



HAL
open science

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

Benjamin Bergougnoux, Mamadou Moustapha Kanté, O-Joung Kwon

► **To cite this version:**

Benjamin Bergougnoux, Mamadou Moustapha Kanté, O-Joung Kwon. An Optimal XP Algorithm for Hamiltonian Cycle on Graphs of Bounded Clique-Width.. 2018. hal-01858180

HAL Id: hal-01858180

<https://hal.science/hal-01858180>

Preprint submitted on 20 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AN OPTIMAL XP ALGORITHM FOR HAMILTONIAN CYCLE ON GRAPHS OF BOUNDED CLIQUE-WIDTH

BENJAMIN BERGOUGNOUX, MAMADOU MOUSTAPHA KANTÉ, AND O-JOUNG KWON

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. (WG 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 $n^{o(k)}$ (SIAM. J. Computing 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.

1. INTRODUCTION

Tree-width is one of the most well-studied graph parameters in the graph algorithm community, due partly to its numerous structural and algorithmic properties (see the survey [3]). A celebrated algorithmic meta-theorem by Courcelle [6] states that every graph problem expressible in monadic second-order logic (MSO₂) can be decided in linear time on graphs of bounded tree-width. Among the problems expressible in MSO₂, there are some well-studied and 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 [8] and the terminology was first introduced by Courcelle and Olariu [10] (see also the book [7]).

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 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. [14] showed that computing clique-width is NP-hard. For a fixed k , the best known approximation algorithm is due to Hliněný and Oum [20]; 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 .

Date: August 15, 2018.

An extended abstract appeared in Algorithms and Data Structures, WADS 2017 [2]. 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), and also supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. NRF-2018R1D1A1B07050294).

Clique-width inherits most of the algorithmic properties of tree-width. Courcelle, Makowsky, and Rotics [9] extended the meta-theorem of Courcelle [6] 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 a clique-width k -expression of it is given.

For MSO_1 problems, clique-width and tree-width have sensibly the same behavior. Indeed, many problems expressible in MSO_1 that admit a $2^{O(k)} \cdot n^{O(1)}$ -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. This includes famous problems like MINIMUM DOMINATING SET and MINIMUM VERTEX COVER [5, 19, 24], or even their connected variants and FEEDBACK VERTEX SET [1, 4, 11].

On the other hand, some fundamental problems are known to be expressible in MSO_2 but not in MSO_1 ; well known examples are MAX-CUT, EDGE DOMINATING SET (EDS), GRAPH COLORING (GC), and HAMILTONIAN CYCLE (HC). 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, Lokshtanov, and Saurabh [15] 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 $W[1] = \text{FPT}$. In 2014, the same authors [16] 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 [16], the authors state (without proving) a lower bound of $f(k) \cdot n^{o(k)}$ for HC under ETH and they left open the 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 $n^{O(k^2)}$ [13] and $n^{O(2^k)}$ [21]. Recently, Fomin et al. [18] 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 1.1. *There exists an algorithm that, given an n -vertex graph G and a k -expression of G , solves HAMILTONIAN CYCLE in time $O(n^{2k+4} \cdot 2^{2k(\log_2(k)+1)} \cdot k^3 \log_2(nk))$.*

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, each partial solution of our algorithm consists of edges which induce a collection of paths. As in [13], 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 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, Gurski, and Wanke [13].

To obtain our $n^{O(k)}$ -time algorithm, we prove that two partial solutions are equivalent if their auxiliary multigraphs have the same connected components, and the paths they induce have the

same number of endpoints labeled i , for all labels i . The running time of our algorithm follows from the fact that there are at most $n^{O(k)} \cdot 2^{O(k \log_2 k)}$ non-equivalent partial solutions.

This equivalence is related to the existence of a two-colored Eulerian trail in multigraphs. Let H be a k -labeled graph arising in the k -expression of G , and let \mathcal{P} be a partial solution, which is a disjoint union of paths that span all vertices of H . Let \mathcal{Q} be a disjoint union of paths such that the edges of each path in \mathcal{Q} belong to $E(G) \setminus E(H)$. Then, it is possible that $\mathcal{P} \cup \mathcal{Q}$ forms a Hamiltonian cycle in G . In order to test whether $\mathcal{P} \cup \mathcal{Q}$ forms a Hamiltonian cycle, we create an auxiliary multigraph by contracting each label class of H by one vertex, see Figures 2 and 3. Informally, the vertex set of the auxiliary multigraph corresponds to the label classes of H , and the edges correspond to either maximal paths in \mathcal{P} , or to paths in \mathcal{Q} that have their end-vertices in common with the paths in \mathcal{P} .

We consider, in the auxiliary multigraph, the edges corresponding to paths in \mathcal{P} as red edges and the edges from paths in \mathcal{Q} as blue edges. Now, if $\mathcal{P} \cup \mathcal{Q}$ forms a Hamiltonian cycle, then it should use paths of \mathcal{P} and paths of \mathcal{Q} that are between end-vertices of paths in \mathcal{P} , alternately. So, we can observe that $\mathcal{P} \cup \mathcal{Q}$ forms a Hamiltonian cycle in G if and only if there exists a red-blue alternating Eulerian trail in the auxiliary multigraph. We deduce the equivalence by proving that a multigraph contains a red-blue alternating Eulerian trail if and only if it is connected and each vertex is incident with the same number of red and blue edges.

Using this equivalence between Hamiltonian cycles and red-blue alternating Eulerian trails, we prove that two partial solutions \mathcal{P} and \mathcal{P}' of H are equivalent whenever their associated auxiliary multigraphs have the same set of connected components, and also same degree sequence. Since, the number of partitions of a k -size set is at most k^k and the size of 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 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 Section 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 Section 3. In Section 4, we prove the equivalence between the existence of Hamiltonian cycles in the input graph and the existence of red-blue alternating 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 Section 5, we show how to obtain from the results of Section 4 an $n^{O(k)}$ -time algorithm for HAMILTONIAN CYCLE. In Section 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 Section 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 $[n] := \{1, 2, \dots, n\}$.

Graph. We essentially follow Diestel's book [12] 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 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 $(V(G), E(G) \setminus F)$. For an edge e of G , we simply write $G - e$ for $G - \{e\}$. The *degree* of a vertex x , denoted by $\deg_G(x)$, is the number of edges incident with x . The set of vertices adjacent to a vertex v is denoted $N_G(x)$. A *cut-edge* in a connected graph G is an edge e such that $G - e$ is disconnected. A graph is *non-trivial* if it contains an edge, otherwise it is said *trivial*.

