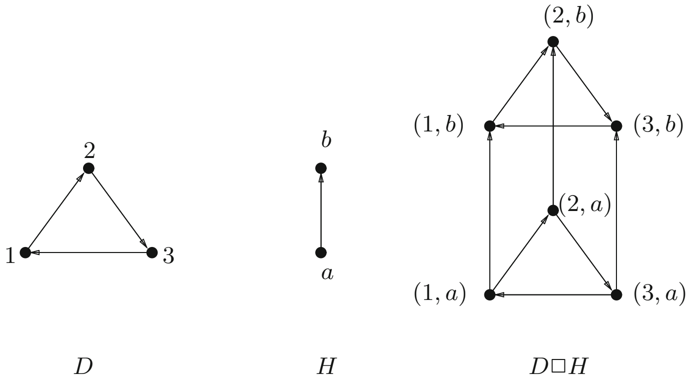


Figure 1.10 The Cartesian product of two digraphs.

The operation of **splitting** a vertex v of a directed multigraph D consists of replacing v by two new vertices v', v'', replacing all arcs of the form xv by an arc xv', replacing all arcs of the form vy by an arc $v''y$ and finally adding the arc $v'v''$. The **subdivision** of an arc uv of D consists of replacing uv by two arcs uw, wv, where w is a new vertex. If H can be obtained from D by subdividing one or more arcs (here we allow subdividing arcs that are already subdivided), then H is a **subdivision** of D. For a positive integer p and a digraph D, the ***p*th power** D^p of D is defined as follows: $V(D^p) = V(D)$,

$x \to y$ in D^p if $x \neq y$ and there are $k \leq p-1$ vertices $z_1, z_2, \ldots, z_k$ such that $x \to z_1 \to z_2 \to \ldots \to z_k \to y$ in D. According to this definition $D^1 = D$. For example, for the digraph $H_n = ([n], \{(i, i+1) :\ i \in [n-1]\})$, we have $H_n^2 = ([n], \{(i,j) :\ 1 \leq i < j \leq i+2 \leq n\} \cup \{(n-1, n)\})$. See Figure 1.11 for the second power of a digraph.

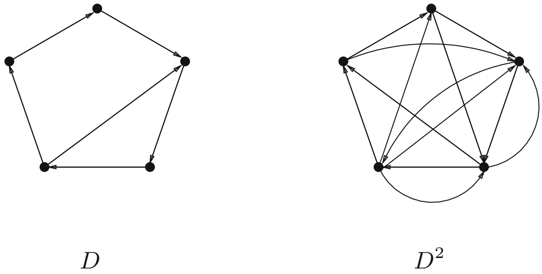


Figure 1.11 A digraph D and its second power D^2.

Let H and L be a pair of directed pseudographs. The **union** $H \cup L$ of H and L is the directed pseudograph D such that $V(D) = V(H) \cup V(L)$ and $\mu_D(x,y) = \mu_H(x,y) + \mu_L(x,y)$ for every pair x, y of vertices in $V(D)$. Here we assume that the function μ_H is naturally extended, i.e., $\mu_H(x,y) = 0$ if at least one of x, y is not in $V(H)$ (and similarly for μ_L). Figure 1.12 illustrates this definition.

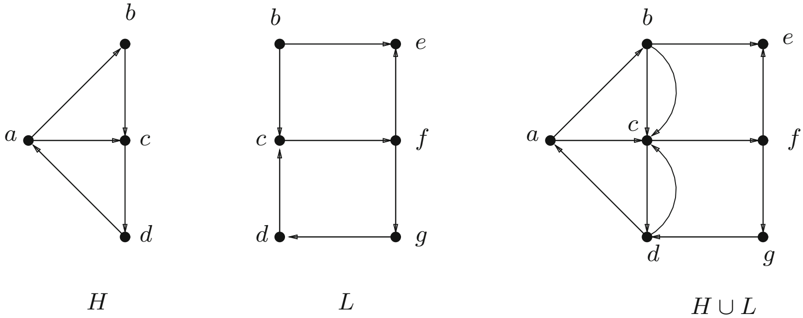


Figure 1.12 The union $D = H \cup L$ of the directed pseudographs H and L.

1.5 Strong Connectivity

In a digraph D a vertex y is **reachable** from a vertex x if D has an (x, y)-diwalk. In particular, a vertex is reachable from itself. By Proposition 1.3.2,

y is reachable from x if and only if D contains an (x, y)-dipath. A digraph D is **strongly connected** (or, just, **strong**) if, for every pair x, y of distinct vertices in D, there exists an (x, y)-diwalk and a (y, x)-diwalk. In other words, D is strong if every vertex of D is reachable from every other vertex of D. We define a digraph with one vertex to be strongly connected. It is easy to see that D is strong if and only if it has a closed Hamiltonian diwalk. As $\vec{C}_n$ is strong, every Hamiltonian digraph is strong.

Recall that a digraph D is vertex-pancyclic if for every $x \in V(D)$ and every integer $k \in \{3, 4, \ldots, n\}$, there exists a k-cycle through x in D. The following basic result on tournaments is due to Moon [30] and is proved in Chapter 2.

Theorem 1.5.1 *Every strong tournament is vertex-pancyclic.*

A digraph D is **semicomplete** if there is an arc between every pair of vertices in D. The class of semicomplete digraphs is a generalization of tournaments and many results for tournaments can be extended to semicomplete digraphs. In particular, it follows from Theorem 1.7.3 and Moon's theorem that every strong semicomplete digraph is vertex-pancyclic. A digraph D is **complete** if, for every pair x, y of distinct vertices of D, both xy and yx are in D. The complete digraph on n vertices will be denoted by $\overleftrightarrow{K}_n$.

A digraph D is **locally in-semicomplete** (**locally out-semicomplete**, respectively) if, for every vertex x of D, all in-neighbours (out-neighbours, respectively) of D induce a semicomplete digraph. It follows from Moon's theorem that every strong tournament is Hamiltonian. The following is an extension of this result by Bang-Jensen, Huang and Prisner [6].

Theorem 1.5.2 *Every strong locally in-semicomplete digraph is Hamiltonian.*

As the converse of every locally out-semicomplete digraph is locally in-semicomplete and the converse of a Hamiltonian dicycle is a Hamiltonian dicycle, Theorem 1.5.2 holds for locally out-semicomplete digraphs as well. Chapter 6 is devoted to results on locally in- and out-semicomplete digraphs.

For a strong digraph $D = (V, A)$, a set $S \subset V$ is a **separator** (or a **separating set**) if $D - S$ is not strong. A digraph D is $\boldsymbol{k}$**-strongly connected** (or $\boldsymbol{k}$**-strong**) if $|V| \geq k+1$ and D has no separator with less than k vertices. It follows from the definition of strong connectivity that a complete digraph with n vertices is $(n-1)$-strong, but is not n-strong. The largest integer k such that D is k-strongly connected is the **vertex-strong connectivity** of D (denoted by $\kappa(D)$). If a digraph D is not strong, we set $\kappa(D) = 0$. For a pair s, t of distinct vertices of a digraph D, a set $S \subseteq V(D) - \{s, t\}$ is an $\boldsymbol{(s, t)}$**-separator** if $D - S$ has no (s, t)-dipaths.

