Journal of Combinatorial Optimization (2020) 40:1008–1019
https://doi.org/10.1007/s10878-020-00644-7

Some algorithmic results for finding compatible spanning circuits in edge-colored graphs

Zhiwei Guo[1,2] · Hajo Broersma[2] · Ruonan Li[1,3] · Shenggui Zhang[1,3]

Published online: 4 September 2020

Abstract
A compatible spanning circuit in a (not necessarily properly) edge-colored graph G is a closed trail containing all vertices of G in which any two consecutively traversed edges have distinct colors. Sufficient conditions for the existence of extremal compatible spanning circuits (i.e., compatible Hamilton cycles and Euler tours), and polynomial-time algorithms for finding compatible Euler tours have been considered in previous literature. More recently, sufficient conditions for the existence of more general compatible spanning circuits in specific edge-colored graphs have been established. In this paper, we consider the existence of (more general) compatible spanning circuits from an algorithmic perspective. We first show that determining whether an edge-colored connected graph contains a compatible spanning circuit is an NP-complete problem. Next, we describe two polynomial-time algorithms for finding compatible spanning circuits in edge-colored complete graphs. These results in some sense give partial support to a conjecture on the existence of compatible Hamilton cycles in edge-colored complete graphs due to Bollobás and Erdős from the 1970s.

Keywords Edge-colored graph · Compatible spanning circuit · NP-complete problem · Polynomial-time algorithm

Mathematics Subject Classification 05C15 · 05C38 · 05C45 · 05C85

Supported by NSFC (Nos. 11671320 and U1803263), CSC (No. 201806290049) and the Fundamental Research Funds for the Central Universities (Nos. 31020180QD124 and 3102019ghjd003).

✉ Hajo Broersma
h.j.broersma@utwente.nl

1 School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, Shaanxi, People's Republic of China

2 Faculty of EEMCS, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

3 Xi'an-Budapest Joint Research Center for Combinatorics, Northwestern Polytechnical University, Xi'an 710129, Shaanxi, People's Republic of China

1 Introduction

In this paper we consider only finite undirected simple graphs. For terminology and notations not defined here, we refer the reader to the textbook of Bondy and Murty (2008).

Let G be a graph. We use $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively. For a vertex v of G, we denote by $E_G(v)$ the set of edges of G incident with v, and we denote by $N_G(v)$ the set of neighbors of v in G. The *degree* of a vertex v in a graph G, denoted by $d_G(v)$, is defined to be the cardinality of $E_G(v)$. We write $\Delta(G) = \max\{d_G(v) \mid v \in V(G)\}$. If no ambiguity can arise, we will denote $E_G(v)$, $N_G(v)$ and $d_G(v)$ by $E(v)$, $N(v)$ and $d(v)$, respectively.

A *spanning circuit* in a graph G is defined as a closed trail that visits (contains) each vertex of G. A *Hamilton cycle* of G refers to a spanning circuit visiting each vertex of G exactly once; an *Euler tour* of G refers to a spanning circuit traversing each edge of G. Hence, a spanning circuit is a common relaxation of a Hamilton cycle and an Euler tour. A graph is said to be *hamiltonian* if it contains a Hamilton cycle, and a graph is said to be *eulerian* if it admits an Euler tour.

An *edge-coloring* of a graph G is defined as a mapping $c : E(G) \to \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers. An *edge-colored graph* refers to a graph with a fixed edge-coloring. Two edges of a graph are said to be *consecutive* with respect to a trail (with a fixed orientation) if they are traversed consecutively along the trail. A *compatible spanning circuit* in an edge-colored graph refers to a spanning circuit in which any two consecutive edges have distinct colors. An edge-colored graph is said to be *properly colored* if any two adjacent edges (i.e., edges sharing exactly one end vertex) of the graph have distinct colors, and an edge-colored graph is *rainbow* if each pair of edges of the graph has distinct colors. Thus, a compatible Hamilton cycle is properly colored, and a properly colored spanning circuit is compatible. Conversely, a compatible spanning circuit is obviously not necessarily properly colored. Thus, a compatible spanning circuit can be viewed as a generalization of a properly colored spanning circuit. Compatible spanning circuits are of interest in graph theory applications, for example, in genetic and molecular biology (Pevzner 2000; Szachniuk et al. 2014, 2009), in the design of printed circuits and wiring boards (Tseng et al. 2010), and in channel assignment of wireless networks (Ahuja 2010; Sankararaman et al. 2014).

Let G be an edge-colored graph. We use $c(e)$ to denote the color appearing on the edge e of G, and we use $C(G)$ to denote the set of colors appearing on the edges of G. Let $d^i_G(v)$ denote the cardinality of the set $\{e \in E_G(v) \mid c(e) = i\}$ for a vertex $v \in V(G)$ and a color $i \in C(G)$. We let $\Delta^{mon}_G(v) = \max\{d^i_G(v) \mid i \in C(G)\}$ for a vertex $v \in V(G)$, and we let $\Delta^{mon}(G) = \max\{\Delta^{mon}_G(v) \mid v \in V(G)\}$; these two parameters are called the *maximum monochromatic degree* of a vertex v of G and the *maximum monochromatic degree* of an edge-colored graph G, respectively. The *color degree* of a vertex v of an edge-colored graph G, denoted by $cd_G(v)$, is defined to be the number of colors appearing on the edges of G incident with v. When no confusion can arise, we will use $d^i(v)$, $\Delta^{mon}(v)$ and $cd(v)$ instead of $d^i_G(v)$, $\Delta^{mon}_G(v)$ and $cd_G(v)$, respectively.

