图与网络分析

第六章

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

第3节

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

最短路问题有多种不同的提法

- 定义
  - 一般的最短路问题

给定一个赋权有向图 $D=(V,A,w)$，对每一个弧 $a=(v_i,v_j)$，相应地有权$w(a)=w_{ij}$，又给定D中的两个顶点v_s,v_t.

设P是D中从v_s到v_t的一条路，定义路P的权$w(P)$为P中所有弧的权之和。最短路问题(**shortest path problem**)就是要在所有从v_s到v_t的路中，求一条权最小的路，即求一条从v_s到v_t的路P^*，使得

$$w(P^*)=\min_P w(P)$$

并称P^*为从v_s到v_t的最短路, 路P^*的权称为从v_s到v_t的距离，记为$d(v_s,v_t)$，可简记为d_{st}。

显然， $d(v_s,v_t)$与$d(v_t,v_s)$不一定相等。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 最短路问题的意义
  - 最短路问题是重要的最优化问题之一，许多最优化问题可化为最短路问题或用最短路的算法作为子程序求解，其用途远远超过其直观意义。
- 最短路的性质
  - 如果P是D中从v_s到v_t的最短路， v_i是P中的一个点，那么，从v_s沿P到v_i的路是D中从v_s到v_i的最短路。
- 最短路问题的分类
  - 非负权时，求从一个顶点到其余各顶点的最短路
  - 任意权时，求从一个顶点到其余各顶点的最短路
  - 任意权时，求所有顶点对之间的最短路
  - 最短路问题的各种推广

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- Dijkstra算法(非负权)
  - 算法概述
    - 该算法由Dijkstra于1959年提出，可用于求解指定两点间的最短路，或从指定点到其余各点的最短路，目前被认为是求非负权网络最短路问题的最好算法。
  - 算法思想
    - 从v_s出发，逐步向外探寻最短路。在算法执行过程中， 对网络中每个顶点赋以一个标号，用来记录从顶点v_s到该顶点的最短路的权（此时称为P标号，即永久标号）或最短路权的上界（此时称为T标号，即临时标号）。
    - 方法的每一步是更新T标号，并把某一个具有T标号的点变为具有P标号的点，这样，至多经过$p-1$步（p为网络中顶点的个数），所有顶点都被赋以P标号，从而求出从v_s到其余各点的最短路。

最短路问题

- Dijkstra算法(非负权)
  - 符号说明

    在Dijkstra算法的具体求解过程中，
    - $P(v), T(v)$分别为v点的P标号和T标号；
    - S_i表示第i步迭代时具有P标号的点的集合；
    - $\lambda(v)$记录从v_s到v的有向路或最短路上点v前面一个相邻顶点的下标，当算法终止时，有：
      - $\lambda(v) = m$ 表示从v_s到v的最短路上v的前一个顶点是v_m
      - $\lambda(v) = M$ 表示D中不含从v_s到v的路
      - $\lambda(v) = 0$ 表示$v = v_s$

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

Dijkstra算法的终止有两种可能情形：
（1）所有顶点都获得了P标号，这时从始点到其余各个顶点j间的最短路及距离均已求得；
（2）并不是所有顶点都获得了P标号，对于未获得P标号的顶点，从始点到这些点没有有向路可达，相应地也就没有最短路。

- Dijkstra算法(非负权)
  - 算法步骤
    - 给定赋权有向图$D=(V,A,w)$,
    - (0) 令$i=1$, $S_i=\{v_s\}$, $P(v_s)=0$, $\lambda(v_s)=0$, 对每一个$v\neq v_s$, 令$T(v)=+\infty, \lambda(v)=M$, 令$k=s$.
    - (1) 若$S_i=V$, 则算法终止, 这时对每个$v\in S_i$, 有$d(v_s,v)=P(v)$; 否则转入(2).
    - (2) 考查每个使$(v_k,v_j)\in A$且$\notin S_i$的点v_j. 若$T(v_j)>P(v_k)+w_{kj}$, 则把$T(v_j)$修改为$P(v_k)+w_{kj}$, 把$\lambda(v_j)$修改为k; 否则转入(3).
    - (3) 令$T(v_{j_i})=\min\limits_{v_j\notin S_i} T(v_j)$. 若$T(v_{j_i})<+\infty$, 则把$v_{j_i}$的$T$标号变为$P$标号$P(v_{j_i})=T(v_{j_i})$, 并令$S_{i+1}=S_i\cup\{v_{j_i}\}$, $k=j_i$, $i=i+1$, 转入(1); 否则算法终止, 这时对每个$v\in S_i$, 有$d(v_s,v)=P(v)$, 而对每个$v\notin S_i$, 有$d(v_s,v)=T(v)$.

