运筹与管理科学丛书 18

网络最优化

谢 政 著

科 学 出 版 社

运筹与管理科学丛书 18

网络最优化

谢 政 著

科学出版社
北 京

内 容 简 介

本书全面系统地介绍了网络最优化中的基本问题和基本算法以及计算复杂性的基本内容和近似算法. 取材恰当, 叙述清晰, 论证严谨, 深入浅出.

全书共十二章, 分为两部分: 第一部分包括前十章, 主要介绍最小树, 最小树形图, 最短路, 最大流, 最小费用流, 最大匹配, 最大权匹配和中国邮递员问题等基本问题的各种多项式算法, 以及线性规划、整数线性规划的基本理论; 第二部分包括后两章, 讨论计算复杂性中的基本概念, NP 完全理论及重要的 NP 完全问题, 还介绍了装箱问题, 平行机排序问题, 旅行商问题, 背包问题等 NP 难问题的近似算法.

本书可作为运筹学专业研究生教材, 也可供应用数学、系统科学、管理科学、计算机科学和军事运筹学等有关专业的教师、研究生和大学高年级学生参考.

图书在版编目(CIP)数据

网络最优化/谢政著. —北京: 科学出版社, 2014.6

(运筹与管理科学丛书; 18)

ISBN 978-7-03-040952-2

I. ①网… II. ①谢… III. ①计算机网络–最佳化 IV. ①TP393

中国版本图书馆 CIP 数据核字 (2014) 第 121943 号

责任编辑: 陈玉琢 / 责任校对: 韩 杨

责任印制: 赵德静 / 封面设计: 王 浩

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

http://www.sciencep.com

北京源海印刷有限责任公司 印刷

科学出版社发行 各地新华书店经销

*

2014 年 6 月第 一 版 开本: 720 × 1000 1/16

2014 年 6 月第一次印刷 印张: 20 1/2

字数: 395 000

定价: 98.00 元

《运筹与管理科学丛书》序

运筹学是运用数学方法来刻画、分析以及求解决策问题的科学. 运筹学的例子在我国古已有之, 春秋战国时期著名军事家孙膑为田忌赛马所设计的排序就是一个很好的代表. 运筹学的重要性同样在很早就被人们所认识, 汉高祖刘邦在称赞张良时就说道:“运筹帷幄之中, 决胜千里之外.”

运筹学作为一门学科兴起于第二次世界大战期间, 源于对军事行动的研究. 运筹学的英文名字 Operational Research 诞生于 1937 年. 运筹学发展迅速, 目前已有众多的分支, 如线性规划、非线性规划、整数规划、网络规划、图论、组合优化、非光滑优化、锥优化、多目标规划、动态规划、随机规划、决策分析、排队论、对策论、物流、风险管理等.

我国的运筹学研究始于20世纪50年代, 经过半个世纪的发展, 运筹学研究队伍已具相当大的规模. 运筹学的理论和方法在国防、经济、金融、工程、管理等许多重要领域有着广泛应用, 运筹学成果的应用也常常能带来巨大的经济和社会效益. 由于在我国经济快速增长的过程中涌现出了大量迫切需要解决的运筹学问题, 因而进一步提高我国运筹学的研究水平、促进运筹学成果的应用和转化、加快运筹学领域优秀青年人才的培养是我们当今面临的十分重要、光荣, 同时也是十分艰巨的任务. 我相信,《运筹与管理科学丛书》能在这些方面有所作为.

《运筹与管理科学丛书》可作为运筹学、管理科学、应用数学、系统科学、计算机科学等有关专业的高校师生、科研人员、工程技术人员的参考书, 同时也可作为相关专业的高年级本科生和研究生的教材或教学参考书. 希望该丛书能越办越好, 为我国运筹学和管理科学的发展做出贡献.

