Trans. Tianjin Univ. 2015, 21: 284-287
DOI 10.1007/s12209-015-2431-y

Three-Rainbow Coloring of Split Graphs*

Hu Yumei(胡玉梅), Liu Tingting(刘婷婷)
(School of Sciences, Tianjin University, Tianjin 300072, China)

Abstract: After a necessary condition is given, 3-rainbow coloring of split graphs with time complexity $O(m)$ is obtained by constructive method. The number of corresponding colors is at most 2 or 3 more than the minimum number of colors needed in a 3-rainbow coloring.

Keywords: edge coloring; 3-rainbow coloring; algorithm; split graph

All graphs considered in this paper are simple, connected and undirected. We follow the terminology and notation of Bondy and Murty [1]. A graph G is a split graph, if $V(G)$ can be partitioned into a clique and an independent set. Let G be a nontrivial connected graph of order n on which an edge coloring is defined, where the adjacent edges may be colored with the same color. A path P is a rainbow path if no two edges of P are colored with the same color. A graph G is rainbow connected if G contains a ($u-v$) rainbow path for every pair u, v of distinct vertices of G. If G is a rainbow connected by coloring c, then c is called a rainbow coloring of G. The rainbow connection number $rc(G)$ of G is the minimum number of colors that results in a rainbow connected graph G, which is introduced by Chartrand *et al*[2]. Nowadays, there have been various investigations on the good upper bounds for rainbow connection number in terms of graph parameters [3-6].

In 2007, Ericksen [7] stated that in the case of emergency, the information transfer paths should be assigned between agencies, which may have other agencies as intermediaries that require a large enough number of passwords and firewalls. The organization is prohibitive to intruders, but small enough to ensure that any path between agencies has no password repeated. This situation can be studied by means of rainbow colorings of graphs.

Subsequently, the rainbow connection number was generalized[8]. A tree T is a rainbow tree if no two edges of T are colored with the same color. Let k be a fixed integer with $2 \leqslant k \leqslant n$. An edge coloring of G is called a k-rainbow coloring if for every set S of k vertices of G, there exists a rainbow tree in G containing the vertices of S. The k-rainbow index $rx_k(G)$ is the minimum number of colors needed in a k-rainbow coloring of G. Clearly, when $k = 2$, $rx_2(G)$ is the rainbow connection number $rc(G)$.

For $k = 3$, Chartrand *et al*[8] determined the precise value of 3-rainbow index for trees and complete graphs.

Lemma 1[8] Let T be a tree of order $n \geqslant 3$, then $rx_3(T) = n-1$.

Lemma 2[8] Let K_n be a complete graph of order $n \geqslant 3$, then $rx_3(G) = 2,\ 3 \leqslant n \leqslant 5$; $rx_3(G) = 3, n \geqslant 6$.

For the complexity of rainbow connection, it was shown that computing the rainbow connection number of an arbitrary graph is NP-hard, and the determination of whether a given edge-colored graph is rainbow connected is NP-complete[9]. Chandran and Rajendraprasad[10] also showed that determining whether $rc(G) \leqslant 3$ remains NP-complete even for split graphs; an algorithm using at most $rc(G)+1$ colors to ensure that G is rainbow connected was also given with time-complexity $O(m)$.

For the k-rainbow index of split graphs, it should be NP-hard to compute the 3-rainbow index of graphs since whether $rc(G) \leqslant 3$ remains NP-complete. Thus, it is unlikely that there exists a polynomial-time algorithm to give a 3-rainbow coloring of split graphs with exact colors. In this paper, we try to show a polynomial-time algorithm of 3-rainbow coloring for a split graph G. First, by a constructive method, the desired algorithm is given in Section 1. Then, by analyzing the necessary condition, we show the effectiveness of the algorithm in Section 2. And we will prove that the time complexity is $O(m)$.

Accepted date: 2014-10-08.
*Supported by the National Natural Science Foundation of China (No.11001196).
Hu Yumei, born in 1977, female, Dr, associate Prof.
Correspondence to Hu Yumei, E-mail : huyumei@tju.edu.cn.