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

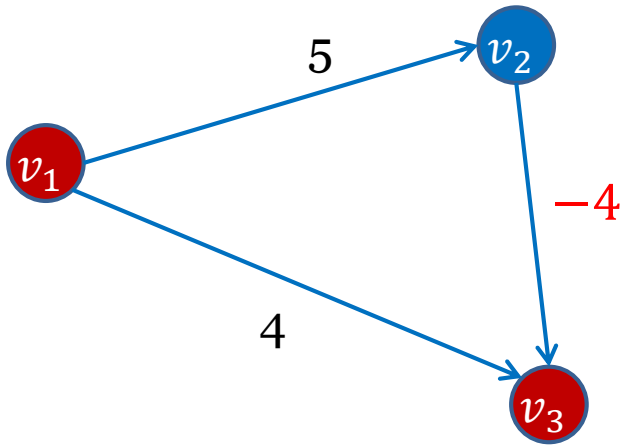
- Dijkstra算法(非负权)
  - 备注
    - 对于赋权无向图$G=(V,E,w)$，边$[v_i,v_j]$可看作是两条弧(v_i,v_j)和(v_j,v_i)，它们 具有相同的权$w[v_i,v_j]=w_{ij}$。若图G中所有的$w_{ij}\geq 0$，则只要把上述Dijkstra算法中“(2) 考查每个使$(v_k,v_j)\in A$且$\notin S_i$的点v_j”改为“(2) 考查每个使$[v_k,v_j]\in E$且$\notin S_i$的点v_j”，同样可求出从v_s到各点的最短路。
    - 当赋权图中存在负权时， Dijkstra算法失效（见下页的示例）。此时可应用《运筹学》（第四版，清华大学出版社）309页给出的一个递推算法(Bellman-Ford)求出从v_s到其余各点的最短路。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- Dijkstra算法(非负权)
  - 备注
    - 示例：当赋权图中存在负权时， Dijkstra算法失效。

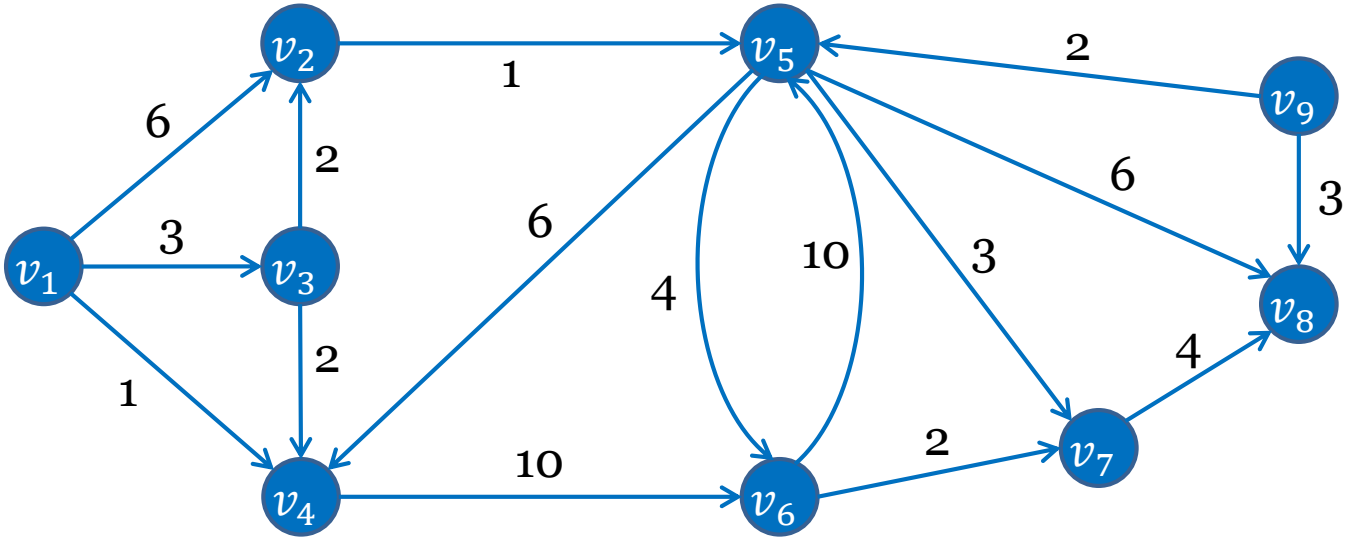
按照Dijkstra算法，得到$P(v_3) = 4$为从v_1到v_3的最短路的权，这显然是错误的，因为$v_s \to v_2 \to v_3$的权只有1.



Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- Dijkstra算法(非负权)
  - 例3 下图所示为单行线交通网，弧旁数字表示通过该单行线所需费用。求从v_1出发，通过该交通网到v_8的总费用最小的旅行路线。

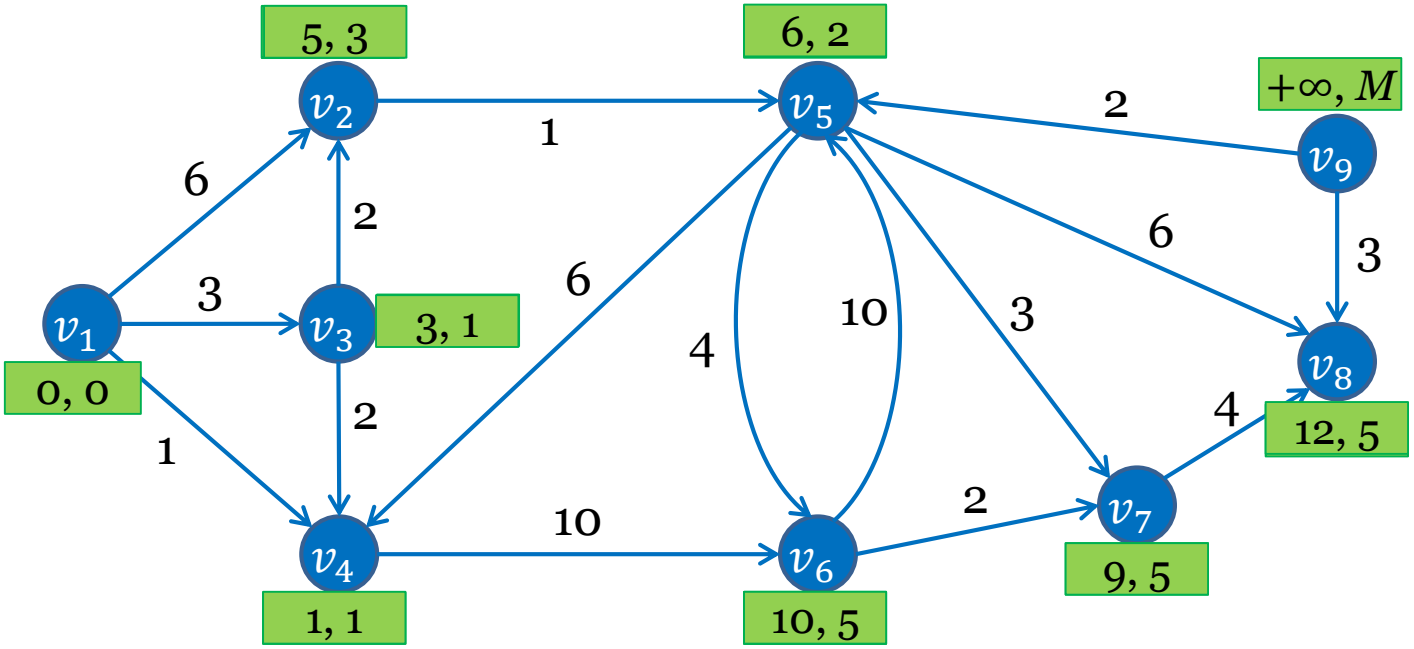


Haiyan Lu
Jiangnan University 4/22/2020

最短路问题

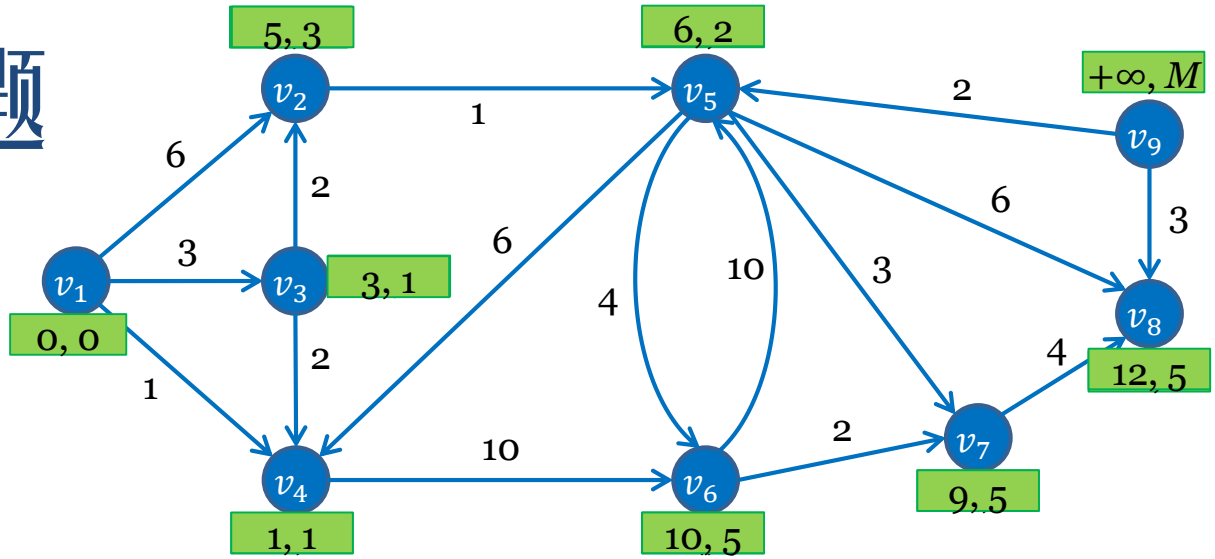
- Dijkstra算法(非负权)
  - 解：

求解过程见下列各图。图中顶点v_i旁方框内的一对数字为$(P(v_i),\ \lambda(v_i))$或$(T(v_i),\ \lambda(v_i))$.



