– Graph Theory and Its Applications –

Trees and Graph Spaces

Xinmin Hou

Department of Computational and Applied Mathematics
xmhou@ustc.edu.cn

Contents

1 Trees and Spanning Trees **4**

Definitions 5

Spanning Forests and Spanning Trees 16

Cotrees and Edge-Cut Sets 19

Exercises 24

2 Vector Spaces of a Graph **25**

Vertex Space and Edge Space 25

Cycle Space and Cut Space 27

Bondy's conjecture 46

Exercises 48

3 The Number of Spanning Trees **50**

Exercises 64

4 The Minimum Connection Problem **66**

Spanning Tree Algorithms 67

Prim's Algorithm 68

Complexity Analysis 73

Exercises 75

5 The Shortest Path Problem **77**

Exercises 86

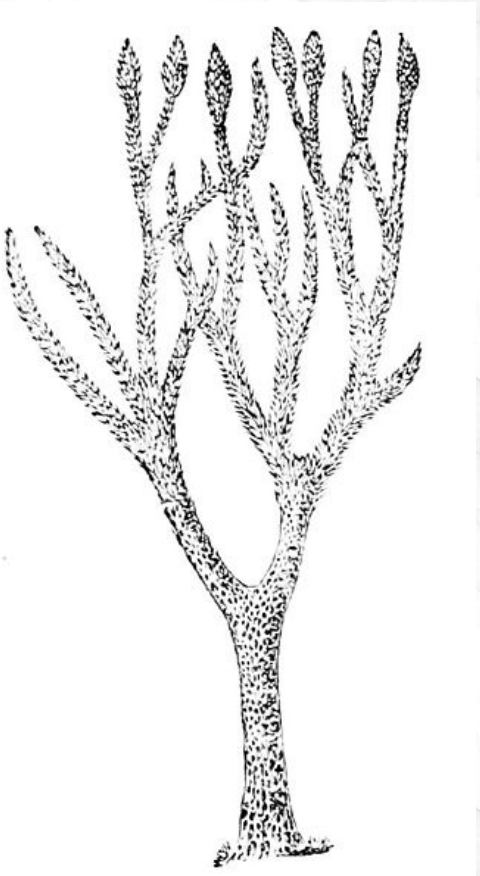
6 Network Equations **88**

7 The Berge–Fulkerson and Berge Conjectures **89**

1 Trees and Spanning Trees







Definitions

- A graph containing no cycle is called a forest, also called an acyclic graph.
- A connected acyclic graph is called a tree. In Figure 1, the graph in (a) is a forest, while those in (b) and (c) are trees; the graph in (c) is also called an **out-tree rooted** at x.
- Since D is a forest $\Leftrightarrow$ every connected component of D is a tree, and D is a tree $\Leftrightarrow$ its underlying graph G is a tree, in this section we state the properties of trees and forests only for undirected graphs G.

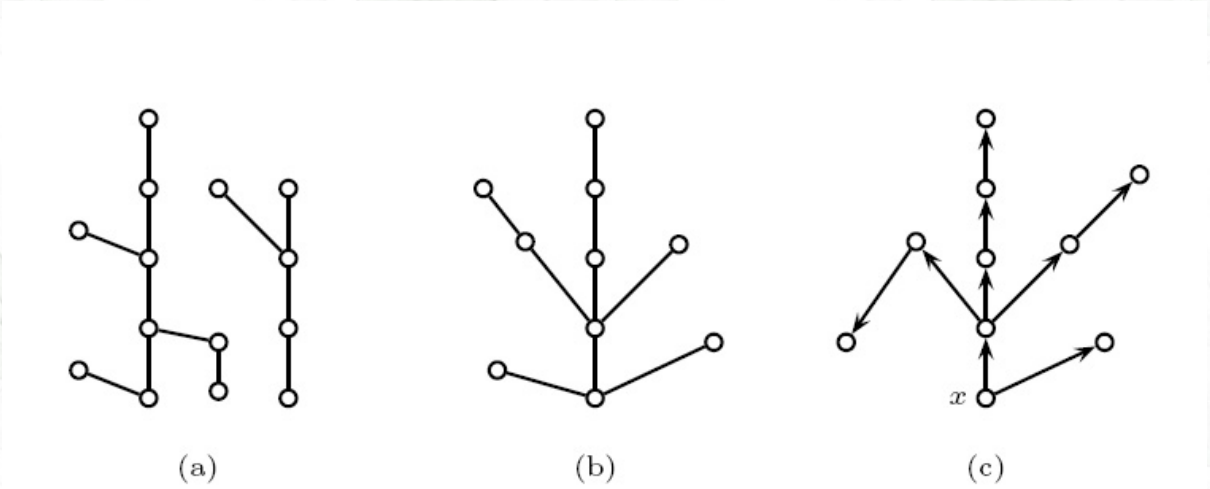


Figure 1: (a) forest (b) tree (c) a tree rooted at x

Theorem 2.1.1 Let G be a simple graph. Then the following statements are equivalent:

(1) G is a tree;

(2) (D. König, 1936) any two distinct vertices of G are joined by exactly one path;

(3) G is connected and $\omega(G-e)=2$ for every $e \in E(G)$;

(4) G has no cycle, but for any two non-adjacent vertices x, y of G, $G+xy$ contains a cycle.

Proof (1) $\Rightarrow$ (2)

- Let $x, y \in V(G)$, and let P_1 and P_2 be two distinct xy-paths in G. Then $P_1 \cup P_2$ is a closed walk. A closed walk can be decomposed into the union of edge-disjoint closed trails, and a closed trail into the union of edge-disjoint cycles; this contradicts the fact that G is acyclic.

(2) $\Rightarrow$ (3)

- That G is connected is clear. Let $e = xy \in E(G)$. Then xey is the unique xy-path in G, so x and y lie in different connected components of $G-e$; hence $\omega(G-e) \geq 2$. Since also $\omega(G-e) \leq 2$, we have $\omega(G-e)=2$.

(3) $\Rightarrow$ (4)

- First we show that G has no cycle. Suppose G contains a cycle C and let $e = xy \in E(C)$. Then $C - e$ is an xy-path in $G - e$, contradicting $\omega(G - e) = 2$.
- Next, for any two non-adjacent vertices x, y in G, $G + xy$ contains a cycle. Since G is connected, it contains an xy-path P, and $P + xy$ is a cycle in $G + xy$.

(4) $\Rightarrow$ (1)

- It suffices to show that G is connected. Suppose G is disconnected, and let G_1, G_2 be two of its components. Take $x \in V(G_1)$ and $y \in V(G_2)$. Then $G + xy$ has no cycle, a contradiction.

Theorem 2.1.2 G is a tree on n vertices $\Leftrightarrow$ G is connected and $e(G) = n-1$.

Proof $(\Rightarrow)$

- Since G is a tree, it is connected.
- We prove $e(G) = n-1$ by induction on $e(G) \geq 0$.
- When $e(G) = 0$, the conclusion is obvious.
- Assume the conclusion holds for every tree with fewer than $e(G)$ edges, and let G be a tree with $e(G)\,(\geq 1)$ edges.
- Take $e \in E(G)$. By Theorem 2.1.1(3), $\omega(G-e) = 2$. Let G_1 and G_2 be the two connected components of $G-e$; both are trees with $e(G_i) < e(G)$, $i = 1,2$. By the induction hypothesis $e(G_i) = |V(G_i)| - 1$. Hence

$$e(G) = e(G_1) + e(G_2) + 1 = n-1.$$

($\Leftarrow$)

- We prove by induction on $n \geq 1$ that G contains no cycle.
- When $n = 1$, $e(G) = 0$ and G is the trivial graph, hence acyclic.
- Now assume every connected graph of order $n (\geq 1)$ with $e(G) = n-1$ is acyclic, and let G be a connected graph of order $n+1$ with $e(G) = n$.
- Since G is connected, $\delta(G) \geq 1$.
- If $\delta(G) \geq 2$, then by Corollary 1.3.1,

$$2n = 2e(G) = \sum_{x \in V} d_G(x) \geqslant 2(n+1),$$

  a contradiction.
- So there exists $x \in V(G)$ with $d_G(x) = 1$.

- Then $G-x$ is a connected graph of order n with $e(G-x)=n-1$. By the induction hypothesis $G-x$ is acyclic, so G is a connected acyclic graph and hence a tree. ∎

Corollary 2.1.3 G is a forest of order $n \Leftrightarrow e(G)=n-\omega(G)$. ∎

Corollary 2.1.4 A forest without isolated vertices has at least 2ω vertices of degree 1. In particular, a nontrivial tree has at least two degree-1 vertices (*called leaves of the tree; a vertex adjacent to a leaf is called a support vertex).*

Proof

- Let G be a forest of order n without isolated vertices; then $\delta(G)\geq 1$.
- Let $X=\{x\in V(G):\ d_G(x)=1\}$.

- By Corollaries 1.3.1 and 2.1.3,

$$2(n-\omega) = 2e(G) = \sum_{x \in V} d_G(x) \geqslant |X| + 2(n-|X|).$$

It follows that $|X| \geq 2\omega$, i.e., G has at least 2ω vertices of degree 1. ∎

As an application of forests and trees, we give an example.

Theorem 2.1.5 [Bondy's Theorem, 1972]
Let $\mathcal{A} = \{A_1, A_2, \cdots, A_n\}$ be a family of n distinct subsets of $X = \{1, 2, \cdots, n\}$. Then there exists $x \in X$ such that $A_1 \setminus \{x\}, A_2 \setminus \{x\}, \cdots, A_n \setminus \{x\}$ are all distinct.

Proof

- Observe that if for some $i \in X$ and $A, B \in \mathcal{A}$ we have $A \setminus \{i\} = B \setminus \{i\}$, then either $A = B \cup \{i\}$ or $B = A \cup \{i\}$; that is, the symmetric difference of A and B is

$$A \Delta B = (A \cup B) \setminus (A \cap B) = \{i\}.$$

- Construct a labeled graph G with $V(G) = \{A_1, A_2, \cdots, A_n\}$ and

$$A_i A_j \in E(G) \text{ with label } x \Leftrightarrow A_i \Delta A_j = \{x\}.$$

  Let $L(E)$ denote the set of edge labels of G, i.e.,

$$L(E) = \{l(e) : e \in E(G)\}.$$

- The claim is equivalent to showing that $L(E)$ is a proper subset of X, i.e., $|L(E)| < n$. If G is acyclic, then $|L(E)| \leq e(G) \leq n-1$ and the claim follows. Now suppose G contains a cycle.

- Let $C = (A_{i_1}, A_{i_2}, \cdots, A_{i_s}, A_{i_1})$ be a cycle in G. From

$$\begin{aligned} A_{i_1} \Delta A_{i_s} &= (A_{i_1} \Delta A_{i_2}) \Delta (A_{i_2} \Delta A_{i_3}) \Delta \cdots \Delta (A_{i_{s-1}} \Delta A_{i_s}) \\ &\subseteq \bigcup_{j=1}^{s-1} (A_{i_j} \Delta A_{i_{j+1}}) \end{aligned}$$

  we see that the label of every edge of C appears at least twice on C.

- Let G' be obtained by deleting any one edge of C; then $L(E(G')) = L(E(G))$.

- If G' still contains a cycle, repeat the operation to obtain G'' with $L(E(G'')) = L(E(G')) = L(E(G))$. Continuing this edge deletion yields a forest F with $L(E(F)) = L(E(G))$.

- Since $e(F) \leq n-1$, we have $|L(E)| \leq n-1$.

