An algorithm on strong edge coloring of K_4-minor free graphs

M.F. van Bommel and Ping Wang
Department of Mathematics and Statistics
St. Francis Xavier University
Antigonish, Nova Scotia, Canada

Abstract. The strong chromatic index $\chi'_s(G)$ of a graph G is the smallest integer k such that G has a proper edge k-coloring with the condition that any two edges at distance at most 2 receive distinct colors. It is known that $\chi'_s(G) \leq 3\Delta - 2$ for any K_4-minor free graph G with $\Delta \geq 3$. We give a polynomial algorithm in order $O(|E(G)|(n\Delta^2 + 2n + 14\Delta))$ to strong color the edges of a K_4-minor free graph with $3\Delta - 2$ colors where $\Delta \geq 3$.

Keywords. K_4-minor free graph; induced match, edge coloring; strong chromatic index

Note to Gary from Martin. Thanks Gary for the conversations, CCC'15 in Kona, and for organizing many hockey pools - winning isn't everything!

Note to Gary from Ping. I came to know Gary in the early 90s in Regina when he was a new to the department of computer science and I was a graduate student in Mathematics. I enjoyed our many discussions about graph theory in the mathematics department lounge, but then Gary moved back to Victoria and I moved to Antigonish. Despite the distance, Gary remains a good friend and a mentor. The paper that we coauthored during a conference in Winnipeg attracted a lot of attention to the fire-fighting problem. Most importantly, I only won the hockey pool once so far by using linear programming while we were working on the linear programming part of our paper in the basement of Annex in Antigonish. Thanks Gary.

1 Introduction

Only simple graphs are considered in this paper. Let G be a graph with vertex set $V(G)$, edge set $E(G)$, minimum degree $\delta(G)$, and maximum degree $\Delta(G)$. A vertex v is called a *k-vertex* (*k^+-vertex*, *k^--vertex*, respectively) if the degree $d_G(v)$ of v is k (at least k, at most k, respectively). Let $N_G(v)$ and $E_G(v)$ denote the set of vertices adjacent to v and the set of edges incident to v, respectively. It is easy to see that $d_G(v) = |N_G(v)| = |E_G(v)|$ for any vertex v of a simple graph G. If no ambiguity arises in the context, $\delta(G)$, $\Delta(G)$, $d_G(v)$, $N_G(v)$, and $E_G(v)$ are written as $\delta, \Delta, d(v), N(v)$, and $E(v)$, respectively.

A *proper edge k-coloring* of a graph G is a mapping $\phi : E(G) \to \{1, 2, \ldots, k\}$ such that $\phi(e) \neq \phi(e')$ for any two adjacent edges e and e'. The *chromatic index* $\chi'(G)$ of G is the smallest integer k such that G has a proper edge k-coloring. The coloring ϕ is called *strong* if any two edges at distance at most two get distinct colors. Equivalently, each color class is an induced matching. The *strong chromatic index* $\chi'_s(G)$ of G is the smallest integer k such that G has a strong edge k-coloring.

Strong edge coloring of graphs was introduced by Fouquet and Jolivet [11] in 1983. It holds trivially that $\chi'_s(G) \geq \chi'(G) \geq \Delta$ for any graph G. In 1985, during a seminar in Prague, Erdős and Nešetřil put forward the following conjecture.

Conjecture 1. *For every graph G with maximum degree Δ,*

$$\chi'_s(G) \leq \begin{cases} \frac{5}{4}\Delta^2, & \text{if } \Delta \text{ is even;} \\ \frac{1}{4}(5\Delta^2 - 2\Delta + 1), & \text{if } \Delta \text{ is odd.} \end{cases}$$

Erdős and Nešetřil provided a construction showing that Conjecture 1 is tight if it were true. In 1997, using probabilistic methods, Molloy and Reed [18] showed that $\chi'_s(G) \leq 1.998\Delta^2$ for a graph G with sufficiently large Δ. The currently best known upper bound for a general graph G is $1.93\Delta^2$, due to Bruhn and Joos [3].