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 $\text{mult}_G : E(G) \rightarrow V(G) \cup [V(G)]^2$, 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 $\text{mult}_G(e) = \{x, y\}$ (or $\text{mult}_G(e) = \{x\}$), then we denote these distinct edges by $\{x, y\}_1, \dots, \{x, y\}_k$ (or $\{x\}_1, \dots, \{x\}_k$).

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

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

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 first and last vertices are the same. Moreover, when the edges of a graph are colored in red or blue, we say that a walk $W = (v_1, e_1, \dots, v_{r-1}, e_{r-1}, v_r)$ is a *red-blue walk*, if for every $i \in \{1, \dots, r-2\}$, the colors of e_i and e_{i+1} are different and the colors of e_1 and e_{r-1} are different, if the walk is closed. We adapt all the notations to multigraphs as well.

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

For convenience sometimes, we use the shortcut $(e_1, e_2, \dots, e_{r-1})$ for a walk $(v_1, e_1, \dots, v_{r-1}, e_{r-1}, v_r)$. This is possible because an edge always determine the two vertices with which it is incident.

A *path* of a graph is a walk where each vertex is used at most once and its end-vertices are distinct. A *cycle* of a graph is a closed walk where each vertex is used at most once. An *Eulerian trail* in a graph is a closed trail containing all edges.

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 $\text{lab}_G^{-1}(i)$ the set of vertices in G with label i . The notion of clique-width is defined by Courcelle and Olariu [10] and is based on the following operations:

- creating a graph, denoted by $i(x)$, with a single vertex x labeled with $i \in \mathbb{N}$;
- for a labeled graph G and distinct labels $i, j \in [k]$, relabeling the vertices of G with label i to j , denoted by $\rho_{i \rightarrow j}(G)$;
- 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

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

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 $(\text{val}(\phi), \text{lab}_{\text{val}(\phi)})$. The *clique-width* of a graph G , denoted by $\text{cw}(G)$, is the minimum k such that G is isomorphic to $\text{val}(\phi)$ 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 [10].

It is worth noticing, from the recursive definition of k -expressions, that one can compute in time linear in $|\phi|$ the labeling function $\text{lab}_{\text{val}(\phi)}$ of $\text{val}(\phi)$, 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 Figure 1.

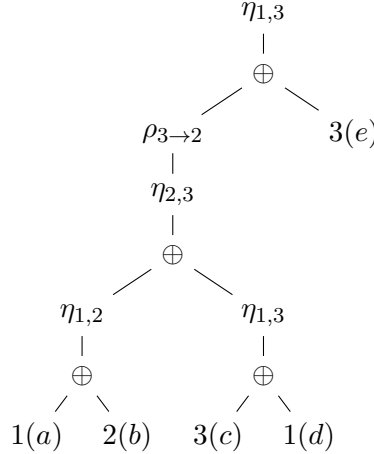


FIGURE 1. An irredundant 3-expression of C_5 .

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

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

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

A k -expression ϕ is called *irredundant* if for every subexpression $\eta_{i,j}(\phi')$ of ϕ , there are no edges in $\text{val}(\phi')$ between the vertices labeled i and the vertices labeled j . Courcelle and Olariu [10] 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 Section 4.

Lemma 2.1. *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 $\text{lab}_H(u) = i$ and $\text{lab}_H(v) = j$, we have the following.*

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

Proof. (1) Assume that $i = j$. Let ϕ' be the subexpression of ϕ such that $H = \text{val}(\phi')$. As u and v have the same label in H , for 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 $N_G(u) \cap (V(G) \setminus V(H)) = N_G(v) \cap (V(G) \setminus V(H))$.

(2) If $uv \in E(G)$ and $i = j$, then we have $uv \in E(H)$ as the operation $\eta_{a,b}$ add edges only between vertices with distinct labels. Thus, if $uv \in E(G) \setminus E(H)$, then we have $i \neq j$.

Assume that $uv \in E(G) \setminus E(H)$. Let ϕ' be the minimal (size wise) subexpression of ϕ such that $uv \in E(\text{val}(\phi'))$. It follows that $\phi' = \eta_{a,b}(\phi^*)$, with $\phi^* \in \text{Sub}(\phi)$, $\text{lab}_{\text{val}(\phi^*)}(u) = a$ and $\text{lab}_{\text{val}(\phi^*)}(v) = b$. Let $D := \text{val}(\phi^*)$. 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 $x \in \text{lab}_H^{-1}(i)$ and $y \in \text{lab}_H^{-1}(j)$, we have $xy \notin E(H)$ and $xy \in E(G)$. \square

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 at most one; in other words, there is no path 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 $E(G) \setminus E(H)$ 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 $V(G) \setminus V(H)$, the degree of v in $G|_{\mathcal{Q}}$ is two. We denote by $\bar{\Pi}(H)$ 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 $V(G) \setminus V(H)$. 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 Figure 2. Note that if G has a Hamiltonian cycle C , then $E(C) \cap E(H)$ is a partial solution and $E(C) \cap (E(G) \setminus E(H))$ 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 .

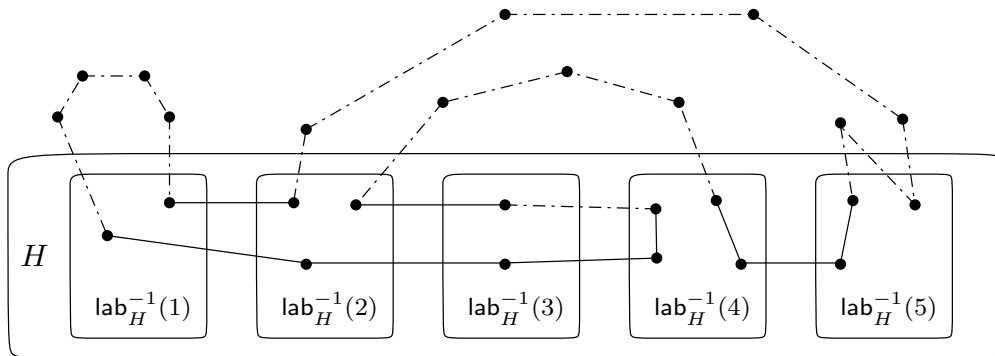


FIGURE 2. Examples of a partial solution \mathcal{P} (solid lines) and a complement solution \mathcal{Q} (dashed lines) forming a Hamiltonian cycle. Observe that $H|_{\mathcal{P}}$ contains 5 maximal paths and $G|_{\mathcal{Q}}$ contains 5 H -paths (and only 4 maximal paths).

