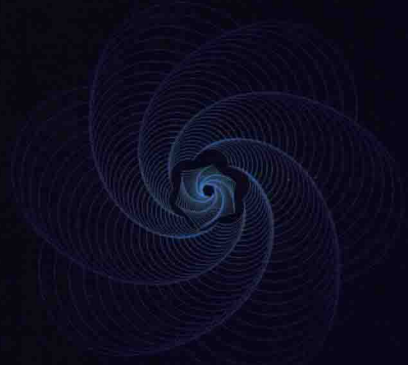
“十三五”国家重点图书出版规划项目
湖北省公益学术著作出版专项资金资助项目
智能制造与机器人理论及技术研究丛书

总主编　丁汉　孙容磊

智能系统新概念数学方法概论

（下册）

朱剑英◎编著



ZHINENG XITONG XINGAINIAN
SHUXUE FANGFA GAILUN

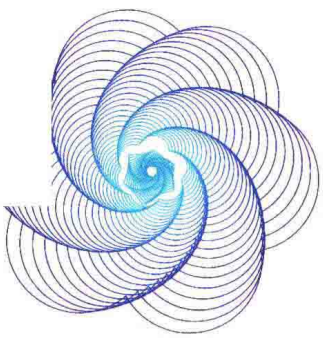
華中科技大學出版社
http://www.hustp.com

“十三五”国家重点图书出版规划项目
湖北省公益学术著作出版专项资金资助项目
智能制造与机器人理论及技术研究丛书
总主编　丁汉　孙容磊

智能系统新概念数学方法概论

（下册）

朱剑英◎编著



ZHINENG XITONG XINGAINIAN
SHUXUE FANGFA GAILUN

華中科技大學出版社
http://www.hustp.com
中国·武汉

内 容 简 介

本书全面、系统汇集并研究了当前和未来在智能系统(包括人工智能)领域所应用的经典与非经典的智能数学方法,至今在国内外尚未见有同类著作发表。本书的特点是:

(1) 从三次数学危机的历史高度出发论证了智能科学、技术、工程的必然发展趋势与创新空间;

(2) 以人工智能科学发展的三大学派——逻辑主义学派、联结主义学派、行为主义学派为线索,介绍与论证了相关的经典与非经典数学方法;

(3) 紧密结合当前与未来人工智能的广泛而深入的应用,精选了十大学科(数理逻辑、集合论、概率论、数理统计、运筹学、图论、组合优化、模糊数学、神经网络、遗传算法)做了全面、系统、精要、启发式的论述与研讨;

(4) 每章都结合所介绍的数学原理和方法,阐述了作者关于创新发展的思悟和建议。

本书适合在智能系统(包括人工智能)领域工作的所有教学、科研、生产人员学习、参考和应用。

图书在版编目(CIP)数据

智能系统新概念数学方法概论:上下册/朱剑英编著. —武汉:华中科技大学出版社,2022.1
(智能制造与机器人理论及技术研究丛书)
ISBN 978-7-5680-5766-0

Ⅰ.①智… Ⅱ.①朱… Ⅲ.①智能系统-数学方法-研究 Ⅳ.①TP18

中国版本图书馆 CIP 数据核字(2021)第 254606 号

智能系统新概念数学方法概论(上下册)
ZHINENG XITONG XINGAINIAN SHUXUE FANGFA GAILUN

朱剑英 编著

策划编辑:俞道凯
责任编辑:戢凤平 刘 飞
封面设计:原色设计
责任监印:周治超
出版发行:华中科技大学出版社(中国·武汉) 电话:(027)81321913
武汉市东湖新技术开发区华工科技园 邮编:430223
录 排:武汉市洪山区佳年华文印部
印 刷:湖北新华印务有限公司
开 本:710mm×1000mm 1/16
印 张:47.75 插页:5
字 数:882 千字
版 次:2022 年 1 月第 1 版第 1 次印刷
定 价:298.00 元(含上下册)