A graph G is called *d-degenerate* if each subgraph of G contains a vertex of degree at most d. Chang and Narayanan [5] proved that if G is a 2-degenerate graph, then $\chi'_s(G) \leq 10\Delta - 10$. Recently, T. Wang [23] strengthened this result to $\chi'_s(G) \leq 6\Delta - 7$ for any 2-degenerate graph G. As a special case, Wang [23] also showed that if all the 3^+-vertices in a graph G induce a forest, then $\chi'_s(G) \leq 4\Delta - 3$. Recall that a *chord* in a graph is an edge that joins two nonconsecutive vertices of a cycle. A graph is said to be *chordless* if there is no cycle in the graph that has a chord. Dębski et al. [7] proved that if G is a chordless graph, then $\chi'(G) \leq 4\Delta - 3$. More recently, Basavarajul and Francis [2] further improved this result to $\chi'(G) \leq 3\Delta$ for any chordless graph G.

Suppose that G is a planar graph. Using the Four-Color Theorem [1] and Vizing Theorem [25], Faudree et al. [10] gave an elegant proof to the result that $\chi'_s(G) \leq 4\Delta + 4$. They also constructed a class of planar graphs G with $\Delta \geq 2$ and $\chi'_s(G) = 4\Delta - 4$. They use their idea of turning an edge coloring problem into a vertex coloring problem to find some polynomial algorithms for some special family of graphs.

In this paper, we focus on the strong edge coloring of K_4-minor free graphs. A graph G has a graph H as a *minor* if H can be obtained from a subgraph of G by contracting edges, and G is called *H-minor free* if G does not have H as a minor. A planar graph is called *outerplanar* if it has an embedding in the Euclidean plane such that all the vertices are located on the boundary of the unbounded face. It is shown by Chartrand and Harary [6] that a graph G is an outerplanar graph if and only if G is K_4-minor free and $K_{2,3}$-minor free. Thus, the class of K_4-minor free graphs is a class of planar graphs that contains the class of outerplanar graphs.

Very recently, Hocquard et al. [14] proved that if G is an outerplanar graph with $\Delta \geq 3$ then $\chi'_s(G) \leq 3\Delta - 3$ and the upper bound $3\Delta - 3$ is tight. On the other hand, it is easy to see that a K_4-minor free graph G is 2-degenerate by the result of Duffin [8] and hence $\chi'_s(G) \leq 6\Delta - 7$ by the result of T. Wang [23]. Y.Q. Wang et al. [24] prove the following.

Theorem 1. *$\chi'_s(G) \leq 3\Delta - 2$ for any K_4-minor free graph G with $\Delta \geq 3$*

Computationally, there are also some very interesting results. For a given graph $G = (V, E)$, an induced matching is a set of $M \subseteq E$ such that there is no edge in E connecting two edges of M. A strong edge coloring of G is an assignment of colors to the edges of G such that each color class is an induced matching. Thus, finding a partition of the edges of G into induced matchings is equivalent to finding a strong edge coloring. It is well known that there are efficient algorithms for finding the maximum matching, such as the Jack Edmonds' "blossom algorithm" [9]. But finding a maximum induced matching is NP-complete, even for bipartite graphs with maximum degree four [22], bipartite with fixed girth g [17] and for 3-regular graphs [16]. It is, however, solvable in polynomial time for trees [12], chordal graphs [4], weakly chordal graphs [21] and circular arc graphs [13]. Some of these algorithms are developed based on the equivalency that an induced matching of G corresponds to an independent set in $L(G)^2$. We can look at the strong edge coloring problem from an equivalency standpoint. For any given graph G, form a new graph $L(G)$ by replacing each edge of G with a vertex, where two vertices in $L(G)$ are adjacent if and only the two associated edges are adjacent in G. So finding a partition of the edge set into induced matchings in G is equivalent to finding the vertex coloring in $L(G)^2$. We note that this idea is used by Kloks et al. [15] for finding $\chi'_s(G)$ where G is a chordal graph.

N. Robertson and P.D. Seymour [19] first introduced the concept of tree width. M. R. Salavatipour [20] proved the following powerful theorem by using the approach of tree-decomposition.

Theorem 2. *For every fixed integer k, there is a deterministic algorithm that, given a graph G with tree width k on n vertices and an integer s, determines in time $O(n(s+1)^{2^{4(k+1)+1}})$ whether G has a strong edge coloring using at most s colors or not, and if so, finds such a strong edge coloring.*

We shall study the family of K_4-minor free graphs in this paper. We use the terms K_4-minor free graphs and series-parallel graphs interchangeably in this paper. We study the series-parallel graphs because they are an interesting class of graphs; that is, they are simpler than planar graphs and have some well described structural properties and, at the same time, they are rich enough so that many problems are non-trivial even when restricted to this class. In other words, series-parallel graphs can be a testing ground for various algorithms.

