CN43-1258/ TP
ISSN 1007-130X

计算机工程与科学
COMPUTER ENGINEERING & SCIENCE

2008 年第 30 卷第 12 期
Vol. 30 ,No. 12 ,2008

文章编号:1007-130X(2008)12-0045-04

基于深度优先搜索的一般图匹配算法*

Matching Algorithms for General Graphs Based on Depth-First Search

虞成诚[1] ,钟　声[1] ,胡绍华[2]

YU Cheng-cheng[1] , ZHONG Sheng[1] , HU Shao-hua[2]

(1. 海南大学计算机科学与技术系,海南 海口 570228;2. 南昌广播电视大学,江西 南昌 330000)

(1. School of Information Science and Technology, Hainan University, Haikou 570228;
2. Nanchang Radio and TV University, Nanchang 330000, China)

摘　要:对于一般图的匹配问题,Edmonds 算法以 Berge 定理为基础,采用广度优先搜索增广路,图中可能存在“花”。遇到这种情况,要对它进行缩减“花”处理,再进行搜索。当找到增广路时,要将缩减图恢复,算法显得复杂。Gabow 等算法使用先给固的顶点和边编号,并使用了不同数组和虚拟顶点,避免了处理花。算法的复杂性为 $O(n^3)$,但增加了空间复杂性。本文提出的基于深度优先搜索算法,在搜索增广路时不会出现“花”的情况,算法相对简单;同时,算法时间效率为 $O(n * degree(n))$, $degree(n)$ 为顶点的平均度数。另外,当图的边动态增减时,使用该算法可以很快调整最大匹配,并且该算法空间复杂性在同一数量级也可以推广到广度优先搜索。

Abstract: For the matching problem of general graphs, the Edmonds algorithm is usually used. When the breadth-first search algorithm is used to find an augmenting-path, a “flower” may be produced. In order to solve this problem, we can cut short the “flower” into a point, and keep on searching, then the obtained graph after finding the augmenting-path is rehabilitated. The time efficiency of the algorithm is $O(n^4)$ since the graph is cut short and rehabilitated. However, when using the Gabow algorithm to find the augmenting-path, we should assign a fixed serial number to node and edges, and use different arrays and pseudo node, thus we need not to deal with the “flower”, and the time efficiency of this algorithm is $O(n^3)$, but the space efficiency of the algorithm is reduced. This paper proposes a depth-first search algorithm, and it is shown that using the proposed algorithm, the "flowers" will not appear when searching an augmenting-path. Moreover, the time efficiency is increased to $O(n * degree(n))$, where $degree(n)$ denotes the average degree of nodes. In addition, when either appending or deleting the edges in the graph, maximum matching can be computed quickly. It should be pointed out that if we follow this idea to consider the breadth-first search, the order of magnitude is identical to the one in depth-first search.

关键词:匹配问题;组合优化算法;图匹配

Key words: matching problem; combinatorial optimization algorithm; graph matching

中图分类号: TP301.6　　**文献标识码**: A

1　引言

一般图 $G=(V, E)$,其中 V 是顶点的集合,E 是边的集合。图的匹配问题就是要找到边集的一个最大不相交子集,即所有边子集中边数最多且任意两条边都没有公共顶点的集合。该问题在工程上有广泛的应用。

一般把图的一个不相交子集称为图的一个匹配,最大不相交子集称为图的最大匹配,包含图的所有顶点的匹配称为完全匹配。显然,完全匹配就是最大匹配。一般把在一个匹配集合里的边称为匹配边,其它边称为自由边;同样,把在一个匹配集合里的边的端点称为饱和点,其它顶点称为自由点(非饱和点)。