From a sufficient condition perspective, the existence of two kinds of extremal compatible spanning circuits, i.e., compatible Hamilton cycles and compatible Euler tours in specific edge-colored graphs has been studied extensively. For more details on the topic, we refer the reader to Alon and Gutin (1997), Bollobás and Erdős (1976), Chen and Daykin (1976), Daykin (1976), Fleischner and Fulmek (1990), Kotzig (1968), Lo (2016) and Shearer (1979). On the other hand, Benkouar et al. (1996), from an algorithmic perspective, considered the existence of compatible Euler tours in edge-colored eulerian graphs. Benkouar et al. (1996) provided a polynomial-time algorithm for finding a compatible Euler tour in an edge-colored eulerian graph G in which $\Delta^{mon}(v) \leq d(v)/2$ for each vertex v of G. Independently, Pevzner (1995) described a similar algorithm for solving the same problem.

In recent work (Guo et al. 2020a, b), sufficient conditions for the existence of more general compatible spanning circuits (i.e., not necessarily a compatible Hamilton cycle or Euler tour) in specific edge-colored graphs have been established.

In this paper, we consider the existence of compatible spanning circuits in edge-colored graphs from an algorithmic perspective. We first prove the following complexity result by a simple reduction from a result due to Garey et al. (1974).

Theorem 1.1 *The decision problem of determining whether an edge-colored connected graph contains a compatible spanning circuit is NP-complete.*

We postpone the proofs of all our results in order not to interrupt the flow of the narrative. Motivated by the above NP-completeness result, we consider the existence of compatible spanning circuits in specific classes of edge-colored graphs from an algorithmic perspective, and we analyze the complexity of the associated algorithms.

A number of sufficient conditions for the existence of compatible Hamilton cycles in edge-colored complete graphs have been obtained (see Alon and Gutin 1997; Bollobás and Erdős 1976; Chen and Daykin 1976; Daykin 1976; Lo 2016; Shearer 1979). In particular, Bollobás and Erdős (1976) considered the problem and proposed the following conjecture on the existence of compatible Hamilton cycles in edge-colored complete graphs back in the 1970s (see Conjecture 1.1). Recently, Lo (2016) proved that this conjecture is true asymptotically. Throughout the rest of this paper, we use K_n^c to denote an edge-colored complete graph on n vertices, where $n \geq 3$.

Conjecture 1.1 (Bollobás and Erdős 1976) If $\Delta^{mon}(K_n^c) < \lfloor n/2 \rfloor$, then K_n^c contains a compatible Hamilton cycle.

In the rest of this paper, we first deal with the existence of compatible spanning circuits (with no restrictions) in graphs K_n^c with $\Delta^{mon}(K_n^c) \leq \lfloor (n-1)/2 \rfloor$, as follows.

Theorem 1.2 *If $\Delta^{mon}(K_n^c) \leq \lfloor (n-1)/2 \rfloor$, then K_n^c contains a compatible spanning circuit. Moreover, such a compatible spanning circuit can be found by an $O(n^4)$ algorithm.*

Remark 1.1 The following example, extended from a construction given by Fujita and Magnant (2011), shows that the bound on $\Delta^{mon}(K_n^c)$ in Theorem 1.2 is tight.

Example 1.1 Let G be a complete graph on n ($n \geq 3$) vertices, and let u be one of the vertices of G. We label the remaining vertices with $v_1, \ldots, v_{n-1}$, respectively, and we color the edge uv_i with color i for each v_i, where $1 \leq i \leq n-1$. Let $H = G - u$, and consider a decomposition of the edges of H into $\lfloor (n-2)/2 \rfloor$ Hamilton cycles (together with one perfect matching M, if n is odd). We arbitrarily orient these Hamilton cycles (and M, if n is odd) such that they become directed cycles (a directed perfect matching). We color the edge $v_i v_j$ with color j if the arc $\overrightarrow{v_i v_j}$ is an arc of one of these Hamilton cycles (perfect matching). This defines an edge-coloring of G, thus a K_n^c.

One can check that the edge-colored complete graph K_n^c of Example 1.1 satisfies $\Delta^{mon}(K_n^c) = \lfloor (n-1)/2 \rfloor + 1$, but it contains no compatible spanning circuit, because such a circuit cannot visit the vertex u compatibly.

We next deal with the existence of compatible spanning circuits visiting every vertex, except for one specific vertex, exactly $(n-2)/2$ times in graphs K_n^c, and we obtain the following result.

Theorem 1.3 *Let n be an even integer such that $n \geq 4$. If $\Delta^{mon}(K_n^c) \leq (n-2)/2$ and $cd(v_0) \geq n - \lfloor (\sqrt{4n-3}+1)/2 \rfloor$ for some vertex v_0 of K_n^c, then K_n^c contains a compatible spanning circuit visiting every vertex of K_n^c, except for v_0, exactly $(n-2)/2$ times. Moreover, such a compatible spanning circuit can be found by an $O(n^4)$ algorithm.*

Remark 1.2 The edge-colored complete graphs on even n ($n \geq 4$) vertices of Example 1.1 also show that the bound on $\Delta^{mon}(K_n^c)$ in Theorem 1.3 is tight. However, we do not know whether the bound on $cd(v_0)$ in Theorem 1.3 is tight.