Auxiliary Multigraph. For $\mathcal{P} \in \Pi(H) \cup \bar{\Pi}(H)$ 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 $\text{aux}_H(\mathcal{P})$, as the multigraph with vertex set $\{v_1, \dots, v_k\}$ and edge set

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

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

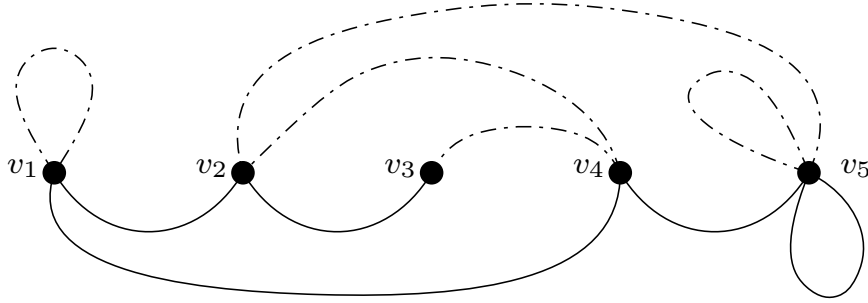


FIGURE 3. The union $G_1 \uplus G_2$ of auxiliary multigraphs G_1 and G_2 associated with the partial solution (solid lines) and the complement solution (dashed lines) represented in Figure 2.

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

Observe that \simeq is an equivalence relation. For a set $\mathcal{A} \subseteq \Pi(H)$, we define $\text{reduce}_H(\mathcal{A})$ as the operation which returns a set containing one representative 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 from the following lemma.

Lemma 4.2. For every $\mathcal{A} \subseteq \Pi(H)$, we have $\text{reduce}_H(\mathcal{A}) \leq n^k \cdot 2^{k(\log_2(k)+1)}$ and we can moreover compute $\text{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk^2 \log_2(nk))$.

Proof. To prove that $\text{reduce}_H(\mathcal{A}) \leq n^k \cdot 2^{k(\log_2(k)+1)}$, 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 $\sum_{i \in [k]} \text{deg}_{\text{aux}_H(\mathcal{P})}(v_i) \leq 2|V(H)|$. First observe that $\sum_{i \in [k]} \text{deg}_{\text{aux}_H(\mathcal{P})}(v_i) = 2|V(H)|$ when $\mathcal{P} = \emptyset$, since each isolated vertex in $H_{|\mathcal{P}}$ gives a loop in $\text{aux}_H(\mathcal{P})$. Moreover, when \mathcal{P} contains an edge, removing an edge from a partial solution \mathcal{P} of H

increases $\sum_{i \in [k]} \deg_{\text{aux}_H(\mathcal{P})}(v_i)$ 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 $\sum_{i \in [k]} \deg_{\text{aux}_H(\mathcal{P})}(v_i) \leq 2|V(H)|$; in particular each vertex of $\text{aux}_H(\mathcal{P})$ has degree at most $2|V(H)|$. As $\text{aux}_H(\mathcal{P})$ contains k vertices, we deduce that there are at most $(2|V(H)|)^k \leq (2n)^k$ possible degree sequences for $\text{aux}_H(\mathcal{P})$.

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

It remains to prove that we can compute $\text{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk \log_2(nk))$. First observe that for every $\mathcal{P} \in \Pi(H)$, we can compute $\text{aux}_H(\mathcal{P})$ in time $O(nk)$. Moreover, we can also compute the degree sequence of $\text{aux}_H(\mathcal{P})$ and the connected components of $\text{aux}_H(\mathcal{P})$ 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 $\text{reduce}_H(\mathcal{A})$ in time $O(|\mathcal{A}| \cdot nk \log_2(|\text{reduce}_H(\mathcal{A})|))$. Since $\log_2(|\text{reduce}_H(\mathcal{A})|) \leq k \log_2(2nk)$, we conclude that $\text{reduce}_H(\mathcal{A})$ is computable in time $O(|\mathcal{A}| \cdot nk^2 \log_2(nk))$. \square

The rest of this section is dedicated to prove that for a set of partial solutions \mathcal{A} of H , the set $\text{reduce}_H(\mathcal{A})$ is equivalent to \mathcal{A} , *i.e.*, if \mathcal{A} contains a partial solution that forms a Hamiltonian cycle with a complement solution, then $\text{reduce}_H(\mathcal{A})$ 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 4.3. *If $\mathcal{P} \in \Pi(H)$ and $\mathcal{Q} \in \overline{\Pi}(H)$ form a Hamiltonian cycle, then the multigraph $\text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$ 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 $M := \text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$. From the definitions of a partial solution and of a complement solution, there is a sequence $(P_1, Q_1, \dots, P_\ell, Q_\ell)$ of paths in \mathcal{P} and \mathcal{Q} such that

- P_1, P_2, \dots, P_ℓ are all the maximal paths in $H|_{\mathcal{P}}$,
- Q_1, Q_2, \dots, Q_ℓ are all the H -paths in $G|_{\mathcal{Q}}$,
- $P_1, Q_1, \dots, P_\ell, Q_\ell$ appear in C in this order,
- for each $x \in [\ell]$, the two end-vertices of P_x are the same as one end-vertex of Q_{x-1} and one 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 $\text{mult}_M(e_x) = \{v_i, v_j\}$ if $i \neq j$ or $\text{mult}_M(e_x) = \{v_i\}$ if $i = j$ such that the edges e_1, \dots, e_ℓ are pairwise different and $E(\text{aux}_H(\mathcal{P})) = \{e_1, \dots, e_\ell\}$. 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 $\text{mult}_M(f_y) = \{v_i, v_j\}$ if $i \neq j$ or $\text{mult}_M(f_y) = \{v_i\}$ if $i = j$ such that the edges f_1, \dots, f_ℓ are pairwise different and $E(\text{aux}_H(\mathcal{Q})) = \{f_1, \dots, f_\ell\}$. It is not difficult to see that $(e_1, f_1, \dots, e_\ell, f_\ell)$ is a red-blue Eulerian trail of $\text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$. \square

Next, we prove the other direction. We use the properties of an irredundant k -expression described in Lemma 2.1.

Lemma 4.4. *Let $\mathcal{P} \in \Pi(H)$. If there exists a complement solution \mathcal{Q} of H such that $\text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$ admits a red-blue Eulerian trail, then there exists $\mathcal{Q}^* \in \overline{\Pi}(H)$ such that \mathcal{P} and \mathcal{Q}^* form a Hamiltonian cycle.*

Proof. Let $T = (v_{a_1}, r_1, v_{c_1}, b_1, v_{a_2}, r_2, v_{c_2}, \dots, v_{a_\ell}, r_\ell, v_{c_\ell}, b_\ell, v_{a_1})$ be a red-blue Eulerian trail of $\text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$ with $r_1, \dots, r_\ell \in E(\text{aux}_H(\mathcal{P}))$ and $b_1, \dots, b_\ell \in E(\text{aux}_H(\mathcal{Q}))$. 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 $P_1, \dots, P_\ell, Q_1, \dots, Q_\ell$ are all pairwise distinct.