- Thus there exists $i \in X$ with $i \notin L(E)$; this i is the required element. ∎

Problem 2.1.6 (Hajnal, 1972) Let $n > k$ and $m = 1 + \sum_{i=0}^{k-1} \binom{n}{i}$. If $\mathcal{A} = \{A_1, A_2, \cdots, A_m\}$ is a family of m distinct subsets of $X = \{1, 2, \cdots, n\}$, then there exists $x \in X$ such that the family $\{A \setminus \{x\} : A \in \mathcal{A}\}$ contains $1 + \sum_{i=0}^{k-1} \binom{n-1}{i}$ distinct subsets.

(It has been proved that there exists an $(n-k)$-subset X' of X such that the family $\{A \setminus X' : A \in \mathcal{A}\}$ contains 2^k distinct subsets (Sauer, 1972).)

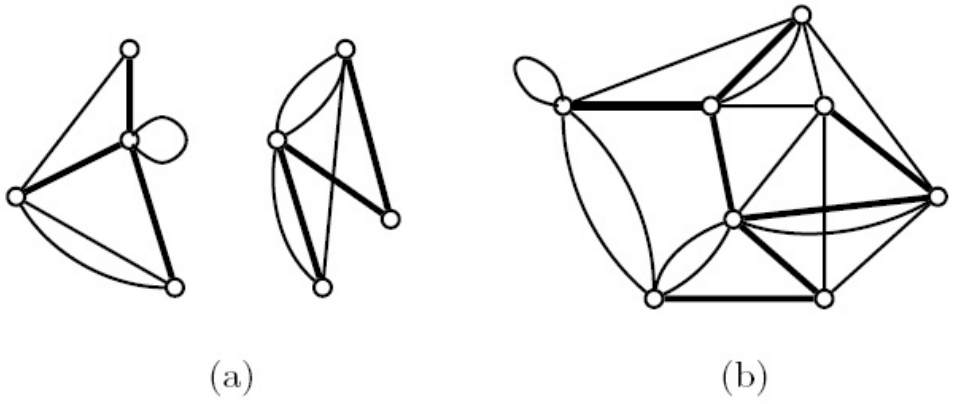


Figure 2: (a) spanning forest (b) spanning tree

Spanning Forests and Spanning Trees

- Let F be a spanning subgraph of a graph D with $\omega(F) = \omega(D)$.
- If F is a forest, it is called a spanning forest of D;
- if F is a tree, it is called a spanning tree of D.

Theorem 2.1.7 (D. König, 1936) Every connected graph contains a spanning tree.

Proof

- Let T be a connected spanning subgraph of the connected graph G with as few edges as possible.
- Then for every $e \in E(T)$ we have $\omega(T - e) = 2$.
- By Theorem 2.1.1(3), T is a tree. ∎

Corollary 2.1.8 Every graph G contains a spanning forest (or spanning tree), and $e(G) \geq |V(G)| - \omega(G)$.

Theorem 2.1.9 (D. König, 1936) Let F be a spanning forest of G. If $e \in E(G) \setminus E(F)$, then $F + e$ contains exactly one cycle (called the fundamental cycle of G with respect to F).

Proof

- Let $e \in E(G) \setminus E(F)$.
- Since $\omega(F) = \omega(G)$, the two endpoints of e lie in one and the same component of F, so $F + e$ contains a cycle.
- Suppose $F + e$ contains two distinct cycles C_1 and C_2. Since F is acyclic, $e \in E(C_1 \cap C_2)$.
- Thus $(C_1 \cup C_2) - e$ is a closed walk in F and hence contains a cycle C', contradicting that F is a forest. Therefore $F + e$ contains exactly one cycle. ∎

Corollary 2.1.10 A graph G of order n contains exactly $e(G) - n + \omega$ distinct fundamental cycles. ∎

The number $e(G)-n+\omega$ is called the cycle rank of G, also known as its Betti number.

Cotrees and Edge-Cut Sets

- Let H be a subgraph of G; then $G-E(H)$ is called the cograph of H in G, denoted by $\overline{H}(G)$.
- It is easy to see that if $G=K_n$, then $\overline{H}(K_n)=H^c$.
- If F is a spanning forest (or tree) of G, then $\overline{F}(G)$ is called the coforest (or cotree) of G.
- When no confusion arises, we abbreviate $\overline{F}(G)$ to $\overline{F}$ and $\overline{T}(G)$ to $\overline{T}$. Sometimes $\overline{F}$ (or $\overline{T}$) is also used in place of $E(\overline{F})$ (or $E(\overline{T})$).
- For example, in Figure 2(a) the thick-edge subgraph is a spanning forest F of G and the thin-edge subgraph is its coforest $\overline{F}(G)$; in

(b) the thick-edge subgraph is a spanning tree T and the thin-edge subgraph is its cotree $\overline{T}(G)$.

- Let B be a nonempty subset of E. If there is a nonempty subset $S \subset V(G)$ such that $B = [S, \overline{S}]$, then B is called an edge-cut set of G.

- If no nonempty proper subset of B is an edge-cut set, then B is called a minimal edge-cut set, also known as a bond.

- An edge-cut set of size 1 is a cut edge, also called a bridge.

- Claim: B is a minimal edge-cut set if and only if $\omega(G-B) = \omega(G)+1$.

  Proof. Deleting a single edge increases the number of components by at most one. If B is a minimal edge-cut set and $\omega(G-B) \geq \omega(G)+2$, then deleting the edges of B one at a time, some proper nonempty subset $B' \subsetneq B$ already increases the number of components (so B' is an edge-cut set), contradicting the minimality of B;

hence $\omega(G-B) = \omega(G)+1$. Conversely, if $\omega(G-B) = \omega(G)+1$, then deleting B splits exactly one component of G into two connected parts with vertex sets S and $\overline{S}$, and $B = [S, \overline{S}]$. Every proper subset of B leaves at least one edge joining S to $\overline{S}$ and therefore is not an edge-cut set; hence B is minimal, i.e. B is a bond.

- If e is a cut edge of G, then e is a bond. Figure 3 shows an edge-cut set and a bond (in thick edges).

Theorem 2.1.11 (D. König, 1936) Let F be a spanning forest of a nonempty graph G and $e \in E(F)$. Then

(i) $\overline{F}$ contains no bond;

(ii) $\overline{F}+e$ contains exactly one bond. (called the fundamental bond of G with respect to F)

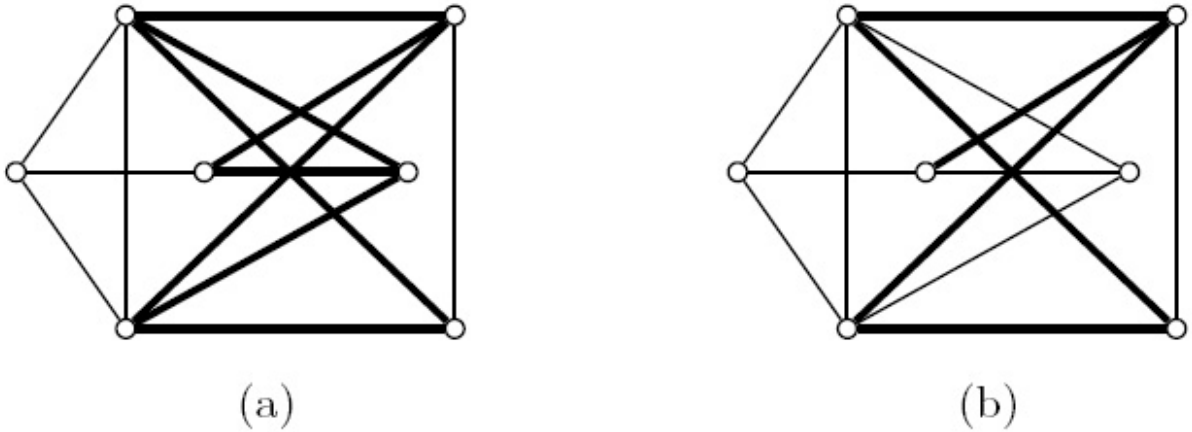


Figure 3: (a) edge-cut set (b) bond

Proof (i) Deleting $\overline{F}$ from G leaves the spanning forest F with $\omega(F) = \omega(G)$, so $\overline{F}$ is not an edge-cut set of G. Hence none of its subsets is an edge-cut set of G, and $\overline{F}$ cannot contain a bond.

(ii)

- Let the tree of F that contains e be split, on deleting e, into two

components with vertex sets S and S'. In G, the only edge of F joining S to S' is e; every other edge of G between S and S' lies in $\overline{F}$. Hence the cut $B = [S, S']$ satisfies $B \subseteq \overline{F} \cup \{e\} = \overline{F} + e$. Since both $G[S]$ and $G[S']$ are connected (each contains a spanning tree inherited from F), B is a bond of G contained in $\overline{F} + e$.

- Now let B' be any bond of G with $B' \subseteq \overline{F} + e$. By (i), B' cannot be contained in $\overline{F}$, so $e \in B'$. A bond containing e has the form $[U, \overline{U}]$ with $G[U]$ and $G[\overline{U}]$ connected, and it separates the two endpoints of e. Because B' contains no edge of F other than e, it cannot separate the endpoints of any edge of $F - e$; consequently all vertices of S lie on one side of the cut and all vertices of S' on the other.

- Thus the cut is forced to be $[S, S'] = B$, independently of the choice of U. Hence B is the unique bond of G contained in $\overline{F} + e$. ∎

Corollary 2.1.12 A graph G contains exactly $|V(G)| - \omega(G)$ distinct fundamental bonds. ∎

Exercises

2.1.3 Let T be a nontrivial tree. Prove:

(a) T is bipartite;

(b) Let $\{X,Y\}$ be the bipartition of T. Then:

(i) if $|X| \geq |Y|$, then X contains at least one vertex of degree 1;

(ii) if $|X| = |Y| + k$, then X contains at least $k+1$ vertices of degree 1.

2.1.10 Let $\mathcal{A} = \{A_1, A_2, \cdots, A_n\}$ be a family of n distinct subsets of $X = \{1, 2, \cdots, n\}$. Prove that there exists $x \in X$ such that $A_1 \cup \{x\}, A_2 \cup \{x\}, \cdots, A_n \cup \{x\}$ are all distinct.

2.1.16 Prove: if B_1 and B_2 are bonds, then their symmetric difference $B_1 \Delta B_2$ is an edge-cut set and hence contains a bond.

2 Vector Spaces of a Graph

Vertex Space and Edge Space

- Let D be a graph with $V(D) = \{u_1, u_2, \cdots, u_n\}$ and $E(D) = \{a_1, a_2, \cdots, a_m\}$.
- The vertex space $\mathcal{V}(D)$ of D is the vector space, under vector addition and scalar multiplication, of all functions from $V(D)$ to the set $\mathbf{R}$ of real numbers; that is,

$$\mathcal{V}(D) = (\{\mathbf{f} \ : \ V(D) \to R\}, \ +, \ \text{scalar multiplication}).$$

- Define functions $\mathbf{u}_i$ by:

$$\mathbf{u}_i(u_j) = \begin{cases} 1, & \text{if } j = i; \\ 0, & \text{if } j \neq i, \end{cases}$$

- Then for $\mathbf{f} \in \mathcal{V}(D)$, if $\mathbf{f}(u_i) = x_i$,

$$\mathbf{f} = \sum_{i=1}^{n} x_i \mathbf{u}_i.$$

  i.e., $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n$ form a basis of $\mathcal{V}(D)$, so $\dim \mathcal{V}(D) = n$. $(x_1, x_2, \cdots, x_n)$ are the coordinates of $\mathbf{f}$ in this basis, and we write $\mathbf{f} = (x_1, x_2, \cdots, x_n)$.