For a strong digraph $D = (V, A)$, a set of arcs $W \subseteq A$ is a **cut** (or a **cutset**) if $D - W$ is not strong. Clearly, every minimal cut is of the form $(X, \bar{X})$, where $X \subset V$ and $\bar{X} = V - X$. A cut $(X, \bar{X})$ is called a $\boldsymbol{(u, v)}$-

cut if $u \in X$ and $v \in \bar{X}$. A digraph D is ***k*-arc-strong** (or ***k*-arc-strongly connected**) if D has no cut with less than k arcs. The largest integer k such that D is k-arc-strongly connected is the **arc-strong connectivity** of D (denoted by $\lambda(D)$). If D is not strong, we set $\lambda(D) = 0$. Note that $\lambda(D) \geq k$ if and only if $d^+(X), d^-(X) \geq k$ for all proper subsets X of V. A collection $\mathcal{P}$ of paths is called **arc-disjoint** if no pair of paths in $\mathcal{P}$ has common arcs.

The following theorem is one of the most fundamental results in graph theory.

Theorem 1.5.3 (Menger's theorem)*[29] Let D be a directed multigraph and let $u, v \in V(D)$ be a pair of distinct vertices. Then the following holds:*

(a) The maximum number of arc-disjoint (u, v)-dipaths equals the minimum number of arcs covering all (u, v)-dipaths and this minimum is attained for some (u, v)-cut $(X, \bar{X})$.
(b) If the arc uv is not in $A(D)$, then the maximum number of internally disjoint (u, v)-dipaths equals the minimum number of vertices in a (u, v)-separator.

A **strong component** of a digraph D is a maximal induced subgraph of D which is strong. If $D_1,\ldots,D_t$ are the strong components of D, then clearly $V(D_1) \cup \ldots \cup V(D_t) = V(D)$ (recall that a digraph with only one vertex is strong). Moreover, we must have $V(D_i) \cap V(D_j) = \emptyset$ for every $i \neq j$ as otherwise all the vertices $V(D_i) \cup V(D_j)$ are reachable from each other, implying that the vertices of $V(D_i) \cup V(D_j)$ belong to the same strong component of D. We call $V(D_1) \cup \ldots \cup V(D_t)$ the **strong decomposition** of D. The **strong component digraph** $SC(D)$ of D is obtained by contracting the strong components of D and deleting any parallel arcs obtained in this process. In other words, if $D_1,\ldots,D_t$ are the strong components of D, then $V(SC(D)) = \{v_i : \ i \in [t]\}$ and $A(SC(D)) = \{v_i v_j : \ (V(D_i), V(D_j))_D \neq \emptyset\}$. The subgraph of D induced by the vertices of a dicycle in D is strong, and hence is contained in a strong component of D. Thus, $SC(D)$ is acyclic. By Proposition 3.1.2 in Chapter 3, the vertices of $SC(D)$ have an acyclic ordering. This implies that the strong components of D can be labelled $D_1,\ldots,D_t$ such that there is no arc from D_j to D_i unless $j < i$. We call such an ordering an **acyclic ordering** of the strong components of D. The strong components of D corresponding to the vertices of $SC(D)$ of in-degree (out-degree) zero are the **initial (terminal) strong components** of D. The remaining strong components of D are called the **intermediate strong components** of D. Figure 1.13 shows a digraph D and its strong component digraph $SC(D)$.

It is easy to see that the strong component digraph of a tournament T is an acyclic tournament. Thus, there is a unique acyclic ordering of the strong components of T, namely, $T_1,\ldots,T_t$ such that $T_i \rightarrow T_j$ for every $i < j$. Clearly, every tournament has only one initial (terminal) strong component.

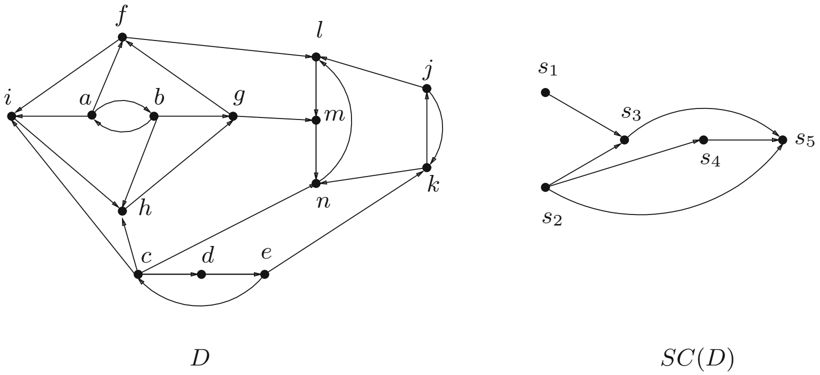


Figure 1.13 A digraph D and its strong component digraph $SC(D)$. The vertices s_1, s_2, s_3, s_4, s_5 are obtained by contracting the sets $\{a, b\}, \{c, d, e\}, \{f, g, h, i\}, \{j, k\}$ and $\{l, m, n\}$ which correspond to the strong components of D. The digraph D has two initial components, D_1, D_2 with $V(D_1) = \{a, b\}$ and $V(D_2) = \{c, d, e\}$. It has one terminal component D_5 with vertices $V(D_5) = \{l, m, n\}$ and two intermediate components D_3, D_4 with vertices $V(D_3) = \{f, g, h, i\}$ and $V(D_4) = \{j, k\}$.

1.6 Linkages

Let $D = (V, A)$ be a digraph and let $s_1, \ldots, s_k, t_1, \ldots, t_k$ be a collection of (not necessarily distinct) vertices of D. A **k-linkage** from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$ is a collection of k internally disjoint dipaths $P_1, \ldots, P_k$ such that, for each $i \in [k]$, P_i is an (s_i, t_i)-dipath if $s_i \neq t_i$ and a dicycle containing s_i if $s_i = t_i$ and s_i, t_i are not internal vertices of P_j for any $j \neq i$. In the case of a cycle C containing s_i, the term internally disjoint means the same as for paths, i.e., no other path or cycle contains vertices $V(C) - \{s_i, t_i\}$. Note that a dicycle with just one vertex must be a loop, not just a vertex itself. A **weak k-linkage** from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$ is a collection of k arc-disjoint dipaths $P_1, \ldots, P_k$ such that, for each $i \in [k]$, P_i is an (s_i, t_i)-dipath if $s_i \neq t_i$ and a dicycle containing s_i if $s_i = t_i$. The next two problems on linkages are fundamental and of central importance in digraph theory.

> k-LINKAGE
> **Input:** A digraph $D = (V, A)$ and not necessarily distinct vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.
> **Question:** Does D contain a k-linkage from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$?

WEAK k-LINKAGE
Input: A digraph $D = (V, A)$ and not necessarily distinct vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.
Question: Does D contain a weak k-linkage from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$?

A digraph is **k-linked** (**weakly k-linked**, respectively) if it has a k-linkage (a weak k-linkage, respectively) for every choice of vertices as above.

