边带权最大独立集问题及其近似算法*)

张 华 朱 洪

(复旦大学计算机工程系智能信息处理实验室 上海200433)

摘 要 区别于传统对带权最大独立集问题的研究，本文从新的角度首先提出了边带权最大独立集问题，给出了完整的定义，证明了它的NP-Complete难解性。并且通过对问题结构的研究，给出了一个近似度为$\frac{1}{\lceil(\Delta'+1)/3\rceil}$的近似算法，$\Delta'$为图中点的最大度数。

关键词 带权最大独立集问题，边带权最大独立集问题，NP-Complete，近似算法

Edge Weighted Maximum Independent Set Problem and an Approximate Algorithm for it

ZHANG Hua ZHU Hong

(Laboratory for Intelligent Information Processing, Fudan University, Shang Hai 200433)

Abstract Different from traditional research domain of Weighted Maximum Independent set Problem, in this paper, we firstly propose Edge Weighted Maximum Independent set Problem, give the formal definition and prove it is Np-Complete. Through researching the structure of problem, we give an approximate algorithm to solve it, which has the approximation ratio $\frac{1}{\lceil(\Delta'+1)/3\rceil}$, where Δ' is the maximum degree of the vertex in graph G.

Keywords Weighted maximum independent set problem, Edge weighted maximum independent set problem, NP-complete, Approximate algorithm

1 引言

最大独立集问题作为NP-Complete类问题中的核心问题，形成时间由来已久。它在工业过程控制，网络设计，大规模集成电路设计及经济模型分析等诸多方面都有着广泛的运用，同时它也是组合优化研究的核心问题，有着极高的研究价值。在过去很长一段时间内不少计算机科学家和数学家在这个领域做了许多卓有成效的研究工作，取得了一系列的研究成果。纵观前人的工作，我们发现它们中的绝大多数在涉及最大独立集的带权情况时(即带权最大独立集问题)都集中在顶点带权的情况，对于边带权的情况却绝少提及。而我们在现实生活中遇到的许多问题，如：计算经济学中的CAP(Combinatorial Allocation Problem)问题等等，实际上可以很容易地归纳为图论中的边带权情况的独立集问题。从这点出发，我们第一次系统地提出了边带权的最大独立集问题，第一次给出了它是NP-complete的证明并且通过对问题结构的发掘，进一步给出了一个近似度为$1/\lceil(\Delta'+1)/3\rceil$的近似算法，$\Delta'$为图中点的最大度数。在丰富独立集问题研究领域的同时也为从计算理论的角度研究其他学科领域中的问题打下坚实的理论基础。

2 边带权最大独立集问题

如前所述，至今以前人们对带权最大独立集问题所作的工作都集中在顶点带权的最大独立集问题研究上，并取得了丰硕的成果。其形式定义如下：

定义1(顶点带权最大独立集问题，WMIS)：给定一个无向图$G=(V,E)$，$V=\{1,2,\cdots,n\}$是G中的顶点集，$E=\{e_{ij}\}$为G的边集，$i,j\in V$。对于$\forall\ i\in V$，$W(i)\in N$是顶点i上的权值。问题是找到V中的一个子集$I\subseteq V$，对于$\forall\ m,n\in I$，在图G中没有边直接相连，即$e_{mn}\notin E$，并且$\sum_{i\in I}W(i)$最大。

我们从新的角度，即边带权而点不带权的情况出发，重新定义了最大独立集问题，形式定义如下：

定义2(边带权最大独立集问题，EWMIS)：给定一个无向图$G=(V,E)$，$V=\{1,2,\cdots,n\}$为G的顶点集，$E=\{e_{ij}\}$为边集$i,j\in V$。同时，对于$\forall\ e_{ij}\in E$，$W(e_{ij})\in N$是边e_{ij}上的权值(e_{ij}和V中的点i,j相关)。问题是找到V中的一个子集$I\subseteq V$，对于$\forall\ m,n\in I$，在图G中没有边直接相连，即$e_{m,n}\notin E$，并且$\sum_{i\in I}\sum_{e_{ij}\in E}W(e_{ij})$最大。