The rest of the paper deals with the proofs of our three results.

2 Proof of Theorem 1.1

Our proof is based on the NP-completeness of the following special case of the Hamilton problem, an early complexity result due to Garey et al. (1974).

Problem 2.1 (Garey et al. 1974)

Instance: A connected graph G with $\Delta(G) = 3$.
Question: Does G contain a Hamilton cycle?

The problem above can easily be reduced to the following special case of the decision problem we stated in Theorem 1.1.

Problem 2.2 **Instance:** An edge-colored connected graph G^c with $\Delta(G^c) = 3$.
Question: Does G^c contain a compatible spanning circuit?

First of all, Problem 2.2 clearly belongs to the class NP: for any candidate subgraph H corresponding to a compatible spanning circuit in G^c, it can be verified in polynomial time whether the subgraph H contains all vertices of G^c, $d_H(v) = 2$ and $\Delta_H^{mon}(v) = 1$ for each vertex v of H.

For any instance G of Problem 2.1, we construct a rainbow edge-colored graph by coloring all edges of G with pairwise distinct colors, to obtain an instance G^c of Problem 2.2. It is obvious that the graph G contains a Hamilton cycle if and only if the edge-colored graph G^c contains a compatible spanning circuit.

It follows directly from our construction that the reduction above is polynomial. This proves that Problem 2.2 is NP-complete. Since Problem 2.2 is a special case of the decision problem we stated in Theorem 1.1, the result is immediate. □

3 Proofs of Theorems 1.2 and 1.3

Before proceeding with our proofs, we first introduce some additional terminology.

For a given trail $T_i = x_1x_2\cdots x_i$ $(i \geq 2)$ of a graph H, we use H_i to denote the (spanning) subgraph of H obtained from H by deleting all the edges of T_i. For a given (compatible) trail $T_i = x_1x_2\cdots x_i$ $(i \geq 2)$ of an edge-colored graph H, an edge x_ix_{i+1} of H is said to be *suitable* for T_i in H if $x_ix_{i+1} \in E_{H_i}(x_i)$ and $c(x_ix_{i+1})$ satisfies that $c(x_ix_{i+1}) \neq c(x_{i-1}x_i)$ and $d^{c_0}_{H_i}(x_i) = \max_{c\neq c(x_{i-1}x_i)} \{d^c_{H_i}(x_i)\}$, where $c_0 = c(x_ix_{i+1})$.

We prove Theorem 1.2 by considering the following polynomial algorithm, and proving its correctness. We use CSC as shorthand for compatible spanning circuit.

Algorithm 1 Finding a CSC in K^c_n with $\Delta^{mon}(K^c_n) \leq \lfloor (n-1)/2 \rfloor$.

Input: A graph K^c_n with $\Delta^{mon}(K^c_n) \leq \lfloor (n-1)/2 \rfloor$;
Output: A CSC T of K^c_n;
Step 1. If n is odd, then let $H = K^c_n$; otherwise, choose an arbitrary perfect matching M of K^c_n, and let $H = K^c_n - M$;
Step 2. Choose an arbitrary vertex x_1 of H, and put $T_1 = x_1$;
Choose the next vertex x_2 such that $c(x_1x_2)$ is (one of) the least frequent colors among the edges of $E_H(x_1)$, and put $T_2 = x_1x_2$;
Step 3. Based on $T_i = x_1x_2\cdots x_i$ $(i \geq 2)$, build up $T_{i+1} = x_1x_2\cdots x_ix_{i+1}$ according to the following **rules**:
if $V(H)\setminus V(T_i) = \emptyset$ and $x_ix_1 \in E_{H_i}(x_i)$, as well as $c(x_{i-1}x_i) \neq c(x_ix_1)$ and $c(x_ix_1) \neq c(x_1x_2)$ **then**
put $T = T_{i+1} = x_1x_2\cdots x_ix_1$, and go to **Step 5**;
else
if there exists a vertex $x_{i+1} \in V(H)\setminus V(T_i)$ such that x_ix_{i+1} is suitable for T_i in H **then**
choose such a vertex x_{i+1} and preferentially choose the vertex x_{i+1} such that $c(x_ix_{i+1}) = c(x_1x_2)$ when $x_i = x_1$, and put $T_{i+1} = x_1x_2\cdots x_ix_{i+1}$;
else
choose a vertex $x_{i+1} \in V(T_i)$ such that x_ix_{i+1} is suitable for T_i in H, and preferentially choose the vertex x_{i+1} such that $c(x_ix_{i+1}) = c(x_1x_2)$ when $x_i = x_1$, and put $T_{i+1} = x_1x_2\cdots x_ix_{i+1}$;
end if
end if
Step 4. $i \leftarrow i+1$, and go to **Step 3**;
Step 5. T is a CSC of K^c_n; terminate the process;
return T.

The ideas behind Algorithm 1 were inspired by similar ideas due to Pevzner (1995) for an efficient algorithm to construct a compatible Euler tour in an edge-colored eulerian graph G in which $\Delta^{mon}(v) \leq d(v)/2$ for each vertex v of G.

