THE 1-2-3-EDGE LABELLING AND VERTEX COLOURS

Mehdi Alaeiyan and Mohammad Reza Farahani
Department of Mathematics, Iran University of Science and Technology (IUST), Narmak, Tehran 16844, Iran

Abstract

A labelling of the edges of a graph is called vertex-colouring if the labelled degrees of the vertices yield a proper colouring of the graph. In this paper, we show that such a labelling is possible from the label with numbers of the set \( \{1, 2, 3\} \) for all graphs not containing components with exactly 2 vertices.

Keywords: edge-labelling, vertex-colouring, ordering algorithm, labelling algorithm, complete graph, latest edge, layer.

1. Introduction

All graphs in this note are simple and the numbers of vertices are finite. For notation is not defined here we refer the reader to [4]. For some \( k \in \mathbb{N} \), let \( f : E(G) \to \{1, 2, \ldots, k\} \) be an integer labelling of the
edges of a graph $G = (V(G), E(G))$, which have $n$ vertices. This labelling is called edge labelling if the labelled degrees $S_i := \sum_{j=1, j \neq i}^{n} f(v_iv_j)$ for all integer $1 \leq i \leq n$ of the vertices yield a proper vertex-colouring of the graph; (i.e., $S_i$'s the sum of the edge labels of vertex $v_i$, where $1 \leq i \leq n$ that it is a vertex colour of $v_i$). It is easy to see that for every graph which does not have a component isomorphic to $K_2$, there exists such a labelling for some integer positive $k$. In 2002, Karonski et al. (see [6, 10]) conjectured that such a labelling with $k = 3$ is possible for all such graphs ($k = 2$ is not sufficient as seen for instance in complete graphs and cycles of length not divisible by four). At first constant bound of $k = 30$ was proved by Addario-Berry et al. [1], which was later improved to $k = 16$ in 2008 by Addario-Berry et al. [2], $k = 13$ by Wang and Yu [11]; and to $k = 6$ by Kalkowski et al. [8]; and Kalkowski et al. introduced the best bound for $k = 5$ in [9].

In this paper, we show a completely different approach by introducing an algorithm for labelling that improves the bound to $k = 3$, that it is finally value for $k$ in the all graphs which are simple and finite. The following theorem is the main result of this paper.

**Theorem 1.1.** For every graph $G$ without components isomorphic to the complete graph $K_2$, there is a labelling $f : E(G) \to \{1, 2, 3\}$, such that the induced vertex weights $S_i := \sum_{j=1, j \neq i}^{n} f(v_iv_j)$ properly colour $V(G)$.

**Conjecture 1.2.** As above notation, there are distinct numbers of $S_i$'s, $1 \leq i \leq n$, of a graph $G$ of order $n$, for a 1-2-3-edge labelling and vertex colours.

**Conjecture 1.3.** For all graph $G$ of order $n$, there are the $\chi(G)$ numbers of $S_i$'s, $1 \leq i \leq n$, with this 1-2-3-edge labelling and vertex colours, where $\chi(G)$ is the number colours of the vertices on the graph $G$. 
2. The Method and Algorithms

In this section, we introduce two algorithms on a graph $G = (V,E)$ of order $n$. At the first algorithm, we present a vertex ordering on a connected components of $G$, which we call it the ordering algorithm and in continue in the second algorithm we label the edges which their vertices be named by the ordering algorithm and this is the labelling algorithm. We now express the first one.

2.1. The ordering algorithm

At first we order all vertices according to their degrees with a descending sequence as follow:

$$L_1 = \{u_1^{(1)}, u_2^{(1)}, \ldots, u_{n_1}^{(1)}\}, \quad L_2 = \{u_1^{(2)}, u_2^{(2)}, \ldots, u_{n_2}^{(2)}\}, \ldots, \quad L_t = \{u_1^{(t)}, u_2^{(t)}, \ldots, u_{n_t}^{(t)}\}$$