考虑到方便和避免重复，我们将边带权最大独立集问题简写为EWMIS，而称一般的带权最大独立集(点带权最大独立集)问题为WMIS。

由于EWMIS是一个最优问题，它所要求的是在给定图G中与其相关的边的权值之和最大的顶点独立集。作为判定问题，我们只需要问，图G中是否存在顶点独立集I，与I相关联的所有边上的权值之和大于等于K，$K\in N$。形式化如下：

EWIS$=\{\langle G,K\rangle$：G中存在顶点独立集I并且与I关联的所有边上的权值之和大于等于$K\}$

命题1 边带权独立集问题(EWIS)$\in$Np-Complete。

证明：首先，我们证明EWIS$\in$NP。对于给定的图$G=(V,E)$，其证据就是顶点独立集I，$I\subseteq V$。验证算法首先对I是否

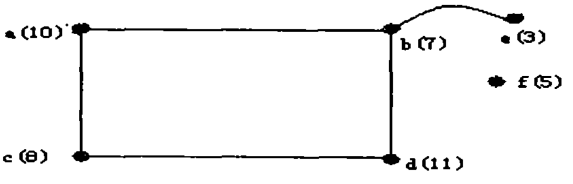
*)本文工作得到科技部基金(No. 2001CCA03000)，国家自然科学基金(No. 60273045)，上海科学技术发展基金(No. 025115032)的支持。张 华 硕士，主要研究方向为计算复杂性理论，近似算法，计算经济学；朱 洪 教授，博士生导师。

为独立集进行验证，检查 I 中任意两个顶点间是否有边相连，这个操作最多可以在 $|V|^2$ 的时间内完成，即遍历图 G 的相邻矩阵即可完成。然后，算出与 I 中所有顶点相关联的边上的权值之和，看是否大于等于 K。这同样可以在最多 $|V|^2$ 的时间内完成，方法与检查顶点相连的方法相同。由此，验证算法总可以在图 G 的多项式时间内完成。EWIS$\in$NP。

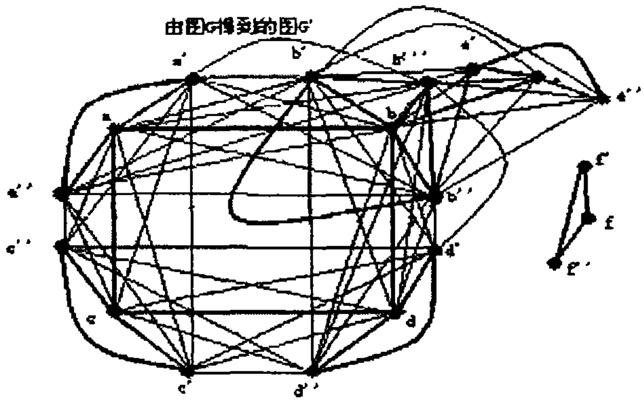
其次，我们证明 WIS$\leqslant_P$EWIS，即 EWIS$\in$NP-Complete。WIS 为点带权独立集问题。给定一个无向图 $G=(V,E)$ 和一个正整数 K，我们总能构造另一个无向图 $G'=(V',E')$，使得，若 G' 中存在一个顶点独立集 I'，所有与它关联的边的权值之和大于等于 K，当且仅当 G 中也存在一个顶点独立集 I，I 中所有点的权值之和大于等于 K。下面是证明过程：

