一种求解平面图的最小顶点覆盖算法①

吴 春 朱国魂 谢玉忠 林 宏 (桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘 要: 最小顶点覆盖问题是图论中经典的组合优化问题，在实际生活中有着广泛的应用价值。根据最小顶点覆盖与最大独立集在图论中事实上是属于等价问题这一特性，从最大独立集的角度出发，根据最大独立集的特性，设计了一种求解简单平面图的最大独立集算法，从而求出最小顶点覆盖。通过实验结果的比对验证算法的正确性和有效性。

关键字: 点覆盖；最小顶点覆盖；最大独立集；平面图；邻域

Minimum Vertex Cover Algorithm for Solving the Planar Graph

WU Chun, ZHU Guo-Hun, XIE Yu-Zhong, LIN Hong

(School of Computer and Control, Guilin University of Electronic and Technology, Guilin 541004, China)

Abstract: The minimum vertex cover problem is a classical problem in combinatorial optimization and has a wide range of values in real life. The problem of the minimum vertex cover and the maximum independent set is actually the equivalence problem in the graph theory. In this paper, the maximum independent set angle and design a maximum independent set algorithm for solving the planar graph is formed, according to characteristics of the maximum independent set, and then the minimum vertex cover is obtained. After many experiments, the results show adequately that the algorithm presented in the thesis is correct and effective.

Keywords: vertex cover; minimum vertex cover; maximum independent set; planar graph; epsilon neighborhood

1 引言

点覆盖问题是一个被广泛研究的问题，国内外研究十分活跃。尽管它的复杂性大，但是在生活当中确是有着广泛的应用。例如，如何用最少的点来监控一个大型的网络，超大规模集成电路芯片的缺陷修复，无线传感器网络节点的布置，分布式控制网络节点的布置等等。

点覆盖是一个经典的 NP 完全性问题[1]，有限顶点、无向图求最小覆盖问题是最早被提出的著名 NPC 问题之一[2]，然而其问题的求解至今还困扰着人们。目前点覆盖的主要研究方向主要集中在两个方面。一个是通过智能计算方法来获取最优近似解集，另一个是发掘和采用更优化的分支限界技术，来提高对问题的核的处理速度。在点覆盖的核心化算法中，NT 算法和皇冠分解一直是两种主要的技术[3]。

最小点覆盖问题、最大独立集问题与最大团问题三者事实上是等价的。这三个问题中任何一个获得的结果都可以立即转化为其他两个问题的等价形式[4]，即，最小点覆盖与最大团等价形式：$a(G)=w(\overline{G})$，最小点覆盖和最大独立集等价形式：$a(G)+b(G)=n$。由此，只要解决三者中的任意一个问题，其他两个问题都可以根据等价形式解决出来。

本文在求解最小覆盖算法从求解最大独立集着手。在一个图中，独立集是在一个点集合中不存在任意两个相连接的顶点。在图论和组合数学中，独立集问题往往就是为了寻找一个基数最大的独立集。大多

① 基金项目:广西区教育厅立项项目(200911LX114)
收稿时间:2010-01-04;收到修改稿时间:2010-01-27

数独立集研究学者，考虑的是通过将现代优化算法引入最大独立集问题中[5]，如人工神经网络算法[6]，模拟退火算法[7]，贪婪算法[8]等等，但是往往求出的是极大独立集，而并非最大独立集。

2 相关研究

定义 1.设 $G(V,E)$ 是简单无向图，$S \subseteq V$ $S \neq \Phi$ ，。E 若中每条边都与 S 中某点关联，则称 S 为 G 的点覆盖(vertex covering)。如果 G 中的任何异于 S 的点覆盖 S'，均有 $|S'|>|S|$，则称 S 为 G 的最小点覆盖。

定义 2.设 S 是简单无向图 $G(V,E)$ 的顶点集 $V(G)$ 的一个非空子集，若 S 中任意两个顶点在 G 中均不邻接，则称 S 为 G 的一个独立集。若 S 是图 G 的独立集，但是任意增加一点就会破坏它的独立性，则称这个独立集 S 为 G 的极大独立集。如果不存在独立集 S'，使 $|S'|>|S|$，则称 S 为 G 的最大独立集，其中 $|S|$ 为集合 S 的基数。

