HEWN 算法的复杂性分析——一点商榷意见*

韩爱丽, 杨志敏

(山东大学 威海分校 计算机科学系,山东 威海 264209)

E-mail: hanal@sdu.edu.cn

http://www.wh.sdu.edu.cn

摘要: 对最大团问题的 HEWN(hierarchical edge-weight network)算法进行复杂性分析.首先通过分析 HEWN 的结构特点和所需进行的操作,设计了一种实现 HEWN 算法的数据结构,指出了在 HEWN 算法中 HEWN 的存储宜采用邻接多重表和二叉链表相结合的链表表示法,然后从 HEWN 的存储结构入手,剖析了 HEWN 的构造过程,在剖析过程中,通过与 MCST(maximum complete sub-graph tree)比较,指出了当 $2j>n$ 时潜在的、指数的生成和修改 GM 的次数存在于 HEWN 算法中.因而,HEWN 算法的时间复杂度是指数的,而不是 $O(n^{8.5})$.

关 键 词: 算法复杂性; NP-完全性;团

中图法分类号: TP301 文献标识码: A

最大团问题已被证明是一个 NP-完全问题,如果最大团问题有多项式时间算法,则所有的 NP-完全问题都有多项式时间算法.对于一个没有自环和平行边的有限无向图 $G(V,E)$,最大团问题可形式化地描述为

Input:有限无向图 $G(V,E)$,正整数 $k(\le|V|)$;

Question:图 $G(V,E)$中是否存在团 C 满足$|C|\ge k$?

对于最大团问题,传统的 MCST 方法描述为:对于含 n 个顶点的图 $G(V,E)$,分层网络(HN)中的顶点按如下方式排列:考虑团的大小为 j,HN 有 j 层,每层上有按序号递增排列的 $n-j+1$ 个顶点.在 HN 的两个相邻层中,较高层中的第 1 个顶点的序号比较低层的多 1,并且较高层中的每个顶点与其左下方或下方的较低层中的所有顶点都有连接,如图 1 所示.沿着 HN 中从顶层到底层的连接,任意的 j-顶点团能被得到,所有可能的路径数为 C_n^j,每条路径上的顶点形成一个顶点集以判断它是否是一个团.为了扩展 HN 中的每条路径,采用树结构.以 HN 中顶层的每个顶点作为一棵树的根,仅当一个顶点与一条路径上的所有顶点都有连接时,它才能被加到这个路径上.相应于 j-层 HN,若任意一棵 j-层树被建成,则 j-顶点团存在.这样的 j-层树被称为最大完全子图树(maximum complete sub-graph tree,简称 MCST).显然,MCST 算法是非多项式的.

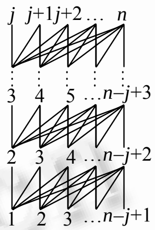

Fig.1 j-Level HN

图 1 j-层 HN

为了在多项式时间内求解最大团问题,一个分层的边权网络(hierarchical edge-weight network,简称 HEWN)在文献[1]中提出来了.它的主要思想是,在任一层次的 MCST 中,存在许多相同的分枝.为了将这些相同的分枝归并成 HN 中的一个单一分枝,每个团的顶点集被固定于 MCST 底层的相应边上,这样的顶点集称为 TL.依附于 MCST 中其他边的 TL 由以下方式决定:MCST 中上方单一分枝上的 TL 是它下方所有分枝上 TL 的并.现在,MCST 中的相同分枝被归并成 HN 中的一条单一边,而所有依附的顶点集也被归并成一个依附于该边的新顶点集,这就得到一个分层的边权网络(HEWN),依附于 HEWN 的每条边的新顶点集称为指导矩阵(GM).借助于

* 收稿日期: 2001-05-28; 修改日期: 2001-12-06

基金项目: 国家自然科学基金资助项目(69874001)

作者简介: 韩爱丽(1966－),女,山东乳山人,副教授,主要研究领域为智能算法,软件工程;杨志敏(1952－),男,黑龙江哈尔滨人,教授,主要研究领域为网络信息安全.

HEWN,最大团问题在多项式时间内解决[1].解最大团问题的 HEWN 算法可简洁地描述为:对于含 n 个顶点的图 $G(V,E)$,HEWN 算法首先试图在多项式时间内构造一个 n-层 HEWN,如果 n-层 HEWN 被成功地构造,则最大团的大小为 n,否则,它试图构造一个(n–1)-层 HEWN,以此类推.为了构造 x-层 HEWN,算法开始于 3-层 HEWN 并逐步增加 HEWN 的层[1].