- The edge space $\mathcal{E}(D)$ of D is the vector space, under vector addition and scalar multiplication, of all functions from $E(D)$ to $\mathbf{R}$; that is,

$$\mathcal{E}(D) = (\{\mathbf{g} : E(D) \to R\}, +, \text{scalar multiplication})$$

- Likewise, let $\mathbf{g} \in \mathcal{E}(D)$ with $\mathbf{g}(a_i) = y_i$. We can write

$$\mathbf{g} = \sum_{i=1}^{m} y_i \mathbf{a}_i, \quad \text{where } \mathbf{a}_i(a_j) = \begin{cases} 1, & \text{if } j = i; \\ 0, & \text{if } j \neq i, \end{cases}$$

i.e., $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m$ form a basis of $\mathcal{E}(D)$, $(y_1, y_2, \cdots, y_m)$ are the coordinates of $\mathbf{g}$ in this basis, and we write $\mathbf{g} = (y_1, y_2, \cdots, y_m)$. Clearly $\dim \mathcal{E}(D) = m$.

Cycle Space and Cut Space

- For $\mathbf{w} \in \mathcal{E}(D)$, $(D, \mathbf{w})$ is called a weighted graph; $\mathbf{w}$ is the weight function and $\mathbf{w}(a)$ is the **weight** of edge a.
- Weights are usually given in the form of a matrix, called a weight matrix. In fact, the weight matrix generalizes the adjacency matrix, since when $\mathbf{w}(a) \equiv 1$ the weight matrix is exactly the adjacency matrix.
- Weighted graphs arise frequently in applications: depending on the problem, a weight may be a distance, a cost, and so on. The corresponding weight matrices are called a distance matrix and a cost matrix, respectively.

- For $B \subseteq E(D)$, write

$$\mathbf{w}(B) = \sum_{a \in B} \mathbf{w}(a).$$

- For $S \subset V(D)$, let $E_D^+(S) = (S, \overline{S})$ be the set of edges of D whose tails lie in S and heads in $\overline{S}$; likewise $E_D^-(S) = (\overline{S}, S)$ is the set of edges whose tails lie in $\overline{S}$ and heads in S.

$$\mathbf{w}^+(S) = \mathbf{w}(E_D^+(S)), \quad \mathbf{w}^-(S) = \mathbf{w}(E_D^-(S)).$$

- Let $\mathbf{f} \in \mathcal{E}(D)$. If

$$\mathbf{f}^+(u) = \mathbf{f}^-(u), \quad \forall\ u \in V(D), \tag{1}$$

then $\mathbf{f}$ is called a cycle vector of D. Figure 4(a) shows a cycle vector $\mathbf{f}$ of D.

- Let C be a cycle in D (not necessarily directed) and fix a positive orientation of C. Let C^+ be the edges of C whose direction agrees

with this orientation and C^- those whose direction is opposite. Define $\mathbf{f}_C \in \mathcal{E}(D)$ by

$$\mathbf{f}_C(a) = \begin{cases} 1, & \text{if } a \in C^+; \\ -1, & \text{if } a \in C^-; \\ 0, & \text{if } a \notin C^+ \cup C^-. \end{cases}$$

It is easy to check that $\mathbf{f}_C$ satisfies (1), so it is a cycle vector of D, called the cycle vector of the cycle C.

- By Corollary 2.1.10, D has $m - n + \omega$ fundamental cycles $C_1, C_2, \cdots, C_{m-n+\omega}$; it is easy to verify that their cycle vectors $\mathbf{f}_{C_1}, \mathbf{f}_{C_2}, \cdots, \mathbf{f}_{C_{m-n+\omega}}$ are linearly independent.

- It is not hard to show that all cycle vectors of D form a subspace of $\mathcal{E}(D)$, called the cycle space of D and denoted $\mathcal{C}(D)$. Clearly $\dim \mathcal{C}(D) \geq m - n + \omega$.

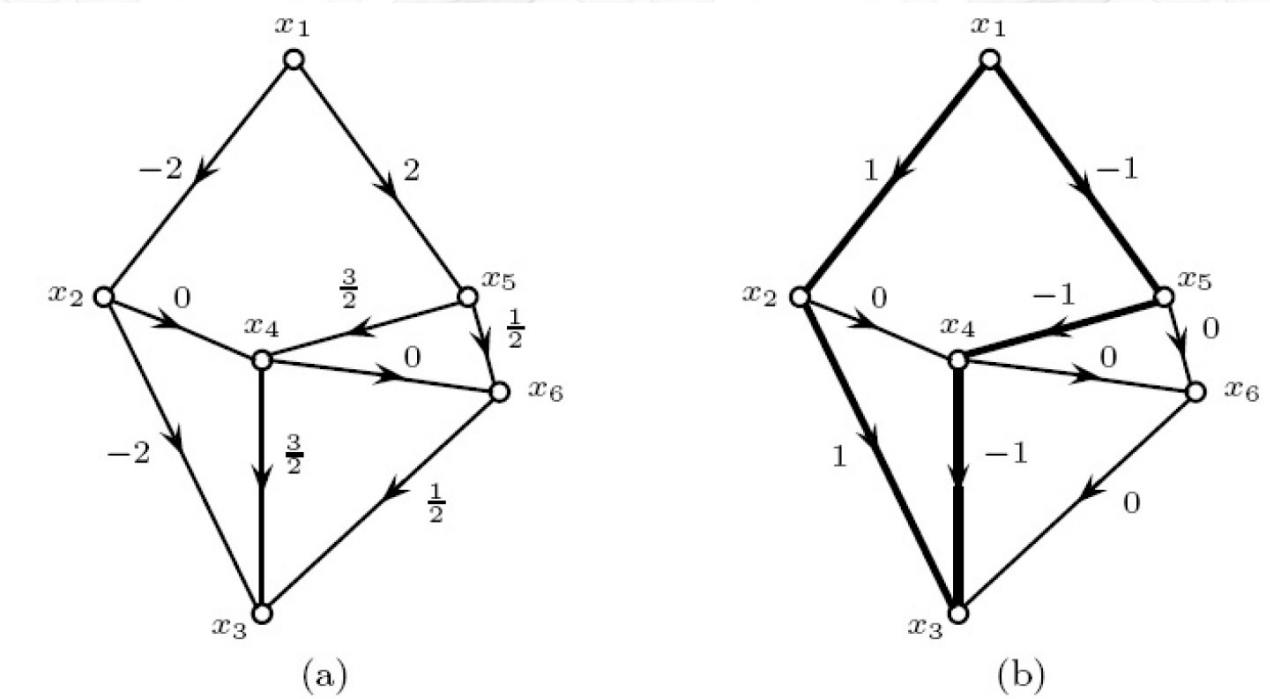


Figure 4: (a) a cycle vector in a weighted graph (b) the cycle vector corresponding to a cycle

- For $\mathbf{p} \in \mathcal{V}(D)$, define $\delta_{\mathbf{p}} \in \mathcal{E}(D)$ by

$$\delta_{\mathbf{p}}(a) = \mathbf{p}(x) - \mathbf{p}(y), \ \forall \ a = (x,y) \in E(D). \tag{2}$$

$\delta_{\mathbf{p}}$ is called a cut vector of D. Figure 5(a) shows $\mathbf{p} \in \mathcal{V}(D)$ and its associated $bf\delta_p$.

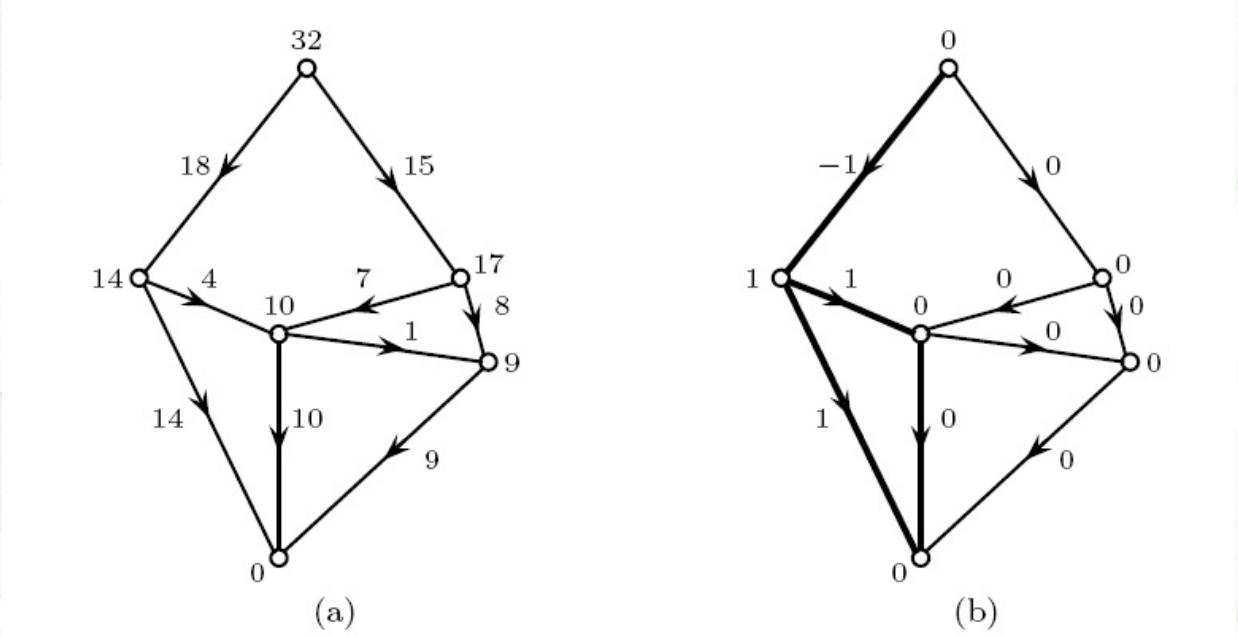


Figure 5: (a) a cut vector in a weighted graph (b) the cut vector corresponding to a bond

- Let $B = E_D[S, \overline{S}]$ be a bond of D. Define $\mathbf{g}_B \in \mathcal{E}(D)$ by

$$\mathbf{g}_B(a) = \begin{cases} 1, & \text{if } a \in (S, \overline{S}); \\ -1, & \text{if } a \in (\overline{S}, S); \\ 0, & \text{if } a \notin B. \end{cases}$$

  If we set

$$\mathbf{p}(u) = \begin{cases} 1, & u \in S; \\ 0, & u \notin S. \end{cases}$$

  then $\mathbf{g}_B = \delta_{\mathbf{p}}$, so $\mathbf{g}_B$ is a cut vector of D, called the cut vector of the bond B. Figure 5(b) shows the cut vector $\mathbf{g}_B$ corresponding to the bond B (in thick edges).

- By Corollary 2.1.12, D has $n - \omega$ fundamental bonds, and it is easy to verify that their associated cut vectors are linearly independent.

- It is not hard to show that all cut vectors of D form a subspace of $\mathcal{E}(D)$, called the cut space of D and denoted $\mathcal{B}(D)$. Clearly $\dim \mathcal{B}(D) \geq n - \omega$.

Theorem 2.2.1 Let $\mathbf{M}$ be the incidence matrix of D. Then $\mathcal{B}(D)$ is the row space $\mathcal{M}$ of $\mathbf{M}$, and $\mathcal{C}(D)$ is its orthogonal complement.