Next, we show the correctness of Algorithm 1 by proving the following lemmas, with the notations H_i, T_i, H and T defined as above.

Lemma 3.1 *We have $\Delta^{mon}_{H_i}(v) \leq d_{H_i}(v)/2$ for each integer i with $i \geq 2$ such that $T_i \neq T$, and each vertex v of H_i, excluding possibly x_1 and x_i.*

Proof Suppose, to the contrary, that the statement of Lemma 3.1 does not hold. Let i_0 be the minimum integer such that Lemma 3.1 fails. Clearly, we have $i_0 > 2$. Thus, for some color c and some vertex v distinct from x_1 and x_{i_0}, we have $d^c_{H_{i_0}}(v) > d_{H_{i_0}}(v)/2$. It is not difficult to see that $v = x_{i_0-1}$; otherwise Lemma 3.1 would already fail for the integer $i_0 - 1$. It follows from $d^c_{H_{i_0}}(x_{i_0-1}) > d_{H_{i_0}}(x_{i_0-1})/2$ and $d_{H_{i_0}}(x_{i_0-1})$ is even that $d^c_{H_{i_0}}(x_{i_0-1}) \geq d_{H_{i_0}}(x_{i_0-1})/2 + 1$.

Obviously, we have $d^c_{H_{i_0-1}}(x_{i_0-1}) \geq d^c_{H_{i_0}}(x_{i_0-1}) \geq d_{H_{i_0}}(x_{i_0-1})/2 + 1 = (d_{H_{i_0-1}}(x_{i_0-1}) - 1)/2 + 1 = (d_{H_{i_0-1}}(x_{i_0-1}) + 1)/2$.

We first prove the following claim in order to complete the proof of Lemma 3.1.

Claim 1 $c(x_{i_0-1}x_{i_0}) = c$.

Proof Suppose, to the contrary, that $c(x_{i_0-1}x_{i_0}) \neq c$. Recall that $d^c_{H_{i_0-1}}(x_{i_0-1}) \geq (d_{H_{i_0-1}}(x_{i_0-1}) + 1)/2$ and $T_{i_0} \neq T$. It follows that $c(x_{i_0-2}x_{i_0-1}) = c$ by the rules of Step 3 of Algorithm 1. Thus, we have $d^c_{H_{i_0-2}}(x_{i_0-1}) = d^c_{H_{i_0-1}}(x_{i_0-1}) + 1 \geq (d_{H_{i_0-1}}(x_{i_0-1}) + 1)/2 + 1 = d_{H_{i_0-2}}(x_{i_0-1})/2 + 1$, contradicting the minimality of i_0. This confirms our claim. □

By Claim 1, we have $c(x_{i_0-2}x_{i_0-1}) \neq c$ and $c(x_{i_0-1}x_{i_0}) = c$. Hence, we have $d^c_{H_{i_0-2}}(x_{i_0-1}) = d^c_{H_{i_0}}(x_{i_0-1}) + 1 \geq d_{H_{i_0}}(x_{i_0-1})/2 + 1 + 1 = (d_{H_{i_0-2}}(x_{i_0-1}) - 2)/2 + 1 + 1 = d_{H_{i_0-2}}(x_{i_0-1})/2 + 1$, contradicting the minimality of i_0. This completes the proof of Lemma 3.1. □

Lemma 3.2 *We have $\Delta^{mon}_{H_i}(x_1) \leq \lceil d_{H_i}(x_1)/2 \rceil$ for each integer i with $i \geq 2$ such that $x_i \neq x_1$.*

Proof By Step 2 of Algorithm 1, we have $\Delta^{mon}_{H_2}(x_1) = \Delta^{mon}_{H_1}(x_1) = \Delta^{mon}_{H}(x_1)$. For the case that n is odd, we have $\Delta^{mon}_{H}(x_1) = \Delta^{mon}_{K^c_n}(x_1) \leq (n-1)/2 = \lceil (n-2)/2 \rceil = \lceil d_{H_2}(x_1)/2 \rceil$. For the case that n is even, we have $\Delta^{mon}_{H}(x_1) \leq \Delta^{mon}_{K^c_n}(x_1) \leq (n-2)/2 = \lceil (n-3)/2 \rceil = \lceil d_{H_2}(x_1)/2 \rceil$. Thus, Lemma 3.2 holds for $i = 2$.

Next, we assume that $i \geq 3$. Suppose, to the contrary, that the statement of Lemma 3.2 does not hold. Let i_0 be the minimum integer such that Lemma 3.2 fails. Since $x_{i_0} \neq x_1$, we conclude that $x_{i_0-1} = x_1$; otherwise Lemma 3.2 would already fail for some integer less than i_0. Thus, for some color c, we have $d^c_{H_{i_0}}(x_1) \geq \lceil d_{H_{i_0}}(x_1)/2 \rceil + 1$. Note that $d_{H_{i_0}}(x_1)$ is odd. Let $d_{H_{i_0}}(x_1) = 2k - 1$ ($k \geq 1$). Obviously, we have $d^c_{H_{i_0-1}}(x_1) \geq d^c_{H_{i_0}}(x_1) \geq \lceil d_{H_{i_0}}(x_1)/2 \rceil + 1 = \lceil (2k-1)/2 \rceil + 1 = k + 1 = d_{H_{i_0-1}}(x_1)/2 + 1$.

