第 22 讲
扫雪问题的数学模型

□ 王树禾

AMCM-90 的问题 B，标题为“扫雪问题”，图 22-1 的地图上，实线表示马里兰州威考密科城中的马路，虚线是高速公路，一场雪后，从位于地图 * 标志的地点以西 4 英里的两车库派出两辆扫雪车，求用两辆扫雪车扫清马路上的雪之有效方法（假设扫雪车不会发生故障，也不停顿，交叉路口不需特别扫雪方法）.

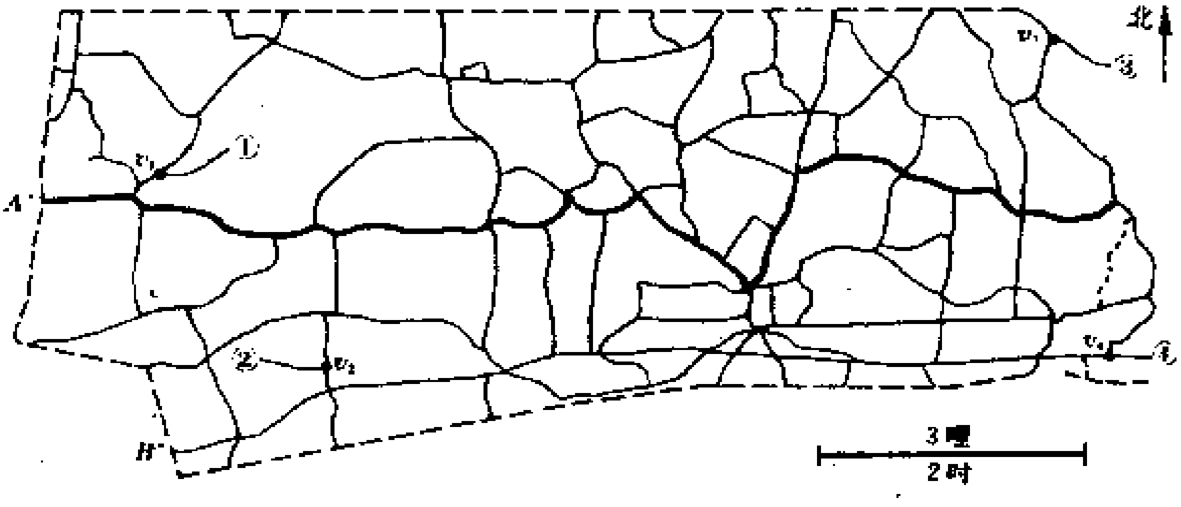

图 22-1

这个题目是该城道路局局长兼工程师 Kirk Banks 从该城实际扫雪工作中提出的.

下面我们对于单车道和双车道的两种道路情况，以及参加扫

雪的车辆是一辆、两辆还是多辆(3 辆以上)分别予以讨论,对于可解者,给出有效算法;对于难解者,证明其为 NPC 的.

§1 单车道单车扫雪

1. 地图是 Euler 图的情形

这种情况,只需求出地图相应的图(graph)上的一条 Euler 回路,且按寻找这条 Euler 回路时边的出现顺序来安排扫雪的工作顺序即可(设寻 Euler 回路时从高速公路上一路口开始,把路口(例如十字路口,丁字路口)视为图的顶点,地图上的道路段视为边).

在 Euler 图上求取 Euler 回路,可用下面的 Fleury 算法.

Fleury 算法:

(1) $\forall v_0 \in V(G)$,令 $W = v_0$.

(2) 设行迹 $W_i = v_0 e_1 v_1 e_2 \cdots e_i v_i$ 已选定,则从 $E(G) - \{e_1, e_2, \cdots, e_i\}$ 中选一边 e_{i+1},使得

(i) e_{i+1} 与 e_i 相邻(共端);(ii)除非已无选择余地,e_{i+1} 不要选 $G_i = G - \{e_1, \cdots, e_i\}$ 的桥.

(3) 直到(2)不能进行为止.

所谓桥,即连通图 G 中一条边 e,使得 $G-e$ 不连通.

