

有向树图的最小 K 点连通扩充

孙雨耕, 吕 航, 郭培生, 吴 雪

(天津大学电气与自动化工程学院, 天津 300072)

摘 要: 本文解决了图论的连通性理论中的一个重要的问题))) 以最小边集扩充一个任意有向树图为 K 点连通图, 证明了该问题在算法上属于 P 问题, 提出了一个时间复杂度为 $O(|V|^3)$ 的有效算法 DIKA, 该算法为可靠通讯网的计算机辅助设计提供了一个基础.

关键词: 有向树图; 最小扩充; K 点连通

中图分类号: TN915

文献标识码: A

文章编号: 0372-2112 (2004) 02-0202-05

The Augmentation Problem of a Directed Tree Graph to a K2Connected Graph

SUN Yu2geng, LV Hang, GUO Pe2sheng, WU Xue

(School of Electrical and Automation Engineering, Tianjin University, Tianjin 300072, China)

Abstract: This paper studies an important problem in Connectivity Theory))) augmenting a directed tree graph to a K2Vertex2Connected graph by adding a minimum set of edges and proves that this problem is P2Problem. An effective algorithm DIKA with complexity of is given. This algorithm contributes a foundation to the Computer Aided Design of reliable communication networks.

Key words: directed tree; minimum augmentation; K2connectivity

1 引言

图的扩充问题在连通性理论中具有非常重要的作用. 1976年, K P Eswaran^[4]证明了任意图的加权 K 点(边)连通扩充问题属于 NP) 完全问题, 并证明任意不加权的无向图的 2 边连通扩充是 P 类问题, 给出了有效算法. 其后, 不加权任意图的 K 点(边)连通扩充是否为 NP) 完全问题, 以及如果是 P 类问题怎样给出其有效算法, 一直是图论、计算数学、离散数学界关注和研究的热点, 除了它在理论上的重要意义外, 还与可靠网络(如通信网)的抗毁性密切相关, 因而也有重要的应用价值. Y S Kajitani, 蔡国瑞和孙雨耕相继解决了任意无向不加权树图的 K 边连通扩充问题和任意无向不加权图的 K 边连通扩充问题^[1].

不加权 K 点连通扩充问题被公认为是比 K 边连通最小扩充难度更大的一类问题, 其关键在于如何建立最优判据以及确定问题的属性, 目前国际上已经做到的最好的结果是已知 K 点连通图, 求其 K+1 点最优连通扩充^[5,6].

本文研究了有向树图的最小 K 点连通扩充问题, 提出了最优判据和有效算法 DIKA, 该算法采用分治策略, 先将树图分解成一系列的子树, 且每个子树最多包含一个内点, 然后把各子树分别扩充成 K 点连通的子图, 再对扩充后的子图进行合并, 合并后所得的图即为有向树图的最小 K 点连通扩充图. 最后对算法的最优性进行了证明并分析算法的复杂度.

2 基本定义和定理

在有向图 $G = (V, E)$ 中, 对一条边 $e = (L, v) \in E(G)$, 称 $t(e)$ 为边 e 的起点或尾($t(e) = L$), $h(e)$ 为边 e 的终点或头($h(e) = v$). 对一个边集 $E_c \subseteq E(G)$, 称 $t(E_c) = \{t(e) | e \in E_c\}$ 为边集 E_c 的尾, $h(E_c) = \{h(e) | e \in E_c\}$ 为边集 E_c 的头. 对一个顶点 $v \in V(G)$, 称 $E^-(v; G) = \{e | h(e) = v\}$ 为以 v 为头的边集, $E^+(v; G) = \{e | t(e) = v\}$ 为以 v 为尾的边集, $\text{Fin}(v; G) = \{L | (L, v) \in E(G)\} = t(E^-(v; G))$ 为 v 的入邻点集, $\text{Fout}(v; G) = \{X | (v, X) \in E(G)\} = h(E^+(v; G))$ 为 v 的出邻点集, $d^+(v; G) = |\text{Fout}(v; G)|$ 为 v 的出度, $d^-(v; G) = |\text{Fin}(v; G)|$ 为 v 的入度, $d(v; G) = d^+(v; G) + d^-(v; G)$ 为 v 的度数. 当在树图中用点 a, ac 剖分某条边时, 并删除边 (a, ac) , 则树图变为两个独立的有向树, 并产生两个新的顶点, 它们是双生子, 为了与原来图中的顶点加以区别, 我们称它为 /Linker0, 并记 $a = \text{twin}(ac)$, $ac = \text{twin}(a)$. $|V|$ 为 G 的顶点数, $|E|$ 为 G 的边的个数.

