

文章编号: 1005—0523(2002)04—0071—02

图的最大完全子图的一个算法

周尚超, 刘二根, 王 森

(华东交通大学 基础科学学院, 江西 南昌 330013)

摘要:利用递归算法和引入点的向后度等概念, 从而加快了图的极大完全子图的计算.

关 键 词:极大完全子图; 点独立数 $\beta_0(G)$; 递归函数.

中国分类号: O157.6

文献标识码: A

1 引言和定义

本文所指的图 G 均为无向简单图, $V(G)$ 为图的顶点集. 其余未定义的术语参见^[1]. 在很多图的问题中, 经常需要计算图 G 的极大完全子图的阶数或 G 的补图的点独立数 β_0 . 例如在计算 Ramsey 数时就必须计算它.

如果 p 点自补图 G 的最大完全子图是 K_n , 则 Ramsey 数 $R(n+1, n+1) \geq p+1$ 在实际问题中, 也会碰到此类问题. 例: 2000 年国际大学生计算机程序设计竞赛上海赛区的一道试题(同年东南欧赛区也有一道相同的题)如下: 某旅游公司组织旅行团, 在 n 个游客中, 有游客 i 不愿与游客 j 在同一旅行团情况, 公司记录了所有这些情况. 公司也不愿意把不愿在同一旅行团的两人安排在同一旅行团. 要求计算人数最多的旅行团的人数. 如果用 $1, 2, \dots, n$ 记录 n 个游客, 把它作为图 G 的 n 个点, 如果游客 i 愿意与游客 j 在同一旅行团, 则记 $A[i, j]=1$; 否则 $A[i, j]=0$; 问题就转化为求图 G 的最大完全子图的阶数.

算法: 如果计算出图 G 有 $m-1$ 阶完全子图, 则计算 G 有无 m 阶完全子图. 如无则最大完全子图的阶数为 $m-1$, 停止计算. 如有则计算 G 有无 $m+1$ 阶完全子图, 这样继续下去就可算出最大完全子图

的阶数.

计算 G 有无 m 阶完全子图用递归算法. 如果我们要找一个 m 阶完全子图, 构成它的顶点集是 $\{x[1], x[2], \dots, x[m]\}$, $x[i] < x[i+1]$. 第 1 个点 $x[1]$ 在 G 的第 1 到第 $n-m+1$ 个点之间找, $\text{degree}[x[1]] \geq m-1$. 第 2 个点 $x[2]$ 在 G 的第 $x[1]+1$ 到第 $n-m+2$ 个点之间找, $\text{degree}[x[2]] \geq m-1$. 等等. 如果 $m=3$, 则程序如下

```
flag:=false;
for x[1]:=1 to n-m+1 do begin
  if flag=true then break;
  if degree[x[1]]<m-1 then continue; //如 x[1] 的度 <
m-1, 则 x[1] 取下一个值
  for x[2]:=x[1]+1 to n-m+2 do begin
    if flag=true then break;
    if degree[x[2]]<m-1 then continue;
    if A[x[2], x[1]]=0 then continue;
    for x[3]:=x[2]+1 to n-m+3 do begin
      if flag=true then break;
      if degree[x[3]]<m-1 then continue;
      if A[x[3], x[1]]=0 then continue;
      if A[x[3], x[2]]=0 then continue;
      flag:=true;
      break;
    end;
  end;
end;
```

收稿日期: 2002—05—10

作者简介: 周尚超(1948—), 男, 云南蒙自人, 教授.

end;

如果程序能运行到 $\text{flag} = \text{true}$ 处, 就说明找到了 3 个点 $x[1], x[2], x[3]$, 其中任意 2 个点的 $A[x[i], x[j]] = 1$, 这 3 点构成 K_3 , 跳出循环, 停止计算. 否则, 找尽所有 3 个点后停止计算. 以 $\text{flag} = \text{false}$ 说明没有 K_3 .

因此程序能计算出是否有 m 阶完全子图. 当 m 较大时, 需要较长的运行时间, 且要写 m 重 for 循环嵌套的程序, m 改变后, 程序也得改变. 使用递归程序, 就只要用 1 个 for 循环就行了.