in which $\deg(u_i^{(j)}) = a_j$ for $j = 1, 2, \ldots, t$ and $i = 1, 2, \ldots, \max\{n_1, n_2, \ldots, n_t\}$. We call $L_k$ as the $K$-th layer, so the degree of all of vertices in $K$-th layer is $a_k$, it is trivial that $\sum_{j=1}^t n_j = n$. We have $1 \leq a_t < \cdots < a_2 < a_1 \leq n - 1$. We want to establish an one-to-one corresponding between $L_{j+1} = \{u_1^{(j+1)}, u_2^{(j+1)}, \ldots, u_{n_{j+1}}^{(j+1)}\} \leftrightarrow \{v_{n_1+n_2+\cdots+n_j+1}, v_{n_1+n_2+\cdots+n_j+2}, \ldots, v_{n_1+n_2+\cdots+n_{j+1}}\}$. In the first step, we select an arbitrary vertex in $L_1$ and call it as $v_1$ and if there exist the another vertex in $L_1$ and it is adjacent to $v_1$ we call it $v_2$. Otherwise (i.e., there exist some vertices in $L_1$ and they are not adjacent to $v_1$) we select one of them arbitrary and call it as $v_2$. In continuation, we find a vertex in $L_1$ like $u_r^{(1)}$ by the following priority and we call it $v_3$,

1. $u_r^{(1)}$ is adjacent to $v_1$ and $v_2$,
2. $u_r^{(1)}$ is adjacent to $v_1$ and it is not adjacent to $v_2$,
3. $u_r^{(1)}$ is adjacent to $v_2$ and it is not adjacent to $v_1$,
4. $u_r^{(1)}$ is not adjacent to $v_1$ and $v_2$. 


we call \( u_r^{(1)} = v_3 \), respectively. And go on this process till all of the vertices in \( L_1 \) are named as \( v_1, v_2, \ldots, v_{n_1} \).

In the second layer, we continue the above process with \( L_2 \) in other words we select a vertex in \( L_2 \) like \( u_s^{(2)} \) with the following priority and call it \( v_{n_1+1} \),

\[- u_s^{(2)} \text{ is adjacent to } v_1, v_2, \ldots, v_{n_1}, \]

\[- u_s^{(2)} \text{ is adjacent to } v_1, v_2, \ldots, v_{n_1-1} \text{ and it is not adjacent to } v_{n_1}, \]

\[- \ldots \]

\[- u_s^{(2)} \text{ is not adjacent to } v_1, v_2, \ldots, v_{n_1}, \]

in each of the above cases we set \( u_s^{(2)} = v_{n_1+1} \), respectively. And go on this process till all of the vertices in \( L_2 \) are named as \( v_{n_1+1}, v_{n_1+2}, \ldots, v_{n_1+n_2} \). Finally, each of the vertices in the all of layers are named and established the following one-to-one corresponding:

\[
\{ u_1^{(1)}, u_2^{(1)}, \ldots, u_{n_1}^{(1)} \} \leftrightarrow \{ v_1, v_2, \ldots, v_{n_1} \},
\]

\[
\{ u_1^{(2)}, u_2^{(2)}, \ldots, u_{n_2}^{(2)} \} \leftrightarrow \{ v_{n_1+1}, v_{n_1+2}, \ldots, v_{n_1+n_2} \},
\]

\[
\{ u_1^{(t)}, u_2^{(t)}, \ldots, u_{n_t}^{(t)} \} \leftrightarrow \{ v_{n_1+n_2+\ldots+n_{t-1}+1}, v_{n_1+n_2+\ldots+n_{t-1}+2}, v_{n_1+n_2+\ldots+n_t} \},
\]

in which \( n_1 + n_2 + \ldots + n_t = n \).

For example, we consider the following graph and apply the ordering algorithm to call its vertices (see Figure 1).

The vertices of degree of 8 : \( L_1 = \{ a, b \} = \{ u_1^{(1)}, u_2^{(1)} \} \).

The vertices of degree of 5 : \( L_2 = \{ c, d, e \} = \{ u_1^{(2)}, u_2^{(2)}, u_3^{(2)} \} \).

The vertices of degree of 4 : \( L_3 = \{ f, g, h, i \} = \{ u_1^{(3)}, u_2^{(3)}, u_3^{(3)}, u_4^{(3)} \} \).
The vertices of degree of $3 : L_4 = \{j, k, l\} = \{u_4^{(4)}, u_2^{(4)}, u_3^{(4)}\}$.