Kühn and Osthus [27] proved the following:

Theorem 1.6.1 *Let $k \geq 2$ be an integer. Every digraph D of order $n \geq 400k^3$ which satisfies $\delta^0(D) \geq n/2 + k - 1$ is k-linked.*

The k-LINKAGE and the WEAK k-LINKAGE problems are both $\mathcal{NP}$-hard even for $k = 2$ [20], Still, somewhat surprisingly, weakly k-linked digraphs are easy to classify due to the following result of Shiloach [35]. Its proof, due to Shiloach, is a beautiful application of Edmonds' branching theorem (Theorem 1.8.2), see [35].

Theorem 1.6.2 *A digraph is weakly k-linked if and only if it is k-arc-strong. Furthermore, there is a polynomial algorithm for finding a weak k-linkage from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$, for any choice of these vertices, in a k-arc-strong digraph.*

So for weak k-linkages, the interesting case is when the arc-strong connectivity is less than k.

1.7 Undirected Graphs and Orientations of Undirected and Directed Graphs

An **undirected graph** $G = (V, E)$ consists of a non-empty finite set $V = V(G)$ of elements called **vertices** and a finite set $E = E(G)$ of unordered pairs of distinct vertices called **edges**. We call $V(G)$ the **vertex set** and $E(G)$ the **edge set** of G. In other words, an edge $\{x, y\}$ is a 2-element subset of $V(G)$. We will often denote $\{x, y\}$ just by xy. If $xy \in E(G)$, we say that the vertices x and y are **adjacent**. Notice that, in the above definition of an undirected graph, we do not allow loops (i.e., pairs consisting of the same vertex) or parallel edges (i.e., multiple pairs with the same end-vertices). The **complement** $\overline{G}$ of an undirected graph G is the undirected graph with vertex set $V(G)$ in which two vertices are adjacent if and only if they are not adjacent in G.

When parallel edges and loops are admissible we speak of **undirected pseudographs**; pseudographs with no loops are **multigraphs**. For a pair u, v of vertices in a pseudograph G, $\mu_G(u, v)$ denotes the number of edges between u and v. In particular, $\mu_G(u, u)$ is the number of loops at u.

A multigraph G is **complete** if every pair of distinct vertices in G are adjacent (that is, $\mu_G(u,v) > 0$ for all $u, v \in V$, $u \neq v$). We will denote the complete undirected graph on n vertices (which is unique up to isomorphism) by K_n. Its complement $\overline{K}_n$ has no edge.

A multigraph H is ***p*-partite** if there exists a partition $V_1, V_2, \ldots, V_p$ of $V(H)$ into p **partite sets** (i.e., $V(H) = V_1 \cup \ldots \cup V_p$, $V_i \cap V_j = \emptyset$ for every $i \neq j$) such that every edge of H has its end-vertices in different partite sets. The special case of a p-partite graph when $p = 2$ is called a **bipartite graph**. We often denote a bipartite graph B by $B = (V_1, V_2; E)$. A p-partite multigraph H is **complete *p*-partite** if, for every pair $x \in V_i$, $y \in V_j$ $(i \neq j)$, an edge xy is in H. A complete graph on n vertices is clearly a complete n-partite graph for which every partite set is a singleton. We denote the complete p-partite *graph* with partite sets of cardinalities $n_1, n_2, \ldots, n_p$ by $K_{n_1,n_2,\ldots,n_p}$. Complete p-partite graphs for $p \geq 2$ are also called **complete multipartite graphs**.

To obtain short proofs of various results on subgraphs of a directed multigraph $D = (V, A)$ the following transformation to the class of bipartite (undirected) multigraphs is extremely useful. Let $BG(D) = (V', V''; E)$ denote the bipartite multigraph with partite sets $V' = \{v' : v \in V\}$, $V'' = \{v'' : v \in V\}$ such that $\mu_{BG(D)}(u', w'') = \mu_D(u, w)$ for every pair u, w of vertices in D. We call $BG(D)$ the **bipartite representation** of D; see Figure 1.14.

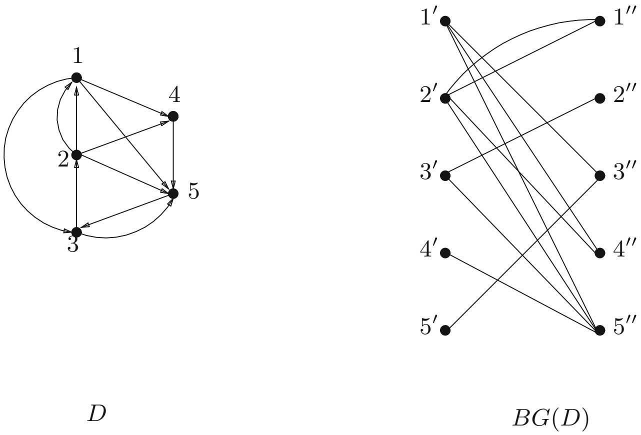


Figure 1.14 A directed multigraph and its bipartite representation.

An **orientation** of an undirected graph G is an oriented graph H obtained from G by replacing every edge xy by either arc (x, y) or arc (y, x). Let D be a directed multigraph. The **underlying multigraph** $UMG(D)$ of D is an undirected multigraph obtained from D by replacing every arc (x, y) with the edge xy. The **underlying graph** $UG(D)$ of D is obtained from $UMG(D)$ by deleting all multiple edges between every pair of vertices apart from one.

For example, for a digraph H with vertices u, v and arcs uv, vu, $UG(H)$ has one edge and $UMG(H)$ has two parallel edges. Chapter 12 is devoted to underlying graphs of digraphs.

A digraph $D = (V, A)$ is **symmetric** if $xy \in A$ implies $yx \in A$. For an undirected graph G, the **complete biorientation** of G is a symmetric digraph $\overleftrightarrow{G}$ obtained from G by replacing each edge $\{x, y\}$ with the pair xy, yx of arcs. Clearly, D is symmetric if and only if D is the complete biorientation of some graph.

An undirected pseudograph G is **connected** if its complete biorientation $\overleftrightarrow{G}$ is strongly connected. Similarly, G is ***k*-connected** if $\overleftrightarrow{G}$ is k-strong. Strong components in $\overleftrightarrow{G}$ are **connected components**, or just **components** in G. A **bridge** in an undirected pseudograph G is an edge whose deletion from G increases the number of connected components. An undirected pseudograph G is ***k*-edge-connected** if the graph obtained from G after deletion of at most $k-1$ edges is connected. Clearly, a connected undirected pseudograph is bridgeless if and only if it is 2-edge-connected. The **neighbourhood** $N_G(x)$ of a vertex x in G is the set of vertices adjacent to x. The **degree** $d(x)$ of a vertex x is the number of edges except loops having x as an end-vertex The **minimum** (**maximum**) **degree** of G is

$$\delta(G) = \min\{d(x) : \ x \in V(G)\} \ (\Delta(G) = \max\{d(x) : \ x \in V(G)\}).$$

We say that G is **regular** (or **$\delta(G)$-regular**) if $\delta(G) = \Delta(G)$. A pair of undirected graphs G and H is **isomorphic** if $\overleftrightarrow{G}$ and $\overleftrightarrow{H}$ are isomorphic.

A digraph is **connected** if its underlying graph is connected. The following well-known theorem is due to Robbins [33]. This theorem is a special case of Theorem 1.7.3.