借助于指导矩阵 GM,在多项式时间内,HEWN 能被分离成 MCST,而 MCST 也能被归并成 HEWN.因此,HEWN 算法的正确性可以得到保证.

本文从 HEWN 的存储结构入手,剖析了 HEWN 的构造过程.在此过程中,通过与 MCST 比较,指出了当 $2j>n$ 时,HEWN 算法是非多项式的.为了易于叙述,本文采用下列缩写: PHEWN(partial HEWN),即顶层仅一个顶点的 HEWN;IHEWN(intermediate HEWN),即构造 x-层 HEWN 时的中间的 HEWN;顶点 k 的 PHEWN,即以顶点 k 为顶层顶点的 PHEWN;顶点 k 的 MCST,即以顶点 k 为根的 MCST;TL(tree's label),即依附于 MCST 边的顶点集.

1 剖析 HEWN 算法

算法+数据结构=程序.要想在计算机中实现 HEWN 算法,需要先确定在计算机中如何表示 HEWN.数据结构在计算机中的表示又称为数据的存储结构.在 HEWN 算法中,指导矩阵 GM 的存储结构为数组,因为它能提供任意两个顶点之间直接的连接信息,而这些信息在 HEWN 算法中被频繁地使用.下面分析 HEWN 的存储结构.

1.1 HEWN的存储结构

在 HEWN 算法中,HEWN 的存储结构宜采用链表表示法.由于 HEWN 在每一次滚动之后均需对顶层顶点进行连接性检测,因而 HEWN 的存储结构只能采用层次结构的链表.在 HEWN 算法中,其顶点的最大度与最小度之间的差别很大,不宜用多重链表来表示,因为如果结点的结构按照最大度的顶点来设计,则大量的存储单元被浪费,如果结点的结构按照它自己的度来设计,则操作是很不方便的.根据HEWN的结构特点和所需进行的操作,HEWN 的存储宜采用邻接多重表和二叉链表相结合的链表表示法[2],即结点的结构与邻接多重表中的相同,而结点之间的连接与二叉链表中的相同,原因如下:

(1) 在 HEWN 算法中,对边的操作频繁地进行,如边的插入和删除、依附于各边的 GM 的并和交等.而在邻接多重表中,对应于每个顶点和每条边都存在相应的顶点结点和边结点.这样,顶点信息和边信息都能很容易地得到.

(2) 在 HEWN 算法中,HEWN 的顶层顶点被频繁地搜索,而其他顶点能沿着从顶层到底层的指针在边结点中得到.为了易于对边进行操作,边结点之间的连接宜采用二叉链表表示法.

根据上面的分析,结点的结构设计如下:

边结点:	*mark*	*ivertex*	*jvertex*	*info*	*jlink*	*ilink*
顶边结点:	*vertex*	*link*				

在边结点中,*ivertex*,*jvertex* 分别表示依附于该边的两个顶点,其中,*ivertex* 在较高层,而 *jvertex* 在较低层;*info* 表示依附于该边的指导矩阵 GM;*ilink* 指向依附于 *ivertex* 的下一条边;*jlink* 指向下一层中依附于 *jvertex* 的第 1 条边;*mark* 表示该边是否已被搜索过(在后面的讨论中 *mark* 被省略).在顶点结点中,*vertex* 表示 HEWN 的顶层顶点;*link* 指向连接到 *vertex* 的第 1 条边.这样,HEWN 的存储结构可形式化地描述为

```
CONST vtxnum=…;{vtxnum denotes the number of vertexes in a graph}
TYPE edglink=↑edgnode;
     edgnode=RECORD
               mark: 0..1;
               ivertex, jvertex: 1..vtxnum;
               info: ARRAY [1..vtxnum, 1..vtxnum] OF 0..1;
               ilink, jlink: edglink
             END;
     vtxnode = RECORD
```

```
            vertex: 1..vtxnum;
            link: edglink
        END;
    graphlist = ARRAY [1..vtxnum] OF vtxnode;
```

1.2 剖析HEWN的构造过程

以图 2 为例,14,13,12,11-层 HEWN 均不能被成功地构造.10-层 HN 如图 3 所示,下面剖析 10-层 HEWN 的构造过程.

