Algorithmica (2020) 82:1654–1674
https://doi.org/10.1007/s00453-019-00663-9

An Optimal XP Algorithm for Hamiltonian Cycle on Graphs of Bounded Clique-Width

Benjamin Bergougnoux[1] · Mamadou Moustapha Kanté[2] · O-joung Kwon[3,4]

Received: 28 September 2018 / Accepted: 9 December 2019 / Published online: 19 December 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

In this paper, we prove that, given a clique-width k-expression of an n-vertex graph, Hamiltonian Cycle can be solved in time $n^{\mathcal{O}(k)}$. This improves the naive algorithm that runs in time $n^{\mathcal{O}(k^2)}$ by Espelage et al. (Graph-theoretic concepts in computer science, vol 2204. Springer, Berlin, 2001), and it also matches with the lower bound result by Fomin et al. that, unless the Exponential Time Hypothesis fails, there is no algorithm running in time $n^{o(k)}$ (Fomin et al. in SIAM J Comput 43:1541–1563, 2014). We present a technique of representative sets using two-edge colored multigraphs on k vertices. The essential idea is that, for a two-edge colored multigraph, the existence of an Eulerian trail that uses edges with different colors alternately can be determined by two information: the number of colored edges incident with each vertex, and the connectedness of the multigraph. With this idea, we avoid the bottleneck of the naive algorithm, which stores all the possible multigraphs on k vertices with at most n edges.

An extended abstract appeared in Algorithms and Data Structures, WADS 2017 [3]. B. Bergougnoux and M.M. Kanté are supported by French Agency for Research under the GraphEN Project (ANR-15-CE40-0009). O. Kwon is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC consolidator grant DISTRUCT, Agreement No. 648527), also supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. NRF-2018R1D1A1B07050294), and also supported by Institute for Basic Sciences (IBS-R029-C1).

✉ O-joung Kwon
ojoungkwon@gmail.com

Benjamin Bergougnoux
benjamin.bergougnoux@uib.no

Mamadou Moustapha Kanté
mamadou.kante@uca.fr

1 University of Bergen, Bergen, Norway

2 LIMOS, CNRS, Université Clermont Auvergne, Aubière, France

3 Department of Mathematics, Incheon National University, Incheon, South Korea

4 Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, South Korea

Keywords Hamiltonian cycle · Eulerian trail · Clique-width · XP algorithm

1 Introduction

Tree-width is one of the most well-studied graph parameters in the graph algorithm community, due to its numerous structural and algorithmic properties (see the survey [4]). A celebrated algorithmic meta-theorem by Courcelle [8] states that every graph problem expressible in monadic second-order logic (MSO_2) can be decided in linear time on graphs of bounded tree-width. Among the problems expressible in MSO_2, there are some well-known NP-hard problems such as MINIMUM DOMINATING SET, HAMILTONIAN CYCLE, and GRAPH COLORING.

Despite the broad interest on tree-width, only sparse graphs can have bounded tree-width. But, on many dense graph classes, some NP-hard problems admit polynomial-time algorithms, and many of these algorithms can be explained by the boundedness of their *clique-width*. Clique-width is a graph parameter that originally emerges from the theory of graph grammars [10] and the terminology was first introduced by Courcelle and Olariu [13] (see also the book [9]).

Clique-width is defined in terms of the following graph operations: (1) addition of a single labeled vertex, (2) addition of all possible edges between vertices labeled i and those labeled j, (3) renaming of labels, and (4) taking the disjoint union of two labeled graphs. The clique-width of a graph is the minimum number of labels needed to construct it. An expression constructing a graph with at most k labels is called a *(clique-width) k-expression*. The modeling power of clique-width is strictly stronger than the modeling power of tree-width. In other words, if a graph class has bounded tree-width, then it has bounded clique-width, but the converse is not true.

Computing the clique-width of a graph is a problem which has received significant attention over the last decade. Fellows et al. [17] showed that computing clique-width is NP-hard. For a fixed k, the best known approximation algorithm is due to Hliněný and Oum [24]; it computes in time $O(f(k)\cdot n^3)$ a $(2^{k+1}-1)$-expression for an n-vertex graph of clique-width at most k.

Courcelle et al. [11] extended the meta-theorem of Courcelle [8] to graphs of bounded clique-width at a cost of a smaller set of problems. More precisely, they showed that every problem expressible in monadic second order logic with formula that does not use edge set quantifications (called MSO_1) can be decided in time $O(f(k)\cdot n)$ in any n-vertex graph of clique-width k, provided that a clique-width k-expression of it is given.

For some MSO_1 problems, clique-width and tree-width have sensibly the same behavior. Indeed, many problems expressible in MSO_1 that admit $2^{O(k)}\cdot n^{O(1)}$-time algorithms parameterized by tree-width have been shown to admit $2^{O(k)}\cdot n^{O(1)}$-time algorithms, when a clique-width k-expression is given. These include famous problems like MINIMUM DOMINATING SET and MINIMUM VERTEX COVER [7,23,28], or even their connected variants and FEEDBACK VERTEX SET [2,5,14].

On the other hand, several classical problems, such as MAX- CUT, EDGE DOMINATING SET (EDS), GRAPH COLORING (GC), and HAMILTONIAN CYCLE (HC), are not expressible in MSO_1. These problems are known to be FPT parameterized

by tree-width thanks to Courcelle's theorem or some variants of this latter theorem [1,6,12]. They are also known to admit XP algorithms parameterized by clique-width [16,20,25].

A natural question is whether these problems admit algorithms with running time $f(k) \cdot n^{O(1)}$ given a k-expression of the input graph. Fomin, Golovach et al. [19] proved the W[1]-hardness of EDS, GC, and HC with clique-width as parameter, which implies that these problems do not admit algorithms with running time $f(k) \cdot n^{O(1)}$, for any function f, unless W[1] = FPT. In 2014, the same authors [20] have proved that MAX-CUT and EDS admit $n^{O(k)}$-time algorithms, and that they do not admit $f(k) \cdot n^{o(k)}$-time algorithms unless ETH fails. In the conclusion of [20], the authors state that HC does not admit $f(k) \cdot n^{o(k)}$-time algorithm under ETH (they gave the proof in [21]) and they left an open question of finding an algorithm with running time $f(k) \cdot n^{O(k)}$. At the time, the best known running time parameterized by clique-width for HC and GC were respectively $n^{O(k^2)}$ [16] and $n^{O(2^k)}$ [25]. Recently, Fomin et al. [21] provided a lower bound of $f(k) \cdot n^{2^{o(k)}}$ for GC.

Our Contribution and Approach In this paper, we prove that there exists an algorithm solving HAMILTONIAN CYCLE in time $n^{O(k)}$, when a clique-width k-expression is given. A *Hamiltonian cycle* of a graph G is a cycle containing all the vertices of G. The problem HAMILTONIAN CYCLE asks, given a graph G, if G contains a Hamiltonian cycle. Specifically, we prove the following.

Theorem 1 *There exists an algorithm that, given an n-vertex graph G and a k-expression of G, solves* HAMILTONIAN CYCLE *in time* $O(n^{2k+5} \cdot 2^{2k(\log_2(k)+1)} \cdot k^3 \log_2(nk))$.

Our algorithm is a dynamic programming one whose steps are based on the k-labeled graphs H arising in the k-expression of G. Observe that the edges of a Hamiltonian cycle of G which belong to $E(H)$ induce either a Hamiltonian cycle or a collection of vertex-disjoint paths in G covering H. Consequently, we define a partial solution as a set of edges of H which induces a collection of paths (potentially empty) covering H. As in [16], with each partial solution $\mathcal{P}$, we associate an auxiliary multigraph such that its vertices correspond to the labels of H and each edge $\{i, j\}$ corresponds to a maximal path induced by $\mathcal{P}$ with end-vertices labeled i and j.

Since H is a k-labeled graph arising in a k-expression of G, we have that two vertices x and y with the same label in H have the same neighborhood in $G - E(H)$ (the graph G without the edges of H). It follows that the endpoints of a path in a partial solution are not important and what matters are the labels of these endpoints. As a result, two partial solutions with the same auxiliary multigraph are equivalent, i.e., if one is contained in a Hamiltonian cycle, then the other also. From these observations, one easily deduces the $n^{O(k^2)}$-time algorithm, due to Espelage et al. [16], because there are at most n possible paths between two label classes and thus there are at most $n^{\mathcal{O}(k^2)}$ possible auxiliary graphs.

To obtain our $n^{O(k)}$-time algorithm, we refine the above equivalence relation. We define that two partial solutions are equivalent if their auxiliary graphs have the connected components on same vertex sets, and the paths they induce have the same

number of endpoints labeled i, for each label i. The motivation of this equivalence relation can be described as follows. Suppose that we have a partial solution $\mathcal{P}$ and a set of edges $\mathcal{Q} \subseteq E(G) \setminus E(H)$ so that $\mathcal{P} \cup \mathcal{Q}$ forms a Hamiltonian cycle, and we consider to make an auxiliary graph of $\mathcal{Q}$, and identify with the one for $\mathcal{P}$. To distinguish edges obtained from $\mathcal{P}$ or $\mathcal{Q}$, we color edges by red if one came from $\mathcal{P}$ and by blue otherwise. Then following the Hamiltonian cycle, we can find an Eulerian trail of this merged auxiliary graph that uses edges of distinct colors alternately. But then if $\mathcal{P}'$ is equivalent to $\mathcal{P}$, then one can observe that if we replace the red part with the auxiliary graph of $\mathcal{P}'$, then it also has such an Eulerian trail, and we can show that $\mathcal{P}'$ can also be completed to a Hamiltonian cycle. So, in the algorithm, for each equivalence class, we store one partial solution. We define this equivalence relation formally in Sect. 3.

Since, the number of partitions of a k-size set is at most k^k and the number of paths induced by a partial solution is always bounded by n, the number of non-equivalent partial solutions is then bounded by $(2n)^k \cdot k^k$ (the maximum degree of an auxiliary multigraph is at most $2n$ because a loop is counted as two edges). The running time of our algorithm follows from the maximum number of non-equivalent partial solutions. The main effort in the algorithm consists then in updating the equivalence classes of this equivalence relation in terms of operations based on the clique-width operations.