如果在 n 点图中没有 K_m , 它从 n 个点中选出 m 个点进行验证, 验证 $C(n, m)$ 次 (从 n 个物体中选 m 个的组合数). 从程序上来看, 验证次数没有这么多, m 个点中的每点必须满足度不小于 $m-1$ 的条件. 当程序运行到

for $x[p] := x[p-1] + 1$ to ... 时

除条件 $\text{degree}[x[p]] \geq m-1$ 外

$x[p]$ 必须与前面的 $p-1$ 个点连接, 要做 $p-1$ 次判断. 如 $x[p]$ 的度大于等于 $m-1$, $x[p]$ 还不能成为满足条件的点, $x[p]$ 必须与 (至少) $m-p$ 个大于 $x[p]$ 的点邻接. 用点 $x[p]$ 的度必须大于 $m-1$ 去筛选, 这种筛选效率是很低的. 如用条件: $x[p]$ 必须与前面的 $p-1$ 个点连接, $x[p]$ 与至少 $m-p$ 个大于 $x[p]$ 的点邻接用这个条件进行筛选, 显然效率要高得多. 在上面的程序中

if $\text{degree}[x[3]] < m-1$ then continue; 可改为

if $\text{Bdegree}[x[3]] < m-3$ then continue;

Bdegree 的定义如下.

定义 1 设 $V(G) = \{1, 2, \dots, n\}$, 点 i 的向后度 $\text{Bdegree}[i]$ 是 $V(G)$ 中大于 i 且与 i 邻接的顶点的个数. 点 i 的向前度 $\text{Fdegree}[i]$ 是 $V(G)$ 中小于 i 且与 i 邻接的顶点的个数.

有了此定义, 则前面的筛选条件为 $x[p]$ 的向后度 $\text{Bdegree}[x[p]] \geq m-p$; 还可加上向前度 $\text{Fdegree}[x[p]] \geq p-1$ 的条件. 但在实际计算中, 加上向前度条件效率不高甚至降低, 因为它增加了 1 次筛选性较低判断, 因而不需向前度条件.

显然点 i 的度 $\text{Degree}[i] = \text{Fdegree}[i] + \text{Bdegree}[i]$

也可定义多维向后度.

定义 2 设 $V(G) = \{1, 2, \dots, n\}$, $i, j \in V(G)$, $i < j$, 2 维向后度 $\text{B}^2\text{degree}[i, j]$ 是与 i 和 j 都邻接且大于 j 的点的个数, 3 维向后度 $\text{B}^3\text{degree}[i, j, k]$ $i < j < k$ 是与 i, j, k 都邻接且大于 k 的点的个数.

4 维和更高的向后度可以类似定义.

在计算机内存允许的情况下可以用更高的向后度来提高效率. 例如 $n=100$ 时, 4 维度占据了 100 兆的内存空间. 下面的引理是显然的.

引理 如果图 H 至少有 m^0 个点其度大于等于 m^0-1 , 但没有 m^0+1 个点其度大于等于 m^0 , 则 H 的极大完全子图的阶数 $\leq m^0$.

定义 3 引理中的 m^0 称为图 H 的极大完全可能值.

定义 4 与 x 连接的大于 x 的点组成的图 H 的极大完全可能值称为点 x 的极大完全可能值.

利用引理可以将向后度条件进行改进. 条件 $x[p]$ 的向后度 $\text{Bdegree}[x[p]] \geq m-p$ 可以加强为: $x[p]$ 的极大完全可能值 $\geq m-p$. 利用此条件, 计算速度显著提高. 在本文的程序中极大完全可能值仍然用 Bdegree 表示

2 程序与计算实例

我们用 Delphi 进行编程, 在 Pentium 75 (内存 32 兆) 机上运行. 编制了计算图的极大完全子图的递归程序. 只要将图的邻接矩阵 (n 行 n 列的 0-1 矩阵) 存入文本文件 *Sboob.txt*. 程序从此文件读入数据即可运算. 程序运算的步骤是: 计算图的 3 阶完全子图, 如有就计算 4 阶完全子图, 如有 m 阶完全子图, 就计算 $m+1$ 阶完全子图, 如无 $m+1$ 阶完全子图, 则极大完全子图的阶是 m .