本文中需要用到的表示符号如下。

$G(V,E)$：简写为 G，表示一个图，其包含一个顶点集 $V(G)$，一个边集 $E(G)$。$|G|$ 表示 $V(G)$ 中顶点的数目，$|E(G)|$ 表示 $V(G)$ 中边的数目。

MinVC：最小点覆盖。G 中最小覆盖点的个数称为点覆盖数，记为 $a(G)$。

MaxIS：最大独立集。G 中最大独立集的点个数称为点独立数，记为 $b(G)$。

$d(v)$：简写为 d，记为顶点 v 的度数，即顶点 v 的边数。

VN1，VN2：邻域集合。顶点 V_i 的邻域是指与顶点 V_i 相邻的顶点构成的集合。

$A(G)$：简写为 A，表示平面图 G 的邻接矩阵。

一个有 n 个顶点的简单图 G，由一个顶点集合 v 和边集合 E 组成，其中 $|v|=n$，任意一条边 e 都是由无序对的截然不同的顶点组成。值得注意的是，这里 G 的定义说明了图中没有环路，重边，而且边集合是有限的。我们可以用整数 1, 2, ..., n 来标识图中的顶点。如果无序对顶点 $\{u,v\}$ 是 G 中的一条边，则 u 是 v 的一个邻接顶点，写成 $uv \in E$。邻接顶点有个很明显的对称关系，$uv \in E$ 当且仅当 $vu \in E$。$d(v)$ 是顶点邻接顶点 v 的个数，最大的 $d(v)$ 表示为 Δ。G 中的独立集 S 表示为一组没有无序对顶点构成一条边的集合。顶点覆盖 C 是指 G 中边 $\{u,v\}$ 至少有一个顶点 u 或 v 在集合 C 中，其中如果 $C-\{v\}$ 的集合仍是 G 的覆盖集合，则 v 是可移动的。

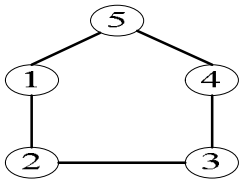

图 1　一个简单无向图

如图 1 所示，是一个顶点数为 5 的简单无向图，其中包含五条边，即{{1 2} {1 5}{2 3}{3 4}{4 5}}。假如一个顶点集合 S 为{1 2 3 4}，则图中的边必与 S 中的某个顶点相关联，例如边{4 5}与 4 关联，所以 S 被称为点覆盖。所谓最小顶点覆盖就是用最少的顶点去关联图中所有的边，这里的最小是指顶点覆盖集合的基数，也是就集合中元素的个数，显然集合{1 2 4}、{1 3 4}、{2 4 5}、{1 3 5}都是最小顶点覆盖。独立集是指集合中的所有顶点互不相连，从图 1 可知，{3 5}、{2 5}、{1 3}、{2 4}都是独立集。当我们将 1 或 2 或 4 放入{3 5}，都会导致这个独立集被破坏，同样的情况也发生在其他 3 个独立集，所以这 4 个都是最大独立集。通过比对最小顶点覆盖和最大独立集，可以发现{1 2 4}+{3 5}={1 2 3 4 5}，即等价公式：$a(G)+b(G)=n$，所以求出两个问题中的任何一个都可以通过等价公式求出另一个问题。

3 算法分析

3.1 现有的算法介绍

一个最小覆盖顶点集合就是不包含可移动的顶点[9]，文献中作者 Ashay Dharwadker 就是用上述的思想来解决最小覆盖顶点问题。具体算法介绍，当 n=i 顶点时，初始化覆盖顶点 $C_i=V-\{i\}$。在 C_i 中逐次寻找个个元素的可移动顶点，并且比较可移动顶点的个数输出个数最大的顶点，假设这个顶点是 j。修改覆盖顶点 $C_i=V-\{i,j\}$，在 C_i 中逐次寻找个个元素的可移动顶点，并且比较可移动顶点的个数输出个数最大的顶点。如此循环，直到 C_i 中没有可移动顶点，这个时候输出 C_i，C_i 就是最小覆盖顶点。