第一步：对于 G 中的任何度数大于等于2而之间又有相邻关系存在的顶点 i,j,k。不失一般性，假设其中 i 的度数为2，即 $e_{ij},e_{ik}\in E$，为了讨论的方便，用 $i-j$，$i-k$ 表示它们。先分别在 i,j 和 i,k 间增加一条路径 $i-i'-j'-j$ 和 $i-i''-k'-k$（其中 i',i'',j',k' 都是新增加的顶点），然后连接 i',i''。我们称 i',i'' 从属于 i，它们和 i 共同构成一个点单位，点单位中的点相互间连接，i 称为该点单位的生成点，i',i'' 称为从属点。同样，j',k' 分别从属于点 j,k。对图 G 中的每条连接两个度数大于等于2的顶点的边都进行这样的操作，我们称这样的操作为边扩张。最后 i,j,k 分别构成了三个不同的点单位。这些操作可以在 $O(|E|)$ 的时间内完成。第二步：把点单位 i 中的每个点和点单位 j 中的每个点进行连接，对于 $i-j$ 和 $i'-j'$ 无需再次连接，因为它们之间已经有边相连。对于 k 所定义的点单位，因为在图 G 中点 i 与点 k 也相连，故点单位 i 中的每个点同样要与点单位 k 中的每个点相连。对图 G 中所有存在这样关系的点单位都进行这样的操作。这些操作可以在 $O(|E|\cdot\Delta^3)$ 的时间内完成。Δ 为图 G 中点的最大度数。第三步：对于图 G 中的悬挂点 x，即 x 的度数为1。不失一般性我们设 $(x-y)\in E$，除对其相应的悬挂边作扩张之外，得到一个从属点 x'，还需增加另一个从属点 x''，这三个点 $\{x,x',x''\}$ 相互连接构成一个点单位。将悬挂点构成的点单位 x 中的每一点连接到 y 所构成的点单位中的每一点上去，已经连接的则无需再连接。对于图 G 中有独立点 z 的情况，即 z 的度数为0，则只需增加两个从属于 z 的点 z' 和 z''，将这三点相互连接，构成一个独立的点单位。这些操作可在 $O(\Delta\cdot|V|)$ 时间内完成。Δ 为图 G 中点的最大度数。第四步：把所有点单位中的生成点 i 上的权值平均分配到它与从属点之间连接的边上去，用 $degree(i)$ 表示点单位 i 中的从属点个数，则这些边上的权为 $\lfloor W(i)/degree(i)\rfloor$，若 $W(i)$ mod degree$(i)\neq 0$，则可将其加到 i 的任意一条从属点与它连接的边上去。对于余下的边，即各对从属点之间的边以及原图 G 中的边，其权值一律赋值0。这些操作可在 $O(|E|\cdot\Delta^3)$ 时间内完成。Δ 为图 G 中点的最大度数。上面这些操作的目的是为了把每个顶点上的权分配到与它相关联的边上去，使得这些边上的权值和等于原来顶点上的权，同时又保证原图 G 中顶点间的相邻关系。

我们举一个例子。原图 G，如下图所示：



括号里的数值为该点上的权。按照前面的方法操作将得到如下的图 G'：



以下是图 G' 中各条边上的权值

点单位 a 中的边：$W'(a-a'')=[W(a)/degree(a)]=5$；$W'(a-a')=[W(a)/degree(a)]=5$；$W'(a'-a'')=0$；

点单位 b 中的边：$W'(b-b')=[W(b)/degree(b)]=2$；$W'(b-b'')=[W(b)/degree(b)]=2$；

$W'(b-b''')=[W(b)/degree(b)]+(W(b)\bmod degree(b))=3$；$W'(b'-b'')=W'(b'-b''')=W'(b''-b''')=0$；

亦可作这样的分配：$W'(b-b')=3$；$W'(b-b'')=2$；$W'(b-b''')=2$；

点单位 c 中的边：$W'(c-c')=[W(c)/degree(c)]=4$；$W'(c'-c'')=0$；$W'(c-c'')=[W(c)/degree(c)]=4$；

点单位 d 中的边：$W'(d-d')=[W(d)/degree(d)]=5$；$W'(d'-d'')=0$；$W'(d-d'')=[W(d)/degree(d)]+(W(d)\bmod degree(d))=6$；

点单位 e 中的边：$W'(e-e')=[W(e)/degree(e)]=1$；$W'(e-e'')=[W(e)/degree(e)]+(W(e)\bmod degree(e))=2$；

$W'(e'-e'')=0$；亦可为，$W'(e-e')=2$，$W'(e-e'')=1$；

点单位 f 中的边：$W'(f-f')=[W(f)/degree(f)]=2$；$W'(f'-f'')=0$；$W'(f-f'')=[W(f)/degree(f)]+(W(f)\bmod degree(f))=3$；亦可为：$W'(f-f')=3$；$W'(f-f'')=2$；

图 G' 中所有剩余的边，如：$W'(a'-b')$，$W'(a'-b)$，$W'(a'-b'')$，$W'(a'-b''')$，$W'(a''-b')$，$W'(a''-b)$，$W'(a''-b'')$，$W'(a''-b''')$ 等等，统统赋值为0。

