# On Sum Coloring of Graphs

Mohammad R. Salavatipour

Department of Computer Science, University of Toronto
10 King's College Road, Toronto, Ontario M5S 3G4, Canada
mreza@cs.toronto.edu

**Abstract**

The sum coloring problem asks to find a vertex coloring of a given graph $G$, using natural numbers, such that the total sum of the colors is minimized. A coloring which achieves this total sum is called an optimum coloring and the minimum number of colors needed in any optimum coloring of a graph is called the strength of the graph. We prove the NP-hardness of finding the vertex strength for graphs with $\Delta = 6$. Polynomial time algorithms are presented for the sum coloring of chain bipartite graphs and $k$-split graphs. The edge sum coloring problem and the edge strength of a graph are defined similarly. We prove that the edge sum coloring and the edge strength problems are both NP-complete for $k$-regular graphs, $k \geq 3$. Also we give a polynomial time algorithm to solve the edge sum coloring problem on trees.

## 1 Introduction

We consider the following kind of graph coloring, called *sum coloring*: for a given graph $G$, find a proper vertex coloring of $G$, using natural numbers, such that the total sum of the colors of vertices is minimized amongst all proper colorings of $G$. This minimum total sum of colors is called the *chromatic sum* of $G$, and is denoted by $\Sigma(G)$. We refer to a vertex coloring whose total sum is $\Sigma(G)$ as an *optimum vertex coloring*. Assume that $P$ is an optimum vertex coloring of a graph $G$ with $k$ colors. It is clear that $k \geq \chi(G)$, but not necessarily $k = \chi(G)$, even for trees (See figure 1). The minimum number of colors needed in any optimum coloring of a graph $G$ is called the *strength* of $G$, and is denoted by $s(G)$. Similar to the vertex sum coloring problem we can define the *edge sum coloring*, to be an edge coloring using natural numbers, such that the total sum of the colors of the edges of the graph is minimized. The *edge chromatic sum*, denoted by $\Sigma'(G)$, and the *edge strength*, denoted by $s'(G)$, are defined in a similar way. Note that there are some graphs for which the edge strength and the chromatic index are not equal, i.e $s'(G) > \chi'(G)$. See [6] for such an example.

The notion of a coloring in which we want to minimize the total sum of colors first appeared in 1987 from two different sources. In theoretical graph theory, Kubika [12] in her Ph.D thesis introduced the chromatic sum of a graph with the above notation. Supowit [18] introduced the *optimum cost chromatic partition* (OCCP) problem, from its application in VLSI design, in which we have to find a proper coloring of a graph, using a given set of colors $\{c_1, c_2, \ldots, c_k\}$, such that the total sum of colors is minimized. There are some other applications for this problem in scheduling and resource allocation, see [1].

In [15], Kubika and Schwenk proved the NP-completeness of the chromatic sum problem and gave a polynomial time algorithm to find the chromatic sum of trees. They showed that for any integer $k$,

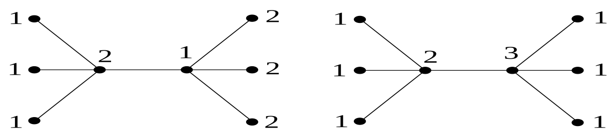

Figure 1: A coloring of a tree with total sum 12 using 2 colors, and a coloring with total sum 11 using 3 colors

there exists a tree $T$, whose strength is $k$, and $k \in \Omega(\log n)$ where $n$ is the size of $T$. In [3] Erdös et al. continued the study of graphs that require many colors in their optimum vertex colorings. In another article, Erdös et al. [19] gave some interesting tight bounds on the chromatic sum of a graph in terms of the number of vertices and edges of the graph. Hajiabolhassan et al. [6] considered optimum vertex colorings of graphs with $s(G)$ colors. A nice result they proved is that $s(G) \leq \Delta$, where $\Delta$ is the maximum degree of $G$, if $G$ is neither a complete graph nor an odd cycle. In [7] Halldórsson and Kortsarz showed that for any $k$-colorable graph $G$, $s(G) \in O(k \log n)$, where $n$ is the number of vertices.

Kroon et al. [13] proved the NP-hardness of the OCCP problem on interval graphs and gave a linear time algorithm for trees. Then, Jansen [9] showed that the OCCP problem for cographs (graphs with no induced $P_4$) and for partial $k$-trees can be solved in time $O(|V|+|E|)$ and $O(|V|\log^{k+1}|V|)$, respectively. On the one hand he showed the NP-completeness of this problem on bipartite graphs and permutation graphs, and on the other hand he proved that the OCCP problem can be solved for cobipartite graphs in polynomial time. We have proved in [17] that the greedy algorithm which finds maximal independent sets consecutively can solve the OCCP problem for $P_4$-reducible graphs (graphs in which each vertex is in at most one $P_4$), which strictly contain cographs. Therefore this result extends the result of Jansen, for cographs.