The vertices of degree of $2 : L_5 = \{m\} = \{u_5^{(1)}\}$.

**Figure 1.** Solution of ordering algorithm for example in this figure.
So in the first step we have \( v_1 = a, v_2 = b \), the next step since \( c \) is adjacent to \( v_1 \) and \( v_2 \) so \( v_3 = c \) also in this layer there exist \( d, e \) but none of them are not adjacent to \( v_1 \), they are adjacent to \( v_2 \), therefore we set \( v_4 = d \) or \( v_4 = e \), etc.

At the end of this process, we have 

\[
v_1 = a, v_2 = b, v_3 = c, v_4 = d, v_5 = e, v_6 = f, v_7 = g, v_8 = h, v_9 = i, v_{10} = j, v_{11} = k, v_{12} = l, v_{13} = m \quad (\text{see Figure 2}).
\]

![Figure 2](image.png)

**Figure 2.** Solution of labelling algorithm for example in Figure 1.

**Definition 2.1.** For a vertex \( v_i(1 \leq i \leq n) \) the latest edge \( v_iv_j \) is an edge of the graph \( G \) such that \( \deg(v_j) \leq \deg(v_k) \), \( \forall v_iv_k \in E(G) \).

**2.2. The labelling algorithm**

After the run of ordering algorithm, we assign the number 3 to all of edges that incident to \( v_1 \), and calculate \( S_1 \). Then continue with vertex \( v_2 \), and assign the number 3 to each edge that it incident to \( v_2 \) and it has not labelled before, then calculate \( S_2 \). If \( v_1v_2 \notin E \) or \( S_1 \neq S_2 \), then
go to vertex $v_3$. Otherwise $v_1v_2 \in E$ and $S_2 = S_1$, in this case we subtract one unit from the latest edge of $v_2$ that has label 3, so $S_2 = S_1 - 1$ and continue with vertex $v_3$, i.e., we assign 3 to all of edges with endpoint $v_3$ that they were not labelled before.

If $v_1v_3 \in E$ and $S_3 = S_1$, then as before we subtract one unit from the latest edge of $v_3$ which it has not the label 1. Similarly if $v_3v_2 \in E$ and $S_2 = S_3$, we do the same process then we have $S_3 = S_2 - 1$.

We continue this process for vertex $v_i$, $4 \leq i \leq n - 1$. Label every edge of $v_i$ that is not labelled with 3. If there exist an index $j (1 \leq j < i)$ such that $v_iv_j \in E$ and $S_i = S_j$, that subtract one unit from the latest edge of $v_i$, that is labelled with number other one, and do as above. In the other words, we do calculate the new $S_i$ by verify $(v_iv_m) \neq 1$ such that $v_i v_m \in E$ for $i < m \leq n$.

Suppose that there is no such vertex $v_m$, $i < m \leq n$, adjacent with vertex $v_i$ such that the edge $v_i v_m$ have a label 1. It means that $\forall \, m, \, i < m \leq n$, if $v_i v_m \in E$ we have $f(v_i v_m) = 1$. Then find the first vertex $v_l$ before of the vertex $v_i$, where $i > l \geq 1$, and subtract one unit from the latest edge of the vertex $v_l$ and check the condition of $v_l$ with the vertices adjacent and verify $S_l$. Now, we start the process for the vertex $v_{l+1}$ without changed the edge labelling and sum of the labelling for all vertices $v_i$, $1 \leq i \leq l$ and continue this process as above until the last vertex $v_{n-1}$. As an example see the following Figure 3, and finally Figure 4.
Figure 3. The 1-2-3-edge labelling and vertex colours of Figure 1.

Figure 4. The 1-2-3-edge labelling for graph $K_3$ and $K_4$ by the special algorithm on $K_n$.

