Relaxations of Hard Graph Problems Using Finite Groups, and Characterizations of Graphs for Efficient Algorithms

西山, 宏

Relaxations of Hard Graph Problems Using Finite Groups, and Characterizations of Graphs for Efficient Algorithms

Hiroshi Nishiyama

Abstract

It was 1971 when the notion of NP-completeness was introduced by Cook. Karp proposed 21 NP-complete problems, and more than half of them are graph problems, including Hamiltonian cycle problem, vertex cover, and chromatic number. These hard problems, however, are solved efficiently when input graphs have specific structures. Understanding graph structures is an important and prospective approach for designing efficient graph algorithms.

Relaxation of a problem is a standard strategy to attack hard problems; consider a *relaxed variant* of a hard problem, find good characterizations for it, and apply the idea of the characterizations to the original hard problem. Motivated by the development of new technologies for relaxation, this paper focuses on relaxations using *finite groups*.

The target of relaxation is not unique; relaxation of the constraints of a problem, relaxation of "for any input" condition (restricting the class of inputs), relaxation of objective function (approximation), and so on. To begin with, this thesis considers a variant of the Hamiltonian cycle problem by relaxing the constraint. We give rise to a new problem, *parity Hamiltonian cycle*, which is a relaxed variant of the Hamiltonian cycle problem. A parity Hamiltonian cycle is a closed walk (possibly using each edge more than once) which visits every vertex an odd number of times. We are concerned with the problem in both undirected graphs and directed graphs. We give some good characterizations for both undirected graphs and directed graphs which have parity Hamiltonian cycles. Based on the characterizations, we present efficient algorithms for both problems. Particularly, the existence of a parity Hamiltonian cycle in a directed graph is characterized by a linear system over GF(2), thus the problem is solved by solving the linear system. Next we consider the Hamiltonian cycle problem for *covering graphs*, which are defined by finite groups. Batagelj et al. (1982) showed a very simple characterization of the Hamiltonicity of the Cartesian product of a tree and a cycle which is represented as a covering graph. We show the same characterization as Batagelj et al.'s is applicable for two larger graph classes.

Finding a spanning tree of minimum weight with bounded diameter is known to be NP-hard. We give rise to a variant of the problem, *the odd depth tree problem.* An odd depth tree is a rooted spanning tree such that the distance of each leaf and the root is odd. We show the NP-hardness for non-bipartite graphs, while we present a complete characterization for bipartite graphs. We also consider the problem for directed graphs, and show similar characterizations and complexity results to the undirected case.

Acknowledgment

First of all the author is very grateful to my supervisor Shuji Kijima. He helped me a lot and gave me advice about my research, presentations, thesis and paperworks. I was on half-way to giving up the Ph. D. course, but he patiently waited for me while I was taking a year off, and when I was back he encouraged me to finish this great work up. Without his effort I could not hang tough to work on this thesis.

I would like to express my sincere gratitude to my ex-supervisor Masafumi Yamashita who kindly gave me advice during my undergraduate and master courses, and the first two years of my Ph. D. course. I also would like to appreciate the associated professor Yukiko Yamauchi for taking great care of me and giving advice about my research.

I am grateful to the professor Eiji Takimoto and the associate professor Naoyuki Kamiyama who cooperatively discussed my research and gave me a lot of helpful advice.

I would like to thank every my current members and ex-colleagues in my laboratory who I have met in the seven years of my life here. I appreciate their kindness which helped me a lot. I have grown in my mind in many aspects throughout the relationship with them.

Finally, I appreciate my family, especially my mother Junko for her economic and mental assistance.

Table of Contents

Chapter 1

Introduction

It was 1971 when Stephen A. Cook posed the notion of NP-completeness, and Leonid Levin independently found essentially the same notion around the same time. Since then, the conjecture P $\neq$ NP has been a major open problem in the theoretical computer science. Graph algorithms have been the central issue in the context. In 1972, Richard R. Karp presented 21 NP-complete problems, in which more than half of them are problems on graphs such as Hamiltonian cycle (HC), vertex cover, chromatic number, etc. However, those hard problems are easily solved for some graphs; the Hamiltonian cycle problem is polynomial time solvable for strongly connected path-mergeable graphs, locally semi-complete graphs [3], cocomparability graphs [17], distance hereditary graphs [32], etc. Understanding graph structures is an important and prospective approach for designing efficient graph algorithms. Then, the ultimate goal of this study in future is to find good characterizations for any problems in NP.

As an approach, it may be a natural idea to begin with considering a *relaxed variant*, of a hard problem, finding good characterizations of it, and applying the idea of the characterizations to the original hard problem. Relaxation is a standard approach to attack a hard problem. The traveling salesman problem (TSP) in a graph, is regarded as a relaxed version of the HC problem, in the sense that the condition of visiting number on each vertex is relaxed to more than once. Another example may be a two-factor (in cubic graphs), which relaxes the condition of the connectivity of an HC, but a two-factor must contain each vertex exactly once (cf. [29, 30, 8, 9]). Motivated by a development of new techniques for relaxation, this paper focuses on relaxations using *finite groups*. In the areas of graph theory and matroid theory, group-labeled graphs recently attract lots of attension, and are intensively investigated [14, 38, 39, 54]. The *parity* is regarded as a cyclic group of order two. Graph problems with parity constraints have been well studied; realizing plane graphs with prescribed parity of degrees of vertices [1], strong connected orienting of undirected graphs with parity constraints for degrees [22], edge-coloring with parity constraint [12], subset feedback set problem with parity condition [35], multi-way cut with parity constraint [44], etc.

The target of relaxation is not unique; relaxation of the constraints of a problem, relaxation of "for any input" condition (restricting the class of inputs), relaxation of objective function (approximation), and so on. In this thesis, we consider a variant of Hamiltonian cycle problem by relaxing the constraint of the Hamiltonian cycle problem. First we introduce *the parity Hamiltonian cycle problem* (PHC), which is a relaxed variant of the Hamiltonian cycle problem. A PHC is a closed walk (possibly using each edge more than once) which visits every vertex an odd number of times. We are concerned with the problem in both undirected graphs and directed graphs. Particularly, a characterization of directed graphs is given by a GF(2) linear system and the problem is easily solved using the linear system. Next we consider the Hamiltonian cycle problem restricting the class of input graphs. The HC problem for graphs possessing some symmetry such as *vertex-transitive graphs* is a major target in graph theory. This thesis is concerned with *covering graphs*, which are defined by finite groups, and gives a new characterization of Hamiltonicity. Also we will be concerned with another NP-hard problem. A spanning tree is a connected acyclic subgraph. While a spanning tree in a graph is easily found, finding the one with bounded diameter is known to be NP-hard. In this thesis we consider the *odd depth tree problem*, a variant of the above problem relaxing the constraint using parity.

1.1 Relaxation of Constraints: Parity Hamiltonian Cycle Problem

1.1.1 Undirected PHC Problem

A *Hamiltonian* cycle (HC) is a cycle which visits every vertex exactly once. The question if a given graph has a Hamiltonian cycle is a celebrated NP-complete problem due to Karp [36]. It could be a natural idea for the HC problem to relax the condition on the visiting number. The traveling salesman problem (TSP) is a celebrated optimization problem to minimize the length of a walk, where the walk must visit every vertex at least once, while an HC visits every vertex exactly once. This thesis proposes another relaxation of the HC problem using parity. The *parity Hamiltonian cycle* (*PHC*), which is a variant of the Hamiltonian cycle: a PHC is a closed walk which visits every vertex an odd number of times (see Section 2.1, for more rigorous description). Note that a closed walk is allowed to use an edge more than once. The PHC problem is to decide if a given graph has a PHC. We remark that if the condition "odd number of times" is changed to "even number of times," the problem becomes trivial: just finding a spanning tree and tracing it twice suffices.

It may not be trivial if the PHC problem is in NP, since the length of a PHC is unbounded in the problem. In Section 2.3 we show that the PHC problem is in P, in fact. Precisely, we give a complete characterization of the graphs which have PHC's.

Table: 1.1: Time complexity of the PHC problem.

Each edge is used at most z times	Complexity	Refinement
$z \geq 4$	P (Thm. 2.7)	
$z = 3$	NP-complete (Thm. 2.10)	$\Rightarrow$ Table 1.2
$z = 2$	NP-complete (Thm. 2.10)	
$z = 1$	NP-complete (Thm. 2.9)	

Table: 1.2: Time complexity of the PHC$_3$ problem.

Edge connectivity	Complexity	Refinement by graph classes
4-edge-connected	P (Thm. 2.11)	
3-edge-connected	unknown	P for P_6-free or $C_{\geq 5}$-free (Section 2.5)
2-edge-connected	NP-complete (Thm. 2.10)	
1-edge-connected	NP-complete (from 2-edge-connected case)	P for P_6-free or $C_{\geq 5}$-free (Section 2.6.1)

Furthermore, we show that if a graph has a PHC then we can find a PHC$_4$ in linear time, where PHC$_z$ for a positive integer z denotes a PHC which uses each edge at most z times. In contrast, Section 2.4 shows that the PHC$_z$ problem is NP-complete for each $z = 1, 2, 3$ (see Table 1.1)[1]. We then further investigate the PHC$_3$ problem. In precise, the PHC$_3$ problem is NP-complete for *two*-edge-connected graphs, while Section 2.4.1 shows that it is solved in polynomial time for *four*-edge-connected graphs. The complexity of the PHC$_3$ for *three*-edge-connected graphs remains unsettled (see Table 1.2).

As an approach to the PHC$_3$ problem for three-edge-connected graphs, we in Section 2.5 utilize the celebrated *ear-decomposition*, which is actually a well-known characterization of *two*-edge-connected graphs. Then, Section 2.5 shows that the PHC$_3$ problem is in P for any two-edge-connected $C_{\geq 5}$-free or P_6-free graphs (see Section 2.2 for the graph classes). The classes of $C_{\geq 5}$-free and P_6-free contain some important graph classes such as chordal, chordal bipartite and cograph. We remark that it is known that the Hamiltonian cycle is NP-complete for $C_{\geq 4}$-free graphs, as well as P_5-free graphs (cf. [10]).

In precise, we in Section 2.5 introduce a stronger notion of *all-roundness* (and *bipartite all-roundness*) of a graph, which is a sufficient condition that a graph has a PHC. Catlin [13] presented a similar notion of *collapsible* in the context of spanning Eulerian subgraphs, and the all-roundness is an extended notion of collapsible. Then, we show that any two-edge-connected $C_{\geq 5}$-free and P_6-free graphs are all-round or bipartite all-

[1]Notice that those hardness results are independent, e.g., "$z = 3$ is hard" does not imply "$z = 2$ is hard," and vice versa.

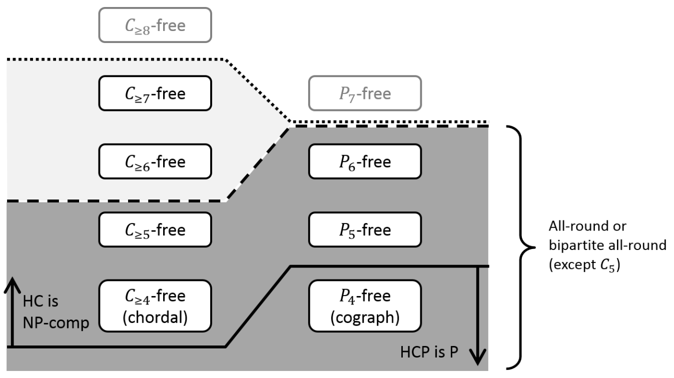


Figure: 1.1: PHC_3 is in P for $C_{\geq 5}$-free graphs and P_6-free graphs.

round. We conjecture that any two-edge-connected $C_{\geq 7}$-free graphs are all-round, while it seems not true for $C_{\geq 8}$-free nor P_7-free.

Section 2.6 is for miscellaneous discussions. Section 2.6.1 shows that the PHC_3 problem is in P for any $C_{\geq 5}$-free or P_6-free graphs, and extends the results in Section 2.5 with an extra argument on bridges. In Section 2.6.2, we remark that a dense graph is also all-round using some techniques in Section 2.5. Before closing the chapter, Section 2.6.3 briefly discusses the connection between the PHC and other problems, such as Hamiltonian cycle or Eulerian cycle, regarding a generalized problem described in Section 2.5.

Related works Here, we refer to the work by Brigham et al. [11], which investigates a similar (or, essentially the same) problem. Brigham et al. [11] showed that any connected undirected graph has a parity Hamiltonian path or cycle (see Theorem 2.8). Their proof is constructive, and they gave a linear time algorithm based on the depth-first-search. As far as we know, it is the uniquely known result on the problem. To be precise, we remark that their argument does not imply that the PHC problem is in P: their study do not consider the case when a graph does not contain a parity Hamiltonian cycle.

Notice that the condition that an HC visits each vertex once, say $1 \in \mathbb{R}$ times, is replaced by $1 \in \mathrm{GF}(2)$ times in the PHC. Modification of the field is found in some graph problems, such as group-labeled graphs or nowhere-zero flows [34, 40]. It was shown that the extension complexity of the TSP is exponential [56, 20, 21], while it is an interesting question if the PHC has an efficient (extended) formulation over GF(2). We also refer to a work on exact k-walk by Jackson and Wormald [33], which is concerned with a spanning closed walk meeting each vertex exactly k times (or at most k times in another version).

1.1.2 Directed PHC problem

It is natural to ask if the PHC problem is efficiently solved in directed graphs. We give two characterizations of graphs which admit PHC's. The characterizations, unlike with the undirected case, are described by linear systems over GF(2). Since a liner system is efficiently solved by Gaussian elimination, the characterizations directly imply that the PHC problem in a directed graph is solved in polynomial time. We also give a polynomial time algorithm to construct a PHC in a directed graph.

For a linear system associated with an $n \times m$ matrix, the Gaussian elimination requires $\mathcal{O}(nm^2)$ time to solve it. We improve the time complexity by proposing a linear time algorithm to solve the PHC problem in directed graphs which is based on the characterization. The algorithm finds a T-join in the bipartite representation of a directed graph, which is essentially equivalent to solving the linear system over GF(2) but faster than it.

A PHC is a closed walk which visits each vertex $1 \bmod 2$ times. For a positive integer p, the problem is generalized as follows: given a directed graph D and a function $f\colon V \to \{0, 1, \dots, p-1\}$, is there a closed walk which visits a vertex v $f(v) \bmod p$ times? We show the generalized problem is characterized similarly to the PHC problem using linear systems over GF(p). The characterization implies the problem is also solved in polynomial time when p is prime or a power of a prime number.

