ELSEVIER

Information Processing Letters 52 (1994) 87–90

Information Processing Letters

Maximizing the number of unused colors in the vertex coloring problem

Refael Hassin *, Shlomo Lahav (Haddad)

Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv 69978, Israel

Communicated by L. Boasson; received 31 May 1994

Keywords: Analysis of algorithms; Combinatorial problems; Computational complexity

1. Introduction

Given an undirected graph $G = (V, E)$, the *vertex coloring problem* is to partition V into a minimum number of subsets, *color classes*, such that no edge of E has both its ends in the same subset. In other words, the problem is to cover V with minimal number of independent sets. Each independent set is colored differently.

Another way to describe the vertex coloring problem of a graph G is as a clique cover problem of its complement graph $G^C = (V, (V \times V) \setminus E)$ since each independent set in G is a clique in G^C. Hence we will sometimes refer to clique cover rather than to vertex coloring.

The problem is known to be NP-complete, and worse than that, a polynomial approximation algorithm with a constant error ratio cannot exist unless P = NP [4]. Recently, Demange, Grisoni, and Paschos [1] attacked the problem from another direction: Clearly, if $|V|$ colors are available, a coloring of the vertices is always possible. However, for a given graph, usually not all of these colors are necessary and the problem can be restated as that of maximizing the number of *unused* colors.

Demange, Grisoni, and Paschos [1] presented a polynomial time algorithm that is guaranteed to save at least $\frac{1}{2}$ of the number of colors saved by an optimal solution. Here, we will present a simple low complexity algorithm with this property. Moreover, we will improve this factor to $\frac{2}{3}$. We also suggest a possible reduction in the algorithm's complexity based on the observation that the maximum weighted matching problem can be approximated within any given error bound in linear time.

2. A simple algorithm

A *vertex coloring* of $G = (V, E)$ is a partition of V into disjoint subsets, *color classes*,

$$C = \{C_i\}_{i=1}^{k}, \quad C_i \subseteq V, \ 1 \leqslant i \leqslant k,$$

such that each subset induces an independent set in G. The size of the coloring is $|C| = k$.

Define a 2-*set* as a pair of vertices that are non-adjacent in G.

* Corresponding author. Email: hassin@math.tau.ac.il.

SSDI 0020-0190(94)00113-D

Algorithm 1.
Input: A graph $G=(V,E)$.
Output: A vertex coloring.
$S := V$;
while (there are 2-sets in S)
{ **find** a 2-set $\{i,j\} \in S$;
assign i and j to a new color class;
$S := S \setminus \{i,j\}$;
}
assign each of the remaining vertices of S to a new color class.

Note that another way to describe the algorithm is by generating a maximal (with respect to inclusion) matching in the complement graph G^C, assigning the two ends of each edge in it to a color class, and finally, assigning each isolated vertex to a new color class. Thus, the clique cover constructed by the algorithm uses only cliques with at most two vertices.

Denote by χ and χ' the size of a coloring of minimum size and the size of the coloring produced by Algorithm 1, respectively. Let $|V| = n$.

Theorem 2.

$$n - \chi' \geqslant \tfrac{1}{2}(n-\chi).$$

Proof. When the algorithm reaches the state in which no 2-sets exist in S, the vertices of S induce a clique in G. Let k be the size of $|S|$ at this stage. Clearly, a coloring of G requires at least k colors. Thus

$$n - \chi' = \tfrac{1}{2}(n-k) \geqslant \tfrac{1}{2}(n-\chi). \qquad \square$$

The time complexity of Algorithm 1 is linear in the number of edges in G^C and hence $O(n^2)$.

3. Improved algorithm

In this section we will improve the algorithm so that

$$n - \chi' \geqslant \tfrac{2}{3}(n-\chi).$$

Define a 3-*set* as a triplet of vertices that are pairwise non-adjacent in G.

Algorithm 3.
Input: A graph $G=(V,E)$.
Output: A vertex coloring.