Note that any K_4-minor free graph has tree width two. If $s = 3\Delta - 2$, then the complexity of this algorithm becomes $O(n(3\Delta - 1)^{2^{13}})$.

We develop a polynomial algorithm to strong edge color any K_4-minor free graph. Our algorithm is based on Theorem 1 and its proof, and provides an improvement over the method based on Theorem 2 for the family of K_4-minor free graphs.

2 A structural property of K_4 minor free graph and its corresponding coloring

Let G be a graph. By a series-parallel reduction we mean any of the following operations:

(i) deletion of a loop,

(ii) deletion of a vertex of degree at most one,

(iii) deletion of a parallel edge, and

(iv) suppression of a vertex of degree two,

where suppression of a vertex v in a graph G means to delete it and add an edge from u to w if u, w are distinct nonadjacent vertices which formed the neighborhood of v.

It follows that a graph is a series-parallel graph if and only it can be reduced to the null graph by repeatedly applying series-parallel reductions. In turn, this yields a linear-time algorithm to test whether a graph is series-parallel. Thus we can determine if a graph G is a K_4-minor free graph in linear time.

Now we are ready to explore some key features of K_4-minor graphs. First, it was proved by Duffin [8] that $\delta(G) \leq 2$ if G is a K_4-minor free graph. The following definition and more detailed structural theorem was presented by Y.Q. Wang et al. [24].

For a vertex $u \in V(G)$, let $n_i(u)$ and $n_{i^+}(u)$ denote the number of i-vertices and i^+-vertices that are adjacent to u in G, respectively. For a vertex $v \in V(G)$, we define:

$$D_G(v) = \{y \mid d(y) \geq 3 \text{ s.t. } vy \in E(G) \text{ or path } vxy \text{ exists with } d(x) = 2\}.$$

Theorem 3. *Let G be a K_4-minor free graph with $\Delta \geq 3$. Then G contains one of the following configurations* (B1), (B2) *and* (B3)*:*

(B1) *a vertex v with $n_1(v) \geq 1$ and $n_{2^+}(v) \leq 2$;*

(B2) *two adjacent 2-vertices;*

(B3) *a vertex v with $d(v) \geq 3$ and $|D_G(v)| \leq 2$.*

Here we outline an algorithm that will locate configurations **(B1), (B2), and (B3)** in a given graph G. Then some elements, vertices or edges of this graph will be removed, and an edge may be added to G. Overall, the size of resultant graph, the total number of vertices and edges, will be reduced. When operating on a graph, the adjacency between vertices of G is either stored in the form of adjacency matrix or in the form of a link list. In this paper, we assume it is stored in the form of linked list. We first sort the degrees into an increasing degree sequence, $d(v_1) \leq d(v_2) \leq \cdots \leq d(v_n)$. This can be done with $n\log(n)$ comparisons where $n = |V(G)|$. The reason we sort it is that we repeatedly have to find a vertex of degree one. In this case, if one exists, it would be the first vertex in the degree sequence. When an edge or a vertex is deleted or an edge added, the action of updating the linked list and the degree sequence can be done in at most Δ time. The other operation frequently employed by the algorithm is to find a particular edge $uv \in E(G)$. Clearly, this can also be done in at most Δ time.

Graph Reduction Procedure

Reduction (I): Locating **B1** in G and reducing G

We search the neighbours of v_1 if $d(v_1) = 1$ to determine if $n_{2^+}(v_1) \leq 2$. Let $H = G - v_1$. This can be done in at most Δ time.

Reduction (II): Locating **B2** in G and reducing G

Locating **B2** in G. We can find a 2-vertex and check its two neighbours. This can be done in $2n$ time.

Let x and y denote the two adjacent 2-vertices. Let x_1 denote the neighbour of x other than y, and y_1 the neighbour of y other than x. If $x_1 = y_1$, then let $H = G - xy$, otherwise let $H = G - y + xy_1$.

Reduction (III): Locating **B3** and reducing G

(B3): a vertex v with $d(v) \geq 3$ and $|D_G(v)| \leq 2$.
First we have to find $D_G(v)$ for each vertex $v \in V(G)$.
This implies we have to check the neighbours of v and the neighbours of vertices in $N(v)$. This can be done in at most $n\Delta(\Delta - 1)$ time.

(IIIa): $|D_G(v)| = 1$ and $D_G(v) = \{x\}$

If v has a neighbor z which is a leaf, let $H = G - z$.

Else let u be a 2-vertex in $N(v)$, $uv, ux \in E(G)$, and let $H = G - uv$.

