

基于三角环的顶点着色问题解法

龚卫华 王元珍

(华中科技大学计算机学院 武汉430074)

摘要 图的着色问题是一个NP难问题,本文着重探讨无向图的顶点的三色问题,提出了用构造三角环的极大独立集方法判断并尝试给出顶点三色问题的可行解,解决了顶点三色的可满足性问题,克服了以前图遍历过程中的回溯问题,以及由此推论顶点四色和五色问题的极大独立集。

关键词 NP完全问题,三角环,极大独立集

A Solution of Vertices Coloring Problem Based on Triangle Clique

GONG Wei-Hua WANG Yuan-Zhen

(School of Computer Science & Technology, HuaZhong University of Science and Technology, Wuhan 430074)

Abstract The coloring problem of graph is a NP-hardness problem. This paper mainly discussed the vertices 3-coloring problem of undirected graph. Proposed a method of using the maximum independent set of constructed triangle clique to judge and present the feasible resolution of the vertices problem. Solved the Satisfiability of the vertices 3-coloring problem. Overcame the previously recursive problem during travelling graph. And deduced the maximum independent set of four-colorable and five-colorable problems from above.

Keywords NP-completeness problem, Triangle clique, Maximum independent set

1 图的NP难问题

已知图 $G=(V, E)$, 其中 $|V|=n, |E|=m$, 对图 G 的所有顶点进行着色时, 要求相邻的两顶点的着色不一样, 问至少需要多少种颜色? 这就是顶点的着色问题。对图 $G=(V, E)$ 的顶点进行着色的结果就是把顶点集 V 划分成若干个不相交的子集, 而且每一子集中的任意两点都不相邻, 这样的集合叫做独立集。设图 G 的某一顶点集合是独立集, 但是任意增加一点就破坏它的独立性, 则称这个独立集是极大的独立集。图中顶点个数最多的极大独立集叫最大独立集, 其顶点数叫做图的独立数。

此外, 与图相关的NP问题有最大圈问题MQ(Maximum Clique), 最大独立集问题MIS(Maximum Independent Set)和最小顶点色数问题MC(Minimum(vertex)Coloring), 对于普通图来说是NP难的, 而对于区间图却是在多项式时间里可解的。

命题1 顶点着色问题是NP完全的(NP-Completeness)。

该命题已被Karp证明^[1]。

2 问题描述

本文讨论的图是指无向图, 通常用 $V(G)$ 和 $E(G)$ 分别表示图 G 的顶点集和边集, 图 G 的一个正常 k -顶点着色, 简称为图的 k -点着色, 是指用 k 种颜色 $1, 2, \dots, k$ 对 G 的各顶点都分配(或称着)不同的颜色, 其中图 G 的顶点集个数 v 与着色种数 k 存在关系 $v \geq k$ 。换句话说讲, 简单图 G 的一个正常 k -点着色, 就是把 $V(G)$ 划分成 k 个独立集的一个分类 $\{V_1, V_2, \dots, V_k\}$, 其中 $V_i (i=1, \dots, k)$ 是 G 的独立集。我们用 $C(k)$ 表示

k 种颜色, 即 $C(k) = \{1, 2, \dots, k\}$ 。现在我们可以更确切地给出图 G 的 k -点着色的定义: 图 G 的 k -点着色, 是从 $V(G)$ 到 $C(k)$ 的一个映射 σ , 当且仅当 $u, v \in V(G)$ 且 $(u, v) \in E(G)$ 时, $\sigma(u) \neq \sigma(v)$, 全体 G 的 k -点着色 σ 构成的集合通常记作 $Cok(G)$, 简记为 $C_k(G)$ 。若 $C_k(G) \neq \emptyset$ (空集), 即 G 至少有一个正常 k -点着色, 就称 G 是 k -点可着色的。

当 $K=3$ 时, 就是我们要讨论的三色问题, 它是一个NP完全问题, 求给定的图 G 中顶点只用三种颜色 $C = \{C_0, C_1, C_2\}$ 着色是否有可行解? 约束条件是要求每两个相邻的顶点必须着不同颜色, 假设图 $G=(V, E), V = \{v_i | 1 \leq i \leq n\}, e_i$ 代表无向边 $(v_i, v_j) \in E$ 且 $v_i < v_j$ 。