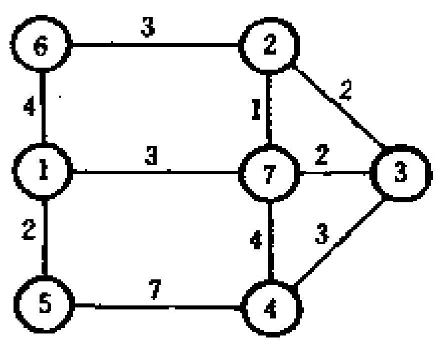

图 22-2

2. 地图不是 Euler 图的情形

这时,相应的图会有偶数个奇次顶,扫雪时某些边(路段)上要通过不止一次.如果一条边 e 需要走两次,就视为这里有两条有公共端点的它们的权(长度)

也一样的边. 设加入的新边集合为 E'，$G'(V,E\cup E')$是 Euler 图，我们是求一个最小的新边集 E'，即在保障$G'(V,E\cup E')$是 Euler 图的前提下，使得 $W(E_1)=\sum\limits_{e\in E_1} w(e)=\min$，其中 $w(e)$是边 e 之权(长度).

$$D=\begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array}\begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{pmatrix} 0 & 4 & 5 & 7 \\ 4 & 0 & 2 & 5 \\ 5 & 2 & 0 & 3 \\ 7 & 5 & 3 & 0 \end{pmatrix} \end{array}$$

下面我们用一个实例交代这种问题的解法. 设道路系统如图 22-2 所示，边旁写的是边权(长度). 记图 22-2 中的图为 G.

1) 求出 G 的奇次顶集 $X_1=\{①\ ②\ ③\ ④\}$.

2) 用 Dijkstra 算法求得 X_1 中各顶对之间的距离如矩阵 D 所示.

3) 构作加权完全图 $K(X_1,E_0)$见图 22-3，边(i,j)的权是矩阵 D 中(i,j)号元素.

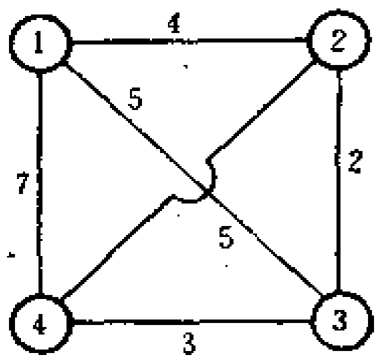

图 22-3

4) 求 $K(X_1,E_0)$的最佳匹配 M，得 $M=\{①②,③④\}$.

5) 用 Dijkstra 算法求 M 顶对间最短轨：

①与②间是①⑦②；③与④间是③④.

6) 在 5)中之最短轨上每边皆加上一个同权新边即得所求之 Euler 图 $G'(V,E\cup E')$，且 $W(E')=\min$.

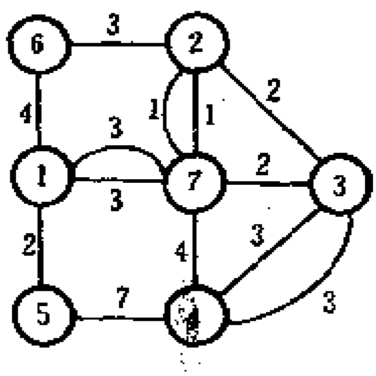

图 22-4 G'

7) 在G'上用 Fleury 算法求一条 Euler 回路即为解答：

⑥ ② ③ ⑦ ② ⑦ ① ⑦ ④ ③ ④ ⑤ ① ⑥ .

[illegible]

$i \leftarrow 0, v \leftarrow s$.

(2) $i \leftarrow i+1, k(v) \leftarrow i$.

(3) 若 v 无未用过的关联边，转(5).

(4) 选一条未用过的与 v 关联的边 $e=uv$，标志 e"用过了"；若 $k(u) \neq 0$，转(3)；否则($k(u)=0$)，$f(u) \leftarrow v, v \leftarrow u$，转(2).

(5) 若 $k(v)=1$，止.

(6) $v \leftarrow f(v)$，转(3).

其中 $k(v)$ 是顶 v 的编码，$f(v)$ 称为 v 的父亲，以父为尾以子为头的边叫作父子边.