Overview In Sect. 2, we give basic definitions and notations concerning (multi)graphs and clique-width. Our notions of partial solutions and of auxiliary multigraphs are given in Sect. 3. In Sect. 4, we prove the equivalence between the existence of Hamiltonian cycles in the input graph and the existence of red–blue Eulerian trails in auxiliary multigraphs, and deduce that it is enough to store $(2n)^k \cdot k^k$ partial solutions at each step of our dynamic programming algorithm. In Sect. 5, we show how to obtain from the results of Sect. 4 an $n^{O(k)}$-time algorithm for Hamiltonian Cycle. In Sect. 6, we give some intuitions for solving in time $n^{O(k)}$ the problems Directed Hamiltonian Cycle given a k-expression. We end up with some concluding remarks and open questions in Sect. 7.

2 Preliminaries

The size of a set V is denoted by $|V|$, and we write $[V]^2$ to denote the set of all subsets of V of size 2. We denote by $\mathbb{N}$ the set of non-negative integers. For two sets A and B, we let

$$A \otimes B := \begin{cases} \emptyset & \text{if } A = \emptyset \text{ or } B = \emptyset, \\ \{X \cup Y \mid X \in A \wedge Y \in B\} & \text{otherwise.} \end{cases}$$

For a positive integer n, let $[n] := \{1, 2, \ldots, n\}$.

Graph We essentially follow Diestel's book [15] for our graph terminology, but we deal only with finite graphs. We distinguish graphs and multigraphs, and for graphs we do not allow to have multiple edges or loops, while we allow them in multigraphs.

The vertex set of a graph G is denoted by $V(G)$ and its edge set is denoted by $E(G) \subseteq [V(G)]^2$. We write xy to denote an edge $\{x, y\}$.

Let G be a graph. For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced by X. For $F \subseteq E(G)$, we write $G_{|F}$ for the subgraph $(V(G), F)$ and $G - F$ for the subgraph $(V(G), E(G) \backslash F)$. For an edge e of G, we simply write $G - e$ for $G - \{e\}$. The *degree* of a vertex x, denoted by $\mathsf{deg}_G(x)$, is the number of edges incident with x. The set of vertices adjacent to a vertex v is denoted by $N_G(x)$.

Multigraph A *multigraph* is essentially a graph, but we allow multiple edges, i.e., edges incident with the same set of vertices. Formally, a *multigraph* G is a pair $(V(G), E(G))$ of disjoint sets, also called sets of vertices and edges, respectively, together with a map $\mathsf{mult}_G : E(G) \to V(G) \cup [V(G)]^2$, which maps every edge to one or two vertices, still called its endpoints. The degree of a vertex x in a multigraph G, is defined analogously as in graphs, except that each loop is counted twice, and similarly for other notions. If there are exactly k edges e such that $\mathsf{mult}_G(e) = \{x, y\}$ (or $\mathsf{mult}_G(e) = \{x\}$), then we denote these distinct edges by $\{x, y\}_1, \ldots, \{x, y\}_k$ (or $\{x\}_1, \ldots, \{x\}_k$).

For two multigraphs G and H on the same vertex set $\{v_1, \ldots, v_k\}$ and with disjoint edge sets, we denote by $G \uplus H$ the multigraph with vertex set $\{v_1, \ldots, v_k\}$, edge set $E(G) \cup E(H)$, and

$$\mathsf{mult}_{G \uplus H}(e) := \begin{cases} \mathsf{mult}_G(e) & \text{if } e \in E(G), \\ \mathsf{mult}_H(e) & \text{otherwise.} \end{cases}$$

If the edges of G and H are colored, then this operation preserves the colors of the edges.

Walk A *walk* of a graph is a sequence of vertices and edges, starting and ending at some vertices, called *end-vertices*, where for every consecutive pair of a vertex x and an edge e, x is incident with e. The vertices of a walk which are not end-vertices are called *internal* vertices. A *trail* of a graph is a walk where each edge is used at most once. A walk (or a trail) is *closed* if its two end-vertices are the same. Moreover, when the edges of a graph are colored red or blue, we say that a walk $W = (v_1, e_1, \ldots, v_{r-1}, e_{r-1}, v_r)$ is a *red–blue walk*, if, for every $i \in \{1, \ldots, r-2\}$, the colors of e_i and e_{i+1} are different and the colors of e_1 and e_{r-1} are different, when the walk is closed. We adapt all the notations to multigraphs as well.

For two walks $W_1 = (v_1, e_1, \ldots, e_{\ell-1}, v_\ell)$ and $W_2 = (v'_1, e'_1, \ldots, e'_{r-1}, v'_r)$ such that $v_\ell = v'_1$, the *concatenation* of W_1 and W_2, denoted by $W_1 - W_2$, is the walk $(v_1, e_1, \ldots, e_{\ell-1}, v_\ell, e'_1, \ldots, e'_{r-1}, v'_r)$.

A *path* of a graph is a walk where each vertex is used at most once. A *cycle* of a graph is a closed walk where each vertex other than the end-vertices is used at most once. An *(closed) Eulerian trail* in a graph G is a closed trail containing all the edges of G. In particular, if the edges of a graph are colored red or blue, then a red–blue Eulerian trail is an Eulerian trail that is a red–blue walk.

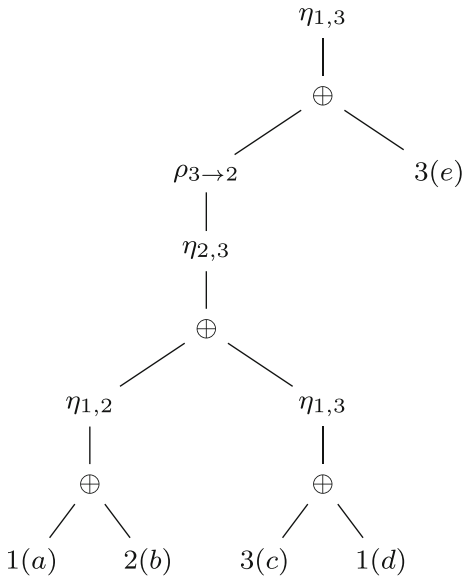

Fig. 1 An irredundant 3-expression of C_5

Clique-Width A *k-labeled graph* is a pair (G, lab_G) of a graph G and a function lab_G from $V(G)$ to $[k]$, called the *labeling function*. We denote by $\mathsf{lab}_G^{-1}(i)$ the set of vertices in G with label i. The notion of clique-width is defined by Courcelle and Olariu [13] and is based on the following operations:

- creating a graph, denoted by $i(x)$, with a single vertex x labeled with $i \in [k]$;
- for a labeled graph G and distinct labels $i, j \in [k]$, relabeling the vertices of G with label i to j, denoted by $\rho_{i\to j}(G)$;
- for a labeled graph G and distinct labels $i, j \in [k]$, adding all the non-existent edges between vertices with label i and vertices with label j, denoted by $\eta_{i,j}(G)$;
- taking the disjoint union of two labeled graphs G and H, denoted by $G \oplus H$, with

$$\mathsf{lab}_{G\oplus H}(v) := \begin{cases} \mathsf{lab}_G(v) & \text{if } x \in V(G), \\ \mathsf{lab}_H(v) & \text{otherwise.} \end{cases}$$

A *clique-width k-expression*, or shortly a *k-expression*, is a finite well-formed term built with the four operations above and using at most k labels. Each k-expression ϕ evaluates into a k-labeled graph $(\mathsf{val}(\phi), \mathsf{lab}_{\mathsf{val}(\phi)})$. The *clique-width* of a graph G, denoted by $\mathsf{cw}(G)$, is the minimum k such that G is isomorphic to $\mathsf{val}(\phi)$ for some k-expression ϕ. We can assume without loss of generality that any k-expression defining a graph G uses $O(n)$ disjoint union operations and $O(nk^2)$ unary operations [13].

It is worth noticing, from the recursive definition of k-expressions, that one can compute in time linear in $|\phi|$ the labeling function $\mathsf{lab}_{\mathsf{val}(\phi)}$ of $\mathsf{val}(\phi)$, and hence we will always assume that it is given.

For example, the cycle $abcdea$ of length 5 can be constructed using the 3-expression represented as a tree-structure in Fig. 1.

The set of subexpressions of a k-expression ϕ, denoted by $\mathsf{Sub}(\phi)$, is defined by the following induction:

$$\mathsf{Sub}(\phi) := \begin{cases} \{\phi\} \text{ if } \phi := i(x) \text{ with } i \in [k], \\ \{\phi\} \cup \mathsf{Sub}(\phi') \cup \mathsf{Sub}(\phi^\star) \text{ if } \phi = \phi' \oplus \phi^\star, \\ \{\phi\} \cup \mathsf{Sub}(\phi') \text{ if } \phi = f(\phi') \text{ with } f \in \{\rho_{i\to j}, \eta_{i,j} \mid i, j \in [k]\}. \end{cases}$$

We say that a k-labeled graph (H, lab_H) arises in a k-expression ϕ if $H = \mathsf{val}(\phi')$ and $\mathsf{lab}_H = \mathsf{lab}_{\mathsf{val}(\phi')}$, for some $\phi' \in \mathsf{Sub}(\phi)$.

A k-expression ϕ is called *irredundant* if, for every subexpression $\eta_{i,j}(\phi')$ of ϕ, there are no edges in $\mathsf{val}(\phi')$ between the vertices labeled i and the vertices labeled j. Courcelle and Olariu [13] proved that given a clique-width k-expression, it can be transformed into an irredundant k-expression in linear time. The following useful properties of an irredundant k-expression will be used in Sect. 4.