命题2 三角形(环)为无向图顶点着三种颜色的极大独立集。

首先, 我们看当用三种颜色着色的极大独立集如图1所示, 当图表示为三角形时要满足相邻顶点着不同颜色, 必须是三角形中每个顶点各取一种颜色。

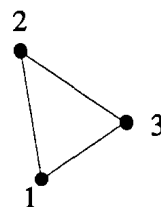


图1 极大独立集合

证明: $G=(V, E), V = \{1, 2, 3\}, E = \{(1, 2), (2, 3), (1, 3)\}, C = \{C_0, C_1, C_2\}, C_i (1 \leq i \leq n)$ 表示顶点着色集, 因为 G 的约束条件是相邻的顶点必须着不同颜色。对于 G 所构成的 $\Delta 123, e_{12} \in E \wedge e_{31} \in E \wedge e_{23} \in E$, 所以 V 中的顶点是两两相邻

的。因此图 G 中顶点 $1, 2, 3$ 的取色方案 $C_1 \cap C_2 \cap C_3 = \varphi$, 即得 $C_1 = \{C_i | 0 \leq i \leq 2 \wedge i \neq j \neq k\}$, $C_2 = \{C_j | 0 \leq j \leq 2 \wedge i \neq j \neq k\}$, $C_3 = \{C_k | 0 \leq k \leq 2 \wedge i \neq j \neq k\}$ 。

另外, 对于不能用三种颜色着色的图, 如图2所示, 为了更加直观我们将图2的平面图转化为图3所示的三维立体图, 显然从这个四面体中我们可以得到四个三角形: $\Delta 123, \Delta 124, \Delta 234, \Delta 134$, 容易得出该图无法用三种颜色着色。

证明: 反证法。假设图2为 $G = (\{1, 2, 3, 4\}, E)$, $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$, 图能够用三种颜色 C 着色。由前面三角形是图着色的极大独立集得, 图3中的三角形为 $\Delta 123, \Delta 124, \Delta 234, \Delta 134$, 在 $\Delta 123$ 和 $\Delta 124$ 包含共四个顶点, 可以看出边 $e_{12} \vee e_{13} \vee e_{23} \in \Delta 123, e_{12} \vee e_{14} \vee e_{24} \in \Delta 124, \Delta 123 \subset G, \Delta 124 \subset G$ 。所以 $\Delta 123 \cap \Delta 124 = \{e_{12}\}$, 在 $\Delta 123$ 中 $\exists C_1 = \{C_0\}, C_2 = \{C_1\}$, 则 $C_3 = \{C_2\}$, 由此推出在 $\Delta 124$ 中, C_1 和 C_2 已经在 $\Delta 123$ 中取值, 所以 $C_4 = \{C_2\}$, 又 $\because \exists e_{34} \in E, \therefore$ 顶点3和4是邻边, $C_3 \cap C_4 \neq \varphi$, 与题目约束条件矛盾。因此图 G 最小着色集合为四种颜色。

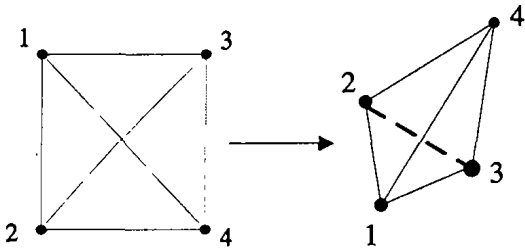


图2 三色无解最小集

图3四面体

因此, 我们得出定理: 如果无向图 G 中存在任意四顶点构成四面体, 则图 G 是非三色图。它是判断图的顶点着三种颜色有否可行解的充要条件, 下面的算法步骤中就是依此判断可行解的条件。

推论 如果无向图 G 的任意两个极大独立集顶点两两相邻, 则图 G 是非三色图。

我们还可以得出当求解图的顶点四色问题时, 四面体为极大独立子集, 实际上四面体由四个三角形所构成的, 因此三角形是四面体的子集。依次类推, 求解五色问题时, 我们也是根据五个四面体所构成的多面体空间图形, 如图4所示。对于 $k \geq 3$ 的图着色问题, 都可由三角形递推得来。