通过上述操作，我们可以看到，在得到的图 G' 中，任何一个点单位中，与生成点 i 关联的所有边权值之和最大，而且等于 i 在图 G 中的权值 $W(i)$。而任何一个从属点 i'，其所有关联边的权值之和小于等于 $\lfloor W(i)/degree(i)\rfloor+(W(i)\bmod degree(i))\leqslant W(i)$。除此之外，对于原来在图 G 中有边相连的两个顶点，比如：$(i-j)\in E$，在图 G' 中，点单位 i 中的每个点与点单位 j 中的每个点必然相互连接。而对于那些在图 G 中没有边直接相连的点，如 $(m-n)\notin E$，在新的图 G' 中，点单位 m 中的任何点和点单位 n 中任何点都不会连接。反过来在图 G' 中，若这两个点单位中的点没有边连接，则这两个点单位的生成点在图 G 中也不会有边直接相连。故，综合上述结论，若图 G 中存在一个独立集 I，其点上的权值之和大于等于 K，则在图 G' 中，I 同样也必然是独立集，而且与 I 中的点关联的所有边的权值总和也必然大于等于 K。反之若图 G' 中存在一个独立集 I'，其相关边的权值之和大于等于 K，则在图 G 中必然也存在一个独立集 I，其点上的权值之和大于等于 K。从图 G 到图 G' 的所有这些操作都可在 $O(|E|\cdot\Delta^3+|V|\cdot\Delta)$ 的时间内完成。已知 WIS$\in$NP-Complete。故 EWIS$\in$NP-Complete。

从上面的定理,我们得到了下面的推论。

推论1 边带权最大独立集问题(EWMIS)∈Np-hard.

3 解边带权最大独立集问题(EWMIS)用到的一些基本知识

对于Np-hard优化问题,由于$P\neq NP$?的关系,我们通常为其设计一个多项式时间的近似算法,通过近似解去逼近最优解。对于求最大值的Np-hard优化问题,我们称一个近似算法的近似度为$\rho(N)$,如果对于任意规模为n的输入,近似算法所得到的解C和问题最优解C^*之间的最小比值为$\rho(n)$,即$\min(C/C^*)\geqslant\rho(n)$。

定理1 无向图$G=(V,E)$,其中$\Delta'=\max(\Delta(i)),i\in V$,$\Delta(i)$表示点$i$上的度数。则存在$G$上的一个划分,可以把$G$分为$\lceil(\Delta'+1)/3\rceil$个推导子图,其中每个子图中任何点在该子图中的度数不会超过2。而且这样的划分最多只需要$O(|V|\cdot\lceil(\Delta'+1)/3\rceil)$的时间[7]。

证明:首先,将V中的点任意划分为$\lceil(\Delta'+1)/3\rceil$个子集,对每个子集中的每个点重复以下的操作:若点$i$在由它所在子集中的点所构成的$G$的推导子图中度数大于2,则不断移动点$i$到其他子集中去,直到它所在子集中所有点所构成的推导子图中,i的度数小于等于2。这样的子集必定存在,否则i的度数将大于等于$3\cdot\lceil(\Delta'+1)/3\rceil\geqslant\Delta'+1$。完成这样的全部操作只需$G$的多项式时间,即$O(|V|\cdot\lceil(\Delta'+1)/3\rceil)$时间内完成。因为,对于$G$中任何一个点$i$来说,它至多移动$\lceil(\Delta'+1)/3\rceil$次就一定可以找到一个符合条件的子集。图$G$中至多有$|V|$个这样的点需要移动。在这个过程中,$\lceil(\Delta'+1)/3\rceil$个子集的数量也不会因为点的移动而减少,因为若一个子集中点的数量少于等于3时,其中的点在其推导子图中的度数绝对不会大于2。

下面是有关WMIS(即点带权最大独立集)问题的一个定理,目的是方便我们在后面对算法思路所作的说明。

定理2 给定无向图$G=(V,E)$,若对于$\forall i\in V,\Delta(i)\leqslant 2$(这里$\Delta(i)$指的是点$i$上的度数),则该图上的点带权最大独立集问题(WMIS)存在多项式时间的精确算法[4]。