To check whether a vertex has a leaf adjacent to it,

one has to go through the linked list of v.

This can be done at most Δ time, so at most 2Δ time in total.

(IIIb): $|D_G(v)| = 2$ and $D_G(v) = \{x, y\}$.

Let $N_G(v) \bigcap N_G(x) = \{x_1, \cdots, x_m\}$, $N_G(v) \bigcap N_G(y) = \{y_1, \cdots, y_n\}$.

We may assume that $m \geq n$.

If v has a leaf, z, then $H = G - z$.

By symmetry, we can be sure both $|D_G(x)| = |D_G(y)| = 2$.

(IIIbi): If $vx \in E(G)$, then let $H = G - vx_1$.

(IIIbii): Else we may assume $vx \notin E(G)$.

If $vy \notin E(G)$, then let $H = G - x_1 + vx$.

If $n \geq 1$, then let $H = G - vy$.

If one of $d(v), d(x), d(y)$ is less or equal to $\Delta - 1$,

Then let $H = G - x_1 + vx$.

It follows that $m = \Delta - 1$ and there exists $x' \in N_G(x) \setminus N_G(v)$.

If $d(x') \leq \Delta - 1$, let $H = G - xx_1$.

If $x'y \in E(G)$, let $H = G - \{x_1 \cdots, x_{\Delta-1}$.

If $x'y \notin E(G)$, let $H = G - \{v, x_1, \cdots, x_{\Delta-1}, x\} + x'y$.

Among the nine if statements in (III), six of them involves checking whether a particular edge is in G or there is a leaf adjacent to it. This can be done in at most $6 * 2\Delta + 3\Delta = 13\Delta$ time. Hence, (III) can be done in at most $n\Delta(\Delta - 1) + max\{2\Delta, 13\Delta\} \leq n\Delta^2 + 13\Delta$ time.

Clearly, H would remain to be a K_4-minor if an edge or a vertex is deleted from G. It is also easy to check H is still K_4-minor free even when an edge is added to H because the existence of an uv-path in G before uv is added to H. Since at least one edge will be removed from G in each iteration, so this reduction algorithm will yield the following complexity.

$$W(n) \leq \Delta + 2n + n\Delta^2 + 13\Delta + W(|E(G)| - 1)$$

and

$$W(n) \leq |E(G)|\,(n\Delta^2 + 2n + 14\Delta)$$

3 Coloring Algorithm

Let G be a K_4-minor free graph with $\Delta \geq 3$. Theorem [24] shows that $\chi'_s \leq 3\Delta - 2$ and the upper bound of χ'_s is tight. Let $C = \{1, 2, \ldots, 3\Delta - 2\}$ denote a set of $3\Delta - 2$ colors where $\Delta \geq 3$. Since $\Delta \geq 3$, it follows that $|C| = 3\Delta - 2 \geq 7$ and thus we can strong color G with $|G| \leq 7$.

Observation 4. *Let P_n be a path with $n \geq 2$ vertices. Then $\chi'_s(P_n) = 1$ if $n = 2$, $\chi'_s(P_n) = 2$ if $n = 3$, and $\chi'_s(P_n) = 3$ if $n \geq 4$.*

Observation 5. *Let C_n be a cycle with $n \geq 3$ vertices. Then $\chi'_s(C_n) = 5$ if $n = 5$, $\chi'_s(C_n) = 3$ if $n \equiv 0 \pmod 3$, and $\chi'_s(C_n) = 4$ otherwise.*

Using Observations 4 and 5, we have the following result on the strong chromatic index of a graph with maximum degree at most two.

Observation 6. *Let G be a graph with $\Delta \leq 2$. Then $\chi'_s(G) \leq 5$, and the equality holds if and only if G contains a component that is a 5-cycle.*

Note that it is trivial to strong edge color of G with $3\Delta - 2$ colors if $\Delta \leq 2$ or $|V(G)| \leq 7$. Thus we shall use these two as **two stop conditions** for the algorithm below.