对正整数 K , 定义 $\text{MK}(G)$ 为 G 中度数大于 K 的点的集合, $\text{NLK}(G)$ 为 G 中度数大于 K 的点的个数. $N(v; G)$ 为 G 中与顶点 v 相邻的点的集合, 简称顶点 v 的邻点集. $E(v, V(G); T)$ 定义为从点 v 到树图 T 的扩充图 G 的点集 $V(G)$ 的有向边的集合, $E(V(G), v; T)$ 定义为从树图 T 的扩充图 G 的点集 $V(G)$ 到点的有向边的集合.

图 G 中对不同的顶点 L 和 v , 称 $P_G(L, v)$ 为 G 中点不相交的 $L-v$ 路径集合. 用 $L_G(L, v)$ 记内部不相交的 $L-v$ 路的最大数目, 用 $C_G(L, v)$ 记最小 L, v 分离边集, 即 $G - C_G(L, v)$ 中没有 $L-v$ 路.

在有向图 G 中, 设 $v \in V(G), A \subset V(G)$, 且 $|A| = m$, 一个 (v, A) fanout 是这样 m 条路径的集合: (1) 在 (v, A) fanout 中, 任何两条不同路径除 v 外没有公共顶点; (2) 在 (v, A) fanout 中, 每条路径均是 $v-v_i$ 的路径, 其中 $v_i \in A$. 一个 (A, v) fanin 是这样的集合: (1) 在 (A, v) fanin 中, 任何两条不同路径除 v 外没有公共顶点; (2) (A, v) fanin 中, 每条路径均是 v_i-v 的路径, 其中 $v_i \in A$.

定理 1^[2] (Menger 定理) 一个 $|V| \geq k+1$ 的图 G 是 k 点连通的当且仅当 G 的任意两个相异顶点至少被 k 条内部不相交的路所连.

定义 1^[7] 对于正整数 n 和 $k (1 \leq k \leq n)$, 一个超连通链 $G_{n,k} = (V, E)$ 定义为: $V = \{0, 1, \dots, n-1\}$, $E = \bigcup_{i=1}^k E_i(G_{n,k})$, 其中 $E_i(G_{n,k}) = \{(j, j+i) \mid 0 \leq j \leq n-1-i, 1 \leq i \leq k\}$

如图 1 画出超连通链 $G_{8,3}$.

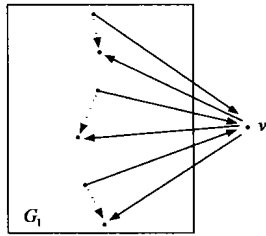
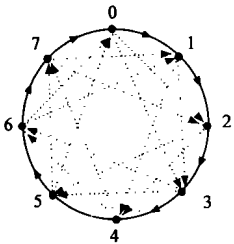


图 1 一个超连通链 $G_{8,3}$ 图 2 虚箭头属于 Ec_1 的边

定理 2^[7] 超连通链 $G_{n,k} (n > k)$ 是一个 k 正则的和 k 点连通的.

3 图的运算

定义两种图的运算: 加点运算 **demerge 1** 及合并运算 **demerge 2**.

定义 2 (demerge 1 运算)^[7] 设有向图 $G_1 = (V_1, E_1)$, 顶点 $v (v \in V_1)$ 和独立边集 $Ec_1 (Ec_1 \subset E_1)$, 则 $G = (V, E) = \text{demerge 1}(v, Ec_1, G_1)$ 定义为:

$$\begin{aligned} V &= V_1 \cup \{v\} \\ E &= (E_1 - Ec_1) \cup \{(t(e), v) \mid e \in Ec_1\} \\ &\quad \cup \{(v, h(e)) \mid e \in Ec_1\} \end{aligned}$$

如图 2 所示.

定义 3 (demerge 2 运算)^[7] 设 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ 是两个独立的有向图, 并有 $d^-(v_1; G_1) = d^+(v_2; G_2) = j$, $d^+(v_1; G_1) = d^-(v_2; G_2) = m$, 则 $G(V, E) = \text{demerge 2}(G_1, v_1, G_2, v_2)$ 定义为:

$$\begin{aligned} V &= V_1 \cup V_2 - \{v_1, v_2\} \\ E &= ((E_1 - Ec_1) \cup (E_1^+(v_1; G_1) \cup E_1^-(v_1; G_1) \\ &\quad \cup E_2^+(v_2; G_2) \cup E_2^-(v_2; G_2))) \\ &\quad \cup \{((Fin(v_1; G_1))_i, (Fout(v_2; G_2))_i) \mid 1 \leq i \leq j\} \end{aligned}$$

$$G \setminus \{((Fin(v_2; G_2))_i, (Fout(v_1; G_1))_i) \mid 1 \leq i \leq m\}$$

如图 3 所示.