$i = 8$时的标号情况，算法终止。

最短路问题

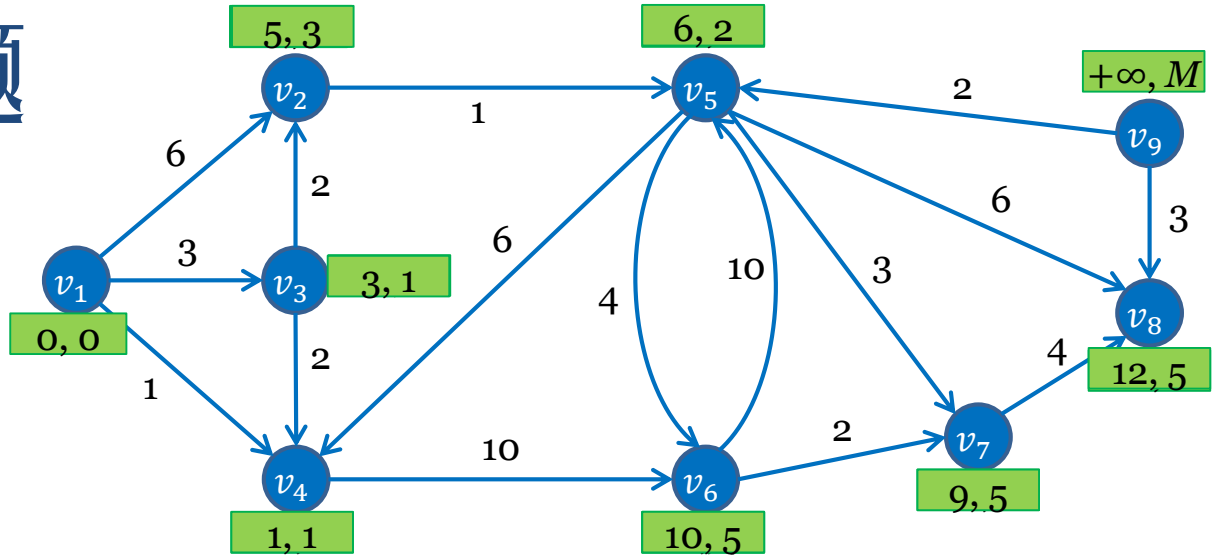


解(续)：算法结束时，所得结果如下：

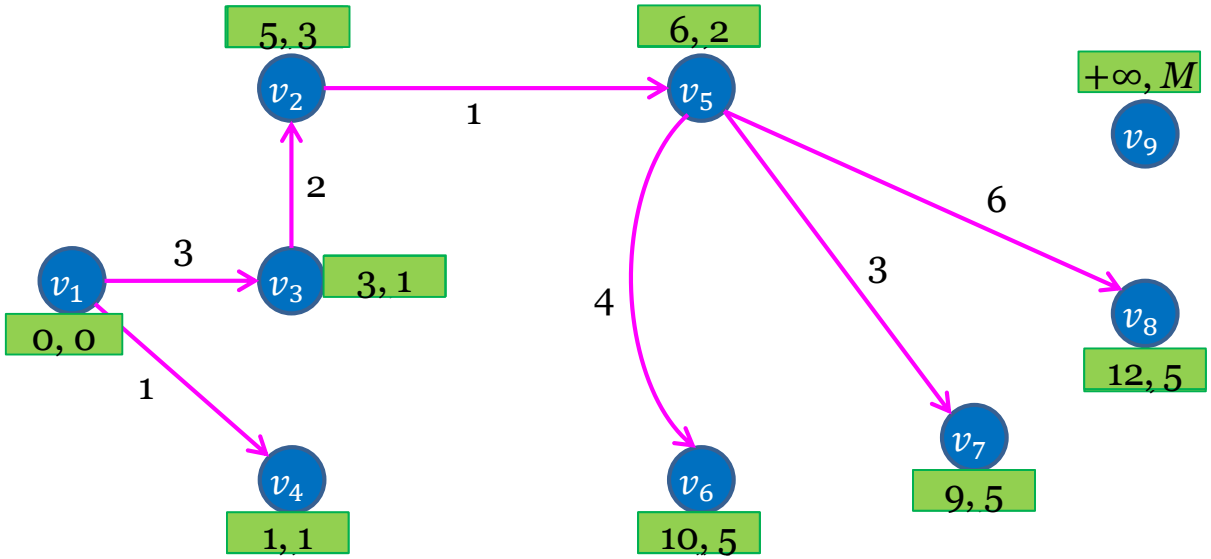
- 从v_1到v_2的最短路长为5，最短路线为(v_1, v_3, v_2);
- 从v_1到v_3的最短路长为3，最短路线为(v_1, v_3);
- 从v_1到v_4的最短路长为1，最短路线为(v_1, v_4);
- 从v_1到v_5的最短路长为5，最短路线为(v_1, v_3, v_2, v_5);
- 从v_1到v_6的最短路长为10，最短路线为$(v_1, v_3, v_2, v_5, v_6)$;
- 从v_1到v_7的最短路长为9，最短路线为$(v_1, v_3, v_2, v_5, v_7)$;
- 从v_1到v_8的最短路长为12，最短路线为$(v_1, v_3, v_2, v_5, v_8)$;
- 从v_1到v_9不存在路。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题



解(续)：此外，由算法求解结果可画出以出发点v_1为根节点的**最短路树**



Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

即《运筹学基础及应用》（第6版）176页所介绍的“矩阵算法”

- 一般标号算法(任意权, 全顶点对)
  - 算法原理
    - 设$D=(V,A,w)$为赋权**有向**图，$W=(w_{ij})_{p\times p}$为权值矩阵(若$(v_i,v_j)\notin A$, 则添加虚拟弧(v_i,v_j), 并令$w_{ij}=\infty$)， p为图 D 的顶点数。显然，**无向**赋权图的权值矩阵 W 为对称阵。
    - **最优性条件**(All-Pairs Shortest Path Optimality Conditions)