Proof

- Let $\mathbf{g} \in \mathcal{B}(D)$. Then there exists $\mathbf{p} \in \mathcal{V}(D)$ with $\delta_{\mathbf{p}} = \mathbf{g}$ and $\mathbf{p}$ satisfying (2).
- Hence, for every $a = (x, y) \in E(D)$,

$$\mathbf{g}(a) = \delta_{\mathbf{p}}(a) = \mathbf{p}(x) - \mathbf{p}(y) = \sum_{u \in V} m_u(a)\mathbf{p}(u),$$

where $m_u(a)$ is the component on edge a of the row vector of $\mathbf{M}$ corresponding to vertex u, namely

$$m_u(a) = \begin{cases} 1, & \text{if } a \in E^+(u); \\ -1, & \text{if } a \in E^-(u); \\ 0, & \text{otherwise.} \end{cases}$$

- Thus $\mathbf{g} = \sum_{u \in V} \mathbf{p}(u)\mathbf{m_u}$ is a linear combination of the row vectors of $\mathbf{M}$, i.e., $\mathcal{B}(D) \subseteq \mathcal{M}$.
- Conversely, let $\mathbf{m}_x = (m_x(a_1), m_x(a_2), \cdots, m_x(a_m))$ be any row vector of $\mathbf{M}$; clearly $\mathbf{m}_x \in \mathcal{E}(D)$.
- Set

$$\mathbf{p}_x(u) = \begin{cases} 1, & \text{if } u = x; \\ 0, & \text{if } u \neq x. \end{cases}$$

  Then $\mathbf{p}_x \in \mathcal{V}(D)$ and $\mathbf{m}_x = \delta_{\mathbf{p}_x}$; therefore, $\mathbf{m}_x \in \mathcal{B}(D)$.
- Since any linear combination of row vectors of M still belongs to $\mathcal{B}(D)$, we have $\mathcal{M} \subseteq \mathcal{B}(D)$.
- Hence $\mathcal{M} = \mathcal{B}(D)$.
- We next show that $\mathcal{C}(D)$ is the orthogonal complement of $\mathcal{B}(D)$ in

$\mathcal{E}(D)$. Let $\mathbf{f} \in \mathcal{C}(D)$. Then

$$\begin{aligned}
\mathbf{f} \text{ satisfies } (1) \quad &\Leftrightarrow \mathbf{f}^+(x) - \mathbf{f}^-(x) = 0, \ \forall\ x \in V(D) \\
&\Leftrightarrow \sum_{a \in E^+(x)} \mathbf{f}(a) - \sum_{a \in E^-(x)} \mathbf{f}(a) = 0 \\
&\Leftrightarrow \sum_{a \in E} m_x(a)\,\mathbf{f}(a) = 0, \ \forall\ x \in V(D), \\
&\Leftrightarrow \mathbf{f} \text{ is orthogonal to every row vector of } \mathbf{M}.
\end{aligned}$$

- Moreover, since $\dim \mathcal{C}(D) + \dim \mathcal{B}(D) \geq m$, $\mathcal{C}(D)$ is the orthogonal complement of $\mathcal{B}(D)$ in $\mathcal{E}(D)$. ∎

Corollary 2.2.2 For every graph D, $\mathcal{E}(D) = \mathcal{B}(D) \oplus \mathcal{C}(D)$. ∎

Corollary 2.2.3 (i)

$$\dim \mathcal{B} = n - \omega;$$

(ii)

$$\dim \mathcal{C} = m - n + \omega.$$

- Let $\mathbf{f} \in \mathcal{E}(D)$. Denote by $D_{\mathbf{f}}$ the subgraph of D induced by the edges on which $\mathbf{f}$ is nonzero; $D_{\mathbf{f}}$ is called the support of $\mathbf{f}$.
- For example, for $\mathbf{f}$ in Figure 4(b), its support $D_{\mathbf{f}}$ is a cycle (in thick edges); while for $\mathbf{g}$ in Figure 5(b), its support $D_{\mathbf{g}}$ is a bond of D (in thick edges).

In general, we have the following result.

Lemma 2.2.4 (i) If a nonzero $\mathbf{f} \in \mathcal{C}(D)$, then $D_{\mathbf{f}}$ contains a cycle.

(ii) If a nonzero $\mathbf{g} \in \mathcal{B}(D)$, then $D_{\mathbf{g}}$ contains a bond.

Proof

(i) Since $0 \neq \mathbf{f} \in \mathcal{C}(D)$, equation (1) gives $d_{D_\mathbf{f}}(u) \geq 2$ for every $u \in V(D_\mathbf{f})$, i.e., $\delta(D_\mathbf{f}) \geq 2$; hence $D_\mathbf{f}$ contains a cycle.

(ii) Since $0 \neq \mathbf{g} \in \mathcal{B}(D)$, there is a nonzero $\mathbf{p} \in \mathcal{V}(D)$ with $\mathbf{g} = \delta_\mathbf{p}$.

Take any $u \in V(D_\mathbf{g})$ and set

$$S = \{w \in V(D) : \ \mathbf{p}(w) = \mathbf{p}(u)\}.$$

Then $E_D[S, \overline{S}]$ is an edge-cut set of D, and $\mathbf{g}(a) \neq 0$ for every $a \in E_D[S, \overline{S}]$, so $E_D[S, \overline{S}] \subseteq E(D_\mathbf{g})$; hence (by Exercise 2.1.16) $D_\mathbf{g}$ contains a bond. ∎

- Let $\mathbf{B}$ be a matrix whose rows are vectors of $\mathcal{E}(D)$, and let $R \subseteq E(D)$. Write $\mathbf{B}|R$ for the submatrix of $\mathbf{B}$ whose columns are restricted to R. To simplify notation, if R is a subgraph of D, we also write $\mathbf{B}|R$ for $\mathbf{B}|E(R)$.

Theorem 2.2.5 Let $\mathbf{B}$ and $\mathbf{C}$ be basis matrices of $\mathcal{B}(D)$ and $\mathcal{C}(D)$, respectively. For any $R \subseteq E(D)$,

(i) the columns of $\mathbf{B}|R$ are linearly independent $\Leftrightarrow D[R]$ is acyclic;

(ii) the columns of $\mathbf{C}|R$ are linearly independent $\Leftrightarrow D[R]$ contains no bond.

Proof (i) Write $\mathbf{B}_a$ for $\mathbf{B}|\{a\}$, the column vector of $\mathbf{B}$ corresponding to edge a.

($\Leftarrow$) (by contradiction)

- Suppose the columns of $\mathbf{B}|R$ are linearly dependent. Then there exists $\mathbf{0} \neq \mathbf{f} \in \mathcal{E}(D)$ with

$$\mathbf{f}(a) \begin{cases} \not\equiv 0, & \text{if } a \in R; \\ = 0, & \text{if } a \notin R \end{cases} \quad \text{such that } \sum_{a \in R} \mathbf{f}(a)\mathbf{B}_a = 0.$$

- That is,

$$\mathbf{0} = \sum_{a \in R} \mathbf{f}(a)\mathbf{B}_a = \sum_{a \in E(D)} \mathbf{f}(a)\mathbf{B}_a = \mathbf{B}\mathbf{f}^T.$$

- Thus $D_{\mathbf{f}} \subseteq D[R]$, and $\mathbf{f}$ is orthogonal to every row vector, hence every vector, of $\mathcal{B}(D)$, i.e., $\mathbf{f} \in \mathcal{B}(D)^{\perp}$.
- By Corollary 2.2.2, $\mathbf{f} \in \mathcal{C}$; by Lemma 2.2.4(i), $D_{\mathbf{f}}$ contains a cycle, hence so does $D[R]$, contradicting the assumption. Therefore the columns of $\mathbf{B}|R$ must be linearly independent.

($\Rightarrow$) (by contradiction)

- If $D[R]$ contains a cycle, let C be such a cycle in $D[R]$. Then $\mathbf{f}_C$ is the nonzero cycle vector of D corresponding to C, i.e., $\mathbf{0} \neq \mathbf{f}_C \in \mathcal{C}(D)$.
- By Corollary 2.2.2, $\mathbf{f}_C$ is orthogonal to every row vector of $\mathbf{B}$, i.e.,

$$\sum_{a \in E(D)} \mathbf{f}_C(a)\mathbf{B}_a = \mathbf{B}\mathbf{f}_C^T = \mathbf{0}.$$

- On the other hand,

$$\mathbf{0} = \sum_{a \in E(D)} \mathbf{f}_C(a)\mathbf{B}_a = \sum_{a \in R} \mathbf{f}_C(a)\mathbf{B}_a = (B|R)(\mathbf{f}_C|R)^T,$$

- i.e., the columns of $\mathbf{B}|R$ are linearly dependent, contradicting the assumption; hence $D[R]$ is acyclic.

- Part (ii) follows by a similar argument using Lemma 2.2.4(ii). ∎

- Let F be a spanning forest of D. Label the edges of D by $a_1, a_2, \cdots, a_m$ so that

$$E(\overline{F}) = \{a_1, a_2, \cdots, a_{m-n+\omega}\},\ E(F) = \{a_{m-n+\omega+1}, \cdots, a_m\}.$$

- By Theorem 2.1.9, for each $a_i \in E(\overline{F})$, $F + a_i$ contains a unique cycle (denoted C_i, called the **fundamental cycle of D corresponding to F**).

- Let $\mathbf{f}_i$ be the cycle vector corresponding to C_i, normalized so that $\mathbf{f}_{C_i}(a_i) = 1$.
- Then the matrix $\mathbf{C}_F$ whose rows are

$$\mathbf{f}_1,\ \mathbf{f}_2,\ \cdots,\ \mathbf{f}_{m-n+\omega} \tag{3}$$

has order $(m-n+\omega) \times m$ and admits the following block form:

$$\mathbf{C}_F = (\mathbf{I}_{m-n+\omega}\ \mathbf{C}_2),$$

Here $\mathbf{I}_{m-n+\omega} = \mathbf{C}_F|\overline{F}$ is the identity matrix of order $m-n+\omega$, and $\mathbf{C}_2 = \mathbf{C}_F|F$ is an $(m-n+\omega) \times (n-\omega)$ matrix.

- Since $\text{rank}(\mathbf{C}_F) = m-n+\omega$, the vectors in (3) form a basis of the cycle space $\mathcal{C}(D)$; $\mathbf{C}_F$ is called the basis matrix of $\mathcal{C}(D)$ corresponding to F.
- Similarly, let $a_j \in E(F)$. By Theorem 2.1.11, $\overline{F} + a_j$ contains a unique bond B_j (called the **fundamental bond of D corresponding to F**).

- Let $\mathbf{g}_j$ be the cut vector corresponding to B_j, normalized so that $\mathbf{g}_{B_j}(a_j) = 1$, $j = m-n+\omega+1, \cdots, m$. The matrix $\mathbf{B}_F$ of order $(n-\omega) \times m$ whose rows are

$$\mathbf{g}_{m-n+\omega+1},\ \mathbf{g}_{m-n+\omega+2},\ \cdots,\ \mathbf{g}_m \tag{4}$$

has the following block form:

$$\mathbf{B}_F = (\mathbf{B}_1\ \mathbf{I}_{n-\omega}),$$

Here $\mathbf{B}_1 = \mathbf{B}_F|\overline{F}$ is an $(n-\omega) \times (m-n+\omega)$ matrix, and $\mathbf{I}_{n-\omega} = \mathbf{B}_F|F$ is the identity matrix of order $n-\omega$.

- Since rank $\mathbf{B}_F = n-\omega$, the vectors in (4) form a basis of the cut space $\mathcal{B}(D)$; $\mathbf{B}_F$ is called the basis matrix of $\mathcal{B}(D)$ corresponding to F.

- For example, for the weighted graph $(D, \mathbf{f})$ in Figure 4, the edge labels and spanning tree T are shown in Figure 6 (thick edges).