下册目录

第 6 章　图论与网络优化　/1

6.1　基本概念　/1

6.1.1　古典问题　/1

6.1.2　基本定义与定理　/2

6.2　树与最小支撑树　/8

6.2.1　树的定义及其性质　/8

6.2.2　支撑树与最小树　/9

6.3　最短路问题　/11

6.3.1　Dijkstra 标号法　/11

6.3.2　福劳德算法　/19

6.4　网络最大流问题　/21

6.4.1　基本概念与基本定理　/22

6.4.2　求解网络最大流的标号法　/24

6.5　最小费用最大流问题　/30

6.6　中国邮递员问题　/32

第 7 章　模糊数学　/35

7.1　模糊集合论的基本概念　/35

7.1.1　经典集合论的基本概念　/35

7.1.2　模糊集合的定义　/42

7.1.3　模糊集合的运算　/44

7.2　模糊集合的分解定理　/48

7.2.1　模糊集合的截集　/48

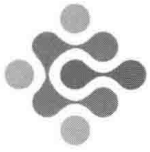
7.2.2　分解定理　/51

7.3　模糊集的隶属度　/53

7.3.1 边界法 /53
7.3.2 模糊统计法 /55
7.3.3 参照法 /57
7.4 模糊集合的扩张原理 /62
7.4.1 经典集合的扩张原理 /62
7.4.2 模糊集合的扩张原理 /63
7.4.3 多元扩张原理 /65
7.5 模糊模式识别 /70
7.5.1 模糊模式识别的直接方法 /70
7.5.2 模糊距离与模糊度 /75
7.5.3 贴近度 /82
7.5.4 多因素模糊模式识别 /88
7.6 模糊关系与聚类分析 /94
7.6.1 经典关系 /94
7.6.2 模糊关系的基本概念 /99
7.6.3 模糊等价关系 /104
7.6.4 模糊传递闭包和等价闭包 /110
7.6.5 求相似矩阵的等价类的直接方法 /116
7.6.6 直接聚类的最大树法 /121
7.6.7 模糊聚类分析 /122
7.6.8 模糊 ISODATA 法 /128
7.7 模糊综合评判 /132
7.7.1 模糊变换 /132
7.7.2 简单模糊综合评判 /133
7.7.3 不完全评判问题 /135
7.7.4 多层次模糊综合评判 /137
7.7.5 广义合成运算的模糊综合评判模型 /140
7.8 模糊逻辑与模糊推理 /141
7.8.1 模糊逻辑 /141
7.8.2 模糊语言 /147
7.8.3 模糊推理 /150
第 8 章 人工神经网络的数学基础 /162
8.1 概述 /162

8.1.1 人工神经网络研究简史 /162
8.1.2 人脑神经元与人工神经元模型 /164
8.1.3 人工神经网络模型 /167
8.1.4 神经网络的学习规则 /167
8.2 前向神经网络 /170
8.2.1 感知器 /170
8.2.2 有导师学习网络 /171
8.2.3 改进的 BP 算法 /176
8.3 Hopfield 网络 /182
8.3.1 离散型 Hopfield 网络 /182
8.3.2 旅行商问题 /187
8.4 自组织神经网络 /191
8.5 随机神经网络 /193
8.5.1 Boltzmann 分布 /193
8.5.2 模拟退火 /193
8.5.3 随机神经网络的概率分布 /194
8.5.4 多层前馈随机网络 /195
8.6 模糊神经网络 /197
8.6.1 模糊神经元模型 /197
8.6.2 模糊 Hopfield 网络 /200
8.7 深度学习:卷积神经网络 /203
8.7.1 概述 /203
8.7.2 卷积神经网络的结构 /205
8.7.3 卷积神经网络的基本算法 /206
8.7.4 卷积神经网络的演变脉络 /213
第 9 章 遗传算法 /216
9.1 概述 /216
9.1.1 遗传算法的生物学基础 /216
9.1.2 遗传算法发展简史 /219
9.1.3 遗传算法的特点 /220
9.2 基本的遗传算法 /221
9.3 遗传算法的基本理论与方法 /226
9.3.1 模式定理 /226

9.3.2 误导问题 /230
9.3.3 编码 /234
9.3.4 群体设定 /238
9.3.5 适应度函数 /239
9.3.6 选择 /243
9.3.7 交换 /245
9.3.8 变异 /247
9.3.9 性能评估 /249
9.3.10 收敛性 /249
9.4 非线性问题寻优的遗传算法 /251
9.4.1 一般非线性优化问题的遗传算法 /252
9.4.2 约束最优化的遗传算法 /254
9.5 背包问题 /255
9.5.1 问题描述 /255
9.5.2 背包问题的遗传算法求解 /256
9.5.3 进一步的讨论 /258
9.6 旅行商问题 /258
9.6.1 编码与适应度 /259
9.6.2 遗传操作 /259
9.6.3 实例 /262
9.7 调度问题 /264
9.7.1 问题概述 /264
9.7.2 调度问题的遗传算法求解 /266
9.8 混合遗传算法 /271
9.8.1 遗传算法优化神经网络 /271
9.8.2 遗传算法优化模糊推理规则 /275
9.9 群体智能算法 /279
9.9.1 概述 /279
9.9.2 蚁群算法 /280
9.9.3 粒子群算法 /285
参考文献 /290
后记 /311



第6章 图论与网络优化

图是最形象的数学语言，它能直观地反映问题。通过图，不但可以简化问题的描述，而且可以很方便地对问题进行分析，并迅速得到问题求解的结果。所以在解决实际问题时，图有十分广泛的应用。图论和网络优化目前已广泛地应用在物理学、化学、控制论、信息论、人工智能、科学管理、计算机科学等诸多领域。在实际生活与生产活动中，许多问题都可以使用图与网络的相关理论与方法来解决。