1.2 Relaxation of "For Any" Condition : HC's in Covering Graphs

Hamiltonicity of symmetric graph is a popular subject in graph theory. Among many kinds of symmetric graphs, the class of Cayley graphs is one of the classes that have been paid attention most. Given a group G and its generating set S, a Cayley graph is an undirected graph on the vertex set G with edges between the pairs of two elements g, h of G with $h = gs$ for some $s \in S$. Due to its construction, a Cayley graph has symmetry. There is a folklore conjecture that every connected Cayley graph has an HC, which has been still unsettled. The Hamiltonicity of Cayley graph is an interesting topic not only from the theoretical point of view, but from the point of application such as network design [43] and word processing [27].

There are some celebrated graph classes containing the class of Cayley graphs as a subclass. The class of *vertex-transitive graph* is one of them, and its Hamiltonicity has been well studied [42]. Another class is the *covering graphs of voltage graphs* (simply *covering graphs*). A covering graph is defined by a *voltage graph*, which is a triple of a graph called a base graph, a group, and labels on each edge of the base graph. Contrary to the vertex-transitive graphs, there are few number of studies on the Hamiltonicity of covering graphs. One of the previous results is by Alspach [2], which completely characterizes the Hamiltonicity of generalized Petersen graphs, which is equivalent to a covering graph of the path of length two. Although the base graph is small, the characterization of Hamiltonicity is complicated, which possibly implies that the Hamiltonicity of the covering graphs of the paths is intractable even when the length is small enough, say three or four.

Another result for the Hamiltonicity of the covering graphs is by Batagelj and Pisanski [4], which deals with the Cartesian product of a tree and a cycle. Their characterization is quite simple; let p be the length of the cycle and Δ be the maximum degree of the tree, the Cartesian product of the two graphs is Hamiltonian if and only if $p \geq \Delta$. The Cartesian product of a tree and a cycle is also viewed as a covering graph; the base graph is a tree having a self-loop on each vertex, the group is the cyclic group $\mathbb{Z}_p$, and the the label of each bridge is zero and the label of each self-loop is one. We note here that there is a study which extends the Hamiltonicity result of Batagelj et al. generalizing the concept of Cartesian product [50].

Contribution In Chapter 4 we extend the result of Batagelj et al.'s [4] in two ways; precisely, we propose two graph classes for which the same characterization of Hamiltonicity as Batagelj's, $p \geq \Delta$, holds. The two classes are covering graphs of trees having a self-loop at each vertex, and defined by some conditions on the labels. The condition of the first class is that the edges of Γ except self-loops can be partitioned into k paths, and for each path, the two end vertices of the path have the same label. The condition of the second class is that the edges of Γ except self-loops can be partitioned into k paths, and in each path, there are two adjacent vertices having the same label. Moreover, for both conditions we require every label to be coprime to the order of the group. For each class, one can check that it contains the class of the Cartesian product of a tree and a cycle by taking the all-one label on the self-loop at every vertex. At last, by putting these two conditions together, we obtain a larger class for which the Hamiltonicity is characterized by $p \geq \Delta$.

Recall that it seems to be intractable to characterize the Hamiltonicity of the covering graphs of the small paths. Our result, however, implies that the Hamiltonicity of arbitrarily long paths is characterized by one quite simple inequality under some conditions on the input graph, as long as the label on each self-loop is coprime to the order of the cyclic group. This is a remarkable aspect of our result.

The two conditions mentioned above, are non-trivial to check. We also propose a linear time algorithm to check if the input voltage graph satisfies each condition.

1.3 Relaxation of Constraints: Spanning Tree Problem

A spanning tree of a graph G is a spanning connected subgraph of G having no cycles. The concept of spanning tree is quite fundamental and it appears in many research areas. Moreover, it has plenty of applications for areas such as network designs, distributed algorithms, and data structures. While it is easy to find a spanning tree (of minimum weight) in a graph, there are considerable number of hard variants. For example, finding a spanning tree having minimum or maximum number of leaves, bounded degrees, bounded diameter, and bounded number of hops, are all known to be NP-hard [23, 25]. Notice that a Hamiltonian path is equivalent to a spanning tree with minimum number of leaves.

There is a previous study concerning a spanning tree with a parity condition. Lovàsz proposed the problem, given a connected graph and a list of disjoint pairs of edges, to find a spanning tree of minimum weight such that for each two edges e, f paired in the list, both e and f are used in the tree or neither of them are used. Lovàsz showed the problem is solved in polynomial time using the linear matroid parity [45, 49].

In this thesis, we consider a variant of the problem to find a spanning tree with bounded diameter relaxing the constraint using parity. An *odd depth tree* is a spanning tree such that every leaf has an odd distance from a prescribed root vertex. The *odd depth tree problem* is to find such a tree in a given undirected graph and a root vertex[2].

Related Topics When some parity (or a finite group) condition is added to a graph problem, it is often generalized to the problems in *group-labeled graphs*. A group-labeled graph is a pair (G, ϕ) where G is a graph and ϕ is a map from $E(G)$ to an element of a group[3]. Consider a group-labeled graph associated with the group $\mathbb{Z}_2$ and every edge has label 1. Consider a path on the group-labeled graph in which the sum of the labels of the edges is not equal to zero. Then it corresponds to a path of odd length in the normal graph. The studies on non-zero paths or cycles in group-labeled graphs have been grown these days; packing non-zero A-paths [14] and its reduction to matroids [54], packing non-zero cycles [38], finding a shortest non-zero A-path [39], etc.

One can suppose that an odd depth tree can be also regarded as a "non-zero" spanning tree in a group-labeled graph. However, paths and spanning trees have a crucial difference; a path is regarded as a walk on a graph, while a spanning tree is not. In other words, the operation of "taking sum of labels along a spanning tree" will make no sense. Thus it does not seem that there is a way to relate the odd depth tree problem with those studies on non-zero paths.

[2]One can also consider the *even* depth tree problem, that is, the problem to find a spanning tree such that every leaf has an even depth, which is essentially equivalent to the odd depth tree problem.

[3]The group-labeled graph is essentially same concept as the voltage graph. There is no mathematical difference between them, they come from different origins.

Contribution Our results on the odd depth tree problem are as follows. For bipartite graphs, we show a Hall-type characterization of graphs having an odd depth tree. We propose a polynomial time algorithm to find an odd depth tree based on the characterization. For non-bipartite graphs, we show the problem is NP-complete. We also study the problem in directed graphs, for which we obtained similar results to the undirected case.

1.4 Organization

This thesis is organized as follows. In Chapter 2 we study the PHC problem in undirected graphs. In Chapter 3 we study the PHC problem in directed graphs. In Chapter 4 we study the HC problem in covering graphs. In Chapter 5 we study the odd depth tree problem. In Chapter 6 we summarize the results and mention future works.

Chapter 2

The PHC Problem in Undirected Graphs

In this chapter we investigate the parity Hamiltonian cycle (PHC) problem in undirected graphs. First we introduce some terminologies and notations, then formally define the PHC problem.

2.1 Preliminaries

An *undirected simple graph* (simply we say "*graph*") $G = (V, E)$ is given by a vertex set V (or we use $V(G)$) and an edge set $E \subseteq \binom{V}{2}$ (or $E(G)$). Let $\delta_G(v)$ denote the set of incident edges to v, and let $d_G(v)$ denote the degree of a vertex v in G, i.e., $d_G(v) = |\delta_G(v)|$. We simply use $\delta(v)$ and $d(v)$ without a confusion.

A *walk* is an alternating sequence of vertices and edges $v_0 e_1 v_1 \cdots v_{\ell-1} e_\ell v_\ell$ with an appropriate $\ell \in \mathbb{Z}_{\geq 0}$, such that $e_i = \{v_{i-1}, v_i\} \in E$ for each i $(1 \leq i \leq \ell)$. Note that each vertex or edge may appear more than once in a walk. A walk is *closed* if $v_\ell = v_0$. A graph G is *connected* if there exists a walk from u to v for any pair of vertices $u, v \in V$.

For a closed walk $w = v_0 e_1 \cdots e_\ell v_\ell$, the *visit number* of $v \in V$, denoted by $\mathrm{visit}(v)$, is the number of times that v appears in the walk but counting out the last v_ℓ (since $v_\ell = v_0$), i.e., $\mathrm{visit}(v) = |\{i \in \{0, 1, \ldots, \ell - 1\} \mid v_i = v\}|$.

2.1.1 Parity Hamiltonian cycle

A *parity Hamiltonian cycle* (*PHC* for short) of a graph G is a closed walk in which $\mathrm{visit}(v) \equiv 1 \pmod 2$ holds for each $v \in V$. Remark again that an edge may appear more than once in a PHC w. Clearly, a graph must be connected to have a PHC, and this paper is basically concerned with connected graphs.

An *edge count vector* $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ of a closed walk w is an integer vector where x_e for

$e \in E$ counts the number of occurrence of e in w. Remark that

$$\text{visit}(v) = \frac{1}{2} \sum_{e \in \delta(v)} x_e \tag{2.1}$$

holds for any closed walk. Thus, we see that a PHC is a closed walk whose edge count vector $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ satisfies the *parity condition*

$$\sum_{e \in \delta(v)} x_e \equiv 2 \pmod 4 \tag{2.2}$$

for each $v \in V$.

2.1.2 A PHC as an Eulerian cycle of a multigraph

As given an arbitrary integer vector $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$, the parity condition (2.2) is a necessary condition for that $\boldsymbol{x}$ is an edge count vector of a PHC. In fact, the following easy but important observation provides an if-and-only-if condition.

Proposition 2.1. *Let $G = (V, E)$ be an undirected simple graph and let $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ be an* arbitrary *integer vector. Let $F = \{e \in E \mid x_e > 0\}$, and then $\boldsymbol{x}$ is an edge count vector of a PHC in G if and only if $\boldsymbol{x}$ satisfies* (2.2) *and the subgraph $H = (V, F)$ of G is connected.*

As a preliminary of the proof of Proposition 2.1, we introduce an *Eulerian cycle* of a *multigraph*. For a simple graph $G = (V, E)$ and any nonnegative integer vector $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$, let $\mathcal{E}_{\boldsymbol{x}}$ be a *multiset* such that $e \in E$ appears x_e times in $\mathcal{E}_{\boldsymbol{x}}$. Then, let $(G, \boldsymbol{x})$ represent a *multigraph* with a vertex set V and a multiedge set $\mathcal{E}_{\boldsymbol{x}}$. Note that $(G, \mathbf{1}) = G$ where $\mathbf{1} \in \mathbb{Z}^E_{\geq 0}$ denotes the all one vector. We say $(G, \boldsymbol{x})$ is connected if a simple graph $H = (V, F)$ is connected where $F = \{e \in E \mid x_e > 0\}$. An *Eulerian cycle* of $(G, \boldsymbol{x})$ is a closed walk which uses each element of the multiset $\mathcal{E}_{\boldsymbol{x}}$ exactly once. It is celebrated fact due to Euler [19] that $(G, \boldsymbol{x})$ has an Eulerian cycle if and only if $(G, \boldsymbol{x})$ is connected and $\boldsymbol{x}$ satisfies the *Eulerian condition*

$$\sum_{e \in \delta(v)} x_e \equiv 0 \pmod 2 \tag{2.3}$$

holds for any $v \in V$.

Proof of Proposition 2.1. The 'only if' part is easy from the definition. We prove the 'if' part. Note that $\boldsymbol{x}$ satisfies (2.3) since $\boldsymbol{x}$ satisfies (2.2). Since H is connected by the hypothesis, the multigraph $(G, \boldsymbol{x})$ has an Eulerian cycle w. Considering (2.1), it is not hard to see that w is a PHC since $\boldsymbol{x}$ satisfies (2.2). □

For convenience, we say $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ *admits a PHC* in G if $\boldsymbol{x}$ is an edge count vector of a PHC in G. In summary, Proposition 2.1 implies the following.

Corollary 2.2. *Let $G = (V, E)$ be an undirected simple graph and let $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ be an* arbitrary *integer vector. Then, $\boldsymbol{x}$ admits a PHC in G if and only if $(G, \boldsymbol{x})$ is connected and $\boldsymbol{x}$ satisfies* (2.2)*.*

2.1.3 A PHC with an edge constraint

As we repeatedly remarked, a PHC may use an edge more than once. For convenience, let PHC$_z$ for $z \in \mathbb{Z}_{>0}$ denote a PHC using each edge at most z times.

2.2 Other graph terminology

This section introduces some other graph terminology which we will use in this chapter. (They will be used in the late chapters as well.)

2.2.1 Fundamental notations

A *simple path* is a walk $w = v_0 e_1 v_1 e_2 \cdots e_\ell v_\ell$ which visits every vertex (and hence every edge) at most once, where ℓ ($\ell \geq 0$) is the *length of the path w*. Similarly, a *simple cycle* is a closed walk $w = v_0 e_1 v_1 e_2 \cdots e_\ell v_0$ which visits every vertex at most once, where ℓ ($\ell \geq 3$) is the *length of the cycle w*. An *odd cycle* is a simple cycle of odd length.

Let $G = (V, E)$ be a graph. For an *edge* subset $F \subseteq E$, let $G - F$ denote a graph $H = (V, E \setminus F)$. For a vertex subset $S \subseteq V$, let $G - S$ denote the subgraph induced by $V \setminus S$, i.e., $G - S$ is given by deleting from G all vertices of S and all edges incident to S. For convenience, we simply use $G - e$ for $e \in E$ instead of $G - \{e\}$, and $G - v$ for $v \in V$ as well. An edge $e \in E$ is a *bridge* if its deletion increases the number of connected components of G. For a pair of graphs G and H, let $G + H = (V(G) \cup V(H), E(G) \cup E(H))$.

2.2.2 T-join

Let $G = (V, E)$ be a graph and let T be a subset of V such that $|T|$ is even. Then, $J \subseteq E$ is a *T-join* if the graph $H = (V, J)$ satisfies

$$d_H(v) \equiv \begin{cases} 1 \pmod 2 & \text{if } v \in T, \\ 0 \pmod 2 & \text{otherwise,} \end{cases} \tag{2.4}$$

for any $v \in V$ [52]. Notice that a graph $H' = (T, J)$ may not be connected, in general. A T-join is a generalized notion of a matching, in the sense that J is a matching when all edges in J are disjoint.

Theorem 2.3 (cf. [41]). *For any graph $G = (V, E)$ and for any $T \subseteq V$ satisfying that $|V(C) \cap T|$ is even for every connected component C of G, G contains a T-join.*

A T-join is found in $O(|V| + |E|)$ time (see Appendix A of [47]).

2.2.3 Edge connectivity

A graph is *k-edge-connected* for a positive integer k if the graph remains connected after removing arbitrary at most $k - 1$ edges. A *k-edge-connected component* H of G is a maximal induced subgraph of G such that H is k-edge-connected.