证明:参见T H,Corman的书Introduction to Algorithms,第二版,其中有详细论述。

由上述两个定理,我们不难发现,如果将图G中的顶点划分成$\lceil(\Delta'+1)/3\rceil$个子集,就可以在多项式时间内得到一个$V$上的划分,其中在每个顶点子集中,点在其生成子图中的度数小于等于2,将整个图G中边上的权叠加到与它相关的点上去。根据定理2,我们可以在多项式时间内求出每个划分后的顶点子集所定义的推导子图中WMIS问题的最优解。从$\lceil(\Delta'+1)/3\rceil$个最优解中选出最大的,就作为近似算法的解。根据这个思路,我们提出了EWMIS问题的近似算法。我们会在本文下一节给出算法的形式化描述,并对其正确性,近似度与及时间复杂度进行证明。

下面是我们在图G受限时得到的另一个结论:

命题2 无向图$G=(V,E)$,若对于$\forall i\in V,\Delta(i)\leqslant 2$(这里$\Delta(i)$指的是点$i$上的度数),则该图上的EWMIS问题存在多项式时间的精确算法,且所求出的最优解I所关联的所有边的权值之和等于图G中所有边上的权值总和。

证明:若图G中任何点上的度数小于等于2,即$\forall i\in V,\Delta(i)\leqslant 2$,则对于$G$中的点来说其拓扑关系只有以下三种:(1)孤立点;(2)由一些点组成的不相交的几条路径;(3)由一些点组成的没有公共点的环。对于第一种情况,孤立点无边,显然不予考虑。而对于第二,三种情况,即对于图G中的路径和环来说,比如$i-j-k-x-y$是图G中的一条路径,则该条路径上边权值之和最大的顶点独立集就只有两个,要么是$\{i,k,y\}$,要么是$\{j,x\}$,而且这两个独立集所关联的边的权值之和相等,就是该条路径上的所有边的权值总和,我们只需取它们中的一个就可。环的情况是类似的。对于图G来说,显然G中边权总和最大的顶点独立集I就是各条边和各个环上的边权总和最大的顶点独立集的并集。而与I相关联的所有边上的权值之和就是图G中所有边的权值总和。这一切都可以通过遍历图G的相邻矩阵得到。在$O(|V|^2)$的时间内得出最优解。

4 EWMIS问题的近似算法

根据我们在上一节对算法思想作的描述,现将它形式化如下:

```
    Approximate Algorithm For EWMIS (G)
  input:G=(V,E),图G的表示形式为它的相邻矩阵A⊆|V|×|V|,其中若点i与点j有边相连,则A[i,j]=d,d∈N,为该边上的权值;若i,j之间没有边,A[i,j]=0;
  output:独立集I,I⊆V;
{1 将G中的所有点V分成⌈(Δ'+1)/3⌉个子集.用V[i]=k表示点i
   属于子集k,1≤k≤⌈(Δ'+1)/3⌉,1≤i≤|V|. //Δ'为图G中点的
   最大度数.
2 for (i=1;i++;i<=|V|) flag[i]=0; /* 这里flag[i]=0,表
     示点i,在所在子集的推导图中度数大于2,反之,flag[i]=1,表
     示其度数小于等于2,符合要求。*/
3 for(i=1;i++;i<=|V|)
  { where (flag[i]≠1) /* 即,将i放到另一个子集中去,改变V[i]
           的值,使其区别于原用 */
    {update(V[i]); /* 过的所有值; */
      if (Δ(i)<=2) flag[i]=1;/* 在当前子图中,即由所有V
         [*]上的值等于V[i]的点组成的推导子图中,检查点i的
         度数,看是否满足小于等于2的要求.若满足则标志位flag
         [i]置1;Δ(i)表示i在当前子图中的度数 */
   }}
4 for(i=1;i++;i<=⌈(Δ'+1)/3⌉)//S[i]用来表示第i个顶点子
  集.
  { S[i]=∅; //这里是把点放到它属于的子集S[*]中去.
   for (j=1; j++; j<=|V|)
   if (V[ j ]==i)将顶点i放入子集S[i]中; }
5 for(i=1; i++; i<=|V|)//将边上的权加到点上去.
  { W[i]=0;
   for(j=1; j++ ; j<=|V|)W[i]=W[i]+A[i,j]; }
6 for (i=1 ; i++;i<=⌈(Δ'+1)/3⌉)
  { 求出由子集S[i]中点所组成的推导子图上的WMIS(点带权最大
    独立集)问题的最优解I[i];
    Weight[i]= I[i]中所有点上的权值之和;}
7 max =1;//从⌈(Δ'+1)/3⌉个推导子图的最优解I[*]中找出权值
  之和最大的最优解I[max];
  for( i=1 ; i++ ;i<=⌈(Δ'+1)/3⌉){if (Weight[i]>Weight
  [max]) max = i ; }
8 output I[max]; //输出I[max];
  }//算法结束
```