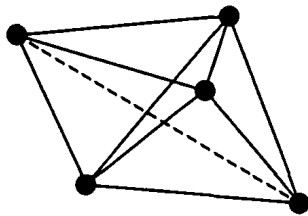


图4

主要思想: 采用图的深度优先遍历改进算法, 对于每个节点只走三步, 次序由小到大。

回答图的可着色(顶点三色)的问题步骤:

```

step1 以邻接表形式输入图 G
      初始化  $G = (V, E), T_{\Delta} = \varphi, E_{\Delta} = \varphi, \neg E_{\Delta} = E$ 。
step2 for i=1 to n do //深度优先遍历图(DFS)找出三角形集合
      { while(graph [vi].next <> nil)
        { j=graph [vi].next; //Vj 为 Vi 的第一邻接点
          while(graph [vj].next <> nil)
            if (graph [vj].next < vj) then graph [vj].next = graph [vj].next.next;
            else { k= graph [vj].next; //Vk 为 Vj 的邻接点(Vi 的第二
    
```

```

邻接点);
while(graph [vk].next <> nil)
  { if (graph [vk].next == Vi) then {
    将  $\Delta ijk$  及其对应的三条边分别加入到  $T_{\Delta}$  和  $E_{\Delta}$  中;
     $T_{\Delta} = \{\Delta ijk | i < j < k\}, E_{\Delta} = \{(i, j), (i, k), (j, k)\}$ 
    break;
    //找到以 Vi 为头长度为3的圈即  $\Delta ijk$ 
    else graph [vk].next = graph [vk].next.next;
  }
  graph [vi].next = graph [vi].next.next; } //end while
step3 判断  $T_{\Delta}$ , if  $\exists \Delta ijk \wedge \Delta ijm \wedge \Delta ikm \wedge \Delta jkm \wedge i < j < k < m$  then
  { output "图 G 是 3-Colorable";
    转 step4. }
  else {
    output "图 G 不是 3-Colorable";
    exit; } //end if
step4 //给出可满足条件的着色方案
      while ( $T_{\Delta} \neq \varphi$ ) do
        { 计算出  $\neg E_{\Delta} = E - E_{\Delta}, \text{Min} \Delta ijk \in T_{\Delta}, //i \leq j \leq k$ , 编序
          初始任取  $C_i = \{C_0\}, C_j = \{C_1\}, C_k = \{C_2\}$ 
        L1: if  $\exists (V_i, V_m) \in \neg E_{\Delta} \vee (V_j, V_m) \in \neg E_{\Delta} \vee (V_k, V_m) \in \neg E_{\Delta}$ 
          then  $C_m = C - C_i \vee C_m = C - C_j \vee C_m = C - C_k$ ; else {
             $T_{\Delta} = T_{\Delta} - \{\Delta ijk\}$ ;
             $E_{\Delta} = E_{\Delta} - \{(i, j), (i, k), (j, k)\}$ ;
          }
        if  $\exists (V_m, V_n) \in \neg E_{\Delta} \vee (V_n, V_m) \in \neg E_{\Delta}$  then  $C_n = C - C_m$ 
        else if  $\exists (V_m, V_n) \in E_{\Delta} \vee (V_n, V_p) \in E_{\Delta}$ 
          then
            { if  $C_m$  已取色  $\wedge C_p$  未取色 then
              {  $C_n = C - C_m$ ;
                 $E_{\Delta} = E_{\Delta} - \{(m, n)\}$ ;
                取出  $\Delta mnp$  goto L1; }
              if  $C_m$  已取色  $\wedge C_p$  已取色 then
                {  $C_n = (C - C_m) \cap (C - C_p)$ ; //  $\Delta mnp$ 
                   $C_n = C_n - (C_m \cap C_p)$ ;  $C_m = C_m - C_n, C_p = C_p - C_n$ ;
                }
            }
        } // end while
    
```