Lemma 1 *Let H be a k-labeled graph arising in an irredundant k-expression ϕ of a graph G. For all $u, v \in V(H)$ with $\mathsf{lab}_H(u) = i$ and $\mathsf{lab}_H(v) = j$, we have the following.*

1. *If $i = j$, then $N_G(u)\backslash V(H) = N_G(v)\backslash V(H)$.*
2. *If $uv \in E(G)\backslash E(H)$, then $i \neq j$ and, for all $x, y \in V(H)$ with $\mathsf{lab}_H(x) = i$ and $\mathsf{lab}_H(y) = j$, we have $xy \in E(G)\backslash E(H)$.*

Proof (1) Assume that $i = j$. Let ϕ' be the subexpression of ϕ such that $H = \mathsf{val}(\phi')$. As u and v have the same label in H, in every subexpression of ϕ having ϕ' as a subexpression, u and v have the same label. Since edges are added only through the operation $\eta_{a,b}$, we conclude that $N_G(u) \cap (V(G)\backslash V(H)) = N_G(v) \cap (V(G)\backslash V(H))$.

(2) Assume that $uv \in E(G)\backslash E(H)$. Then, we have $i \neq j$ because the operation $\eta_{a,b}$ adds edges only between vertices with distinct labels. Let ϕ' be the minimal (size wise) subexpression of ϕ such that $uv \in E(\mathsf{val}(\phi'))$. It follows that $\phi' = \eta_{a,b}(\phi^\star)$, with $\phi^\star \in \mathsf{Sub}(\phi)$, $\mathsf{lab}_{\mathsf{val}(\phi')}(u) = a$ and $\mathsf{lab}_{\mathsf{val}(\phi')}(v) = b$. Let $D := \mathsf{val}(\phi^\star)$. Observe that we have $V(H) \subseteq V(D)$ and $E(H) \subseteq E(D)$. Moreover, all vertices labeled i in H are labeled a in D and those labeled j in H are labeled b in D. Since ϕ is irredundant, there are no edges in D between a vertex labeled a and one labeled b. Thus, for all vertices $x \in \mathsf{lab}_H^{-1}(i)$ and $y \in \mathsf{lab}_H^{-1}(j)$, we have $xy \notin E(H)$ and $xy \in E(G)$.

□

3 Partial Solutions and Auxiliary Graphs

Let G be a graph and (H, lab_H) be a k-labeled graph such that H is a subgraph of G.

A *partial solution* of H is a set of edges $\mathcal{P} \subseteq E(H)$ such that $H_{|\mathcal{P}}$ is a union of vertex-disjoint paths, i.e., $H_{|\mathcal{P}}$ is acyclic and, for every vertex $v \in V(H)$, the degree of v in $H_{|\mathcal{P}}$ is at most two. We denote by $\Pi(H)$ the set of all partial solutions of H. We say that a path P in $H_{|\mathcal{P}}$ is *maximal* if the degree of its end-vertices in $H_{|\mathcal{P}}$ is

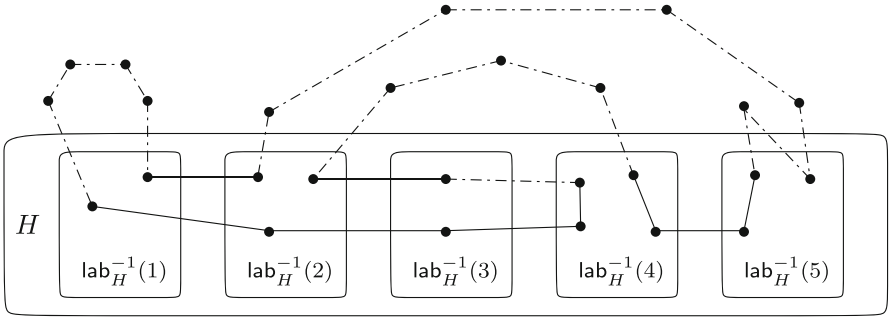

Fig. 2 Examples of a partial solution $\mathcal{P}$ (solid lines) and a complement solution $\mathcal{Q}$ (dashed lines) forming a Hamiltonian cycle. Observe that $H_{|\mathcal{P}}$ contains 5 maximal paths and $G_{|\mathcal{Q}}$ contains 5 H-paths (and only 4 maximal paths)

at most one; in other words, there is no path P' in $H_{|\mathcal{P}}$ such that $V(P) \subsetneq V(P')$. Observe that an isolated vertex in $H_{|\mathcal{P}}$ is considered as a maximal path.

A *complement solution* of H is a subset $\mathcal{Q}$ of $E(G)\backslash E(H)$ such that $G_{|\mathcal{Q}}$ is a union of vertex-disjoint paths with end-vertices in $V(H)$; in particular, for every vertex v in $V(G)\backslash V(H)$, the degree of v in $G_{|\mathcal{Q}}$ is two. We denote by $\overline{\Pi}(H)$ the set of all complement solutions of H. A path P in G with at least 2 vertices is an *H-path* if the end-vertices of P are in $V(H)$ and the internal vertices of P are in $V(G)\backslash V(H)$. By definition, isolated vertices in $V(H)$ are not H-paths. Observe that, for a complement solution $\mathcal{Q}$, we can decompose each maximal path Q of $G_{|\mathcal{Q}}$ with at least 2 vertices into H-paths (not necessarily one).

Examples of a partial solution and a complement solution are given in Fig. 2. Note that if G has a Hamiltonian cycle C and $E(C) \nsubseteq E(H)$, then $E(C) \cap E(H)$ is a partial solution and $E(C) \cap (E(G)\backslash E(H))$ is a complement solution. We say that a partial solution $\mathcal{P}$ and a complement solution $\mathcal{Q}$ form a Hamiltonian cycle if $(V(G), \mathcal{P} \cup \mathcal{Q})$ is a cycle containing all the vertices of G.

Auxiliary Multigraph For $\mathcal{P} \in \Pi(H) \cup \overline{\Pi}(H)$ and $i, j \in [k]$, we define ℓ_{ij} and ℓ_i as follows.

- If $\mathcal{P}$ is a partial solution of H, then ℓ_{ij} is the number of maximal paths in $H_{|\mathcal{P}}$ with end-vertices labeled i and j, and ℓ_i is the number of maximal paths in $H_{|\mathcal{P}}$ with both end-vertices labeled i.
- If $\mathcal{P}$ is a complement solution of H, then ℓ_{ij} is the number of H-paths in $G_{|\mathcal{P}}$ with end-vertices labeled i and j, and ℓ_i is the number of H-paths in $G_{|\mathcal{P}}$ with both end-vertices labeled i.

Now, we define the auxiliary multigraph of $\mathcal{P}$, denoted by $\mathsf{aux}_H(\mathcal{P})$, as the multigraph with vertex set $\{v_1, \ldots, v_k\}$ and edge set

$$\bigcup_{\substack{i,j\in[k]\\ i\neq j}} \{\{v_i, v_j\}_1, \ldots, \{v_i, v_j\}_{\ell_{ij}}\} \cup \bigcup_{i\in[k]} \{\{v_i\}_1, \ldots, \{v_i\}_{\ell_i}\}.$$

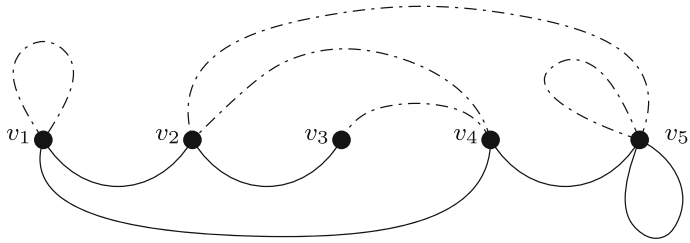

Fig. 3 The union $G_1 \uplus G_2$ of auxiliary multigraphs G_1 and G_2 associated with the partial solution (solid lines) and the complement solution (dashed lines) represented in Fig. 2

Moreover, if $\mathcal{P}$ is a partial solution of H, then we color all the edges of $\mathsf{aux}_H(\mathcal{P})$ in red, and if $\mathcal{P}$ is a complement solution, then we color the edges of $\mathsf{aux}_H(\mathcal{P})$ in blue. An example of an auxiliary multigraph is given in Fig. 3.

4 Relations between Hamiltonian Cycles and Eulerian Trails

Let G be an n-vertex graph and ϕ be an irredundant k-expression of G. Let H be a k-labeled graph arising in the k-expression ϕ. Observe that H is a subgraph of G (disregarding the labels). This section is dedicated to prove the properties of the following relation between partial solutions of H based on the degree sequence and the connected components of their auxiliary multigraphs.

Definition 1 Let $\mathcal{P}_1, \mathcal{P}_2 \in \Pi(H)$. We write $\mathcal{P}_1 \simeq \mathcal{P}_2$ if $\mathsf{aux}_H(\mathcal{P}_1)$ and $\mathsf{aux}_H(\mathcal{P}_2)$ have the same set of connected components and for each vertex v in $\{v_1, \ldots, v_k\}$, $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P}_1)}(v) = \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P}_2)}(v)$.

Observe that $\simeq$ is an equivalence relation. For a set $\mathcal{A} \subseteq \Pi(H)$, we define $\mathsf{reduce}_H(\mathcal{A})$ as the operation which returns a set containing one element of each equivalence class of $\mathcal{A}/\simeq$.

The main idea of our algorithm is to call reduce_H at each step of our dynamic programming algorithm in order to keep the size of a set of partial solutions manipulated small, i.e., bounded by $n^{O(k)}$. The running time of our algorithm follows mostly from the following lemma.

Lemma 2 *For every $\mathcal{A} \subseteq \Pi(H)$, we have $|\mathsf{reduce}_H(\mathcal{A})| \leq n^k \cdot 2^{k(\log_2(k)+1)}$ and we can moreover compute $\mathsf{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk^2 \log_2(nk))$.*

Proof To prove that $\mathsf{reduce}_H(\mathcal{A}) \leq n^k \cdot 2^{k(\log_2(k)+1)}$, it is enough to bound the number of equivalence classes of $\Pi(H)/\simeq$.