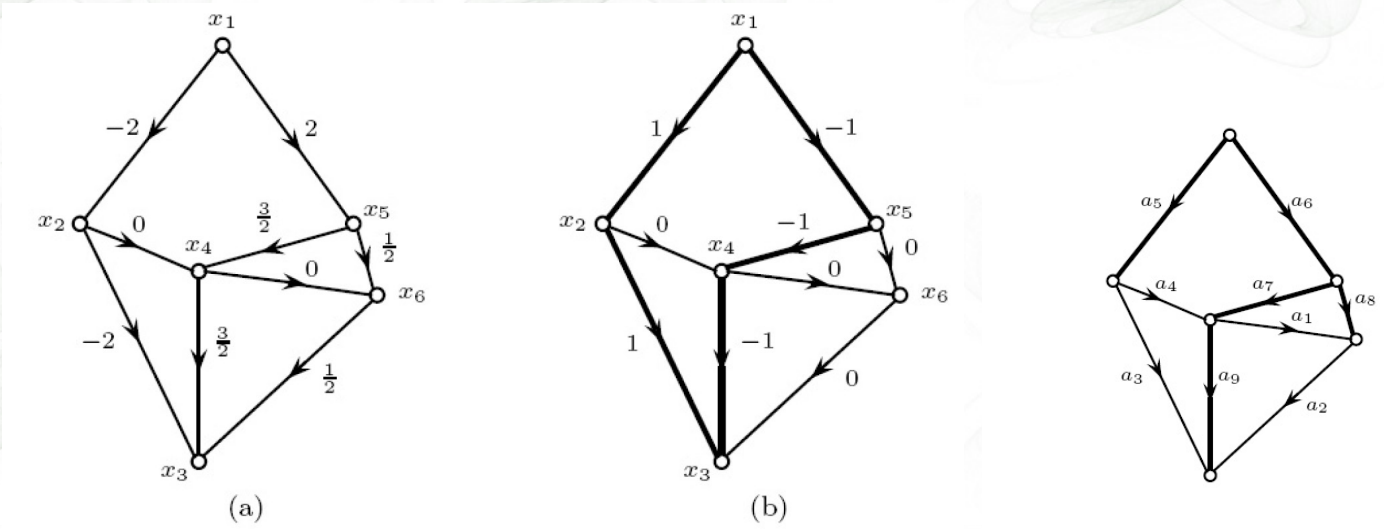


Figure 6: A digraph D and one of its spanning trees

$E(\overline{T}) = \{a_1, a_2, a_3, a_4, \}$, the fundamental cycle vectors $\mathbf{f}_i$ $(i = 1,2,3,4)$ of D corresponding to T are:

$$\begin{aligned}
\mathbf{f}_1 &= (1, \quad 0, \quad 0, \quad 0, \quad 0, \quad 0, \quad 1, \quad -1, \quad 0) \\
\mathbf{f}_2 &= (0, \quad 1, \quad 0, \quad 0, \quad 0, \quad 0, \quad -1, \quad 1, \quad -1) \\
\mathbf{f}_3 &= (0, \quad 0, \quad 1, \quad 0, \quad 1, \quad -1, \quad -1, \quad 0, \quad -1) \\
\mathbf{f}_4 &= (0, \quad 0, \quad 0, \quad 1, \quad 1, \quad -1, \quad -1, \quad 0, \quad 0)
\end{aligned} \tag{5}$$

- the basis matrix of $\mathcal{C}(D)$ corresponding to T

$$\mathbf{C}_T = \begin{array}{c} \begin{array}{ccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 \end{pmatrix} \end{array} = (\mathbf{I}_4, \mathbf{C}_2). \tag{6}$$

- The cycle vector $\mathbf{f}$ in Figure 4 can be expressed as

$$\begin{aligned} \mathbf{f} = & \quad (\mathbf{f}(a_1), \mathbf{f}(a_2), \mathbf{f}(a_3), \cdots, \mathbf{f}(a_8), \mathbf{f}(a_9)) \\ = & \quad \left(0,\ \tfrac{1}{2},\ -2,\ 0,\ -2,\ 2,\ \tfrac{3}{2},\ \tfrac{1}{2},\ \tfrac{3}{2}\right) \\ = & \quad 0 \cdot \mathbf{f}_1 + \tfrac{1}{2} \cdot \mathbf{f}_2 + (-2) \cdot \mathbf{f}_3 + 0 \cdot \mathbf{f}_4 \\ = & \quad \sum_{i=1}^{4} \mathbf{f}(a_i)\, \mathbf{f}_i. \end{aligned}$$

- This shows that the values of $\mathbf{f}$ are uniquely determined by its values on $\mathbf{f}|\overline{T}$. In fact, the same holds for every cycle vector, as follows immediately from the block form $\mathbf{C}_F = (\mathbf{I}\ \mathbf{C}_2)$.

- For example, for the digraph D and spanning tree T (thick edges) in Figure 6, $E(T) = \{a_5, a_6, a_7, a_8, a_9\}$; the fundamental cut vectors $\mathbf{g}_i$ $(i = 5,6,7,8,9)$ of D corresponding to T are:

$$\begin{array}{rl} \mathbf{g}_5 = & (\ 0,\ \ 0,\ -1,\ -1,\ \ 1,\ 0,\ 0,\ 0,\ 0) \\ \mathbf{g}_6 = & (\ 0,\ \ 0,\ \ 1,\ \ 1,\ 0,\ 1,\ 0,\ 0,\ 0) \\ \mathbf{g}_7 = & (-1,\ \ 1,\ \ 1,\ \ 1,\ 0,\ 0,\ 1,\ 0,\ 0) \\ \mathbf{g}_8 = & (\ 1,\ -1,\ \ 0,\ \ 0,\ 0,\ 0,\ 0,\ 1,\ 0) \\ \mathbf{g}_9 = & (\ 0,\ \ 1,\ \ 1,\ \ 0,\ 0,\ 0,\ 0,\ 0,\ 1) \end{array} \tag{7}$$

- the basis matrix of $\mathcal{B}(D)$ corresponding to the spanning tree T

$$\begin{array}{rl} & \begin{array}{ccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{array} \\ \mathbf{B}_T = & \begin{pmatrix} 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\ = & (\mathbf{B}_1\ \mathbf{I}_5). \end{array} \tag{8}$$

- The cut vector $\mathbf{g}$ in Figure 5 can be expressed as

$$\begin{aligned}\mathbf{g} &= (1,\ 9,\ 14,\ 4,\ 18,\ 15,\ 7,\ 8,\ 10)\\ &= 18\cdot\mathbf{g}_5 + 15\cdot\mathbf{g}_6 + 7\cdot\mathbf{g}_7 + 8\cdot\mathbf{g}_8 + 10\cdot\mathbf{g}_9\\ &= \sum_{j=5}^{9}\mathbf{g}(a_j)\,\mathbf{g}_j.\end{aligned}$$

- This shows that the values of $\mathbf{g}$ are uniquely determined by its values on $\mathbf{g}|T$; in fact the same holds for every cut vector, as follows immediately from the block form $\mathbf{B}_F = (\mathbf{B}_1\ \mathbf{I})$.

Bondy's conjecture

- Denote $\mathcal{C}_k(G)$ as the subspace of $\mathcal{C}(G)$ spanned by the cycles of length exactly k in G.
- A graph G on n vertices is said to be Hamilton-generated if $\mathcal{C}_n(G) = \mathcal{C}(G)$.

- A natural question is to determine the condition under which $\sum_{k\in I} C_k(G) = C(G)$ for some finite positive integer set I. A still open conjecture originating with Bondy states that:

Conjecture 2.1 (Bondy, 1979) *For every $d \in \mathbb{Z}$, in every 3-connected graph G with $|V(G)| \geq 2d$ and $\delta(G) \geq d$,*

$$\sum_{k\geq 2d-1} C_k(G) = C(G).$$

- Hartman[1](1983) proved $\sum_{k\geq d+1} C_k(G) = C(G)$ for each 2-connected graph G with minimum degree d and $G \neq K_{d+1}$ if d is odd.

---

[1] I. B.-A. Hartman. Long cycles generate the cycle space of a graph. *European J. Combin.*, 4:237–246, 1983.

- Locke[2](1985) proved that the conjecture of Bondy is true under the additional assumption of 'G non-Hamiltonian or $|V(G)| > 4d-5$'.

Theorem 2.2 (H., Yin, 2025+) *There exists $n_0 \in \mathbb{N}^+$ such that for any odd $n > n_0$, if G is a graph with n vertices that has a minimum degree $\delta(G) \geq \frac{n-1}{2}$ and is Hamilton-connected, then $\mathcal{C}_n(G) = \mathcal{C}(G)$.*

Corollary 2.3 *If G is a graph with odd n vertices and $\delta(G) > n/2$, then $\mathcal{C}_n(G) = \mathcal{C}(G)$ for sufficiently large n.*

Exercises

Example 2.2.6 Let $\mathbf{K}$ be obtained from the incidence matrix $\mathbf{M}$ of a connected graph D by deleting any one row; then $\mathbf{K}$ is a basis matrix of the cut space $\mathcal{B}(D)$.

---

[2]S. C. Locke. A basis for the cycle space of a 3-connected graph. *Ann. Discrete Math.*, 27:381–397, 1985.

2.2.2 Let C be a cycle and B a bond of D. Prove:

(a) $\mathbf{f}_C \in \mathcal{E}(D)$ is a cycle vector of D, and all cycle vectors of D form a subspace of $\mathcal{E}(D)$;

(b) $\mathbf{g}_B \in \mathcal{E}(D)$ is a cut vector of D, and all cut vectors of D form a subspace of $\mathcal{E}(D)$.

3 The Number of Spanning Trees

- Let $\tau(D)$ denote the number of spanning trees in a connected loopless graph D.
- This section gives a formula for $\tau(D)$; the proof presented here is due to Tutte (1965).
- Let D be a connected loopless digraph and $\mathbf{B}$ a basis matrix of its cut space $\mathcal{B}(D)$.
- By Theorem 2.2.5(i), if $R \subseteq E(D)$ and $|R| = n-1$, then the square submatrix $\mathbf{B}|R$ is invertible $\Leftrightarrow$ the induced subgraph $D[R]$ is a spanning tree of D.
- Thus $\tau(D)$ equals the number of invertible $(n-1)\times(n-1)$ submatrices of $\mathbf{B}$.
- Let $\mathbf{M}$ be an $n \times m$ matrix. If the determinant of every full-order

(i.e., order $\min\{n,m\}$) square submatrix of $\mathbf{M}$ is 0, -1, or 1, then $\mathbf{M}$ is called a unimodular matrix.

- $\mathbf{M}$ is called a totally unimodular matrix if every square submatrix (of every order) has determinant 0, 1, or -1; in particular, a totally unimodular matrix is itself unimodular.

Proposition 2.2.7 Let $\mathbf{B}$ be a basis matrix of the cut space $\mathcal{B}(D)$, F a spanning forest of D, and $\mathbf{B}_F$ the basis matrix of $\mathcal{B}(D)$ corresponding to F. Then $\mathbf{B}$ is uniquely determined by $\mathbf{B}|F$, and $\mathbf{B} = (\mathbf{B}|F)\mathbf{B}_F$.

Proof

- Since both $\mathbf{B}$ and $\mathbf{B}_F$ are basis matrices of $\mathcal{B}(D)$, there is a unique invertible matrix $\mathbf{P}$ with $\mathbf{B} = \mathbf{P}\mathbf{B}_F$.
- Hence $\mathbf{B}$ is uniquely determined by P.

- Since $\mathbf{B}|F = (\mathbf{P}\mathbf{B}_F)|F = \mathbf{P}(\mathbf{B}_F|F) = \mathbf{P}$, we have $\mathbf{B} = \mathbf{P}\mathbf{B}_F = (\mathbf{B}|F)\mathbf{B}_F$. ∎

Similarly, one can prove

Proposition 2.2.8 Let $\mathbf{C}$ be a basis matrix of the cycle space $\mathcal{C}(D)$, F a spanning forest of D, and $\mathbf{C}_F$ the basis matrix of $\mathcal{C}(D)$ corresponding to F. Then $\mathbf{C}$ is uniquely determined by $\mathbf{C}|\bar{F}$, and $\mathbf{C} = (\mathbf{C}|\bar{F})\mathbf{C}_F$.

Proposition 2.2.9 Let D be a loopless digraph, F a spanning forest of D, and $\mathbf{B}_F$, $\mathbf{C}_F$ the basis matrices of $\mathcal{B}(D)$ and $\mathcal{C}(D)$ corresponding to F. Then both $\mathbf{B}_F$ and $\mathbf{C}_F$ are unimodular.