For every $i \in [\ell]$, we construct from Q_i an H -path Q_i^* 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 $xy \in E(G) \setminus E(H)$. By Lemma 2.1, we have $uv \in E(G) \setminus E(H)$. In this case, we define $Q_i^* = (u, uv, v)$.
- Assume now that $Q_i = (x, xy, y, \dots, w, wz, z)$ with $w, y \in V(G) \setminus V(H)$. Since x and u have the same label in H , by Lemma 2.1, we have $N_G(x) \setminus V(H) = N_G(u) \setminus V(H)$. Hence, u is also adjacent to y . Symmetrically, we have v is adjacent to w . In this case, we define $Q_i^* = (u, uy, y, \dots, w, wv, v)$, *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^* 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

$$P_1 - Q_1^* - \dots - P_\ell - Q_\ell^*$$

is a Hamiltonian cycle.

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

It is well known that a connected multigraph contains an Eulerian trail if and only if every vertex has even degree. The following lemma shows that a connected two-edge colored multigraph contains a red-blue Eulerian trail if and only if each vertex is incident with the same number of edges for both colors.

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 $\text{rdeg}_G(v)$ and $\text{bdeg}_G(v)$ 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 4.5. *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 , $\text{bdeg}_G(v) = \text{rdeg}_G(v)$.*

Proof. One easily checks that if G has a red-blue Eulerian trail, then for every vertex v , $\text{bdeg}_G(v) = \text{rdeg}_G(v)$. Indeed, if $T = (v_1, e_1, v_2, \dots, v_\ell, e_\ell, v_1)$ is a red-blue Eulerian trail, then for $2 \leq i \leq \ell$, e_{i-1} and e_i have different colors, and because 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 m be the number of edges in G . If $m = 0$, then G contains exactly one vertex v , because it is connected, and (v) is a red-blue Eulerian trail of G . Assume now that $m > 0$ and for every vertex v , $\text{bdeg}_G(v) = \text{rdeg}_G(v)$. Observe that we have $m > 2$.

We choose a red-blue trail $T := (v_1, e_1, v_2, e_2, \dots, v_i, e_i, v_{i+1})$ with maximum i such that

- (1) e_1 is colored red, and
- (2) the multigraph $G_i := G - \left(\bigcup_{j \in [i]} \{e_j\} \right)$ contains exactly one non-trivial component C and this component C contains v_1 and v_{i+1} .

We first observe that such a trail T exists. We choose any red edge $e = vw$. If e is not a cut-edge, then $G - e$ is connected, and (v, e, w) is such a red-blue trail. Suppose that e is a cut-edge of G . As every vertex in G has even degree, the two connected components of $G - e$ contain, each, exactly one vertex with odd degree (the endpoints of e). But, this is not possible since the sum of degrees of vertices of a connected graph is even and thus every connected graph has an even number of vertices with odd degree.

Now, we consider the properties of $G_i = G - \left(\bigcup_{j \in [i]} \{e_j\}\right)$. Since T is a red-blue trail, we have the following:

- For each $v \in V(G) \setminus \{v_1, v_{i+1}\}$, $\text{rdeg}_{G_i}(v) = \text{bdeg}_{G_i}(v)$.
- If $v_{i+1} \neq v_1$ and e_i is colored red (resp. blue), then $\text{bdeg}_{G_i}(v_{i+1}) = \text{rdeg}_{G_i}(v_{i+1}) + 1$ (resp. $\text{rdeg}_{G_i}(v_{i+1}) = \text{bdeg}_{G_i}(v_{i+1}) + 1$) and $\text{bdeg}_{G_i}(v_1) = \text{rdeg}_{G_i}(v_1) + 1$.
- If $v_{i+1} = v_1$ and e_i is colored red (resp. blue), then $\text{bdeg}_{G_i}(v_1) = \text{rdeg}_{G_i}(v_1) + 2$ (resp. $\text{bdeg}_{G_i}(v_1) = \text{rdeg}_{G_i}(v_1)$).

We claim that $i = m - 1$. Suppose for contradiction that $i < m - 1$. Note that there are at least two edges in $E(G_i)$ because $i < m - 1$.

We observe that there is at most one cut-edge in G_i incident with v_{i+1} . Indeed, if there are two cut-edges incident with v_{i+1} , then one cut-edge f satisfies that $G_i - f$ has a connected component C that contains neither v_1 nor v_{i+1} . However, all vertices v in C satisfy $\text{rdeg}_{G_i}(v) = \text{bdeg}_{G_i}(v)$ by the assumption, and in particular, they have even degrees in G_i . Then it becomes a connected component of $G_i - f$ having exactly one vertex of odd degree, which is not possible. Therefore, there is at most one cut-edge in G_i incident with v_{i+1} .

Suppose that $v_{i+1} \neq v_1$ and e_i is colored red. The proof is symmetric when e_i is colored blue. If v_{i+1} is incident with at least two blue edges in G_i , then we can choose a blue edge that is not a cut-edge to extend the red-blue trail. Thus, we may assume that there is only one blue edge e incident with v_{i+1} and it is a cut-edge of G_i . By the condition, v_{i+1} is not adjacent to any red edge in G_i , and thus it has degree 1 in G_i . Therefore, $G_i - e$ has only one non-trivial component, and it implies that we can choose e as a next edge to extend the red-blue trail.

Suppose that $v_{i+1} = v_1$ and e_i is colored red. Since there are at least two blue edges incident with v_{i+1} in G_i , we can choose a blue edge e that is not a cut-edge.

Assume now that $v_{i+1} = v_1$ and e_i is colored blue. Note that $\text{rdeg}_{G_i}(v_1) \neq 0$ by the condition that there is at least one non-trivial connected component containing v_1 and v_{i+1} , and thus there is at least one red edge incident with v_{i+1} in G_i . We claim that there is at least one red edge among them that is not a cut-edge. Suppose for contradiction that there is only one red edge incident to v_{i+1} in G_i and it is a cut-edge in G_i . As $\text{bdeg}_{G_i}(v_1) = \text{rdeg}_{G_i}(v_1)$, there is exactly one blue edge incident to v_{i+1} in G_i . Since one of the two edges incident to v_{i+1} is not a cut-edge, the other one is also not a cut-edge because all the vertices have even degree in G_i . This is a contradiction with the assumption. Thus, we can choose a red edge e that is not a cut-edge to extend the red-blue trail.