The only results about the edge sum coloring problem, as far as we know, appeared in [6] and [1]. The edge sum coloring problem is introduced independently in each of these articles as the vertex sum coloring of line graphs. Hajiabolhassan et al. [6] introduced the notion of edge strength, and similar to Vizing's theorem, they proved that for every graph $G$, $s'(G) \leq \Delta + 1$.

The organization of the paper is as follows: In the next section we show that finding the strength of a graph with $\Delta = 6$ is NP-hard. This is the first complexity result on the strength of graphs. In section 3 some algorithms for various restricted classes of graphs are presented. We show that there is a linear time algorithm for finding an optimum vertex coloring of chain bipartite graphs. The author proves in [17] that the sum coloring problem is NP-complete for split graphs. Here we show that if we bound the outgoing degree of each vertex in one of the parts of a split graph, the clique part or the independent set part, then there is a polynomial time algorithm for them. The time complexity of the edge sum coloring problem and the edge strength problem are both proved to be NP-complete in section 4. Finally in the last section we provide a polynomial time algorithm to solve the edge sum coloring problem for trees. We note that the author has some results on the generalization of sum coloring to list sum coloring in [17].

## 2 Complexity of finding the vertex strength

Since the sum coloring problem seems no easier than the standard vertex coloring problem, one can expect that finding the vertex strength is NP-hard. We prove that in fact it is NP-hard to find the vertex strength even for a graph with $\Delta = 6$. Our proof is very similar to the proof of Kubika and Schwenk [15] for proving the NP-completeness of finding the chromatic sum.

**Theorem 2.1** *The vertex strength problem restricted to the class of graphs with maximum degree 6 is NP-hard.*

**Proof:** First note that we do not know whether this problem belongs to NP or not. To show the NP-hardness, we are going to reduce the vertex 3-coloring of graphs with maximum degree 4 problem, which is NP-complete even for the class of planar graphs [5], to this problem. Consider an instance of the vertex coloring problem. Let $G(V, E)$ be the graph of this instance. Construct $G'(V', E')$, the cartesian product of $G \times K_3$, as follows: take three copies of $G$, called $G_1$, $G_2$, and $G_3$. For a vertex $v \in V$, let $v_1$, $v_2$, and $v_3$ be the corresponding vertices of $G_1$, $G_2$, and $G_3$, respectively, and connect them together using three new edges. Since the degree of each vertex in $G$ is at most 4, and each $v_i \in G'$ is connected to two more vertices, therefore the maximum degree of $G'$ is at most 6. It is clear that we can construct $G'$ in polynomial time. Now we claim that:

$$\chi(G) \leq 3 \Longleftrightarrow s(G') = 3.$$

First suppose that $s(G') = 3$. It means that there is an optimum vertex coloring of $G'$ in which only 3 colors are used. Since $G$ is a subgraph of $G'$, this coloring induces a proper 3-coloring for $G$. Therefore $\chi(G) \leq 3$.

Now assume that $\chi(G) \leq 3$. Therefore we can color $G_1$ with 3 colors, independently. To obtain a proper 3-coloring of $G'$, we use the same partition of vertices of $G_1$ for $G_2$ and $G_3$, with the modification that the color of the $j$'th class of $G_i$ is $(i + j - 2 \textit{ mod } 3) + 1$, instead of $j$. Thus each $G_i$ $(1 \leq i \leq 3)$ is colored with colors $1, 2, 3$, and also this is a proper coloring of $G'$. It is not difficult to see that this is an optimum vertex coloring for $G'$. This follows since each complete subgraph of size 3 of $G'$, which contains the corresponding vertices of copies of $G$, requires at least 3 colors, in any proper coloring of $G'$. In this coloring each of these complete subgraphs are colored with colors $1, 2, 3$, which clearly gives the least possible sum of colors. Thus this is an optimum vertex coloring of $G'$, and clearly uses the least possible number of colors. Therefore $s(G') = 3$. ■

**Corollary 2.2** *For a given graph $G$, it is NP-hard to determine if $s(G) \leq k$, for any fixed $k \geq 3$.*

We don't know the time complexity of deciding if the strength of a given graph is equal to 2, but we expect it to be NP-hard.