      - 设$V=\{1,2,\ldots,n\}$，对任意顶点对$\{i,j\}$，令$d(i,j)$表示图D中从顶点i到顶点j的一条有向路的长度，则$d(i,j)$为从顶点i到顶点j的**最短路**的长度d_{ij}**当且仅当**$d(i,j)$满足:

$$d(i,j)\le d(i,k)+d(k,j)\quad (i,j,k\text{为图}D\text{中任意顶点})$$

一般标号算法通过逐步迭代（**修正标号**）距离标号$d(i,j)$使之最终满足上述**最优性条件**，从而求得网络中任意顶点对间的最短路。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 算法步骤
    - 由上述最优性条件，**一般标号算法**首先给出**直接距离矩阵**：

$$D^{(0)} = (d^0(i,j))_{n\times n} = (w_{ij})_{n\times n} = W$$

    - 然后，利用$D^{(0)}$和下述公式

$$d^1(i,j) = \min_r\{d^0(i,r) + d^0(r,j)\}$$

    计算得到

$$D^{(1)} = (d^1(i,j))_{n\times n}$$

    - 重复上述过程，直至出现 $D^{(m+1)} = D^{(m)}$为止，此时得到**距离矩阵**$D = (d_{ij})_{n\times n}$，算法结束：

$$D = (d_{ij})_{n\times n} = D^{(m+1)} = (d^{m+1}(i,j))_{n\times n}$$

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 备注
    - 算法中所用的迭代公式为

$$d^k(i,j) = \min_r\{d^{k-1}(i,r) + d^{k-1}(r,j)\}$$

这里$d^k(i,j)$表示从顶点i到顶点j的包含至多2^k-1个中间节点的最短路的长度。（**Why?课后思考**）

    - 设图的顶点数为n，一般至多计算到$D^{(k)}$算法可结束，其中:

$$2^{k-1}-1 < n-2 \le 2^k-1$$

即

$$k-1 < \frac{\lg(n-1)}{\lg 2} \le k$$

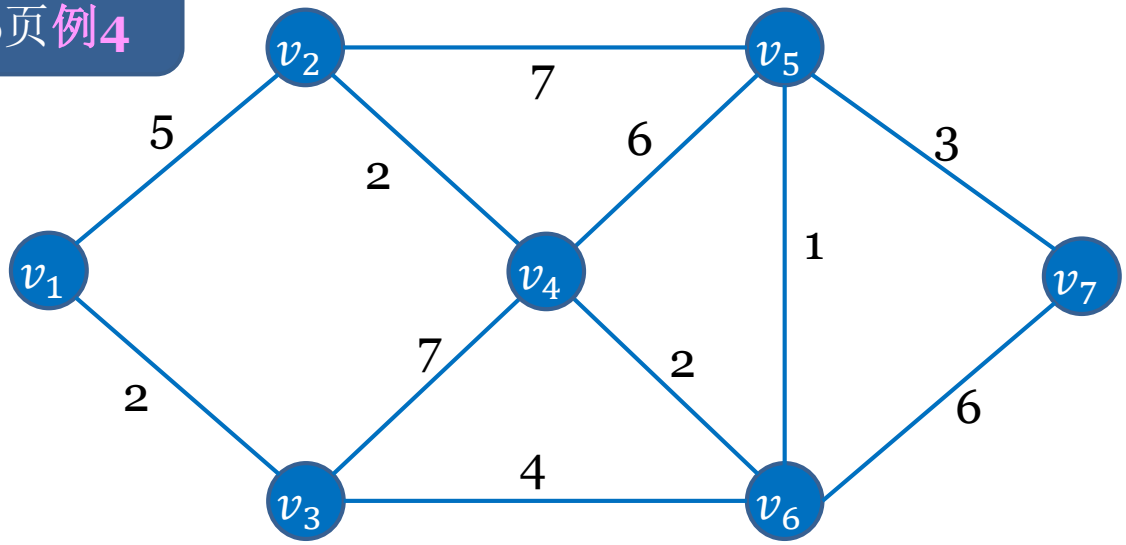
对于有**负权**的情形，则该式可能会失效。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 例4 用一般标号算法求下图中各顶点对之间的最短距离：

《运筹学基础及应用》（第6版） 176页例4



Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 解： 求解过程及结果见下列各表

$$k-1<\frac{\lg(7-1)}{\lg 2}\approx 2.6\le k$$
故 $k=3$

	$k=0$							$k=1$							$k=2$							$k=3$						
$D^{(k)}=(d^k(i,j))$	0	5	2	∞	∞	∞	∞	0	5	2	7	12	6	∞	0	5	2	7	7	6	10	0	5	2	7	7	6	10
	5	0	∞	2	7	∞	∞	5	0	7	2	7	4	10	5	0	7	2	5	4	8	5	0	7	2	5	4	8
	2	∞	0	7	∞	4	∞	2	7	0	6	5	4	10	2	7	0	6	5	4	8	2	7	0	6	5	4	8
	∞	2	7	0	6	2	∞	7	2	6	0	3	2	8	7	2	6	0	3	2	6	7	2	6	0	3	2	6
	∞	7	∞	6	0	1	3	12	7	5	3	0	1	3	7	5	5	3	0	1	3	7	5	5	3	0	1	3
	∞	∞	4	2	1	0	6	6	4	4	2	1	0	4	6	4	4	2	1	0	4	6	4	4	2	1	0	4
	∞	∞	∞	∞	3	6	0	∞	10	10	8	3	4	0	10	8	8	6	3	4	0	10	8	8	6	3	4	0