构造一般图的最大匹配问题算法,比较典型的是 Ed-

* **收稿日期**:2008-08-18;**修订日期**:2008-09-10
基金项目:海南省自然科学基金资助项目(80636);海南省教育厅资助项目(Hjkj200705)
作者简介:虞成诚(1974-),男,江西九江人,硕士生,研究方向为算法理论与软件方法、图论及网络的计算机辅助设计;钟声,教授,研究方向为算法理论与软件方法、图论及网络的计算机辅助设计。
通讯地址:570228 海南省海口市海南大学信息科学技术学院;Tel:13136029997;E-mail:shzhong@hainu.edu.cn
Address: School of Information Science and Technology, Hainan University, Haikou, Hainan 570228, P. R. China

monds[1]提出的构造匹配的基本方法和 Gabow[2]对 Edmonds 方法的有效实现算法。其它一些有效的标号方法[4~6]所使用的方法与 Gabow 相似,算法复杂性可达到 $O(n^{5/2})$。

Edmonds 算法以 Berge 定理为基础。Berge 定理指出某一匹配是最大的,当且仅当不存在相对于此匹配的增广路径。这样,给定了一个图和一个初始匹配:找出一条相对于该匹配 M 的增广路径 P,就得到此多一条边的匹配 $M \oplus P$。对于这个新的匹配,找出一条增广路径,处理方法同前。如此继续下去,直至得到一个不存在增广路径的匹配为止。由 Berge 定理可知,这个匹配是最大的。

Edmonds 算法最重要的概念就是"花":如果在搜索可选集中某个顶点时,发现此顶点与另一个已在可选集的顶点相邻接,这就产生了一个奇回路(具有奇数长度的回路)。这个具有奇数长度的闭合交错路径叫做花。一旦生成了花,就用一个单顶点来代替花中的所有顶点,这个新顶点叫做伪顶点,与花中一个或多个顶点相邻接的所有顶点都与这个新顶点相邻接,可以得到一个缩减图。在这个缩减图上,继续寻找增广路径。当又找到一朵花时,就重复这个过程。如果最后在某个缩减图上找不到一个增广路径,那么就意味着原始图中,相对于当前匹配的增广路径也不存在。于是,当前匹配是最大的。如果在某个缩减图中找到一条增广路径,那么意味着在原始图中也存在一条增广路径。为了求出路径 P,需要通过展开前面所得到的花来回溯。算法中花的收缩和展开的时间开销,使这个算法的复杂性为 $O(n^4)$。这里,n 是图中顶点的数目。

Gabow 等算法使用先给固的顶点和边编号,并使用了不同数组和虚拟顶点,避免了处理花。算法的复杂性为 $O(n^3)$,但增加了空间复杂性。

本文提出的算法基于图的遍历搜索算法,使用不同的遍历搜索标记来完成增广路的搜索。

2 搜索算法

在介绍搜索方法前,介绍两个术语:

(1)交错路:一条路径中,匹配边和自由边交叉出现,即匹配边的相邻边是自由边,自由边的相邻边是匹配边,并且路径的匹配边和自由边数目相等。

(2)增广路:一条路径中,匹配边和自由边交叉出现,并且路径的起点和终点均为自由点。

为了搜索增广路,我们将图的搜索算法中使用的访问标记增加到五个,不同的标记决定下一步搜索方法,使用深度优先搜索,不会出现"花",提高了算法的时间效率,使用的中间变量也与一般图的遍历算法相当。

我们用 $f_i(1 \le i \le n)$ 表示顶点 i 的搜索标记,不同值表示的意义为:f_i 为 -2,表示没有访问过的顶点;f_i 为 -1,表示自由点;f_i 为 0,表示饱和点,进入方向未搜索;f_i 为 1,表示饱和点,进入方向已搜索;f_i 为 2,表示不能匹配自由点。

对于某一匹配 M 中的匹配边 e_{ij},在搜索增广路时,如果先访问顶点 i,接着访问顶点 j,则称顶点 i 为进入方向搜索,顶点 j 为离开方向搜索。初始设置 $f_i = -2(1 \le i \le n)$,初始匹配 $M_0 = \Phi$。