# 3 Algorithms for some restricted classes of graphs

Unlike the standard vertex coloring, the sum coloring problem is NP-complete for bipartite graphs, permutation graphs, and split graphs [9, 2, 17]. So it is natural to consider this problem for restricted families of graphs. In this section we show that we have polynomial time algorithms to solve this problem on chain bipartite graphs, and $k$-split graphs.

## 3.1 Vertex sum coloring of chain bipartite graphs

The notion of chain graphs was introduced by Yannakakis [20, 21].

**Definition 3.1** *A bipartite graph $G(X \cup Y, E)$ is called a chain graph if for every two vertices $x_i, x_j \in X$, we have either $N(x_i) \subseteq N(x_j)$ or $N(x_j) \subseteq N(x_i)$.*

In other words, there is an ordering of the vertices of $X$, $x_{\pi_1}, x_{\pi_2} \ldots, x_{\pi_n}$, where $|X| = n$, such that $N(x_{\pi_i}) \subseteq N(x_{\pi_{i+1}})$, $1 \leq i < n$. We call this property, the *chain* property. It is not difficult to see that if we have an ordering of the vertices of $X$, having the chain property, we can find an ordering of the vertices of $Y$ with this property, as well. We have to note that recognition of chain bipartite graphs and finding an ordering of each part having the chain property can be easily done in linear time.

Let $G(X \cup Y, E)$ be a chain bipartite graph. Without loss of generality, we assume that the graph is connected. Also, let $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$ ($|X| = n$ and $|Y| = m$) be the orderings of $X$ and $Y$ respectively, having the chain property. By $G_{ij}$ we mean the induced subgraph of $G$ on vertices $\{x_1, x_2, \ldots, x_i\} \cup \{y_1, y_2, \ldots, y_j\}$. For a pair $(i, j)$ where $G_{ij}$ has no edges, consider the following vertex sum coloring of $G$: assign color 1 to all of the vertices of $G_{ij}$. If $n - i \geq m - j$ then assign color 2 to $x_{i+1}, \ldots, x_n$ and assign color 3 to $y_{j+1}, \ldots, y_m$. Otherwise, if $n - i < m - j$ then assign color 3 to $x_{i+1}, \ldots, x_n$ and assign color 2 to $y_{j+1}, \ldots, y_m$. Let $S_{ij}$ be the total sum of this coloring. We call a pair $(i, j)$ a *proper pair* if the set of vertices of $G_{ij}$ is a maximal independent set of $G$. Let $S_{min}$ be the minimum of $S_{ij}$ over all proper pairs $(i, j)$. The algorithm computes $S_{min}$ and returns the minimum of $\{S_{min}, 2n + m, n + 2m\}$ as the chromatic sum of $G$. The last two values are the total cost of the trivial 2-colorings of $G$. We refer to figure 1 to see why a 2-coloring does not necessarily give an optimum coloring of a chain bipartite graph.

**Theorem 3.2** *The chromatic sum of $G$ is equal to the minimum of $\{S_{min}, 2n + m, n + 2m\}$.*

**Proof:** Let $C$ be an optimum vertex coloring of $G$. We denote the color of vertex $v$ by $c(v)$ and its neighborhood by $N(v)$. If no vertex in $X$ has color 1, then all of the vertices in $Y$ must have color 1 and so all the vertices in $X$ must have color 2. Thus the total cost of $C$ will be $2n + m$. Similarly if no vertex in $Y$ has color 1, then the total cost of $C$ will be $n + 2m$.

Now assume that there is at least one vertex with color 1 in each part. Let $i$ be the largest index such that $c(x_i) = 1$, and let $j$ be the largest index such that $c(y_j) = 1$. We know that $C$ is an optimum vertex coloring, and $N(x_k) \subseteq N(x_i)$, for $k < i$. Therefore, all the vertices $x_1, x_2, \ldots, x_{i-1}$ must have color 1, too. Similarly all the vertices $y_1, y_2, \ldots, y_{j-1}$ must have color 1. Thus the vertex set of $G_{ij}$ is an independent set. It follows from the definition of $i$ and $j$ that the color of the vertices $x_{i+1}, \ldots, x_n$ and $y_{j+1}, \ldots, y_m$ must be greater than 1. So each of them must be connected to a vertex having color 1, otherwise we could simply change its color to 1. In particular, $x_{i+1}$ is connected to $y_a$ for some $a \leq j$, and $y_{j+1}$ is connected to $x_b$ for some $b \leq i$. Because of the chain property it follows that $x_i$ is connected to all of the vertices $y_{j+1}, \ldots, y_m$, and $y_j$ is connected to all of the vertices $x_{i+1}, \ldots, x_n$, and the subgraph $G - G_{ij}$ is a complete bipartite subgraph. Therefore $G_{ij}$ is a maximal independent set. Thus $(i, j)$ is a proper pair. Note that if $k > i$ and $l > j$, then no two vertices $x_k$ and $y_l$, can have the same color. Now it is clear that the vertices of the larger of the two sets $\{x_{i+1}, \ldots, x_n\}$ and $\{y_{j+1}, \ldots, y_m\}$ must be colored with color 2, and the vertices of the smaller one with color 3. This kind of coloring is the same as the one we use in the algorithm when we select a proper pair. ■