We first prove the following claim in order to complete the proof of Lemma 3.2.□

Claim 2 $c(x_1x_{i_0}) = c$.

Proof Suppose, to the contrary, that $c(x_1x_{i_0}) \neq c$. Recall that $x_{i_0-1} = x_1$ and $d^c_{H_{i_0-1}}(x_1) \geq d_{H_{i_0-1}}(x_1)/2+1$. It follows that $c(x_{i_0-2}x_1) = c$ by the rules of Step 3 of Algorithm 1. Thus, we have $d^c_{H_{i_0-2}}(x_1) = d^c_{H_{i_0-1}}(x_1) + 1 \geq d_{H_{i_0-1}}(x_1)/2 + 1 + 1 = k + 1 + 1 = \lceil d_{G_{i_0-2}}(x_1)/2\rceil + 1$, contradicting the minimality of i_0. This confirms our claim. □

By Claim 2, we have $c(x_{i_0-2}x_1) \neq c$ and $c(x_1x_{i_0}) = c$. Hence, we have $d^c_{H_{i_0-2}}(x_1) = d^c_{H_{i_0}}(x_1)+1 \geq \lceil d_{H_{i_0}}(x_1)/2\rceil+1+1 = k+1+1 = \lceil d_{H_{i_0-2}}(x_1)/2\rceil+1$, contradicting the minimality of i_0. This completes the proof of Lemma 3.2. □

From Lemmas 3.1 and 3.2, we obtain the following lemma immediately.

Lemma 3.3 *For each integer i with $i \geq 2$ such that $T_i \neq T$, we have*

$$\Delta^{mon}_{H_{i-1}}(x_i) \leq \begin{cases} d_{H_{i-1}}(x_i)/2, & \text{if } x_i \neq x_1; \\ (d_{H_{i-1}}(x_i) + 1)/2, & \text{if } x_i = x_1. \end{cases}$$

Lemma 3.3 implies that for each trail $T_i \neq T$, there always exists an edge x_ix_{i+1} that is suitable for T_i in H.

Next, we show that Algorithm 1 will terminate, by proving the following lemma.

Lemma 3.4 *For an integer i with $i \geq 3$ such that $x_i \neq x_1$ and $E_{H_i}(x_i) = E_{H_i}(x_1) = \{x_ix_1\}$, we have $V(T_i) = V(H)$, as well as $c(x_{i-1}x_i) \neq c(x_ix_1)$ and $c(x_ix_1) \neq c(x_1x_2)$.*

Proof Let i be an integer with $i \geq 3$ such that $x_i \neq x_1$ and $E_{H_i}(x_i) = E_{H_i}(x_1) = \{x_ix_1\}$.

We first claim that $V(T_i) = V(H)$. Suppose, to the contrary, that there exists a vertex $v \in V(H)\backslash V(T_i)$. Obviously, we have $v \notin \{x_1, x_i\}$. Recall that K^c_n is a complete graph on n vertices, where $n \geq 3$. It follows from the construction of H that at least one of vx_i and vx_1 is an edge of H_i, contradicting the fact that $E_{H_i}(x_i) = E_{H_i}(x_1) = \{x_ix_1\}$. Thus as we claimed, we have $V(T_i) = V(H)$.

Recall that $E_{H_i}(x_i) = \{x_ix_1\}$. It follows that $\Delta^{mon}_{H_{i-1}}(x_i) = 1$ by Lemma 3.3. Therefore, we conclude that $c(x_{i-1}x_i) \neq c(x_ix_1)$.

Next, we prove the assertion that $c(x_ix_1) \neq c(x_1x_2)$. Suppose, to the contrary, that $c(x_ix_1) = c(x_1x_2) = c_0$. Let us consider T_i as the oriented trail in the direction from x_1 to x_2 (see Fig. 1). Note that $d_H(x_1) = n - 1$, if n is odd, and $d_H(x_1) = n - 2$, otherwise. We use $y_1, y_2, \ldots, y_{n-3}, y_{n-2}$ (and y_{n-1}, if n is odd) to denote the vertices of H adjacent to the vertex x_1 according to the order in which they are visited by the oriented trail T_i (see Fig. 1a, b).

We prove the following claim in order to complete the proof of Lemma 3.4.

Claim 3 There exists an integer j with $2 \leq j \leq n - 4$ ($2 \leq j \leq n - 3$, if n is odd) such that $c(\overrightarrow{y_jx_1}) \neq c_0$ and $c(\overrightarrow{x_1y_{j+1}}) \neq c_0$.