Besides, it is worth mentioning some related results[11-15]. Hammer and Simeone[11] showed that for a split graph with the degree sequence $d_1 \geqslant d_2 \geqslant \cdots \geqslant d_n$, $\{v_i \in V(G): d_i \geqslant i-1\}$ is a maximum clique in G and $\{v_i \in V(G): d_i \leqslant i-1\}$ is a maximum independent set in G. Furthermore, the vertices of a graph can be sorted on the basis of degrees in $O(n)$ times. Therefore, when we input a split graph, we assume that the order of the vertices, even a maximum clique and a maximum independent set also can be given as input to our algorithm.

1 Algorithm

Algorithm 1 Coloring split graphs

Input $G([n],E)$, a connected split graph with a maximum clique K.

Output A 3-rainbow coloring $c_G(G) \to \{1,\cdots,rx_3(G)+2\}$ or $\{1,\cdots,rx_3(G)+3\}$

$N_j = \{v_i \in V(G)\setminus K, d_i = j\}, n_j = |N_j|$, for $j=1,2$

$N_3 = \{v_i \in V(G)\setminus K, d_i \geqslant 3\}, n_3 = |N_3|$

$l = n_1 + n_2 + n_3$

if $l \geqslant 3$ then

let e_i^j be the edges incident with N_j, $j=1,2,3$

$c_G(e_i^1) = i$, for every edge $e_i^1 \in \{e_1^1, e_2^1, \cdots, e_{n_1}^1\}$

for $v_i \in N_3$ do

$c_G(e_1^3)=1, c_G(e_2^3)=2, c_G(e^3)=3, e^3 \in \{e_3^3,\cdots,e_{d_i}^3\}$

end for

$t = \min\{x : n_1 x + x(x-1) \geqslant n_2\}$

for $v_i \in N_2$ do

let $\{e_{i_1}^2, e_{i_2}^2\}$ be the edges incident with v_i.

$(c_G(e_{i_1}^2), c_G(e_{i_2}^2)) = (p,q)$, where $p \in \{1,2,\cdots,n_1,n_1+1,\cdots,n_1+t\}$

$q \in \{n_1+1,\cdots,n_1+t\}, p \neq q.$ And for different i,i', $(c_G(e_{i_1}^2),\ c_G(e_{i_2}^2)) \neq (c_G(e_{i'_1}^2), c_G(e_{i'_2}^2))$

// Since $n_1 t + t(t-1) \geqslant n_2$, we can guarantee that different vertices correspond to different (p,q).

end for

$r = n_1 + t$

end if

if $l \leqslant 2$ then

label the l vertex in $N_1 \cup N_2 \cup N_3$ by $1,2,\cdots,l$, the label of v_i is denoted by $code(i)$, $c_G(e) = code(i)$, for all edges incident with v_i.

$r = l$

end if

$c_G(e) = c'(e)$, for all edges $e \in E(K)$.

// c' is the 3-rainbow coloring of Lemma 2,

if $3 \leqslant |K| \leqslant 5$, $c'(e) \in \{r+1, r+2\}$;

if $|K| \geqslant 6$, $c'(e) \in \{r+1, r+2, r+3\}$

return c_G

2 Effectiveness

For a given connected split graph G, let E_i be the edge set that is incident with $N_i, i=1,2,3,$ where N_i is introduced in Section 1. Let c be any 3-rainbow coloring and $c(E_1)$ the set of colors used in E_1.

Lemma 3 Let e_1^1 and e_2^1 be any two edges in E_1, then every 3-rainbow coloring of G must assign different colors to e_1^1 and e_2^1.

Proof Without loss of generality, we assume that $e_1^1 = v_1 v_1'$, $e_2^1 = v_2 v_2'$, where $v_1, v_2 \in N_1$. We choose $\{v_1, v_2\} \in S$, then the tree connecting S must contain e_1^1, e_2^1, thus there is no rainbow tree connecting S, a contradiction. Hence, the Lemma 3 holds.

Lemma 4 For any 3-rainbow coloring c and any vertex v in N_2, $|c(E_1) \cap (c(e_1^2) \cup c(e_2^2))| \leqslant 1$, where e_1^2 and e_2^2 are two edges that are incident with the vertex v.

Proof Assume that $|c_1(E_1) \cap (c_1(e_1^2) \cup c_1(e_2^2))| \geqslant 2$, i.e., there is some vertex v' and 3-rainbow coloring c_1 such that the two edges incident with vertex v' receive different colors all used in E_1 in the 3-rainbow coloring c_1. Then we choose vertex v' and vertices in N_1 corresponding to edges obtaining the above different colors. Clearly, there is no rainbow tree connecting them, a contradiction.