袁亚湘

2007 年 9 月

前　　言

网络是指由若干对象及其相互关系构成的一个系统. 从数学上讲, 网络是由若干个点及其点之间的带箭头连线表示的一个有向图, 并且常常都赋以权. 现代社会在很大程度上是一个由信息网络、通信网络、运输网络、能源和物资分配网络构成的巨大的复杂系统. 网络最优化则是综合运用图论、线性规划和算法理论来解决图和网络上的最优化问题, 因此它能为人们更有效地设计、控制和管理复杂的社会网络系统提供一套科学的方法.

由于其视角的独特性、模型的直观性、内容的趣味性和方法的技巧性, 以及计算机科学和信息科学的有力推动, 网络最优化业已成长为运筹学中引人入胜、富有挑战、充满活力的一个分支, 它的理论和方法广泛地渗透于运筹学、信息论、控制论、管理科学和计算机科学等领域, 并在工程技术、经济、军事等诸多方面都有着极为重要的应用.

随着大规模网络最优化问题的产生, 寻找有效最优算法便成为网络最优化的重点. 由于许多问题都难以找到有效最优算法, 计算复杂性和近似算法的研究已成为网络最优化的热点. 本书的编写正是兼顾以上重点和热点, 力求突出三大特色:

一是化繁为简, 激发学习热情. 从实际应用问题引入网络最优化模型, 直观、自然地建立基本概念, 用通俗而又严谨的语言来描述基本理论和基本方法.

二是夯实基础, 拓宽创新视野. 将传统知识与现代技术进行有机融合, 以经典内容为主, 注重基础性和完备性, 介绍学科的前沿研究成果.

三是注重应用, 培养研究能力. 从图论和线性规划两个不同的角度讨论网络最优化问题, 介绍算法设计技巧和复杂性分析方法, 强调模型、算法与计算复杂性的紧密结合, 使读者能清楚地了解各种网络最优化问题之间的关系以及研究方法的本质.

本书根据作者 2003 年在国防科技大学出版社出版的《网络算法与复杂性理论》(第 2 版) 修订而成, 它全面地介绍了网络最优化的基本问题和基本方法, 计算复杂性理论的基本概念和一些常见的 NP 完全问题及其近似算法. 全书共十二章, 分为两部分. 第一部分包括前十章, 主要介绍网络最优化的概念、模型和算法, 并始终强调算法复杂性分析. 第二部分包括后两章, 介绍 NP 完全理论和近似算法. 每章最后都配置一定数量的习题, 其中一些难度较大的作为对正文内容的补充.

阅读本书只需具备线性规划的基本知识.

本书的出版得到了科学出版社的支持, 以及陈玉琢编辑的帮助, 在此表示由衷的感谢.

谢　政
2014 年 6 月 2 日
于国防科技大学

目　　录

第1章 图 与 算 法

从一定意义上讲, 现代社会是一个由信息网络、通信网络、运输网络、能源和物资分配网络构成的巨大的复杂系统. 网络最优化能为人们有效地设计、控制和管理这个网络系统提供一套科学方法.

本章主要介绍图、网络、算法及算法复杂性的一些基本概念, 给出八个典型的网络最优化问题及其相关的实际例子. 这些内容是全书的基础.

1.1 图的基本概念

图 (graph)G 是指由非空有限集合 $V(G)$ 和 $V(G)$ 中某些元素的无序对的集合 $E(G)$ 构成的二元组 $(V(G),E(G))$.$V(G)$ 称为 G 的顶点集 (vertex set), 其中的元素称为 G 的顶点 (vertex). $E(G)$ 称为 G 的边集 (edge set), 其中的元素称为 G 的边 (edge). 在不混淆的情况下, 记 $V=V(G),E=E(G),G=(V,E)$. 如果 $V=\{v_1,v_2,\cdots,v_n\}$, 则 E 中元素 e 与 V 中某两个元素构成的无序对 $\{v_i,v_j\}$ 相对应, 记 $e=v_iv_j$ 或 $e=v_jv_i$.

图可以用图形来表示, 用小圆圈表示顶点, 用小圆圈之间的连线表示边. 例如, 图 1.1 就表示图 $G=(V,E):V=\{v_1,v_2,v_3,v_4,v_5\}$, $E=\{e_1,e_2,e_3,e_4,e_5,e_6,e_7\}$, 其中 $e_1=v_1v_2,e_2=v_1v_2,e_3=v_2v_3,e_4=v_3v_4,e_5=v_4v_5,e_6=v_5v_2,e_7=v_4v_4$.