In the end of this section, we exhibit the following definition:

**Definition 2.2.** A subset of the vertices of a graph $G$ that have the same degree $a_i$ is called a layer $i$. 
Definition 2.3. Let $G = (V, E)$ be a graph. The number of error of the vertex $v$ of $G$ which we denote by $ER_v$, is the number of adjacent vertex of $v$ that are labelled before $v$, and also sum of their labels is equal to $S_v$ in the self step of the algorithm.

Definition 2.4. Let $G = (V, E)$ be a graph with $|V| = n$. For a vertex $v$ of $G$ that is $i$-th in the ordering of the algorithm, define $X_v$ to be the number of vertices that are adjacent and before $v$.

It is easy to see that $X_v \leq i - 1$ and $X_v \leq \deg(v)$, and also $ER_v \leq X_v$ for every vertex $v$.

In the next section, we first prove the algorithms in the special case of the complete graph $K_n$.

3. Special Case of the Algorithm on $K_n$

That is trivial that the run of the algorithms in a regular graph specially in the complete graphs is harder than to any other graphs, because these graphs have one layer. In the following lemma, we will show that the algorithms are correct for the complete graph $K_n$.

Lemma 3.1. There exists an edge-labelling with numbers 1, 2, and 3 for the complete graph $K_n$ with $n$ vertices such that the sum of the labels in all vertices is different.

Proof of Lemma 3.1. First we express the algorithm for $K_n$, because in the next section we will use the lower bound in the general case. These algorithms calculate the maximum sum that there exist for all vertices of the graph $G$. Indeed this lower bound is for the regular graphs, which is the harder case and here we have only one layer. We will prove that the lower bound of $S_n$ in the complete graph $K_n$ is $S_n \geq n + 2$. 
3.1. Algorithm for graph $K_n$

Since all vertices of $K_n$ are adjacent with together and they have the same degree $n - 1$, so we don’t need to run the ordering algorithm, we consider the second algorithm only. Mark the vertices with numbers $1, 2, \ldots, n$, we introduce the modified version of the above algorithm for the complete graph $K_n$.

Label every edges of $v_1$ with 3, so $S_1 = 3n - 3$. For the vertex $v_2$, label every edge that is not labelled before, with 3, and label edge $v_2v_n$ with 2, and then we have $S_2 = 3n - 4$. For the vertex $v_3$, label every edge as above and label edge $v_3v_n$ with 1, so $S_3 = 3n - 5$. For the vertex $v_4$, labelled every edge that is not labelled, with 3 and label edge $v_4v_n$ with 1, and label edge $v_4v_{n-1}$ with 2, so $S_4 = 3(n - 1) - 3$.

For $4 \leq i \leq n - 1$, we continue this process for every vertex $v_i$, in which we subtract one unit from the latest edges (which are not equal to 1) so that $S_i = S_{i-1} - 1$. Now we consider two cases.

- If $i$ is even, that is $i = 2m$, then set $v_iv_n = v_iv_{n-1} = \cdots = v_iv_{n-m+1} = 1$ and label other edges with 3. Now, if $S_i \neq S_{i-1}$ continue this process for $v_{i+1}$, otherwise when $S_i = S_{i-1}$, then set $v_iv_{n-m} = 2$, and we have $S_i = S_{i-1} - 1$.

- If $i$ is odd, that is $i = 2m + 1$, then we set $v_iv_n = v_iv_{n-1} = \cdots = v_iv_{n-m+1} = 1$ and $v_iv_{n-m} = 2$; and label other edges with 3. Now if $S_i \neq S_{i-1}$, we go to the vertex $v_{i+1}$ and we continue as above. If $S_i = S_{i-1}$, then we set $v_iv_{n-m} = 1$, and hence $S_i = S_{i-1} - 1$.