上述算法的时间复杂度为 $O(|E(G)|)$.

图 22-5 所示的是 DFS 过程，容易证明，在 DFS 中每边恰通过两次，又返回出发点，且父子边们导出一棵生成树.

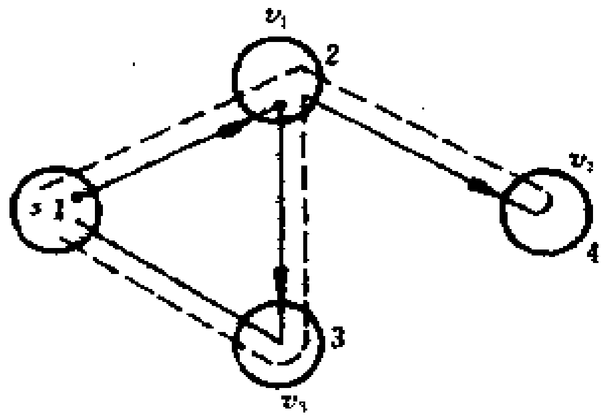

图 22-5

图 22-5 中箭头边是父子边，虚线是返回路线.

联系地图是未知的情形，这时扫雪车按右侧通行交通规则，依 DFS 算法的过程安排扫雪工作程序即可.

§3 双车道双车扫雪

若能把道路系统划分成两部分，使得两部分都是连通的，又第

一部分的总长度与第二部分的总长度相等，两辆扫雪车都在各自分得的那部分任务的一个顶点处，则变成双车道单车扫雪问题. 可惜对一般图(道路系统)，这并不总是可行的，例如道路是"星形"的，如图 22-6 所示.

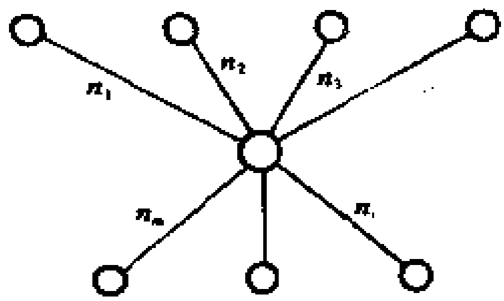

图 22-6

图 22-6 中任给了 m 条边，每边之权皆自然数，欲在此图上实现上述"等分"，这是 PART 问题，我们知道(见本书第 21 讲)，PART$\in$NPC. 所以一般而言，这个两车扫雪问题，既使是双车道行驶，也是 NPC 问题!

当然，这次竞赛题只是这一问题中的一个实例，根据道路的具体情况，我们可以把道路系统划分成"南片"与"北片"两个分别连通的"工作区"，且使两片的工作量(路长)相等，观察此具体形势，从 A 车出发点向东行，粗实线(图 22-1)以北归 A 车扫，以南归 B 车，粗实线路线也是 A 车的任务，经核算，若两者工作量不等，则在分界线(粗实线)附近做些调配，使其分别连通且工作量一致.

这种做法比较实惠，易于理解，但建立不了一般地可以复用的算法. 下面我们给出一种非一次性的对于所谓 k 边连通图皆适用的"k 等分算法".

所谓图 G 的边连通度为 k，即至少要删除 k 条边才能使一个连通图 G 变成不连通图，记成 $\kappa'(G)=k$.

在图 22-1 中，若暂不考虑①②③④四条街道，且认为高速公路上行驶时工作量为零，即与高速公路接头的街道的接头点是公共的，于是可以认为 A，B 两车从同一处出发，且道路系统中除①②③④外，再无这种边(街)，把它删除时则图不再连通，所以此图视为 2-边连通图，即 $\kappa'(G)\geqslant 2$.

定义 设 $G(V,E)$ 是连通加权图，每边 $e\in E(G)$，有权 $w(e)$

$\in \mathbf{N}$，且 $G_1,G_2,\cdots,G_n$ 是 G 的连通子图，又

(i) $v_0\in V(G_i)$，$i=1,2,\cdots,n$； (ii) $G=\bigcup_{i=1}^{n}G_i(n>1)$；

(iii) $E(G_i)\cap E(G_j)=\varnothing$，$i\neq j$； (iv) $W(G_i)=\frac{1}{n}W(G)$，

其中 $W(G)=\sum_{e\in E(G)}w(e)$，则称 $G(V,E)$ 被连通同根等分为 $G_1,G_2,\cdots,G_n$；v_0 称为 G_i 们的根，$G_1,\cdots,G_n$ 叫作同根 n 等分连通子图.