算法中的步骤1,2,3目的是为了将G划分为$\lceil(\Delta'+1)/3\rceil$个点不相交的子图,而且在每个子图中任何点的度数小于等于2。它的可行性我们在定理1中已经有了详细的证明。步骤4,5,6是为了求出每个子图上WMIS(点带权最大独立子集)问题的最优解,目的是从这些最优解中找出最大的一个来作为EWMIS(边带权最大独立集)问题在图G上的近似解。定理2也已经证明这些可以在多项式时间内做到。而7,8两步则就是找到最大的解并输出。我们在后面对算法的分析中会看到这是一个近似度为$\frac{1}{\lceil(\Delta'+1)/3\rceil}$的近似算法,$\Delta'$为图中点的最大度数。

5 近似算法的分析

命题3 算法的输出,$I[max]$,是一个独立集。

证明:这是显然成立的,因为$I[max]$实际是图G中某个

推导子图上 WMIS 问题的最优解。

命题4 该近似算法的近似度为$1/\lceil(\Delta'+1)/3\rceil$，其中$\Delta'$是图$G$中点的最大度数。

证明：将图G划分为$\lceil(\Delta'+1)/3\rceil$个子图，$\Delta'$为图中点的最大度数。将每个点在图$G$中相关的所有边上的权值求和，作为该点的权值，求出每个子图中 WMIS(点带权最大独立集)问题的最优解$I[i]$。我们会看到图G上 EWMIS(边带权最大独立集)问题的最优解I^*最多等于这$\lceil(\Delta'+1)/3\rceil$个局部最优解的并集。因为经过划分后，$I^*$中每个点都会属于$\lceil(\Delta'+1)/3\rceil$个顶点子集中的其中一个，它们所关联的边的权值总和不会超过它们所在的子集中的 WMIS 问题的最优解的权值。即使几个I^*中的点同时属于划分后的某个子集，情况也是如此。

我们从这$\lceil(\Delta'+1)/3\rceil$个最优解中找出权值最大的解$I[max]$。用$Weight(I[i])$表示子集$I[i]$中所有点的权值之和，则有$Weight(I[max])>=Weight(I[i])$，$1\leqslant i\leqslant\lceil(\Delta'+1)/3\rceil$，进而有$\lceil(\Delta'+1)/3\rceil Weight(I(max))\geqslant\sum_{1\leqslant i\leqslant\lceil(\Delta'+1)/3\rceil}Weight(I[i])$；$Weight(I^*)$表示图$G$中 EWMIS 问题最优解$I^*$关联的所有边的权值之和。根据上面我们的分析，有$Weight(I^*)\leqslant\sum_{1\leqslant i\leqslant\lceil(\Delta'+1)/3\rceil}Weight(I[i])$，最后推出，$\rho(n)=Weight(I[max])/Weight(I^*)\geqslant 1/\lceil(\Delta'+1)/3\rceil$，故该算法的近似度为$1/\lceil(\Delta'+1)/3\rceil$。

命题5 算法的运行时间为$O(|V|^2)$

证明：步骤1可以在常数时间内完成，步骤2运行时间为$|V|$，步骤3运行时间最多为$|V|\cdot\lceil(\Delta'+1)/3\rceil$，步骤4运行时间为$|V|$，步骤5，6，7可以在$O(|V|^2)$的时间内完成。所以，算法总的运行时间可以在$O(|V|^2)$时间内完成。