Finally, we obtain $S_1, S_2, \ldots, S_n$, for $v_1, v_2, \ldots, v_n$. Therefore, all edges of the complete graph $K_n$ labelled with 1, 2, and 3 and having the desired property. In fact, for the graph $K_n$, we find $n$ series with finite
sequences of numbers 1, 2, and 3. In the other words, for every 
$i, 1 \leq i \leq n$ set $S_i := \sum_{j=1}^{n} f(v_iv_j)$ such that $f(v_iv_i) = 0$ and 
f$(v_iv_j) \in \{1, 2, 3\}$ for $i \neq j$. Then 
\[(n - 1) \leq S_n < S_{n-1} < \cdots < S_2 < S_1 \leq 3(n - 1).\]
So it is sufficient to show that we can find such labelling for integer $n$
such that $S_n = \min\{S_i|_{i=1}^{n} \geq n - 1.\}$

Now we will show that $S_n \geq n + 2$, where $n \geq 4$, and this means
that “we can label one edge of the vertex $v_n$ with 3, one edge with 2 and
$n - 3$ edges with 1”.

For $n = 1, 2$, there is no such edge-labelling, so $n \geq 3$. Also for the
set of size less than 3, such an edge-labelling not exist. But for $n = 3$, we
will have the edge $v_2v_3$ with label 1 and the edge $v_1v_3$ with 2 and the edge
$v_1v_2$ with 3. Since for $n \geq 4$, we have $n - 1 < n + 2 \leq S_n < S_{n-1} < \cdots <$ 
$S_2 < S_1 \leq 3(n - 1)$, it is sufficient to show that $S_1 - S_n \leq 2n - 6 < 2n - 1 \Rightarrow S_2 - S_n \leq 2n - 5 < 2n - 3$.

Now we prove this by induction on $n$. If $n = 4$, then $G$ is the complete
graph $K_4$, $S_4 = 6 \geq 4 + 2$ and $S_2 - S_4 = 8 - 6 < 3 = 2 \times 4 - 5$, see
Figure 3.

Now suppose that it holds for the complete graph $K_{n-1} = (V', E')$.
Hence $S_2' - S_{n-1}' \leq 2(n - 1) - 5 = 2n - 7$, where that $S_i'$ is sum of the
edge labels of $v_iS_i' = \sum_{j \neq i} f(v_iv_j)$ and $v_iv_j \in E'$ and $f(v_iv_j) \in \{1, 2, 3\}$.

Finally, we are ready to label the complete graph $K_n$, and calculate
each $S_i$ for all vertices $v_i(1 \leq i \leq n)$, by running the algorithm. So we
have $S_1 = 3n - 3$, $S_2 = 3n - 4$, and so on.
Remove the vertex \( v_1 \) of the graph \( K_n \). Suppose that \( R_1 \) is the sum of labels of vertex \( u_1 \) in the subgraph \( K_{n-1} \), which vertex \( u_1 \) is the vertex \( v_2 \) on the graph \( K_n \). So we will have \( R_i = S_{i+1} - 3 \), and \( R_1 = S_2 - 3 = (3(n - 1) - 1) - 3 = 3n - 7 \), such that for the vertex \( u_1 \), \( R_1 = S_1' - 1 \). Since \( u_2v_n = 2 \) in the graph \( K_n \), add the number 1 to the label of the latest edge \( u_1u_{n-1} \) of the vertex \( u_1 \), and modify \( R_1 = (3n - 7) + 1 \).

For the vertex \( u_2 \) add one number to its latest edge \( u_2u_{n-1} \), because \( R_2 = S_2' - 1 \). So the new sum is \( R_2 = S_3 - 3 + 1 = 3n - 7 \).

On the other hand, for the vertex \( u_{n-1} \), since \( R_{n-1} = S_n - 3 \), we can add 2 numbers to two edges of \( K_{n-1} \), so that we have \( S_{n-1}' < S_n \). Thus for adding two edges to it we have \( R_{n-1} = S_n - 3 + 2 = S_n - 1 \).

The relations \( R_2 = S_3 - 2 \) and \( R_{n-1} = S_n - 1 \); imply that

\[
2(n - 1) - 5 = 2n - 7 \geq R_2 - R_{n-1} = S_3 - 2 - S_n + 1.
\]

Since by run of the labelling algorithm on the \( K_n \), \( S_3 = S_2 - 1 \), \( S_2 - 3 - S_n + 1 = S_2 - S_n - 2 \leq 2n - 7 \). Therefore \( S_2 - S_n \leq 2n - 5 \). The proof is now complete for \( K_n \).

We encourage the readers to see other references [3-5] on the edge-labelling and Vertex-coloring of the complete graph \( K_n \).