下面给出同根 n 等分连通子图的算法.

不妨设 $w(e)\equiv1$，不然加些新顶，把长为 n_0 的边分割成 n_0 条边，每边之权变成 1，且设 $W(G)=0(\bmod n)$；如果 $W(G)\neq0(\bmod n)$，则取原长度单位的 $\frac{1}{n}$ 为新的长度单位，又设 $\kappa'(G)\geqslant n\geqslant2$（$G$ 为 n 边连通图）.

算法 1：找到第一棵生成树 $T^{(1)}$.

(i) $\forall\ v_0v_1\in E(G)$，令 $T_1=G[v_0v_1]$，$i\leftarrow1$.

(ii) 若树 T_i 已取得，对于 $(V(T_i),\overline{V(T_i)})$ 中的每条边 e，在图 $G_i=G-(T_i+e)$ 中求

$$\min_{v\in V(G)-\{v_0\}} M_i(v_0,v)=\mu_i,$$

取使 $\mu_i\geqslant n-1$ 的边 e_i，若 $i<|V(G)|-1$，令 $T_i\leftarrow T_i+e_i$，$i\leftarrow i+1$，转(ii)；否则转(iii).

(iii) 止，记所得的树为 $T^{(1)}$.

其中 $\overline{V(T_i)}$ 表示 $V(G)-V(T_i)$，$(V(T_i),\overline{V(T_i)})$ 表示一端在 $V(T_i)$ 另端不在 $V(T_i)$ 的边们的集合，$M_i(v_0,v)$ 表示 G_i 中 v_0 与 v 之间的最小截量，即删除 M_i 条边可使 v_0 与 v 在 G_i 中不连通，但删除 M_i-1 条边仍有 v_0 与 v 连通. $M_i(v_0,v)$ 的计算有有效算法[1].

算法 2：找到两两无公共边的生成树 $T^{(1)}\cdots,T^{(n)}$.

(i) $G\leftarrow G-T^{(1)}$，$n\leftarrow n-1$，转算法 1，得 $T^{(2)}$.

(ii) 若 $T^{(1)},T^{(2)},\cdots,T^{(k)}$ 已得到，$k<n$，令 $n\leftarrow n-k$，$G\leftarrow G-\bigcup\limits_{i=1}^{k}T^{(i)}$，转算法 1，得，$T^{(k-1)}$；若 $k+1<n$，$k\leftarrow k+1$，转(ii)，否则转(iii).

(iii) 止，得树 $T^{(1)},T^{(2)},\cdots,T^{(n-1)},T^{(n)}$.

其中 $G-T^{(i)}$ 表示从 G 中删去 $T^{(i)}$ 之边.

算法 3：处理 $W(T^{(k)})>\frac{1}{n}W(G)$.

(i) $T\leftarrow T^{(k)}$，任取 T 的一个叶 $l\neq v_0$，若

$$W(T-l)>\frac{1}{n}W(G),$$

则 $T\leftarrow T-l$ 转(i).

(ii) 若 $W(T-l)=\frac{1}{n}W(G)$，把 $T-l=\hat{T}^{(k)}$ 染成 k 色，止.

我们用算法 3 把 $T^{(1)},T^{(2)},\cdots,T^{(n)}$ 中凡是 $W(T^{(\cdot)})>\frac{1}{n}W(G)$ 者皆"减轻"成 $\hat{T}^{(\cdot)}$，使得 $W(\hat{T}^{(\cdot)})=\frac{1}{n}W(G)$. 且把 $\hat{T}^{(\cdot)}$ 染成第$(\cdot)$色；再把其余的树 $T^{(:)}$ 染成第$(:)$色.

算法 4：处理 $W(T^{(m)})<\frac{1}{n}W(G)$.