Theorem 1.7.1 *A connected graph G has a strongly connected orientation if and only if G has no bridge.*

Here is a well-known characterization of Eulerian directed multigraphs (clearly, the deletion of loops in a directed pseudograph D does not change the property of D of being Eulerian or otherwise): A directed multigraph D is Eulerian if and only if D is connected and $d^+(x) = d^-(x)$ for every vertex x in D [4]. Eulerian directed multigraphs are considered in Chapter 4.

The notions of walks, trails, paths and cycles in undirected pseudographs are analogous to those for directed pseudographs (we merely disregard orientations). An ***xy*-path** in an undirected pseudograph is a path whose end-vertices are x and y. An undirected graph is a **forest** if it has no cycle. A connected forest is a **tree**. It is easy to see that every connected undirected graph has a **spanning tree**, i.e., a spanning subgraph, which is a tree.

A **matching** M in a directed (an undirected) pseudograph G is a set of arcs (edges) with no common end-vertices. We also require that no element of M is a loop. If M is a matching, then we say that the edges (arcs) of

M are **independent**. A matching M in G is **maximum** if M contains the maximum possible number of edges. A maximum matching is **perfect** if it has $n/2$ edges, where n is the order of G. A set Q of vertices in a directed or undirected pseudograph H is **independent** if the graph $H\langle Q\rangle$ has no edges (arcs). The **independence number**, $\alpha(H)$, of H is the maximum integer k such that H has an independent set of cardinality k. A **(proper) colouring** of a directed or undirected graph H is a partition of $V(H)$ into (disjoint) independent sets. The minimum number, $\chi(H)$, of independent sets in a proper colouring of H is the **chromatic number** of H.

In Section 1.4, the operation of composition of digraphs was introduced. Similarly, we can define the operation of **composition** of undirected graphs. Let H be a graph with vertex set $\{v_i : i \in [n]\}$, and let $G_1, G_2, \ldots, G_n$ be graphs which are pairwise vertex-disjoint. The composition $H[G_1, G_2, \ldots, G_n]$ is the graph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and edge set

$$\cup_{i=1}^{n} E(G_i) \cup \{g_i g_j : \ g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in E(H)\}.$$

If none of the graphs $G_1, \ldots, G_n$ in this definition of $H[G_1, \ldots, G_n]$ have edges, then $H[G_1, \ldots, G_n]$ is an **extension** of H.

We conclude this section with the notion of an orientation of a digraph, which extends the notion of an orientation of an undirected graph. An **orientation** of a digraph D is a subgraph of D obtained from D by deleting exactly one arc between x and y for every pair $x \neq y$ of vertices such that both xy and yx are in D. See Figure 1.15 for an illustration of this definition.

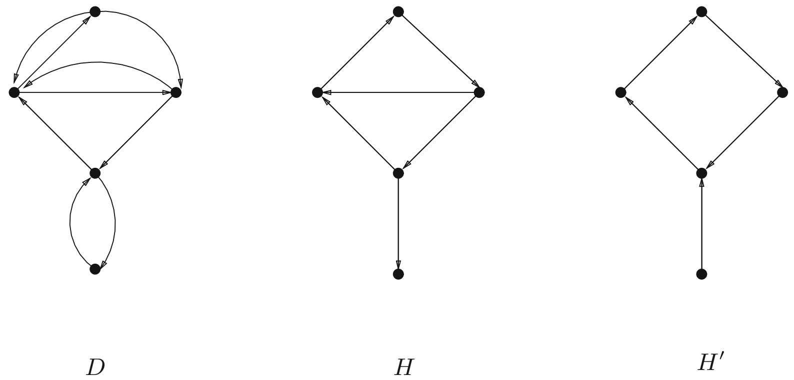


Figure 1.15 A digraph D and subgraphs H and H' of D. The digraph H is an orientation of D but H' is not.

Lemma 1.7.2 *Let D be a strong digraph and x, y vertices of D such that both xy and yx are arcs. Then either $D - xy$ or $D - yx$ is strong if and only if e is not a bridge in $UG(D)$.*

Proof: If e is a bridge in $UG(D)$, then clearly neither $D - xy$ nor $D - yx$ is strong. Assume that e is not a bridge in $UG(D)$ and consider $D' = D - \{xy, yx\}$. If D' is strong, then clearly both $D - xy$ and $D - yx$ are strong. Thus, assume that D' is not strong. Since e is not a bridge, D' is connected. Let $L_1, L_2, \ldots, L_k$ be strong components of D'. Since D is strong, there is only one initial strong component, say L_1, and only one terminal strong component, say L_k. Since D is strong, one of the vertices x and y is in L_1 and the other in L_k. Without loss of generality, x is in L_1 and y is in L_k. Then $D - xy$ is strong. $\square$

This lemma immediately implies the following theorem of Boesch and Tindell [12], which generalizes Theorem 1.7.1.

Theorem 1.7.3 *A strong digraph D has a strong orientation if and only if $UG(D)$ has no bridge.*

1.8 Trees in Digraphs

A digraph D is an **oriented forest** (**tree**) if D is an orientation of a forest (tree). A digraph T is an **out-tree** (an **in-tree**) if T is an oriented tree with just one vertex s of in-degree zero (out-degree zero). The vertex s is the **root** of T. A digraph F is an **out-forest** (an **in-forest**) if F is the vertex disjoint union of out-trees (in-trees).

If an out-tree (in-tree) T is a spanning subgraph of D, T is called an **out-branching** (an **in-branching**). (See Figure 1.16.)

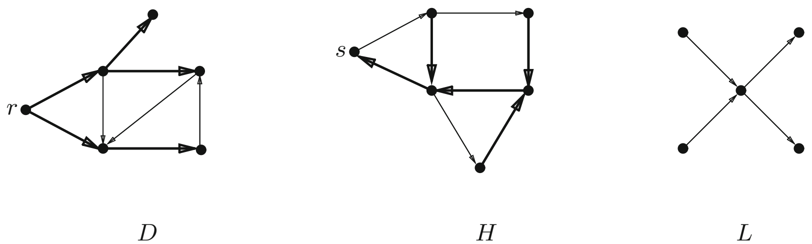


Figure 1.16 The digraph D has an out-branching with root r (shown in bold); H contains an in-branching with root s (shown in bold); L possesses neither an out-branching nor an in-branching.

Since each spanning oriented tree R of a connected digraph is acyclic, R has at least one vertex of out-degree zero and at least one vertex of in-degree zero (see Proposition 3.1.1 of Chapter 3). Hence, the out-branchings and in-branchings capture the important cases of uniqueness of the corresponding vertices. The following is a characterization of digraphs with in-branchings (out-branchings).

Proposition 1.8.1 *A connected digraph D contains an out-branching (in-branching) if and only if D has only one initial (terminal) strong component.*

Proof: We prove this characterization only for out-branchings since the second claim follows from the first one by considering the converse of D.

Assume that D contains at least two initial strong components and suppose that D has an out-branching T. Observe that the root r of T is an initial strong component of D. Let x be a vertex in another initial strong component of D. Since r is the root of T, there is a path from r to x in T and, thus, in D, which is a contradiction to the assumption that r and x are in different initial strong components of D.