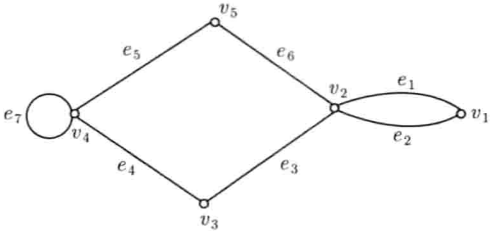

图 1.1 图的图形表示

设 $G=(V,E)$ 是一个图, 若 $e=v_iv_j\in E$, 则称顶点 v_i 和 v_j 是相邻的 (adjacent), 并称 v_i,v_j 为边 e 的端点 (end), 也称 e 与 v_i,v_j 关联 (incident). 若 $e_1,e_2\in E$, 且 e_1 和 e_2 有公共的端点, 则称 e_1 与 e_2 是相邻的. 我们把 G 中所有与顶点 v 相邻的顶点的集合称为 v 的邻域 (neighbour), 记为 $N_G(v)$ 或简记为 $N(v)$.

两个端点重合的边称为环 (loop). 如果有两条边的端点是同一对顶点, 则称这

两条边为重边 (multiple edge). 既没有环也没有重边的图称为简单图 (simple graph). 没有环的图称为无环图 (unlooped graph). 边集为空集的图称为空图 (empty graph). 至少有一条边的图称为非空图 (nonempty graph). 一个图的顶点数称为该图的阶 (order).

图 (G) 中顶点 v 的度 (degree) 定义为和 v 关联的边的数目 (与 v 关联的每个环算作两条边), 记为 $d_G(v)$. 称 $d_G(v)$ 是偶数的顶点 v 为偶点 (even vertex), 称 $d_G(v)$ 是奇数的顶点 v 为奇点 (odd vertex). $d_G(v) = 0$ 的顶点 v 称为孤立点 (isolated vertex). $d(v) = 1$ 的顶点 v 称为悬挂点 (pendant vertex). 容易得到下面的定理:

定理 1.1 设 $G = (V, E)$ 是一个图, 则

$$\sum_{v\in V} d_G(v) = 2|E|. \qquad \square$$

如果图 G 的某些顶点和边可以排成非空的有限序列 $W = v_1e_1v_2\cdots v_ke_kv_{k+1}$, 这里 $v_i \in V(G)(1 \leqslant i \leqslant k+1), e_j \in E(G)(1 \leqslant j \leqslant k)$, 并且 $e_i = v_iv_{i+1}(1 \leqslant i \leqslant k)$, 则称 W 为 G 的一条途径 (walk). v_1 称为 W 的起点 (origin), v_{k+1} 称为 W 的终点 (terminus), $v_i(2 \leqslant i \leqslant k)$ 称为 W 的内部顶点 (internal vertex), 也称 W 为 G 的 (v_1, v_{k+1}) 途径, k 称为 W 的长 (length). 有时把途径 $v_1e_1v_2\cdots v_ke_kv_{k+1}$ 简单地记为 $v_1v_2\cdots v_kv_{k+1}$. 值得注意的是, 由于以顶点 v_i 和 v_{i+1} 为端点的边可能不止一条, 因此 $v_1v_2\cdots v_kv_{k+1}$ 可能同时表示若干条不同的途径; 但是简单图中的途径 $v_1e_1v_2\cdots v_ke_kv_{k+1}$ 由 $v_1v_2\cdots v_kv_{k+1}$ 完全确定.