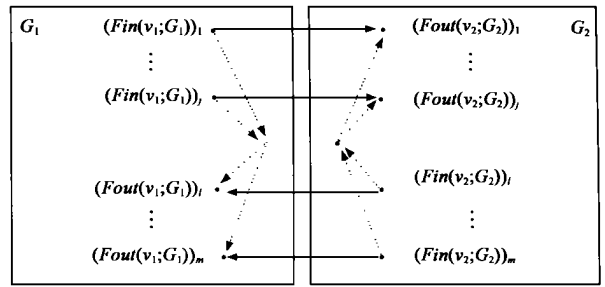


图 3 demerge 2 运算例图

虚箭头: $E_1^+(v_1; G_1) \cup E_1^-(v_1; G_1) \cup E_2^+(v_2; G_2) \cup E_2^-(v_2; G_2)$

实箭头: $\{((Fin(v_1; G_1))_i, (Fout(v_2; G_2))_i) \mid 1 \leq i \leq j\}$

$$G \setminus \{((Fin(v_2; G_2))_i, (Fout(v_1; G_1))_i) \mid 1 \leq i \leq m\}$$

引理 1^[7] 一有向图 $G = (V, E)$ 是 m 点连通的, 当且仅当 (1) $|V| \geq m+1$;

(2) 对任意顶点 v 和任意点集 $V' (A \subset V)$ 且 $|V'| = m$, 存在一个 (v, V') fanout.

引理 2^[7] 一有向图 $G = (V, E)$ 是 m 点连通的, 当且仅当 (1) $|V| \geq m+1$;

(2) 对任意顶点 v 和任意点集 $V' (A \subset V)$ 且 $|V'| = m$, 存在一个 (v, V') fanin.

定理 3 设有向图 $G_1 = (V_1, E_1)$, Ec_1 是 G_1 中的独立边集, 且 $|Ec_1| = K$, v 是任意不属于 V_1 的顶点, 若 G_1 是 K 点连通图, 则 $G = (V, E) = \text{demerge 1}(v, Ec_1, G_1)$ 也是 K 点连通的.

证明 根据定理 1, 只要 G 中任意不相邻的顶点 v_1, v_2 间至少存在 K 条点不相交的路径, 即 $|P_G(v_1, v_2)| \geq K$, 即可得证, 以下分三种情况证明:

(1) $v_1 = v$. 因为 v_1, v_2 是不相邻的, 即 $(v_1, v_2) \notin E$, 而 $h(Ec_1)$ 必与 v (即 v_1) 相邻, 所以, $v_2 \in h(Ec_1)$, 记 FO 是 $(v_1, h(Ec_1))$ fanout, 则 $FO = \{v_1, h(e) \mid e \in Ec_1\}$, 由于 G_1 是 K 点连通图, 根据引理 1, 存在一个 $(h(Ec_1), v_2)$ fanin, 记 $FI = \{h(e), v_2 \mid e \in Ec_1\}$, 又因 $v_2 \in h(Ec_1)$, 所以据 FI 不含 Ec_1 中得任何一条边, 因此 K 条点不相交的路径可由 FO 和 FI 构成, 于是 $|P_G(v_1, v_2)| \geq K$ 成立.

(2) $v_2 = v$. 证明与情况 (1) 类似.

(3) $v_1 \neq v$ 和 $v_2 \neq v$. 记 $P_G(v_1, v_2) = \{L_i\}$, $Ec_1 = \{e_i\}, i = 1, 2, \dots, K$, 用归纳法证明, 对 Ec_1 分布于 $P_G(v_1, v_2)$ 上的路径条数 n 进行归纳, 若 $Ec_1 = \{e_i\}, i = 1, 2, \dots, K$ 分布在某一 L_j 上, 即 $n = 1$, 则 G 中有 K 条点不相交的路径, 其中 $K-1$ 条为: $L_i (i = 1, 2, \dots, K, i \neq j)$ 另外 1 条为 $\beta v_1, \dots, t(e_j), v, h(e_k), \dots, v_2$, 其中 $l(v_1, t(e_j)) = \min l(v_1, t(e_i)), l(v_1, t(e_k)) = \max l(v_1, t(e_i))$, 即按从 v_1 到各独立边的头点的距离远近对独立边进行编号. 显然, $n = 1$ 时结论成立, 如图 4 所示.