Proof

- Let $\mathbf{P}$ be any full-order square submatrix of $\mathbf{B}_F$.
- By Corollary 2.2.3(i), $\mathbf{P}$ has order $n - \omega$.

- Let R be the edge subset of D corresponding to $\mathbf{P}$, and let the subgraph $D[R]$ induced by R be F_1; then $\mathbf{P} = \mathbf{B}_F|F_1$.
- If F_1 contains a cycle, Theorem 2.2.5(i) gives $\det \mathbf{P} = 0$.
- Now suppose F_1 is a spanning forest of D, and let $\mathbf{B}_1$ be the basis matrix of $\mathcal{B}(D)$ corresponding to F_1.
- By Proposition 2.2.7,

$$(\mathbf{B}_F|F_1)\mathbf{B}_1 = \mathbf{B}_F.$$

- Restricting both sides to F gives

$$(\mathbf{B}_F|F_1)(\mathbf{B}_1|F) = \mathbf{B}_F|F.$$

Taking determinants and noting that $\mathbf{B}_F|F$ is the identity matrix gives

$$\det(\mathbf{B}_F|F_1) \cdot \det(\mathbf{B}_1|F) = 1. \tag{9}$$

- Since both determinants in (9) are determinants of integer matrices, their values are integers. It follows that

$$\det \mathbf{P} = \det(\mathbf{B}_F | F_1) = \pm 1.$$

  This proves that $\mathbf{B}_F$ is unimodular.

- The same argument shows that $\mathbf{C}_F$ is unimodular. (Exercise!) ∎

Proposition 2.3.1 Let $\mathbf{M}$ be the incidence matrix of a loopless digraph D and $\mathbf{K}$ a submatrix obtained from $\mathbf{M}$ by deleting any one row; then $\mathbf{K}$ is unimodular.

Proof

- Let $\mathbf{P}$ be a full-order square submatrix of $\mathbf{K}$. We prove by induction that $\mathbf{P}$ is totally unimodular.

- Since $\mathbf{M}$ is the incidence matrix of D, every entry of $\mathbf{M}$ (and hence of $\mathbf{P}$) is 0, -1, or 1; thus every 1×1 submatrix of $\mathbf{P}$ is unimodular.

- Assume every $k \times k$ $(k \geq 1)$ submatrix of $\mathbf{P}$ is unimodular. We must show that the determinant of every $(k+1) \times (k+1)$ submatrix of $\mathbf{P}$ is also 0, -1, or 1.
- Let $\mathbf{Q} = (q_{ij})$ $(1 \leq i, j \leq k+1)$ be a $(k+1) \times (k+1)$ submatrix of $\mathbf{P}$.
- By the definition of the incidence matrix $\mathbf{M}$, each column of $\mathbf{Q}$ has at most two nonzero entries.
- If every column of $\mathbf{Q}$ has exactly two nonzero entries, the sum of all row vectors of $\mathbf{Q}$ is the zero vector, so $\det \mathbf{Q} = 0$.
- Now suppose $\mathbf{Q}$ has at least one column with at most one nonzero entry, say q_{ij} $(= 0, -1, \text{ or } 1)$.
- Expand $\det \mathbf{Q}$ along column j: $\det \mathbf{Q} = (-1)^{i+j} q_{ij} \cdot \det \mathbf{Q}_{ij}$, where $\det \mathbf{Q}_{ij}$ is a $k \times k$ submatrix of $\mathbf{Q}$.
- By the induction hypothesis, $\det \mathbf{Q}_{ij} = 0, -1, \text{ or } 1$, so $\det \mathbf{Q} = 0, \pm 1$.

- Since k is arbitrary, $\det \mathbf{P} = 0, \pm 1$.
- Since $\mathbf{P}$ is arbitrary, $\mathbf{K}$ is unimodular. ∎

Theorem 2.3.2 Let D be a connected loopless digraph, and let $\mathbf{B}$ and $\mathbf{C}$ be unimodular basis matrices of $\mathcal{B}(D)$ and $\mathcal{C}(D)$, respectively. Then

$$\tau(D) = \det(\mathbf{B}\mathbf{B}^T) = \det(\mathbf{C}\mathbf{C}^T),$$

where $\mathbf{B}^T$ and $\mathbf{C}^T$ denote the transposes of $\mathbf{B}$ and $\mathbf{C}$.

Proof

- Applying the Binet–Cauchy formula for the determinant of a product of two matrices gives

$$\det(\mathbf{B}\mathbf{B}^T) = \sum_{\substack{R \subseteq E(D) \\ |R| = n-1}} (\det(\mathbf{B}|R))^2. \qquad (10)$$

- On the one hand, by Theorem 2.2.5(i), the number of nonzero terms on the right of (10) equals $\tau(D)$.
- On the other hand, since D is connected and $\mathbf{B}$ is unimodular, every nonzero term on the right of (10) equals 1.
- Therefore $\tau(D) = \det(\mathbf{B}\mathbf{B}^T)$.
- Similarly, $\tau(D) = \det(\mathbf{C}\mathbf{C}^T)$. ∎

Corollary 2.3.3 Let D be a connected loopless digraph, and let $\mathbf{B}$ and $\mathbf{C}$ be unimodular basis matrices of $\mathcal{B}(D)$ and $\mathcal{C}(D)$, respectively. Then

$$\tau(D) = \pm\det \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix}.$$

Proof

- By Theorem 2.3.2,

$$\tau(D)^2 = \det(\mathbf{B}\mathbf{B}^T) \cdot \det(\mathbf{C}\mathbf{C}^T) = \det \begin{pmatrix} \mathbf{B}\mathbf{B}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{C}\mathbf{C}^T \end{pmatrix}.$$

- Since $\mathcal{B}$ and $\mathcal{C}$ are orthogonal, $\mathbf{C}\mathbf{B}^T = \mathbf{B}\mathbf{C}^T = \mathbf{O}$.
- Thus

$$\begin{aligned} \tau(D)^2 = & \det \begin{pmatrix} \mathbf{B}\mathbf{B}^T & \mathbf{B}\mathbf{C}^T \\ \mathbf{C}\mathbf{B}^T & \mathbf{C}\mathbf{C}^T \end{pmatrix} \\ = & \det \left(\begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{B}^T & \mathbf{C}^T \end{pmatrix} \right) \\ = & \det \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} \cdot \det \begin{pmatrix} \mathbf{B}^T & \mathbf{C}^T \end{pmatrix} \\ = & \left(\det \begin{pmatrix} \mathbf{B} \\ \mathbf{C} \end{pmatrix} \right)^2 \end{aligned}$$

- This proves the corollary. ∎

Corollary 2.3.4 Let D be a connected loopless digraph, T a spanning tree of D, $\mathbf{B}_T$ and $\mathbf{C}_T$ the basis matrices of $\mathcal{B}(D)$ and $\mathcal{C}(D)$ corresponding to T, and $\mathbf{K}$ the matrix obtained from the incidence matrix $\mathbf{M}$ of D by deleting any one row. Then

$$\tau(D) = \det\left(\mathbf{B}_T\mathbf{B}_T^T\right) = \det\left(\mathbf{C}_T\mathbf{C}_T^T\right) = \det\left(\mathbf{K}\mathbf{K}^T\right).$$

Proof

- By Propositions 2.2.7 and 2.2.8, both $\mathbf{B}_T$ and $\mathbf{C}_T$ are unimodular.
- By Theorem 2.3.2,

$$\tau(D) = \det(\mathbf{B}_T\mathbf{B}_T^T) = \det(\mathbf{C}_T\mathbf{C}_T^T).$$

- By Example 2.2.6, $\mathbf{K}$ is a basis matrix of $\mathcal{B}(D)$.

- By Proposition 2.3.1, $\mathbf{K}$ is unimodular.
- Again by Theorem 2.3.2, $\tau(D) = \det(\mathbf{K}\mathbf{K}^T)$. ∎

Example 2.3.5 Consider the digraph D shown in Figure 7. Its incidence matrix is

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

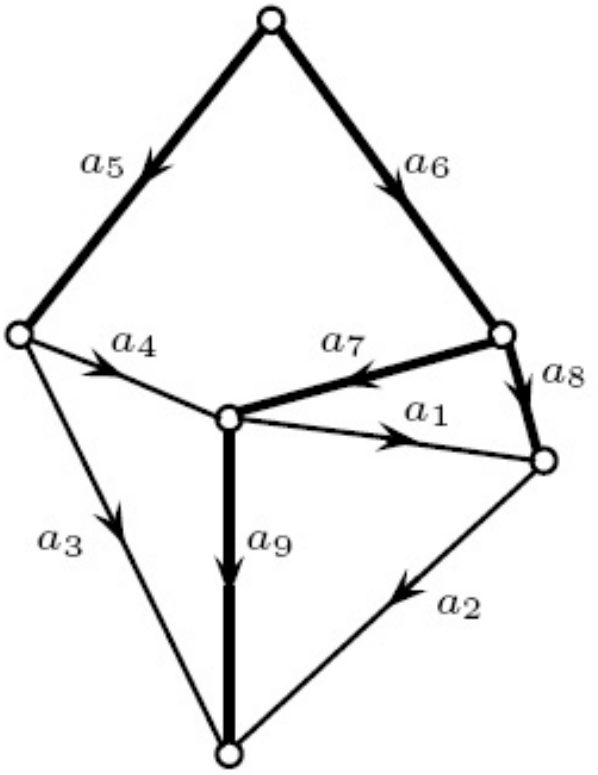


Figure 7: A digraph D and one of its spanning trees

Let $\mathbf{K}$ be obtained from $\mathbf{M}$ by deleting its last row; then

$$\det \mathbf{K}\mathbf{K}^T = \det \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix} = 66$$

Let T be the spanning tree induced by the edge subset $\{a_5, a_6, a_7, a_8, a_9\}$ (thick edges in Figure 7). The basis matrices $\mathbf{C}_T$ and $\mathbf{B}_T$ of $\mathcal{C}(D)$ and $\mathcal{B}(D)$ corresponding to T are given in (5) and (7), respectively. Then

$$\mathbf{B}_T \mathbf{B}_T^T = \begin{pmatrix} 3 & -2 & -2 & 0 & -1 \\ -2 & 3 & 2 & 0 & 1 \\ -2 & 2 & 5 & -2 & 2 \\ 0 & 0 & -2 & 3 & -1 \\ -1 & 1 & 2 & -1 & 3 \end{pmatrix},$$

$$\mathbf{C}_T\mathbf{C}_T^T = \begin{pmatrix} 3 & -2 & -1 & -1 \\ -2 & 4 & 2 & 1 \\ -1 & 2 & 5 & 3 \\ -1 & 1 & 3 & 4 \end{pmatrix}.$$

Hence $\det(\mathbf{B}_T\mathbf{B}_T^T) = 66 = \det(\mathbf{C}_T\mathbf{C}_T^T)$. ∎

Corollary 2.3.6 Let T_n be a tournament on n $(n \geq 2)$ vertices; then $\tau(T_n) = n^{n-2}$.

Proof By Corollary 2.3.4,

$$\begin{aligned} \tau(T_n) &= \det \mathbf{K}\mathbf{K}^T \\ &= \det \begin{pmatrix} n-1 & -1 & \cdots & -1 & -1 \\ -1 & n-1 & \cdots & -1 & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & \cdots & n-1 & -1 \\ -1 & -1 & \cdots & -1 & n-1 \end{pmatrix}_{n-1} \\ &= n^{n-2}. \end{aligned}$$

∎

Corollary 2.3.7 (Matrix–Tree Theorem, Kirchhoff, 1847) Let G be a connected loopless undirected graph with vertex set $\{x_1, x_2, \cdots, x_n\}$. Then $\tau(G) = \det \mathbf{K}\mathbf{K}^T$, where $\mathbf{K}$ is obtained by deleting any one row from the incidence matrix $\mathbf{M}(D)$ of any orientation D of G.