途径 $W = v_1e_1v_2\cdots v_ke_kv_{k+1}$ 的节 (section) 是指由 W 中相继项构成的子序列 $v_ie_iv_{i+1}\cdots v_{j-1}e_{j-1}v_j$, 它也是一条途径, 这一子序列又称为 W 的 (v_i, v_j) 节. 途径 $W = v_1e_1v_2\cdots v_ke_kv_{k+1}$ 的逆向途径 (inversion walk) 是指途径 $v_{k+1}e_kv_k\cdots v_2e_1v_1$, 记为 W^{-1}. 设 $W_1 = v_1e_1v_2\cdots v_ke_kv_{k+1}$ 和 $W_2 = v_{k+1}e_{k+1}v_{k+2}\cdots v_le_lv_{l+1}$ 是图 G 的两条途径, 则称途径

$$W = v_1e_1v_2\cdots v_ke_kv_{k+1}e_{k+1}v_{k+2}\cdots v_le_lv_{l+1}$$

为 W_1 与 W_2 的衔接 (concatenation), 记作 $W = W_1W_2$, 也称 W 可以表示为 W_1 和 W_2 的并 (union).

如果图 G 中途径 W 上的边互不相同, 则称 W 为 G 的迹 (trail). 如果图 G 中的途径 W 上的顶点互不相同, 则称 W 为 G 的链 (chain). 易知, 图 G 中的链必定是 G 的迹, 但 G 中的迹不一定是 G 的链.

如果途径的长至少为 1, 且起点和终点重合, 则称该途径为闭途径 (closed walk). 起点和终点重合且长至少为 1 的迹称为闭迹 (closed trail). 起点、内部顶点互不相同的闭迹称为圈 (cycle). 长为偶数的圈称为偶圈 (even cycle), 长为奇数的圈称为奇圈 (odd cycle).

如果图 G 和图 H 满足 $V(H) \subseteq V(G), E(H) \subseteq E(G)$, 则称 H 是 G 的子图 (subgraph), 记为 $H \subseteq G$. 特别地, 若 $V(H) = V(G), E(H) = E(G)$, 则记 $H = G$. 如果 $H \subseteq G$, 且 $V(H) = V(G)$, 则称 H 为 G 的支撑子图 (spanning subgraph). 图 G 中的链和圈都可以看做是 G 的子图.

设 $V' \subseteq V(G), V' \neq \varnothing$, 令 $E' = \{e \in E(G) | e$ 的两个端点均属于 $V'\}$, 则称 G 的子图 $G' = (V', E')$ 为由 V' 导出的子图 (induced subgraph), 记为 $G' = G[V']$.

设 $E' \subseteq E(G), E' \neq \varnothing$, 令 $V' = \{v \in V(G) | v$ 是 E' 中某条边的端点 $\}$, 则称 G 的子图 $G' = (V', E')$ 为由 E' 导出的子图 (edge-induced subgraph), 记为 $G' = G[E']$.

如果对于图 G 的任意两个顶点 v_i 和 v_j, G 中都存在 (v_i, v_j) 链, 则称 G 是连通图 (connected graph). 不连通的图称为非连通图 (disconnected graph).

如果 $H \subseteq G$, 且 H 是连通图, 则称 H 是 G 的连通子图 (connected subgraph). 图 H 称为图 G 的极大连通子图 (maximal connected subgraph) 是指 H 为 G 的连通子图, 并且 G 中不存在连通子图 H' 使得 $H \subseteq H', H' \neq H$. 图 G 的极大连通子图又称为 G 的连通分支 (connected component). 由此可知, 连通图恰有一个连通分支, 而非连通图则有两个或两个以上的连通分支.

下面介绍图的几种运算.

若 $E_1 \subseteq E(G)$, 从图 G 中删去 E_1 的所有边得到的图记为 $G - E_1$, 通常把 $G - \{e\}$ 简记为 $G - e$. 若 $H \subseteq G, E_2 \subseteq E(G) \setminus E(H)$ 且 E_2 中每条边的端点都属于 $V(H)$, 在图 H 中添加 E_2 的所有边得到的图记为 $H + E_2$. 常常简单地用 $H + e$ 表示 $H + \{e\}$.

若 $V_1 \subset V(G)$, 则把从图 G 中删去 V_1 的所有顶点以及与 V_1 中顶点关联的边得到的图记为 $G - V_1$.$G - \{v\}$ 常常简记为 $G - v$. 容易知道 $G[V'] = G - (V(G) \setminus V')$.