In each case, we can extend the red-blue trail, which contradicts the maximality. Therefore $i = m - 1$, and v_{i+1} should be connected to v_1 by a blue edge $f := v_{i+1}v_1$. Thus,

$$(v_1, e_1, v_2, e_2, \dots, v_i, e_i, v_{i+1}, f, v_1)$$

is a required red-blue Eulerian trail. □

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

Definition 4.6. 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 $\text{aux}_H(\mathcal{P}_1) \uplus \mathcal{M}$ admits a red-blue Eulerian trail, there exists $\mathcal{P}_2 \in \mathcal{A}$ such that $\text{aux}_H(\mathcal{P}_2) \uplus \mathcal{M}$ 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 $\mathcal{A} \subseteq \Pi(H)$ such that $\mathcal{A} \lesssim_H \Pi(H)$ with bounded size. Indeed, by Lemmas 4.3 and 4.4, $\mathcal{A} \lesssim_H \Pi(H)$ 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 $\mathcal{P}^* \in \mathcal{A}$ and $\mathcal{Q}^* \in \overline{\Pi}(H)$ such that \mathcal{P}^* and \mathcal{Q}^* form a Hamiltonian cycle. The following lemma is the key of our algorithm.

Lemma 4.7. Let $\mathcal{A} \subseteq \Pi(H)$. Then $\text{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{A}$.

Proof. Let $\mathcal{P} \in \mathcal{A}$ and \mathcal{M} be a multigraph whose edges are colored blue such that $\text{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ admits a red-blue Eulerian trail. By definition, $\text{reduce}_H(\mathcal{A})$ contains a partial solution \mathcal{P}^* such that $\mathcal{P} \simeq \mathcal{P}^*$.

As $\text{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ contains a red-blue Eulerian trail, by Lemma 4.5, we have that

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

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

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

We conclude by Lemma 4.5, that $\text{aux}_H(\mathcal{P}^*) \uplus \mathcal{M}$ admits a red-blue Eulerian trail.

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

Lemma 4.8. *Let $\mathcal{A}, \mathcal{B} \subseteq \Pi(H)$. If $\mathcal{A} \lesssim_H \mathcal{B}$, then $\text{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{B}$.*

Proof. One easily checks that \lesssim_H is a transitive relation. Now, assuming that $\mathcal{A} \lesssim_H \mathcal{B}$, we have $\text{reduce}_H(\mathcal{A}) \lesssim \mathcal{B}$ because $\text{reduce}_H(\mathcal{A}) \lesssim_H \mathcal{A}$ by Lemma 4.7. \square

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 5.1. *Let $H = \rho_{i \rightarrow j}(D)$. 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 $\bar{\Pi}(H) = \bar{\Pi}(D)$. 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 $\text{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ contains a red-blue Eulerian trail T . To prove the lemma, it is sufficient to prove that there exists $\mathcal{P}^* \in \mathcal{A}_D$ such that $\text{aux}_H(\mathcal{P}^*) \uplus \mathcal{M}$ contains a red-blue Eulerian trail.

Let f be a bijective function such that

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

By construction of $\text{aux}_D(\mathcal{P})$ and $\text{aux}_H(\mathcal{P})$, 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 $\text{aux}_D(\mathcal{P})$ 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 $\text{aux}_D(\mathcal{P})$, 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 $\text{aux}_D(\mathcal{P}) \uplus \mathcal{M}'$. Since $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists $\mathcal{P}^* \in \mathcal{A}_D$ such that $\text{aux}_D(\mathcal{P}^*) \uplus \mathcal{M}'$ contains a red-blue Eulerian trail. Observe that $\text{aux}_H(\mathcal{P})$ (respectively \mathcal{M}) is obtained from $\text{aux}_D(\mathcal{P}^*)$ (resp. \mathcal{M}') by replacing each edge associated with $\{v_i, v_k\}$ or $\{v_i\}$ in $\text{aux}_D(\mathcal{P}^*)$ (resp. \mathcal{M}') with an edge associated with $\{v_j, v_k\}$ or $\{v_j\}$ respectively. We conclude that $\text{aux}_H(\mathcal{P}^*) \uplus \mathcal{M}$ admits a red-blue Eulerian trail. \square

Lemma 5.2. *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 $\text{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) = \text{aux}_H(\mathcal{P}_D) \uplus \text{aux}_H(\mathcal{P}_F)$. 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 $\text{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) \uplus \mathcal{M}$. It is sufficient to prove that there exist $\mathcal{P}_D^* \in \mathcal{A}_D$ and $\mathcal{P}_F^* \in \mathcal{A}_F$ such that $\text{aux}_H(\mathcal{P}_D^* \cup \mathcal{P}_F^*) \uplus \mathcal{M}$ admits a red-blue Eulerian trail.

We begin by proving that there exists $\mathcal{P}_D^* \in \mathcal{A}_D$ such that $\text{aux}_H(\mathcal{P}_D^* \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains a red-blue Eulerian trail. In order to do so, we construct from $\text{aux}_H(\mathcal{P}_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ 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 $\text{aux}_H(\mathcal{P}_F) \uplus \mathcal{M}$, remove these edges and add a blue edge between the two end-vertices of W . Obviously, for every $\mathcal{P}'_D \in \Pi(D)$, if $\text{aux}_D(\mathcal{P}'_D) \uplus \mathcal{M}'$ admits a red-blue Eulerian trail, then $\text{aux}_H(\mathcal{P}'_D \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains a red-blue Eulerian trail also.

By construction of \mathcal{M}' , the multigraph $\text{aux}_D(\mathcal{P}_D) \uplus \mathcal{M}'$ contains a red-blue Eulerian trail. As $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists \mathcal{P}_D^* such that $\text{aux}_D(\mathcal{P}_D^*) \uplus \mathcal{M}'$ contains a red-blue Eulerian trail. We conclude that $\text{aux}_H(\mathcal{P}_D^* \cup \mathcal{P}_F) \uplus \mathcal{M}$ contains also a red-blue Eulerian trail.