结束语 本文从新的角度出发，发展了原有的带权最大独立集问题的领域，第一次系统地提出了边带权最大独立集问题，而且通过对其属于 NP-Complete 性质的证明以及问题结构的研究分析，进一步给出了一个近似度为$1/\lceil(\Delta'+1)/3\rceil$近似算法。我们未来的研究重点将放在边带权最大独立集问题特有性质的进一步发掘上，希望通过问题特有的结构找到具有更好近似度的算法。

参考文献

1 Hochbaum D S. Approximation Algorithm for Np-hard Problem. PWS Publishing Company ,1997

2 Corman T H,Leiserson C E,Rivest R L,Stein C. Introduction to Algorithms,Second Edition,The MIT Press,2001

3 Arkin E M,Hassin R. On local search for weighted k-set packing. ESA'97

4 Baker B S. Approximation algorithm for NP-complete problems on planar graphs. J. ACM,1994

5 Halldorsson M M,Lau H C. Low-degree graph partitioning via local search with applications to constraint satisfaction,max cut,and 3-coloring. J. Graph Algo. Applic. 1997

(上接第134页)

```
{
    LSInterestExp interestExp;
    List<LSInterestExp> dataServerExp;
    sockaddr dataServerLoc;
    int nPort;
    int nAddrType;
}
```

类 LSInterestExp 中的4个值分表代表一个绑定盒在X、Y两个坐标轴上的最大最小值；类 LSPacketQuest 中的 interestExp 表示用户感兴趣的区域，我们只用一个简单兴趣区域表达式来表示用户的兴趣区域；类 LSPacketReply 中的 dataServerLoc 和 nPort 表示一个服务器的地址和端口号，服务器可能是数据服务器，也可能是定位服务器，nAddrType 用于标记服务器的类型；interestExp 表示客户兴趣区域和 dataServerLoc 表示的数据服务器的兴趣区域的相交部分，dataServerExp 则表示返回的数据服务器所对应的空间的兴趣区域表达式描述，可能是一个，也可能是多个简单兴趣区域表达式。

在我们的实现中，定位服务机制和数据分布是相互独立的，而且定位服务是一种在应用层之上的服务，可以作为一个应用程序在计算机上运行，也就是说同一台计算机既可以作为数据服务器，也可以作为定位服务器。

定位服务和数据的网络传输会造成一定的延迟，降低了客户端响应的实时性。一种有效的办法是在客户端使用行为预测方法，根据用户的视点历史位置，对视点将来的位置做预测，客户端根据预测的位置向定位服务器发送请求，然后向对应的数据服务器预取数据。在数据的传输和渲染时，还需要累进式传输(Progressive Transmission)、累进式渲染(Progressive Rendering)等相关技术，我们将另外著文说明。

结论和进一步工作 本文提出了一种用于分布式虚拟环境的定位服务机制，可以在客户端只需知道自己的空间位置和感兴趣区域的情况下，提供对虚拟环境中分布式存放的静态数据的有效、可靠的访问。这种方法可以独立于分布式虚拟环境动态通讯结构，具有一定的通用性。我们已将这种机制应用到了我们的分布式虚拟环境 AIMNET 中。

目前的定位机制中，各定位服务器的服务区域都是预先配置的，而且这种机制只能处理静态数据的精确定位，下一步我们将研究动态的服务区域配置算法以及分布式虚拟环境中的动态数据的精确定位机制。另外，我们的虚拟环境在安全服务和用户权限管理方面有待加强，我们会将安全服务和用户权限管理引入分布式虚拟环境，并将其和定位服务机制相结合。

参考文献

1 潘志庚，姜晓红，张明敏，石教英．分布式虚拟环境综述．软件学报，2000，11(4)：461～467

2 孙元浩．可扩展主动兴趣管理：[南京大学硕士毕业论文]．2003

3 龚震宇．分布式虚拟环境中的事件模型：[南京大学硕士毕业论文]．2003

4 姜晓红，潘志庚，杨孟洲，石教英．多服务器/客户 DVE 系统的分区数据管理模式．计算机学报，2000，23(8)：872～876

5 杨孟洲，姜晓红，潘志庚，石教英．分布式虚拟环境中的信息服务和资源定位．计算机研究与发展，2000，37(10)：1227～1232

6 Leonhardi A,Rothermel K. Architecture of a Large-scale Location Service. In:Proc. of the 22 nd Intl. Conf. on Distributed Computing Systems (ICDCS'02)