若 $H_1 \subseteq G, H_2 \subseteq G$, 则把顶点集为 $V(H_1) \cup V(H_2)$、边集为 $E(H_1) \cup E(H_2)$ 的图称为 H_1 与 H_2 的并, 记为 $H_1 \cup H_2$. 若 $E(H_1) \cap E(H_2) = \varnothing$, 则称 $H_1 \cup H_2$ 为 H_1 与 H_2 的和, 记为 $H_1 + H_2$.

设 e 是图 $G = (V, E)$ 的一条边, 且 $e = v_i v_j$, 我们从 G 中删去 e, 并把顶点 v_i 和 v_j 重合为一个顶点 y, 得到一个新的顶点集 $V' = (V \setminus \{v_i, v_j\}) \cup \{y\}$, 而把 G 中所有与 v_i 或与 v_j 关联的边都改为与 y 关联, G 中既不与 v_i 也不与 v_j 关联的边不变, 得到一个新的边集 E', 这样产生的新图 $G' = (V'E')$ 称为 G 关于边 e 的收缩 (condensation). 顶点 y 称为收缩 e 产生的人造顶点 (artificial vertex).

设 $G = (V, E)$ 是一个图, S 为 V 的一个非空真子集, $\overline{S} = V \setminus S$, 记

$$[S, \overline{S}] = \{v_i v_j \in E | v_i \in S, v_j \in \overline{S}\}.$$

如果 $[S, \overline{S}] \neq \varnothing$, 则称 $[S, \overline{S}]$ 是 G 的一个边割 (edge-cut). 若 E' 是 G 的一个边割,

但 E' 的任何真子集都不是 G 的边割, 则称 E' 为 G 的极小边割 (minimaledge-cut). G 的极小边割又称为 G 的补圈 (cocycle).

一个图的边割与圈有如下的关系:

定理 1.2 设 C 和 $[S,\overline{S}]$ 分别是图 G 的圈和边割, 则

$$|E(C)\cap[S,\overline{S}]|\equiv 0(\text{mod}2).$$

证明 当 $E(C)\cap[S,\overline{S}]=\varnothing$ 时, $|E(C)\cap[S,\overline{S}]|=0$. 当 $E(C)\cap[S,\overline{S}]\neq\varnothing$ 时, 设 $C=v_1e_1v_2e_2v_3\cdots v_pe_pv_1, v_1\in S$, 且 $E(C)\cap[S,\overline{S}]=\{e_{i_1},e_{i_2},\cdots,e_{i_k}\}$, 其中 $1\leqslant i_1<i_2<\cdots i_k\leqslant p$. 记 $\widehat{V}=\{v_{i_1},v_{i_2},\cdots,v_{i_k}\}$, 由 $v_1\in S$ 知, $v_{i_1}\in S$, $v_{i_k+1}\in S$. 根据 $[S,\overline{S}]$ 的定义有

$$\{v_{i_j}\in\widehat{V}|j\text{ 为奇数 }\}\subseteq S,\quad \{v_{i_j}\in\widehat{V}|j\text{ 为偶数 }\}\subseteq\overline{S}.$$

因 $v_{i_k+1}\in S$, 故 $v_{i_k}\in\overline{S}$, 即知 k 为偶数. 于是 $|E(C)\cap[S,\overline{S}]|$ 为偶数. □

1.2 有向图的基本概念

有向图 (digraph)D 是指由一个非空的有限集合 $V(D)$ 和 $V(D)$ 中某些元素的有序对的集合 $A(D)$ 构成的二元组 $(V(D),A(D)).V(D)$ 称为 D 的顶点集, 其中的元素称为 D 的顶点. $A(D)$ 称为 D 的弧集 (arc set), 其中的元素称为 D 的弧 (arc). 在不混淆时, 记 $V=V(D),A=A(D),D=(V,A)$.