用图的深度优先搜索时,如果某一顶点的邻接点标记为 -2,则修改该邻接点的标记为 -1,将该邻接点变成当前顶点继续往下搜索,在往下搜索前,如果当前顶点与前一顶点的标记均为 -1,则将它们的标记均修改为 0,并将以这两个顶点为端点的边加入到初始匹配集合。

我们不妨假设处理的图是连通的,则算法 1 可以遍历所有顶点,并且构造初始匹配 M_0,$f_i = -1(1 \le i \le n)$ 的点是自由点,$f_i = 0(1 \le i \le n)$ 的点是饱和点。

算法 1 深度优先搜索算法构造初始匹配 M_0

```
Set & InitMatching() //返回初始匹配(集合) M0
{ M0 = Φ;for (i = 1; i ≤n; i + +) f i = - 2; //初始化
S[0] = 1; f S[0] = - 1; i = 0;//任取一顶点记为 s[0]
while (i ≥0)
if exist( f j = - 2 and e s[i]j ∈E)
//如果存在顶点 j 没有遍历过且是顶点 s[i]的邻接点
{ i = i + 1; S[i] = j;
if (f S[i - 1] = - 1)
{ f S[i - 1] = 0; f S[i] = 0; M0 = M0 ∪e s[i - 1]s[i];}
else f S[i] = - 1;}
else i = i - 1;
return M0;
```

在搜索增广路时,从某一自由顶点出发,修改该顶点的标记为 2,将它作为当前顶点,搜索与其邻接的且标记为 0 的邻接饱和顶点,将该顶点的标记修改为 1,再找到这个饱和顶点所在匹配边的另一个饱和顶点,将最后找到的这个顶点作为当前搜索出发点。如果当前顶点存在标记为 -1 的邻接点,则找到一条增广路,否则继续往下搜索。当前顶点没有标记为 0 或 -1 的邻接点时,回溯到搜索路径中前两个顶点,继续往下搜索,当回溯到初始出发的自由顶点都找不到合适的邻接点时,可以确定该自由顶点作为初始出发点的增广路径不存在。搜索增广路算法见算法 2

算法 2 搜索增广路

```
int IncMatching(Sequence & IncPath, Set & M)
{ //返回增广路顶点序列 IncPath,函数值为 0,成功
//搜索到增广路,否则,搜索失败
if exist( f k = - 1) //存在 k,满足 f k = - 1
{ S[0] = k; f k = 2; i = 0;
while (i ≥0)
{ if exist( f j = - 1 and e s[i]j ∈E)
//如果存在 s[i]的自由邻接点 j 则找到增广路
{ i = i + 1; S[i] = j; f i = 0;
IncPath = s; IncPath. length = i + 1;
return 1;
}
if exist( f j = 0 and e ij ∈E and e jk ∈M)
//如果存在 s[i]的饱和邻接点 j 且没有从
//进入方向搜索过,e jk是当前匹配 M 的边
{ i = i + 1; s[i] = j; i = i + 1; s[i] = k; f j = 1;
}
else i = i - 2;
}
}
else return 0;
}
```

图匹配算法见算法 3。

算法 3 匹配问题算法

```
Matching()
{ M = InitMatching(); //构造初始匹配
while (IncMatching(Sequence &IncPath, Set & M) = 0)
{//只要找到增广路,调整匹配,匹配边数增加一条
i = 0; M = M ∪e IncpathIncpath; f i = 0;
for(i = 1; i < IncPath. length - 2; i + = 2)
{ M = M - e IncpathIncpath;
M = M ∪e IncpathIncpath;
```

```
f_i = 0; f_{i+1} = 0; }
for(i = 1; i ≤ n; i + + ) if (f_i = 1) f_i = 0;}
}
```