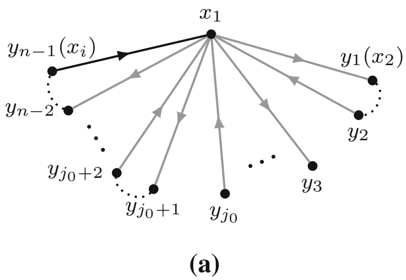

Fig. 1 **a** The oriented trail T_i for n odd; **b** the oriented trail T_i for n even

Proof Suppose, to the contrary, that at least one of $c(\overrightarrow{y_j x_1})$ and $c(\overrightarrow{x_1 y_{j+1}})$ is c_0 for every integer j with $2 \leq j \leq n-4$ ($2 \leq j \leq n-3$, if n is odd). It follows that $d_H^{c_0}(x_1) \geq (n-4)/2 + 2 > (n-2)/2$ ($d_H^{c_0}(x_1) \geq (n-3)/2 + 2 > (n-1)/2$, if n is odd), contradicting the fact that $\Delta^{mon}(K_n^c) \leq \lfloor (n-1)/2 \rfloor$. This confirms our claim. □

Let $j_0 = \max\{j \mid c(\overrightarrow{y_j x_1}) \neq c_0 \text{ and } c(\overrightarrow{x_1 y_{j+1}}) \neq c_0\}$. We suppose that $y_{j_0} = x_k$. Thus, we have $x_{k+1} = x_1$. We conclude that $d_{H_{k+1}}^{c_0}(x_1) \geq (n-3-j_0-1)/2 + 1$ and $d_{H_{k+1}}(x_1) = n-2-j_0$ ($d_{H_{k+1}}^{c_0}(x_1) \geq (n-2-j_0-1)/2+1$ and $d_{H_{k+1}}(x_1) = n-1-j_0$, if n is odd), implying that $d_{H_{k+1}}^{c_0}(x_1) \geq d_{H_{k+1}}(x_1)/2$. Recall that $c(\overrightarrow{y_{j_0} x_1}) \neq c_0$. By definition, the edge of $E_{H_{k+1}}(x_1)$ with color c_0 is suitable for T_{k+1} in H. It follows that $c(\overrightarrow{x_1 y_{j_0+1}}) = c_0$ by the rules of Step 3 of Algorithm 1. However, as supposed, we have $c(\overrightarrow{x_1 y_{j_0+1}}) \neq c_0$, a contradiction. This completes the proof of Lemma 3.4. □

Lemma 3.4 implies that Algorithm 1 will terminate in the case that $E_{H_i}(x_i) = E_{H_i}(x_1) = \{x_i x_1\}$ for some integer i with $i \geq 3$ such that $x_i \neq x_1$. However, it is possible that Algorithm 1 terminates earlier. It is not difficult to check that in all cases the output T of Algorithm 1 is a compatible spanning circuit of K_n^c.

Now, we analyze the time complexity of Algorithm 1. It is obvious from the structure of the algorithm that the combination of Step 4 and Step 3 dominates and determines its time complexity. Since each edge of H is traversed at most once, Step 3 is performed at most $|E(H)| = O(n^2)$ times according to Step 4 of Algorithm 1. In Step 3 of Algorithm 1, it requires at most $O(n^2)$ time to choose a vertex x_{i+1} such that $x_i x_{i+1}$ is suitable for T_i in H: this requires checking and comparing these colors that appear on $x_{i-1}x_i$ and the at most $O(n^2)$ edges of $E_H(x_i)$. Thus, Step 3 takes at most $O(n^2)$ time, yielding an overall time complexity $O(n^4)$. This completes the proof of Theorem 1.2. □

Proof of Theorem 1.3

We prove Theorem 1.3 by using a known algorithm due to Pevzner (1995) as a subroutine to construct an $O(n^4)$ algorithm, and proving its correctness.

Pevzner (1995) provided a polynomial algorithm for constructing a compatible Euler tour (CET for short) in an edge-colored eulerian graph G in which $\Delta^{mon}(v) \leq d(v)/2$ for each vertex v of G (see Algorithm 2 below). Benkouar et al. (1996) independently described a different algorithm for solving the same problem, requiring solving a perfect matching problem for a specific class of complete k-partite graphs.

We use Algorithm 2 as a subroutine to construct an $O(n^4)$ algorithm for finding a compatible spanning circuit visiting every vertex of K_n^c, except for one specific vertex, exactly $(n-2)/2$ times (SCSC for short), subject to the conditions that n is an even integer and $n \geq 4$, as well as the graph K_n^c satisfies $\Delta^{mon}(K_n^c) \leq (n-2)/2$ and $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$ for some vertex v_0 of K_n^c (see Algorithm 3 below).

In Algorithm 3, we denote $G' = K_n^c - v_0$, where v_0 is a specific vertex of K_n^c with $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$. It is not difficult to see that $\Delta_{G'}^{mon}(v) \leq \Delta^{mon}(K_n^c) \leq (n-2)/2 = d_{G'}(v)/2$ for each vertex v of G'. Thus, the graph G' satisfies the conditions of Algorithm 2. This implies that we can use Algorithm 2 as a subroutine in Step 2 of Algorithm 3 to construct a compatible Euler tour T' of G'.

After presenting the pseudocode of the two algorithms, we show the correctness of Algorithm 3 by stating and proving Lemma 3.5. This is followed by a short analysis of the time complexity and some concluding remarks.

Algorithm 2 (Pevzner 1995) Finding a CET in an edge-colored eulerian graph G with $\Delta^{mon}(v) \leq d(v)/2$ for each vertex v.