4. Main Results: The Proof of Theorem 1.1

We prove that, there is an edge-labelling with number 1, 2, and 3 for any connected component \( G = (V, E) \) with \( n \) vertices of a graph \( H \), such that sum of the edge-labels on adjacent vertices are not equal.

It is obvious that for each vertex \( v_i \), of degree \( a_i \) for \( 1 \leq i \leq n \), the sum of \( S_i \) is at least \( a_i \), because there exist the label one and this implies that the algorithm is convergence by the following proof.
We do this labelling with the mentioned algorithms in Section 2. In fact, we will show that the algorithms are convergence for each vertex $v_i$ for integer $i(1 \leq i \leq n)$. In the first, we order the vertices with their degrees in a descending sequence by the ordering algorithm. This labelling algorithm runs on vertex $v_1$ and edge-labels with number 3. So $S_1 = 3a_1$ in which $a_1 = \deg(v_1)$.

Now run the algorithm for the vertex $v_2$. If the vertex $v_2$ is co-layer with $v_1$; i.e., $\deg(v_2) = a_1$ and $v_1, v_2$ are adjacent ($v_1v_2 \in E$), reduce one unit from the latest edge of $v_2$, and $S_2 = 3a_1 - 1 \neq S_1$. and if $v_1v_2 \notin E$, $S_2 = 3a_1$. Also if the vertex $v_2$ is not co-layer with $v_1$, then $\deg(v_2) = a_2 \neq a_1$, and $S_2 = 3a_2$.

Suppose that the algorithm runs for the vertex $v_3, \ldots, v_{i-1}$, for each integer $i$, and then obtain $S_3, \ldots, S_{i-1}$. Now we show that it acts for any vertex $v_i$. Suppose that $\deg(v_i) = d_i$, that $v_i$ is in the layer $r$, where $d_i = a_r$. So these vertices are adjacent with $X_i$ or $X_{v_i}$ vertex $v_1$ until $v_{i-1}$. Thus, it is adjacent with $(d_i - X_i)$ numbers of the vertices $v_{i+1}$ until $v_n$ that are not coloured and edges are not labelled. Suppose that $Y_i$ be the sum of labels of $X_i$ preceding edges of $v_i$, that is, $Y_i = \sum_{j=1}^{i-1} f(v_iv_j)$.

Since the labels are from numbers 1, 2, and 3, we have

(I) : $X_i \leq Y_i \leq 3X_i$.

Also from Definition 2.4,

(II) : $ER_i \leq X_i \leq d_i$.

Hence with respect to the algorithm labels all of $(d_i - X_i)$ remaining edges have number 3. So

$$S_i = 3(d_i - X_i) + Y_i = 3d_i + Y_i - 3X_i.$$
Now (I) implies that
\[ 3d_i + X_i - 3X_i \leq S_i = 3d_i - X_i + Y_i \leq 3d_i + 3X_i - X_i \leq 3d_i + 2X_i. \]

It follows that,
\[ 3d_i - 2X_i \leq S_i = 3d_i - X_i + Y_i. \]

Now from (II), we have
\[ 3d_i - S_i \leq 2X_i \leq 2d_i \Rightarrow d_i \leq S_i \leq 3d_i. \]

For the above inequality, we can show that the labelling algorithm holds for \( v_i, 1 \leq i \leq n - 1 \). This means that for all vertex \( v_i \) the labelling algorithm return \( S_i \) in the favorite bound, such that for all \( v_i v_j \in E \), \( f(v_i v_j) \in \{1, 2, 3\} \) and \( S_i \neq S_j \).

When \( i = n \), we have \( Y_n = \sum_{j=1}^{n-1} f(v_i v_j) \) and
\[ a_i = d_n = X_n \leq n - 1, \]

and
\[ Y_n = S_n = 3(d_n - X_n) + Y_n. \]

Thus (I) implies that
\[ d_n \leq Y_n = S_n \leq 3d_i = 3 \deg(v_n). \]

Now the proof of Theorem 1.1 is complete.

References


   http://dx.doi.org/10.14317/jami.2014.001