Symmetrically, we can prove that there exists $\mathcal{P}_F^* \in \mathcal{A}_F$ such that $\text{aux}_H(\mathcal{P}_D^* \cup \mathcal{P}_F^*) \uplus \mathcal{M}$ contains a red-blue Eulerian trail. This proves the lemma. \square

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 $\mathcal{E}_{i,j}^H$ the set of edges whose endpoints are labeled i and j , i.e., $\{uv \in E(H) \mid \text{lab}_H(v) = i \wedge \text{lab}_H(v) = j\}$. 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 $\mathcal{E}_{i,j}^H$. 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

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

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

Lemma 5.3. *Let $H = \eta_{i,j}(D)$ such that $E(D) \cap \mathcal{E}_{i,j}^H = \emptyset$. If $\mathcal{A}_D \lesssim_D \Pi(D)$, then we have $\mathcal{A}_D^0 \cup \dots \cup \mathcal{A}_D^n \lesssim_H \Pi(H)$.*

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

Claim 5.3.1. *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 $\text{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ admits a red-blue Eulerian trail. Take $e = vw \in \mathcal{E}_{i,j}^H$ such that $\mathcal{P}' := \mathcal{P} - e$ belongs to \mathcal{B} and $v \in \text{lab}_H^{-1}(i)$ and $w \in \text{lab}_H^{-1}(j)$. 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 $\text{aux}_H(\mathcal{P}') \uplus \mathcal{M}'$ admits a red-blue Eulerian trail. Note that there is a path $P \in \mathcal{P}$ containing the edge e , and it is divided into two maximal subpaths in $P - e$, say P_1 and P_2 . Without loss of generality, we may assume that P_1 contains v and P_2 contains w . Let v' and w' be the other end vertices of P_1 and P_2 , respectively, and let $i' := \text{lab}_H(v')$ and $j' := \text{lab}_H(w')$. Note that $\text{aux}_H(\mathcal{P}')$ can be obtained from $\text{aux}_H(\mathcal{P})$ by removing an edge $v_i v_{j'}$ and adding two edges $v_i v_{i'}$ and $v_j v_{j'}$. So, we can obtain a red-blue Eulerian trail of $\text{aux}_H(\mathcal{P}') \uplus \mathcal{M}'$ from a red-blue Eulerian trail of $\text{aux}_H(\mathcal{P}) \uplus \mathcal{M}$ by replacing $v_i v_{j'}$ with the sequence $(v_{i'}, v_{i'} v_i, v_i, v_i v_j, v_j, v_j v_{j'}, v_{j'})$ where $v_i v_j$ is the new blue edge in \mathcal{M}' . It implies the claim.

Now, since $\mathcal{A} \lesssim_H \mathcal{B}$, there exists $\mathcal{P}^* \in \mathcal{A}$ such that $\text{aux}_H(\mathcal{P}^*) \uplus \mathcal{M}'$ 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 $H|_{\mathcal{P}^*}$ 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}^* by adding the edge in $\mathcal{E}_{i,j}^H$ 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 $\text{aux}_H(\widehat{\mathcal{P}}) \uplus \mathcal{M}$ admits a red-blue Eulerian trail. We conclude that $\mathcal{A} + (i, j) \lesssim_H \mathcal{B} + (i, j)$. \square

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 $\mathcal{E}_{i,j}^H$ of size at most n , we have $\Pi(H) = \bigcup_{0 \leq t \leq n} (\Pi(D) + t(i, j))$.

Since $V(D) = V(H)$ and $E(D) \subseteq E(H)$, we have $\mathcal{A}_D^0 = \mathcal{A}_D \lesssim_H \Pi(D) + 0(i, j)$. Let $\ell \in \{1, \dots, n\}$ and suppose that $\mathcal{A}_D^{\ell-1} \lesssim_H \Pi(D) + (\ell-1)(i, j)$. From Claim 5.3.1, we have $\mathcal{A}_D^{\ell-1} + (i, j) \lesssim_H \Pi(D) + \ell(i, j)$. By Lemma 4.8, we deduce that $\mathcal{A}_D^\ell = \text{reduce}(\mathcal{A}_D^{\ell-1} + (i, j)) \lesssim_H \Pi(D) + \ell(i, j)$.

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

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

Theorem 5.4. *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. We assume that G is connected and it has at least 3 vertices, otherwise we can automatically say that it is a NO-instance. 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 $\mathcal{A}_H := \text{reduce}_H(\mathcal{A}_D \otimes \mathcal{A}_F)$.
- If $H = \eta_{i,j}(D)$, then we set $\mathcal{A}_H := \text{reduce}_H(\mathcal{A}_D^0 \cup \dots \cup \mathcal{A}_D^n)$.

For the last three cases, we deduce, by induction, from Lemma 4.8 and Lemmas 5.1–5.3 that $\mathcal{A}_H \lesssim_H \Pi(H)$. Moreover by the use of the function reduce_H , by Lemma 4.2, 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 $\ell \in [k] \setminus \{i, j\}$, we have $\text{deg}_{\text{aux}_D(\mathcal{P})}(v_\ell) = 0$ and $\text{deg}_{\text{aux}_H(\mathcal{P})}(v_i) = \text{deg}_{\text{aux}_H(\mathcal{P})}(v_j) > 0$.

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 $i, j \in [k]$,
- $E(C) \setminus E(D) \neq \emptyset$.

It follows that $E(C) \setminus E(D) \subseteq \mathcal{E}_{i,j}^H$. 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 4.3, the multigraph $\text{aux}_D(\mathcal{P}) \uplus \text{aux}_D(\mathcal{Q})$ contains a red-blue Eulerian trail. Since $\mathcal{A}_D \lesssim_D \Pi(D)$, there exists $\mathcal{P}^* \in \mathcal{A}_D$ such that $\text{aux}_D(\mathcal{P}^*) \uplus \text{aux}_D(\mathcal{Q})$ contains a red-blue Eulerian trail. As $\mathcal{Q} \subseteq \mathcal{E}_{i,j}^H$, we deduce that for every $\ell \in [k] \setminus \{i, j\}$, we have $\text{deg}_{\text{aux}_D(\mathcal{P}^*)}(v_\ell) = 0$ and $\text{deg}_{\text{aux}_H(\mathcal{P}^*)}(v_i) = \text{deg}_{\text{aux}_H(\mathcal{P}^*)}(v_j)$.

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 $\deg_{\text{aux}_H(\mathcal{P})}(v_i)$ many edges between the set of vertices labeled i and the set of vertices labeled j . Observe that $\text{aux}_H(\mathcal{Q})$ consists of $\deg_{\text{aux}_H(\mathcal{P})}(v_i)$ many edges between v_i and v_j . Therefore, by Lemma 4.5, $\text{aux}_H(\mathcal{P}) \uplus \text{aux}_H(\mathcal{Q})$ admits a red-blue Eulerian trail. Then by Lemma 4.4, there exists $\mathcal{Q}^* \in \overline{\Pi}(H)$ such that \mathcal{P} and \mathcal{Q}^* 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 $H = \rho_{i \rightarrow j}(D)$, then we compute \mathcal{A}_H in time $O(1)$.