Now we assume that D contains only one initial strong component D_1, and r is an arbitrary vertex of D_1. We prove that D has an out-branching rooted at r. In $SC(D)$, the vertex x corresponding to D_1 is the only vertex of in-degree zero and, hence every vertex v of $SC(D)$ is reachable from x (the longest path to v must start at x). Thus, every vertex of D is reachable from r. We construct an oriented tree T as follows. In the first step T consists of r. In Step $i \geq 2$, for every vertex y appended to T in the previous step, we add to T a vertex z, such that $y \to z$ and $z \notin V(T)$, together with the arc yz. We stop when no vertex can be included in T. Since every vertex of D is reachable from r, T is spanning. Clearly, r is the only vertex of in-degree zero in T. Hence, T is an out-branching. $\square$

The following theorem is a very important result, which can be viewed as just a fairly simple generalization of Menger's theorem. However, it has many important consequences, see the book [4] by Bang-Jensen and Gutin for many such applications of the theorem.

Theorem 1.8.2 (Edmonds' branching theorem) *[citeedmonds1973] A directed multigraph $D = (V, A)$ with a special vertex z has k arc-disjoint out-branchings rooted at z if and only*[2] *if*

$$d^-(X) \geq k \;\; \forall \;\; \emptyset \neq X \subseteq V - z. \tag{1.1}$$

There exists a polynomial algorithm for finding k arc-disjoint out-branchings from a given root s in a directed multigraph which satisfies (1.1).

A **leaf** in an out-tree (in-tree) is a vertex of out-degree (in-degree) zero. The minimum (maximum, respectively) number of leaves in an out-branching

[2] By Menger's theorem (Theorem 1.5.3), (1.1) is equivalent to the existence of k arc-disjoint dipaths from z to every other vertex of D.

of a digraph D will be denoted by $\ell_{\min}(D)$ ($\ell_{\max}(D)$, respectively). Clearly, the problem of finding $\ell_{\min}(D)$ is $\mathcal{NP}$-hard as even the problem of deciding whether $\ell_{\min}(D) = 1$ is $\mathcal{NP}$-complete as it is equivalent to the Hamilton dipath problem. The following theorem of Las Vergnas gives a bound to the minimum number of leaves in an out-branching. Recall that for a digraph D, $\alpha(D)$ denotes the maximum number of vertices without an arc between them.

Theorem 1.8.3 ([28]) *Let D be a digraph and let $\ell_{\min}(D)$ be the minimum number of leaves in an out-branching of D. Then $\ell_{\min}(D) \le \alpha(D)$.*

This theorem implies the Gallai–Milgram theorem (Theorem 1.8.4), for a proof of this fact see the paper [5] by Bang-Jensen and Gutin.

The problem of finding $\ell_{\max}(D)$ is $\mathcal{NP}$-hard; Alon, Fomin, Gutin, Krivelevich and Saurabh showed that it in fact remains $\mathcal{NP}$-hard when restricted to acyclic digraphs [1]. Daligault and Thomassé [17] designed a 92-approximation algorithm for the (general) problem and Daligault, Gutin, Kim and Yeo [16] obtained an $O^*(3.72^k)$-time algorithm for deciding whether a digraph D contains an out-branching with at least k leaves.

Rédei's theorem (Theorem 2.2.4) can be rephrased as saying that every digraph with independence number one has a Hamiltonian dipath and hence has path covering number one. Gallai and Milgram generalized this as follows.

Theorem 1.8.4 (Gallai–Milgram theorem) *[21] For every digraph D the path covering number is at most its independence number, that is $pc(D) \le \alpha(D)$.*

In fact, the following stronger result holds. It can be useful in certain applications, see, e.g., Section 3.10.3.

Theorem 1.8.5 (Gallai–Milgram theorem) *[21] Let D be a digraph, let $P = P_1 \cup \dots P_\ell$ be a dipath factor of D, and let $I(P)$ and $T(P)$ denote the sets of initial and terminal vertices, respectively, of dipaths of P. If $\ell > \alpha(D)$, then D contains a dipath factor P' with $\ell - 1$ paths and such that $I(P') \subset I(P)$ and $T(P') \subset T(P)$.*

1.9 Flows in Networks

A **network** $\mathcal{N}$ is a digraph $D = (V, A)$ in which each arc a is associated with a **capacity** $u(a)$. A **flow** in a network $\mathcal{N}$ associates each arc a of $\mathcal{N}$ with a non-negative number which must not exceed the capacity $u(a)$ of the arc. Flows in networks are widely used to model systems in which some quantity passes through channels (arcs in the network) that meet at junctions (vertices); examples include traffic in a road system, fluids in pipes, or electrical current in circuits. Here is a formal definition of networks and flows in these.

A **network** is a tuple $\mathcal{N} = (V, A, l, u, c)$, where $D = (V, A)$ is a digraph with vertex set V and arc set A, and $l : A \to \mathbb{Z}_0$, $u : A \to \mathbb{Z}_0$ and $c : A \to \mathbb{R}$ are functions. Intuitively, l and u represent **lower bounds** and **capacities** (also called **upper bounds**), respectively, on how much flow can pass through each arc, and c represents the **cost** associated with each unit of flow in each arc. If there are no costs specified and $l(a) = 0$ for each $a \in A$, then we omit the relevant letters from the notation. For example, if $\mathcal{N} = (V, A, u, c)$, then $l(a) = 0$ for each $a \in A$. Sometimes we also specify a function $b : V \to \mathbb{Z}$ such that $\sum_{v \in V} b(v) = 0$. This is called a **balance vector** and if this is also specified, we denote the network by $\mathcal{N} = (V, A, l, u, c, b)$.

Given a network $\mathcal{N} = (V, A, l, u, c)$ (or $\mathcal{N} = (V, A, l, u, c, b)$), a function $x : A \to \mathbb{R}_0$ is called a **flow** in $\mathcal{N}$; it is an **integer flow** if $x(a) \in \mathbb{Z}_0$ for each $a \in A$. For a flow x, define the **balance vector** b_x as follows:
$b_x(v) = \sum_{v' \in N^+(v)} x(vv') - \sum_{v' \in N^-(v)} x(v'v)$ for every $v \in V$. For two distinct vertices $s, t \in V$, a flow x is an (s,t)-**flow** if $b_x(s) = -b_x(t) \geq 0$ and $b_x(v) = 0$ for each $v \in V \setminus \{s, t\}$. The **value** of an (s,t)-flow x is $|x| = b_x(s)$. A flow x is a **circulation** if $b_x(v) = 0$ for every $v \in V$. The **cost** of a flow x is given by $c(x) = \sum_{vv' \in A} c(vv')x(vv')$. A flow x is **feasible** in $\mathcal{N} = (V, A, l, u, c, b)$ if the following conditions are satisfied:

(a) $l(a) \leq x(a) \leq u(a)$ for every $vv' \in A$;
(b) $b_x(v) = b(v)$ for every $v \in V$.

If no balance constraint is specified, that is, $\mathcal{N} = (V, A, l, u, c)$, then a feasible flow in $\mathcal{N}$ just has to satisfy (a) above.

See Figure 1.17 for an example of a feasible flow.