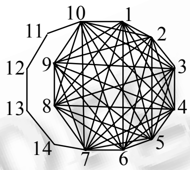

Fig.2 An example
图 2 一个例子

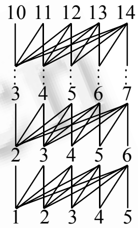

Fig.3 10-level HN
图 3 10-层 HN

Step1. 构造 3-层 PHEWN,并将它们组合成一个 3-层 IHEWN.

Step2. 构造基于 3-层 IHEWN 的 4-层 PHEWN,并将它们组合成一个 4-层 IHEWN.

(1) 构造顶点 4 的 4-层 PHEWN.

对于由 step1 构造的 3-层 IHEWN,经过 3 次分解-滚动-组合,得到一个 3-层 IHEWN,即路径 3-2-1.当顶点 4 被连接到这个路径的顶端时,$GM_{3\text{-}2}$ 和 $GM_{2\text{-}1}$ 被修改,而生成 $GM_{4\text{-}3}$.生成和修改 GM 的次数为 3,它等于顶点 4 的 4-层 MCST 中 TL 的数目.

(2) 构造顶点 5 的 4-层 PHEWN.

对于由 step1 构造的 3-层 IHEWN,经过 3 次分解-滚动-组合,得到一个如图 4 所示的 3-层 IHEWN.当顶点 5 被连接到该 3-层 IHEWN 顶层的两个顶点之后,得到一个 4-层 PHEWN,其存储结构如图 5 所示.

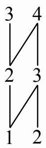

Fig.4
图 4

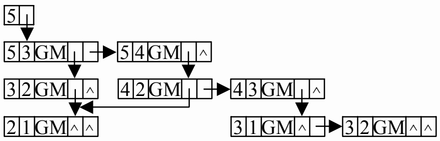

Fig.5 Storage of the 4-level PHEWN of vertex 5
图 5 顶点 5 的 4-层 PHEWN 的存储结构

根据图 5,不难理解,当顶点 5 被连接至图 4 顶层的两个顶点时,这两个边结点的插入只能依次进行,它们不能同时被插入到存储结构中.当插入边 5-3 的结点时,需生成 $GM_{5\text{-}3}$ 且修改 $GM_{3\text{-}2}$ 和 $GM_{2\text{-}1}$;当插入边 5-4 的结点时,需生成 $GM_{5\text{-}4}$ 且修改 $GM_{4\text{-}2}$, $GM_{2\text{-}1}$,$GM_{4\text{-}3}$,$GM_{3\text{-}1}$ 和 $GM_{3\text{-}2}$.注意:$GM_{2\text{-}1}$ 被修改了两次.这样,当构造顶点 5 的 4-层 PHEWN 时,生成和修改 GM 的次数为 (1+2)+(1+5)=9,它等于顶点 5 的 4-层 MCST 中 TL 的数目.

(3) 构造顶点 6 的 4-层 PHEWN.

对于由 Step1 构造的 3-层 IHEWN,经过 3 次的分解-滚动-组合,得到一个如图 6 所示的 3-层 IHEWN.当顶点 6 被连接至该 3-层 IHEWN 顶层的 3 个顶点后,得到一个 4-层 PHEWN,其存储结构如图 7 所示.

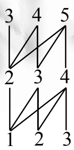

Fig.6
图 6

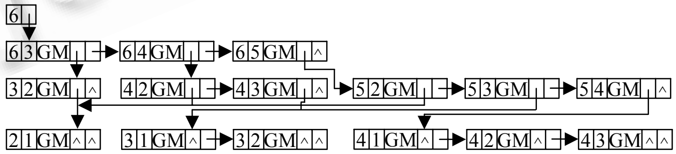

Fig.7 Storage of 4-level PHEWN of vertex 6
图 7 顶点 6 的 4-层 PHEWN 的存储结构

根据图 7 不难理解,当顶点 6 被连接至图 6 顶层的 3 个顶点时,这 3 个边结点的插入也只能依次进行.当插入边 6-3 的结点时,顶点 3 的 3-层 PHEWN 中的所有 GM 被修改;当插入边 6-4 的结点时,顶点 4 的 3-层 PHEWN 中的所有 GM 被修改;当插入边 6-5 的结点时,顶点 5 的 3-层 PHEWN 中的所有 GM 被修改.这样,当构造顶点 6 的 4-层 PHEWN 时,生成和修改 GM 的次数为(1+2)+(1+5)+(1+9)=19,它等于顶点 6 的 4-层 MCST 中 TL 的数目.