We claim that, for every $\mathcal{P} \in \Pi(H)$, we have $\sum_{i\in[k]} \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) \leq 2|V(H)|$. First observe that $\sum_{i\in[k]} \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) = 2|V(H)|$ when $\mathcal{P} = \emptyset$, since each isolated vertex in $H_{|\mathcal{P}}$ gives a loop in $\mathsf{aux}_H(\mathcal{P})$. Moreover, when $\mathcal{P}$ contains an edge, removing an edge from a partial solution $\mathcal{P}$ of H increases $\sum_{i\in[k]} \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i)$ by two; indeed, this edge removal splits a maximal path of $H_{|\mathcal{P}}$ into two maximal paths. Therefore, any partial solution $\mathcal{P}$ satisfies that $\sum_{i\in[k]} \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) \leq 2|V(H)|$; in particular

each vertex of $\mathsf{aux}_H(\mathcal{P})$ has degree at most $2|V(H)|$. As $\mathsf{aux}_H(\mathcal{P})$ contains k vertices, we deduce that there are at most $(2|V(H)|)^k \leq (2n)^k$ possible degree sequences for $\mathsf{aux}_H(\mathcal{P})$.

Since the number of partitions of $\{v_1, \ldots, v_k\}$ is bounded by $2^{k \log_2 k}$. We conclude that $\simeq$ partitions $\Pi(H)$ into at most $n^k \cdot 2^{k(\log_2 k+1)}$ equivalences classes.

It remains to prove that we can compute $\mathsf{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk \log_2(nk))$. First observe that, for every $\mathcal{P} \in \Pi(H)$, we can compute $\mathsf{aux}_H(\mathcal{P})$ in time $O(nk)$. Moreover, we can also compute the degree sequence of $\mathsf{aux}_H(\mathcal{P})$ and the connected components of $\mathsf{aux}_H(\mathcal{P})$ in time $O(nk)$. Thus, by using the right data structures, we can decide whether $\mathcal{P}_1 \simeq \mathcal{P}_2$ in time $O(nk)$. Furthermore, by using a self-balancing binary search tree, we can compute $\mathsf{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk \log_2(|\mathsf{reduce}_H(\mathcal{A})|))$. Since $\log_2(|\mathsf{reduce}_H(\mathcal{A})|) \leq k \log_2(2nk)$, we conclude that $\mathsf{reduce}_H(\mathcal{A})$ is computable in time $O(|\mathcal{A}| \cdot nk^2 \log_2(nk))$. □

The rest of this section is dedicated to prove that, for a set of partial solutions $\mathcal{A}$ of H, the set $\mathsf{reduce}_H(\mathcal{A})$ is equivalent to $\mathcal{A}$, i.e., if $\mathcal{A}$ contains a partial solution that forms a Hamiltonian cycle with a complement solution, then $\mathsf{reduce}_H(\mathcal{A})$ also. Our results are based on a kind of equivalence between Hamiltonian cycles and red–blue Eulerian trails. The following observation is one direction of this equivalence.

Lemma 3 *If $\mathcal{P} \in \Pi(H)$ and $\mathcal{Q} \in \overline{\Pi}(H)$ form a Hamiltonian cycle, then the multigraph $\mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$ admits a red–blue Eulerian trail.*

Proof Suppose that $\mathcal{P} \in \Pi(H)$ and $\mathcal{Q} \in \overline{\Pi}(H)$ form a Hamiltonian cycle C. Let $M := \mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$. From the definitions of a partial solution and of a complement solution, there is a sequence $(P_1, Q_1, \ldots, P_\ell, Q_\ell)$ of paths in $\mathcal{P}$ and $\mathcal{Q}$ such that

- $P_1, P_2, \ldots, P_\ell$ are all the maximal paths in $H_{|\mathcal{P}}$,
- $Q_1, Q_2, \ldots, Q_\ell$ are all the H-paths in $G_{|\mathcal{Q}}$,
- $P_1, Q_1, \ldots, P_\ell, Q_\ell$ appear in C in this order,
- for each $x \in [\ell]$, the first end-vertex of P_x is the last end-vertex of Q_{x-1} and the last end-vertex of P_x is the first end-vertex of Q_x (indices are considered modulo ℓ).

Observe that each maximal path P_x in $H_{|\mathcal{P}}$ with end-vertices labeled i and j is associated with a red edge in M, say e_x with $\mathsf{mult}_M(e_x) = \{v_i, v_j\}$ if $i \neq j$ or $\mathsf{mult}_M(e_x) = \{v_i\}$ if $i = j$ such that the edges $e_1, \ldots, e_\ell$ are pairwise distinct and $E(\mathsf{aux}_H(\mathcal{P})) = \{e_1, \ldots, e_\ell\}$. Similarly, each H-path Q_y of $G_{|\mathcal{Q}}$ with end-vertices labeled i and j is associated with a blue edge f_y in M with $\mathsf{mult}_M(f_y) = \{v_i, v_j\}$ if $i \neq j$ or $\mathsf{mult}_M(f_y) = \{v_i\}$ if $i = j$ such that the edges $f_1, \ldots, f_\ell$ are pairwise distinct and $E(\mathsf{aux}_H(\mathcal{Q})) = \{f_1, \ldots, f_\ell\}$. It is not difficult to see that $(e_1, f_1, \ldots, e_\ell, f_\ell)$ is a red–blue Eulerian trail of $\mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$. □

Next, we prove the other direction. We use the properties of an irredundant k-expression described in Lemma 1.

Lemma 4 *Let $\mathcal{P} \in \Pi(H)$. If there exists a complement solution $\mathcal{Q}$ of H such that $\mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$ admits a red–blue Eulerian trail, then there exists $\mathcal{Q}^\star \in \overline{\Pi}(H)$ such that $\mathcal{P}$ and $\mathcal{Q}^\star$ form a Hamiltonian cycle.*

Proof Let $T = (v_{a_1}, r_1, v_{c_1}, b_1, v_{a_2}, r_2, v_{c_2}, \ldots, v_{a_\ell}, r_\ell, v_{c_\ell}, b_\ell, v_{a_1})$ be a red–blue Eulerian trail of $\mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$ with $r_1, \ldots, r_\ell \in E(\mathsf{aux}_H(\mathcal{P}))$ and $b_1, \ldots, b_\ell \in E(\mathsf{aux}_H(\mathcal{Q}))$. In the following, the indexes are modulo ℓ.

For each $i \in [\ell]$, we associate r_i with a maximal path P_i of $H_{|\mathcal{P}}$ with end-vertices labeled a_i and c_i and we associate b_i with an H-path Q_i of $G_{|\mathcal{Q}}$ with end-vertices labeled c_i and a_{i+1}, such that $P_1, \ldots, P_\ell, Q_1, \ldots, Q_\ell$ are all pairwise distinct.

For every $i \in [\ell]$, we construct from Q_i an H-path $Q_i^\star$ of G by doing the following. Let u, v be respectively the last end-vertex of P_i and the first end-vertex of P_{i+1}. Observe that u and the first vertex of Q_i are labeled c_i, and v and the last vertex of Q_i are labeled a_{i+1}. We distinguish two cases:

- Suppose that $Q_i = (x, xy, y)$, i.e., Q_i uses only one edge. Since $\mathcal{Q}$ is a complement solution of H, we have $xy \in E(G) \backslash E(H)$. By Lemma 1, we have $uv \in E(G) \backslash E(H)$. In this case, we define $Q_i^\star = (u, uv, v)$.
- Assume now that $Q_i = (x, xy, y, \ldots, w, wz, z)$ with $w, y \in V(G) \backslash V(H)$ (possibly, $w = y$). Since x and u have the same label in H, by Lemma 1, we have $N_G(x) \backslash V(H) = N_G(u) \backslash V(H)$. Hence, u is also adjacent to y. Symmetrically, we have v is adjacent to w. In this case, we define $Q_i^\star = (u, uy, y, \ldots, w, wv, v)$, i.e., the path with the same internal vertices as Q_i and with end-vertices u and v.

In both cases, we end up with a path $Q_i^\star$ that uses the same internal vertices as Q_i and whose end-vertices are the last vertex of P_i and the first vertex of P_{i+1}. We conclude that

$$P_1 - Q_1^\star - \cdots - P_\ell - Q_\ell^\star$$

is a Hamiltonian cycle.

Let $\mathcal{Q}^\star$ be the set of edges used by the paths $Q_1^\star, \ldots, Q_\ell^\star$. By construction, we have $\mathcal{Q}^\star \subseteq E(G) \backslash E(H)$, and thus $\mathcal{Q}^\star \in \overline{\Pi}(H)$. Observe that, for every $i \in [\ell]$, the labels of the end-vertices of $Q_i^\star$ are the same as those of Q_i. Consequently, we have $\mathsf{aux}_H(\mathcal{Q}^\star) = \mathsf{aux}_H(\mathcal{Q})$. □

It is well known that a connected multigraph contains an Eulerian trail if and only if every vertex has even degree. As an extension, Kotzig [26] proved that a connected two-edge colored graph (without loops and multiple edges) contains a red–blue Eulerian trail if and only if each vertex is incident with the same number of edges for both colors. This result can be easily generalized to multigraphs by replacing red edge with a path of length 3 whose colors are red, blue, red in the order, and replacing blue edge with a path of length 3 whose colors are blue, red, blue in the order. For the completeness of our paper, we add its proof.

Let G be a multigraph whose edges are colored red or blue, and let R and B be respectively the set of red and blue edges. For a vertex $v \in V(G)$, we let $\mathsf{rdeg}_G(v)$ and $\mathsf{bdeg}_G(v)$ be respectively the degree of v in $G_{|R}$ and $G_{|B}$. Recall that a loop is counted twice in the degree of a vertex.

Lemma 5 (Kotzig [26]) *Let G be a connected multigraph whose edges are colored red or blue. Then G has a red–blue Eulerian trail if and only if, for every vertex v, $\mathsf{bdeg}_G(v) = \mathsf{rdeg}_G(v)$.*

Proof One can easily check that if G has a red–blue Eulerian trail, then for every vertex v, $\mathsf{bdeg}_G(v) = \mathsf{rdeg}_G(v)$. Indeed, if $T = (v_1, e_1, v_2, \ldots, v_\ell, e_\ell, v_1)$ is a red–blue Eulerian trail, then, for $2 \leq i \leq \ell$, e_{i-1} and e_i have different colors, and e_1 and e_ℓ have different colors, we can conclude that the blue edges incident with a vertex v are in 1-to-1 correspondence with the red edges incident with v (by counting twice the loops).