如果 $V(D)=\{v_1,v_2,\cdots,v_n\}$ 那么 $A(D)$ 中元素 a 与 $V(D)$ 中某两个元素构成的有序对 $\{v_i,v_j\}$ 相对应, 记 $a=(v_i,v_j)$, 其中 v_i 称为 a 的尾 (tail), v_j 称为 a 的头 (head); a 称为 v_i 的出弧 (out-arc), 也称为 v_j 的入弧 (in-arc); 称 v_j 为 v_i 的出邻点, 称 v_i 为 v_j 的入邻点. 以 $N_D^+(v)$ 表示顶点 v 的所有出邻点的集合, 称为 v 的出邻域 (out-neighbour); 以 $N_D^-(v)$ 表示顶点 v 的所有入邻点的集合, 称为 v 的入邻域 (in-neighbour). 顶点 v 在 D 中出弧的数目记为 $d_D^+(v)$, 称之为 v 的出度 (out-degree); 顶点 v 在 D 中入弧的数目记为 $d_D^-(v)$, 称之为 v 的入度 (in-degree). 显然有

定理 1.3 设 $D=(V,A)$ 是有向图, 则

$$\sum_{v\in V}d_D^+(v)=\sum_{v\in V}d_D^-(v)=|A|.\qquad\square$$

有向图也可以用一个图形来表示. 用小圆圈表示顶点, 用小圆圈之间带有箭头的连线表示弧, 连线的箭头由弧的尾指向弧的头. 例如, 图 1.2 就表示一个有向图 $D=(V,A):V=\{v_1,v_2,\cdots,v_6\},A=\{a_1,a_2,\cdots,a_{11}\}$, 其中 $a_1=(v_1,v_1),a_2=$

$(v_1, v_2), a_3 = (v_3, v_2), a_4 = (v_3, v_4), a_5 = (v_4, v_3), a_6 = (v_2, v_4), a_7 = (v_5, v_4), a_8 = (v_4, v_6), a_9 = (v_6, v_5), a_{10} = (v_6, v_2), a_{11} = (v_6, v_2)$.

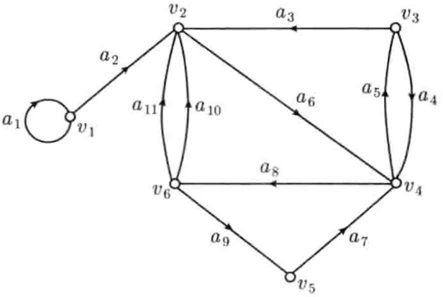

图 1.2 有向图的图形表示

尾和头重合的弧称为环. 若两条弧有相同的头和相同的尾, 则称这两条弧为重弧 (multiple-arc). 既没有环也没有重弧的有向图称为简单有向图 (simple digraph).

把有向图 D 中每条弧上的箭头去掉, 即把 D 的每条弧 (v_i, v_j) 用边 v_iv_j 代替, 得到的图称为 D 的基础图 (underlying graph). 反之, 给定一个图 G, 如果把图 G 的每条边都规定一个方向, 即把顶点的无序对按规定的方向改为有序对, 得到的有向图称为 G 的定向图 (oriented graph).

有向图中的许多概念可以通过它的基础图去描述. 例如, 设 G 是有向图 D 的基础图, 如果顶点 v_i 和 v_j 在 G 中相邻, 则称 v_i 和 v_j 在 D 中相邻; 若 G 是连通的, 则称 D 是连通的; D 中的途径是指 D 中这样的一些顶点和弧构成的序列: 使这些顶点与这些弧相对应的边构成 G 的途径. 类似地, 定义有向图的迹、链、闭途径、闭迹和圈.

仿照图中子图、支撑子图和导出子图的定义, 同样可以给出有向图的子图、支撑子图和导出子图的定义. 另外, 1.1 节介绍的图的一些运算也可以照搬到有向图上来. 这些都留给读者去完成.

由于有向图中,“方向”是很重要的, 因此我们要给出有向图中与方向有关的一些概念.