To find a proper pair $(i, l)$ for a fixed $i$, first we find the largest number $j$ such that $x_i y_j \notin E$. If such a number does not exists then it's clear that there is no proper pair having $i$ as the first element. We now show that $(i, j)$ is a proper pair and that it is the only proper pair having $i$ as the first element. This follows from the fact that, if $x_a y_b \in E$, $1 \leq a \leq i$ and $1 \leq b \leq j$, then because of the chain property, $x_i y_b \in E$. Since $b \leq j$, this implies that $x_i y_j \in E$, which is a contradiction. Therefore, to find a proper pair with $(i, j)$ for a fixed $i$, it takes at most $O(deg(x_i))$ time. So overall, the time complexity of finding all proper pairs is $O(|E|)$. We have proved the following theorem:

**Theorem 3.3** *The chromatic sum of chain bipartite graphs can be found in time $O(|E| + |V|)$.*

### 3.2 Vertex sum coloring of $k$-split graphs

A graph is a split graph if its vertices can be partitioned into a clique and an independent set. The author has proved in [17] that the sum coloring problem is NP-complete for split graphs, and therefore for chordal graphs, by giving a reduction from the *Exact Cover by 3-sets*. We introduce two subclasses of split graphs where the sum coloring problem can be solved for them efficiently. The following two lemmas will be used later in our algorithms:

**Lemma 3.4** *If $G(C \cup I, E)$ is a split graph, where $C$ is the complete part and $I$ is the independent part and $|C| = n_C$, then $s(G) \leq n_C + 1$.*

**Proof:** Consider an optimum vertex coloring of $G$. Let $i$ be the smallest positive number such that none of the vertices of $C$ have color $i$. In this case, no vertex in $I$ can have a color greater than $i$, otherwise we can simply change the color of that vertex to $i$. So the only colors (possibly) used in $I$ are $1, 2, \ldots, i$. It follows that there can't be any color greater than $n_C + 1$ in $C$, otherwise there is some color $x$ $(i + 1 \leq x \leq n_C + 1)$ that is not used in $C$, and we can simply change the color greater than $n_C + 1$ to $x$. ■

**Lemma 3.5** *For split graph $G(C \cup I, E)$, where $|C| = n_C$ and $|I| = n_I$, we have:*
*(i)* $\Sigma(G) \leq \frac{n_C(n_C+1)}{2} + n_C + n_I$.
*(ii) If* $s(G) = n_C + 1$ *then* $\Sigma(G) = \frac{n_C(n_C+1)}{2} + n_C + n_I$.

**Proof:** **(i)** Consider the vertex sum coloring of $G$ in which all of the vertices of $I$ have color 1, and the vertices of $C$ are colored by colors $2, 3, \ldots, n_C + 1$. It is trivial that this is a proper coloring and the total sum of colors is $\frac{n_C(n_C+1)}{2} + n_C + n_I$.

**(ii)** If $s(G) = n_C + 1$ then there are $n_C + 1$ vertices such that the set of colors of these vertices is exactly $\{1, 2, \ldots, n_C + 1\}$, and the total sum of the colors of the other vertices is at least $n_I - 1$. Therefore:

$$\Sigma(G) \geq \frac{(n_C + 1)(n_C + 2)}{2} + n_I - 1 = \frac{n_C(n_C + 1)}{2} + n_C + n_I.$$

The result follows from part (i). ■

**Definition 3.6** *A split graph $G(C \cup I, E)$, where $C$ is a complete subgraph and $I$ an independent set, is a $k_I$-split graph if the degree of each vertex of $I$ is at most $k$. It is a $k_C$-split graph if the number edges of each vertex in $C$ going out to the vertices in $I$ is at most $k$. We call a graph $k$-split if it is either $k_C$-split or $k_I$-split graph.*

Let $G(C \cup I, E)$ be a $k_I$-split graph. By lemma 3.4 and the fact that $C$ is a clique of size $n_C$, we have: $n_C \leq s(G) \leq n_C + 1$. Lemma 3.5 gives the exact value of $\Sigma(G)$ and its proof gives an optimum vertex coloring for the case $s(G) = n_C + 1$.