Let us now prove the other direction. Let $T := (v_1, e_1, v_2, e_2, \ldots, v_i, e_i, v_{i+1})$ be a longest red–blue trail. We may assume that e_1 is colored red. First observe that $v_1 = v_{i+1}$. Otherwise, $\mathsf{bdeg}_T(v_1) + 1 = \mathsf{rdeg}_T(v_1)$ and thus, there is a blue edge in $E(G)\backslash E(T)$ incident with v_1. So, we can extend T by adding this edge, a contradiction. Thus, $v_1 = v_{i+1}$.

Next we show that e_i is colored blue. Suppose e_i is colored red. Then $\mathsf{bdeg}_T(v_1) + 2 = \mathsf{rdeg}_T(v_1)$ and thus, there is a blue edge in $E(G)\backslash E(T)$ incident with v_1. So, we can extend T by adding this edge, a contradiction. Thus, e_i is colored red, and it implies that T is a closed red–blue trail. It means that T can be considered as a closed red–blue trail starting from any vertex of T and following the same order or the reverse order of T.

Now, we show that $V(G) = V(T)$. Suppose $V(G)\backslash V(T)$ is non-empty. Since G is connected, there is an edge vw with $v \in V(T)$ and $w \in V(G)\backslash V(T)$. If vw is a red edge, then starting from this edge and following T from a blue edge incident with v, we can find a red–blue trail longer than T, a contradiction. The same argument holds when vw is a blue edge. Therefore, we have that $V(G) = V(T)$. By a similar argument, one can show that $E(G) = E(T)$; if there is an edge vw in $E(G)\backslash E(T)$, we can extend T by putting vw at the beginning. So, $E(G) = E(T)$.

We conclude that T is a red–blue Eulerian trail, as required. □

In order to prove the correctness of our algorithm, we need the following relation between subsets of partial solutions.

Definition 2 Let $\mathcal{A}$ and $\mathcal{B}$ be two subsets of $\Pi(H)$. We write $\mathcal{A} \lesssim_H \mathcal{B}$ if, for every multigraph $\mathcal{M}$ whose edges are colored blue, whenever there exists $\mathcal{P}_1 \in \mathcal{B}$ such that $\mathsf{aux}_H(\mathcal{P}_1) \uplus \mathcal{M}$ admits a red–blue Eulerian trail, there exists $\mathcal{P}_2 \in \mathcal{A}$ such that $\mathsf{aux}_H(\mathcal{P}_2) \uplus \mathcal{M}$ admits a red–blue Eulerian trail.

The main idea of our algorithm for Hamiltonian Cycle, is to compute, for every k-labeled graph H arising in ϕ, a set $\mathcal{A} \subseteq \Pi(H)$ of small size such that $\mathcal{A} \lesssim_H \Pi(H)$. Indeed, by Lemmas 3 and 4, $\mathcal{A} \lesssim_H \Pi(H)$ implies that if there exist $\mathcal{P} \in \Pi(H)$ and $\mathcal{Q} \in \overline{\Pi}(H)$ such that $\mathcal{P}$ and $\mathcal{Q}$ form a Hamiltonian cycle, then there exist $\mathcal{P}^\star \in \mathcal{A}$ and $\mathcal{Q}^\star \in \overline{\Pi}(H)$ such that $\mathcal{P}^\star$ and $\mathcal{Q}^\star$ form a Hamiltonian cycle. The following lemma is the key of our algorithm.

Lemma 6 *Let $\mathcal{A} \subseteq \Pi(H)$. Then $\mathsf{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{A}$.*

Proof Let $\mathcal{P} \in \mathcal{A}$ and $\mathcal{M}$ be a multigraph whose edges are colored blue such that $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ admits a red–blue Eulerian trail. By definition, $\mathsf{reduce}_H(\mathcal{A})$ contains a partial solution $\mathcal{P}^\star$ such that $\mathcal{P} \simeq \mathcal{P}^\star$. As $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ contains a red–blue Eulerian trail, by Lemma 5, we have that

– $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ is connected and
– for every $i \in [k]$, $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) = \mathsf{deg}_{\mathcal{M}}(v_i)$.

Since $\mathsf{aux}_H(\mathcal{P})$ has the same set of connected components as $\mathsf{aux}_H(\mathcal{P}^\star)$, the multigraph $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}$ is also connected. Moreover, for every $i \in [k]$, we have

$$\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) = \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P}^\star)}(v_i) = \mathsf{deg}_{\mathcal{M}}(v_i).$$

By Lemma 5, we conclude that $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}$ admits a red–blue Eulerian trail.

Thus, for every $\mathcal{P} \in \mathcal{A}$ and multigraph $\mathcal{M}$ with blue edges such that $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ admits a red–blue Eulerian trail, there exists $\mathcal{P}^\star \in \mathsf{reduce}_H(\mathcal{A})$ such that $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}$ admits a red–blue Eulerian trail. Hence, we have $\mathsf{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{A}$. □

Lemma 7 *Let $\mathcal{A}, \mathcal{B} \subseteq \Pi(H)$. If $\mathcal{A} \lesssim_H \mathcal{B}$, then $\mathsf{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{B}$.*

Proof One easily checks that $\lesssim_H$ is a transitive relation. Now, assuming that $\mathcal{A} \lesssim_H \mathcal{B}$, we have $\mathsf{reduce}_H(\mathcal{A}) \lesssim \mathcal{B}$ because $\mathsf{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{A}$ by Lemma 6. □

5 Hamiltonian Cycle Problem

In this section, we present our algorithm solving Hamiltonian Cycle. Our algorithm computes recursively, for every k-labeled graph H arising in the k-expression of G, a set $\mathcal{A}_H$ such that $\mathcal{A}_H \lesssim_H \Pi(H)$ and $|\mathcal{A}_H| \leq n^k \cdot 2^{k(\log_2(k)+1)}$. In order to prove the correctness of our algorithm, we need the following lemmas which prove that the operations we use to compute sets of partial solutions preserve the relation $\lesssim_H$.

Lemma 8 *Let $H = \rho_{i\to j}(D)$. If $\mathcal{A}_D \lesssim_D \Pi(D)$, then $\mathcal{A}_D \lesssim_H \Pi(H)$.*

Proof First, observe that H has the same set of vertices and edges as D. Thus, we have $\Pi(H) = \Pi(D)$ and $\overline{\Pi}(H) = \overline{\Pi}(D)$. Suppose that $\mathcal{A}_D \lesssim_D \Pi(D)$.

Let $\mathcal{P} \in \Pi(H)$ and $\mathcal{M}$ be a multigraph whose edges are colored blue such that $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ contains a red–blue Eulerian trail T. To prove the lemma, it is sufficient to prove that there exists $\mathcal{P}^\star \in \mathcal{A}_D$ such that $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}$ contains a red–blue Eulerian trail.

Let f be a bijective function such that

– for every edge e of $\mathsf{aux}_D(\mathcal{P})$ with endpoints v_ℓ and v_i, for some ℓ, $f(e)$ is an edge of $\mathsf{aux}_H(\mathcal{P})$ with endpoints v_ℓ and v_j, and
– for every loop e with endpoint v_i, $f(e)$ is a loop of $\mathsf{aux}_H(\mathcal{P})$ with endpoint v_j.

By construction of $\mathsf{aux}_D(\mathcal{P})$ and $\mathsf{aux}_H(\mathcal{P})$, such a function exists.

We construct the multigraph $\mathcal{M}'$ from $\mathcal{M}$ and T by successively doing the following:

– For every edge e of $\mathsf{aux}_D(\mathcal{P})$ with endpoints v_ℓ and v_i, take the subwalk $W = (v_\ell, f(e), v_j, e_a, v_a)$ of T. Replace e_a in $\mathcal{M}$ by an edge e'_a with endpoints v_i and v_a.

– For every loop e with endpoint v_i in $\mathsf{aux}_D(\mathcal{P})$, take the subwalk $W = (v_a, e_a, v_j, f(e), v_j, e_b, v_b)$ of T. Replace e_a (respectively e_b) in $\mathcal{M}$ by an edge e'_a (resp. e'_b) with endpoints v_a and v_i (resp. v_i and v_b).

By construction, one can construct from T a red–blue Eulerian trail of $\mathsf{aux}_D(\mathcal{P}) \uplus \mathcal{M}'$. Since $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists $\mathcal{P}^\star \in \mathcal{A}_D$ such that $\mathsf{aux}_D(\mathcal{P}^\star) \uplus \mathcal{M}'$ contains a red–blue Eulerian trail. Observe that $\mathsf{aux}_H(\mathcal{P})$ (respectively $\mathcal{M}$) is obtained from $\mathsf{aux}_D(\mathcal{P}^\star)$ (resp. $\mathcal{M}'$) by replacing each edge associated with $\{v_i, v_k\}$ or $\{v_i\}$ in $\mathsf{aux}_D(\mathcal{P}^\star)$ (resp. $\mathcal{M}'$) with an edge associated with $\{v_j, v_k\}$ or $\{v_j\}$ respectively. We conclude that $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}$ admits a red–blue Eulerian trail. □

Lemma 9 *Let $H = D \oplus F$. If $\mathcal{A}_D \lesssim_D \Pi(D)$ and $\mathcal{A}_F \lesssim_F \Pi(F)$, then $(\mathcal{A}_D \otimes \mathcal{A}_F) \lesssim_H \Pi(H)$.*

Proof Observe that $V(D)$ and $V(F)$ are disjoint. Consequently, we have $\Pi(H) = \Pi(D) \otimes \Pi(F)$, and, for all $\mathcal{P}_D \in \Pi(D)$ and $\mathcal{P}_F \in \Pi(F)$, we have $\mathsf{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) = \mathsf{aux}_H(\mathcal{P}_D) \uplus \mathsf{aux}_H(\mathcal{P}_F)$. Suppose that $\mathcal{A}_D \lesssim_D \Pi(D)$ and $\mathcal{A}_F \lesssim_F \Pi(F)$.