有向图 D 中的有向途径 (directed walk) 是指 D 中某些顶点和弧组成的非空有限序列 $W = v_1a_1v_2\cdots v_ka_kv_{k+1}$, 其中 $v_i \in V(D)(1 \leqslant i \leqslant k+1), a_j \in A(D)(1 \leqslant j \leqslant k)$, 且 $a_i = (v_i, v_{i+1})(1 \leqslant i \leqslant k)$.$v_1$ 称为 W 的起点, v_{k+1} 称为 W 的终点, W 上的其他顶点称为 W 的内部顶点, 并称 W 为 D 的有向 (v_1, v_{k+1}) 途径. k 称为 W 的长. 有向途径 $v_1a_1v_2\cdots v_ka_kv_{k+1}$ 常常简单地用 $v_1v_2\cdots v_kv_{k+1}$ 来表示.

有向途径 $W = v_1a_1v_2\cdots v_ka_kv_{k+1}$ 的节是指由 W 中相继项构成的子序列 $v_ia_iv_{i+1}\cdots v_{j-1}a_{j-1}v_j$, 这一子序列也称为 W 的 (v_i, v_j) 节. 如果 W_1 和 W_2 都

是有向图 D 中的有向途径, 且 W_1 的终点正好是 W_2 的起点, 则把 W_2 接在 W_1 的后面就得到 D 的一条新的有向途径 W, 记作 $W = W_1W_2$, 并称 W 为 W_1 与 W_2 的衔接, 或称 W 可表示为 W_1 与 W_2 的并.

如果有向图 D 中的有向途径 W 上的弧互不相同, 则称 W 为 D 的有向迹 (directed trail). 与图的链、闭途径、闭迹和圈的定义一样, 完全类似地定义有向图中的有向链、有向闭途径、有向闭迹和有向圈. 为方便计, 我们常常把有向链称为路 (path), 把有向圈称为回路 (circuit).

如果对于有向图 D 中的任意两个顶点 v_i 和 v_j, D 中既存在 (v_i, v_j) 路, 也存在 (v_j, v_i) 路, 则称 D 是强连通的 (strongly connected). 易知, 若有向图 D 是强连通的, 则 D 是连通的, 反之不然. 同图的连通分支类似, 可以定义有向图的强连通分支 (strongly component).

可以用一个形象的比喻来解释连通有向图和强连通有向图的差异. 设想有一个公路网连接若干个城镇, 且每条公路只允许单向行驶, 这样的公路网称为单行公路网. 若把城镇看做顶点, 单行公路看做弧, 则单行公路网可看做一个有向图 D, D 连通相当于从任一城镇出发, 不管公路规定的行驶方向, 驱车可到达任一其他城镇; 而 D 强连通则等同于从任一城镇驾车出发, 严格按照规定的方向行驶, 可以到达任一其他城镇.

图中边割的概念也可以推广到有向图上来. 设 $D = (V, A)$ 是一个有向图, $\forall S, T \subseteq V$, 定义

$$(S, T) = \{(v_i, v_j) \in A | v_i \in S, v_j \in T\}.$$

如果 S 为 V 的非空真子集, $T = V \setminus S$, 则称 (S, T) 为 D 的截集 (cut set).

1.3 几类重要的图

本节介绍一些重要的图类.

任何一对顶点都相邻的简单图称为完全图 (complete graph). n 阶完全图记为 K_n.

设 $G = (V, E)$ 为 n 阶简单图, K_n 为 n 阶完全图, 且 $V(K_n) = V$, 则称简单图 $K_n - E$ 为 G 的补图 (complement), 记为 $\overline{G}$.

设图 $G = (V, E)$, 如果存在 V 的一个划分 X, Y, 使得 G 的任何一条边的一个端点在 X 中, 另一个端点在 Y 中, 则称 G 为二部图 (bipartite graph), 记作 $G = (X, Y, E)$. 完全二部图 (complete bipartite graph) 是指这样的二部图 $G = (X, Y, E)$, 其中 $X \neq \varnothing, Y \neq \varnothing$, 且 X 的每个顶点都与 Y 的每个顶点之间恰有一条边. 若 $|X| = m, |Y| = n$, 则完全二部图 $G = (X, Y, E)$ 记为 $K_{m,n}$.