Now suppose that $s(G) = n_C$. We denote the set of neighbors of a vertex $c \in C$ that are in set $I$ by $N_I(c)$. Let $P$ be an optimum coloring of $G$ with $n_C$ colors. Clearly $C$ is colored with colors $1, 2, \ldots, n_C$. Let $v_{c_i}$ be the vertex of $C$ that has color $i$ in $P$. The set of vertices in $I$ that have color 1 are exactly those vertices that are not connected to $v_{c_1}$, i.e the vertices in $N_I(v_{c_1})$ are colored by at least 2. Among them, those that are not connected to $v_{c_2}$ have color 2, and the remainder has color at least 3. In general, the number of vertices having color greater than $i$ in $I$, is:

$$|N_I(v_{c_1}) \cap N_I(v_{c_2}) \ldots \cap N_I(v_{c_i})|.$$

Note that since the degree of each vertex in $I$ is bounded by $k$, no vertex in $I$ has a color greater than $k+1$ in any optimum vertex coloring of $G$. Let:

$$S = |N_I(v_{c_1})| + |N_I(v_{c_1}) \cap N_I(v_{c_2})| + \ldots + |\bigcap_{i=1}^{k} N_I(v_{c_i})|.$$

Therefore, the total sum of colors in $I$ is $n_I + S$ and the total sum of colors of coloring $P$ is:

$$\frac{n_C(n_C+1)}{2} + n_I + S.$$

So, to find an optimum vertex coloring of $G$, using $n_C$ colors, we must minimize the term $S$. We can simply consider all permutations $\pi$ with $k$ elements, such that each element is a vertex of $C$, and compute the value of $S$ for each permutation by assigning color $i$ to the vertex $v_{c_{\pi(i)}}$ of $C$, and then taking the minimum over all values of $S$. It is straightforward to compute the value of $S$ for each permutation in time $O(n_I)$ and the number of such permutations is $O(n_C^k)$. The chromatic sum of $G$ is the minimum between the minimum value of $S$ and $n_C$, plus the term $\frac{n_C(n_C+1)}{2} + n_I$.. Hence:

**Theorem 3.7** *Let $G(C \cup I, E)$ be a $k_I$-split graph, for fixed $k$. Then the chromatic sum and also an optimum vertex coloring of $G$ can be computed in $O(n_I \times n_C^k)$.*

Now, let $G(C \cup I, E)$ be a $k_C$-split graph. Again, if $s(G) = n_C + 1$ then we know the exact value of $\Sigma(G)$. Assume that $s(G) = n_C$ and consider an optimum vertex coloring of $G$, called $P$. Let $v_{c_1}$ be the vertex in $C$ that has color 1 in $P$. Clearly every vertex in $I - N_I(v_{c_1})$ also has color 1. We show that no vertex in $N_I(v_{c_1})$ can have a color greater than $k+1$. Otherwise, let $v_x$ be a vertex in $N_I(v_{c_1})$ with color $x$, such that $k+1 < x \leq n_C$. Therefore there exists a color $y$ such that $y \leq k+1$ and $y$ has not appeared on any vertex in $N_I(v_{c_1})$. Let $v_{c_x}$ and $v_{c_y}$ be the vertices of $C$ having colors $x$ and $y$, respectively. By exchanging the colors of $v_{c_x}$ and $v_{c_y}$ we can assign color $y$ to all the vertices of $N_I(v_{c_1})$ colored $x$. This exchange reduces the total sum of the colors of $P$, which is a contradiction. So, to find an optimum coloring of $G$, using $n_C$ colors, we select one of the vertices of $C$, call $v_{c_1}$, and assign color 1 to it and to all the vertices in $I - N_I(v_{c_1})$. Then we consider all the possible assignments of colors $2, 3, \ldots, k+1$ to the vertices of $N_I(v_{c_1})$. Note that since the degree of $v_{c_1}$ is at most $k$, there is a constant number of such assignments. Also, for each assignment of colors to the vertices in $N_I(v_{c_1})$, we have to find a coloring for the uncolored vertices of $C$ that is feasible with coloring of $N_I(v_{c_1})$. To do so we construct a bipartite graph $G'(X \cup Y, E')$, such that $X = \{x_1, x_2, \ldots, x_{n_C - 1}\}$, $Y = \{y_2, y_3, \ldots, y_{k+1}\}$, and $x_i y_j \in E'$ if and only if there is no edge between the $i$th uncolored vertex of $C$ and the vertex in $N_I(v_{c_1})$ with color $j$. It is not difficult to see that by using a bipartite matching in $G'$ that covers all the vertices in $Y$ (if there exists such a matching), we can find those vertices of $C$ that will have colors $2, 3, \ldots, k+1$, and then color the rest of vertices of $C$ with colors $k+2, \ldots, n_C$ arbitrarily. By taking the minimum between the total sum of the colors of each of these colorings, we find the sum of the optimum coloring using $n_C$ colors. Finally, we have to take the minimum between this amount and the total sum of the coloring using $n_C + 1$ colors. Finding a maximum matching in $G'$ can be done in time $O(n_C)$, by applying the augmenting path algorithm $k$ times. Therefore, for each assignment of colors to the vertices in $N_I(v_{c_1})$ we spend $O(n_C)$ time to complete the coloring. Also, at the first stage, there are $n_C$ choices for selecting the vertex $v_{c_1}$. Thus, overall we spend $O(n_C^2)$ time to find the minimum sum of colors between all colorings using $n_C$ colors.