Let $\mathcal{P}_D \in \Pi(D)$ and $\mathcal{P}_F \in \Pi(F)$, and let $\mathcal{M}$ be a multigraph whose edges are colored blue such that there exists a red–blue Eulerian trail T in $\mathsf{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) \uplus \mathcal{M}$. It is sufficient to prove that there exist $\mathcal{P}^\star_D \in \mathcal{A}_D$ and $\mathcal{P}^\star_F \in \mathcal{A}_F$ such that $\mathsf{aux}_H(\mathcal{P}^\star_D \cup \mathcal{P}^\star_F) \uplus \mathcal{M}$ admits a red–blue Eulerian trail.

We begin by proving that there exists $\mathcal{P}^\star_D \in \mathcal{A}_D$ such that $\mathsf{aux}_H(\mathcal{P}^\star_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains a red–blue Eulerian trail. In order to do so, we construct from $\mathsf{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ a multigraph $\mathcal{M}'$ by successively repeating the following: take a maximal sub-walk W of T which uses alternately blue edges and red edges from $\mathsf{aux}_H(\mathcal{P}_F) \uplus \mathcal{M}$, remove these edges and add a blue edge between the two end-vertices of W.

By construction of $\mathcal{M}'$, for every $\mathcal{P}'_D \in \Pi(D)$, if $\mathsf{aux}_D(\mathcal{P}'_D) \uplus \mathcal{M}'$ admits a red–blue Eulerian trail, then $\mathsf{aux}_H(\mathcal{P}'_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains a red–blue Eulerian trail also. We also deduce from this construction that the multigraph $\mathsf{aux}_D(\mathcal{P}_D) \uplus \mathcal{M}'$ contains a red–blue Eulerian trail. As $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists $\mathcal{P}^\star_D$ such that $\mathsf{aux}_D(\mathcal{P}^\star_D) \uplus \mathcal{M}'$ contains a red–blue Eulerian trail. We conclude that $\mathsf{aux}_H(\mathcal{P}^\star_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains also a red–blue Eulerian trail.

Symmetrically, we can prove that there exists $\mathcal{P}^\star_F \in \mathcal{A}_F$ such that $\mathsf{aux}_H(\mathcal{P}^\star_D \cup \mathcal{P}^\star_F) \uplus \mathcal{M}$ contains a red–blue Eulerian trail. This proves the lemma. □

For two k-labeled subgraphs H and D arising in the k-expression of G such that $H = \eta_{i,j}(D)$, we denote by $\mathcal{E}^H_{i,j}$ the set of edges whose endpoints are labeled i and j, i.e., $\{uv \in E(H) \mid \mathsf{lab}_H(v) = i \ \wedge \ \mathsf{lab}_H(v) = j\}$. For $\mathcal{P} \in \Pi(H)$, we denote by $\mathcal{P} + (i, j)$ the set of all partial solutions $\mathcal{P}'$ of $\Pi(H)$ obtained by the union of $\mathcal{P}$ and an edge uv in $\mathcal{E}^H_{i,j}$. Observe that u and v must be the endpoints of two distinct maximal paths of $H_{|\mathcal{P}}$. We extend this notation to sets of partial solutions; for every $\mathcal{A} \subseteq \Pi(H)$, we denote by $\mathcal{A} + (i, j)$, the set $\bigcup_{\mathcal{P} \in \mathcal{A}} (\mathcal{P} + (i, j))$. It is worth noting that $\Pi(D) \subseteq \Pi(H)$ and thus the operator $+(i, j)$ is well defined for a partial solution in $\Pi(D)$.

Moreover, for every $\mathcal{A} \subseteq \Pi(D)$ and integer $t \geq 0$, we define the set $\mathcal{A}^t$ as follows

$$\mathcal{A}^t := \begin{cases} \mathcal{A} & \text{if } t = 0, \\ \mathsf{reduce}_H(\mathcal{A}^{t-1} + (i, j)) & \text{otherwise.} \end{cases}$$

Observe that each set $\mathcal{P}$ in $\mathcal{A}^t$ is a partial solution of H and $|\mathcal{P} \cap \mathcal{E}^H_{i,j}| = t$.

Lemma 10 *Let $H = \eta_{i,j}(D)$ such that $E(D) \cap \mathcal{E}^H_{i,j} = \emptyset$. If $\mathcal{A}_D \lesssim_D \Pi(D)$, then we have $\mathcal{A}^0_D \cup \cdots \cup \mathcal{A}^n_D \lesssim_H \Pi(H)$.*

Proof Suppose that $\mathcal{A}_D \lesssim_D \Pi(D)$. We begin by proving the following claim.

Claim Let $\mathcal{A}, \mathcal{B} \subseteq \Pi(H)$. If $\mathcal{A} \lesssim_H \mathcal{B}$, then $\mathcal{A} + (i, j) \lesssim_H \mathcal{B} + (i, j)$.

Proof Suppose that $\mathcal{A} \lesssim_H \mathcal{B}$. Let $\mathcal{P} \in \mathcal{B} + (i, j)$ and $\mathcal{M}$ be a multigraph with blue edges such that $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ admits a red–blue Eulerian trail. Take $xy \in \mathcal{E}^H_{i,j}$ such that $\mathcal{P}' := \mathcal{P} - xy$ belongs to $\mathcal{B}$ and $x \in \mathsf{lab}_H^{-1}(i)$ and $y \in \mathsf{lab}_H^{-1}(j)$. Let $\mathcal{M}'$ be the multigraph obtained by adding a blue edge f with endpoints v_i and v_j to $\mathcal{M}$.

We claim that the multigraph $\mathsf{aux}_H(\mathcal{P}') \uplus \mathcal{M}'$ admits a red–blue Eulerian trail. Note that there is a path $P \in \mathcal{P}$ containing the edge xy, and when we remove xy from $\mathcal{P}$, we divide P into two maximal subpaths, say P_1 and P_2. Without loss of generality, we may assume that P_1 contains x and P_2 contains y. Let x' and y' be the other end-vertices of P_1 and P_2, respectively, and let $i' := \mathsf{lab}_H(x')$ and $j' := \mathsf{lab}_H(y')$. Note that $\mathsf{aux}_H(\mathcal{P}')$ can be obtained from $\mathsf{aux}_H(\mathcal{P})$ by removing an edge e associated with $\{v_{i'}, v_{j'}\}$ and adding two edges e_1 and e_2 associated with $\{v_{i'}, v_i\}$ and $\{v_j, v_{j'}\}$ respectively. So, we can obtain a red–blue Eulerian trail of $\mathsf{aux}_H(\mathcal{P}') \uplus \mathcal{M}'$ from a red–blue Eulerian trail of $\mathsf{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ by replacing $(v_{i'}, e, v_{j'})$ with the sequence $(v_{i'}, e_1, v_i, f, v_j, e_2, v_{j'})$ where f is the blue edge we add to $\mathcal{M}$ to obtain $\mathcal{M}'$. It implies the claim.

Now, since $\mathcal{A} \lesssim_H \mathcal{B}$, there exists $\mathcal{P}^\star \in \mathcal{A}$ such that $\mathsf{aux}_H(\mathcal{P}^\star) \uplus \mathcal{M}'$ admits a red–blue Eulerian trail T. Let W be the subwalk of T such that $W = (v_a, e_a, v_i, f, v_j, e_b, v_b)$. Take two maximal paths P_1 and P_2 in $H_{|\mathcal{P}^\star}$ such that the end-vertices of P_1 (respectively P_2) are labeled a and i (resp. b and j). Let $\widehat{\mathcal{P}}$ be the partial solution of H obtained from $\mathcal{P}^\star$ by adding the edge in $\mathcal{E}^H_{i,j}$ between the end-vertex of P_1 labeled i and the end-vertex of P_2 labeled j. By construction, we have $\widehat{\mathcal{P}} \in \mathcal{A} + (i, j)$ and $\mathsf{aux}_H(\widehat{\mathcal{P}}) \uplus \mathcal{M}$ admits a red–blue Eulerian trail. We conclude that $\mathcal{A} + (i, j) \lesssim_H \mathcal{B} + (i, j)$. □

Let $\Pi(D) + t(i, j)$ be the set of partial solutions of H obtained by applying t times the operation $+(i, j)$ on $\Pi(D)$. Since every partial solution of H is obtained from the union of a partial solution of D and a subset of $\mathcal{E}^H_{i,j}$ of size at most n, we have $\Pi(H) = \bigcup_{0 \le t \le n}(\Pi(D) + t(i, j))$.

Since $V(D) = V(H)$ and $E(D) \subseteq E(H)$, we have $\mathcal{A}^0_D = \mathcal{A}_D \lesssim_H \Pi(D) + 0(i, j)$. Let $\ell \in \{1, \ldots, n\}$ and suppose that $\mathcal{A}^{\ell-1}_D \lesssim_H \Pi(D) + (\ell - 1)(i, j)$. From Claim 5, we have $\mathcal{A}^{\ell-1}_D + (i, j) \lesssim_H \Pi(D) + \ell(i, j)$. By Lemma 7, we deduce that $\mathcal{A}^\ell_D = \mathsf{reduce}(\mathcal{A}^{\ell-1}_D + (i, j)) \lesssim_H \Pi(D) + \ell(i, j)$.

Thus, by recurrence, for every $0 \le \ell \le n$, we have $\mathcal{A}^\ell_D \lesssim_H \Pi(D) + \ell(i, j)$. We conclude that $\mathcal{A}^0_D \cup \cdots \cup \mathcal{A}^n_D \lesssim_H \Pi(H)$. □

We are now ready to prove the main theorem of this paper.

Theorem 2 *There exists an algorithm that, given an n-vertex graph G and a k-expression ϕ of G, solves* Hamiltonian Cycle *in time $O(n^{2k+5} \cdot 2^{2k(\log_2(k)+1)} \cdot k^3 \cdot \log_2(nk))$.*

Proof Since every k-expression can be transformed into an irredundant k-expression in linear time, we may assume that ϕ is an irredundant k-expression.