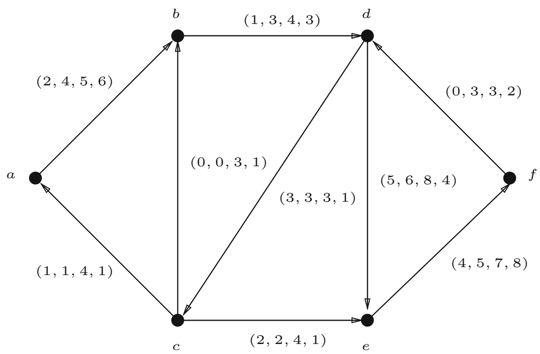


Figure 1.17 A network $\mathcal{N} = (V, A, l, u, c)$ with a feasible flow x specified. The specification on each arc uv is $(l(vw), x(vw), u(vw), c(vw))$. The cost of the flow is 109.

The following two simple propositions allow us to reduce problems about general feasible flows to problems about feasible (s,t)-flows. See [4, Section 4.2].

Proposition 1.9.1 *Let $\mathcal{N} = (V, A, l, u, b, c)$ be a network.*

(a) Suppose that the arc $ij \in A$ has $l(ij) > 0$. Let $\mathcal{N}'$ be obtained from $\mathcal{N}$ by making the following changes: $b(j) := b(j) + l(ij)$, $b(i) := b(i) - l(ij)$, $u(ij) := u(ij) - l(ij)$, $l(ij) := 0$. Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}'$ and vice versa. Furthermore, the costs of these two flows are related by $c(x) = c(x') + l(ij)c(ij)$.

(b) There exists a network $\mathcal{N}_{l\equiv 0}$ in which all lower bounds are zero such that every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{l\equiv 0}$ and vice versa. Furthermore, the costs of these two flows are related by $c(x) = c(x') + \sum_{ij \in A} l(ij)c(ij)$.

Proposition 1.9.2 *Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network. Let $M = \sum_{\{v:b(v)>0\}} b(v)$ and let $\mathcal{N}_{st}$ be the network defined as follows: $\mathcal{N}_{st} = (V \cup \{s,t\}, A', l' \equiv 0, u', b', c')$, where*

(a) $A' = A \cup \{sr : b(r) > 0\} \cup \{rt : b(r) < 0\}$,
(b) $u'(ij) = u(ij)$ for all $ij \in A$, $u_{sr} = b(r)$ for all r such that $b(r) > 0$ and $u(qt) = -b(q)$ for all q such that $b(q) < 0$,
(c) $c'(ij) = c(ij)$ for all $ij \in A$ and $c' = 0$ for all arcs leaving s or entering t,
(d) $b'(v) = 0$ for all $v \in V$, $b'(s) = M$, $b'(t) = -M$.

Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{st}$ and vice versa. Furthermore, the costs of x and x' are the same.

For a function $f:\ A \to \mathbb{Z}$ and a proper subset X of V, let $\overline{X} = V \setminus X$ and $f(X, \overline{X}) = \sum_{yz \in (X,\overline{X})} f(yz)$. It is not hard to see that given a network $\mathcal{N} = (V, A, l, u)$ if $l(\overline{S}, S) > u(S, \overline{S})$ then $\mathcal{N}$ has no feasible circulation. Hoffman [26] proved that the converse holds as well.

Theorem 1.9.3 (Hoffman's circulation theorem) *Let $\mathcal{N} = (V, A, l, u)$ be a network with lower bounds on the arcs, then $\mathcal{N}$ has a feasible circulation if and only if the following holds for every proper subset S of V:*

$$l(\overline{S}, S) \leq u(S, \overline{S}). \tag{1.2}$$

1.10 Polynomial and Exponential Time Algorithms, SAT and ETH

Unless explicitly stated otherwise, when we say that an algorithm is polynomial, respectively that a problem is polynomial, we mean that the running time of the algorithm is polynomial in the size of the input, respectively that there exists a polynomial algorithm for solving the problem.

Recall that a **CNF formula** is a conjunction of clauses. Each **clause** is a disjunction of literals, each of which is either a variable or its negation. A CNF formula F is **satisfiable** if there is a truth assignment to the variables of F such that every clause contains at least one literal equal true. In **k-CNF** formula every clause has exactly k literals. For $k \geq 2$, the problem **k-SAT** is stated as follows: Given a k-CNF formula F, decide whether F is satisfiable. It is well-known that while 2-SAT is polynomial-time solvable (see e.g. Section 17.5 in [4]), k-SAT is $\mathcal{NP}$-complete for every $k \geq 3$. The following variations of 3-SAT are also $\mathcal{NP}$-hard. In **NAE-3-SAT**, we are to decide whether there is a truth assignment for which each clause of a 3-CNF formula F has a literal equal true and a literal equal false. The problem **monotone-NAE-3-SAT** is a special case of NAE-3-SAT in which a 3-CNF formula contains no negations of variables. Finally, in **1-in-3-SAT**, given a 3-CNF formula F, decide whether there is a truth assignment making exactly one literal true in each clause of F.

It is widely believed that $\mathcal{P} \neq \mathcal{NP}$ and thus there are no polynomial time algorithms for $\mathcal{NP}$-complete problems. Unfortunately, many problems in graph theory are $\mathcal{NP}$-complete and just declaring them intractable seems too simplistic. In this and the next two sections we will briefly consider modern approaches for dealing with $\mathcal{NP}$-hard problems. We will consider only theory-based methods largely ignoring many heuristic approaches, which are of great interest in graph theory applications, but unfortunately are outside the scope of this book.

It seems that the oldest practical way to deal with $\mathcal{NP}$-hard problems is to use exponential time algorithms such as branch-and-bound. The theoretical foundations of such algorithms have been largely ignored for a while, but in the last two decades the situation has changed and many approaches and results on exponential-time algorithms have been obtained, see, e.g., [19] which is the only monograph on the topic. One such example is Schöning's randomized k-SAT algorithm [34] and its derandomization by Moser and Scheder [31]. The runtimes of Schöning's algorithm and of its derandomization are $O^*((\frac{2(k-1)}{k})^n)$ and $O^*((\frac{2(k-1)}{k}+\varepsilon)^n)$, where n is the number of variables and ε is an arbitrary positive number. As customary in the area of exponential algorithms, we used above O^* which hides not only constant factors, but also polynomial ones. Note that the obvious brute-force algorithm for k-SAT is of runtime $O^*(2^n)$.

Recently many lower bound results for the complexity of exponential time algorithms have been proved under the assumption that the Exponential Time Hypothesis (ETH) (see [15]) holds. ETH claims that there exists a real number $\delta > 0$ such that 3-SAT cannot be solved in time $O(2^{\delta n})$, where n is the number of variables in the CNF formula of 3-SAT. For example, Cygan, Fomin, Golovnev, Kulikov, Mihajlin, Pachocki and Socala [14] proved that, subject to ETH, there is no $2^{o(n \log n)}$-time algorithm deciding whether an n-vertex graph H is a subgraph of another n-vertex graph G (the obvious brute-force algorithm solves this problem in time $2^{O(n \log n)}$).