(i) $\hat{T}\leftarrow T^{(m)}$，任取无色边 e，若

$$W(\hat{T}+e)<\frac{1}{n}W(G),$$

令 $\hat{T}\leftarrow\hat{T}+e$，把 e 染成 m 色，转(i).

(ii) 若 $W(\hat{T}+e)=\frac{1}{n}W(G)$，把 $\hat{T}+e=\hat{T}^{(m)}$ 的边都染成 m 色，止.

算法 1 到算法 4 的时间复杂度为 $O(|V||E|^2)$.

定理 1 算法 1 至算法 4 得出的 n 个不同色的子图即为 n 边连通图 G 的以 v_0 为根的同根连通 n 等分子图($n\geqslant 2$).

证 由于 $\kappa'(G)\geqslant n\geqslant 2$，故$\forall\ S\subset V(G)$，$v_0\in S$，$S\neq V(G)$时，

有$|(S,\overline{S})|\geqslant n$，算法1中$G[v_0v_1]$是一棵树$T$，且在$G-T$中$|(S,\overline{S})|\geqslant n-1$. 为证明定理，只需证下面的命题($\alpha$)：

(α)　$(V(T),\overline{V(T)})$中存在一条边e，使得$T+e$满足：$\forall S\subset V(G)$，$v_0\in S$，$S\neq V(G)$，则在$G-(T+e)$中，$|(S,\overline{S})|\geqslant n-1$；其中$T$是$G$上一个树子图.

若能证明(α)是真的，由递推得知，在G中有一生成树T_1，使得$G-T_1$中，$|(S,\overline{S})|\geqslant n-1$，进而在$G-T_1$中有生成树$T_2,\cdots$，在$G$中有两两边集交为空集的$n$棵生成树$T_1,T_2,\cdots,T_n$.

我们考虑这样的集合S：

(β)　$S\cup V(T)\neq V(G)$，在$G-T$中，$|(S,\overline{S})|=n-1$.

(1) 若(β)中的S不存在，$\forall e\in(V(T),\overline{V(T)})$，则$T+e$使命题($\alpha$)成立. 事实上，这时，若($\alpha$)不成立，则存在$S\subset V(T)$，$S\neq V(G)$，$v_0\in S$，但在$G-(T+e)$中，$|(S,\overline{S})|<n-1$，即在$G-T$中，$|(S,\overline{S})|<n$，$|(S,\overline{S})|\leqslant n-1$，又$G-T$中，$|(S,\overline{S})|\geqslant n-1$，故在$G-T$中，$|(S,\overline{S})|=n-1$，若继续证出$S\cup V(T)\neq V(G)$，则$S$满足($\beta$)，与满足($\beta$)之$S$不存在矛盾. 下面验证$S\cup V(G)\neq V(G)$. 记$e=uv$，$v\notin V(T)$，由于$G-(T+e)$中$|(S,\overline{S})|<n-1$，在$G-T$中$|(S,\overline{S})|=n-1$，故$v\notin S$，于是$S\cup V(T)\neq V(G)$.

(2) 若满足(β)的S存在，令φ是满足(β)的极大顶子集，不难看出，在$G-T$与G中皆成立

$$|(\varphi\cup V(T),\overline{\varphi\cup V(T)})|\geqslant n.$$

故在$G-T$中

$$|(\varphi\cup V(T),\overline{\varphi\cup V(T)})|>|(\varphi,\overline{\varphi})|.$$

上面不等式指出，$\exists\ e=uv\in E(G-T)$，使得

$$u\in\varphi\cup V(T),e\in(\varphi\cup V(T),\overline{\varphi\cup V(T)}),$$
$$e\notin(\varphi,\overline{\varphi}),u\in(\overline{\varphi}\cap V(T)),v\in\overline{V(T)}\cap\overline{\varphi}.$$

下面验证$T+e$满足命题(α).