**Theorem 3.8** *If $G(C \cup I, E)$ is a $k_C$-split graph, for fixed $k$, then the chromatic sum and also an optimum vertex coloring of $G$ can be computed in $O(n_C^2)$.*

**Corollary 3.9** *The chromatic sum of $k$-split graphs can be computed in time $O(n_I \times n_C^k)$.*

## 4 Complexity of the edge chromatic sum and the edge strength problems

In [1] Bar-noy et al. proved that the edge sum coloring problem is NP-hard for general multigraphs, but the complexity of this problem was left open for simple graphs. In this section, we study edge sum coloring of $k$-regular graphs for $k \geq 3$. We prove that finding the edge chromatic sum and the edge strength of a $k$-regular graph are both NP-complete. Holyer [8] proved that the chromatic index problem restricted to cubic graphs is NP-complete. Later, Leven and Galil [16] generalized this result to $k$-regular graphs, $k \geq 3$. By Vizing's theorem and Hajiabolhassan et al. [6] we know that the chromatic index and the edge strength of a $k$-regular graph are either $k$ or $k+1$. First we show that finding the edge chromatic sum of a $k$-regular graph is NP-complete.

**Instance:** A $k$-regular graph $G$ of size $n$, $k \geq 3$.

**Question:** Is $\Sigma'(G) = \frac{nk(k+1)}{4}$?

Note that since the degree of each vertex of a $k$-regular graph is $k$, in any edge coloring of $G$ the total sum of the colors used to color the incident edges of any vertex is at least $\frac{k(k+1)}{2}$. Therefore, $\Sigma'(G) \geq \frac{nk(k+1)}{4}$.

**Theorem 4.1** *The edge chromatic sum problem is NP-complete for $k$-regular graphs, $k \geq 3$.*

**Proof:** First of all, it is trivial that this problem belongs to NP. To prove the NP-completeness we use a reduction from the chromatic index problem. We prove that for $k$-regular graph $G$:

$$\Sigma'(G) = \frac{nk(k+1)}{4} \Longleftrightarrow \chi'(G) = k.$$

First, assume that $\chi'(G) = k$. This means that there exists a $k$-edge coloring of $G$, called $C$. Since the degree of each vertex in $G$ is $k$, all numbers $1, 2, \ldots, k$ must appear on the edges incident with each vertex. So the total sum of the colors in $C$ is equal to $\frac{nk(k+1)}{4}$. Therefore $\Sigma'(G) = \frac{nk(k+1)}{4}$.

Now, suppose that $\Sigma'(G) = \frac{nk(k+1)}{4}$. It suffices to prove that any optimum edge coloring of $G$ is a $k$-edge coloring of $G$. Assume, by way of contradiction, that $C$ is an optimum edge coloring of $G$ with $k+1$ colors. So there exists at least one vertex such that color $k+1$ is the color of one of its incident edges. Therefore the total sum of the colors of the edges incident with that vertex is more than $\frac{k(k+1)}{2}$. We have the lower bound $\frac{k(k+1)}{2}$ for the sum of the colors of the edges of every other vertex. Therefore the total sum of colors of the edges in $G$ will be strictly greater than $\frac{nk(k+1)}{4}$, which is a contradiction. Therefore, any optimum edge coloring of $G$ is also a $k$-edge coloring of $G$, and so $\chi'(G) = k$.