1.11 Parameterized Algorithms and Complexity

Parameterized algorithms and complexity is one of the approaches for dealing with $\mathcal{NP}$-hard problems. The main idea of this approach is that using only the size of the problem in the complexity bound for the problem is often too simplistic as the instances of the problem under consideration which are of our interest, often have some small parameter k (such as the maximum semi-degree of a digraph or the treewidth of an undirected graph). Problems with parameters are called **parameterized** problems; an instance of a parameterized problem is a pair (I, k), where I is an instance of the problem (no parameter) and k is the value of the parameter.For a parameterized problem with parameter k, an algorithm of runtime $O^*(f(k)) := O(f(k)n^c)$, where $f(k)$ is an arbitrary computable function, n is the size of the problem and c is a constant (independent of k and n), can be viewed as a generalization of a polynomial algorithm and, thus, an efficient algorithm (especially when $f(k)$ grows relatively slowly and c is of moderate value). Such algorithms are called **fixed-parameter tractable (FPT)** and parameterized problems admitting such algorithms are also called FPT. The class of FPT problems is denoted by FPT.

From the practical point of view, the chosen parameters should be relatively small on practically-interesting instances of the problem under consideration. The Directed rural postman problem (DRPP) is formulated as follows: Given a strongly connected directed multigraph $D = (V, A)$ with nonnegative integral weights on the arcs, a subset R of required arcs and a nonnegative integer ℓ, we are to decide whether D has a closed directed walk of weight at most ℓ containing every arc of R. DRPP is $\mathcal{NP}$-hard. Let k be the number of connected components in the subgraph of $UG(D)$ induced by R. In [37] Sorge, van Bevern, Niedermeier and Weller commented that "k is presumably small in a number of applications" and Sorge [36] noted that in planning for snow plowing routes for Berliner Stadtreinigung, k is only between 3 and 5. Gutin, Wahlström and Yeo [25] developed an $O^*(2^k)$-time randomized algorithm for DRPP. Unfortunately, the existence of a deterministic FPT algorithm for DRPP parameterized by k still remains "a more than thirty years open ... question with significant practical relevance" (see [37]).

When the runtime $O(f(k)n^c)$ is replaced by the much more powerful $n^{O(f(k))}$, we obtain the class **XP** where each problem is polynomial-time solvable for any fixed value of k. There are a number of parameterized complexity classes between FPT and XP (for each integer $t \geq 1$, there is a class W[t]) and they form the following tower:

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[P] \subseteq XP.$$

Here W[P] is the class of all parameterized problems (with parameter k) that can be solved in $f(k)n^{O(1)}$ time by a non-deterministic Turing machine that makes at most $f(k)\log n$ non-deterministic steps for some function f. For the definition of classes W[t], see, e.g., the monographs [15] by Cygan, Fomin, Kowalik, Lokshtanov, Marx, Pilipczuk, Pilipczuk and Saurabh, and [18] by Downey and Fellows. It is widely believed that FPT $\neq$ W[1]. One reason for this is that if FPT $=$ W[1], then ETH fails, see, e.g., [18]. The problem of deciding whether a graph has a clique with k vertices is W[1]-complete [15, 18], so it is highly unlikely that the problem is FPT.

For parameterized problems Π and Π', a **bikernelization** is a polynomial algorithm that maps an instance (I,k) of Π to an instance (I',k') of Π' (the **bikernel**) such that (i) $(I,k) \in \Pi$ if and only if $(I',k') \in \Pi'$, (ii) $k' \leq g(k)$, and (iii) $|I'| \leq g(k)$ for some function g. The function $g(k)$ is called the **size** of the bikernel. When $\Pi' = \Pi$, a bikernel is called a **problem kernel** or just a **kernel**. It is well-known that a parameterized problem Π is fixed-parameter tractable if and only if it is decidable and admits a kernelization [15, 18]. The same holds if "kernel" is replaced by a "bikernel" (see [2] by Alon, Gutin, Kim, Szeider and Yeo).

Due to applications, low degree polynomial size kernels are of main interest. Unfortunately, many FPT problems do not have kernels of polynomial size unless $\mathcal{NP} \subseteq \text{co}\mathcal{NP}/\text{poly}$, which is highly unlikely as $\mathcal{NP} = \text{co}\mathcal{NP}/\text{poly}$ would imply that the polynomial hierarchy collapses to its third level; for definitions and more information, see, e.g., [15, 18]. In particular, the problem of whether a digraph contains a k-dipath is FPT but has no polynomial kernel unless $\text{co}\mathcal{NP} \subseteq \mathcal{NP}/\text{poly}$ [11]. Binkele-Raible, Fernau, Fomin, Lokshtanov, Saurabh and Villanger [10] proved that the problem of deciding whether a digraph D and a vertex $v \in V(D)$ has an out-tree rooted at v with least k leaves admits a problem kernel with at most $O(k^3)$ vertices (and, hence, at most $O(k^6)$ arcs). Interestingly, Binkele-Raible *et al.* [10] also proved that if we allow the out-tree to be rooted at any vertex of D, then the "unrooted" problem does not admit a polynomial kernel unless $\text{co}\mathcal{NP} \subseteq \mathcal{NP}/\text{poly}$. For further background and terminology on parameterized complexity we refer the reader to the monographs [15, 18].

Let us consider a couple of recent results on parameterized complexity of problems on digraphs.

Bang-Jensen and Yeo [7] asked whether the following problem is FPT.

CONNECTIVITY PRESERVING PATH CONTRACTIONS	**Parameter:** k
Input: A strongly connected digraph D.	
Question: Can we path-contract k arcs from D such that D remains strongly connected?	

Gutin, Ramanujan, Reidl and Wahlström [24] proved that the problem is, in fact, W[1]-hard. However, the problem is FPT if the operation of path-contraction is replaced by deletion, which was proved by Basavaraju, Misra, Ramanujan and Saurabh [8].

We complete this section with an open questions on the parameterized complexity of the following digraph problem introduced by Bezáková, Curticapean, Dell and Fomin [9].

Problem 1.11.1 *For given vertices s and t of a digraph D, and an integer (parameter) k, decide whether D has an (s,t)-path in D that is at least k longer than a shortest (s,t)-path.*

If "at least" is replaced by "exactly", then the problem is FPT [9]. However, it is unknown whether the original problem is even in XP.

1.12 Approximation Algorithms

There are several situations when the use of exact optimization algorithms does not seem to be a good idea. One is when the time is greatly limited or the problem should be solved online. Another is when the data is not exact or the objective function is not well-defined and, thus, we cannot get an optimal solution even by exhaustive search. In such situations, we can use approximation algorithms for finding a solution that is often not optimal, but we have some performance guarantee in each case.

Let P be a combinatorial optimization problem, and let $\mathcal{A}$ be an approximation algorithm for P. Let $X(I)$ denote the set of all feasible solutions for some instance $I \in P$ and let $|I|$ be the size of I. We denote the solution obtained by $\mathcal{A}$ for an instance I of P by $x(I)$. Furthermore let $opt(I)$ denote the optimal solution of I. The weight of a solution y of P will be denoted by $w(y)$.