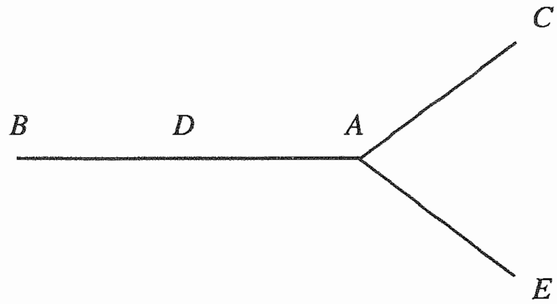

Fig. 1.

$S := V$;
while (there are 3-sets in S)
{ **find** a 3-set $\{i,j,k\} \in S$;
assign i, j and k to a new color class;
$S := S \setminus \{i,j,k\}$;
}
compute a maximum matching on the complement of the graph induced by S;
assign a new color class to each pair in the matching;
assign a new color class to each of the remaining vertices of S.

Note that Algorithm 3 produces a clique cover in G^C using cliques of at most three vertices. Denote by χ and χ' the size of a coloring of minimum size and the size of the coloring produced by Algorithm 3, respectively. Let $|V| = n$.

Theorem 4.

$$n - \chi' \geqslant \tfrac{2}{3}(n-\chi).$$

Proof. The proof is similar to the proof of Theorem 2. Let k be the size of S when no 3-sets can be found. Let c be the minimum number of colors needed to color these vertices. Then, $\chi \geqslant c$ or, $n-\chi \leqslant n-c$. However, Algorithm 3 colors these vertices with exactly c colors so that $\chi' = (n-k)/3 + c$ and

$$\begin{aligned} n - \chi' &= \tfrac{2}{3}(n-k) + k - c \\ &= \tfrac{2}{3}(n-c) + \tfrac{1}{3}(k-c) \geqslant \tfrac{2}{3}(n-\chi). \qquad \square \end{aligned}$$

The next example shows that the bound is tight.

Example 5. Consider the graph G described in Fig. 1. An optimal coloring is $\{C,D,E\}$, $\{A,B\}$. Thus $\chi = 2$. If the algorithm picks the 3-set $\{B,C,E\}$ then $\chi' = 3$ colors will be used. The bound is tight since

$5 - 3 = \frac{2}{3}(5 - 2)$.

The complexity of the algorithm is $O(f(G) + h(G))$, where $f(G)$ is the time required to form a maximal (with respect to inclusion) set of disjoint cycles of length 3 in the complement graph G^C, and $h(G)$ is the time required to compute a maximum cardinality matching in this graph. A naive algorithm can be used to guarantee for a graph with n vertices and m edges that $f(G) = O(nm)$. A maximum matching can be computed in $h(G) = O(\sqrt{n}m)$ time [3,5,6]. Therefore, we can guarantee a total complexity of $O(nm)$, where m is the number of edges in G^C.

Remark 6. The following is a simple improvement of Algorithm 3 that does not affect its complexity or worst case behavior but improves its practical performance. Instead of coloring the 3-set with a new color and ignoring it later, simply condense the three vertices into a single vertex. This vertex is connected to another vertex, say v, by an edge if at least one of its forming vertices was connected to v by an edge. The algorithm proceeds as before. This way larger classes of colors, containing more than 3 vertices, may be formed. A similar modification can also be applied to Algorithm 1.

Remark 7. Given an undirected graph $G = (V, E)$, the *edge coloring problem* is to partition E into a minimum number of subsets, *color classes*, such that no vertex of V is incident to more than a single edge from any class. In other words, the problem is to cover E with minimal number of matchings. Each matching set is colored differently.

An edge coloring is just a vertex coloring of the line graph of G were each edge of G is represented by a vertex and two vertices are connected by an edge if the corresponding edges in G are incident. Therefore, as observed in [1], any algorithm for the vertex coloring problem can be used to form an algorithm for edge coloring with the same factor of unused colors. The complexity of the edge coloring algorithm is that of the vertex coloring algorithm when m replaces n and $\sum_i (d_i)^2 = O(mn)$ replaces m (where d_i is the degree of vertex i).