■

**Corollary 4.2** *For a $k$-regular graph $G$, $k \geq 3$, we have: $s'(G) = k \Longleftrightarrow \chi'(G) = k$.*

**Proof:** If $s'(G) = k$ then trivially $\chi'(G) = k$. Now assume that $\chi'(G) = k$. From the arguments we had in the proof of theorem 4.1 it follows that if $s'(G) > k$ then $\Sigma'(G) > \frac{nk(k+1)}{4}$. Also we know that the sum of any $k$-edge coloring of $G$ is $\frac{nk(k+1)}{4}$. This proves that in this case $s'(G) = k$. ■

Since finding the chromatic index is NP-complete for the class of $k$-regular graphs, $k \geq 3$, therefore:

**Theorem 4.3** *Finding the edge strength of $k$-regular graphs is NP-complete, $k \geq 3$.*

## 5 Edge sum coloring of trees

In this section, we give a polynomial time algorithm that finds the edge chromatic sum of trees. We can find an optimum edge coloring as well, by storing some extra information in the data tables.

Assume that we are given tree $T$ of size $n$, with a Breath First Search ordering of it. For vertex $v$ of $T$, we denote the subtree rooted at $v$ by $T_v$. By *lower edges* of $v$, we mean the set of edges that connect $v$ to its children. We denote the degree of vertex $v$ by $deg(v)$ and assume that the maximum degree is $\Delta$. If $C$ is an edge coloring of $T$, the set of colors used for the lower edges of $v$, is denoted by $L_v$. Also, as we mentioned in the previous section, we know that $s'(T) \leq \Delta + 1$.

The algorithm uses the dynamic programming method. We have a $n \times (\Delta + 1)$ table, called $S$, such that:

$$S[v, j] = \text{The cost of an optimum edge coloring of } T_v \text{ such that } j \notin L_v.$$

For $1 \leq j \leq \Delta + 1$, the initial values of $S$ are filled as:

$$\begin{cases} S[x, j] = 0 & x \text{ is a leaf} \\ S[x, j] = \infty & \text{otherwise.} \end{cases}$$

The algorithm computes the values of this table in a bottom-up way, from the leaves of the tree up to the root. It computes the value $S[v, j]$ for each internal node $v$, after it has computed the values for the children of $v$. Suppose that $u_1, u_2, \ldots, u_k$ are the children of internal node $v$. Assume that $S[u_i, j]$ is computed, for $1 \leq i \leq k$ and $1 \leq j \leq \Delta + 1$, and we want to compute the value of $S[v, m]$ $(1 \leq m \leq \Delta + 1)$, the cost of an optimum edge coloring for $T_v$, such that $m \notin L_v$. Construct the complete weighted bipartite graph $G_v = (A \cup B, E')$, where $A = \{a_1, a_2, \ldots, a_k\}$, $B = \{b_j | j \neq m, 1 \leq j \leq \Delta + 1\}$, and the weight of edge $a_i b_j$ is $w(a_i b_j) = S[u_i, j] + j$. Now find a min-weighted maximum matching in $G_v$, and call it $M$. This matching covers all the vertices of $A$. We assign the colors of the lower edges of $v$, according to the following rule:

*The color of $vu_i$ is $c_i$, iff edge $a_i b_{c_i}$ is in $M$.*

It is easy to see that by this assignment the value of $S[v, m]$ is equal to the sum of the weights of $M$, which is:

$$\sum_{e \in M} w(e) = \sum_{a_i b_{c_i} \in M} S[u_i, c_i] + c_i.$$

Knowing how to compute the value of $S[v, m]$, from the computed values of children of $v$, the algorithm starts from the leaves of $T$, and fills in the table, from bottom to up, until it computes the value of $S[r, \Delta + 1]$, where $r$ is the root of $T$. One can easily verify that the minimum value of $S[r, j]$, for $1 \leq j \leq \Delta + 1$, is the edge chromatic sum of the tree.

To find an optimum edge coloring for $T$, we only need to keep track of the colors of the lower edges of each vertex $v$, when we compute $S[v, m]$. This can be easily done by storing this extra information for each entry of the table $S$.