推而广之, 所谓 k 部图 (k-partite graph)G 是指存在 $V(G)$ 的一个划分 $V_1, V_2, \cdots, V_k$, 使得 V_i 的任何两个顶点在 G 中都不相邻, $i = 1, 2, \cdots, k$. 又若 $V_i \neq \varnothing (1 \leqslant i \leqslant k)$, 且 V_i 中任一顶点与 V_j 中任一顶点之间恰有一条边 $(1 \leqslant i < j \leqslant k)$, 则称 G 为完全 k 部图 (complete k-partite graph).

二部图有下面的一个重要特征.

定理 1.4 图 G 是二部图当且仅当 G 中不含奇圈.

证明 ($\Rightarrow$) 设二部图 $G = (X, Y, E), C = v_1 v_2 \cdots v_k v_1$ 是 G 中的任意一个圈. 不妨设 $v_1 \in X$. 因 $v_1 v_2 \in E, G$ 是二部图, 故 $v_2 \in Y$. 同理 $v_3 \in X$. 一般说来, $v_{2i-1} \in X, v_{2i} \in Y$. 由于 $v_1 \in X$, 因此 $v_k \in Y$, 所以有某个 i 使 $k = 2i$, 即 C 是偶圈.

($\Leftarrow$) 不妨设 G 连通 (否则, 逐个考察它的连通分支), 并且 G 不含奇圈.

由于 G 连通, 因此 $\forall v_i, v_j \in V(G), G$ 中存在 (v_i, v_j) 链. 我们把 G 中边数最少的 (v_i, v_j) 链称为 G 的最短 (v_i, v_j) 链, 并把它的长记为 $d_G(v_i, v_j)$.

任取 $v \in V(G)$, 令

$$X = \{v_i \in V(G) | d_G(v, v_i) \text{ 是偶数 }\},$$
$$Y = \{v_i \in V(G) | d_G(v, v_i) \text{ 是奇数 }\},$$

从而 X, Y 是 $V(G)$ 的一个划分.

$\forall x_1, x_2 \in X, x_1 \neq x_2$, 设 P 是最短 (v, x_1) 链, Q 是最短 (v, x_2) 链, 记 P 和 Q 的最后一个公共顶点为 v_1. 因 P 和 Q 是最短链, 则用反证法不难证明 P 和 Q 的 (v, v_1) 节都是最短 (v, v_1) 链, 故它们的长度相等. 又因 P 和 Q 的长度都是偶数, 从而 P 的 (v_1, x_1) 节 P_1 的长和 Q 的 (v_1, x_2) 节 Q_1 的长必有相同的奇偶性, 由此推知 (x_1, x_2) 链 $P_1^{-1} Q_1$ 的长为偶数. 如果 x_1 与 x_2 相邻, 则 $P_1^{-1} Q_1 x_2 x_1$ 是一个奇圈, 此与假设矛盾. 因此 X 中任何两个顶点都不相邻. 同理可证 Y 中任何两个顶点也不相邻. □

不含圈的图称为无圈图 (acyclic graph). 连通的无圈图称为树 (tree). 如果 T 是图 G 的一个支撑子图, 且 T 是树, 则称 T 为 G 的支撑树 (spanning tree). 同样, 我们把有向图 D 中连通无圈的支撑子图也称为 D 的支撑树.

如果图 G 中存在包含一切边的闭迹 W, 则称 G 是 Euler 图 (Euler graph), W 称为 G 的 Euler 闭迹 (Euler closed trail). 如果有向图 D 中存在包含一切弧的有向闭迹 W, 则称 D 是有向 Euler 图 (Euler digraph), W 称为 D 的有向 Euler 闭迹 (directed Euler closed trail).

如果图 G 中存在包含一切顶点的圈 C, 则称 G 是 Hamilton 图 (Hamilton graph), C 称为 G 的 Hamilton 圈 (Hamilton cycle). 包含图 G 中一切顶点的链称为 G 的 Hamilton 链 (Hamilton chain). 类似地, 在有向图中定义 Hamilton 回路