用归纳法:

假设 $n = q$ 时结论成立, 证明 $n = q+1$ 时的结论. 设 L_j 上有 Ec_1 中的 m 条边 $\{e_1, e_2, \dots, e_m\}$, 其中 $(1 \leq m \leq K-q)$, 且 $l(v_1, t(e_1)) = \min l(v_1, t(e_i)), l(v_1, t(e_m)) = \max l(v_1, t(e_i))$.

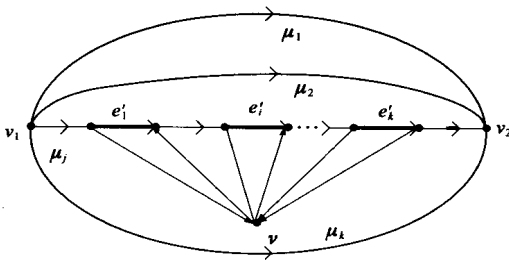


图 4 证明所用例图

($\mathfrak{e}_i$)), $1 \leq i \leq m$; 首先将分布在 $\{L_1, \dots, L_{j-1}, L_{j+1}, \dots, L_{q+1}\}$ 上的边集 $E_{c1} = \{\mathfrak{e}_1, \mathfrak{e}_2, \dots, \mathfrak{e}_m\}$ 做 demerge1 运算, 所得图记为 G_1 , 由归纳法假设, G_1 存在 K 条 $v_1) v_2$ 点独立的路径, 且除 L_j 外, 其余 $K-1$ 条路径最多有一条过 v 点, 然后再对 $\{\mathfrak{e}_1, \mathfrak{e}_2, \dots, \mathfrak{e}_m\}$ 做 demerge1 运算, 得图 G .

(1) 若 G_1 中没有过 v 点的 $v_1) v_2$ 路径, 显然, 对 $\{\mathfrak{e}_1, \mathfrak{e}_2, \dots, \mathfrak{e}_m\}$ 做 demerge1 运算后, 有 $|P_G(v_1, v_2)| \leq K$ 成立.

(2) G_1 中存在一条过 v 点的 $v_1) v_2$ 路径, 即除了 L_j 外至少还有 $K-2$ 条独立且不过 v 点的 $v_1) v_2$ 路径, 对 $\{\mathfrak{e}_1, \mathfrak{e}_2, \dots, \mathfrak{e}_m\}$ 做 demerge1 运算后得到图 G , 则 G 中仍至少存在 $K-2$ 条不过 v 点的点独立的 $v_1) v_2$ 路径, 记为 $P_{G_1}^{K-2}(v_1, v_2)$. 因为 G_1 是 K 点连通的, 所以 $|P_{G_1}(t(\mathfrak{e}_1), h(\mathfrak{e}_m))| \leq K$, 那么 $|P_{G_1}(t(\mathfrak{e}_1), h(\mathfrak{e}_m)) - 3t(\mathfrak{e}_1) \mathfrak{e}_1 h(\mathfrak{e}_1), \dots, t(\mathfrak{e}_i) \mathfrak{e}_i h(\mathfrak{e}_i), \dots, t(\mathfrak{e}_m) \mathfrak{e}_m h(\mathfrak{e}_m)| \leq K-1$, 因此, 对 E_{c1} 进行 demerge1 运算后, G 中至少有 $K-1$ 条点独立的且不过 v 点的路径 $t(\mathfrak{e}_1)) h(\mathfrak{e}_m)$, 记为 $P_G^{K-1}(t(\mathfrak{e}_1), h(\mathfrak{e}_m))$. 又考虑极限情况, 当 $P_G^{K-2}(v_1, v_2)$ 与 $P_G^{K-1}(t(\mathfrak{e}_1), h(\mathfrak{e}_m))$ 全部相交时至少有 $K-1$ 个相交部分 (记为 $A_i, i=1, 2, \dots, K-1$), 且这 $K-1$ 个相交部分之间必互不相交, 即 $A_i \cap A_j = \emptyset, i \neq j$, 否则将不能保证 $|P_{G_1}(t(\mathfrak{e}_1), h(\mathfrak{e}_m)) - 3t(\mathfrak{e}_1) \mathfrak{e}_1 h(\mathfrak{e}_1), \dots, t(\mathfrak{e}_i) \mathfrak{e}_i h(\mathfrak{e}_i), \dots, t(\mathfrak{e}_m) \mathfrak{e}_m h(\mathfrak{e}_m)| \leq K-1$, 因此, $|P_G(v_1, v_2)| \leq K$, 其中至少有 $K-1$ 条分别经过 $A_i (i=1, 2, \dots, K-1)$ 的点不相交的路径; 另一条 $v_1) v_2$ 路径为 $3v_1, \dots, t(\mathfrak{e}_1) v h(\mathfrak{e}_m), \dots, v_2$.