Corollary 2.3.8 (Cayley's formula, 1889) Let K_n be the complete graph on vertex set $\{x_1, x_2, \cdots, x_n\}$; then $\tau(K_n) = n^{n-2}$, $n \geq 2$.

Exercises

2.3.7 Let D be a nontrivial connected loopless digraph and a any edge of D. Prove:

$$\tau(D) = \tau(D \cdot a) + \tau(D - a).$$

Here $D \cdot a$ is the digraph obtained from D by contracting a: delete a, identify its endpoints u and v as a single vertex, and retain all other edges, with their directions preserved.

4 The Minimum Connection Problem

- Suppose a road system connecting several towns is to be built in a region. The cost of a direct road between towns x_i and x_j is c_{ij}. Design a construction plan of minimum cost.
- Such problems abound, e.g., the design of gas, water, power, and communication networks in a city. Such problems are collectively called the minimum connection problem.
- Construct a weighted simple graph $(G, \mathbf{w})$ in which town x_i is the vertex x_i of G,

$$x_i x_j \in E(G) \Leftrightarrow c_{ij} < \infty$$

(if $c_{ij} = \infty$, no road can be built between x_i and x_j), and set $\mathbf{w}(x_i x_j) = c_{ij}$.

- Thus the minimum connection problem reduces to finding a spanning tree of minimum total weight in $(G, \mathbf{w})$; such a spanning tree is called a minimum tree.

- Clearly, the minimum connection problem has a solution $\Leftrightarrow$ G is connected.
- One could enumerate all spanning trees of G, compare their total weights, and find a minimum tree; this certainly works in principle.
- But in general, when the number n of vertices is large, listing all $\tau(G)$ spanning trees is itself very hard (by Corollary 2.3.8, $\tau(G) \leq \tau(K_n) = n^{n-2}$). It is therefore necessary to seek efficient algorithms for minimum trees.

Spanning Tree Algorithms

First consider the simplest case, where $\mathbf{w}(e) = 1$ for every $e \in E(G)$; then every spanning tree of G is a minimum tree, so it suffices to construct any spanning tree. The following recursive algorithm does so:

1. Choose any $x_0 \in V(G)$ and set $V_0 = \{x_0\}$, $T_0 = x_0$, $k = 0$.

2. Once V_{k-1} and T_{k-1} are chosen, take any $e_k \in [V_{k-1}, \overline{V}_{k-1}]$; then there exist $u \in V_{k-1}$ and $x_k \in \overline{V}_{k-1}$ with $e_k = ux_k$.

Set $V_k = V_{k-1} \cup \{x_k\}$ and $T_k = T_{k-1} + e_k$.

3. If $k < n-1$, go to step 2; if $k = n-1$, stop.

This algorithm is valid because at each step it builds a subtree T_k with vertex set V_k; since G is connected, it must terminate with $V_{n-1} = V(G)$ and a spanning tree T_{n-1} of G.

Remark. Two other well-known search algorithms are breadth-first search (BFS) and depth-first search (DFS).

Prim's Algorithm

Prim generalized this algorithm to arbitrary edge weights.

Prim's algorithm:

1. Choose any $x_0 \in V(G)$, set $\ell(x_0) = 0$, $\ell(x) = \infty$ for $x \neq x_0$, $V_0 = \{x_0\}$, $T_0 = x_0$, and $k = 0$.

2. For every $x \in N_G(x_{k-1}) \cap \overline{V}_{k-1}$, if $\mathbf{w}(x_{k-1}x) < \ell(x)$, replace $\ell(x)$ by $\mathbf{w}(x_{k-1}x)$.

Choose $x_k \in \overline{V}_{k-1}$ with $\ell(x_k) = \min\{\ell(x) : x \in \overline{V}_{k-1}\}$.

Let $e_k = ux_k$ with $u \in V_{k-1}$ satisfy $\mathbf{w}(e_k) = \ell(x_k)$, and set

$$V_k = V_{k-1} \cup \{x_k\},\ T_k = T_{k-1} + e_k.$$

3. If $k < n-1$, go to step 2; if $k = n-1$, stop.

Example 2.4.1 Consider the weighted graph $(G, \mathbf{w})$ in Figure 8(a). The construction of a minimum tree T_{n-1} by Prim's algorithm is illustrated in Figure 8: (b) shows step 1, while (c)–(h) show the six iterations of step

2, each producing a new vertex x_k and a new edge e_k (in thick edges). The resulting minimum tree T_{n-1} is shown in Figure 8(h) (thick edges), with $\mathbf{w}(T_{n-1}) = 21$ (the sum of the labels $\mathbf{l}(x)$). ∎

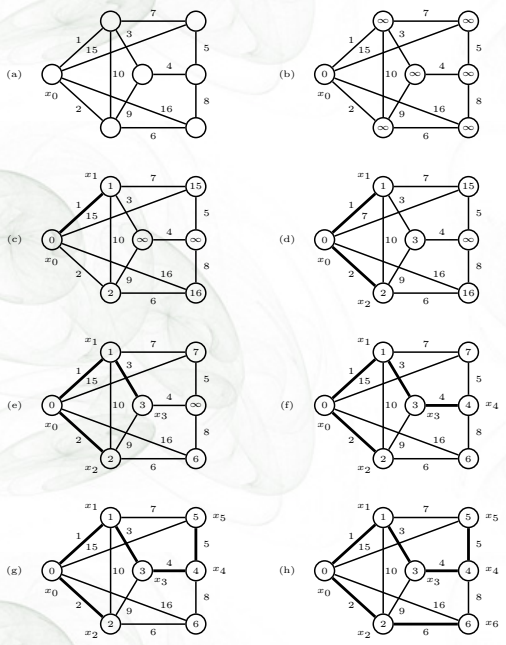


Figure 8: An application of Prim's algorithm

The following theorem guarantees that the T_{n-1} produced by Prim's algorithm is a minimum tree of G.

Theorem 2.4.2 The T_{n-1} produced by Prim's algorithm is a minimum tree of the connected weighted graph $(G, \mathbf{w})$.

Proof

- It suffices to show that T_k is a subgraph of some minimum tree of G, by induction on $k \geq 0$. The claim is clear for $k = 0$. Assume it holds for $k-1$, and consider $T_k = T_{k-1} + e_k$.
- First, we claim that T_k is a tree.
- By the induction hypothesis T_{k-1} is a tree; since $V(T_{k-1}) = V_{k-1}$ and $e_k \in [V_{k-1}, \overline{V}_{k-1}]$, T_k is connected.
- Moreover, as $e_k = ux_k$ with $u \in V_{k-1}$ and $x_k \in \overline{V}_{k-1}$, x_k has degree 1 in T_k, i.e., $T_k = T_{k-1} + e_k$ is acyclic and hence a tree.

- We now show T_k is a subgraph of some minimum tree of G. By hypothesis T_{k-1} is a subgraph of a minimum tree T^*, and $T_k = T_{k-1} + e_k$.
- If $e_k \in T^*$, then T_k is a subgraph of T^*.
- If $e_k \notin T^*$, then by Theorem 2.1.9, $T^* + e_k$ contains a unique cycle C.
- Since $e_k \in [V_{k-1}, \overline{V}_{k-1}]$, there is an edge $e'_k \in C$ with $e'_k \in [V_{k-1}, \overline{V}_{k-1}]$ and $e'_k \neq e_k$.
- Set

$$T' = (T^* + e_k) - e'_k,$$

  then $T_k \subseteq T'$.
- It remains to show that T' is also a minimum tree of G.

- Since T' is connected and $e(T') = n-1$, Theorem 2.1.2 shows that T' is another spanning tree of G, with

$$\mathbf{w}(T') = \mathbf{w}(T^*) + \mathbf{w}(e_k) - \mathbf{w}(e'_k). \tag{11}$$

- The edge $e_k \in [V_{k-1}, \overline{V}_{k-1}]$ chosen by Prim's algorithm has minimum weight, so

$$\mathbf{w}(e_k) \leqslant \mathbf{w}(e'_k). \tag{12}$$

- Combining (11) and (12) gives $\mathbf{w}(T') \leqslant \mathbf{w}(T^*)$, so T' is also a minimum tree of G. By induction, the theorem follows. ∎

Complexity Analysis

How does one judge the quality of an algorithm? Edmonds (1965) proposed the following criterion.

- A graph algorithm is called a polynomial algorithm if, on every graph, the number of operations it requires is bounded above by a polynomial (e.g., $3n^2m$) in the number n of vertices and m of edges. This bound is customarily called the computational complexity, or simply the complexity.

- For example, Prim's algorithm has complexity $O(n^2)$ (it needs $n-1$ iterations; at the k-th iteration it performs at most $n-k$ label updates, at most $n-k$ comparisons to choose $x_k \in \overline{V}_{k-1}$, k comparisons to choose $e_k \in [V_{k-1}, \overline{V}_{k-1}]$, and two assignments for V_k and T_k, so the complexity is at most $(n-1)(2n) = O(n^2)$).

- An algorithm whose complexity is exponential (e.g., 2^n) or $n!$ is inefficient when n is large.

- A polynomial algorithm is also called an efficient algorithm, or a good algorithm; Prim's algorithm is efficient.

- Standard notation for orders of complexity (each inequality holds

for some positive constant(s) and all sufficiently large n):

$$f = O(g): \quad f(n) \leq c\,g(n);$$

$$f = \Omega(g): \quad f(n) \geq c\,g(n);$$

$$f = \Theta(g): \quad c_1 g(n) \leq f(n) \leq c_2 g(n).$$

Exercises

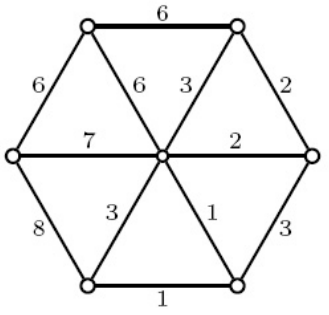
2.4.3 The **Kruskal algorithm** for a minimum tree in a connected weighted simple graph $(G, \mathbf{w})$ is as follows:

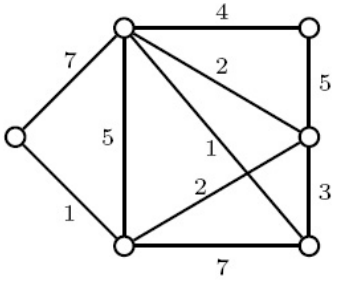
1. Choose $e_1 \in E(G)$ with $\mathbf{w}(e_1)$ as small as possible;
2. once $e_1, e_2, \cdots, e_i$ are chosen, pick $e_{i+1} \in E(G) \setminus \{e_1, \cdots, e_i\}$ so that $G[\{e_1, \cdots, e_i, e_{i+1}\}]$ is acyclic and $\mathbf{w}(e_{i+1})$ is as small as possible;

3. if $i < n-1$, go to step 2; if $i = n-1$, stop.

(a) Prove that the subgraph of G built by Kruskal's algorithm is a minimum tree of G.

(b) Use Kruskal's algorithm to find a minimum tree in the following graph.





5 The Shortest Path Problem

- Suppose a railway system connects several cities, with a fixed city x_0 (e.g., the capital or a provincial capital). Find the shortest route from x_0 to every other city; this is called the shortest path problem.
- Represent the railway system by a simple weighted undirected graph $(G, \mathbf{w})$ with a positive weight function $\mathbf{w} \in \mathcal{E}(G)$; the value $\mathbf{w}(e)$ on an edge $e = xy$ is the railway distance between cities x and y. The problem then becomes finding the shortest path in $(G, \mathbf{w})$ from x_0 to every other vertex of G.
- More generally, let $(D, \mathbf{w})$ be a simple digraph with positive weights and x_0 a fixed vertex of D; find the shortest directed path from x_0 to every vertex of D.
- Clearly, a necessary condition for the shortest path problem to be solvable is that D has a spanning out-tree rooted at x_0; conversely, if

such a spanning out-tree exists, the problem always has a solution. Henceforth assume D has a spanning out-tree rooted at x_0.