6.1 基本概念

6.1.1 古典问题

1. “哥尼斯堡的七座桥”问题

图论的研究已有200多年的历史，早期图论与“数学游戏”有着密切关系，“哥尼斯堡的七座桥”问题便是其中之一。

200多年前的东普鲁士有一座哥尼斯堡城（现属俄罗斯加里宁格勒），城中有一条河叫普雷格尔河，河中有两个岛屿共建七座桥，如图6.1.1所示。平时城中居民大都喜欢来这里散步，有人提出这样一个问题：一个散步者能否经过每座桥恰好一次最后回到原出发点。

当时有许多人都探讨了这个问题，但不得其解。著名数学家欧拉（Euler）于1736年发表了图论方面的第一篇论文，将此难题化成了一个数学问题：用点表示两岸或小岛，用点之间的连线表示陆地之间的桥，这样就得到了图6.1.2所示的一个图。从而，问题变为：在这个图中，是否可能从某一点出发只经过各条边一次且仅仅一次而又回到出发点？即一笔画问题。

欧拉否定了这种可能性，原因是图中与每一个点相关联的线都是奇数条（要回到出发点，每一点相连的边，必须是偶数条）。

2. 哈密尔顿图

1859年，哈密尔顿提出了一种游戏：在一个实心的12面体（见图6.1.3）的

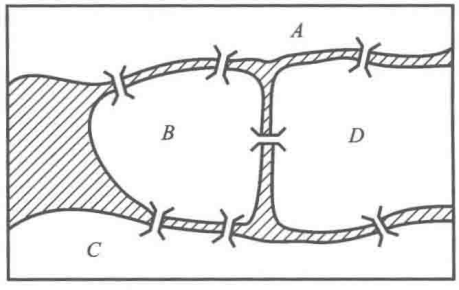


图 6.1.1

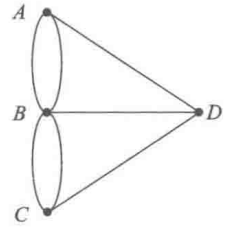


图 6.1.2

20 个顶点上标以世界上著名的城市名称，要求游戏者从某一城市出发，遍历各城市恰恰一次而返回原地，这就是“绕行世界问题”。我们可以按此问题作出如图 6.1.4 所示的图形，该问题变为：从某一点出发寻找一条路径，过所有 20 个点仅一次，再回到出发点。解决这个问题可以按图中的序号 1→2→3→4→…→20→1 所形成的一个闭合路径来完成。此路径称为哈密尔顿图。虽然这个“绕行世界问题”解决了，但是由此引出的“对于给定的连通图（见定义 6.1.5），让它成为哈密尔顿图的充要条件是什么？”至今尚无定论，这是图论中一个著名的尚未解决的问题。

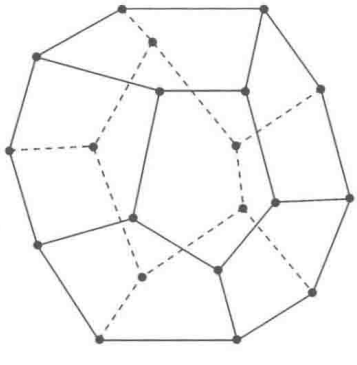


图 6.1.3

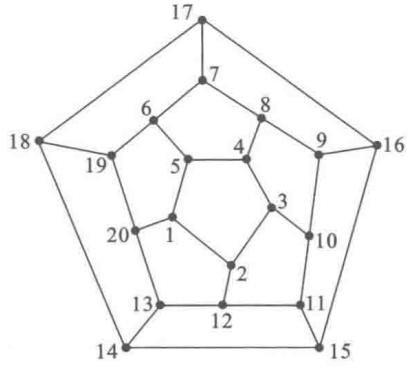


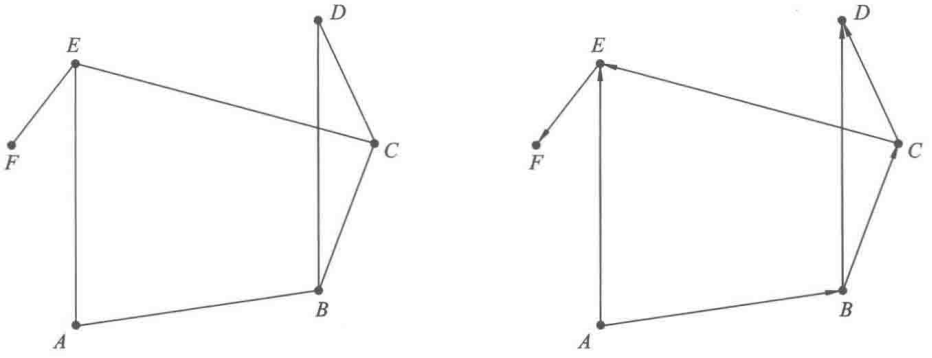
图 6.1.4

由此可见，图论中所研究的图是由实际问题抽象出来的逻辑关系图，这种图与几何中的图形和函数论中所研究的图形是不同的。逻辑关系图的画法具有一定的随意性，在保持相对位置和相互相连关系不变的前提下，点的位置不一定按实际情形来画，线的长度也不一定表示实际的长度，而且画成直线或曲线都可以。通俗地说，这种图是一种关系示意图。