The *ear decomposition* is a cerebrated characterization of two-edge-connected graphs. An *ear* $P = v_0e_1v_1e_2 \cdots e_\ell v_\ell$ of a graph G is a simple path, or a simple cycle (where $v_0 = v_\ell$), of length at least one where $d(v_i) = 2$ for each $i = 1, \ldots, \ell - 1$ and v_0 and v_ℓ are in the same two-edge-connected component of $G - \{v_1, \ldots, v_{\ell-1}\}$, or $G - e_1$ when $\ell = 1$. A *cycle graph*, which consists of a simple cycle only, is two-edge-connected. It is not difficult to see that any two-edge-connected graph, except for a cycle graph, has an ear. By the definition, a graph obtained by deleting an ear P except for v_0 and v_ℓ from a two-edge-connected graph G is again two-edge-connected unless G is a cycle graph. By recursively deleting ears from a two-edge-connected graph G, we eventually obtain a single vertex. The sequence of ears obtained by the operation is called an *ear decomposition* of G. The following fact is well-known.

Theorem 2.4 (cf. [52]). *A graph G is two-edge-connected if and only if G has an ear decomposition.*

2.2.4 Graph classes

Let P_n ($n \geq 2$) denote a graph consisting of a simple path on n vertices. Notice that the *length* of the path P_n is $n - 1$. Let C_n ($n \geq 3$) denote a cycle graph on n vertices. A graph is *P_k-free* (resp. *C_k-free*) if it does not contain P_k (resp. C_k) as an induced subgraph. A graph is *$C_{\geq k}$-free*[1] if G is $C_{k'}$-free for all $k' \geq k$. Clearly, any P_k-free graph is also P_{k+1}-free, as well as any $C_{\geq k}$-free graph is also $C_{\geq k+1}$-free. We can also observe that any P_k-free graph is $C_{\geq k+1}$-free. However, any $C_{\geq k}$-free is not included in P_l-free for any l, since a tree, which is clearly $C_{\geq 3}$-free, admits a path of any length.

Many important graph classes are known to be characterized as P_k-free or $C_{\geq k}$-free. For instance, the class of *cographs* is equivalent to P_4-free, *chordal* is equivalent to $C_{\geq 4}$-free, and *chordal bipartite* is $C_{\geq 6}$-free bipartite (cf. [10])[2].

[1] $C_{\geq k}$-free is often denoted by C_{n+k}-free [10].

[2] Here, we omit the definitions of cograph, chordal and chordal bipartite. This paper requires the properties of P_k-free or $C_{\geq k}$-free, only.

2.3 The PHC_4 problem

It may not be trivial if the PHC problem is in NP, since the length of a closed walk is unbounded. In this section we begin with the PHC_4 problem, and give a complete characterization of the graphs which have PHC_4's. Then we see that the PHC problem is in P as a corollary. The following is the key theorem to characterize the PHC_4 problem.

Theorem 2.5. *A connected graph $G = (V, E)$ contains a* PHC *if and only if the order $|V|$ is even or G is non-bipartite.*

In fact, we prove a slightly stronger theorem (Theorem 2.6). Given a graph $G = (V, E)$ and $S \subseteq V$, an *S-odd walk* is a closed walk of G which visits every vertex of S an odd number of times and visits every other vertex an even number of times. Clearly, a V-odd walk is a PHC of G.

Theorem 2.6. *For any connected graph G and any $S \subseteq V(G)$, G contains an S-odd walk if and only if $|S|$ is even* or *G is non-bipartite. Furthermore, we can find an S-odd walk of G which uses each edge at most* four *times.*

Proof. First, we show the 'only if' part. Suppose that $G = (U, V; E)$ is a bipartite graph, and that $|S|$ is odd. Assume for a contradiction that G has an S-odd walk. Without loss of generality, we may assume that $|U \cap S|$ is odd, and hence $|V \cap S|$ is even since $|S|$ is odd. Then, $\sum_{v \in U \cap S} \mathrm{visit}(v)$ is odd since $\mathrm{visit}(v)$ of an S-odd walk is odd for each $v \in S$ and $|U \cap S|$ is odd, while $\sum_{v \in V \cap S} \mathrm{visit}(v)$ is even since $|V \cap S|$ is even. Furthermore, $\sum_{v \in U \setminus S} \mathrm{visit}(v)$ is even (as well as $\sum_{v \in V \setminus S} \mathrm{visit}(v)$) since $\mathrm{visit}(v)$ of an S-odd walk is even for each $v \notin S$. Thus, $\sum_{v \in U} \mathrm{visit}(v)$ is odd while $\sum_{v \in V} \mathrm{visit}(v)$ is even. On the other hand, any closed walk of G satisfies that $\sum_{u \in U} \mathrm{visit}(u) = \sum_{v \in V} \mathrm{visit}(v)$ since G is bipartite. Contradiction.

Next, we show the 'if' part. When $|S|$ is even, let J be an S-join of G. Let $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ be given by

$$x_e = \begin{cases} 2 & \text{if } e \in J, \\ 4 & \text{otherwise.} \end{cases}$$

Then, $(G, \boldsymbol{x})$ has an Eulerian cycle, which in fact an S-odd walk since J is an S-join of G.

When $|S|$ is odd and G is non-bipartite, let C be an odd cycle of G. Let

$$T = (S \setminus V(C)) \cup (V(C) \setminus S),$$

i.e., T is the symmetric difference between S and $V(C)$. Then, it is not difficult to observe that $|T| \equiv |S| + |V(C)| \pmod 2$ holds, and hence $|T|$ is even since both $|S|$ and $|V(C)|$ are respectively odd by the assumption. Let J' be a T-join of G, and let $\boldsymbol{x}' \in \mathbb{Z}^E_{\geq 0}$

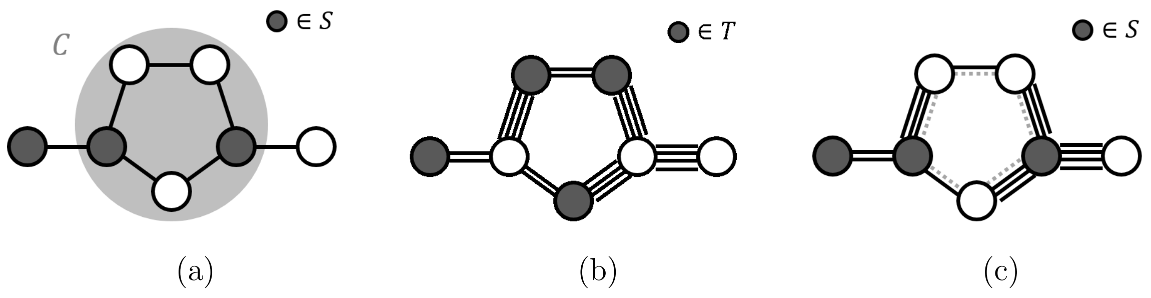


Figure: 2.1: Example for the proof of Theorem 2.5 when $|S|$ is odd and G is non-bipartite. (a) Given graph G, vertex set S, and an odd cycle C, (b) $\boldsymbol{x}'$ given by (2.5) and T, (c) $\boldsymbol{x}''$ given by (2.6).

be given by

$$x'_e = \begin{cases} 2 & \text{if } e \in J', \\ 4 & \text{otherwise,} \end{cases} \tag{2.5}$$

and then $(G, \boldsymbol{x}')$ satisfies the Eulerian condition, and any Eulerian cycle is a T-odd walk since J' is a T-join. Now, let $\boldsymbol{x}''$ be given by

$$x''_e = \begin{cases} x'_e - 1 & \text{if } e \in E(C), \\ x'_e & \text{otherwise.} \end{cases} \tag{2.6}$$

We remark that $x_e \leq 4$ for any $e \in E$, as well as $x_e > 0$, where the latter observation implies $(G, \boldsymbol{x}'')$ is connected. Now, it is not difficult to observe that $(G, \boldsymbol{x}'')$ has an S-odd walk since S is a symmetric difference between T and $V(C)$. □

Since a T-join of a graph is found in linear time, the following is derived from the proof of Theorem 2.6.

Corollary 2.7. *A PHC$_4$ is found in linear time for any connected graph G if the order $|V|$ is even or G is non-bipartite.* □

Finally, we remark that Theorem 2.6 is a generalization of the following theorem due to Brigham et al. [11]. They proved it by presenting an algorithm based on a depth-first-search, without using the notion of T-join.

Theorem 2.8 (cf. [11]). *Every connected graph has a closed walk visiting all vertices such that the walk visits at least $|V| - 1$ vertices an odd number of times and uses each edge at most four times. Such a walk is found in linear time.*

2.4 The PHC_z problems when $z = 1, 2, 3$

In Section 2.3, we obtained a good characterization for the PHC_4 problem. This section shows that the PHC_z problem is NP-complete for each $z \in \{1, 2, 3\}$. We remark that these hardness results are independent, e.g., the fact that PHC_3 is NP-complete does not imply the fact that PHC_2 is NP-complete, and vice versa.

Theorem 2.9. *The* PHC_1 *problems is* NP*-complete, even when a given graph is three-connected, planar and cubic.*

Proof. It is not difficult to observe that the PHC_1 problem is exactly the same as the HC problem for a cubic graph. It is known that the HC problem is NP-complete even when a given graph is three-connected, planar and cubic [24]. □

Theorem 2.10. *The* PHC_2 *and* PHC_3 *problems are respectively* NP*-complete, even when a given graph is two-edge-connected.*

Proof. It is easy to see that both problems are in NP. We reduce the HC problem for cubic graphs to the PHC problems. Let G be a two-edge-connected cubic graph, which is an input of the HC problem. Then, we construct a graph H as an input of the PHC problems, as follows (see also Figure 2.2):

- Subdivide every edge $e = \{v, u\} \in E(G)$ into a path of length three, i.e., remove e and add vertices v_e, u_e and edges $\{v, v_e\}$, $\{v_e, u_e\}$, $\{u_e, u\}$.
- For each vertex $v \in V(G)$, attach a cycle of length four, i.e., add vertices w_{v1}, w_{v2}, w_{v3} and edges $\{v, w_{v1}\}$, $\{w_{v1}, w_{v2}\}$, $\{w_{v2}, w_{v3}\}$, $\{w_{v3}, v\}$.

For convenience, let V denote the set of original vertices, i.e., $V = V(G)$, let V_{s} denote the set of vertices u_e, v_e added by the subdivision, and let V_{c} denote the set of vertices w_{v1}, w_{v2}, w_{v3} added by the cycle attachment, i.e., $V(H) = V \cup V_{\mathrm{s}} \cup V_{\mathrm{c}}$ where $|V_{\mathrm{s}}| = 2|E(G)|$ and $|V_{\mathrm{c}}| = 3|V(G)|$. Similarly, let E_{s} denote the set of edges $\{v, v_e\}$, $\{v_e, u_e\}$, $\{u_e, u\}$ given by the subdivision for each $v \in V$, and let E_{c} denote the set of edges $\{v, w_{v1}\}$, $\{w_{v1}, w_{v2}\}$, $\{w_{v2}, w_{v3}\}$, $\{w_{v3}, v\}$ given by the cycle attachment for each $v \in V$, i.e., $E(H) = E_{\mathrm{s}} \cup E_{\mathrm{c}}$ where $|E_{\mathrm{s}}| = 3|E(G)|$ and $|E_{\mathrm{c}}| = 4|V(G)|$.

In the following, we will prove (i) if G has an HC then H has a PHC_2, and (ii) if H has a PHC_3 then G has an HC. Since a PHC_2 is also a PHC_3, (i) implies that (i') if G has an HC then H has a PHC_3, as well as (ii) implies that (ii') if H has a PHC_2 then G has an HC. Thus, (i) and (ii') imply the NP-hardness of the PHC_2 problem, and (i') and (ii) imply the NP-hardness of the PHC_3 problem.

First, we show (i) if G has an HC then H has a PHC_2, by presenting an edge count vector $\boldsymbol{x} \in \mathbb{Z}_{\geq 0}^{E(H)}$. For the edges in E_{s}, set $x_{\{v,v_e\}} = x_{\{v_e,u_e\}} = x_{\{u_e,u\}} = 1$ if the HC uses $e = \{v, u\} \in E(G)$, otherwise set $x_{\{v,v_e\}} = x_{\{u_e,u\}} = 2$ and $x_{\{v_e,u_e\}} = 0$. For each edge $e' \in E_{\mathrm{c}}$, set $x_{e'} = 1$ (see Figure 2.2, right). It is not difficult to observe that $(G, \boldsymbol{x})$ is connected and satisfies the Eulearian condition. Furthermore, an Eulerian cycle of $(G, \boldsymbol{x})$ visits every vertex of $V(H)$ an odd number of times; it visits every vertex of V exactly three times, and every other vertices exactly once. Thus, $\boldsymbol{x}$ admits a PHC$_2$ of H.

Next, we show (ii) if H has a PHC$_3$ then G has an HC. Let $\boldsymbol{x}$ be the edge count vector of the PHC$_3$ of H. In order to visit each vertex of V_{c} an odd number of times, we can observe that $(x_{\{v,w_{v1}\}}, x_{\{w_{v1},w_{v2}\}}, x_{\{w_{v2},w_{v3}\}}, x_{\{w_{v3},v\}})$ should be $(1,1,1,1)$ or $(3,3,3,3)$ for each cycle attached to $v \in V$ since $d_H(v') = 2$ for $v' \in V_{\mathrm{c}}$. Similarly, there are three possible assignments of $(x_{\{v,v_e\}}, x_{\{v_e,u_e\}}, x_{\{u_e,u\}})$ for the subdivided path between v and u (i.e., $\{v,u\} \in E(G)$), namely $(1,1,1)$, $(2,0,2)$ or $(3,3,3)$. Recall the hypothesis that G is cubic, and let $a, b, c \in E(G)$ denote the edges incident to v in G, meaning that three subdivided paths are incident to $v \in V$ in H. To be precise, $d_H(v)$ is 5, where $\{v, v_a\}$, $\{v, v_b\}$, $\{v, v_c\}$ in E_{s} and $\{v, w_{v1}\}$, $\{v, w_{v3}\}$ in E_{c} are incident to v. The assumption that $\boldsymbol{x}$ is the edge count vector of a PHC$_3$, which visits every vertex an odd number of times, implies that

$$x_{\{v,v_a\}} + x_{\{v,v_b\}} + x_{\{v,v_c\}} + x_{\{v,w_{v1}\}} + x_{\{v,w_{v3}\}} \equiv 2 \pmod 4$$

holds. As we saw, $x_{\{v,w_{v1}\}} = x_{\{v,w_{v3}\}} = 1$ or 3 hold, which implies

$$x_{\{v,v_a\}} + x_{\{v,v_b\}} + x_{\{v,v_c\}} \equiv 0 \pmod 4.$$

Then, the possible multisets of values of $\{\{x_{\{v,v_a\}}, x_{\{v,v_b\}}, x_{\{v,v_c\}}\}\}$ are $\{\{1,1,2\}\}$ or $\{\{3,3,2\}\}$ where $\{\{\cdot\}\}$ denotes a multiset. This means that exactly two of subdivided paths incident to each vertex of V are assigned $(1,1,1)$ or $(3,3,3)$, while the other is assigned $(2,0,2)$. Now it is easy to observe that we obtain an HC of G by choosing edges corresponding to $(1,1,1)$ or $(3,3,3)$ paths. □

2.4.1 The PHC$_3$ problem for four-edge-connected graphs

The PHC$_3$ problem is NP-complete for *two*-edge-connected graphs, as we have shown in Theorem 2.10. For *four*-edge-connected graphs, we obtain the following.