对顶点 7 和顶点 8 作同样的分析,不难发现,当构造任意的基于 3-层 IHEWN 的 4-层 PHEWN 时,生成和修改 GM 的次数总是等于 4-层 MCST 中 TL 的数目.如果 4-层 IHEWN 的存储结构也被考虑的话,则生成和修改 GM 的次数总是大于 4-层 MCST 中 TL 的数目.

Step 3. 构造基于 4-层 IHEWN 的 5-层 PHEWN,并将它们组合成 5-层 IHEWN.

(1) 构造顶点 5 的 5-层 PHEWN.

对于由 Step2 构造的 4-层 IHEWN,经过 4 次分解-滚动-组合,得到一个 4-层 IHEWN,即路径 4-3-2-1.当顶点 5 被连接至该路径的顶端时,生成和修改 GM 的次数为 4.由于构造顶点 4 的基于 3-层 IHEWN 的 4-层 PHEWN 时生成和修改 GM 的次数为 3,因此,构造顶点 5 的基于 3-层 IHEWN 的 5-层 PHEWN 时,生成和修改 GM 的次数大于 4,即大于顶点 5 的 5-层 MCST 中 TL 的数目.

(2) 构造顶点 6 的 5-层 PHEWN.

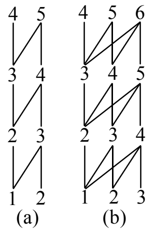

Fig.8 Two IHEWNs
图 8 两个 IHEWN

对于由 Step2 构造的 4-层 IHEWN,经过 4 次分解-滚动-组合,得到一个如图 8(a)所示的 4-层 IHEWN,然后顶点 6 被连接至该 4-层 IHEWN 的顶层.当边 6-4 的结点被插入时,生成 GM_{6-4},而顶点 4 的 4-层 PHEWN 中的所有 GM 被修改;当边 6-5 的结点被插入时,生成 GM_{6-5},而顶点 5 的 4-层 PHEWN 中的所有 GM 被修改.由于构造顶点 4 和顶点 5 的基于 3-层 IHEWN 的 4-层 PHEWN 时生成和修改 GM 的次数分别为 3 和 9,因此构造顶点 6 的基于 3-层 IHEWN 的 5-层 PHEWN 时,生成和修改 GM 的次数大于(1+3)+(1+9)=14,即大于顶点 6 的 5-层 MCST 中 TL 的数目.

(3) 构造顶点 7 的 5-层 PHEWN.

对于由 step2 构造的 4-层 IHEWN,经过 4 次分解-滚动-组合,得到一个如图 8(b)所示的 4-层 IHEWN,然后顶点 7 被连接至该 4-层 IHEWN 的顶层.当插入边 7-4 的结点时,生成 GM_{7-4},而顶点 4 的 4-层 PHEWN 中的所有 GM 被修改;当插入边 7-5 的结点时,生成 GM_{7-5},而顶点 5 的 4-层 PHEWN 中的所有 GM 被修改;当插入边 7-6 的结点时,生成 GM_{7-6},而顶点 6 的 4-层 PHEWN 中的所有 GM 被修改.由于构造顶点 4,5,6 的基于 3-层 IHEWN 的 4-层 PHEWN 时生成和修改 GM 的次数分别为 3,9,19,因此,构造顶点 7 的基于 3-层 IHEWN 的 5-层 PHEWN 时,生成和修改 GM 的次数大于(1+3)+(1+9)+(1+19)=34,即大于顶点 7 的 5-层 MCST 中 TL 的数目.

对顶点 8 和顶点 9 作同样的分析,不难发现,当构造任意基于 3-层 IHEWN 的 5-层 PHEWN 时,生成和修改 GM 的次数总是大于 5-层 MCST 中 TL 的数目.

Step4. 以同样的方式构造 6,7,8,9,10-层 IHEWN 或 HEWN.在构造过程中不难发现,当构造任意的基于 3-层 IHEWN 的 x-层 PHEWN 时,生成和修改 GM 的次数总是大于 x-层 MCST 中 TL 的数目.

在上述构造 10-层 HEWN 的过程中不难发现,当构造 6-层 PHEWN 时所有的连接都存在,因为 10-层 HEWN 的第 6 层的最后一个顶点是 10,它等于图中最大团的大小.