证毕

定理 4^[7] 设有向图 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ 是独立的, 且均为 K 点连通图, $K \geq 2, v_1 \in V_1, v_2 \in V_2$, 且有 $d^-(v_1; G_1) = d^+(v_2; G_2) = d^-(v_1; G_1) = d^+(v_2; G_2) = K$, 那么, $G = \text{demerge2}(G_1, v_1, G_2, v_2)$ 是 K 点连通的.

4 最优判据

下面定义中都满足 $T < G$.

定义扩充入度为:

$$5^-(v; G) = d^-(v; G) - d^-(v; T), \forall v \in V(T)$$

$$5^-(G) = \sum_{v \in V(T)} 5^-(v; G)$$

定义扩充出度为:

$$5^+(v; G) = d^+(v; G) - d^+(v; T), \forall v \in V(T)$$

$$5^+(G) = \sum_{v \in V(T)} 5^+(v; G)$$

定理 5 (最优判据) 如果一个 K 点连通扩充图 $G = (V,$

$E + E_c)$ 是由树图 $T = (V, E)$ 通过添加边集 E_c 而成, 那么必有

$$|E_c| = \max \left(\sum_{v \in V} \max(K - d^-(v; T), 0), \right.$$

$$\left. \sum_{v \in V} \max(K - d^+(v; T), 0) \right) \quad (1)$$

成立, 反之, 对于上述 T , 可构造 E_c , 其边数满足式 (1) 要求, 使得 $G = (V, E + E_c)$ 为 K 点连通.

证明:

设 $G = (V, E + E_c)$ 是由 $T = (V, E)$ 通过添加边集 E_c 后得到的 K 点连通扩充图, 则对每一个顶点 v 都满足:

$$d^-(v; G) \geq \max(d^-(v; T), K)$$

$$d^+(v; G) \geq \max(d^+(v; T), K)$$

则

$$5^-(v; G) \geq \max\{0, K - d^-(v; T)\},$$

$$5^+(v; G) \geq \max\{0, K - d^+(v; T)\}$$

那么,

$$5^-(G) \geq \sum_{v \in V} \max\{0, K - d^-(v; T)\}$$

$$5^+(G) \geq \sum_{v \in V} \max\{0, K - d^+(v; T)\}$$

而每一条扩充边或者是一条入边或者是一条出边, 因此有:

$$|E_c| \geq 5^-(G) \text{ 和 } |E_c| \geq 5^+(G)$$

那么,

$$|E_c| \geq \max \left(\sum_{v \in V} \max(K - d^-(v; T), 0), \right.$$

$$\left. \sum_{v \in V} \max(K - d^+(v; T), 0) \right)$$

如果等式成立, 则 $|E_c|$ 必然最小, 即 G 为最小扩充图.

我们将提出一个算法, 构造一个扩充图 $G = (V, E + E_c)$, 使得 G 为 K 点连通, 且使得

$$|E_c| = \max \left(\sum_{v \in V} \max(K - d^-(v; T), 0), \right.$$

$$\left. \sum_{v \in V} \max(K - d^+(v; T), 0) \right)$$

成立.

5 扩充算法 DTKA

将有向树图 $T_0 = (V, E)$ 最小的扩充为 K 点连通图 $G = (V, E + E_c)$ 的算法 DTKA 由 3 部分组成: 首先将有向树图分解成一系列具有特殊性质的子图; 然后将每个子图按超连通链的构造方法扩充为 K 正则的子图; 最后将所有的子图和并成为一个图 G .

5.1 扩充有向图的分解

Procedure split(T_0)

首先建立一个子图库 SUBGRAPH0 和一个堆栈 STACK, 算法开始时均为空, 在运行的过程中, 所有的 $NLK(T) \geq 1$ 的子图均存放到 SUBGRAPH0 中, 其它的子图均放到 STACK 中等待进一步的分解.