Let ϕ is a strong edge coloring of H where ϕ: $E(H) \to C = \{1, 2, \ldots, 3\Delta - 2\}$. Then we will extend this strong edge coloring ϕ from H to G according the graph reduction operation preformed on G described in the previous section. For an edge $e \in E(G) \setminus E(H)$, we use $F(e)$ to denote the set of colors forbidden on e when e is considered to be colored. That is, $F(e) = \{\phi(e') \in C \mid e' \in E(H)$ has distance at most two to e in $G\}$. Moreover, let $f(e) = |F(e)|$. When $e \in E(G)$ is considered to be colored, it is proved in [24] that there is always a color in $C \setminus F(e)$ available to color e by showing $f(e) \leq \Delta - 3$. Since the set different $C \setminus F(e)$ is used here, we have to consider the complexity of this operation. For any color in $F(e)$ it will take no more than $3\Delta - 2$ comparisons to delete it from C. Hence, this operation can be done in at most $(3\Delta - 2)^2$ time.

Algorithm Color(G)
Input: A K_4-minor free graph G.
Output: A strong edge $(3\triangle - 2)$-coloring of G.

if $\triangle(G) \leq 2$ **then** ▷ Observation 6
    Construct a strong edge 5-coloring as per Observation 6
else if G contains a 1-vertex u with $uv \in G$ ▷ I
        and the number of 2^+-vertices in $N(v)$ is at most 2 **then**
    Remove u and uv from G
    Color(G)
    Add u and uv back to G
    Color uv with a color in $C \setminus F(vw)$ where $w \neq u$
else if G contains two adjacent 2-vertices x and y **then**
    if x and y have a common neighbour z **then** ▷ II
        Remove xy from G
        Color(G)
        Add xy back to G
        Color xy with a color in $C \setminus F(xy)$
    else ▷ II
        Where edges $ux, yv \in G$ where $u \neq v$, $u \neq y$, $x \neq v$:
        Contract edge xy in G producing vertex z
        Color(G)
        Split vertex z to x and y recreating edges $ux, xy, yv \in G$
        Color xy with a color in $C \setminus F(xy)$
        Color ux with color of uz
        Color yv with color of uv
    end if

else if G contains a vertex v with $d(v) \geq 3$ and $|D_G(v)| \leq 2$ **then**
  if $|D_G(v)| = 1$, say $D_G(v) = \{x\}$ **then** ▷ IIIa
    if v is adjacent to a leaf z **then**
      Remove z and vz from G
      Color(G)
      Add z and vz back to G
      Color vz with a color in $C \setminus F(vz)$
    else
      Where u is a 2-vertex that is adjacent to both v and x:
      Remove uv from G
      Color(G)
      Add uv back to G
      Color uv with a color in $C \setminus F(uv)$
    end if
  else if $|D_G(v)| = 2$, say $D_G(v) = \{x, y\}$ **then** ▷ IIIb
    if v is adjacent to a leaf z **then**
      Remove z and vz from G
      Color(G)
      Add z and vz back to G
      Color vz with a color in $C \setminus F(vz)$
    else
      Where $x_1, x_2, \ldots x_m \in N(v)$ are degree 2 and adjacent to x:
      Where $y_1, y_2, \ldots y_n \in N(v)$ are degree 2 and adjacent to y:
      if $vx \in E(G)$ **then** ▷ IIIbi
        Remove vx_1 from G
        Color(G)
        Add vx_1 back to G
        Color vx_1 with a color in $C \setminus F(vx_1)$
      else ▷ IIIbii
        Add vx to G
        Remove x_1, vx_1, and xx_1 from G
        Color(G)
        Add x_1, vx_1, and xx_1 back to G
        Color xx_1 with color of vx
        Remove vx from G
        if $vy \notin E(G)$ **or** $n \geq 1$ **or** $xy \in E(G)$
        or $d(v) < \triangle$ **or** $d(x) < \triangle$ **or** $d(y) < \triangle$ **then**
          Color vx_1 with a color in $C \setminus F(vx_1)$
        else
          Where $x' \neq x_1, \ldots, x_m$ is adjacent to x:
          if $x'y \in E(G)$ **then**
            Color $xx_1, xx_2, \ldots, xx_{m-1}$ with $C(y) \setminus \{\phi(x'y), \phi(vy)\}$
            Color $vx_1, vx_2, \ldots, vx_{m-1}$ with $C(x') \setminus \{\phi(x'y), \phi(xx')\}$

Color xx_m, vx_m with colors not on $E(y) \cup E(x')$
else
Color both xx' and vy with $\phi(x'y)$
Color $xx_1, xx_2, \ldots, xx_m$ with $C(y) \setminus \{\phi(x'y)\}$
Color $vx_1, vx_2, \ldots, vx_{m-1}$ with $C(x') \setminus \{\phi(x'y), \phi(xx_1)\}$
end if
end if
end if
end if
end if
end if

This algorithm can color any K_4-minor graph with at most $3\Delta - 2$ colors. Note that $\chi'_s \leq 3\Delta - 2$ is a tight upper bound for χ'_s. The complexity is $O(\,|E(G|\,(9n\Delta^4)\,)$. This is substantially better than the previous known result $O(n * (3\Delta - 1)^{2^{13}})$ in [20] if Δ is a function of n. As one can see that graph reduction, that is, locating configurations **(B1), (B2), and (B3)** in G is the most costly part. Furthermore, with minor modifications, this algorithm can be used to find a partition of the edges of G into induced matchings.