2 HEWN 算法的复杂性分析

首先,在 HEWN 算法中,由于 HEWN 的每一次滚动之后均需对顶层顶点进行连接性检测,因此 HEWN 的存储只能采用层次结构的链表.本文给出的层次结构的链表是最简单的存储方法,若采用其他存储方法,则操作次数更多,因而复杂度更高.

其次,对于含有 n 个顶点的图 $G(V,E)$,假设其最大团的顶点数为 j,HEWN 算法依次构造 n~j 层 HEWN.为了构造 x-层 HEWN,算法开始于 3-层 HEWN 并逐步增加 HEWN 的层.每增加一层 HEWN,算法都要进行分解-滚动-组合操作,而每一次的分解、组合操作都是通过矩阵的“交”、“并”操作来完成的.文献[1]以此为依据将“并/交”操作视为复杂性分析的基本操作,并由“并/交”操作次数是多项式的来推出整个 HEWN 算法是多项式的.事实上,HEWN 算法的实例输入只有一项——图的邻接矩阵,HEWN 算法建立的各个层次的 HEWN 及依附于各边的 GM 均不是问题的实例输入,这些数据需要由计算机一步一步地组织,而组织这些数据的操作是指数的.我们知道,如果算法的某一步是多项式的,则整个算法不一定是多项式的,但如果算法的某一步是指数的,则整个算法就是指数的.

在 HEWN 算法中,尽管“并/交”操作被频繁地使用(其操作次数为 $O(n^{8.5})$[1]),但是复杂性分析的基准应当是生成和修改 GM 的次数,而不是“并/交”操作的次数.假设图 $G(V,E)$的顶点数为 n 而最大团的大小为 j.由于最少连接性优先的策略对 HEWN 算法在最坏情况下的时间复杂度没有影响,因此,为了易于得出结论,图 $G(V,E)$中最

大团的顶点被首先编号.当采用 HEWN 算法求最大团时,依次构造 n~j-层 HEWN,如果其中之一的构造次数是指数的,则整个算法是指数的.下面分析构造 j-层 HEWN 的复杂性.

令 $i(3\le i\le j)$表示 IHEWN 的层数,考虑当 $i=j-(n-j+1)+1=2j-n$ 时构造 i-层 PHEWN 的复杂性.当 $i=2j-n$ 时,如果 $2j>n$,则 j-层 HEWN 的第$(2j-n)$层的最后一个顶点为 j,它等于图中最大团的大小,因而在$(2j-n)$-层 PHEWN 中,所有的连接都存在.在前面的分析中已经知道,当构造基于 3-层 IHEWN 的$(2j-n)$-层 PHEWN 时,生成和修改 GM 的次数大于$(2j-n)$-层 MCST 中 TL 的数目,因此下面先分析$(2j-n)$-层 MCST 中 TL 的数目.由于$(2j-n)$-层 MCST 中路径的数目为 C_j^{2j-n},而每条路径上只有一个叶结点,因而$(2j-n)$-层 MCST 底层上的 TL 的数目为 C_j^{2j-n}.同理,$(2j-n)$-层 MCST 的倒数第 2 层上 TL 的数目为 C_{j-1}^{2j-n-1},依此类推.由此可知,$(2j-n)$-层 MCST 上 TL 的总数为 $C_j^{2j-n}+C_{j-1}^{2j-n-1}+\dots+C_{n-j+2}^{2}$.由于构造基于 3-层 IHEWN 的$(2j-n)$-层 PHEWN 时生成和修改 GM 的次数大于$(2j-n)$-层 MCST 中 TL 的数目,因此,构造$(2j-n)$-层 PHEWN 时生成和修改 GM 的次数大于

$$E(n,j)=C_j^{2j-n}+C_{j-1}^{2j-n-1}+\dots+C_{n-j+2}^{2}=O(C_j^{2j-n}).$$

令 $P(n,j)$表示在其他步骤中生成和修改 GM 的次数,则 HEWN 算法的时间复杂度大于

$$T(n,j)=E(n,j)+P(n,j)=O(C_j^{2j-n})+P(n,j).$$

显然,当 $2j>n$ 时 $T(n,j)$是指数的.因此,HEWN 算法的时间复杂度是指数的,而不是 $O(n^{8.5})$.