6.1.2 基本定义与定理

图论中的图由点和边组成，点和边取决于实际问题的需要。例如，在“哥尼

斯堡的七座桥”问题中，点表示陆地和岛屿，边表示桥。而在“绕行世界”问题中，点则表示城市，边则表示城市之间的通路。又如，若干球队比赛，可以用点表示球队，点间的连线表示哪两个球队比赛，如图 6.1.5(a)所示，这个图称为无向图。若要表示胜负结果，则可以用箭头表示：箭头指向的一方为负方，如图 6.1.5(b)所示，这种图称为有向图。



(a) 无向图　　(b) 有向图

图 6.1.5

定义 6.1.1　图由表示具体事物的点(顶点)的集合 $\boldsymbol{V}=\{v_1,v_2,\cdots,v_n\}$ 和表示事物之间关系边的集合 $\boldsymbol{E}=\{e_1,e_2,\cdots,e_m\}$ 所组成，且 $\boldsymbol{E}$ 中的元素 e_i 用 $\boldsymbol{V}$ 中的无序元素对 $[v_i,v_j]$ 表示，即 $e_i=[v_i,v_j]$，记为 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$，并称这类图为无向图。

例如，在图 6.1.6 中，有 8 条边，6 个顶点，即

$$\boldsymbol{V}=\{v_1,v_2,\cdots,v_6\},\quad \boldsymbol{E}=\{e_1,e_2,\cdots,e_8\}$$

其中：

$$e_1=[v_1,v_2]=[v_2,v_1]$$
$$e_2=[v_2,v_3]=[v_3,v_2]$$
$$e_3=[v_3,v_4]=[v_4,v_3]$$
$$e_4=[v_4,v_4]$$
$$e_5=[v_4,v_2]=[v_2,v_4]$$
$$e_6=[v_4,v_5]=[v_5,v_4]$$
$$e_7=[v_2,v_5]=[v_5,v_2]$$
$$e_8=[v_2,v_5]=[v_5,v_2]$$

定义 6.1.2

(1) 顶点数和边数：图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 中，$\boldsymbol{V}$ 中元素的个数称为图 $\boldsymbol{G}$ 的顶点数，记作 $p(\boldsymbol{G})$ 或简记为 p；$\boldsymbol{E}$ 中元素的个数称作图 $\boldsymbol{G}$ 的边数，记为 $q(\boldsymbol{G})$，或简记

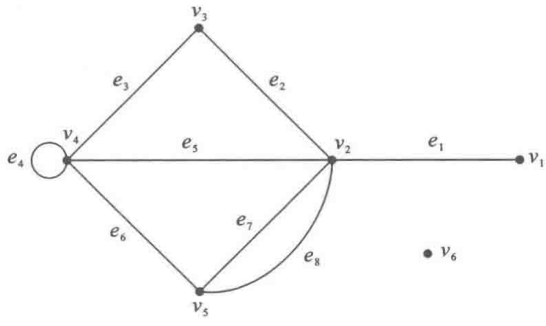


图 6.1.6

为 q。

(2) 端点和关联边：若 $e_i=[v_i,v_j]\in \boldsymbol{E}$，则称点 v_i,v_j 是边 e_i 的端点，边 e_i 是点 v_i 和 v_j 的关联边。

(3) 相邻点和相邻边：同一条边的两个端点称为相邻点，简称邻点；有公共端点的两条边称为相邻边，简称邻边。

(4) 多重边与环：具有相同端点的边称为多重边或平行边；两个端点落在一个顶点的边称为环。

(5) 多重图和简单图：含有多重边的图称为多重图；无环也无多重边的图称为简单图。

(6) 次：以 v_i 为端点的边的条数称为点 v_i 的次，记作 $d(v_i)$。

(7) 悬挂点和悬挂边：次为 1 的点称为悬挂点；与悬挂点相连的边称为悬挂边。

(8) 孤立点：次为零的点称为孤立点。

(9) 奇点与偶点：次为奇数的点称为奇点；次为偶数的点称为偶点。

例如，在图 6.1.6 中，$p(\boldsymbol{G})=6$，$q(\boldsymbol{G})=8$；$e_3=[v_4,v_3]$，v_4 与 v_3 是 e_3 的端点，e_3 是 v_4 和 v_3 的关联边；v_2 与 v_5 是邻点，e_3 与 e_2 是邻边；e_7 与 e_8 是多重边，e_4 是一个环；图 6.1.6 是一个多重图；v_1 是悬挂点，e_1 是悬挂边；v_6 是孤立点；v_2 是奇点，v_3 是偶点。

定理 6.1.1 图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 中，所有点的次之和是边数的两倍，即

$$\sum_{v_i\in V} d(v_i)=2q$$

定理 6.1.1 是显然的，因为在计算各点的次时，每条边都计算了两次，于是图 $\boldsymbol{G}$ 中全部顶点的次之和就是边数的两倍。

定理 6.1.2 任一图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$ 中，奇点的个数为偶数。

证明 设 $\boldsymbol{V}_1,\boldsymbol{V}_2$ 分别是 $\boldsymbol{G}$ 中奇点和偶点的集合，由定理 6.1.1 可知

$$\sum_{v_i \in V_1} d(v_i) + \sum_{v_i \in V_2} d(v_i) = \sum_{v_i \in V} d(v_i) = 2q \tag{6.1.1}$$