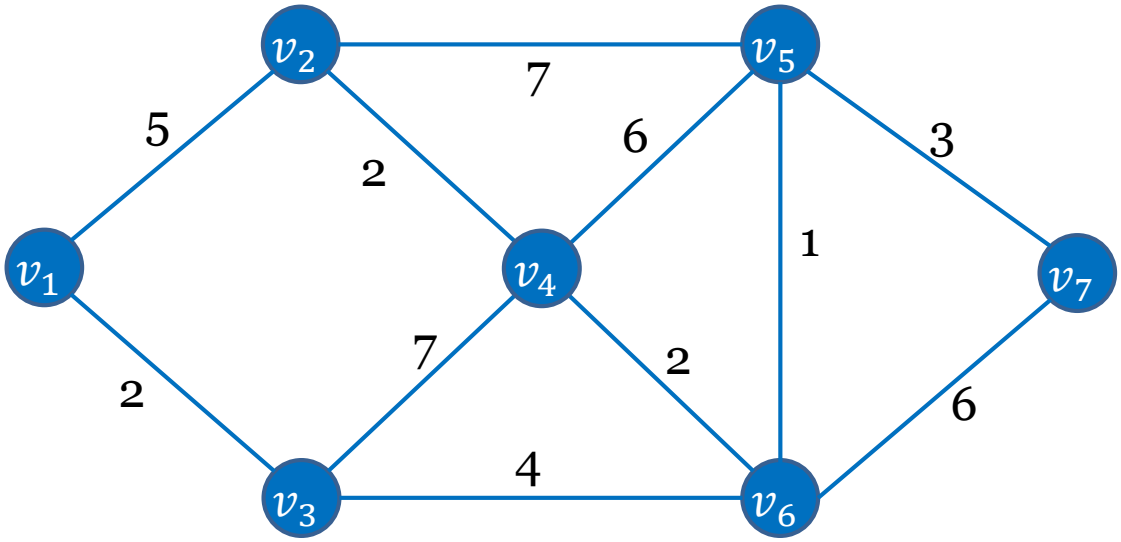
由于$D^{(3)}=D^{(2)}$，计算结束，得到距离矩阵$D^{(3)}=D=(d_{ij})_{n\times n}$，
其中各元素$d^3(i,j)$即为所给图中从顶点i到顶点j的最短距离d_{ij}。

通过对最终的距离矩阵进行“反向追踪”可找出各条最短路。
比如： v_3到v_7的最短路的长度为8，相应的最短路为(v_3,v_6,v_5,v_7)

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 例5 下图(同例4)中$v_1, v_2, v_3, v_4, v_5, v_6, v_7$为7个村子，决定要办一所联合小学，已知各村的小学生人数分别为30,40,25,20,50,60,60，问小学应建在哪个村子，使小学生上学走的总路程为最短？



最短路问题

- 一般标号算法(任意权, 全顶点对)
  - **解：**
    - 将**例4**中所得的最终距离矩阵$D^{(3)}$的第一行乘以v_1村的小学生人数30，则所得数字为假定小学建于 $v_1, v_2, v_3, v_4, v_5, v_6, v_7$各村时， v_1村所有小学生单程所走的最短路程长度。
    - 依此类推，可得假定小学建于各村时v_i (i = 2, … , 7)村所有小学生单程所走的最短路程长度，计算结果见下页表格。
    - 将表中各列元素之和置于最后一行，则得到假定小学建于v_i (i = 1, … , 7)村时，7个村子小学生累计的一次单程上学所走最短路程的长度。
    - 由计算结果可知，该小学应建在v_6村。

最短路问题

- 一般标号算法(任意权, 全顶点对)
  - 解(续):

小学建于下列各村时小学生上学所走最短路程长度							
	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	150	60	210	210	180	300
v_2	200	0	280	80	200	160	320
v_3	50	175	0	150	125	100	200
v_4	140	40	120	0	60	40	120
v_5	350	250	250	150	0	50	150
v_6	360	240	240	120	60	0	240
v_7	600	480	480	360	180	240	0
$\sum$	1,700	1,335	1,430	1,070	835	770	1,330

Haiyan Lu
Jiangnan University

1、注意与前面的“一般标号算法”的区别！
2、Floyd-Warshall算法也称为Floyd算法

最短路问题

- Floyd-Warshall算法(任意权, 全顶点对)
  - 算法原理
    - 设$D=(V,E,w)$为赋权**有向**图，$W=(w_{ij})_{p\times p}$为权值矩阵（若$(v_i,v_j)\notin A$，则添加虚拟弧(v_i,v_j), 并令$w_{ij}=\infty$），p为图D的顶点数。显然，**无向**赋权图的权值矩阵W为对称阵。
    - 设$V=\{1,2,\dots,n\}$，令$d^k(i,j)$表示从顶点i到顶点j的仅可以顶点$1,\dots,k$作为中间节点的最短路的长度。显然，$d^n(i,j)$表示该图中从顶点i到顶点j的**实际最短路**的长度，即
    $$d_{ij}=d^n(i,j)$$
    - 给定$d^{k-1}(i,j)$，算法计算过程中用到了如下**性质**：

$$d^k(i,j)=\min\{d^{k-1}(i,j),\ d^{k-1}(i,k)+d^{k-1}(k,j)\}$$

三角运算

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- Floyd-Warshall算法(任意权, 全顶点对)
  - 算法步骤
    - 首先计算直接距离矩阵