3 结果与讨论

HEWN 算法的时间复杂度与图的疏密度有很大的关系.对于一个没有自环和平行边的顶点数为 n 的有限无向图 $G(V,E)$,假设其最大团的大小为 j,当 $2j-n$ 趋于 $j/2$ 时,HEWN 算法的时间复杂度最高.下面通过两个实例来比较 HEWN 算法和 MCST 算法的时间复杂度.

对于如图 9 所示的一个简单实例,采用 HEWN 算法求解其最大团时,依次判断是否存在 8,7,6,5 顶点团,总的生成和修改 GM 的次数为 164 次,而采用 MCST 算法求解其最大团时,总的连接性检测次数为 82 次.

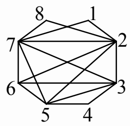

Fig.9 Instance 1
图 9 实例 1

对于图 2 所示的稍微复杂一点的实例,采用 HEWN 算法求解其最大团时,判断是否存在 14,13,12,11,10 顶点团所需生成和修改 GM 的次数分别为 291,2410,5563,9232,12578,其中,在构造 10-层 HEWN 的过程中,当构造基于 5-层 IHEWN 的 6-层 PHEWN 时,仅顶点 10 的 6-层 PHEWN 的生成和修改 GM 的次数就达 1 751 次.HEWN 算法生成和修改 GM 的总次数为 3.01×10^4;采用 MCST 算法求解其最大团时,为了比较在最坏情况下的复杂度,把顶点 14~1 重新编号为 1~14,判断是否存在 14,13,12,11,10 顶点团所需进行的连接性检测次数分别为 46, 297,983,2167,45,总的连接性检测次数为 3.54×10^3.

Table 1 Comparison of the basic operation times between HEWN and MCST algorithm

表 1 HEWN 算法和 MCST 算法的基本操作次数比较

Instance① \ Algorithm②		The times of creating and modifying GM in HEWN algorithm③	The times of testing connection in MCST algorithm④
Instance 1⑤	n=8 j=5	164	82
Instance 2⑥	n=14 j=10	3.01×10^4	3.54×10^3

①实例,②算法,③HEWN 算法中生成和修改 GM 的次数,
④MCST 算法中连接性检测的次数,⑤实例 1,⑥实例 2.

通过比较不难看出,当求解图的最大团时,HEWN 算法生成和修改 GM 的次数远大于 MCST 算法进行连接性检测的次数.事实上,HEWN 算法的时间复杂度是指数的,且其复杂度是非常高的.

致谢 山东大学计算机学院的朱大铭教授对本文的工作给予了细心的指导,栾峻峰博士对本文的完成提出了有益的建议,复旦大学电子工程系的唐璞山教授为本文的写作提供了有益的资料,在此一并表示感谢.

References:

[1] Tang, Pu-shan, Huang, Zhi-jun. HEWN: a polynomial algorithm for CLIQUE problem. Journal of Computer Science and Technology, 1998,13(Supplement):33~44.

[2] Yan, Wei-min, Wu, Wei-min. Data Structure. 2nd ed., Beijing: Tsinghua University Press, 1992 (in Chinese).

附中文参考文献:

[2] 严蔚敏,吴伟民.数据结构(第 2 版).北京:清华大学出版社,1992.

Complexity Analysis for the HEWN Algorithm—A Discussible Opinion*

HAN Ai-li, YANG Zhi-min

(*Department of Computer Science*, *Shandong University at Weihai*, *Weihai* 264209, *China*)

E-mail: hanal@sdu.edu.cn

http://www.wh.sdu.edu.cn

Abstract: In this paper, the time complexity of the HEWN algorithm for the maximum clique problem is analyzed. Firstly, a sort of data structure to implement the HEWN algorithm is designed by analyzing the structural characters of HEWN and the needed operations. It is pointed out that the storage structure of HEWN should use the linked list which combine adjacency multi-list with binary linked list. And then, the building procedure of HEWN is anatomized by starting with the storage structure of HEWN. In the anatomizing procedure, by comparing with the MCST, it is pointed out that underlying exponential times of creating and modifying GM exists in the HEWN algorithm when $2j>n$. Therefore, the time complexity of the HEWN algorithm is exponential instead of $O(n^{8.5})$.

Key words: algorithm complexity; NP-completeness; clique

* Received May 28, 2001; accepted December 6, 2001
Supported by the National Natural Science Foundation of China under Grant No.69874001