Note that according to Lemma 4, the choice of color sets in Section 1 is reasonable.

Lemma 5 For any 3-rainbow coloring, if one edge that is incident with some vertex in N_2 obtains color p (the color used in E_1), and the other edge that is incident with the same vertex receives color q that is not used in E_1, then there is no vertex corresponding to two edges colored by color p, q in N_2.

Proof Assume that there are two vertices in N_2 such that their corresponding edges obtain colors p, q, then we choose the two vertices and the vertex in N_1 whose corresponding edge receives color p to be set S. Then we cannot find any tree containing S, a contradiction. Hence, Lemma 5 holds.

We call a set of colors to be shown completely in vertices if there is no other color appearing on the edges incident with the vertices besides the set of colors.

Lemma 6 For any 3-rainbow coloring, any two colors that are not used in E_1 are shown to be completely in at most two vertices of N_2.

Proof If there are three vertices v_i, v_j, v_k such that $(c_G(e_{i_1}^2),\ c_G(e_{i_2}^2)) \in \{p,q\} \times \{p,q\}$, $(c_G(e_{j_1}^2),\ c_G(e_{j_2}^2)) \in \{p,q\} \times \{p,q\}$, $(c_G(e_{k_1}^2),\ c_G(e_{k_2}^2)) \in \{p,q\} \times \{p,q\}$, where $p,q \in \{n_1+1,\cdots,n_1+t\}$. By pigeonhole principle, the tree containing the three vertices must include the same color edge, a contradiction. The conclusion holds.

Note that in view of Lemma 5 and Lemma 6, in Section 1 the format $(c_G(e_{i_1}^2), c_G(e_{i_2}^2)) = (p,q)$ is justifiable, where $p \in \{1,2,\cdots,n_1,n_1+1,\cdots,n_1+t\}$, $q \in \{n_1+1,\cdots,n_1+t\}$, $p \neq q$.

Now we can derive our main result as follows.

Theorem 1 Let G be a connected split graph, then the coloring obtained by Algorithm 1 is 3-rainbow coloring of G, using at most $rx_3(G)+2$ or $rx_3(G)+3$ colors. Moreover, the corresponding time-complexity is $O(m)$.

Proof Firstly, we show that the c_G obtained by Algorithm 1 is a 3-rainbow coloring. According to the definition, for any three vertices of G, we only need to find a rainbow tree connecting them in G. Assume $S = \{v_1, v_2, v_3\}$. If $S \subseteq K$, then there is a rainbow tree connecting S in G. For $\{v_1, v_2\} \in K, v_3 \in V \setminus K$, since G is connected, there exists an edge e connecting v_3 and K. Let $e = v_3v_4$ (in particular, $v_4 = v_1$ or v_2). By the above analysis, we can find a rainbow tree T' containing $\{v_1, v_2, v_4\}$. By the coloring schemes, $c_G(e)$ receives the color different from T'. Thus $T = T' \cup e$ is the rainbow tree containing S. For $v_1 \in K, \{v_2, v_3\} \subseteq V \setminus K$, we can prove that the conclusion holds similarly. For $S \subseteq V \setminus K$, from the coloring schemes, whatever the three vertices of S belong to $N_1 \cup N_2 \cup N_3$, there exist three edges e_1, e_2, e_3 colored by different colors connecting the three vertices in S and some vertices in S' in K. Obviously, there is T' connecting S'. S' may contain one, two, or three vertices, then the corresponding T' may be a vertex or a path. Hence, $T = T' \cup \{e_1, e_2, e_3\}$ is our desired tree.

Then, we check that at most $rx_3(G)+2$ or $rx_3(G)+3$ colors are used. For a given connected split graph G, the set N_1, N_2, N_3 of the algorithm may be empty, so we consider the following cases.

Case 1 $N_1 \neq \varnothing, N_2 = N_3 = \varnothing$

$G = K \cup E_1$. By Lemma 1, $rx_3(G) \geqslant n_1 = r$. If $3 \leqslant |K| \leqslant 5$, then $r+2$ colors are used. If $|K| \geqslant 6$, then $r+3$ colors are used.