References

[1] K. Appel, W. Haken, Every planar graph map is four colorable, Bull. Amer. Math. Soc. 82 (1976) 711-712.

[2] M. Basavarajul, M. C. Francis, Strong chromatic index of chordless graphs, J. Graph Theory 80 (2015) 58-68.

[3] H. Bruhn, F. Joos, A stronger bound for the strong chromatic index, Electron. Notes Discrete Math 49 (2015) 277-284.

[4] K. Cameron, Induced matchings, Discrete Applied Math, 24 (1989) 97-102.

[5] G. J. Chang, N. Narayanan, Strong chromatic index of 2-degenerate graphs, J. Graph Theory 73 (2013) 119-126.

[6] G. Chartrand, F. Harary, Planar permutation graphs, Ann. Inst. H. Poincaŕe Sect. B (N. S.) 3 (1967) 433-438.

[7] M. Dębski, J. Grytczuk, M. Śleszyńska-Nowak, The strong chromatic index of sparse graphs, Inform. Process. Lett. 115 (2015) 326-330.

[8] J. Duffin, Topology of series-parallel networks, J. Math. Anal. Appl. 10 (1965) 303-318.

[9] J. Edmonds, Paths, trees, and flowers, Cdn. J. of Math. 17 (1965) 449-467.

[10] R. J. Faudree, A. Gyárfás, R. H. Schelp, Zs. Tuza, The strong chromatic index of graphs, Ars Combin. 29(B) (1990) 205-211.

[11] J. L. Fouquet, J. L. Jolivet, Strong edge-colorings of graphs and applications to multi-k-gons, Ars Combin. 16(A) (1983) 141-150.

[12] G. Fricke, R. Lasker, Strong matchings on trees, Numer. 89 (1992) 239-243.

[13] M.C. Golumbic, R. Lasker, Irredundancy in circular arc graphs, Disc. Appl. Math, 44 (1993), 79-89.

[14] H. Hocquard, P. Ochem, P. Valicov, Strong edge-coloring and induced matchings, Inform. Process. Lett. 113 (2013) 836-843.

[15] T. Kloks, S.-H. Poon, C.-T. Ung, Y.-L. Wang, On the strong chromatic index and maximum induced matching of tree-cographs, permutation graphs and chordal bipartite graphs, Discrete Alg. 30 (2015) 21-28.

[16] C.W. Ko, F.B. Shepherd, Adding an identity to total unimodular matrix, London School of Economics Operations Research working paper, LSEOR 92.14, July 1994.

[17] M. Mahdian, On computational compexity of strong edge coloring, Disc. App. Math, 118 (2002) 239-248.

[18] M. Molloy, B. Reed, A bound on the strong chromatic index of a graph, J. Combin. Theory Series B 69 (1997) 103-109.

[19] N. Robertson and P.D. Seymour, Graph Minors III. Planar Tree-Width, J. Combin. Theory, Series B. 36 (1984), 49-64.

[20] M. R. Salavatipour, A polynomial time algorithm for strong coloring of partial k-trees, Disc. App. Math, 143 (2004) 285-291.

[21] R. Sritharan, Y. Tang, Finding a maximum induced matching in weakly chordal graphs, Horizons Combinatorics/16th Shanks Lecture series, May 21-24, 2001.

[22] L.J. Stockmeyer, V.V. Vazirani, NP-completeness of some generalizations of the maximum matching problems, Info. Proc. Letters 15 1 (1982), 14-19.

[23] T. Wang, Strong chromatic index of k-degenerate graphs, Disc. Math 330 (2014) 17-19.

[24] Y. Q. Wang, P. Wang, W. F. Wang, Strong chromatic index of K_4-minor free graphs, Info. Proc. Letters 129 (2018), 53-56.

[25] V. G. Vizing, On an estimate of the chromatic index of a p-graph, Diskret. Anal. 3 (1964) 25-30.