(2.1)　若$e\notin(S,\overline{S})$，则在$G-T$与$G-(T+e)$中皆

$$|(S,\overline{S})|\geqslant n-1. \tag{γ}$$

(2.2) 若 $e\in(S,\overline{S})$，$e=uv$，$u\in S$，$v\in\overline{S}$，对 $\forall S_1\subset V(G)$，$\forall S_2\subset V(G)$，在 $G-T$ 中

$$|(S_1\cup S_2,\overline{S_1\cup S_2})|+|(S_1\cap S_2,\overline{S_1\cap S_2})|$$
$$\leqslant|(S_1,\overline{S}_1)|+|(S_2,\overline{S}_2)|. \tag{δ}$$

又在 $G-T$ 中，

$$|(\varphi,\overline{\varphi})|=n-1,|\varphi\cap S,\overline{\varphi\cap S}|\geqslant n-1. \tag{ε}$$

由$(\gamma)(\varepsilon)(\delta)$得知在 $G-T$ 中

$$|(S\cup\varphi,\overline{S\cup\varphi})|+|(\varphi\cap S,\overline{\varphi\cap S})|\leqslant|(S,\overline{S})|+|(\varphi,\overline{\varphi})|,$$
$$|(S\cup\varphi,\overline{S\cup\varphi})|\leqslant|(S,\overline{S})|+n-1-|(\varphi\cap S,\overline{\varphi\cap S})|$$
$$\leqslant|(S,\overline{S})|.$$

又 $u\in S$，$u\in\overline{\varphi}\cap V(T)$，得 $u\notin\varphi\cap S$，故 $u\notin\varphi$，S 不是 φ 的子集，于是

$$|(S\cup\varphi|>|\varphi|, \tag{η}$$

又 $v\in\overline{S}$，$v\in\overline{\varphi}$，$v\in\overline{V(T)}$，故

$$\varphi\cup S\cup V(T)\neq V(G).$$

(例如 $v\in V(G)$，但 $v\notin\varphi\cup S\cup V(T)$)，由 φ 之极大性，在 $G-T$ 中(由(η)，$S\cup\varphi$ 比 φ 大)有

$$|(S\cup\varphi,\overline{S\cup\varphi})|\geqslant n,$$

从而 $|(S,\overline{S})|\geqslant n$，故在 $G-(T+e)$ 中，$|(S,\overline{S}|\geqslant n-1$，命题$(\alpha)$成立，定理证毕.

具体到 AMCM-90*B* 扫雪问题，设删去图 22-1 中的四条"死胡同"①②③④后的图为 G，用算法 1 到算法 4 得到了 G 的两个二等分连通生成子图 G_1(北片)与 G_2(南片)，由于①与②长度一致，把①加到 G_1 上，把②加到 G_2 上；③比④略长一些，把③加到 G_1 上，把④加到 G_2 上，这样 G_1 的总长比 G_2 的大了一些，从 G_1 的边上割让一段给 G_2，即可使加上① ② ③ ④后，两车工作量一致. 在两车效率一致的条件下，全城扫雪完成得最早.

上面的讨论还可以给出在 n 边连通图上 n 车双车道扫雪时的

最早完成任务的解法.

实际生活中，还有一些实际问题与扫雪问题有一样的数学模型. 例如邮局有几位投递员同时去投递信件，全城每条街道都要投递，投递员完成任务后返回邮局，问如何分配投递路线，使得完成全城投递的时间最早，就是具有这种数学模型的问题.

§4 单车道双车扫雪

若道路系统是单行道，这时，既使所对应的图是 Euler 图，一般而言也是一个极难的问题. 欲使完工最早，要求每街通过恰一次，两车同时在某条道路上对头相遇或回到各自的出发点.

1. 要求每车回到各自的车库，且每边恰通过一次，同时完工

这个问题的数学模型是在 Euler 图 G 上寻求两个等长的闭行迹 C_1,C_2，使得 $E(C_1)\cup E(C_2)=E(G)$，且 $E(C_1)\cap E(C_2)=\varnothing$，$W(C_1)=W(C_2)$，我们仍设 $w(e)=1$，$\forall\ e\in E(G)$，我们称此问题为“Euler 二等分问题”.

定理 2 Euler 二等分问题是 NPC 的.

证 不难看出，在多项式时间内，可由不确定 Turing 机解决 Euler 二等分问题，所以这一问题是 NP 的.