Input: An edge-colored eulerian graph G with $\Delta^{mon}(v) \leq d(v)/2$ for each vertex v;
Output: A CET T of G;
Step 1. $i \leftarrow 1$;
$\mathcal{C} \leftarrow \emptyset$;
Step 2. Choose an arbitrary vertex x_1 of G, and put $T_1 = x_1$;
Choose the next vertex x_2 such that $c(x_1x_2)$ is the most frequent color among the edges of $E(x_1)$, and put $T_2 = x_1x_2$;
while there exists an edge x_jx_{j+1} suitable for T_j in G **do**
Based on $T_j = x_1x_2\cdots x_j$ $(j \geq 2)$, build up $T_{j+1} = x_1x_2\cdots x_jx_{j+1}$ by choosing a vertex x_{j+1} such that x_jx_{j+1} is suitable for T_j in G, and preferentially choosing the vertex x_{j+1} such that $c(x_jx_{j+1}) = c(x_1x_2)$ when $x_j = x_1$;
$j \leftarrow j+1$;
end while
$C_i \leftarrow T_j$ (note that T_j is a closed trail);
$\mathcal{C} \leftarrow \mathcal{C} \cup \{C_i\}$;
$G \leftarrow G - E(C_i)$;
Step 3. **if** $E(G) \neq \emptyset$ **then**
$i \leftarrow i+1$, and go to **Step 2**;
end if
Step 4. To construct T, if $\mathcal{C} \setminus \{C_1\} \neq \emptyset$, then start walking along C_1, **until** an intersection vertex with another closed trail C_p of $\mathcal{C}$ is found;

Continue walking along C_p while preserving the compatibility on the intersection vertex in the case of walking into C_p and walking out of C_p, **until** C_p is entirely walked out;

Then continue walking along the remaining part of C_1, **until** C_1 is entirely walked out;

We use C_1 to denote the new closed trail that is the combination of C_1 and C_p, and continue to combine the remaining elements of $\mathcal{C}$, if any, in this way, **until** all elements of $\mathcal{C}$ have been combined into the closed trail T;
return T.

Algorithm 3 Finding a SCSC in K_n^c (even $n \geq 4$) with $\Delta^{mon}(K_n^c) \leq (n-2)/2$ and $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$ for some vertex v_0.

Input: An edge-colored graph K_n^c (even $n \geq 4$) with $\Delta^{mon}(K_n^c) \leq (n-2)/2$ and $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$ for some vertex v_0;
Output: A compatible spanning circuit T of K_n^c visiting every vertex of K_n^c, except for one specific vertex, exactly $(n-2)/2$ times;
Step 1. Choose a specific vertex v_0 of K_n^c with $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$, and let $G' = K_n^c - v_0$;
Step 2. Perform Algorithm 2 on G' to produce a compatible Euler tour of G', denoted by $T' = x_1x_2\cdots x_1$;
Step 3. Choose an edge x_ix_{i+1} of T' such that $c(x_{i-1}x_i) \neq c(x_iv_0)$ and $c(x_iv_0) \neq c(v_0x_{i+1})$, as well as $c(v_0x_{i+1}) \neq c(x_{i+1}x_{i+2})$, where the subscripts are taken modulo $\binom{n-1}{2}$;
Step 4. Let $T = T' \cup \{x_iv_0, v_0x_{i+1}\} \setminus \{x_ix_{i+1}\}$;
return T.

The correctness of Algorithm 3 follows directly from the following lemma (and the correctness of Algorithm 2 due to Pevzner (1995)).

Lemma 3.5 *There exists an edge x_ix_{i+1} of T' such that $c(x_{i-1}x_i) \neq c(x_iv_0)$ and $c(x_iv_0) \neq c(v_0x_{i+1})$, as well as $c(v_0x_{i+1}) \neq c(x_{i+1}x_{i+2})$, where the subscripts are taken modulo $\binom{n-1}{2}$.*

Proof Suppose, to the contrary, that for each integer i such that $c(x_iv_0) \neq c(v_0x_{i+1})$, either $c(x_{i-1}x_i) = c(x_iv_0)$, or $c(v_0x_{i+1}) = c(x_{i+1}x_{i+2})$.

Let $cd(v_0) = n-\ell \geq n-\lfloor(\sqrt{4n-3}+1)/2\rfloor$. Thus, we have $\ell \leq (\sqrt{4n-3}+1)/2$. Let $P = \{\{v_0x_i,\ v_0x_{i+1}\} \subset E_{K_n^c}(v_0) \mid c(v_0x_i) \neq c(v_0x_{i+1})\}$. We can conclude that $|P| \geq \binom{n-1}{2} - \binom{\ell}{2} = ((n-1)(n-2))/2 - (\ell(\ell-1))/2 = (n^2-3n+2-\ell^2+\ell)/2$.

As supposed, we have either $c(x_{i-1}x_i) = c(x_iv_0)$, or $c(v_0x_{i+1}) = c(x_{i+1}x_{i+2})$ for each pair $\{v_0x_i,\ v_0x_{i+1}\}$ of P. Note that the graph G' is a complete graph on $n-1$ vertices. It follows from $\ell \leq (\sqrt{4n-3}+1)/2$ that $\frac{|P|}{n-1} \geq \frac{n^2-3n+2-\ell^2+\ell}{2(n-1)} \geq \frac{n-3}{2} > \frac{n-4}{2}$. Therefore, there exists a vertex v of G' such that $d_{K_n^c}^{c_0}(v) \geq (n-4)/2+1+1 > (n-2)/2$, where $c_0 = c(v_0v)$, contradicting that $\Delta^{mon}(K_n^c) \leq (n-2)/2$. This completes the proof of Lemma 3.5. □

Lemma 3.5 clearly shows that we can always find an edge satisfying the requested conditions at Step 3 of Algorithm 3.