By Lemma 4.2, for every $\mathcal{A} \subseteq \Pi(H)$, we can compute $\text{reduce}_H(\mathcal{A})$ 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 $\mathcal{A}_H := \text{reduce}_H(\mathcal{A}_D \otimes \mathcal{A}_F)$ 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 4.2, for every $\ell \in \{0, \dots, n-1\}$, 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 $\mathcal{A}_D^1, \dots, \mathcal{A}_D^n$ in time $O(n^{k+4} \cdot 2^{k(\log_2(k)+1)} \cdot k^2 \log_2(nk))$. The running time to compute \mathcal{A}_H from \mathcal{A}_D follows from $|\mathcal{A}_D^0 \cup \dots \cup \mathcal{A}_D^n| \leq n^{k+1} \cdot 2^{2k(\log_2(k)+1)}$.

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

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

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 [8], 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 edges from vertices with label i to vertices with label j (so we do not add edges 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 the same approach, we can show the following.

Theorem 6.1. *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 $n^{O(k)}$.*

Similar to the proof of Theorem 5.4, we recursively compute a set \mathcal{A}_H of bounded size such that \mathcal{A}_H represents the set of all partial solutions of H , for every k -labeled graph H arising in the k -expression of G . For DIRECTED HAMILTONIAN CYCLE, auxiliary graphs should be directed graphs. Thus, to adapt our idea for undirected graphs, we need to prove the directed version of Lemma 4.5, and prove auxiliary lemmas.

Let G be a directed 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 define that

- $\text{rdeg}_G^+(v)$ and $\text{rdeg}_G^-(v)$ are respectively the outdegree and indegree of v in $G|_R$,
- $\text{bdeg}_G^+(v)$ and $\text{bdeg}_G^-(v)$ are respectively the outdegree and indegree of v in $G|_B$.

Lemma 6.2. *Let G be a connected directed multigraph whose edges are colored red or blue. Then G has a red-blue Eulerian trail if and only if for every vertex v , $\text{bdeg}_G^+(v) = \text{rdeg}_G^-(v)$ and $\text{bdeg}_G^-(v) = \text{rdeg}_G^+(v)$.*

Proof. The walks here are all directed. The idea for proving Lemma 6.2 is the same as the one of Lemma 4.5. If $T = (v_1, e_1, v_2, \dots, v_\ell, e_\ell, v_1)$ is a red-blue Eulerian trail, then for $2 \leq i \leq \ell$, e_{i-1} and e_i have different colors, and moreover e_1 and e_ℓ have distinct colors. Thus, we can conclude that

- for every vertex v , $\text{bdeg}_G^+(v) = \text{rdeg}_G^-(v)$ and $\text{bdeg}_G^-(v) = \text{rdeg}_G^+(v)$.

For the other direction, suppose that for every vertex v , $\text{bdeg}_G^+(v) = \text{rdeg}_G^-(v)$ and $\text{bdeg}_G^-(v) = \text{rdeg}_G^+(v)$. Since each vertex has equal indegree and outdegree, the graph contains an Eulerian trail (not necessarily a red-blue one), and it implies that G is strongly connected.

We choose a red-blue trail $T := (v_1, e_1, v_2, e_2, \dots, v_i, e_i, v_{i+1})$ with maximum i such that

(1) e_1 is colored red, and

(2) the multigraph $G_i := G - \left(\bigcup_{j \in [i]} \{e_j\} \right)$ contains strongly connected components C_1, C_2, \dots, C_t ,

where at least one is non-trivial, and such that

- $v_{i+1} \in V(C_1)$ and $v_1 \in V(C_t)$,
- for each $x \in [t-1]$, there is exactly one edge from C_x to C_{x+1} and there are no other edges between distinct strongly connected components in C_1, C_2, \dots, C_t .

We first observe that such a trail T exists. We choose any red edge $e = (v, w)$. If $G - e$ is strongly connected, then we are done. Otherwise, let C_1, C_2, \dots, C_t be the strongly connected components of $G - e$ which form an acyclic ordering; that is, there is an edge from C_i to C_j with $i \neq j$ if and only if $i < j$. Note that C_1 contains only outgoing edges in $G - e$. If C_1 does not contain v and w , then each vertex of C_1 has equal indegree and outdegree, and it is a contradiction. Since v also contributes an outdegree in the edge vw , we can deduce that w is contained in C_1 . In the similar reason, v is contained in C_t .

Note that every vertex of C_1 has same indegree and outdegree in G . Thus, there is exactly one outgoing edge from C_1 to $C_2 \cup \dots \cup C_t$. If there is an outgoing edge from C_1 to a component other than C_2 , then C_2 have only outgoing edges to $C_3 \cup \dots \cup C_t$, which is impossible. Thus, there is exactly one edge from C_1 to C_2 . The same argument can be repeated, and we can deduce that for each $x \in [t-1]$, there is exactly one edge from C_x to C_{x+1} and there are no other edges between distinct strongly connected components in C_1, C_2, \dots, C_t . Therefore (v, e, w) is a required red-blue trail.

Now, we consider the properties of $G_i = G - \left(\bigcup_{j \in [i]} \{e_j\} \right)$. Since T is a red-blue trail, we have the following:

- For each $v \in V(G) \setminus \{v_1, v_{i+1}\}$, $\text{bdeg}_{G_i}^+(v) = \text{rdeg}_{G_i}^-(v)$ and $\text{bdeg}_{G_i}^-(v) = \text{rdeg}_{G_i}^+(v)$.
- If $v_{i+1} \neq v_1$ and e_i is colored red (resp. blue), then
 - $\text{bdeg}_{G_i}^+(v_{i+1}) = \text{rdeg}_{G_i}^-(v_{i+1}) + 1$ (resp. $\text{rdeg}_{G_i}^+(v_{i+1}) = \text{bdeg}_{G_i}^-(v_{i+1}) + 1$),
 - $\text{bdeg}_{G_i}^-(v_{i+1}) = \text{rdeg}_{G_i}^+(v_{i+1})$ (resp. $\text{rdeg}_{G_i}^-(v_{i+1}) = \text{bdeg}_{G_i}^+(v_{i+1})$),

- $\text{bdeg}_{G_i}^-(v_1) = \text{rdeg}_{G_i}^+(v_1) + 1$ and $\text{bdeg}_{G_i}^+(v_1) = \text{rdeg}_{G_i}^-(v_1)$.
- If $v_{i+1} = v_1$ and e_i is colored red (resp. blue), then
 - $\text{bdeg}_{G_i}^-(v_1) = \text{rdeg}_{G_i}^+(v_1) + 1$ (resp. $\text{bdeg}_{G_i}^-(v_1) = \text{rdeg}_{G_i}^+(v_1)$),
 - $\text{bdeg}_{G_i}^+(v_1) = \text{rdeg}_{G_i}^-(v_1) + 1$ (resp. $\text{bdeg}_{G_i}^+(v_1) = \text{rdeg}_{G_i}^-(v_1)$).