又已知 PART$\in$NPC，只欠证任给 PART 的一个输入 I，在多项式 $Q(|I|)$时间内，可以把 I 映射成 $f(I)$，$f(I)$作为 Euler 二等分问题的一个输入，当且仅当 PART 对于 I 回答 yes 时，Euler 二等分问题对于 $f(I)$也回答 yes.

设 I 为$\{w_1,w_2,\cdots,w_t\}\subset\mathbf{N}$，取 $f(I)$为加权图 $G(V,E)$及 $v_0\in V(G)$：

$G=\bigcup_{i=1}^{\varepsilon} C_i, E(C_i)\cap E(C_j)=\varnothing, i\neq j; v_0\in V(C_i), i=1,2,\cdots,\varepsilon$,
且 $|E(C_i)|=w_i$，每边 e 权皆为 1，见图 22-7，其中 C_i 是圈.

对于 PART，I yes，即$\exists\ J\subset\{1,2,\cdots\varepsilon\}\subset N$，使得 $\sum_{i\in J} w_i=\sum_{i\notin J} w_i$；这时取

$$C_{甲}=\bigcup_{i\in J} C_i, C_{乙}=\bigcup_{i\notin J} C_i$$

于是 I yes 则知 $f(I)$ 也 yes.

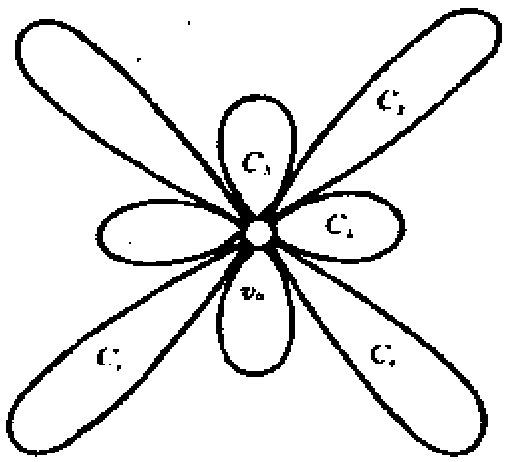

图 22-7

反之，若对于 Euler 二等分问题，$f(I)$ yes，即$\exists\ J\subset\{1,2,\cdots,\varepsilon\}$，使得

$$W(\bigcup_{i\in J} C_i)=W(\bigcup_{i\notin J} C_i).$$

这时得

$$\sum_{i\in J} w_i=\sum_{i\notin J} w_i,$$

即对于 PART，I 也 yes，由 NP 完备性，得知 Euler 二等分问题属于 NPC，证毕.

相似地可以提出 Euler n 等分问题，$n\in \mathbf{N}$，且可以证明 Euler 等分问题也是 NPC 的.

可见对于要求二车同时完工，每街恰扫一次，且返回各自车库的问题，除一些特殊情形可以利用道路的特点加以一次性地解决外，是尚无有效算法的，由于是 NPC 的问题，使得这种问题的难度十分之大.

2. 要求每边恰通过一次，同时完工，但未必回到各自车库.

这个问题的数学模型是在图 G 上求取两个等长行迹 P_1,P_2，使得 $E(P_1)\cup E(P_2)=E(G)$，$E(P_1)\cap E(P_2)=\varnothing$，且 $W(P_1)=$

$W(P_2)=\frac{1}{2}W(G)$，这个问题可以类似地证明也是 NPC 的.

3. 不要求同时完工，但要求最后一个完工者最早完工，要求每街恰扫一次

这个问题的数学模型是在连通加权图 $G(V,E)$ 上求取两个行迹 P_1,P_2，使得

$E(P_1)\cup E(P_2)=E(G)$，$E(P_1)\cap E(P_2)=\varnothing$，且

$$\max_{i=1,2}\sum_{e\in E(P_i)} w(e)=\min.$$

这个问题可以证明也是 NPC 的.

参 考 文 献

[1] 王树禾，图论及其算法，中国科学技术大学出版社，1990.

[2] B. Andrasfai，图论导引，郭照人译，高等教育出版社，1985.

[3] Edmonds，J. and Johnson E. L.，Matching，*Euler tours and the Chinese Postman*，Math·Programming，5，88－124，1973.

[4] C. Berge，*Graphs*，North-Holland，1985.