Case 2 $N_3 \neq \varnothing, N_2 = \varnothing$

$G = K \cup E_1 \cup E_3$, where E_3 must be unempty and E_1 may be empty. In this case, if $n_1 \geqslant 3$, by Lemma 3, the conclusion holds. Otherwise, $n_1 \leqslant 2$. When $l \geqslant 3$, we choose the three vertices of $N_1 \cup N_3$ as the set S. Then the rainbow tree connecting S has size at least 3, thus we get $rx_3(G) \geqslant 3$. In such circumstances, the algorithm uses 5 or 6 colors. When $l \leqslant 2$, it is easy to see that a tree containing three vertices has size at least 2, i.e., $rx_3(G) \geqslant 2$. It can be seen that Algorithm 1 uses 4 or 5 colors. From the above discussion, the conclusion holds.

Case 3 $N_2 \neq \varnothing$

$G = K \cup E_1 \cup E_2 \cup E_3$, where E_2 must be unempty, E_1 and E_3 may be empty.

When $l \geqslant 3$, according to Lemmas 4, 5 and 6, we know that at least $\min\{x : n_1x + x(x-1) \geqslant n_2\}$ colors are needed to guarantee that there exists a rainbow tree connecting the vertices in N_2. When $l \leqslant 2$, it can be easily checked. In summary, if there is a rainbow tree connecting any three vertices in $N_1 \cup N_2 \cup N_3$, then the colors used in $E_1 \cup E_2 \cup E_3$ have at least r colors, so $rx_3(G) \geqslant r$. The algorithm uses at most $r+2$ or $r+3$ colors.

Finally, the algorithm visits each edge exactly once and $t = \min\{x : n_1x + x(x-1) \geqslant n_2\}$ can be solved in constant time. Hence, the time-complexity is $O(m)$.

3 Conclusions

A 3-rainbow coloring of split graphs is given in this paper, which uses at most $rx_3(G)+2$ or $rx_3(G)+3$ colors. Moreover, the time-complexity is $O(m)$.

References

[1] Bondy J A, Murty U S R. *Graph Theory* [M]. Springer, UK, 2008.

[2] Chartrand G, Johns G L, MeKeon K A *et al.* Rainbow connection in graphs [J]. *Mathematica Bohemica*, 2008, 133(1): 85-98.

[3] Caro Y, Lev A, Roditty Y *et al.* On rainbow connection [J]. *Electronic Journal of Combinatorics*, 2008, 15(1): R57.

[4] Krivelevich M, Yuster R. The rainbow connection of a graph is (at most) reciprocal to its minimum degree [J]. *Journal of Graph Theory*, 2010, 63(3):185-191.

[5] Li X, Shi Y, Sun Y. Rainbow connections of graphs: A survey [J]. *Graphs and Combinatorics* , 2013, 29(1):1-38.

[6] Li X, Sun Y. *Rainbow Connections of Graphs* [M]. Springer Briefs in Mathematics. Springer, New York, USA, 2012.

[7] Ericksen A B. A matter of security [J]. *Graduating Engineer and Computer Careers*, 2007: 24-28.

[8] Chartrand G, Okamoto F, Zhang P. Rainbow trees in graphs and generalized connectivity [J]. *Networks*, 2010, 55(4): 360-367.

[9] Chakraborty S, Fischer E, Matsliah A *et al*. Hardness and algorithms for rainbow connection [J]. *Journal of Combinatorial Optimization*, 2011, 21(3): 330-347.

[10] Chandran L S, Rajendraprasad D. Rainbow coloring of spliting and threshold graphs [C]. In: *18th Annual International Conference*, COCOON 2012. Sydney, Australia, 2012.

[11] Hammer P L, Simeone B. The splittance of a graph [J]. *Combinatorica*, 1981,1(3):275-284.

[12] Liu T, Hu Y. The 3-rainbow index of graph operations [J]. *WSEAS Transactions on Mathematics,* 2014, 13:161-170.

[13] Liu T, Hu Y. Some upper bounds for 3-rainbow index of graphs [J]. *Journal of Combinatorial Mathematics and Combinatorial Computing*, accepted.

[14] Seward H H. Information Sorting in the Application of Electronic Digital Computers to Business Operations [D]. Massachusetts Institute of Technology, Cambridge, USA, 1954.

[15] He J, Liang H. On rainbow-*k*-connectivity of random graphs [J]. *Information Processing Letters*, 2012, 112(10): 406-410.

(Editor: Wu Liyou)