It is not difficult to check that the closed trail returned by Algorithm 3 is a desired compatible spanning circuit of K_n^c.

In order to analyze the time complexity of Algorithm 3, we first need to analyze the time complexity of Algorithm 2 due to Pevzner, since we use it as a subroutine. In fact, it is clear that Algorithm 2 is the dominating factor regarding the time complexity of Algorithm 3. Due to the similarity with Algorithm 1, it is not difficult to see that Algorithm 2 has time complexity $O(n^4)$ (in case the graph G is a complete graph on n vertices). Therefore, the whole time complexity of Algorithm 3 is $O(n^4)$. This completes the proof of Theorem 1.3. □

4 Conclusions and final remarks

In this work, we considered the existence of more general compatible spanning circuits in edge-colored graphs from an algorithmic perspective. We first proved that the decision problem of determining whether an edge-colored connected graph contains a compatible spanning circuit is NP-complete, even within graphs with maximum degree 3. We then developed two polynomial-time algorithms for finding compatible spanning circuits (with certain properties) in specific edge-colored complete graphs. In particular, our Algorithm 1 returns a compatible spanning circuit (with no restrictions) directly. In previous work from literature, this was done in two steps. We also presented Algorithm 3 for finding compatible spanning circuits visiting every vertex, except for one specific vertex, exactly $(n-2)/2$ times in edge-colored complete graphs G on even n $(n \geq 4)$ vertices with $\Delta^{mon}(G) \leq (n-2)/2$ and $cd(v_0) \geq n - \lfloor(\sqrt{4n-3}+1)/2\rfloor$ for some vertex v_0 of G.

In future work, we look forward to establishing polynomial-time algorithms for finding compatible spanning circuits in other classes of edge-colored graphs. As another future direction, a more challenging problem is to develop polynomial-time algorithms for finding compatible spanning circuits visiting every vertex exactly (or at least) a specified number of times in some specific classes of edge-colored graphs.

Acknowledgements We thank the anonymous referees for their careful reading, and for their useful comments on an earlier version that improved the presentation.

References

Ahuja SK (2010) Algorithms for routing and channel assignment in wireless infrastructure networks, Ph.D. thesis, Univ. Arizona

Alon N, Gutin G (1997) Properly colored Hamilton cycles in edge-colored complete graphs. Random Struct Algorithm 11:179–186

Benkouar A, Manoussakis Y, Paschos VT, Saad R (1996) Hamiltonian problems in edge-colored complete graphs and Eulerian cycles in edge-colored graphs: some complexity results. RAIRO Oper Res 30:417–438

Bollobás B, Erdős P (1976) Alternating hamiltonian cycles. Isr J Math 23:126–131

Bondy JA, Murty USR (2008) Graph theory, vol 244. Graduate texts in mathematics. Springer, New York

Chen CC, Daykin DE (1976) Graphs with hamiltonian cycles having adjacent lines different colors. J Combin Theory Ser B 21:135–139

Daykin DE (1976) Graphs with cycles having adjacent lines of different colors. J Combin Theory Ser B 20:149–152

Fleischner H, Fulmek M (1990) P(D)-compatible eulerian trails in digraphs and a new splitting lemma. In: Contemporary methods in graph theory. Mannheim, Germany, Bibliographisches Institut, pp 291–303

Fujita S, Magnant C (2011) Properly colored paths and cycles. Discrete Appl Math 159:1391–1397

Garey MR, Johnson DS, Stockmeyer L (1974) Some simplified NP-complete problems. In: Proceedings of sixth annual ACM symposium on theory of computing. ACS, Washington, DC, pp 47–63
Guo Z, Broersma HJ, Li, B, Zhang S (2020a) Almost eulerian compatible spanning circuits in edge-colored graphs (submitted)
Guo Z, Li B, Li X, Zhang S (2020b) Compatible spanning circuits in edge-colored graphs. Discrete Math 343:111908
Kotzig A (1968) Moves without forbidden transitions in a graph. Mat Časopis Sloven Akad Vied 18:76–80
Lo A (2016) Properly coloured hamiltonian cycles in edge-coloured complete graphs. Combinatorica 36:471–492
Pevzner PA (1995) DNA physical mapping and alternating eulerian cycles in colored graphs. Algorithmica 13:77–105
Pevzner PA (2000) Computational molecular biology: an algorithmic approach. MIT Press, Cambridge
Sankararaman S, Efrat A, Ramasubramanian S (2014) On channel discontinuity-constraint routing in wireless networks. Ad Hoc Netw 13:153–169
Shearer J (1979) A property of the colored complete graph. Discrete Math 25:175–178
Szachniuk M, De Cola MC, Felici G, Blazewicz J (2014) The orderly colored longest path problem—a survey of applications and new algorithms. RAIRO-Oper Res 48:25–51
Szachniuk M, Popenda M, Adamiak RW, Blazewicz J (2009) An assignment walk through 3D NMR spectrum. In: Proceedings of 2009 IEEE symposium on computational intelligence in bioinformatics and computational biology, pp 215–219
Tseng IL, Chen HW, Lee CI (2010) Obstacle-aware longest path routing with parallel MILP solvers. Proc WCECS-ICCS 2:827–831