因为 $\sum_{v_i \in V} d(v_i)$ 是偶数，而 $\sum_{v_i \in V_2} d(v_i)$ 也是偶数，故 $\sum_{v_i \in V_1} d(v_i)$ 必是偶数，由于偶数个奇数才能导致偶数，所以奇点的个数必须为偶数。

定义 6.1.3

(1) 链：在一个图 $G=(V,E)$ 中，一个由点与边构成的交错序列 $(v_{i1}, e_{i1}, v_{i2}, e_{i2}, \cdots, v_{i(k-1)}, e_{i(k-1)}, v_{ik})$ 如果满足 $e_{it}[e_{it}, e_{it+1}]$ $(t=1,2,\cdots,k-1)$，则称此序列为一条联结 v_{i1}, v_{ik} 的链，记为 $\mu=(v_{i1}, v_{i2}, \cdots, v_{ik})$，称点 $v_{i2}, v_{i3}, \cdots, v_{i(k-1)}$ 为链中的中间点。

(2) 闭链与开链：若链 μ 中 $v_{i1}=v_{ik}$，即始点与终点重合，则称此链为闭链(圈)；否则，称之为开链。

(3) 简单链和初等链：若链 μ 中所含的边均不相同，则称之为简单链；若链 μ 中，顶点 $v_{i1}, v_{i2}, \cdots, v_{ik}$ 都不相同，则称此链为初等链。除非特别交代，以后我们讨论的均指初等链。

例如，图 6.1.6 中，$\mu_1=(v_2, e_2, v_3, e_3, v_4, e_6, v_5)$ 是一条链，由于链 μ_1 里所含的边和点均不相同，故是一条初等链；而 $\mu_2=(v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_5, v_2, e_1, v_1)$ 是一条闭链。

定义 6.1.4

(1) 回路：一条闭的链称为回路。

(2) 通路：一条开的初等链称为通路。

(3) 简单的回路和初等回路：若回路中的边都互不相同，则称为简单回路；若回路中的边和顶点都互不相同，则称为初等回路或圈。

定义 6.1.5 一个图 G 的任意两个顶点之间，如果至少有一条通路将它们连接起来，则这个图 G 就称为连通图，否则称为不连通图。

例如，图 6.1.6 中，v_1 与 v_6 没有一条通路把它们连接起来，故此图是不连通图。本章以后讨论的图，除特别声明外，均指连通图。

定义 6.1.6

(1) 子图：设 $G_1=\{V_1, E_1\}$，$G_2=\{V_2, E_2\}$，如果 $V_1 \subseteq V_2$，又 $E_1 \subseteq E_2$，则称 G_1 为 G_2 的子图。

(2) 真子图：若 $V_1 \subset V_2$，$E_1 \subset E_2$，即 G_1 中不包含 G_2 中所有的顶点和边，则称 G_1 是 G_2 的真子图。