BEGIN

(1) push T_0 to STACK;

(2) pop (STACK to T);

(3) 如果 $NLK(T) \neq 1$, 存放 T 于 $SUBGRAPH\ 0$ 并转 (6), 否则转 (4);

(4) 从 $NLK(T)$ 中选择两个顶点 V_1, V_2 , 如果 $e(V_1, V_2) \in E$ 或 $e(V_2, V_1) \in E$, 用点 a, ac 剖分该条边, 并删除边 (a, ac) ; 否则, 寻找 T 中的 $V_1 - V_2$ 路, 如果存在 $V_1 - V_2$ 路, 比如 $p = 3 V_1, u, \dots, V_2$, 用点 a, ac 剖分 $(V_1 - u)$, 并删除边 (a, ac) , 否则不考虑有向树的方向, 变为其基础图 T_c , 重新寻找 $V_1 - V_2$ 路, 比如 $p = 3 V_1, u, \dots, V_2$, 用点 a, ac 剖分 $(V_1 - u)$, 并删除边 (a, ac) , 并重新变为有向树, 这样树 T 被分解成为两个子树 T_a, T_{ac} ;

(5) push T_0 于 $STACK$, set T_{ac} 于 T 转 (3);

(6) 如果 $STACK$ 非空, 转 (2);

(7) 返回.

END

5.1.2 各子树的扩充

5.1.2.1 子树的顶点标号 任一子树 T_i 中, 首先假设 $NLK(T_i) = u$, 称 u 为内点. 若 u 的度 $d^-(u; T_i) \neq K$ 且 $d^+(u; T_i) \neq K$ 则按如下顶点次序标号:

$$f(v) = \begin{cases} [0, \dots, d^-(u; T_i) - 1], v \in \text{Fin}(u; T_i) \\ d^-(u; T_i), v = u \\ [(f(u) + 1), \dots, (f(u) + d^+(u; T_i))], v \in \text{Fout}(u; T_i) \end{cases}$$

若 $d^-(u; T_i) > K$ 或 $d^+(u; T_i) > K$, 则

(1) 删除 $d^-(u; T_i) - K$ 或 $(和) d^+(u; T_i) - K$ 条边, 并用 Ed_i 表示删除的边集, 删除后出现的孤立点集用 Vd_i 表示;

(2) 对 $V(T_i) - Vd_i$ 中的各点按 $f(v)$ 标号, 对 Vd_i , $f(v)$ 定义为: $[|V(T_i) - Vd_i|, \dots, |V(T_i) - 1|] v \in Vd_i$.

5.1.2.2 子树的扩充 在这一过程, 扩充 $SUBGRAPH\ 0$ 中的每一子树, 然后将扩充图存放在另一个图库中 $SUBGRAPH\ 1$. 算法如下:

Procedure AUGMWT(T)

BEGIN

(1) 如果 $NLK(T) = 1$ 转 (3), 否则转 (2);

(2) 选择 $K + 1$ 个相邻的点集 $A = \{V_1, V_2, \dots, V_{K+1}, A \cap V(T)\}$. 将 $G(A)$ 扩充成一个有向完全图, 转 (5);

(3) 设度大于或等于 K 的顶点为 NK , 令 $A = \{NK\} \cap G \cap \{NK; T\}$, 按照超连通链的构造方法, 将 A 中的顶点标号;

(4) 将 $G(A)$ 构造成一个超连通链;

(5) 如果 $|A| = |V(T)|$, 转 (6); 否则对 $v \in V(T) - V(G)$ 的各点做:

° 若 $E(v, V(G); T) = 1$ 或 $E(V(G), v; T) = 1$, 即 v 与 $u \in V(G)$ 在 T 中有一条边 e 存在, 则选择 K 条独立边 Ec_1 , 使 $u = t(e)$ 时, Ec_1 中含有 $E^+(u, G)$ 中的一条扩充边, 或当 $u = h(e)$ 时, Ec_1 中含有 $E^-(u, G)$ 中的一条扩充边, 按 $demerge1$ 运算, 将 v 合并到 $G(A)$ 中, 使 $G \leftarrow demerge1(v, Ec_1, G(A))$ 转 (5);

° 否则转 (5);

(6) $SUBGRAPH\ 1 \leftarrow G$;

(7) 返回.

END

每个子图经过扩充后, 所形成的图也是 K 正则图.

5.1.3 子图的合并

经过扩充阶段, 每一个子图均扩充到 K 点连通, 这样, 两个相邻的子图 G_1 和 G_2 就可以通过合并运算合并成 G .