We do a bottom-up traversal of the k-expression and at each k-labeled graph H arising in ϕ, we compute a set $\mathcal{A}_H \subseteq \Pi(H)$ such that $|\mathcal{A}_H| \leq n^k 2^{k(\log(k)+1)}$ and $\mathcal{A}_H \lesssim_H \Pi(H)$, by doing the following:

- If $H = i(v)$, then we have $\Pi(H) = \{\emptyset\}$ because $E(H) = \emptyset$. In this case, we set $\mathcal{A}_H := \{\emptyset\}$. Obviously, we have $\mathcal{A}_H \lesssim_H \Pi(H)$.
- If $H = \rho_{i,j}(D)$, then we set $\mathcal{A}_H := \mathcal{A}_D$.
- If $H = D \oplus F$, then we set $\mathcal{A}_H := \mathsf{reduce}_H(\mathcal{A}_D \otimes \mathcal{A}_F)$.
- If $H = \eta_{i,j}(D)$, then we set $\mathcal{A}_H := \mathsf{reduce}_H(\mathcal{A}_D^0 \cup \cdots \cup \mathcal{A}_D^n)$.

For the last three cases, we deduce, by induction, from Lemmas 7–10 that $\mathcal{A}_H \lesssim_H \Pi(H)$. Moreover by the use of the function reduce_H, by Lemma 2, we have $|\mathcal{A}_H| \leq n^k \cdot 2^{k(\log(k)+1)}$.

We now explain how our algorithm decides whether G admits a Hamiltonian cycle. We claim that G has a Hamiltonian cycle if and only if there exist two k-labeled graphs H and D arising in ϕ with $V(H) = V(G)$ and $H = \eta_{i,j}(D)$, and there exists $\mathcal{P} \in \mathcal{A}_D$ such that, for every $\ell \in [k]\backslash\{i, j\}$, we have $\mathsf{deg}_{\mathsf{aux}_D(\mathcal{P})}(v_\ell) = 0$ and $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i) = \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_j) > 0$.

First suppose that G contains a Hamiltonian cycle C and take the k-labeled graph H arising in ϕ such that $E(C) \subseteq E(H)$ and $|E(H)|$ is minimal. Note that the operations of taking the disjoint union of two graphs or relabeling cannot create a Hamiltonian cycle. Thus, by minimality, we have $H = \eta_{i,j}(D)$ such that

- D is a k-labeled graph arising in ϕ and $i, j \in [k]$,
- $E(C) \nsubseteq E(D)$.

It follows that $E(C)\backslash E(D) \subseteq \mathcal{E}_{i,j}^H$. Let $\mathcal{P} := E(C) \cap E(D)$ and $\mathcal{Q} := E(C) \cap \mathcal{E}_{i,j}^H$. Observe that $\mathcal{P} \in \Pi(D)$ and $\mathcal{Q} \in \overline{\Pi}(D)$. By Lemma 3, the multigraph $\mathsf{aux}_D(\mathcal{P}) \uplus \mathsf{aux}_D(\mathcal{Q})$ contains a red–blue Eulerian trail. Since $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists $\mathcal{P}^\star \in \mathcal{A}_D$ such that $\mathsf{aux}_D(\mathcal{P}^\star) \uplus \mathsf{aux}_D(\mathcal{Q})$ contains a red–blue Eulerian trail. As $\mathcal{Q} \subseteq \mathcal{E}_{i,j}^H$, we deduce that, for every $\ell \in [k]\backslash\{i, j\}$, we have $\mathsf{deg}_{\mathsf{aux}_D(\mathcal{P}^\star)}(v_\ell) = 0$ and $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P}^\star)}(v_i) = \mathsf{deg}_{\mathsf{aux}_H(\mathcal{P}^\star)}(v_j)$.

For the other direction, suppose that the latter condition holds. Let $\mathcal{Q}$ be the graph on the vertex set $V(G)$ such that it contains exactly $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i)$ many edges between the set of vertices labeled i and the set of vertices labeled j. Observe that $\mathsf{aux}_H(\mathcal{Q})$ consists of $\mathsf{deg}_{\mathsf{aux}_H(\mathcal{P})}(v_i)$ many edges between v_i and v_j. Therefore, by Lemma 5, $\mathsf{aux}_H(\mathcal{P}) \uplus \mathsf{aux}_H(\mathcal{Q})$ admits a red–blue Eulerian trail. Then, by Lemma 4, there exists $\mathcal{Q}^\star \in \overline{\Pi}(H)$ such that $\mathcal{P}$ and $\mathcal{Q}^\star$ form a Hamiltonian cycle, as required.

Running Time Let H be a k-labeled graph arising in ϕ. Observe that if $H = i(v)$ or $H = \rho_{i\to j}(D)$, then we compute $\mathcal{A}_H$ in time $O(1)$.

By Lemma 2, for every $\mathcal{A} \subseteq \Pi(H)$, we can compute $\mathsf{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk^2 \log_2(nk))$. Observe that, for every k-labeled graph D arising in ϕ and such that $\mathcal{A}_D$ is computed before $\mathcal{A}_H$, we have $|\mathcal{A}_D| \leq n^k \cdot 2^{k(\log_2(k)+1)}$. It follows that:

– If $H = D \oplus F$, then we have $|\mathcal{A}_D \otimes \mathcal{A}_F| \leq n^{2k} \cdot 2^{2k(\log_2(k)+1)}$. Thus, we can compute $\mathcal{A}_H := \mathsf{reduce}_H(\mathcal{A}_D \otimes \mathcal{A}_F)$ in time

$$O(n^{2k+1} \cdot 2^{2k(\log_2(k)+1)} \cdot k^2 \log_2(nk)).$$

– If $H = \eta_{i,j}(D)$, then we can compute $\mathcal{A}_H$ in time

$$O(n^{k+4} \cdot 2^{k(\log_2(k)+1)} \cdot k^2 \log_2(nk)).$$

First observe that, for every partial solution $\mathcal{P}$ of H, we have $|\mathcal{P} + (i, j)| \leq n^2$ and we can compute the set $\mathcal{P} + (i, j)$ in time $O(n^2)$. Moreover, by Lemma 2, for every $\ell \in \{0, \ldots, n-1\}$, we have $|\mathcal{A}_D^\ell| \leq n^k \cdot 2^{k(\log_2(k)+1)}$ and thus, we deduce that $|\mathcal{A}_D^\ell + (i, j)| \leq n^{k+2} \cdot 2^{k(\log_2(k)+1)}$ and that $\mathcal{A}_D^{\ell+1}$ can be computed in time $O(n^{k+3} \cdot 2^{k(\log_2(k)+1)} \cdot k^2 \log_2(nk))$. Thus, we can compute the sets $\mathcal{A}_D^1, \ldots, \mathcal{A}_D^n$ in time $O(n^{k+4} \cdot 2^{k(\log(k)+1)} \cdot k^2 \log_2(nk))$. The running time to compute $\mathcal{A}_H$ from $\mathcal{A}_D$ follows from $|\mathcal{A}_D^0 \cup \cdots \cup \mathcal{A}_D^n| \leq n^{k+1} \cdot 2^{2k(\log_2(k)+1)}$.

Since ϕ uses at most $O(n)$ disjoint union operations and $O(nk^2)$ unary operations, we deduce that the total running time of our algorithm is

$$O(n^{2k+5} \cdot 2^{2k(\log_2(k)+1)} \cdot k^4 \log_2(nk)).$$

□

6 Directed Hamiltonian Cycle

In this section, we explain how to adapt our approach for directed graphs. A *k-labeled digraph* is a pair (G, lab_G) of a digraph G and a function lab_G from $V(G)$ to $[k]$. Directed clique-width is also defined in [10], and it is based on the four operations, where the three operations of creating a digraph, taking the disjoint union of two labeled digraphs, and relabeling a digraph are the same, and the operation of adding edges is replaced with the following:

– For a labeled digraph G and distinct labels $i, j \in [k]$, add all the non-existent arcs from vertices with label i to vertices with label j (so we do not add arcs of both directions altogether).

A *directed clique-width k-expression* and *directed clique-width* are defined in the same way. A *directed Hamiltonian cycle* of a digraph G is a directed cycle containing all the vertices of G. The Directed Hamiltonian Cycle problem asks, for a given digraph G, whether G contains a directed Hamiltonian cycle or not.

With a similar approach, we can show the following.

Theorem 3 *There exists an algorithm that, given an n-vertex digraph G and a directed clique-width k-expression of G, solves* Directed Hamiltonian Cycle *in time* $n^{O(k)}$.

For Directed Hamiltonian Cycle, auxiliary graphs should be directed graphs. We define partial solutions and auxiliary multigraphs similarly, at the exception that a directed maximal path (resp. H-path) from a vertex labeled i to a vertex labeled j in a partial solution (resp. complement solution) corresponds to an arc (v_i, v_j) in its auxiliary multigraph.

Let G be an n-vertex directed graph and ϕ be a directed irredundant k-expression of G. Similar to the Proof of Theorem 2, for every k-labeled graph H arising in ϕ, we recursively compute a set $\mathcal{A}_H$ of small size such that $\mathcal{A}_H$ represents $\Pi(H)$ which is the set of all partial solutions of H.

It is not hard to see that Lemmas 3 and 4 hold also in the directed case. That is, we have an equivalence between directed Hamiltonian cycle in graphs and directed red–blue Eulerian trail in multigraphs. Thus, to adapt our ideas for undirected graphs, we only need to characterize the directed multigraphs which admit a red–blue Eulerian trail. Fleischner [18, Theorem VI.17] gave such a characterization for directed graphs without loops and multiple arcs, but the proof can easily be adapted for directed multigraphs.

Let M be a directed multigraph whose arcs are colored red or blue, and let R and B be respectively its set of red and blue arcs. We denote by M^* the digraph derived from M with $V(M^*) := \{v_1, v_2 \mid v \in V(M)\}$ and $E(M^*) := \{(v_1, w_2) \mid (v, w) \in B\} \cup \{(v_2, w_1) \mid (v, w) \in R\}$. For a digraph G and a vertex v of G, we denote by $\mathsf{deg}^+_G(v)$ and $\mathsf{deg}^-_G(v)$, respectively, the outdegree and indegree of v in G.