3 几个定理

定理 1 从饱和点 i 进入方向搜索不到终点为自由点的交错路,则不存在包含 i 点进入方向搜索的增广路。

证明 假设存在包含 i 点进入方向搜索的增广路,因为增广路的两个端点均为自由点,所以存在从饱和点 i 进入方向搜索的交错路,这与已知条件矛盾,定理成立。 □

这个定理说明,在搜索增广路时,匹配边的两个端点只要分别在进入方向搜索一次且必须搜索一次,就能决定该匹配边是否能够作为当前搜索增广路的一条边。

定理 2 从自由点 i 出发搜索不到增广路,则其它自由点 j 出发找到增广路调整匹配后,也不存在从 i 点开始的增广路。

证明 假设从其它自由点 j 出发找到增广路调整匹配后,存在从 i 点开始的增广路。令 $(i_0, i_1, \cdots, i_m)$ 和 $(j_0, j_1, \cdots, j_n)$ 分别为从 i 和 j 出发的增广路的顶点序列,两条路一定有公共边,否则找到增广路调整匹配将互不影响。设 $j_s = i_k$ 是它们的第一个公共顶点,在调整匹配前,边 (i_{k-1}, i_k) 不可能是匹配边,因为如果边 (i_{k-1}, i_k) 是匹配边,则边 (j_{s-1}, j_s) 和边 (j_s, j_{s+1}) 一定不是匹配边。根据交错特性,这样 $(j_0, j_1, \cdots, j_n)$ 就不是增广路,与假设矛盾。因此,边 (j_{s-1}, j_s) 和边 (j_s, j_{s+1}) 只有一个是匹配边,如果边 (j_{s-1}, j_s) 是匹配边,则 $(i_0, i_1, \cdots, i_k, j_{s-1}, \cdots, j_0)$ 是增广路,如果边 (j_s, j_{s+1}) 是匹配边,则 $(i_0, i_1, \cdots, i_k, j_{s+1}, \cdots, j_n)$ 是增广路,这与已知条件矛盾,定理成立。 □

该定理说明,只需要一次从自由点 i 出发搜索增广路,其它增广路对匹配的调整将不会影响从自由点 i 出发的增广路的存在性。

推论 如果增广路存在,本文提出的算法一定可以搜索到增广路。

证明 根据图的遍历搜索算法,一定可以访问图的所有顶点,由定理 1 和定理 2 可知,如果增广路存在,则一定可以搜索得到,证明了算法的正确性。 □

4 算法扩展

当图的顶点或边发生添加或删除操作时,使用我们提出的算法可以很快计算出新的最大匹配。下面分几种情况讨论:

(1) 添加一个顶点。当添加一个顶点和一些与该顶点相关的边时,添加的顶点是自由点。从这点出发,搜索增广路,当增广路存在时,调整最大匹配即可。由定理 2 可知,调整匹配后,不会影响其它自由点出发的增广路的存在性。

(2) 删除一个顶点。当删除顶点为自由点时,不影响最大匹配。删除的顶点是饱和点时,将与删除点所在匹配边的另一个饱和点变成自由点,从该自由点出发,搜索增广路即可。

(3) 添加一条边。当添加边的两个端点均为自由点时,将添加的边放到匹配集合即可。当添加边有一个端点为饱和点,一个端点为自由点时,从饱和点出发,按进入方向搜索终点为自由点但不是添加边的自由端点的交错路,如果存在则找到从添加边的自由端点出发的增广路。当添加边的两个端点均为饱和点时,从这两个饱和点出发,按进入方向分别搜索终点不是同一自由点的交错路,如果两个交错路都存在,则找一条增广路。