We claim that $i = m - 1$. Suppose for contradiction that $i < m - 1$. Let C_1, C_2, \dots, C_t be the strongly connected components stated in the second condition. We divide cases depending on whether $t = 1$ or not.

(Case 1. $t = 1$.) In this case, C_1 is strongly connected. So we can simply extend T by choosing a next edge which has the color differ from the color of e_i .

(Case 2. $t \geq 2$.) If C_1 consists of one vertex, then there is a unique edge from v_1 to C_2 in G_i which color differs from the color of e_i , and we can choose this edge as a next edge. Suppose C_1 contains at least two vertices. Then we can choose a next edge as an edge in C_1 that has a color different from that of e_i .

In each case, we can extend the red-blue trail, which contradicts the maximality. Therefore $i = m - 1$, and v_{i+1} should be connected to v_1 by a blue edge $f := v_{i+1}v_1$. Thus,

$$(v_1, e_1, v_2, e_2, \dots, v_i, e_i, v_{i+1}, f, v_1)$$

is a required red-blue Eulerian trail. □

We define partial solutions and auxiliary multigraphs similarly at the exception that a directed path from a vertex labeled i to a vertex labeled j in a partial solution corresponds to a directed edge (v_i, v_j) in the auxiliary multigraph. Based on Lemma 6.2, the set of connected components and the degree sequence are important. 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 $n^{O(k)}$. At this point, the behavior of clique-width towards fundamental problems expressible in MSO_2 are well-understood. One major open question related to clique-width remains the complexity of its computation and its approximation within a constant factor.

We can avoid this long standing open question by using a related parameter like *rank-width*. This parameter was introduced by Oum and Seymour [23] 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, and in some graph, clique-width can be exponentially bigger than rank-width. On the other hand, rank-width is harder to manipulate and, to the best of our knowledge, there is no optimal algorithm known for rank-width under ETH, even for basic problems such as VERTEX COVER and DOMINATING SET, where the best algorithms run in time $2^{O(k^2)} \cdot n^{O(1)}$ and the best lower bound under ETH are $2^{o(k)} \cdot n^{O(1)}$.

One successful way to work with rank-width is through the notion of *d-neighbor equivalence* introduced in [5]. The d -neighbor equivalence was the key in the design of efficient algorithms to solve some well-studied and well-known problems expressible in MSO_1 such as VERTEX COVER, DOMINATING SET [5, 22], their connected variants, and FEEDBACK VERTEX SET [1]. Is this notion also useful for the design of fast XP algorithms, parameterized by rank-width/cliue-width, for problems expressible in MSO_2 but not in MSO_1 ?

We conclude with one question. A digraph D is an *out-tree* if D is an oriented tree 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 [17] 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 $O(k)$.

REFERENCES

- [1] B. Bergougnoux and M. M. Kanté. Rank based approach on graphs with structured neighborhood. *CoRR*, abs/1805.11275, 2018.
- [2] B. Bergougnoux, M. M. Kanté, and O. Kwon. An optimal xp algorithm for hamiltonian cycle on graphs of bounded clique-width. In *Algorithms and Data Structures*, pages 121–132. Springer International Publishing, 2017.
- [3] H. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.
- [4] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inform. and Comput.*, 243:86–111, 2015.
- [5] B.-M. Bui-Xuan, J. A. Telle, and M. Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theoret. Comput. Sci.*, 511:66–76, 2013.
- [6] B. Courcelle. The monadic second-order logic of graphs IV: definability properties of equational graphs. *Annals of Pure and Applied Logic*, 49(3):193–255, 1990.
- [7] B. Courcelle and J. Engelfriet. *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.
- [8] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.
- [9] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.
- [10] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.
- [11] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. 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*, pages 150–159. IEEE Computer Soc., Los Alamitos, CA, 2011.
- [12] R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, third edition, 2005.
- [13] W. Espelage, F. Gurski, and E. Wanke. 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 Comput. Sci.*, pages 117–128. Springer, Berlin, 2001.
- [14] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Clique-width is NP-complete. *SIAM J. Discrete Math.*, 23(2):909–939, 2009.
- [15] F. V. Fomin, P. A. Golovach, D. Lokshtanov, and S. Saurabh. Intractability of clique-width parameterizations. *SIAM J. Comput.*, 39(5):1941–1956, 2010.
- [16] F. V. Fomin, P. A. Golovach, D. Lokshtanov, and S. Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM J. Comput.*, 43(5):1541–1563, 2014.
- [17] R. Ganian, P. Hliněný, and J. Obdržálek. Clique-width: when hard does not mean impossible. In *28th International Symposium on Theoretical Aspects of Computer Science*, volume 9 of *LIPICs. Leibniz Int. Proc. Inform.*, pages 404–415. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011.
- [18] P. A. Golovach, D. Lokshtanov, S. Saurabh, and M. Zehavi. Cliqueswidth III: the odd case of graph coloring parameterized by cliqueswidth. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 262–273, 2018.
- [19] F. Gurski. A comparison of two approaches for polynomial time algorithms computing basic graph parameters. *CoRR*, abs/0806.4073, 2008.
- [20] P. Hliněný and S. Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008.
- [21] D. Kobler and U. Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.

- [22] S. Oum, S. H. Sæther, and M. Vatshelle. Faster algorithms for vertex partitioning problems parameterized by clique-width. *Theoret. Comput. Sci.*, 535:16–24, 2014.
- [23] S. Oum and P. Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006.
- [24] J. M. M. van Rooij, H. L. Bodlaender, and P. Rossmanith. 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*, pages 566–577, 2009.

UNIVERSITÉ CLERMONT AUVERGNE, LIMOS, CNRS, AUBIÈRE, FRANCE
Email address: `benjamin.bergognoux@uca.fr`

Email address: `mamadou.kante@uca.fr`

DEPARTMENT OF MATHEMATICS, INCHEON NATIONAL UNIVERSITY, INCHEON, SOUTH KOREA
Email address: `ojoungkwon@gmail.com`