该算法的流程图如图5。

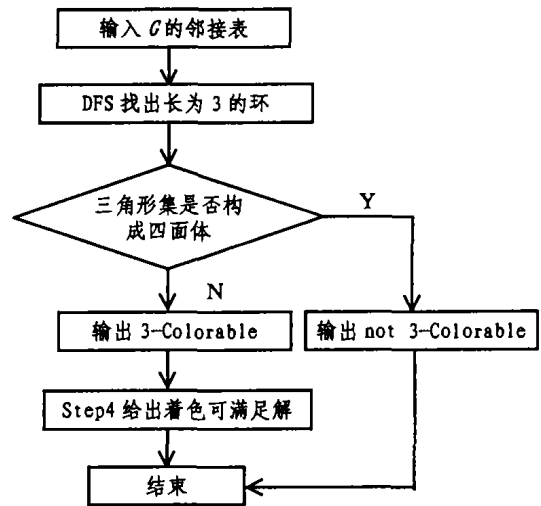


图5 着色流程图

算法特点: 该算法解决了两个问题, 一个是回答了图是否有三色着色方案的可行解问题也即图的 3-Colorable 问题, 另一个是给出了图三色着色可行解也就是可满足性问题 (Satisfiability)。

3 评价及结论

首先由算法可知该方法没有以前常用图的局部搜索算法所存在的回溯问题, 步骤中每个顶点只需判断有以该顶点出发长度为3的圈或者三角形, 因此此算法通过遍历邻接表判断三个相邻的顶点是否形成环或者三角形, 如果长度为3出发点和终止点相同就证明已形成三角形而没有回溯问题, 然后进行下一步的处理。而且以前的算法的回溯可能导致图中

(下转第93页)

3.3 实验结果

在数据进行模拟前先对数据预处理,数据归一化,归一化后的数据分为两部分,第一部分为训练数据,第二部分为测试数据,测试数据约占整个数据的10%左右。在模型建立中,我们使用的核函数为径向基函数(RBF),同时将标准的算法简称为SSVM,分解合作加权算法简称为DCWSVM,实验结果如图2所示,图中横坐标的1至5个点表示 (σ, c, ϵ) 参数的不同组合, σ 表示核函数RBF的参数, c 表示调整参数, ϵ 表示误差。图中实线表示DCWSVM算法结果,虚线表示SSVM算法结果,左图是MAE的比较,中图是NMSE的比较,右图是为计算时间的比较。

从图2数据可以看出,在预测误差保持大体一致的情况下,DCWSVM算法在运行速度方面要大大优于SSVM算法;这也说明本文中提出的DCWSVM算法是有效的。

(上接第78页)

所有顶点的颜色调整,由此造成NP难(NP-Hardness)问题,此算法不会引起该问题而在多项式时间内得出可行解或无解的结果。

例:如图6所示,求图G是否为三色图。

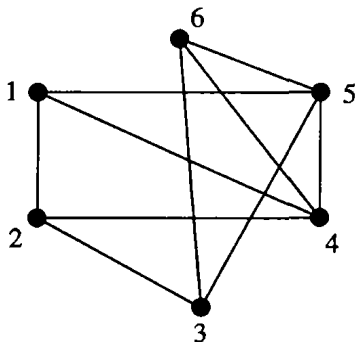


图6 无向图G

求解过程如下:

Step1 输入图G的邻接表如图7所示,并初始化集合 $G=(V, E), T_{\Delta}=\varphi, E_{\Delta}=\varphi, \neg E_{\Delta}=E$ 。

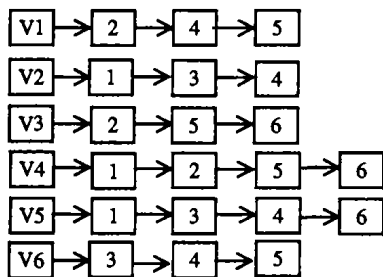


图7 图G的邻接表

Step2 首先从顶点V1开始遍历 $V1 \rightarrow V2$,然后再从V2开始遍历第二个结点V3(因为链表 $V2 > V1$,所以跳过V1到下一结点),最后遍历以V3开头的链表,查找是否有 $V3 \rightarrow V1$ 的边。此时形成三角形的长度为3。表中可以看出不存在V1到V3的边,因此返回结点V2的下一邻边V4即 $V2 \rightarrow V4$ 。再从V4出发,可以找到 $V4 \rightarrow V1$,也就形成了 $V1 \rightarrow V2 \rightarrow V4 \rightarrow V1$ 的三角环。这样 $T_{\Delta} = \{\Delta 124\}, E_{\Delta} = \{(1,2), (1,4), (2,4)\}, \neg E_{\Delta} = E - E_{\Delta}$ 。其次再继续从V1的下一个邻边 $V1 \rightarrow V4$,找到