(4) 删除一条边。当删除的边是自由边时,对最大匹配没有影响。当删除的是匹配边时,将这条边的两个端点变成自由点,从这两个自由点出发分别搜索增广路即可。

将该算法思想用到广度优先搜索,标记为 0 的饱和点可以往下搜索。这样,每条匹配边允许在搜索树上出现两次,即从两个端点的进入方向进行搜索。

5 算法实例

图 1a 是给定需要计算最大匹配的原始图,采用求初始匹配的搜索算法,搜索树如图 1b,得到初始匹配结果如图 1c。这时,5 和 8 顶点是自由点,边 a、c、g 构成初始匹配,其它顶点为饱和点,其它边为自由边。从顶点 5 出发,使用本文提出的增广路搜索算法,搜索树如图 1d 所示,搜索过程从左分支到右分支进行深度优先搜索,最后得到增广路 5-3-4-6-8,将增广路上自由边变成匹配边,匹配边变成自由边,得到调整后的匹配 $\{a, e, f, h\}$ 也是这个图的最大匹配和完全匹配,如图 1e 所示。

图 1f 使用 Edmonds 算法出现"花"的情况,也是从自由点 5 出发进行广度优先搜索,定点 5、3、4 构成奇回路,将其缩减成一个顶点 H(也即是"花"),从 H 出发继续搜索可以得到增广路 H-6-7-8,将 g 变成自由边,f、h 变成匹配边后,再展开"花"。可以看出,还要将边 c、e 的自由边和匹配边性质对调,即得到最大匹配。

如果将本文提出的思想用广度优先搜索寻找增广路,对于图 1a 原始图和图 1b 得初始匹配,从顶点 5 出发进行搜索,得到的搜索树与图 1d 完全一样。

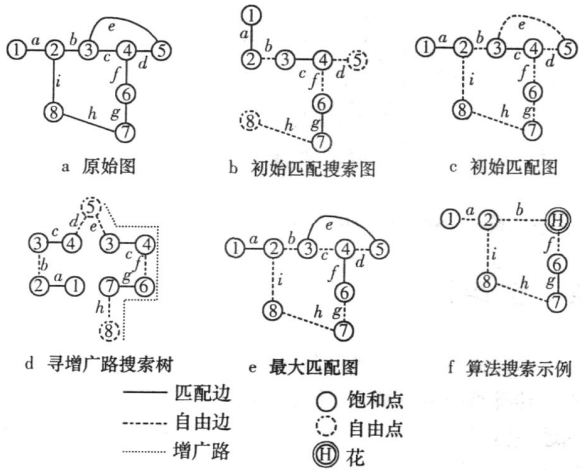

图 1 算法搜索示例

6 结束语

本文提出的基于深度优先搜索算法在搜索增广路时不

会出现“花”的情况,算法相对简单。当搜索初始匹配得到的搜索树是一线性链时,初始匹配是最大匹配。算法时间效率与深度优先搜索遍历算法是同一数量级,为 $O(n * degree(n))$, $degree(n)$ 为顶点的平均度数。另外,当图的边动态增减时,使用该算法可以很快调整最大匹配,并且该算法很容易推广到广度优先搜索。

参考文献:

[1] Edmonds J. Paths, Trees and Flowers[J] Canadian. Journal of Mathematics, 1965,17:449-467.

[2] Gabow H N. An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs[J]. Journal of the ACM,1976,23(2):221-234.

[3] Bajinski M L. Labelling to Obtain a Maximum Matching[M]// Bose R C, Dowling T A, eds Combinatorial Mathematics and Its Applications. Chappel Hill, N. C University of North Carolina Press,1967:585-602.

[4] Witzgall D, Zahn C T Jr. Modification of Edmonds' Algorithm for Maximum Matching of Graphs[J]. Journal of the Research National Bureau of Standards,1965,69B:91-98.

[5] Kameda T, Munro I. A $O(|V| * |E|)$ Algorithm for Maximum Matching of Graph[J]. Computing, 1974,12:91-98.