$$D^{(0)} = (d^0(i,j))_{n\times n} = W$$

    - 然后利用$D^{(0)}$，计算

$$D^{(1)} = (d^1(i,j))_{n\times n}$$

    - 重复上述过程，直至得到距离矩阵

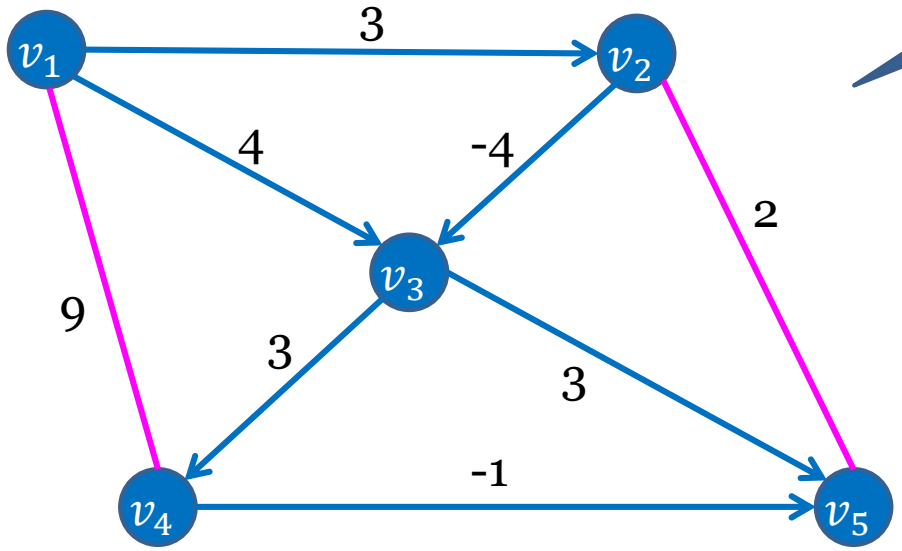
$$D = (d_{ij})_{n\times n} = D^{(n)} = (d^n(i,j))_{n\times n}$$

    算法结束。

Haiyan Lu 4/22/2020
Jiangnan University

最短路问题

- Floyd-Warshall算法(任意权, 全顶点对)
  - 例6 用 Floyd-Warshall 算法求下图中各顶点对之间的最短距离：



该图为混合图

Haiyan Lu
Jiangnan University

从$k=1$开始，$D^{(k-1)}$中的第k行和第k列称为关键行和列，把关键行和列中的元素复制到矩阵$D^{(k)}$的第k行和第k列中，$D^{(k)}$中的其余元素通过比较$D^{(k-1)}$中$d^{k-1}(i,j)$与其对应于关键行列中的两元素之和$d^{k-1}(i,k)+d^{k-1}(k,j)$得到（称为三角运算），取其中较小者作为$d^{k}(i,j)$。

最短路问题

- Floyd-Warshall算法(任意权, 全顶点对)
  - 解：用Floyd-Warshall算法求解的计算结果见下表:

$D^{(k)}=(d^k(i,j))$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
	0 3 4 9 ∞	0 3 4 9 ∞	0 3 -1 9 5	0 3 -1 2 2	0 3 -1 2 1	0 3 -1 2 1
	∞ 0 -4 ∞ 2	∞ 0 -4 ∞ 2	∞ 0 -4 ∞ 2	∞ 0 -4 -1 -1	8 0 -4 -1 -2	8 0 -4 -1 -2
	∞ ∞ 0 3 3	∞ ∞ 0 3 3	∞ ∞ 0 3 3	∞ ∞ 0 3 3	12 15 0 3 2	12 4 0 3 2
	9 ∞ ∞ 0 -1	9 12 13 0 -1	9 12 8 0 -1	9 12 8 0 -1	9 12 8 0 -1	9 1 -3 0 -1
	∞ 2 ∞ ∞ 0	∞ 2 ∞ ∞ 0	∞ 2 -2 ∞ 0	∞ 2 -2 1 0	10 2 -2 1 0	10 2 -2 1 0

$D^{(5)}=D$给出了图中各点之间的**最短路长**，并可由计算过程，通过从最终的**距离矩阵**进行“**反向追踪**”可找出最短路。

比如，从v_1到v_5的最短路长为1，相应的最短路为(v_1,v_2,v_3,v_4,v_5);

从v_5到v_1的最短路长为10，相应的最短路为(v_5,v_2,v_3,v_4,v_1).

可按照《运筹学教程》（第5版）240-241页给出的方法来记录最短路的信息。

课堂练习

- **判断题**

（1）若图中从v_1至各点均有唯一的最短路，则连接v_1至其他各点的最短路在去掉重复部分后，恰好构成该图的最小支撑树。 错

（2）求图的最小支撑树以及求图中一点至另一点的最短路问题，都可以归结为整数规划问题。 对

Thank you!

谢谢！