3.2 本文的算法介绍

一个简单无向图 $G(V,E)$，$A(G)$ 是 G 的 n*n 邻接矩阵，矩阵中 1 表示顶点 u 和顶点 v 相连，0 表示不

相连。根据独立集的定义，相连接的顶点 u 和顶点 v，不可能同时出现在独立集 S 中，因此定义了两个邻域集合 VN1 和 VN2。一个简单无向图 G 中，一个顶点 v 的 d 可以是在区间$[1,n-1]$中的任意整数，可以根据邻接矩阵中的每行中元素 1 的个数来得知 d。假设 A 中每行每列都有标识符 1，2，...，n，其中 i 表示图中第 i 个顶点。首先，判断顶点 i 是否在邻域集合 VN1 或 VN2 中，有四种情况：i 在两集合都不存在； i 在 VN1 中； i 在 VN2 中； i 在两集合中都存在。根据不同的情况，对 VN1 和 VN2 做不同的操作。如此循环，直到所有顶点判断完，然后在 VN1 和 VN2 集合中找出具有相同值的元素，删除。最后，比较$|VN1|$和$|VN2|$的大小，如果$|VN2|$大，则将 G 中顶点数$V-VN2$，得到的就是最小覆盖顶点集合。其中要注意两点，在将 i 的邻域放入邻域集合 VN1 或 VN2 时，如果 VN1 或 VN2 中存在要被放入的元素，则不用放入元素。另一点，邻接矩阵存在特性：$A=A^T$，A 被称为是对角矩阵，则在循环的时候，顶点 i 内的循环可以从 i+1 开始，在 n 结束。

3.2.1 算法具体实现步骤

For i=1, 2, ..., n in turn

Step1 判断 i 是否属于 VN1 或 VN2。

Step2 如果不属于 VN1 和 VN2，则将 i 放入 VN1，将 i 的领域放入 VN2；如果属于 VN1，则将 i 的领域放入 VN2；如果属于 VN2，则将 i 的领域放入 VN1；如果都属于 VN1 和 VN2，则直接跳过，进入下一个循环。

Step3 找寻 i 顶点中的领域，并根据 step2 的原则，将 i 和 i 的领域放入 VN1 和 VN2。

Step4 循环完毕，将 VN1 和 VN2 中相同的元素删除。

Step5 比较 VN1 和 VN2 集合中元素个数大小。

Step6 将 G 中的顶点集合减去 step5 比较出来的结果集合，输出结果就是最小覆盖顶点集合。

3.2.2 算法举例说明算法举例说明

如图 2 所示，是一个 n=6 的八面体图（the Octahedron Graph）。图中标有颜色的点表示最小覆盖顶点，也就是说这个八面体图的最小覆盖顶点为 4。

图中的邻接矩阵 A={{0 1 1 0 1 1}, {1 0 1 1 0 1}, {1 1 0 1 1 0}, {0 1 1 0 1 1}, {1 0 1 1 0 1}, {1 1 0 1 1 0}}，调用本算法步骤如下：

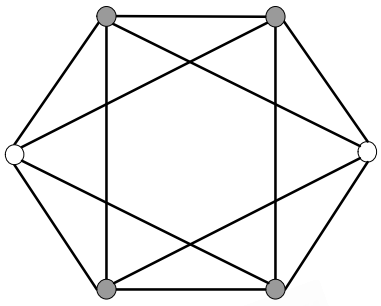

图 2 the Octahedron Graph

i=1：判断 i 是否属于 VN1 或 VN2；执行判断之后，VN1={1}，VN2={2,3,5,6}。

i=2：判断 i 是否属于 VN1 或 VN2；执行判断之后，VN1={1,3,4,6}，VN2={2,3,5,6}。

i=3：判断 i 是否属于 VN1 或 VN2；因为 3 在两个集合中都有，所以 3 的邻域情况不考虑。

i=4：判断 i 是否属于 VN1 或 VN2；执行判断之后，VN1={1,3,4,6}VN2={2,3,5,6}。

i=5：判断 i 是否属于 VN1 或 VN2；执行判断之后，VN1={1,3,4,6}VN2={2,3,5,6}。

判断 VN1 和 VN2 相同的元素，把 3 和 6 删除，则 VN1={1,4}，VN2={2,5}。

输出 Vertex Cover (4): 2 3 5 6 ，其中括号中的 4 表示最小顶点覆盖数。

4 测试结果与算法分析

算法的仿真是在 windows xp 系统下进行的，硬件配置为 Intel Pentium 3.0GHz，1G 内存，软件开发平台是 Visual Studio2005。数据来源文献[9]，运行结果见表 1。