4. Possible improvements

We now claim that $h(G)$ can be replaced by a lower complexity function. When Algorithm 3 reaches the stage where no more 3-sets exist, it computes a matching in the complement graph. This way the uncolored vertices at this stage are colored by the minimum possible number of colors. Let the size of this matching be M and denote by k the number of uncolored vertices. Then, the number of unused colors in an optimal solution is at most $(n - k) + M$.

Suppose that instead of a maximum matching of size M we approximate the maximum matching by a matching of size at least $\frac{2}{3}M$. Then, the number of unused colors in the approximate solution produced by Algorithm 3 is at least $(2/3)(n - k) + (2/3)M$ and the proof is still valid. To maintain the bound claimed by the theorem we only need to approximate the maximum matching by a factor $\alpha = \frac{2}{3}$.

We now show that a factor of $\frac{2}{3}$, or in fact any given constant factor $\alpha < 1$, can be obtained in $O(m)$ time, where m is the number of edges in G^C.

Consider a graph $G = (V, E)$. The algorithms of Even and Kariv [3] and Micali and Vazirani [5,6] for the maximum cardinality matching are based on Hopcroft and Karp's algorithm [2]. They construct in each phase a maximal set of disjoint minimum length augmenting paths. As shown by [2] only $\sqrt{|V|}$ phases are needed to find a maximum matching. Each phase is implemented in $O(|E|)$ time.

The size of the shortest augmenting path strictly increases from a phase to the next. If the length of the shortest augmenting path at a given phase is d, then after its termination, no augmenting paths of size d or less exist. In particular, after l phases, no augmenting path containing l edges of the matching (i.e., of size $2l + 1$) or less exists.

Theorem 8. *Let M' be the matching at a given step of the algorithm, when the shortest augmenting path is of length $2l+1$ (i.e., it contains exactly l edges from the current matching). Let M be a maximum matching. Then, $|M'| \geqslant |M|(1 - 1/l)$.*
$l+1$

Proof. Consider $M' \cup M$. It consists of pairwise disjoint alternating cycles and paths. Each cycle contains an equal number of edges from M' and M. This is also the case for paths of even length. Paths of odd length are augmenting paths with respect to M'. Consider such a path. Let r denote its length. Then, $r \geqslant 2l+1$, and the path contains $(r-1)/2$ edges from M' and $(r+1)/2$ edges from M. Thus the ratio of M' edges to M edges in this path is

$$\frac{r-1}{r+1} \geqslant \frac{l}{l+1}$$

and the claim follows from this observation. □

Corollary 9. *In* $O(l|E|)$ *time we can compute an approximation* M' *satisfying* $|M'| \geqslant |M|(1-1/l)$.

Acknowledgement

We thank Saharon Rosset for simplifying our original proof of Theorem 2.

References

[1] M. Demange, P. Grisoni and V.Th. Paschos, Approximation results for the minimum graph coloring problem, *Inform. Process. Lett.* **50** (1994) 19–23.

[2] J.E. Hopcroft and R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* **4** (1973) 225–231.

[3] S. Even and O. Kariv, An $O(n^{2.5})$ algorithm for maximum matching in general graphs, in: *Proc. 16th IEEE Symp. on Foundations of Computer Science* (1975) 100–112.

[4] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, in: *Proc. 25th ACM Symp. on Theory of Computing* (1993) 286–293.

[5] S. Micali and V.V. Vazirani, An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs, in: *Proc. 21st IEEE Symp. on Foundations of Computer Science* (1980) 17–27.

[6] V.V. Vazirani, A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{|V|}|E|)$ general graph matching algorithm, in: R. Kannan and W.R. Pulleybkank, eds., *Proc. Conf. on Integer Programming and Combinatorial Optimization*, University of Waterloo (1990) 509–535.