Theorem 2.11. *A* four-*edge-connected graph $G = (V, E)$ contains a* PHC$_3$ *if and only if the order $|V|$ is even or G is non-bipartite.*

To prove Theorem 2.11, we use the following celebrated theorem.

Theorem 2.12 ([46, 28])**.** *Every four-edge-connected graph has two edge disjoint spanning trees.*

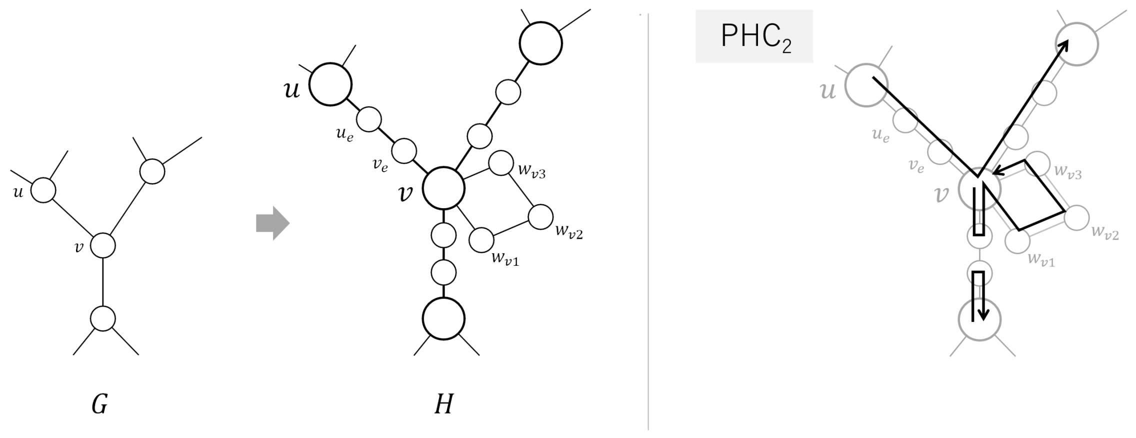


Figure: 2.2: A PHC$_2$ around vertex v.

Proof of Theorem 2.11. The 'only-if' part follows from that of Theorem 2.5. We show the 'if' part, in a constructive way. Suppose that G is four-edge-connected. Then, let τ and τ' be a pair of edge disjoint spanning trees of G, implied by Theorem 2.12. Intuitively, we construct a closed walk on τ, and control the parity condition using edges in τ', then we obtain a PHC$_3$.

Let $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ be given by

$$x_e = \begin{cases} 2 & \text{if } e \in E(\tau), \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Then, $(G, \boldsymbol{x})$ is connected, and has an Eulerian cycle, say w. Let S be the set of vertices with even degree in τ, i.e., S is the entire set of vertices each of which w visits an even number of times. We also remark that $|V \setminus S|$ is even, by the handshaking lemma. In the following, we consider two cases whether $|V|$ is even or odd.

If $|V|$ is even, then $|S|$ is even. Let $J \subseteq E(\tau')$ be an S-join in the tree τ'. Then, let $\boldsymbol{x}' \in \mathbb{Z}^E_{\geq 0}$ be defined by

$$x'_e = \begin{cases} x_e + 2 & \text{if } e \in J, \\ x_e & \text{otherwise.} \end{cases} \tag{2.8}$$

It is easy to see that $\boldsymbol{x}'$ satisfies the parity condition (2.2) for each vertex of V since J is an S-join. Clearly $(G, \boldsymbol{x}')$ is connected, and $\boldsymbol{x}'$ admits a PHC by Corollary 2.2. Notice that $J \subseteq E(\tau')$ is disjoint with $E(\tau)$, meaning that (2.7) and (2.8) imply that $x'_e \leq 2$ for each $e \in E$. We obtain the claim in the case.

If $|V|$ is odd, then $|S|$ is odd. By the hypothesis, G is non-bipartite and hence G

contains an odd cycle, say C. Let

$$T = (S \setminus V(C)) \cup (V(C) \setminus S),$$

i.e., T is the symmetric difference between S and $V(C)$. Then, it is not difficult to observe that $|T| \equiv |S| + |C| \pmod 2$ holds, and hence $|T|$ is even since both $|S|$ and $|C|$ are respectively odd by the assumption. Let $J' \subseteq E(\tau')$ be an T-join of τ', and let $\boldsymbol{x}'' \in \mathbb{Z}^E_{\geq 0}$ be defined by

$$x''_e = \begin{cases} x_e + 2 & \text{if } e \in J', \\ x_e & \text{otherwise.} \end{cases} \tag{2.9}$$

It is not difficult to see that $(G, \boldsymbol{x}'')$ has a $(V \setminus V(C))$-odd walk. Finally, we modify $\boldsymbol{x}''$ to $\boldsymbol{x}''' \in \mathbb{Z}^E_{\geq 0}$ by

$$x'''_e = \begin{cases} x''_e + 1 & \text{if } e \in E(C), \\ x''_e & \text{otherwise,} \end{cases} \tag{2.10}$$

and then we obtain a PHC$_3$. □

It is known (cf. [52]) that a pair of edge disjoint spanning trees τ and τ' in a four-edge-connected graph is found in polynomial time (e.g., $\mathcal{O}(|E(G)|^2)$ time, due to Roskind, Tarjan [51]). Thus, the proof of Theorem 2.11 also implies a polynomial time (e.g., $\mathcal{O}(|E(G)|^2)$) algorithm to find a PHC$_3$ in a four-edge-connected graph G.

2.5 The PHC$_3$ Problem for Two-Edge-Connected Graphs

The PHC$_3$ problem is NP-complete for *two*-edge-connected graphs (Theorem 2.10), while it is solved in polynomial time for any *four*-edge-connected graph (Theorem 2.11). We devote this section to dig into the PHC$_3$ problem in two-edge-connected graphs (which includes the class of three-edge-connected graphs). The goal of this section is to establish the following theorem.

Theorem 2.13. *Suppose that a two-edge-connected graph G is P_6-free or $C_{\geq 5}$-free. Then, $G = (V, E)$ contains a* PHC$_3$ *if and only if the order $|V|$ is even or G is non-bipartite.*

As mentioned before, $C_{\geq 5}$-free contains some important graph classes such as chordal (equivalent to $C_{\geq 4}$-free), chordal bipartite (equivalent to $C_{\geq 5}$-free bipartite), and cograph (equivalent to P_4-free). We also remark that the Hamiltonian cycle problem is NP-complete for $C_{\geq 4}$-free graphs, as well as P_5-free graphs (cf. [10]).

2.5.1 All-roundness : Preliminary

We introduce the notion of the *all-roundness* of a graph. This notion is used throughout the rest part of this section to characterize the graphs having PHC$_3$'s.

Generalized problem

Section 2.5 is actually concerned with the following problem, which is a slight generalization of the PHC$_3$ problem.

Problem 1. *Given a graph $G = (V, E)$ and a map $f\colon V \to \{0, 1, 2, 3\}$, find $\boldsymbol{x} \in \{0, 1, 2, 3\}^E$ satisfying the conditions that*

$$\sum_{e\in\delta(v)} x_e \equiv f(v) \pmod 4 \qquad \text{for any } v \in V, \tag{2.11}$$

$$(G, \boldsymbol{x}) \text{ is connected.} \tag{2.12}$$

Clearly, the PHC$_3$ problem is given by setting $f(v) = 2$ for any $v \in V$ (recall Corollary 2.2). For convenience, we call $\boldsymbol{x} \in \{0, 1, 2, 3\}^E$ a *mod-4 f-factor* of G if $\boldsymbol{x}$ satisfies (2.11). A mod-4 f-factor $\boldsymbol{x} \in \{0, 1, 2, 3\}^E$ is *connected* if it satisfies (2.12), i.e., Problem 1 is a problem to find a *connected mod-4 f-factor*. We remark the following two facts.

Proposition 2.14. *A graph $G = (V, E)$ has a mod-4 f-factor only when the map f satisfies that*

$$\sum_{v\in V} f(v) \text{ is even.} \tag{2.13}$$

Proof. Summing up the both sides of (2.11) over V, we obtain

$$\sum_{v\in V} f(v) \equiv \sum_{v\in V}\sum_{e\in\delta(v)} x_e \pmod 4. \tag{2.14}$$

It is not difficult to see that

$$\sum_{v\in V}\sum_{e\in\delta(v)} x_e = 2\sum_{e\in E} x_e \tag{2.15}$$

holds, in an analogy with the handshaking lemma, that is $\sum_{v\in V}\sum_{e\in\delta(v)} 1 = \sum_{e\in E} 2$. By (2.14) and (2.15), we obtain that

$$\sum_{v\in V} f(v) \equiv 2\sum_{e\in E} x_e \pmod 4$$

which implies the claim. □

We will later show in Lemma 2.23 that (2.13) is also sufficient for any connected non-bipartite graph to have a mod-4 f-factor. For bipartite graphs, we need an extra necessary condition on f.

Proposition 2.15. *A bipartite graph $G = (U, V; E)$ has a mod-4 f-factor only when the map f satisfies that*

$$\sum_{v\in U} f(v) \equiv \sum_{v\in V} f(v) \pmod 4. \tag{2.16}$$

Proof. Since $G = (U, V; E)$ is bipartite,

$$\sum_{v\in U}\sum_{e\in\delta(v)} x_e = \sum_{v\in V}\sum_{e\in\delta(v)} x_e \tag{2.17}$$

is required. Summing up (2.11) over U and V, respectively, we obtain

$$\begin{aligned}\sum_{v\in U} f(v) &\equiv \sum_{v\in U}\sum_{e\in\delta(v)} x_e \pmod 4, \quad \text{and}\\ \sum_{v\in V} f(v) &\equiv \sum_{v\in V}\sum_{e\in\delta(v)} x_e \pmod 4\end{aligned}$$

hold, which with (2.17) implies the claim. □

Notice that the condition (2.16) implies (2.13). We will show in Lemma 2.23 that (2.16) is also sufficient for any connected bipartite graph.

All-roundness : Definition

Then, we introduce the following two notions.

Definition 2.16. *A graph G is* all-round *if G has a connected mod-4 f-factor for any map f satisfying* (2.13)*.*

Definition 2.17. *A bipartite graph G is* bipartite all-round *if G has a connected mod-4 f-factor for any map f satisfying* (2.16)*.*

It is not difficult to see that Definitions 2.16 and 2.17 are (too strong) sufficient conditions that a graph contains a PHC$_3$. In the rest of Section 2.5, we will show the following theorems, which immediately imply Theorem 2.13.

Theorem 2.18. *Every two-edge-connected $C_{\geq 5}$-free graph is either all-round or bipartite all-round.*

Theorem 2.19. *Every two-edge-connected P_6-free graph, except for C_5, is either all-round or bipartite all-round.*

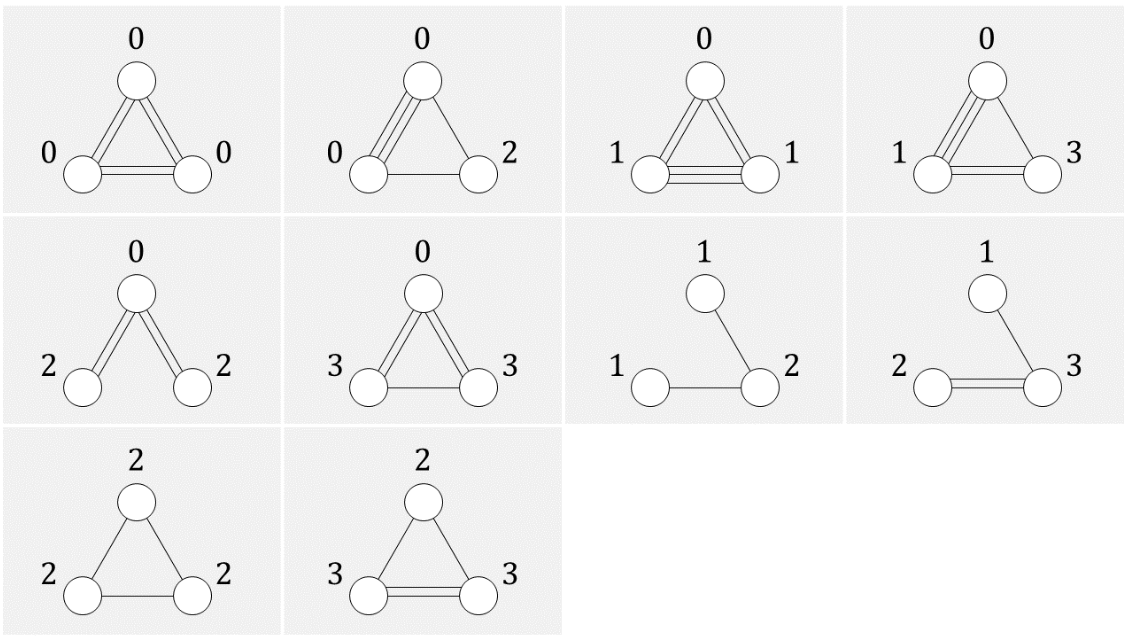


Figure: 2.3: K_3 is all-round. A number beside a vertex denotes the value of f. There are $4^3/2 = 32$ possible assignment of f, but by the symmetry, it suffices to check these 10 cases.

All-roundness of small graphs

As a preliminary step of the proof, we remark the following facts, which are easily confirmed by the exhaustive search.

Proposition 2.20. *C_3 (i.e., K_3) is all-round (See Figure 2.3).*

Proposition 2.21. *C_4 and C_6 are bipartite all-round, respectively.*

The following fact may be counterintuitive.