表 1 中每个算法都有两栏运行时间，这些运行时间都是每个图在经过十次运行之后出现次数最多的。当图的顶点数较小时，算法的运行时间偶出现 0ms 或 15ms，基本上两个运行时间的差值在 15ms。

在运行现有算法的程序时，我们发现该算法必须输入一个由人为判断的覆盖数 K，如果输入的 K 值小于输出的最小覆盖数，则程序一直重复地输出 the vertex cover size，这时你需要重新输入 K 值。如果 K 值大于输出的最小覆盖数，在顶点数 n 较大的时候会出现另一个最小覆盖数。例如 Benchmark Graph1，如果输入 K 值是 1 到 40 的任意一个整数，

表 1　与现有算法运行结果时间的比对

图 G 名称	顶点数	文献[9]运行时间/ms		最小点覆盖数	本文算法运行时间/ms		最小点覆盖数
Octahedron Graph	6	0	0	4	0	0	4
Cube Graph	8	0	15	4	0	0	4
Petersen Graph	10	0	15	6	0	0	6
Icosahedron Graph	12	0	16	9	0	0	9
Ramsey Graph	17	16	16	14	0	0	14
Folkman Graph	20	47	62	10	0	16	10
Dodecahedron Graph	20	47	62	12	0	16	12
Tutte-coxeter Graph	30	172	172	15	15	15	15
Thomassen Graph	34	235	250	20	16	16	20
Benchmark Graph1	60	1188	1203	40	16	31	41
		984	969	41			
Benchmark Graph2	450	167297	168187	423	78	78	426
		78735	80778	424			

则输出的最小覆盖数是 40；如果输入 K 值是 41 到 60 的任意一个整数，则输出的最小覆盖数是 41，而在文献[9]中作者给出的最小覆盖数是 40。同样的情况也出现在 Benchmark Graph2，如果输入 K 值小于 423，则得出的最小覆盖数是 423；如果输入 K 值大于 424，则得出的最小覆盖数是 424，而在在文献[9]中作者给出的最小覆盖数是 420。因此，我们猜想该算法在顶点数 n 较大的情况下，会出现两个或多个最小覆盖数。

本算法同样在输入邻接矩阵之后，不需要输入一个由人为判断的覆盖数 K 就可以直接得出结果。从表 1 中可以看出在顶点数 n 较小时，得出的最小覆盖数是相同的。但是，在 n 较大的情况下，也许会出现极小的误差，如在 Benchmark Graph2 误差为 0.78%。

5　结束语

通过表 1 的数据可以直观的看出，所提出的算法在时间上的优势，而且在所求顶点数越大，所花费的时间对比越明显。但是在顶点数较大的情况下，会有很小的误差，下一步的工作将分析导致误差的原因，改进所提出的算法。

参考文献

1 Lu LM, Wang RS, Li WG. A target detection method in range-Doppler domain from SAR Echo data. Proc. of the 16th Int’l Conf. on Pattern Recognition (ICPR). 2002,1:91－94.

2 张秀梅. 偶图求最小覆盖的一种算法. 辽宁工学院学报, 2007,27(2):132－135.

3 常乐. 参数化点覆盖及最小点覆盖问题研究[硕士学位论文].长沙:中南大学, 2007.

4 Bondy JA, Murty USR, 吴望名等译.图论及其应用.北京:科学出版社, 1984.

5 李勤丰.最大独立集问题的一类进化算法研究[硕士学位论文]. 南京:河海大学, 2007.

6 卓仲畅,刘玉峰.用神经网络研究图的最大独立集问题.东北重型机械学院学报,1995,19(4):326－329.

7 王知人, 刘玉峰. 图的最大独立集问题的模拟退火算法. 东北重型机械学院学报,1997,21(3):274－277.

8 李勤丰,李尤丰,丁根宏.一个新的极大独立集算法及独立数的界.计算机工程与应用,2008,44(26):48－50.

9 Dharwadker A. The Vertex Cover Algorithm. 2006. http://www.geocities.com/dharwadker/vertex_cover/