The most time consuming step of the algorithm is to find the min-weighted matching. If the vertex $v$ has $k$ children, then the size of the bipartite graph $G_v$, is of $O(\Delta)$. The fastest min-weighted maximum matching algorithm works in time $O(|E||V| \log |V|)$ and is due to Galil et al. [4]. By using this algorithm, for each vertex $v$, and each value $m \leq \Delta + 1$, we spend $O(deg(v)\Delta^2 \log \Delta)$ time. Thus the total amount of time for computing the entries of the row $v$ of the table $S$ is $O(deg(v)\Delta^3 \log \Delta)$. Summing up these values for all $v$, we have the bound $O(n^4 \log n)$ for the time complexity of this algorithm. Therefore, we can say:

**Theorem 5.1** *An optimum edge sum coloring of a tree, and therefore its edge chromatic sum, can be found in time* $O(n^4 \log n)$.

**More general version:** We note that the above algorithm, with a few modifications, can also be used to solve the more general case of the edge sum coloring problem, similar to the OCCP problem for vertex sum coloring, where the color costs are from a given set $C = \{c_1, c_2, \ldots, c_l\}$.

**Acknowledgments.** I would like to thank my supervisor, Derek Corneil, and Mike Molloy for their helpful advice and suggestions, and the Department of Computer Science at University of Toronto, for financial assistance. I extend my appreciation to the referees for their useful comments which resulted in an improved paper.

## References

[1] A. Bar-Noy, M. Bellare, M. M. Halldorsson, H. Shachnai, T. Tamir, On chromatic sums and distributed resource allocation, Information and computing 140 (1998) 183-202.

[2] A. Bar-Noy, G. Kortsarz, The minimum color sum of bipartite graphs, Journal of Algorithms 28 (1998) 339-365.

[3] P. Erdös, E. Kubika, A. Schwenk, Graphs that require many colors to achieve their chromatic sum, Congressus Numerantium 71 (1990) 17-28.

[4] Z. Galil, S. Micali, H. Gabow, An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs, SIAM J. Comp. 15 (1986) 120-130.

[5] M.R. Garey and D.S. Johnson , Computers and Intractability, (Freeman, San Francisco, 1979).

[6] H. Hajiabolhassan, M.L. Mehrabadi, R. Tusserkani, Minimal coloring and strength of graphs, Discrete Math. 215 (2000) 265-270.

[7] M. M. Halldórsson and G. Kortsarz Multicoloring Planar Graphs and Partial $k$-Trees, Proceedings of the Second International Workshop on Approximation algorithms (APPROX '99). LNCS Vol. 1671, Springer-Verlag (1999).

[8] I. Holyer, The NP-completeness of edge coloring, SIAM J. Comp. 10 (1981) 718-720.

[9] K. Jansen, Complexity results for the optimum cost chromatic partition problem, Preprint.

[10] K. Jansen, Approximation results for the optimum cost chromatic partition problem, Journal of Algorithms 34 (2000) 54-89.

[11] T. Jiang and D. West, Coloring of trees with minimum sum of colors, Journal of Graph Theory 32 (1999) 354-358.

[12] E. Kubika, The chromatic sum of a graph, Ph.D Thesis, Western Michigan University (1989).

[13] L.G. Kroon, A. Sen, H. Deng, A. Roy, The optimum cost chromatic partition problem for trees and interval graphs, Workshop on graph theoretical concepts in computer science LNCS 1197 (1996).

[14] K. Kubika, G. Kubika, and D. Kountanis, Approximation algorithms for the chromatic sum, Proceedings of the First Great Lakes Computer Science Conference, LNCS 507 (1989) 15-21.

[15] E. Kubika, A.J. Schwenk, An introduction to chromatic sums, Proceedings of the seventeenth Annual ACM Comp. Sci., Conf. ACM Press (1989) 39-45.

[16] D. Leven, Z. Galil, NP-completeness of finding the chromatic index of regular graphs, J. Algorithms 4, 35-44 (1983).

[17] M. Salavatipour, On sum coloring of graphs, M.Sc. Thesis, Department of Computer Science, University of Toronto (2000).

[18] K.J. Supowit, Finding a maximum planar subset of nets in a channel, IEEE Trans. on Computer Aided Design CAD 6, 1 (1987) 93-94.

[19] C. Thomassen, P. Erdös, Y. Alavi, P.J. Malde, A.J. Schwenk, Tight bounds on the chromatic sum of a connected graph, Journal of Graph Theory 13 (1989) 353-357.

[20] M. Yannakakis, Computing the minimum fill-in is NP-complete, SIAM J. Alg. Disc. Meth. 2 (1981) 77-79.

[21] M. Yannakakis, Node-deletion problems on bipartite graphs, SIAM J. Comp. 10 (1981) 310-327.