Proposition 2.22. *C_5 is* not *all-round.*

Figure 2.4 shows an example of f for which Problem 1 does not have a solution. Notice that C_5 clearly has a PHC$_3$, which implies that the all-roundness is not a necessary condition for a graph to have a PHC$_3$.

2.5.2 All-roundness of $C_{\geq 5}$-free: Proof of Theorem 2.18

This section shows Theorem 2.18, presenting some useful idea on a mod-4 f-factor of a graph, and all-roundness or bipartite all-roundness. To begin with, we show for any appropriate map f that it is easy to find a mod-4 f-factor, which may be disconnected.

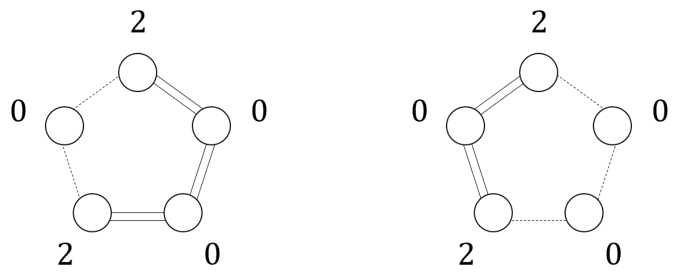


Figure: 2.4: A counterexample of the all-roundness (C_5). A number beside a vertex v is the value of $f(v)$. For this f, there are two ways to suffice the (mod 4) degree condition on every vertex, neither of the factors are connected.

Lemma 2.23. *Any connected non-bipartite graph has a mod-4 f-factor for any map f satisfying* (2.13). *Any connected bipartite graph has a mod-4 f-factor for any map f satisfying* (2.16).

Proof. We give a constructive proof. Let $T := \{v \in V \mid f(v) \text{ is odd}\}$. We remark that $|T|$ is even since $\sum_{v\in V} f(v)$ is even by (2.13). Then, let $J \subseteq E$ be a T-join, and let $\boldsymbol{x} \in \{0,1,2,3\}^E$ be given by

$$x_e = \begin{cases} 1 & \text{if } e \in J, \\ 0 & \text{otherwise.} \end{cases}$$

Let $f'\colon V \to \{0,1,2,3\}$ be

$$f'(v) = \left(f(v) - \sum_{e\in\delta(v)} x_e \right) \bmod 4. \tag{2.18}$$

Remark that $f'(v)$ is even for any $v \in V$, i.e., $f'(v) = 0$ or 2 for any $v \in V$. Let $S = \{v \in V \mid f'(v) = 2\}$. If $|S|$ is even, then let J' be an S-join and let $\boldsymbol{x}' \in \{0,1,2,3\}^E$ be defined by

$$x'_e = \begin{cases} x_e + 2 & \text{if } e \in J' \\ x_e & \text{otherwise.} \end{cases}$$

It is not difficult to observe that $\boldsymbol{x}'$ satisfies (2.11), and $x'_e \leq 3$ holds for any $e \in E$. Thus, we obtain the claim in the case. Here we remark that if G is bipartite then $|S|$ is even, since (2.16) implies

$$\begin{aligned}\sum_{v\in U\cup V} f'(v) &= \sum_{v\in U} f'(v) + \sum_{v\in V} f'(v)\\ &\equiv \sum_{v\in U} f'(v) - \sum_{v\in V} f'(v)\\ &\equiv 0 \pmod 4.\end{aligned}$$

If $|S|$ is odd, we need an extra process. Notice that G is non-bipartite in the case, by the above argument. Let C be an odd cycle of G and let $\boldsymbol{x}'' \in \{0,1,2,3\}^{E(G)}$ be

$$x''_e = \begin{cases} x_e + 1 & \text{if } e \in E(C)\\ x_e & \text{otherwise.}\end{cases}$$

Let $f''\colon V \to \{0,1,2,3\}$ be

$$f''(v) = \left(f(v) - \sum_{e\in\delta(v)} x''_e\right) \bmod 4. \tag{2.19}$$

Let $S' = \{v \in V(G) \mid f''(v) = 2\}$. Then, S' is the symmetric difference between S and $V(C)$, which implies $|S'|$ is even since $|S|$ and $|V(C)|$ are respectively odd. Let $J' \subseteq E(G)$ be an S'-join and let $\boldsymbol{x}''' \in \{0,1,2,3\}^{E(G)}$ be

$$x'''_e = \begin{cases} x''_e + 2 \pmod 4 & \text{if } e \in J'\\ x''_e & \text{otherwise.}\end{cases}$$

Then, we obtain a mod-4 f-factor. □

To obtain a *connected* mod-4 f-factor, the notions of all-roundness and bipartite all-roundness play important roles. The following lemma is easy to see from the definition.

Lemma 2.24 (connecting lemma). *Let $\boldsymbol{x}$ be a mod-4 f-factor of G, and let H_1 and H_2 be a distinct pair of connected components of $(G, \boldsymbol{x})$. Suppose that there is a connected subgraph H of G such that $V(H)$ intersects both $V(H_1)$ and $V(H_2)$, and that H is all-round or bipartite all-round. Then, G has another mod-4 f-factor $\boldsymbol{x}'$ such that H_1 and H_2 are connected in $(G, \boldsymbol{x}')$ where other connected components are respectively kept being connected.*

Proof. As given $\boldsymbol{x} \in \{0,1,2,3\}^{E(G)}$ and H described in the hypothesis, we define a map $f_H\colon V(H) \to \{0,1,2,3\}$ by

$$f_H(v) = \sum_{e\in\delta_H(v)} x_e \bmod 4.$$

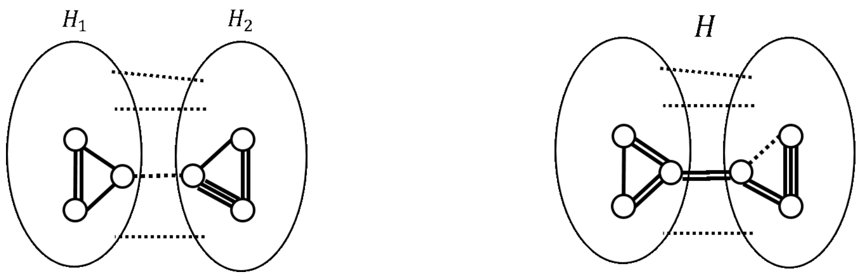


Figure: 2.5: Connecting lemma. Left figure shows that H_1 and H_2 are disconnected in $(G, \boldsymbol{x})$. Right figure shows that a mod-4 f-factor in H is replaced by another *connected* one. The hypothesis that H is all-round or bipartite all-round implies such a connected mod-4 f-factor in H.

Let $\boldsymbol{y} \in \{0,1,2,3\}^{E(H)}$ be defined by $y_e = x_e$ for each $e \in E(H)$, and then clearly $\boldsymbol{y}$ is a mod-4 f_H-factor of H. Proposition 2.14 implies that f_H satisfies (2.13), as well as Proposition 2.15 implies that f_H satisfies (2.16) if H is bipartite. The hypothesis that H is all-round or bipartite all-round implies that there is a *connected* mod-4 f_H-factor $\boldsymbol{y}' \in \{0,1,2,3\}^{E(H)}$ of H. Let $\boldsymbol{x}' \in \{0,1,2,3\}^{E(G)}$ be

$$x'_e = \begin{cases} y'_e & \text{if } e \in E(H), \\ x_e & \text{otherwise,} \end{cases}$$

and then, it is not difficult to observe that $\boldsymbol{x}'$ is a desired mod-4 f-factor of G (See also Figure 2.5). □

It is not difficult to see that Lemmas 2.23 and 2.24 imply the following useful lemma.

Lemma 2.25. *Let G be a graph. Suppose that for any edge $e \in E(G)$ there exists a subgraph H of G such that $e \in E(H)$ and H is all-round or bipartite all-round. Then, G is all-round, or bipartite all-round.*

Proof. For any appropriate f, meaning that f satisfies (2.13), and (2.16) if G is bipartite, Lemma 2.23 implies a mod-4 f-factor $\boldsymbol{x}$. Since the hypothesis, we can obtain a connected mod-4 f-factor by iteratively applying the connecting lemma (Lemma 2.24) to $\boldsymbol{x}$. □

In fact, the all-roundness of $C_{\geq 5}$-free (Theorem 2.18) is immediate from Lemma 2.25.

Proof of Theorem 2.18. Let G be a two-edge-connected $C_{\geq 5}$-free graph. Then, any edge e of G is contained in C_3 or C_4. Since C_3 is all-round (Proposition 2.20), and C_4 is bipartite all-round (Proposition 2.21), Lemma 2.25 implies that G is all-round or bipartite all-round. □

We will use Lemma 2.25 again in the proof of Theorem 2.19, with some extra arguments. We also show another example in Section 2.6.2 where we apply Lemma 2.6.2 to dense graphs.

2.5.3 All-roundness of P_6-free: Proof of Theorem 2.19

We cannot directly apply Lemma 2.25 to a P_6-free graph G, since G may contain C_5, which is *not* all-round (recall Proposition 2.22). To prove Theorem 2.19, we investigate the all-roundness (or bipartite all-roundness) of two-edge-connected graphs, considering ear-decomposition in Section 2.5.3.

Ear-decomposition for mod-4 all-round graphs

This section presents three lemmas, which claim that adding a short ear preserves the all-roundness or bipartite all-roundness.

Lemma 2.26. *Let G and G' be two-edge-connected non-bipartite graphs, where G' is given by adding to G an ear of length at most seven. If G is all-round, then G' is all-round.*

Proof. The proof idea is similar to Lemma 2.25: construct a mod-4 f-factor which may not be connected (Lemma 2.23), and let it be connected using the connecting lemma (Lemma 2.24) on assumption that G is all-round. A technical issue of the proof idea is that the ear is not bipartite all-round. We show that a desired mod-4 f-factor always exists if the ear is short.

Let $p = v_0e_1v_1e_2\cdots e_\ell v_\ell$ denote the ear added to G, and let $P = (\{v_0, v_1, \dots, v_\ell\}, \{e_1, \dots, e_\ell\})$, where $\ell \le 7$. Remark that $v_0, v_\ell \in V(G)$. As given an arbitrary map $f\colon V(G') \to \{0,1,2,3\}$ satisfying (2.13), let $\boldsymbol{x} \in \{0,1,2,3\}^{E(G')}$ be a mod-4 f-factor implied by Lemma 2.23. For convenience, let $\boldsymbol{x}^{(0)} = (x_1, \dots, x_\ell) \in \{0,1,2,3\}^{E(P)}$ where x_i denotes x_{e_i} for simplicity. We also define $\boldsymbol{x}^{(a)} \in \{0,1,2,3\}^{E(P)}$ for each $a \in \{1,2,3\}$ by

$$x_i^{(a)} = \begin{cases} (x_i + a) \bmod 4 & \text{if } i \text{ is odd,} \\ (x_i - a) \bmod 4 & \text{if } i \text{ is even.} \end{cases}$$

Notice that $\boldsymbol{x}^{(a)}$ for each $a \in \{0,1,2,3\}$ is a mod-4 $f_P^{(a)}$-factor for a map $f_P^{(a)}\colon V(P) \to \{0,1,2,3\}$ given by

$$f_P^{(a)}(v) = \begin{cases} (x_1 + a) \bmod 4 & \text{if } v = v_0, \\ (x_i + x_{i+1}) \bmod 4 = f(v) & \text{if } v \in \{v_1, \dots, v_{\ell-1}\}, \\ (x_\ell + (-1)^\ell a) \bmod 4 & \text{if } v = v_\ell, \end{cases}$$

when $v_0 \neq v_\ell$, whereas

$$f_P^{(a)}(v) \;=\; \begin{cases} (x_1 + x_\ell + a + (-1)^\ell a) \bmod 4 & \text{if } v = v_0 = v_\ell, \\ (x_i + x_{i+1}) \bmod 4 = f(v) & \text{if } v \in \{v_1, \ldots, v_{\ell-1}\}, \end{cases}$$

when $v_0 = v_\ell$. At the same time, let $f_G^{(a)} \colon V(G) \to \{0,1,2,3\}$ be defined for each $a \in \{0,1,2,3\}$ by

$$f_G^{(a)}(v) \;=\; \begin{cases} \left(f(v) - f_P^{(a)}(v)\right) \bmod 4 & \text{if } v \in \{v_0, v_\ell\}, \\ f(v) & \text{otherwise,} \end{cases}$$

and then G has a *connected* mod-4 $f_G^{(a)}$-factor $\boldsymbol{y}^{(a)} \in \{0,1,2,3\}^{E(G)}$ since G is all-round by the hypothesis. If $(P, \boldsymbol{x}^{(a)})$ consists of at most two connected components, i.e., $x_i^{(a)} = 0$ holds for at most one $i \in \{1,2,\ldots,\ell\}$, then $\boldsymbol{x}^{(a)}$ and $\boldsymbol{y}^{(a)}$ imply a connected mod-4 f-factor of G'.

Then, we claim that there is $a \in \{0,1,2,3\}$ such that $(P, \boldsymbol{x}^{(a)})$ consists of at most two connected components. Remark that $\{x_i^{(0)}, x_i^{(1)}, x_i^{(2)}, x_i^{(3)}\} = \{0,1,2,3\}$ holds for each $i \in \{1,2,\ldots,\ell\}$. By a version of the pigeon hole principle, the hypothesis of $\ell \leq 7$ implies that there is an index $a \in \{0,1,2,3\}$ such that $x_j^{(a)} = 0$ holds for at most one $j \in \{1,2,\ldots,\ell\}$; otherwise the multiset $\{\{x_j^{(a)} \mid a \in \{0,1,2,3\},\ j \in \{1,2,\ldots,\ell\}\}\}$ contains 8 or more 0's. The $\boldsymbol{x}^{(a)}$ is the desired mod-4 $f_P^{(a)}$-factor for P, and we obtain the claim. □

The following lemma is a version of Lemma 2.26 for bipartite graphs. The proof is essentially the same, and we omit it.

Lemma 2.27. *Let G and G' be two-edge-connected bipartite graphs, where G' is given by adding to G an ear of length at most seven. If G is bipartite all-round, then G' is bipartite all-round.* □

Finally, we show the following lemma, which claims a connection between a bipartite all-round graph and an all-round graph.