运算实例: 随机构造一个 50×50 的 0-1 矩阵 A , 即随机产生 2 500 个小于 100 的整数, 如果其值大于 10 就将它置为 1, 否则为 0. 其对应图的最大度为 47, 如果不利用向后度, 需要运行 540 秒, 使用 (2, 3 维) 向后度, 运行时间为 10 秒. 将向后度改为极大完全可能值, 运行时间为 1 秒. 计算出的极大完全子图的阶数是 19. 又对一个 60×60 的 0-1 矩阵运算, 使用时间是不利用向后度为 988 秒, 利用向后度为 317 秒. 利用极大完全可能值为 99 秒. 这个 60 点图的最大度是 58. 计算出的极大完全子图的阶数是 27. 点集 $\{1\ 3\ 4\ 8\ 10\ 11\ 14\ 16\ 18\ 21\ 22\ 24\ 25\ 27\ 31\ 33\ 35\ 36\ 37\ 42\ 45\ 51\ 52\ 53\ 54\ 56\ 57\}$ 生成一个 27 点的极大完全子图.

参考文献:

- [1] F. Harary. Graph Theory [M]. Addison Wesley. Reading, MA, 1969.

(下转第 82 页)

期的过程,需要企业领导者和人力资源部门进行持续和有效的管理.③没有激励的创新是不能持久的,创新需要动力.激励有外部和内部之分,外部的激励主要来自物质奖励方面,这就要求建立一套公平、合理的奖励制度;内部的激励主要来自员工对某项活动的激情和兴趣,这需要一套方法、政策和文化氛围.④组织能取到保证作用.根据项目成立工作团队^[5],并建立团队工作业绩评估制度,同时给予支持、鼓励.

4) 实施科学的创新原则

一项创新的好坏,不仅在于该项创新的新颖性或技术含量的高低,而且在于商业上的成功性.所谓商业上成功是指:必须能够获得原始投资回报以及额外利润,简单地讲就是因为企业采用该项创新能为用户创造价值,也能为企业自身带来更大效益和竞争优势.Amidon^[6]把现代管理的关键要素的关系想象为:一个是知识,它由数据和信息演进而来,它提供描述内容的方式,需要对此加以管理;第二个是作为过程的创新,它由思想活动构成,而非技术本身的进步;增加知识的水准和确保通过创新过程取得商业成果的唯一途径是实时学习(Real Time Learning)的方法论.企业同时管理知识内容和创新

过程,价值的积累就会在企业内部建立起来.

创新平台中三个支撑点在一定程度上对竞争优势发挥积极作用,但因产业的不同,每个支撑点对竞争优势影响的程度均不一样.如目前电脑和家用电器的生产企业,服务对竞争优势能起关键的作用.对于象饭店、宾馆这样的企业,后勤可能在一定程度上不存在.对于批发商,后勤管理最重要.创新是创新过程的结果,这过程可以被定义为导向适销的新产品及服务以及新的生产及交货系统的综合活动^[2],但都必须围绕“价值”进行.

参考文献:

- [1] 彼得·F·德鲁克. 创业精神与创新[M]. 北京:工人出版社,1989.
- [2] 施培公. 后发优势—模仿创新的理论与实证研究[M]. 北京:清华大学出版社,1999.
- [3] 傅家骥. 技术创新学[M]. 北京:清华大学出版社,1998.
- [4] 罗伯特·A·伯吉尔曼—技术与创业的战略,1998 彭宜新译,华中理工大学管理学院.
- [5] 胡树华—国内外产品创新管理研究综述[J]. 中国管理科学 1999, (1).
- [6] Debra M·Amidon; Innovation Strategy for the Knowlwdge Economy—The Ken Awakening, Butterworth-Heinemann, 1997.

(上接第 72 页)

An Algorithm of Maxmum Clique on An Arbitrary Graph

ZHOU Shang-chao, LIU Er-gen, WANG Sen

(School of Natural Science, East China Jiaotong University, Nanchang 330013, China)

Abstract: By using the conception of point backward degree, the calculating of extremal great complete graph of graph is speeded up.

Key word: extremal great complete graph; point independence number; recursive function.