The theoretical performance of an approximation algorithm is normally measured by the (worst case) **performance ratio**. Usually, upper or lower bounds for the worst case performance ratio are obtained, where the performance ratio is defined as

$$\max_{I\in P:|I|=n} \left\{ \frac{w(x(I))}{w(opt(I))}, \frac{w(opt(I))}{w(x(I))} \right\}$$

The performance ratio defined in this way has its advantage in the fact that it is always at least 1 (for both minimization and maximization problems).

We normally require that an approximation algorithm has a polynomial running time. Some approximation algorithms provide a good performance guarantee. For example, the well-known Christofides algorithm [13] for the symmetric TSP[3] with triangle inequality (i.e., $w_{ij} + w_{jk} \geq w_{ik}$ for every

[3] The symmetric TSP is the problem of finding a minimum weight Hamilton cycle in a weighted complete undirected graph.

triple i, j, k of vertices, where w_{ij} is the weight of an edge ij) has performance ratio 1.5. Unless $\mathcal{P}$=$\mathcal{NP}$, there are no approximation algorithms of constant performance ratio for the (general) symmetric TSP [3].

A **polynomial-time approximation scheme (PTAS)** is an algorithm which takes an instance of a minimization problem $\mathcal{Q}$ and a parameter $\varepsilon > 0$ and, in polynomial time, returns a solution that is within a factor $1 + \varepsilon$ of being optimal. The definition remains the same for maximization problems, but the solution must be within a factor $1 - \varepsilon$ of being optimal. It is well-known that MaxSNP-hard problems do not admit PTAS unless $\mathcal{P} = \mathcal{NP}$.

For many results on approximation algorithms and in-approximability, see, e.g., the monograph [38] by Williamson and Shmoys.

References

1. N. Alon, F.V. Fomin, G. Gutin, M. Krivelevich, and S. Saurabh. Spanning directed trees with many leaves. *SIAM J. Discrete Math.*, 23(1):466–476, 2009.
2. N. Alon, G. Gutin, E.J. Kim, S. Szeider, and A. Yeo. Solving MAX-r-SAT above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.
3. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer-Verlag, Berlin, 1999.
4. J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer-Verlag, London, 2nd edition, 2009.
5. J. Bang-Jensen and G. Gutin. Out-branchings with extremal number of leaves. *Ramanujan Math. Soc. Lect. Notes*, 13:91–99, 2010.
6. J. Bang-Jensen, J. Huang, and E. Prisner. In-tournament digraphs. *J. Combin. Theory Ser. B*, 59(2):267–287, 1993.
7. J. Bang-Jensen and A. Yeo. The minimum spanning strong subdigraph problem is fixed parameter tractable. *Discrete Appl. Math.*, 156:2924–2929, 2008.
8. M. Basavaraju, P. Misra, M.S. Ramanujan, and S. Saurabh. On finding highly connected spanning subgraphs. *CoRR*, arXiv:1701.02853, 2017.
9. I. Bezáková, R. Curticapean, H. Dell, and F.V. Fomin. Finding detours is fixed-parameter tractable. In *ICALP 2017*, volume 80 of *LIPIcs*, pages 54:1–54:14, 2017.
10. D. Binkele-Raible, H. Fernau, F.V. Fomin, D. Lokshtanov, S. Saurabh, and Y. Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Trans. Algorithms*, 8(4):38:1–38:19, 2012.
11. H.L. Bodlaender, R.G. Downey, M.R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
12. F. Boesch and R. Tindell. Robbins's theorem for mixed multigraphs. *Amer. Math. Mon.*, 87(9):716–719, 1980.
13. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report CS-93-13, Carnegie Mellon University, 1976.
14. M. Cygan, F.V. Fomin, A. Golovnev, A.S. Kulikov, I. Mihajlin, J. Pachocki, and A. Socala. Tight bounds for graph homomorphism and subgraph isomorphism. In *SODA 2016: 27th ACM-SIAM Symposium on Discrete Algorithms*, pages 1643–1649, 2016.
15. M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

16. J. Daligault, G. Gutin, E.J. Kim, and A. Yeo. FPT algorithms and kernels for the directed k-leaf problem. *J. Comput. Syst. Sci.*, 76:144–152, 2010.
17. J. Daligault and S. Thomassé. On finding directed trees with many leaves. In *IWPEC 2009*, volume 5917 of *Lect. Notes Comput. Sci.*, pages 86–97. Springer, 2009.
18. R.G. Downey and M.R. Fellows. *Fundamentals of Parameterized Complexity.* Springer, 2013.
19. F.V. Fomin and D. Kratsch. *Exact Exponential Algorithms.* Springer, 2010.
20. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
21. T. Gallai and A.N. Milgram. Verallgemeinerung eines graphentheoretischen Satzes von Rédei. *Acta Sci. Math. Szeged*, 21:181–186, 1960.
22. G. Gutin. Characterization of complete n-partite digraphs that have a Hamiltonian path. *Kibernetika (Kiev)*, no. 1:107–108, 1988.
23. G. Gutin and R. Li. Seymour's second neighbourhood conjecture for quasi-transitive oriented graphs. *CoRR*, abs/1704.01389, 2017.
24. G. Gutin, M.S. Ramanujan, F. Reidl, and M. Wahlström. Path-contractions, edge deletions and connectivity preservation. In *ESA 2017: 25th Annual European Symposium on Algorithms*, volume 87 of *LIPIcs*, pages 47:1–47:13, 2017.
25. G. Gutin, M. Wahlström, and A. Yeo. Rural postman parameterized by the number of components of required edges. *J. Comput. Syst. Sci.*, 83(1):121–131, 2017.
26. A.J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. In R. Bellman and M. Hall, editors, *Combinatorial Analysis*, pages 113–128. American Mathematical Society, Providence, RI, 1960.
27. D. Kühn and D. Osthus. Linkedness and ordered cycles in digraphs. *Combin. Prob. Comput.*, 17:411–422, 2008.
28. M. Las Vergnas. Sur les arborescences dans un graphe orienté. *Discrete Math.*, 15(1):27–39, 1976.
29. K. Menger. Zur allgemeinen Kurventheorie. *Fund. Math.*, 10:96–115, 1927.
30. J.W. Moon. On subtournaments of a tournament. *Can. Math. Bull.*, 9:297–301, 1966.
31. R.A. Moser and D. Scheder. A full derandomization of Schöning's k-SAT algorithm. In *STOC 2011: 43rd ACM Symposium on Theory of Computing*, pages 245–252, 2011.
32. L. Rédei. Ein kombinatorischer Satz. *Acta Litt. Szeged*, 7:39–43, 1934.
33. H.E. Robbins. A theorem on graphs with an application to a problem on traffic control. *Amer. Math. Mon.*, 46:281–283, 1939.
34. U. Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *FOCS 1999: 40th IEEE Symposium on Foundations of Computer Science*, pages 410–414, 1999.
35. Y. Shiloach. Edge-disjoint branching in directed multigraphs. *Inform. Process. Lett.*, 8(1):24–27, 1979.
36. M. Sorge. Some algorithmic challenges in arc routing, May 2013. Talk at NII Shonan Seminar no. 18.
37. M. Sorge, R. van Bevern, R. Niedermeier, and M. Weller. A new view on rural postman based on eulerian extension and matching. *J. Discrete Algor.*, 16:12–33, 2012.
38. D.P. Williamson and D.B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.