Lemma 2.28. *Let G be a two-edge-connected bipartite graph, and let G' be a two-edge-connected non-bipartite graph given by adding to G an ear of length at most three. If G is bipartite all-round, then G' is all-round.*

Proof. The proof is similar to Lemma 2.26. Let $p = v_0 e_1 v_1 e_2 \cdots e_\ell v_\ell$ denote the ear added to G where $\ell \leq 3$, and let $P = (\{v_0, v_1, \ldots, v_\ell\}, \{e_1, \ldots, e_\ell\})$. Remark that $v_0, v_\ell \in V(G)$. As given an arbitrary map $f \colon V(G') \to \{0,1,2,3\}$ satisfying (2.13), let $\boldsymbol{x} \in \{0,1,2,3\}^{E(G')}$ be a mod-4 f-factor implied by Lemma 2.23. For convenience, let

$\boldsymbol{x}^{(0)} = (x_1, \ldots, x_\ell) \in \{0,1,2,3\}^{E(P)}$ where x_i denotes x_{e_i} for simplicity. We also define $\boldsymbol{x}^{(2)} \in \{0,1,2,3\}^{E(P)}$ by

$$x_i^{(2)} = (x_i + 2) \bmod 4.$$

Notice that $\boldsymbol{x}^{(2)}$ is also a mod-4 $f_P^{(2)}$-factor for a map $f_P^{(2)} \colon V(P) \to \{0,1,2,3\}$ given by

$$f_P^{(2)}(v) \;=\; \begin{cases} (x_1 + 2) \bmod 4 & \text{if } v = v_0, \\ (x_i + x_{i+1}) \bmod 4 = f(v) & \text{if } v \in \{v_1, \ldots, v_{\ell-1}\}, \\ (x_\ell + 2) \bmod 4 & \text{if } v = v_\ell, \end{cases}$$

when $v_0 \neq v_\ell$, whereas

$$f_P^{(2)}(v) \;=\; \begin{cases} (x_1 + x_\ell + 2) \bmod 4 & \text{if } v = v_0 = v_\ell, \\ (x_i + x_{i+1}) \bmod 4 = f(v) & \text{if } v \in \{v_1, \ldots, v_{\ell-1}\}, \end{cases}$$

when $v_0 = v_\ell$. At the same time, let $f_G^{(a)} \colon V(G) \to \{0,1,2,3\}$ be defined for each $a \in \{0,2\}$ by

$$f_G^{(a)}(v) \;=\; \begin{cases} \left(f(v) - f_P^{(a)}(v)\right) \bmod 4 & \text{if } v \in \{v_0, v_\ell\}, \\ f(v) & \text{otherwise,} \end{cases}$$

and then $f_G^{(0)}$ clearly satisfies (2.16), and $f_G^{(2)}$ as well. Thus, G has a *connected* bipartite mod-4 $f_G^{(a)}$-factor $\boldsymbol{y}^{(a)} \in \{0,1,2,3\}^{E(G)}$ for each $a \in \{0,2\}$ since G is bipartite all-round by the hypothesis. If $(P, \boldsymbol{x}^{(a)})$ consists of at most two connected components, i.e., $x_i^{(a)} = 0$ holds at most one $i \in \{1,2,\ldots,\ell\}$, then $\boldsymbol{x}^{(a)}$ and $\boldsymbol{y}^{(a)}$ imply a connected mod-4 f-factor of G'.

Then, we claim that one of $(P, \boldsymbol{x}^{(0)})$ or $(P, \boldsymbol{x}^{(2)})$ consists of at most two connected components. Remark that $\{x_i^{(0)}, x_i^{(2)}\} = \{0,2\}$ or $\{1,3\}$ holds for each $i \in \{1,2,\ldots,\ell\}$. Since $\ell \leq 3$, it is not difficult to observe the claim. □

Proof of Theorem 2.19

Now, we show Theorem 2.19. In fact, we prove the following lemma, which immediately implies Theorem 2.19 together with Lemma 2.25.

Lemma 2.29. *Let $G = (V, E)$ be a two-edge-connected P_6-free graph such that $G \neq C_5$. For any edge $e \in E$, there is a subgraph H of G such that H contains e and it is all-round or bipartite all-round.*

Proof. Suppose that an edge e is contained in a cycle C of length ℓ. If $\ell = 3, 4, 6$, then C itself is all-round or bipartite all-round, and we obtain the claim. Thus, we will show the cases of $\ell = 5$ and $\ell \geq 7$.

First, we show the case $\ell \geq 7$. To begin with, we remark that the cycle C has at least two chords, say f and g, otherwise, let v be an end vertex of the unique chord and then the induced subgraph removing v contains P_6 (or longer) as an induced subgraph. For convenience, we say a chord *separates* C *into* $(a, \ell - a)$ ($a \in \{2, 3, \dots, \lfloor \ell/2 \rfloor\}$) if the chord connects vertices in the distance a along C. The proof idea is an induction on ℓ, i.e., we show a shorter cycle containing e. Recall that C_3 is all-round (Proposition 2.20), and C_4 and C_6 are bipartite all-round (Proposition 2.21), while C_5 is not all-round (Lemma 2.22).

In the case of $\ell = 7$. If a chord separates C into $(2, 5)$, then e is contained in C_3 or C_6, and we obtain the claim. If both chords f and g separate C into $(3, 4)$. then we claim that $H = C + f + g$ is all-round. If f and g share a common vertex, then it is easy to observe that e is contained in C_4 or C_3. Otherwise, H is isomorphic to one of two graphs in Figure 2.6. In the left graph, we can find an ear decomposition consisting of C_4 (1-5-6-7-1, in Figure 2.6) and ears of length 3 (1-2-3-7) and 2 (3-4-5). Let H' denote the graph consisting of C_4 (1-5-6-7-1) and an ear of length 3 (1-2-3-7). Then, Lemma 2.27 implies that H' is bipartite all-round. Since H is given by H' and an ear of length two, Lemma 2.28 implies that H is all-round. In the right graph, we can find an ear decomposition consisting of C_4 (1-5-6-7-1), an ear of length 2 (5,4,7), and an ear of length 3 (1,2,3,4). Then, H is also all-round by a similar argument.

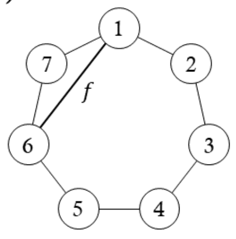
In the case of $\ell = 8$. Then, a chord possibly separates C into $(2, 6)$, $(3, 5)$ or $(4, 4)$. The cases of $(2, 6)$ and $(3, 5)$ are easy; in the case of $(2, 6)$, e is contained in C_3 or C_7, where the latter case is reduced to the above case of $\ell = 7$. In the case of $(3, 5)$, e is contained in C_4 or C_6. Finally, suppose that both f and g separate C into $(4, 4)$. Notice that f and g cannot share a common vertex in the case. Then, $H = C + f + g$ is isomorphic to one of two graphs in Figure 2.7. One graph (upper one) consists of C_4 (1-5-4-8-1) and two ears of length 3 (1-2-3-4 and 5-6-7-8), and it is all-round by Lemmas 2.26 and 2.28. The other consists of C_6 (1-5-4-3-7-8-1) and ears of length 2 (1-2-3 and 5-6-7), and it is bipartite all-round, by Lemma 2.27. Thus we obtain the claim in the case.

In the case of $\ell = 9$. Then, a chord possibly separates C into $(2, 7)$, $(3, 6)$ or $(4, 5)$. The cases of $(2, 7)$ or $(3, 6)$ are easy, since e is contained in C_3, C_8, C_4 or C_7. Suppose that each of f and g separate C into $(4, 5)$. If f and g share a common vertex, then e is contained in C_6. Otherwise, $H = C + f + g$ is isomorphic to one of three graphs in Figure 2.8. We can observe that each graph has an ear decomposition consisting of C_6 (1-2-3-4-5-6-1) with ears of length two and three. Then, each graph is all-round by Lemmas 2.26, 2.27 and 2.28.

In the case of $\ell \geq 10$. Suppose that a chord separates C into $(a, \ell - a)$ where $a \leq \ell - a$, then e is in C_{a+1} or $C_{\ell-a+1}$. Unless $e \in C_5$, i.e., $a = 4$ and $e \in C_{a+1}$, the case is reduced to a shorter cycle. Suppose $e \in C_{a+1}$ where $a = 4$. Then, $\ell - a + 1 \geq 7$ implies that $C_{\ell-a+1}$ contains another chord. Using the chord, we can reduce the case to a shorter cycle.

Next, we show the remaining case of $\ell = 5$. Suppose that e is contained in a cycle C

(2,5)



(3,4)

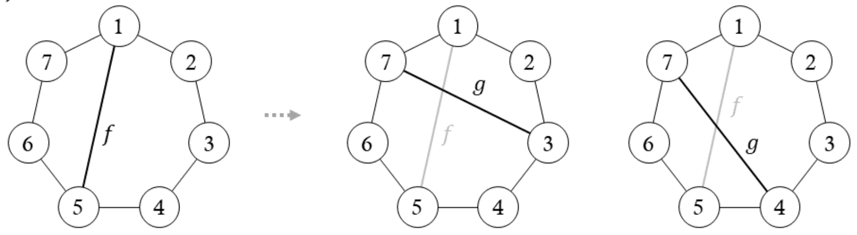


Figure: 2.6: $\ell = 7$.

of length five. We consider two cases: C is the unique cycle which contains e, or there is another cycle containing e.

First, we consider the former case. Since there is no cycle containing e other than C, C has no chord. Furthermore, since $G \neq C_5$, there exists a vertex $v \in V(C)$ that is contained in another cycle, which implies there exists an edge $f \in \delta(v) \setminus E(C)$. Since G is a two-edge-connected P_6-free graph, f is contained in a C_3: Otherwise G has a P_6 as an induced subgraph. Then, Lemma 2.26 implies that $C + C_3$ is all-round, and we obtain the claim in the case.

Next, we consider the second case. If there exists another cycle containing e and its length is not five, the argument for $\ell \neq 5$ establishes Lemma 2.29 for e. Suppose every cycle containing e has length five. Let C'' be one of the cycles containing e other than C. Then C and C'' has common edges. Let k be the number of the common edges. If $k = 2$ or 3, Lemma 2.28 implies that $C + C''$ is all-round. If $k = 1$, $C + C'' - e$ is a cycle of length eight. Then $C + C'' - e$ has a chord $f \neq e$ because G is P_6-free. By the argument of the case $\ell = 8$, $C + C'' + f$ is all-round. □

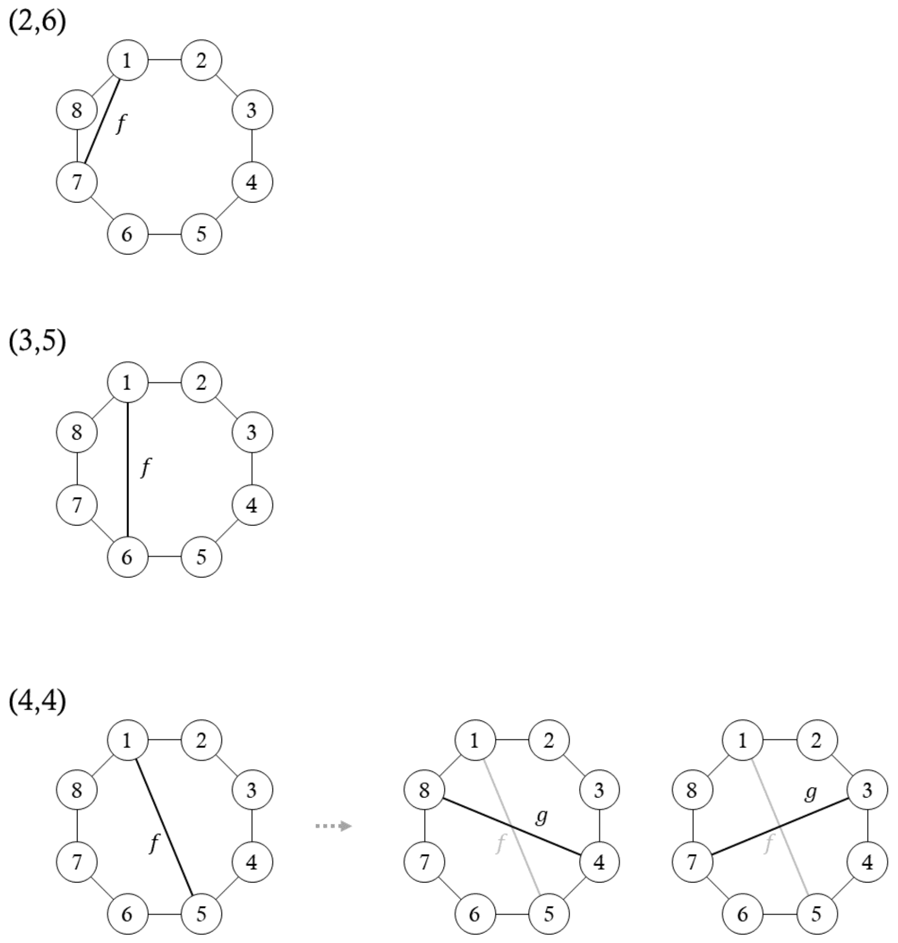


Figure: 2.7: $\ell = 8$.