Lemma 11 (Fleischner [18]) *Let M be a directed multigraph whose arcs are colored red or blue. The following are equivalent.*

1. *M has a red–blue Eulerian trail.*
2. *M^* has an Eulerian trail.*
3. *The underlying undirected graph of M^* has at most one connected component containing an edge, and, for each vertex v of M^*, $\mathsf{deg}^+_{M^*}(v) = \mathsf{deg}^-_{M^*}(v)$.*

In (3), the condition that "for each vertex v of M^*, $\mathsf{deg}^+_{M^*}(v) = \mathsf{deg}^-_{M^*}(v)$" can be translated to that, for each vertex v of M, the number of blue incoming arcs is the same as the number of red outgoing arcs, and the number of red incoming arcs is the same as the number of blue outgoing arcs. However, an important point is that instead of having that the underlying undirected graph of M^* has at most one connected component containing an edge, the condition that "the underlying graph of M is connected" or "M is strongly connected" is not sufficient, because the connectedness of M^* depends on the colors of arcs. We give an example. Let G be a graph on $\{x, y, z\}$ with red arcs (x, y), (y, z) and blue arcs (z, y), (y, x). Even though G is strongly connected, it does not have a red–blue Eulerian trail, and one can check that M^* has two connected components containing an edge.

To decide whether the underlying undirected graph of M^* has at most one connected component containing an edge, multiple arcs are useless. So, it is enough to keep one partial solution $\mathcal{P}$ for each degree sequence in $\mathsf{aux}_H(\mathcal{P})$ and for each set $\{\mathsf{mult}_{\mathsf{aux}_H(\mathcal{P})}(e) \mid e \in E(\mathsf{aux}_H(\mathcal{P}))\}$.

Let $\simeq$ be the equivalence relation on $\Pi(H)$ such that $\mathcal{P}_1 \simeq \mathcal{P}_2$ if the following are satisfied:

– for every pair (v, w) of vertices in $\{v_1, \ldots, v_k\}$, there exists an arc from v to w in $\mathsf{aux}_H(\mathcal{P}_1)$ if and only if there exists one in $\mathsf{aux}_H(\mathcal{P}_2)$,
– for every vertex v in $\{v_1, \ldots, v_k\}$, $\mathsf{deg}^+_{\mathsf{aux}_H(\mathcal{P}_1)}(v) = \mathsf{deg}^+_{\mathsf{aux}_H(\mathcal{P}_2)}(v)$ and $\mathsf{deg}^-_{\mathsf{aux}_H(\mathcal{P}_1)}(v) = \mathsf{deg}^-_{\mathsf{aux}_H(\mathcal{P}_2)}(v)$.

From Lemma 11 and the definition of M^*, we deduce the following fact.

Fact 1 *Let $\mathcal{P}_1, \mathcal{P}_2 \in \Pi(H)$. If $\mathcal{P}_1 \simeq \mathcal{P}_2$, then $\mathcal{P}_1$ is part of a directed Hamiltonian cycle in G if and only if $\mathcal{P}_2$ is part of a directed Hamiltonian cycle in G.*

From the definition of $\simeq$, one can show that $|\Pi(H)/\simeq| \leq n^{2k} \cdot 2^{k^2}$. Thus we can follow the lines of the proof for undirected graphs, and easily deduce that one can solve Directed Hamiltonian Cycle in time $n^{O(k)}$, where k is the directed clique-width of the given digraph.

7 Conclusion

We have proved that, given a k-expression, one can solve Hamiltonian Cycle in time $n^{O(k)}$, and also prove a similar variant for directed graphs.

One major open question related to clique-width is to determine whether it can be approximated within a constant factor. One can bypass this long-standing open problem by using a related parameter called *rank-width*. This parameter was introduced by Oum and Seymour [27] and admits an efficient algorithm to approximate it within a constant factor. Moreover, the clique-width of a graph is always bigger than its rank-width. On the other hand, rank-width is harder to manipulate than clique-width. To the best of our knowledge, there is no optimal algorithm known for basic problems such as Vertex Cover and Dominating Set, where the best algorithms run in time $2^{O(k^2)} \cdot n^{O(1)}$ [7] and the best lower bounds state that we can not solve these problems in time $2^{o(k)} \cdot n^{O(1)}$ unless ETH fails. Improving these algorithms or these lower bounds would be the natural way of continuing the works done on clique-width.

Recently, Bergougnoux and Kanté [2] proved that the Max Cut problem is solvable in time $n^{O(k)}$ where k is the clique-width of the input graph without assuming that the graph is given with a k-expression. For doing so, they used a related parameter called $\mathbb{Q}$-rank-width and the notion of d-neighbor equivalence. It would be interesting to know whether the same approach can be used for the Hamiltonian Cycle problem.

We conclude with one explicit question. A digraph D is an *out-tree* if D is an oriented tree (an undirected tree with orientations on edges) with only one vertex of indegree zero (called the root). The vertices of out-degree zero are called leaves of D. The *Min Leaf Out-Branching* problem asks for a given digraph D and an integer ℓ, whether there is a spanning out-tree of D with at most ℓ leaves. This problem generalizes Hamiltonian Path (and also Hamiltonian Cycle) by taking $\ell = 1$. Ganian, Hliněný, and Obdržálek [22] showed that there is an $n^{2^{O(k)}}$-time algorithm for solving Min Leaf Out- Branching problem, when a clique-width k-expression of a digraph D is given. We ask whether it is possible to drop down the exponential blow-up from $2^{O(k)}$ to $O(k)$.

Acknowledgements The authors would like to thank the anonymous reviewer for pointing out the previous mistake on red–blue Eulerian trails for directed graphs and for indicating proper citations.

References

1. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms **12**(2), 308–340 (1991)
2. Bergougnoux, B., Kanté, M.M.: More applications of the d-neighbor equivalence: connectivity and acyclicity constraints. In: 27th Annual European Symposium on Algorithms, ESA 2019, pp. 17:1–17:14. Munich/Garching, Germany, September 9–11, 2019. https://doi.org/10.4230/LIPIcs.ESA.2019.17
3. Bergougnoux, B., Kanté, M.M., Kwon, O.: An optimal XP algorithm for Hamiltonian cycle on graphs of bounded clique-width. In: Algorithms and Data Structures—15th International Symposium, WADS 2017, St. John's, NL, Canada, July 31–August 2, 2017, Proceedings, pp. 121–132 (2017)
4. Bodlaender, H.: A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci. **209**(1–2), 1–45 (1998)
5. Bodlaender, H.L., Cygan, M., Kratsch, S., Nederlof, J.: Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. Inform. Comput. **243**, 86–111 (2015)
6. Borie, R.B., Parker, R.G., Tovey, C.A.: Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. Algorithmica **7**(5&6), 555–581 (1992)
7. Bui-Xuan, B.-M., Telle, J.A., Vatshelle, M.: Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. Theor. Comput. Sci. **511**, 66–76 (2013)
8. Courcelle, B.: The monadic second-order logic of graphs IV: definability properties of equational graphs. Ann. Pure Appl. Log. **49**(3), 193–255 (1990)
9. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic. Volume 138 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge (2012). A language-theoretic approach, with a foreword by Maurice Nivat
10. Courcelle, B., Engelfriet, J., Rozenberg, G.: Handle-rewriting hypergraph grammars. J. Comput. Syst. Sci. **46**(2), 218–270 (1993)
11. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000)
12. Courcelle, B., Mosbah, M.: Monadic second-order evaluations on tree-decomposable graphs. Theor. Comput. Sci. **109**(1–2), 49–82 (1993)
13. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discrete Appl. Math. **101**(1–3), 77–114 (2000)
14. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time (extended abstract). In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011, pp. 150–159. IEEE Computer Society, Los Alamitos, CA (2011)
15. Diestel, R.: Graph Theory. Number 173 in Graduate Texts in Mathematics, 3rd edn. Springer, Berlin (2005)
16. Espelage, W., Gurski, F., Wanke, E.: How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In: Graph-theoretic concepts in computer science (Boltenhagen, 2001). Volume 2204 of Lecture Notes in Computer Science, pp. 117–128. Springer, Berlin (2001)
17. Fellows, M.R., Rosamond, F.A., Rotics, U., Szeider, S.: Clique-width is NP-complete. SIAM J. Discrete Math. **23**(2), 909–939 (2009)
18. Fleischner, H.: Eulerian Graphs and Related Topics, vol. 1. Elsevier, Amsterdam (1990)
19. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S.: Intractability of clique-width parameterizations. SIAM J. Comput. **39**(5), 1941–1956 (2010)
20. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S.: Almost optimal lower bounds for problems parameterized by clique-width. SIAM J. Comput. **43**(5), 1541–1563 (2014)
21. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S., Zehavi, M.: Clique-width III: hamiltonian cycle and the odd case of graph coloring. ACM Trans. Algorithms **15**(1), 9:1–9:27 (2019)

22. Ganian, R., Hliněný, P., Obdržálek, J.: Clique-width: when hard does not mean impossible. In: 28th International Symposium on Theoretical Aspects of Computer Science. Volume 9 of LIPIcs Leibniz International Proceedings in Informatics, pp. 404–415. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern (2011)
23. Gurski, F.: A comparison of two approaches for polynomial time algorithms computing basic graph parameters. CoRR (2008). arXiv:0806.4073
24. Hliněný, P., Oum, S.: Finding branch-decompositions and rank-decompositions. SIAM J. Comput. **38**(3), 1012–1032 (2008)
25. Kobler, D., Rotics, U.: Edge dominating set and colorings on graphs with fixed clique-width. Discrete Appl. Math. **126**(2–3), 197–221 (2003)
26. Kotzig, A.: Moves without forbidden transitions in a graph. Matematickỳ časopis **18**(1), 76–80 (1968)
27. Oum, S., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theory Ser. B **96**(4), 514–528 (2006)
28. van Rooij, J.M.M., Bodlaender, H.L., Rossmanith, P.: Dynamic programming on tree decompositions using generalised fast subset convolution. In: Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7–9, 2009. Proceedings, pp. 566–577 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.