(3) 部分图：若 $V_1=V_2$，$E_1 \subset E_2$，即 G_1 中不包含 G_2 中所有的边，则称 G_1 是 G_2 的一个部分图。

(4) 支撑子图：若 G_1 是 G_2 的部分图，且 G_1 是连通图，则称 G_1 是 G_2 的支

撑子图。

(5) 生成子图:若 G_1 是 G_2 的真子图,且 G_1 是不连通图,则称 G_1 是 G_2 的生成子图。

例如,图 6.1.7 中,图(b)是图(a)的真子图,图(c)是图(a)的部分图,图(d)是图(a)的支撑子图,图(e)是图(a)的生成子图。

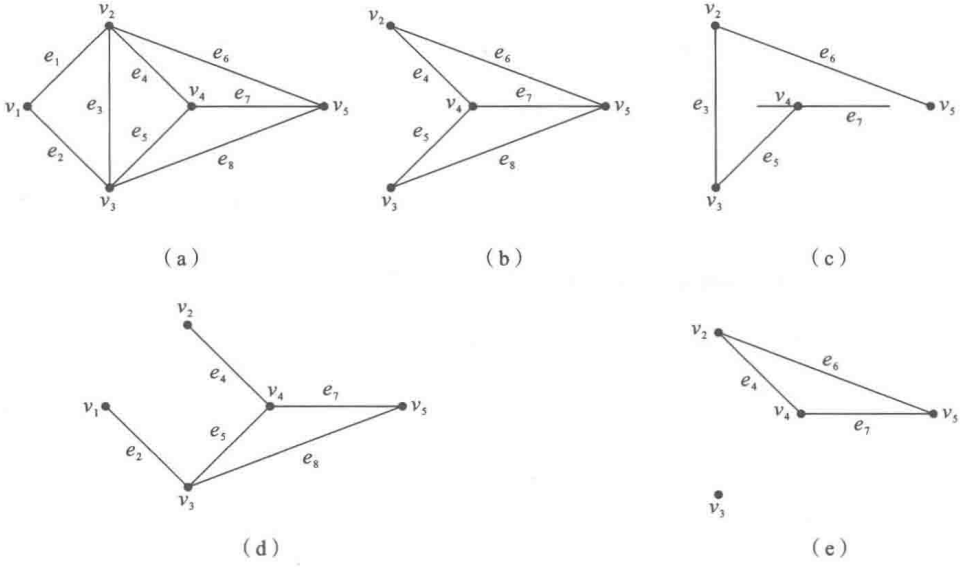


图 6.1.7

定义 6.1.7 设 $G=(V,E)$ 中,对于任意一条边 $e\in E$,如果相应都有一个权值 $W(e)$,则称 G 为赋权图,$W(e)$ 称为边 e 的权。

图 6.1.8 是一个赋权图。

$$e_1=[v_1,v_2],\ W(e_1)=1;\quad e_2=[v_1,v_3],\ W(e_2)=4$$
$$e_3=[v_2,v_3],\ W(e_3)=2;\quad e_4=[v_2,v_4],\ W(e_4)=3$$
$$e_5=[v_3,v_4],\ W(e_5)=1;\quad e_6=[v_2,v_5],\ W(e_6)=5$$
$$e_7=[v_4,v_5],\ W(e_7)=2;\quad e_8=[v_3,v_5],\ W(e_8)=3$$

可见,赋权图不仅指出各点之间的邻接关系,而且也表示各点之间连接的数量关系,所以赋权图在图的理论及应用方面有着重要的地位。

在很多实际问题中,事物之间的联系是带有方向性的。如图 6.1.5(b)所示,箭头的方向指向负方的球队。又如图 6.1.9 所示,v_1 表示某一水系的发源地,v_6 表示这个水系的入海口,图中的箭头则表示各支流的水流方向,图 6.1.9 是水系流向图。图 6.1.5(b)和图 6.1.9 称为有向图。下面给出有向图和赋权有向图的正规定义。

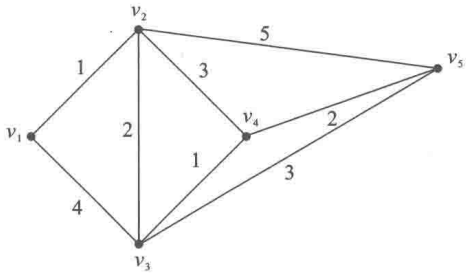


图 6.1.8

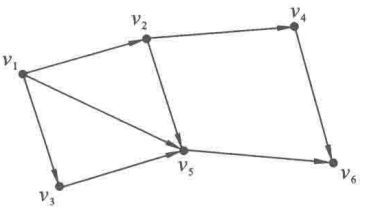


图 6.1.9

定义 6.1.8 设 $\boldsymbol{V}=\{v_1,v_2,\cdots,v_n\}$ 是由 n 个顶点组成的非空集合，$\boldsymbol{A}=\{a_1,a_2,\cdots,a_m\}$ 是由 m 条边组成的集合，且有 $\boldsymbol{A}$ 中元素 a_i 是 $\boldsymbol{V}$ 中的一个有序元素对 (v_i,v_j)，则称 $\boldsymbol{V}$ 和 $\boldsymbol{A}$ 构成了一个有向图，记作 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{A})$，$a_i=(v_i,v_j)$ 表明 v_i 和 v_j 分别为边 a_i 的起点和终点，称有方向的边 a_i 为弧（在图中用带有箭头的线表示）。

例如，图 6.1.9 中，(v_1,v_2)，(v_1,v_3)，(v_2,v_5) 都是 $\boldsymbol{A}$ 中的元素，$\boldsymbol{A}$ 是弧的集合。

与无向图类似，在有向图中也可以定义多重边、环、简单图、链等概念。只是在无向图中，链与路、闭链与回路概念是一致的，而在有向图中，这两个概念却不能混为一谈。概括地说，一条路必定是一条链。然而在有向图中，一条链未必是一条路，只有在相邻的两弧的公共结点是其中一条弧的终点，同时又是另一条弧的始点时，这条链才能叫做一条路。

例如，图 6.1.9 中 $\{v_1,(v_1,v_2),v_2,(v_2,v_5),v_5,(v_5,v_6)\}$ 是一条链，也是一条路，而 $\{v_1,(v_1,v_2),v_2,(v_2,v_5),v_5,(v_3,v_5),v_3\}$ 是一条链但不是一条路。