(上接第 22 页)

因在于 ILP 和 DGA 未针对指定的覆盖度进行优化。

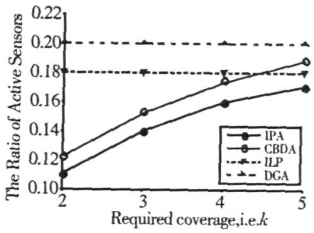

图 3 与 ILP、DGA 在不同覆盖度下激活节点比例的比较结果

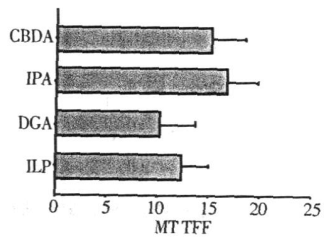

图 4 ILP、DGA、IPA 和 CBDA 的 MTTFF 比较

6 结束语

本文针对感知方向可调的有向感知模型,首次提出了有向传感器网络中基于概率感知模型的最小连通 k 覆盖集问题,并设计了基于 0-1 整数规划和最小生成树的集中式近似算法(IPA)和基于覆盖效益探测的分布式近似算法(CBDA),分别证明两种算法最终得到的是 MCKS 问题的可行解,并分析了算法的时间复杂度、性能比和通信复杂度。通过仿真实验并与 ILP 算法和 DGA 算法进行比较的结果表明,在基于概率感知模型的条件下,IPA 和 CBDA 能够有效实现 DSN 中的连通 k 覆盖,并且激活节点数目较少,从而延长了网络寿命。以后我们的工作将试图引入概率通信模型,进一步考虑有向传感器网络中同时基于概率感知模型和概率通信模型的最小连通 k 覆盖集问题。

参考文献:

[1] Ren Y, Zhang S D, Zhang H K. Theories and Algorithms of Coverage Control for Wireless Sensor Networks [J]. Journal of Software, 2006,17(3):422-433.

[2] Hefeeda M, Bagheri M. Randomized k-Coverage Algorithms for Dense Sensor Networks [C]// Proc of INFOCOM'07, 2007:2376-2380.

[3] Yang S, Dai F, Cardei M, et al. On Connected Multiple Point Coverage in Wireless Sensor Networks [J]. Journal of Wireless Information Networks, 2006,13(4): 289-301.

[4] Anthony S, Yinyu Ye. On Solving Coverage Problems in a Wireless Sensor Network Using Voronoi Diagrams [C]// Proc of WINE'05,2005:584-593.

[5] Cai Y, Lou W, Li M, et al. Target-Oriented Scheduling in Directional Sensor Networks[C]// Proc of INFOCOM'07, 2007: 1550-1558.

[6] Hekmat R, van Mieghem P. Connectivity in Wireless Ad-Hoc Networks with a Log-Normal Radio Model [J]. Mobile Networks and Applications, 2006,11(3): 351-360.

[7] Wu Y A, Li M, Cai Z P, et al. A Distributed Algorithm to Approximate Node-Weighted Minimum α-Connected (θ, k)-Coverage in Dense Sensor Networks[C]// Proc of International Frontiers of Algorithmics Workshop,2008.

[8] Ai J, Abouzeid A A. Coverage by Directional Sensors in Randomly Deployed Wireless Sensor Networks [J]. Journal of Combinational Optimization,2006,11(1): 21-41.

[9] He T, Huang C, Lum B, et al. Range-Free Localization Schemes for Large Scale Sensor Networks [C]// Proc of MOBICOM'03,2003: 81-95.

[10] Ganeriwal S, Kumar R, Srivastava M B. Network-Wide Time Synchronization in Sensor Networks [R]. Technical Report, NESL,2003.

[11] Jiang J, Fang L, Zhang H Y, et al. An Algorithm for Minimal Connected Cover Set Problem in Wireless Sensor Networks [J]. Journal of Software,2006,17(2):175-184.

[12] Ye Y. An $O(n^3 l)$ Potential Reduction Algorithm for Linear Programming [J] Mathematical Programming, 1991,50: 239-258.