结论 运用基于支持向量机的分解合作加权算法对股票指数时间序列数据作预测,实验和理论都证明在保证泛化能力的情况下,其运算速度比标准算法要快,同时也大大地减少了内存的开销。

参考文献

- 1 Vapnik V N. Statistical learning theory [M]. New York: Wiley, 1998
- 2 Vapnik V N. The Nature of Statistical Learning Theory [M]. New York: Springer, 1999
- 3 Cristianini N, Taylor J S. An introduction to support vector machines and other kernel-based learning methods [M]. New York: Cambridge Press, 2000
- 4 Tay F E H, Cao L J. Modified support vector machines in financial time series forecasting [J]. Neurocomputing, 2002, 48: 847~861
- 5 Tay F E H, Cao L J. Application of support vector machines in financial time series forecasting [J]. omega, 2001, 29: 309~317

$\Delta 145$,因此 $T_{\Delta} = \{\Delta 124, \Delta 145\}, E_{\Delta} = \{(1,2), (1,4), (2,4), (1,5), (4,5)\}, \neg E_{\Delta} = E - E_{\Delta}$ 。然后用同样的方法依次遍历以V2, V3, V4, V5, V6开头的链表。实际上只需到结点V4就可终止了,因为以V5和V6开头的结点构成的三角环长度 < 3 了,它们所包含的三角环已在前面结点找到了。这样最终结果集为: $T_{\Delta} = \{\Delta 124, \Delta 145, \Delta 356, \Delta 456\}, E_{\Delta} = \{(1,2), (1,4), (2,4), (1,5), (4,5), (3,5), (3,6), (5,6), (4,6)\}, \neg E_{\Delta} = E - E_{\Delta} = \{(2,3)\}$ 。

Step3 判断 T_{Δ} ,得出集合中的三角形不构成四面体,所以输出给定的图G是3-Colorable的。

Step4 给出可满足条件的着色方案(进行3-Coloring)。首先,从 T_{Δ} 中取出 $\Delta 124$,假设 $C_1 = \{C_0\}, C_2 = \{C_1\}, C_4 = \{C_2\}, \because (2,3) \in \neg E_{\Delta}, \therefore C_3 = C - \{C_1\} = \{C_0, C_2\}$ 。然后取出 $\Delta 145, \because C_1$ 和 C_4 已取色, $C_1 \cap C_4 \cap C_5 = \varphi, \therefore C_5 = \{C_1\}$ 。再取出 $\Delta 356$,又 $\because C_5$ 已取色, $(V_5, V_6) \in E_{\Delta}, C_5 \cap C_6 = \varphi, \therefore C_6 = C - C_5 = \{C_0, C_2\}$ 。最后取出 $\Delta 456, \because C_4$ 和 C_5 已取色, $C_4 \cap C_5 \cap C_6 = \varphi, \therefore C_6 = \{C_0\}$,又 $\because C_3 \cap C_6 = \varphi, \therefore C_3 = \{C_2\}$ 。所以得出其中符合条件的一种着色方案为: $C_1 = \{C_0\}, C_2 = \{C_1\}, C_3 = \{C_2\}, C_4 = \{C_2\}, C_5 = \{C_1\}, C_6 = \{C_0\}$ 。

参考文献

- 1 Karp R M. Reducibility among combinatorial problems. In: Raymond E. Miller and James W. Thatcher, eds. Complexity of Computer Computations, Plenum Press, 1972. 85~103
- 2 Khot S. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002
- 3 Guruswami V, Khanna S. On the hardness of 4-coloring a 3-colorable graph. IEEE, 2000
- 4 Bogdanov A, Obata K, Trevisan L. A lower bound for testing 3-colorability in bounded-degree graphs. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002
- 5 Guruswami V, Hastad J, Sudan M. Hardness of approximate hypergraph coloring. IEEE, 2000
- 6 Guruswami V, Hastad J, Sudan M. Hardness of approximate hypergraph coloring. IEEE, 2000
- 7 Beigel R, Eppstein D. 3-coloring in time $O(1.3446^n)$: a no-MIS algorithm. IEEE, 1995