Procedure MERGE

BEGIN

(1) 任选 $SUBGRAPH\ 1$ 中的一个子图作为操作图 G , 将 G 从 $SUBGRAPH\ 1$ 中删除;

(2) 如果 $SUBGRAPH\ 1$ 为空, 则转 (6);

(3) 选择一个 $Linker$ 顶点 $v \in G$, 找到 $SUBGRAPH\ 1$ 中的子图 G_c , 使得 $vc = twin(v) \in G_c$;

(4) $G \leftarrow wdemerge2(G, v, G_c, vc)$;

(5) $G \leftarrow G^*$, 从 $SUBGRAPH\ 1$ 中删除 G_c , 转 (2);

(6) 将删除的边集 $Ed = Ed_1 \cup G \cup Ed_2 \cup G$, $G \cup Ed_r$ 再恢复到 G 中, 则加入边集 Vd_i 入度或出度将增加;

(7) 找出各 Ed_i 对应的 Vd_i , 检查各 Vd_i 中各点的入度和出度, 将入度大于 K 的点重新放入一个点集 V_{λ} 中, 将出度大于 K 的点重新放入一个点集 V_{μ} 中.

(8) 如果 V_{λ} 和 V_{μ} 都不为 $\emptyset$, 即只要都有点存在, 则从 V_{λ} 中任取一点 v_1 , 找出一条满足 $v_1 = h(e_1)$ 的扩充边 e_1 所对应的另一点 x , 再从 V_{μ} 中任取一点 v_2 , 找出一条满足 $v_2 = t(e_2)$ 的扩充边 e_2 所对应的另一点 y , 删除 e_1 和 e_2 , 连接 x 与 y , 构成边 e 使得 $x = t(e)$, $y = h(e)$, 从 V_{λ} 中删除 v_1 , 从 V_{μ} 中删除 v_2 , 然后转 (8).

END

算法结束, G 即为 T_0 的最小 K 点连通扩充图.

6 算法的最优性证明及时间复杂度估计

定理 6 由算法 DTKA 得到的图 G 是 T 的最小连通扩充图.

证明 首先将证明 G 是 K 点连通的.

算法 DTKA 由分解、扩充、合并三阶段组成. 树图 T 分解成各个子树, 每个子树按照超连通链的方法进行扩充, 超连通链是一个 K 正则的和 K 点连通的图; 在合并阶段, 根据定理 4 可知, 最后的图 G 也是 K 点连通扩充图.

下面证明 $|E(G) - E(T)|$ 最小. 在扩充阶段, 首先要对每个子树重新标号, 主要是将入度或(和)出度大于 K 的内点通过删除边集变为入度或(和)出度等于 K 的内点, 设因内点入度大于 K 而删除的边集的元素个数为 m_1 , 因内点出度大于 K 而删除的边集的元素个数为 m_2 , 那么, 到合并阶段第 (6) 步以前, 得到一个 K 点连通的正则图, 此时, 扩充边数为:

$$\begin{aligned} |Ed| &= \sum_{v \in V} \max(K - d^-(v; T), 0) + m_2 \\ &= \sum_{v \in V} \max(K - d^+(v; T), 0) + m_1 \end{aligned}$$

因为在标号时, 如果某些点的入度大于 K 时, 需要删除树图的入边, 所以 $d^-(v; T)$ 的值应该减小, 而上面公式中没有改变 $d^-(v; T)$ 的值, 所以计算扩充边数时, 应该加上 m_1 , 同理, 对于出边也是如此.

第(6)步加入删除边集后, 以前由于内点入度大于 K 而删除边集所形成的孤立节点, 其出度就要增加, 而起先由于内点出度大于 K 而删除边集所形成的孤立点, 其入度就要增加, 但是, 加入的删除边集是树图本身就有的, 扩充边数仍为

$$\begin{aligned} |Ed| &= \sum_{v \in V} \max(K - d^-(v; T), 0) + m_2 \\ &= \sum_{v \in V} \max(K - d^+(v; T), 0) + m_1 \end{aligned}$$

通过第(7)、(8)步后, 进入孤点 v_1 的扩充边与出孤点 v_2 的扩充边合并为一条扩充边, 下面分三种情况讨论:

(1) $m_1 = m_2$, 扩充边, $|Ec| = |Ed| - m_1 = |Ed| - m_2$, 则

$$\begin{aligned} |Ec| &= \sum_{v \in V} \max(K - d^-(v; T), 0) \\ &= \sum_{v \in V} \max(K - d^+(v; T), 0) \end{aligned}$$

(2) $m_1 > m_2$, 扩充边 $|Ec| = |Ed| - m_2$, 则

$$\begin{aligned} |Ec| &= \sum_{v \in V} \max(K - d^-(v; T), 0) \\ &= \sum_{v \in V} \max(K - d^+(v; T), 0) + m_1 - m_2 \end{aligned}$$