- Note: a solution of the shortest path problem need not be a minimum spanning tree. Even for undirected graphs Prim's algorithm cannot solve it. For example, the thick edges in Figure 8(h) form the spanning tree T given by Prim's algorithm, which is a minimum tree. Along the edges of T, the path P from x_0 to x_5 has length 13. But the path Q from x_0 through x_1 and along the edge x_1x_5 to x_5 has length 8. Thus P is not the shortest x_0x_5 path in G, so another method is needed for the shortest path problem.

The following algorithm was discovered independently by Moore (1957), Dijkstra (1959), Dantzig (1960), and Whiting & Hillier (1960). Based on Theorem 1.4.2, it modifies step 2 of Prim's algorithm. Throughout, we build a function $\ell \in \mathcal{V}(D)$, a sequence S_k of vertex subsets containing x_0, and a sequence T_k of out-trees rooted at x_0 with vertex set S_k, for

$k = 0, 1, \cdots, v-1$. Initially, set

$$\ell(x) = \begin{cases} 0, & \text{if } x = x_0; \\ \infty, & \text{if } x \neq x_0, \end{cases} \qquad S_0 = \{x_0\}.$$

As the algorithm runs, the values $\ell(x)$ are repeatedly updated to recursively construct S_k and the directed paths P_k starting at x_0.

Moore–Dijkstra algorithm

1. Set $\ell(x_0) = 0$, $\ell(x) = \infty$ for $x \neq x_0$, $S_0 = \{x_0\}$, $T_0 = x_0$, and $k = 0$.

2. For every $x \in N_D^+(x_k) \cap \overline{S}_k$, set

$$\ell(x) := \min\{\ell(x), \ell(x_k) + \mathbf{w}(x_k, x)\}.$$

Choose $x_{k+1} \in N_D^+(S_k) \cap \overline{S}_k$ such that

$$\ell(x_{k+1}) = \min_{x \in \overline{S}_k} \{\ell(x)\},$$

and choose $x_j \in S_k$ $(j \leq k)$ with $(x_j, x_{k+1}) \in E(D)$ that attains this minimum, i.e. $\ell(x_{k+1}) = \ell(x_j) + \mathbf{w}(x_j, x_{k+1})$.

Set $S_{k+1} = S_k \cup \{x_{k+1}\}$ and $T_{k+1} = T_k + (x_j, x_{k+1})$.

3. If $k = n-1$, stop; if $k < n-1$, replace k by $k+1$ and return to step 2.

We illustrate this algorithm with an example.

Example 2.5.1 Consider the weighted graph $(D, \mathbf{w})$ in Figure 9(a); the Moore–Dijkstra algorithm runs as in Figure 9.

As Figure 9 shows, running the Moore–Dijkstra algorithm amounts to growing a spanning out-tree of D rooted at x_0. We will prove that the label $\ell(x_i)$ of each vertex x_i is the weighted distance $d_D(x_0, x_i)$ from x_0 to x_i in D.

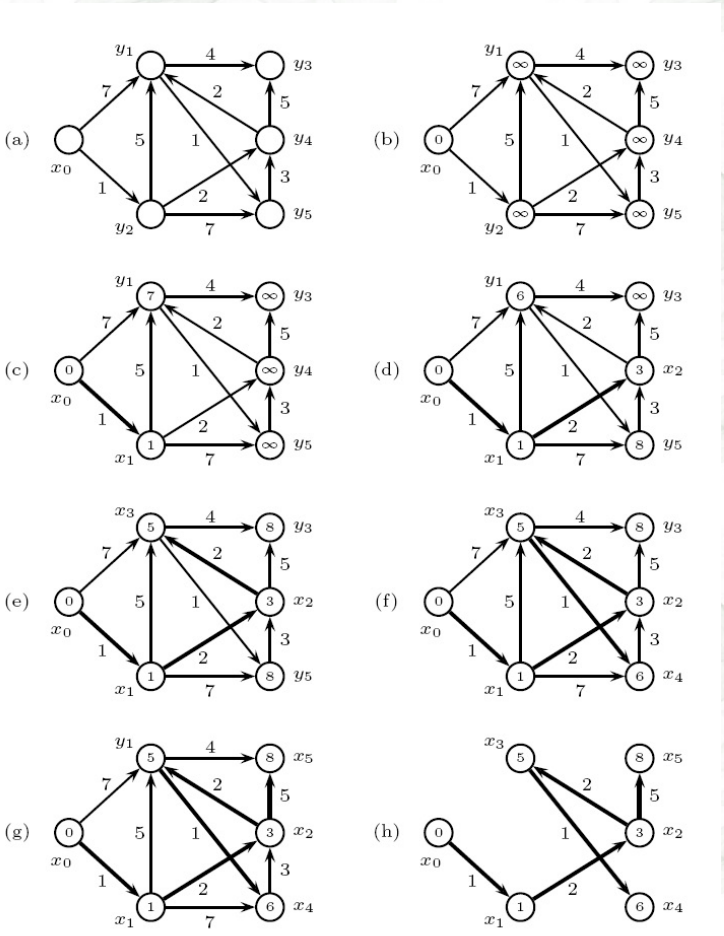


Figure 9: An application of the shortest path algorithm

The subgraph T built when the Moore–Dijkstra algorithm terminates has n vertices and $n-1$ edges, is acyclic and connected, and hence is a spanning tree of D. Since $d_T^-(x_0) = 0$ and $d_T^-(x_i) = 1$ for $i \neq 0$, T is an out-tree rooted at x_0. The following theorem guarantees that, for every $x_i \in V(D)$, the (x_0, x_i) path in T is the shortest (x_0, x_i) path in D.

Theorem 2.5.2 Let T be the spanning out-tree rooted at x_0 produced by the Moore–Dijkstra algorithm in $(D, \mathbf{w})$. For every $x \in V(D)$, the (x_0, x) path in T is the shortest (x_0, x) path in D.

Proof

- It suffices to show $d_D(x_0, x) = \ell(x)$ for every $x \in V(D)$.
- On the one hand, by the construction of T, T contains an (x_0, x) path P of length $\ell(x)$; hence $d_D(x_0, x) \leq \ell(x)$.
- On the other hand, we show $\ell(x) = d_D(x_0, x)$ for the vertex x chosen at each iteration. Suppose x is chosen at the k-th iteration and is the first vertex with $\ell(x) > d_D(x_0, x)$.

- Let $Q = (x_0, x_1, \cdots, x_k = x)$ be a shortest (x_0, x) path in D, and let x_i be the first vertex common to Q and $\overline{S}_k$.
- We claim that $\ell(x_i) = d_D(x_0, x_i)$.

  Indeed, since x is the first vertex with $\ell(x) > d_D(x_0, x)$, every earlier selected vertex—in particular $x_{i-1} \in S_k$ (with $i \geq 1$)—satisfies $\ell = d_D$. Hence

$$\ell(x_i) \leq \ell(x_{i-1}) + \mathbf{w}(x_{i-1}, x_i) = d_D(x_0, x_{i-1}) + \mathbf{w}(x_{i-1}, x_i)$$

$$= d_D(x_0, x_i).$$

  The last equality uses the fact that a subpath of a shortest path is itself a shortest path.
- Hence

$$\ell(x) > d_D(x_0, x) \geq d_D(x_0, x_i) \geq \ell(x_i),$$

  contradicting the choice of x. ∎

The Moore–Dijkstra algorithm is a good algorithm with complexity $O(n^2)$.

This algorithm also yields a polynomial-time algorithm for the diameter of a graph:

- The Moore–Dijkstra algorithm finds the shortest path and its distance $\ell(x)$ from x_0 to every other vertex x. Since x_0 is arbitrary, running the algorithm n times, once with each vertex as x_0, yields the diameter $d(D)$ of D.
- For example, on the weighted digraph $(D, \mathbf{w})$ in Figure 9, six runs produce the spanning out-trees rooted at x_i $(i = 0, 1, 2, 3, 4, 5)$, with the one rooted at x_0 shown in Figure 9(h). For each such out-tree T_i $(0 \leq i, j \leq 5)$, the label on x_j is the distance $d_D(x_i, x_j)$. The 6×6

matrix whose (i,j)-entry is $d_D(x_i,x_j)$,

$$\begin{pmatrix} 0 & 1 & 3 & 5 & 6 & 8 \\ 12 & 0 & 2 & 4 & 7 & 7 \\ 10 & 11 & 0 & 2 & 3 & 5 \\ 8 & 9 & 4 & 0 & 1 & 4 \\ 13 & 14 & 3 & 5 & 0 & 8 \\ 17 & 18 & 13 & 9 & 10 & 0 \end{pmatrix}$$

is called the **distance matrix**. Comparing its entries gives the diameter of D,

$$d(D) = \max\{d_D(x_i,x_j) : 0 \leqslant i,j \leqslant 5\} = 18.$$

- Since computing $d(D)$ requires n runs of the Moore–Dijkstra algorithm followed by comparing the n^2 entries of the distance matrix, the complexity of finding the diameter is $O(n \cdot n^2 + n^2) = O(n^3)$.
- The Moore–Dijkstra algorithm requires nonnegative edge weights and is not guaranteed to work when negative weights occur, even

if the graph has no negative cycle; in that situation one should use the Bellman–Ford algorithm (provided that no negative cycle is reachable). If walks are allowed to repeat edges, a reachable negative cycle makes the shortest distance unbounded below ($-\infty$); when paths are required to be simple, deciding shortest paths in the presence of negative cycles is NP-hard.

- Remark: Negating every edge weight transforms a longest-path problem into a shortest-path problem. If walks may repeat edges, a reachable positive cycle makes the longest distance unbounded above ($+\infty$); when paths are required to be simple, the longest path problem is NP-hard even when there is no positive cycle).

Exercises

2.5.5 A company has branches in six cities $x_1, x_2, \cdots, x_6$. The direct airfare from x_i to x_j is the (i, j)-entry of the matrix below (∞ means no direct flight). Construct a table of the cheapest route between every

pair of cities for the company.

$$\begin{pmatrix} 0 & 50 & \infty & 40 & 25 & 10 \\ 50 & 0 & 15 & 20 & \infty & 25 \\ \infty & 15 & 0 & 10 & 20 & \infty \\ 40 & 20 & 10 & 0 & 10 & 25 \\ 25 & \infty & 20 & 10 & 0 & 55 \\ 10 & 25 & \infty & 25 & 55 & 0 \end{pmatrix}$$

6 Network Equations

This section is left for self-study.

7 The Berge–Fulkerson and Berge Conjectures

A matching of a graph $G = (V,E)$ is an edge subset in which no two edges are adjacent. Clearly, in an edge coloring of G, the set of edges of any one color is a matching of G.

A matching of G with exactly $\frac{|V(G)|}{2}$ edges is called a perfect matching of G.

Conjecture 2.7.1 (Berge–Fulkerson conjecture) Every bridgeless cubic graph admits six perfect matchings such that every edge is covered exactly twice.

Conjecture 2.7.2 (Berge conjecture) Every bridgeless cubic graph admits five perfect matchings such that every edge belongs to at least one of them.

Exercises:

(1) Both conjectures hold for cubic graphs that are 3-edge-colorable.

(2) Both conjectures hold for cubic Hamiltonian graphs (graphs containing a Hamilton cycle). A Hamilton cycle of G is a cycle containing every vertex of G.

(3) Do the above conjectures hold for cubic hypohamiltonian graphs?

A graph G is hypohamiltonian if G itself is not Hamiltonian, but deleting any single vertex from G yields a Hamiltonian graph.

Thank you!