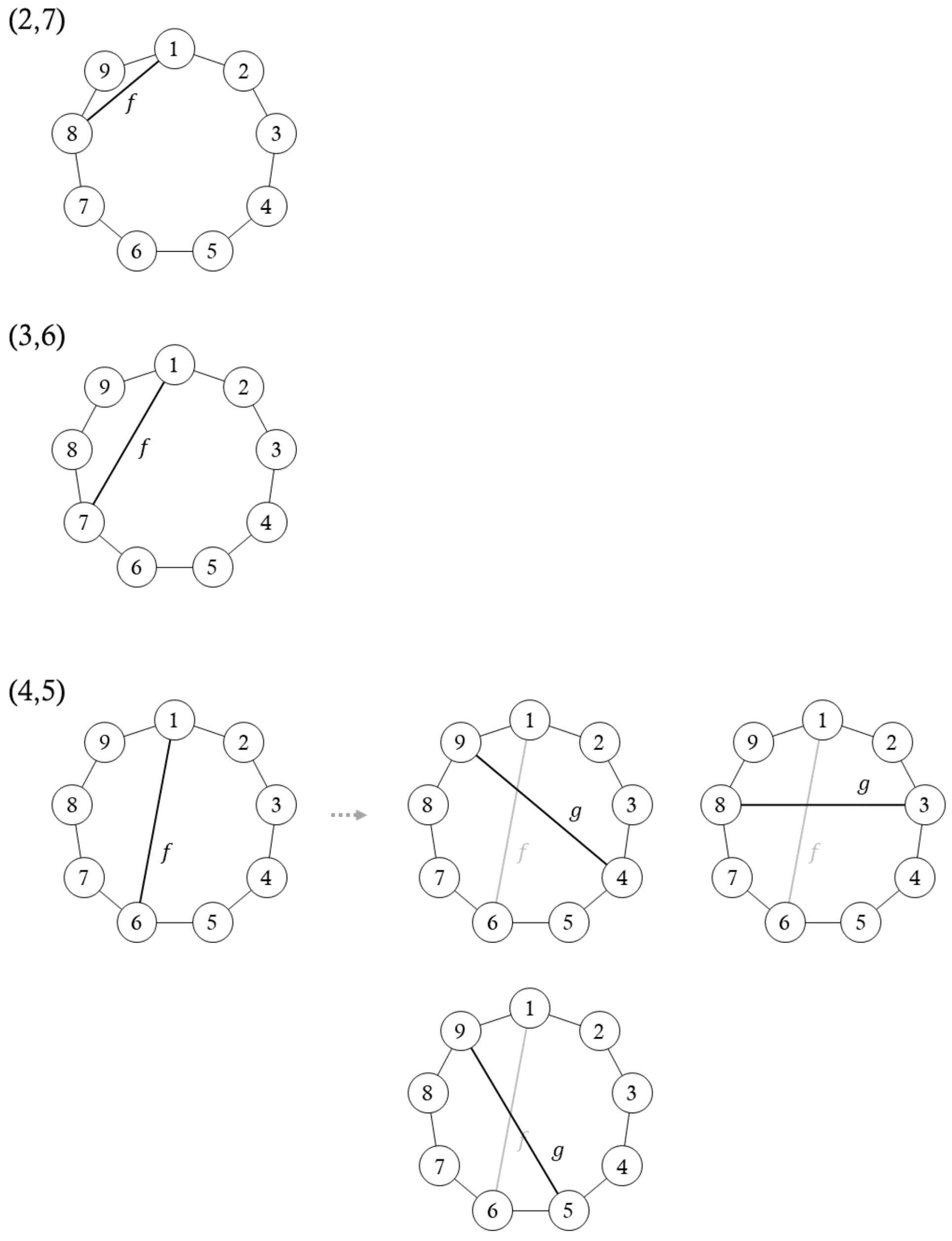


Figure: 2.8: $\ell = 9$.

2.6 Miscellaneous Discussions

This section remarks three related topics. We have introduced the notion of all-roundness in Section 2.5, and have shown that the PHC$_3$ problem is in P for *two-edge-connected* P_6-free or $C_{\geq 5}$-free graphs using an ear decomposition. Section 2.6.1 shows that the PHC$_3$ problem is in P for any P_6-free or $C_{\geq 5}$-free graphs, using their all-roundness with an extra argument on bridges. Section 2.6.2 remarks an all-roundness of dense graphs applying Lemma 2.25. Section 2.6.3 briefly explains a connection between the PHC problem and other problems such as the HC problem.

2.6.1 All-roundness of graphs with bridges

Theorem 2.30. *The* PHC$_3$ *problem is in* P *for P_6-free or $C_{\geq 5}$-free graphs.*

Proposition 2.31. *If a graph $G = (V, E)$ has a* PHC$_3$*, then its edge count vector $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ satisfies the condition that $x_e = 2$ for any bridge e in G.*

Proof. The edge count vector $\boldsymbol{x} \in \mathbb{Z}^E_{\geq 0}$ of any *closed* walk satisfies the condition that x_e is positive and even for any bridge $e \in E$. Since $x_e \leq 3$ for a PHC$_3$, we obtain the claim. □

Proposition 2.32. *Suppose that a graph $G = (V, E)$ has a vertex $v \in V$ such that $\delta(v) \subseteq B$ where $B \subseteq E$ denotes the set of bridges of G. Then, G has a* PHC$_3$ *only when the degree of v is odd.*

Proof. Suppose that every edge incident to a vertex v is a bridge of G. Proposition 2.31 implies that the visiting number (2.1) at v is equal to its degree if G has a PHC$_3$. It must be odd. □

Proof of Theorem 2.30. Without loss of generality, we may assume that an input graph $G = (V, E)$ is connected. We give an algorithmic proof, consisting of essentially two steps. Algorithm 2.6.1 shows the summary. As a preliminary step (Step 1 in Algorithm 2.6.1), we check the condition implied by Proposition 2.32; if G has a vertex v such that $\delta(v) \subseteq B$ where B denotes the set of bridges B and $d(v)$ is even, then G cannot have a PHC$_3$. It takes polynomial time, namely in $\mathcal{O}(|E|^2)$ time in a naive way.

The second step (Step 2 in Algorithm 2.6.1) is the heart of the proof. For convenience, let B denote the set of bridges of G, and let $D_1, \ldots, D_\ell$ denote two-edge-connected components, which are also found in polynomial time, namely in $\mathcal{O}(|E|^2)$ time in a naive way. Considering Proposition 2.31, let a map $f_i \colon V(D_i) \to \{0, 1, 2, 3\}$ for $i = 1, \ldots, \ell$ be given by

$$f_i(v) \equiv (2 - 2|B \cap \delta(v)|) \pmod 4 \tag{2.20}$$

Algorithm 2.6.1

```
INPUT: A connected $P_6$-free or $C_{\geq 5}$-free graph $G = (V, E)$.
QUESTION: If $G$ has a PHC$_3$?
Step 1. If $G$ has a vertex $v \in V$ such that $\delta(v) \subseteq B$ and $d(v)$ is even, then return FALSE,
        where $B$ denotes the set of bridges of $G$.
Step 2. For each two-edge-connected component $D_1, \dots, D_\ell$ of $G$,
    set $f_i(v) \equiv (2 - 2|B \cap \delta(v)|) \pmod 4$ for $v \in V(D_i)$.
    Case (a): $D_i$ is a cycle of length 5 ($C_5$). Unless $D_i$ has a mod-4 $f_i$-factor, return FALSE.
    Case (b): $D_i$ ($\neq C_5$) is non-bipartite. Unless $f_i$ satisfies the condition (2.13), return FALSE.
    Case (c): $D_i$ is bipartite. Unless $f_i$ satisfies the condition (2.16), return FALSE.
Step 3. Return TRUE.
```

for $v \in V(D_i)$. It is not difficult to observe that G has a PHC$_3$ if and only if every D_i has a mod-4 f_i-factor. Notice that each D_i is two-edge-connected and it is P_6-free or $C_{\geq 5}$-free, meaning that D_i is all-round or bipartite all-round unless D_i is a cycle graph of length 5 (C_5) by Theorems 2.18 and 2.19. If $D_i = C_5$, then we can check if D_i has a mod-4 f_i-factor in a constant time, by an exhaustive check of all assignments of $\{0, 1, 2, 3\}^5$. Unless $D_i \neq C_5$, we only need to check if f_i satisfies conditions (2.13), and (2.16) when D_i is bipartite. Propositions 2.14 and 2.15 imply the condition is necessary, and Theorems 2.18 and 2.19 imply that it suffices. Clearly, it takes only polynomial time, namely $\mathcal{O}(|V(D_i)|)$ time for each $i = 1, \dots, \ell$. □

Remark that Theorem 2.30 only claims that the *decision* problem is in P. If we know an appropriate ear decomposition then we can also *find* a PHC$_3$ in polynomial time by carefully tracing the proofs of Theorems 2.18, 2.19 and 2.30. It is an interesting question if we can find a PHC$_3$, beyond the existence, in polynomial time for P_6-free or $C_{\geq 5}$-free graphs.

2.6.2 All-roundness of dense graphs

This section shows another application of Lemma 2.25.

Proposition 2.33. *Let $G = (V, E)$ be a connected graph where $|V| \geq 3$ and the minimum degree of G is at least $|V|/2$. Then, G is all-round, or bipartite all-round if G is bipartite.*

Proof. We show that every edge $e = \{u, v\} \in E$ is contained in a cycle of length three or four, and then Lemma 2.25 implies that G is all-round. If $|V|$ is odd, the degree of each u and v is strictly greater than $|V|/2$ by the hypothesis. This implies that u and v

have a common neighbor w by the pigeon hole principle. Thus any edge is contained in a cycle of length three.

Suppose $|V|$ is even. If u and v have a common neighbor, then we obtain the claim. If it is not the case, we can observe that $|N(u) \setminus \{v\}| = |N(v) \setminus \{u\}| = |V|/2 - 1$ holds. Let $w \in N(u) \setminus \{v\}$, and then $d(w) \geq |V|/2$ implies that w is connected to a vertex in $N(v) \setminus \{u\}$ by the pigeon hole principle. Thus we obtain a cycle of the length four in the case. □

In fact, we can show the following stronger theorem with some complicated arguments (see [47] for the proof).

Theorem 2.34. *Let $G = (V, E)$ be a connected graph where $|V| \geq 4$ and the minimum degree of G is at least $|V|/3$. Then, G is all-round, or bipartite all-round if G is bipartite.*

2.6.3 Connection of Parity Hamiltonian, Hamiltonian, Eulerian

We remark the connection between the PHC problem and the HC problem, or other related topics[3]. In fact, we are concerned with the following generalized version of Problem 1.

Problem 2 (connected mod-d f-factor with edge capacity constraints)**.** *Given a graph $G = (V, E)$, a map $z\colon E \to \mathbb{Z}_{\geq 0}$, a positive integer d, and a map $f\colon V \to \mathbb{Z}_{\geq 0}$, find $\boldsymbol{x} \in \mathbb{Z}_{\geq 0}^E$ satisfying the conditions that*

$$\sum_{e \in \delta(v)} x_e \equiv f(v) \pmod{d} \qquad \textit{for any } v \in V, \tag{2.21}$$

$$(G, \boldsymbol{x}) \textit{ is connected.} \tag{2.22}$$

$$x_e \leq z(e) \qquad \textit{for any } e \in E. \tag{2.23}$$

The PHC problem is given by setting $d = 4$ and $f(v) = 2$ for any $v \in V$ with an appropriate capacity constraint. Problem 1 is given by setting $d = 4$ and $z(e) = 3$ for any $e \in E$. The Hamiltonian cycle problem is represented by setting $d = n$, $f(v) = 2$ for any $v \in V$, and $z(e) = 1$ for any $e \in E$. We remark that the Hamiltonian cycle problem is also given (in its original form) by setting $d = \infty$, $f(v) = 2$ for any $v \in V$, and removing (2.23) (or setting $z(e) = \infty$ for any $e \in E$). The Eulerian cycle problem is given by setting $d = 2$, $f(v) = 0$ for any $v \in V$, and replacing (2.23) with $x_e = 1$ for any $e \in E$.

A two-factor plays a key role in the arguments of the HC problem in cubic graphs, where the connectivity constraint is relaxed [29, 30, 8, 9]. Motivated by a *connected*

[3]The argument of this section might be appropriate appearing in the section of Concluding Remarks. However, it is too long to put there, and we discuss just before the concluding remark.

“factor,” this paper has investigated connected mod-4 factors. A mod-d factor for prime d is an interesting future work.

Chapter 3

The PHC Problem in Directed Graphs

In this chapter we consider the PHC problem in *directed* graphs. First we introduce terminologies necessary for this chapter. Then we give a characterization of digraphs which have PHC's in Section 3.2 and show the problem is solved in polynomial time. In Section 3.2.1, we propose a linear time algorithm to solve the PHC problem based on the characterization. In Section 3.3 we extend the PHC problem to "modulo p" version, and give a characterization which is decidable in polynomial time.

3.1 Preliminaries

A *directed graph* (*digraph* for short) $D = (V, A)$ is given by a vertex set V and an arc set A (we write $V(D)$ and $A(D)$ to clarify which graph we are discussing). Let $\delta^+(v)$ (resp. $\delta^-(v)$) for $v \in V$ denote the set of *outgoing* (resp. *incoming*) arcs; that is, arcs that leave v (resp. enter v). Their sizes $|\delta^+(v)|$ and $|\delta^-(v)|$ are called the *out-degree* and the *in-degree* of v, respectively.

A *directed walk* is a sequence of vertices and arcs $v_0 a_1 \cdots a_\ell v_\ell$, where $a_i = (v_{i-1}, v_i) \in A$ for each i $(1 \leq i \leq \ell)$. A directed walk is *closed* if $v_\ell = v_0$. A *directed path* is a directed walk which contains each vertex at most once except the start vertex v_0 and the end vertex v_ℓ. A directed closed path is called a *directed cycle.* A digraph D is *strongly connected* if there exists a directed path from u to v for any pair of vertices $u, v \in V(D)$. For convenience, we often represent a directed closed walk by an integer vector $\tilde{\boldsymbol{x}} \in \mathbb{Z}^A_{\geq 0}$, in which $\tilde{x}(a)$ denotes the number of occurrences of arc a in the closed walk.

A *cycle basis* of D is a set of directed cycles $\{C_1, C_2, \ldots, C_k\}$ which satisfies conditions: (i) Their incidence vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \{0, 1\}^A$ are linearly independent over GF(2), and (ii) For the incidence vector $\boldsymbol{\gamma}$ of each cycle in D[1], there exist coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{0, 1\}$ such that $\boldsymbol{\gamma} = \alpha_1 \boldsymbol{c}_1 + \alpha_2 \boldsymbol{c}_2 + \cdots + \alpha_k \boldsymbol{c}_k$ over GF(2). We call each

[1] $\boldsymbol{\gamma}$ can be an incidence vector of a cycle which is not directed.

cycle C_i a *fundamental cycle*. It is known that the size k of a cycle basis is equal to $|A(D)| - |V(D)| + 1$ [3]. A cycle basis of a digraph is found in linear time [3, 53].

For a digraph D, let $M^+ = [m^+_{va}]$ and $M^- = [m^-_{va}] \in \{0,1\}^{|V|\times|A|}$ be matrices respectively defined by

$$m^+_{va} = \begin{cases} 1 & \text{if } a \in \delta^+(v), \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad m^-_{va} = \begin{cases} 1 & \text{if } a \in \delta^-(v), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\tilde{\boldsymbol{x}} \in \mathbb{Z}^A_{\geq 0}$ be the edge count vector of a directed closed walk in a digraph. Then $\tilde{\boldsymbol{x}}$ is the edge count vector of a PHC if and only if $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \mathbf{1} \pmod 2$ holds.

We define a matrix M over $\{0,1\}^{2|V|\times|A|}$ by

$$M = \begin{bmatrix} M^+ \\ M^- \end{bmatrix}.$$

Let $\{C_1, C_2, \ldots, C_k\}$ $(k = |A| - |V| + 1)$ be a cycle basis of D and let $\boldsymbol{c}_1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \{0,1\}^A$ be their incidence vectors. Let $R = [\boldsymbol{c}_1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_k]$. We define a matrix $Q \in \{0,1\}^{|V|\times k}$ by

$$Q = M^+R. \tag{3.1}$$

Remark that $Q = M^-R$ holds, since for each i, the column vector $\boldsymbol{q}_i \in \{0,1\}^V$ of Q is a vector such that $q_i(v) = 1$ if and only if C_i contains $v \in V$.