说明:

$$\sum_{v \in V} \max(K - d^-(v; T), 0) > \sum_{v \in V} \max(K - d^+(v; T), 0)$$

(3) $m_1 < m_2$, 扩充边 $|Ec| = |Ed| - m_1$, 则

$$\begin{aligned} |Ec| &= \sum_{v \in V} \max(K - d^+(v; T), 0) \\ &= \sum_{v \in V} \max(K - d^-(v; T), 0) + m_2 - m_1 \end{aligned}$$

说明:

$$\sum_{v \in V} \max(K - d^+(v; T), 0) > \sum_{v \in V} \max(K - d^-(v; T), 0)$$

因此有:

$$|Ec| = |E_G - E| = \max\left(\sum_{v \in V} \max(K - d^-(v; T), 0), \sum_{v \in V} \max(K - d^+(v; T), 0)\right)$$

由定理 5, G 是 T 的最小 K 点连通扩充图。证毕

时间复杂度估计: 此算法时间复杂度为 $O(|V|^3)$ 。

算法分成三个阶段:

(1) 分解阶段 边分裂步骤(4)将要运行 $NK(T) - 1$ 次, 即不超过 $|V|$ 次, 找一条 $v_1 - v_2$ 路的时间复杂度为 $O(|V|^2)$, 所以子程序 SPLIT(T)的时间复杂度不超过 $O(|V|^3)$;

(2) 扩充阶段 由于分解阶段完成后, 至多有 $|V|$ 个子图, 此算法至多运行 $|V|$ 次, 从第(1)步到第(4)步, 主要目的是构造一个超连通链, 它的时间复杂度为 $O(|V|^2)$, 第(5)步加点运算, 此过程至多运行 $|V|$ 次, 而检查 v 点是否与超连通链之间有边相连, 需要 $O(|V|)$ 次, 因此加点运算时间也为 $O(|V|^2)$, 所以整个扩充阶段算法的时间复杂度不超过 $|V| \cdot (O(|V|) + O(|V|^2))$, 即为 $O(|V|^3)$ 为;

(3) 合并阶段 分解后的子图个数小于 $|V|$, 则扩充后的子图两两合并次数也将小于 $|V|$, 每个 demerge 1 运算也要运行 $O(|V|)$ 次, (6), (7), (8) 步是解决标号后称为孤立点的顶点间入度或出度大于 K 的情况, 至多需要运行 $O(|V|^2)$ 时间, 所以算法 MERGE 的复杂度不超过 $|V| \cdot (O(|V|) + O(|V|^2))$, 即 $O(|V|^3)$ 。

因此, 算法总的时间复杂度应为 $O(|V|^3)$ 。

限于篇幅, 本算法的例题图解没有列出, 感兴趣的可以与作者联系。

参考文献:

- [1] 蔡国瑞, 孙雨耕. Minimum augmentation of any graph to a K_2 -edge connected graph[J]. Networks, 1989, 19(1): 151-172.
- [2] B Bollobas. Extremal Graph Theory[M]. London: Academic Press, 1978.
- [3] Andras Frank. Augmenting graph to meet edge connectivity requirements[J]. SIAM J. DISC. Math, 1992, 5(1): 23-53.
- [4] K P Eswaran, R E Tarjan. Augmentation problem[J]. SIAM COMPUT, 1976, 5(4): 663-665.
- [5] Tsai Sheng Hsu, V Ramachandran. Finding a smallest augmentation to biconnect a graph[J]. SIAM COMPUT, 1993, 22(5): 889-912.
- [6] Tirot Jordan. On the optimal vertex connectivity augmentation[J]. Journal of Combination Theory, Series B 1995, 63: 8-20.
- [7] T Masuzawa. An optimal time algorithm for the K_2 -vertex connectivity unweighted augmentation problem for rooted directed trees[J]. Discrete Applied Math, 1987, 17: 67-105.

作者简介:



孙雨耕 男, 1940 年出生于吉林, 教授、博士生导师, 研究方向为: 图论与系统优化、电路理论、可靠性通信网络系统优化设计、通信网中的流量控制工程与基于无线公网的无线数据通信技术。



吕航 男, 1976 年出生湖北蕲春, 1998 年起于天津大学电气自动化与能源工程学院攻读硕博连读制博士学位, 研究方向为通信网(含计算机网络)的性能优化、流量工程与网络规划。
Email: lvhang@tju.edu.cn