图 6.1.10 是表示某地区交通运输公路分布、走向及相应费用的有向图。箭头表示走向，箭头旁边的数字表示费用。这类图称为赋权有向图。

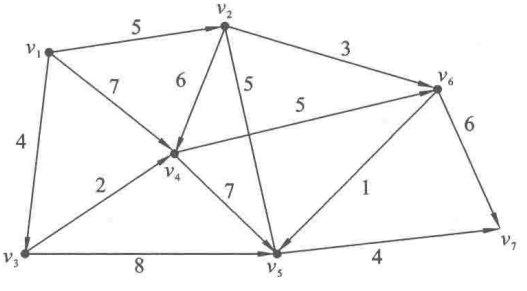


图 6.1.10

定义 6.1.9 设在有向图 $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{A})$ 中，对于任意一条弧 $a_{ij}\in\boldsymbol{A}$，如果相应都有一个权值 $W(a_{ij})$，则称 $\boldsymbol{G}$ 为赋权有向图，$W(a_{ij})$ 称为弧 a_{ij} 的权，简记为 W_{ij}

(权可以表示距离、费用和时间等)。

在实际工作中,有很多问题的可行解方案都可以通过一个赋权有向图来表示,例如,物流渠道的设计、物资运输路线的安排、装卸设备的更新、排水管道的铺设等。所以,赋权图被广泛应用于解决工程技术及科学管理等领域的最优化问题。

通常,我们称赋权图为网络,称赋权有向图为有向网络,称赋权无向图为无向网络。我们讨论的目标是要解决这些网络的计划与优化问题,其数学方法是:最小支撑树、最短路、网络最大流、最小费用等问题的求解方法。下面逐节加以介绍。

6.2 树与最小支撑树

6.2.1 树的定义及其性质

树是各图中最简单的一种图,由于模型简单,它在现实中应用非常广泛。

例 6.2.1 某企业组织机构如图 6.2.1(a)所示:

如果用图表示,则如图 6.2.1(b)所示,它是一个呈树枝形状的图,"树"的名称由此而来。

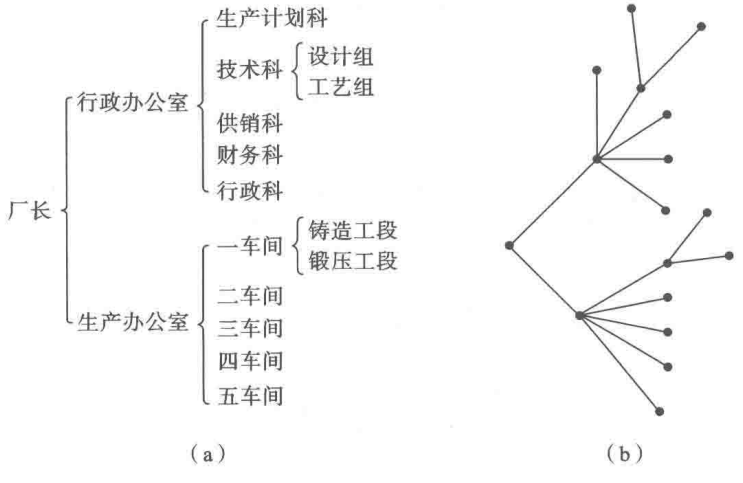


图 6.2.1

定义 6.2.1 一个无回路(圈)的连通无向图称为树。

树的性质如下:

(1) 树必连通,但无回路(圈);

(2) n 个顶点的树必有 $n-1$ 条边;

(3) 树中任意两点间,恰有一条初等链;

(4) 树连通,但去掉任一条边,必变为不连通;

(5) 树无回路(圈),但不相邻顶点连一条边,恰得一回路(圈)。

6.2.2 支撑树与最小树

定义 6.2.2 设图 $G_1=(V,E_1)$ 是图 $G=\{V,E\}$ 的支撑子图,如果 G_1 是一棵树,记 $T=(V,E_1)$,则称 T 是 G 的一棵支撑树。

定理 6.2.1 图 G 有支撑树的充分必要条件是图 G 的连通。

证明 必要性是显然的。

充分性的证明如下:

设 G 是连通图。

(1) 如果不含圈,由定义 6.2.1 可知,G 本身就是一棵树,从而 G 是它自身的支撑树。

(2) 如果 G 含圈,任取一圈,从圈中任意去掉一条边,得到 G 的一个支撑子图 G_1。如果 G_1 不含圈,那么 G_1 是 G 的一棵支撑树(因为易见 G_1 是连通的);如果 G_1 仍含圈,那么从 G_1 中任取一个圈,从圈中再任意去掉一条边,得到 G 的一个支撑子图 G_2。如此重复,最终可以得到 G 的一个支撑子图 G_k,它不含圈,则 G_k 是 G 的一棵支撑树。

由以上充分性的证明中,提供了一个寻求连通图的支撑树的方法,这种方法称为“破圈法”。

例 6.2.2 在图 6.1.7(a)中,用破圈法求出图的一棵支撑树。

解 取一圈 $\{v_1\ e_1\ v_2\ e_3\ v_3\ e_2\ v_1\}$ 去掉 e_3;取一圈 $\{v_1\ e_1\ v_2\ e_4\ v_4\ e_5\ v_3\ e_2\ v_1\}$ 去掉 e_5;取一圈 $\{v_2\ e_4\ v_4\ e_7\ v_5\ e_6\ v_2\}$ 去掉 e_7;取一圈 $\{v_1\ e_1\ v_2\ e_6\ v_5\ e_8\ v_3\ e_2\ v_1\}$ 去掉 e_6。

如图 6.2.2 所示,此图是图 6.1.7(a)的一个支撑子图,且为一棵树(无圈),所以我们找到一棵支撑树 $T_1=\{V,E_1\}$,其中,$E_1=\{e_1,e_4,e_2,e_8\}$。

不难发现,图的支撑树不是唯一的,对于上例若这样做:

取一圈 $\{v_1\ e_1\ v_2\ e_3\ v_3\ e_2\ v_1\}$ 去掉 e_3;$\{v_1\ e_1\ v_2\ e_4\ v_4\ e_5\ v_3\ e_2\ v_1\}$ 去掉 e_4;取一圈 $\{v_1\ e_1\ v_2\ e_6\ v_5\ e_8\ v_3\ e_2\ v_1\}$ 去掉 e_6;取一圈 $\{v_4\ e_7\ v_5\ e_8\ v_3\ e_5\ v_4\}$ 去掉 e_8。

如图 6.2.3 所示,得到图 6.1.7(a)的另一棵支撑树 $T_2=\{V,E_2\}$,其中 $E_2=\{e_1,e_2,e_5,e_7\}$。

求图 G 的支撑树还有另外一种方法“避圈法”,主要步骤是在图中任取一条边 e_1,找出一条不与 e_1 构成圈的边 e_2,再找出不与 $\{e_1,e_2\}$ 构成圈的边 e_3……一般地,设 e 有 $\{e_1,e_2,\cdots,e_k\}$,找出一条不与 $\{e_1,e_2,\cdots,e_k\}$ 构成圈的边 e_{k+1},重复

这个过程，直到不能进行下去为止，这时，由所有取出的边所构成的图是图 G 的一棵支撑树。

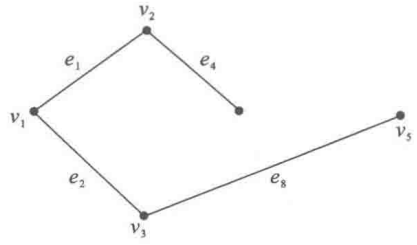


图 6.2.2

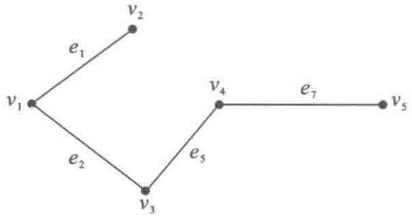


图 6.2.3

定义 6.2.3 设 $T=(V,E')$ 是赋权图 $G=(V,E)$ 的一棵支撑树，称 E' 中全部边上的权数之和为支撑树 T 的权，记为 $W(T)$，记

$$W(T)=\sum_{[v_i,v_j]\in T} w_{ij} \tag{6.2.1}$$

如果支撑树 T^* 的权 $W(T^*)$ 是 G 的所有支撑数的权中最小者，则称 T^* 是 G 的最小支撑树，简称为最小树，即

$$W(T^*)=\min_{T}\{W(T)\} \tag{6.2.2}$$

式中对 G 的所有支撑树 T 取最小。

求最小树通常用以下两种方法。

(1) 破圈法：在给定连通图 G 中，任取一圈，去掉一条最大权边（如果有两条或两条以上的边都是权最大的边，则任意去掉其中一条），在余图中（是图 G 的支撑子图）任取一圈，去掉一条最大权边，重复下去，直到余图中无圈为止，即可得到图 G 的最小树。

例 6.2.3 用破圈法求图 6.1.7(a)（即图 6.1.8）的最小树。

解 取一圈 $\{v_1\ e_1\ v_2\ e_3\ v_3\ e_2\ v_1\}$ 去掉 e_2；取一圈 $\{v_2\ e_6\ v_5\ e_8\ v_3\ e_3\ v_2\}$ 去掉 e_6；取一圈 $\{v_2\ e_4\ v_4\ e_5\ v_3\ e_3\ v_2\}$ 去掉 e_4；取一圈 $\{v_4\ e_7\ v_5\ e_8\ v_3\ e_5\ v_4\}$ 去掉 e_8。

如图 6.2.4 所示，得到一棵支撑树，即为所求的最小树 T^*，$W(T^*)=1+2+1+2=6$。

(2) 避圈法(Kruskal 算法)：在连通图 G 中，任取权值最小的一条边（若有两条或两条以上权相同且最小，则任取一条），在未选边中选一条权值最小的边，要求所选边与已选边不构成圈，重复下去，直到不存在与已选边不构成圈的边为止，那么已选边与顶点构成的图 T 就是所求最小树。

算法的具体步骤如下：

第 1 步：令 $i=1$，$E_0=\varnothing$（空集）。

第 2 步：选一条边 $e_i\in E\backslash E_i$，且 e_i 是使图 $G_i=(V,E_{i-1}\cup\{e\}$ 中不含圈的所