3.2 Characterization

The following is the characterization for the PHC problem in digraphs.

Theorem 3.1. *Let $D = (V, A)$ be a strongly connected digraph. The following three conditions are equivalent:*

(a) *D has a PHC,*

(b) *$M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ has a solution $\boldsymbol{x} \in \{0,1\}^A$,*

(c) *$Q\boldsymbol{\beta} \equiv \mathbf{1} \pmod 2$ has a solution $\boldsymbol{\beta} \in \{0,1\}^k$,*

where $k = |A| - |V| + 1$ and $\mathbf{1}$ denotes the all-one vector.

Proof. The proofs of (c) $\Rightarrow$ (a) and (a) $\Rightarrow$ (b) are easy, while the other way (b) $\Rightarrow$ (a) and (a) $\Rightarrow$ (c), as well as (b) $\Rightarrow$ (c) directly are not trivial. First we show (a) $\Leftrightarrow$ (b), then we show (a) $\Leftrightarrow$ (c).

(a) $\Rightarrow$ (b) Let $\tilde{\boldsymbol{x}} \in \mathbb{Z}^A_{\geq 0}$ be a vector in which $\tilde{x}(a)$ denotes the number of uses of $a \in A$ in a PHC. By the parity condition of PHC, we have $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \mathbf{1} \pmod 2$. Then let $\boldsymbol{x} \in \{0,1\}^A$ be defined by $\boldsymbol{x} \equiv \tilde{\boldsymbol{x}} \pmod 2$, we have $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \mathbf{1} \pmod 2$, and thus $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$.

(b) $\Rightarrow$ (a) Suppose that $\boldsymbol{x} \in \{0,1\}^A$ is a solution of $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$, then we explain how to construct a PHC. Remark that a graph indicated by $\boldsymbol{x}$ satisfies the parity condition of the visiting number on each vertex, but may not satisfy the Eulerian condition, meaning that $\sum_{a\in\delta^+(v)} x(a) = \sum_{a\in\delta^-(v)} x(a)$ may not hold for some vertex v, and connectivity.

First, we construct $\boldsymbol{x}' \in \mathbb{Z}^A_{\geq 0}$ satisfying both of the parity condition $M\boldsymbol{x}' \equiv \mathbf{1} \pmod 2$ and the Eulerian condition $\sum_{a\in\delta^+(v)} x'(a) = \sum_{a\in\delta^-(v)} x'(a)$ for each $v \in V$. Let $\phi(v) = \sum_{a\in\delta^+(v)} x(a) - \sum_{a\in\delta^-(v)} x(a)$ for each $v \in V$, denoting the difference between out-degree and in-degree of v in $\boldsymbol{x}$. Then $\boldsymbol{x}$ is Eulerian if and only if $\phi(v) = 0$ for all v. Notice that $\sum_{v\in V} \phi(v) = 0$ holds since the total of out-degrees is equal to the total of in-degrees. We also remark that $\phi(v)$ is even for each $v \in V$, since $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ implies that both of out-degree ($\sum_{a\in\delta^+(v)} x(a)$) and in-degree ($\sum_{a\in\delta^-(v)} x(a)$) are odd. Then we apply the following Procedure 1 to $\boldsymbol{x}$:

Procedure 1.

1. Find $u, v \in V$ such that $\phi(u) < 0$ and $\phi(v) > 0$.
2. Find a directed path P from u to v (P always exists since D is strongly connected).
3. $x(a) := x(a) + 2$ for each $a \in A(P)$.

Procedure 1 preserves the parity condition $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$, and decreases the value of $\sum_{v\in V} |\phi(v)|$ (by four). By recursively applying Procedure 1 until $\sum_{v\in V} |\phi(v)|$ is zero, we obtain a desired closed walk $\boldsymbol{x}'$.

If $\boldsymbol{x}'$ suggests a connected walk, we obtain a PHC. Suppose that $\boldsymbol{x}'$ is not connected. Then we apply the following Procedure 2 to $\boldsymbol{x}'$:

Procedure 2.

1. Find $u, v \in V$ which are in distinct connected components.
2. Find directed paths P from u to v and P' from v to u.
3. $x'(a) := x'(a) + 2$ for each $a \in A(P) \cup A(P')$.

Procedure 2 preserves the parity condition $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ and the Eulerian condition, and decreases the number of connected components. By recursively applying Procedure 2, we obtain a connected walk, which is in fact a PHC.

(c) $\Rightarrow$ (a) We construct a PHC from the solution $\boldsymbol{\beta} \in \{0,1\}^k$. Let

$$\alpha_i = \begin{cases} 1 & \text{if } \beta_i = 1, \\ 2 & \text{if } \beta_i = 0, \end{cases} \tag{3.2}$$

and set $\tilde{\boldsymbol{x}} = R\boldsymbol{\alpha}$. Notice that $\tilde{\boldsymbol{x}}$ indicates a closed walk since it is a sum of cycles. We claim that the closed walk indicated by $\tilde{\boldsymbol{x}}$, say γ, is a PHC. The walk γ is connected since γ uses all edges of D at least once, and D is strongly connected. Then, we have

$$M^{+}\tilde{\boldsymbol{x}} = M^{+}R\boldsymbol{\alpha} = Q\boldsymbol{\alpha} \equiv Q\boldsymbol{\beta} \equiv \mathbf{1} \pmod 2,$$

where the second last congruence comes from (3.2) and the last congruence follows from the assumption that $\boldsymbol{\beta}$ is a solution of $Q\boldsymbol{\beta} \equiv \mathbf{1} \pmod 2$. Hence, γ satisfies the parity condition, and thus γ is a PHC.

(a) $\Rightarrow$ **(c)** To show the necessity we show the following lemma.

Lemma 3.2. *Let γ be any closed walk of D, and let $\tilde{\boldsymbol{x}} \in \mathbb{Z}^{A}_{\geq 0}$ be a vector in which $\tilde{x}(a)$ denotes the number of uses of $a \in A$ in γ. Then $R\boldsymbol{\beta} \equiv \tilde{\boldsymbol{x}} \pmod 2$ has a solution $\boldsymbol{\beta} \in \{0,1\}^{k}$.*

Proof. Since γ is an Eulerian walk in a multi-digraph, meaning that γ consists of simple cycles, $\tilde{\boldsymbol{x}}$ is represented by

$$\tilde{\boldsymbol{x}} = \sum_{j=1}^{\ell} \alpha_j \boldsymbol{\gamma}_j, \tag{3.3}$$

with appropriate positive integer ℓ, where each α_j is a nonnegative integer and each $\boldsymbol{\gamma}_j \in \{0,1\}^{A}$ is the incidence vector of a directed cycle of D. Remark that each $\boldsymbol{\gamma}_j$ is represented by a linear combination of incidence vectors of fundamental cycles $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$, such that $\boldsymbol{\gamma}_j \equiv \sum_{i=1}^{k} \beta'_{ij}\boldsymbol{c}_i \pmod 2$ for some 0-1 coefficients β'_{ij} for each j. Let $\beta_i \in \{0,1\}$ be defined by $\beta_i \equiv \sum_{j=1}^{\ell} \beta'_{ij}\alpha_j \pmod 2$, then

$$\tilde{\boldsymbol{x}} \equiv \sum_{j=1}^{\ell} \alpha_j \sum_{i=1}^{k} \beta'_{ij}\boldsymbol{c}_i \equiv \sum_{i=1}^{k} \beta_i \boldsymbol{c}_i \pmod 2$$

holds. Notice that $\sum_{i=1}^{k} \beta_i \boldsymbol{c}_i = R\boldsymbol{\beta}$, then we obtain the claim. □

Suppose that γ is a PHC of D, and that $\tilde{\boldsymbol{x}} \in \mathbb{Z}^{A}_{\geq 0}$ is a vector in which $\tilde{x}(a)$ denotes the number of uses of $a \in A$ in γ. Since a PHC is a closed walk, Lemma 3.2 implies that there is a vector $\boldsymbol{\beta} \in \{0,1\}^{k}$ such that $\tilde{\boldsymbol{x}} \equiv R\boldsymbol{\beta} \pmod 2$. Then

$$Q\boldsymbol{\beta} = M^{+}R\boldsymbol{\beta} \equiv M^{+}\tilde{\boldsymbol{x}} \equiv \mathbf{1} \pmod 2,$$

where the last congruence comes from the fact that $\tilde{\boldsymbol{x}}$ indicates a PHC. □

3.2.1 Recognition in Linear Time

By Theorem 3.1, we can decide if a given directed graph D has a PHC in polynomial time, by solving a linear system $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ or $Q\boldsymbol{\beta} \equiv 1 \pmod 2$, both of which cost $\mathcal{O}(|V||A|^2)$ time. In this section we improve the time complexity to linear.

For a digraph $D = (V, A)$, let $BG(D)$ be an undirected bipartite graph $(V^+, V^-; E)$, where V^+ and V^- are the copies of V and $E = \{u^+v^- \mid (u, v) \in A\}$. It is easy to see that the class of D's and $BG(D)$'s have a one-to-one corresponding. Observe that M of D coincides with the incidence matrix of $BG(D)$, and hence $|E| = |A|$.

Lemma 3.3. *$M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ has a solution if and only if $BG(D) = (V^+, V^-; E)$ has a $(V^+ \cup V^-)$-join.*

Proof. Let F be any subset of E, and let $\boldsymbol{x}_F \in \{0,1\}^E$ be its incidence vector. Since M is the incidence matrix of $BG(D)$, the v-th entry of the vector $M\boldsymbol{x}_F$, $(M\boldsymbol{x}_F)_v$, denotes the degree of v in $\boldsymbol{x}_F$. Let $\boldsymbol{x} \in \{0,1\}^E$ be a solution of $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$. Then $\boldsymbol{x}$ indicates a subgraph of $BG(D)$ in which every vertex has odd degree, which is a $(V^+ \cup V^-)$-join of $BG(D)$. Conversely, if $\boldsymbol{x}_F \in \{0,1\}^E$ is the incidence vector of a $(V^+ \cup V^-)$-join F, $\boldsymbol{x}_F$ satisfies $M\boldsymbol{x}_F \equiv \mathbf{1} \pmod 2$. □

Since both $BG(D)$ and a T-join are constructed in linear time, we see the following.

Theorem 3.4. *The PHC problem in digraphs is solved in linear time.*

Thus we have verified that it is decided in linear time if the input directed graph has a PHC. Finally we argue the time complexity to construct a PHC in a given directed graph D. The proof of Lemma 3.3 implies that we can obtain a solution $\boldsymbol{x} \in \{0,1\}^A$ of $M\boldsymbol{x} \equiv \mathbf{1} \pmod 2$ by finding a $(V^+ \cup V^-)$-join of $BG(D)$. Once we obtain a solution $\boldsymbol{x}$, we can construct a PHC according to the proof of Theorem 3.1 for (b) $\Rightarrow$ (a). The algorithm is summarized in Algorithm 3.2.1.

It takes $\mathcal{O}(|A|)$ time in line 1. In line 2, we repeatedly find paths, each path is found in $\mathcal{O}(|A|)$ time and repeated $\mathcal{O}(|A|)$ time thus $\mathcal{O}(|A|^2)$ time in total. In line 3, we repeatedly find pairs of paths, each is done in $\mathcal{O}(|A|)$ and repeated $\mathcal{O}(|V|)$ time, thus $\mathcal{O}(|V||A|)$ time in total. Consequently, the time complexity of Algorithm 3.2.1 is $\mathcal{O}(|A|^2)$.

Algorithm 3.2.1 Constructing a PHC in a digraph.

```
1: Find a $(V^+ \cup V^-)$-join $J$ of $BG(D)$ and $\boldsymbol{x} \leftarrow \chi_J$
2: Repeat Procedure 1 until $\boldsymbol{x}$ satisfies the Eulerian condition
3: Repeat Procedure 2 until $\boldsymbol{x}$ becomes connected
4: return $\boldsymbol{x}$
```

3.3 Extension to $\mathrm{GF}(p)$

This section is concerned with the following problem, a generalization of the PHC problem: Given a digraph D and an integer p and an integer vector $\boldsymbol{r} \in \{0, 1, \ldots, p-1\}^A$, decide if there exists a closed walk which visits each vertex v $r(v)$ times modulo p. In other words, the problem asks to find a connected closed walk that satisfies the condition $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod p$, where $\tilde{\boldsymbol{x}} \in \mathbb{Z}^A_{\geq 0}$ is a vector in which $\tilde{x}(a)$ denotes the number of uses of arc a in the closed walk. It is easy to see that the problem is equivalent to the PHC problem when $p = 2$ and $\boldsymbol{r} = \mathbf{1}$. The following is the characterization for the generalized problem which is similar to Theorem 3.1 (b).

Theorem 3.5. *A strongly connected digraph D has a connected closed walk which satisfies $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod p$ if and only if $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod p$ has a solution $\boldsymbol{x} \in \{0, \ldots, p-1\}^A$.*

Proof. The proof is similar to (a) $\Leftrightarrow$ (b) of Theorem 3.1.

Necessity. Let $\tilde{\boldsymbol{x}} \in \mathbb{Z}^A_{\geq 0}$ be a vector in which $\tilde{x}(a)$ denotes the number of uses of $a \in A$ in a connected closed walk which satisfies $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod p$. Then let $\boldsymbol{x} \in \{0, \ldots, p-1\}^A$ be defined by $\boldsymbol{x} \equiv \tilde{\boldsymbol{x}} \pmod p$, we have $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod p$.

Sufficiency. Suppose that $\boldsymbol{x} \in \{0, \ldots, p-1\}^A$ is a solution of $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod p$, then we explain how to construct a closed walk which satisfies $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod p$. Remark that a graph indicated by $\boldsymbol{x}$ may not satisfy the Eulerian condition, meaning that $\sum_{a\in\delta^+(v)} x(a) = \sum_{a\in\delta^-(v)} x(a)$ may not hold for some vertex v.

First, we construct $\boldsymbol{x}' \in \mathbb{Z}^A_{\geq 0}$ satisfying the condition $M^+\boldsymbol{x}' \equiv M^-\boldsymbol{x}' \equiv \boldsymbol{r} \pmod p$ and the Eulerian condition $\sum_{a\in\delta^+(v)} x'(a) = \sum_{a\in\delta^-(v)} x'(a)$ for each $v \in V$. Let $\phi(v) = \sum_{a\in\delta^+(v)} x(a) - \sum_{a\in\delta^-(v)} x(a)$ for each $v \in V$, denoting the difference between out-degree and in-degree of v in $\boldsymbol{x}$. Then $\boldsymbol{x}$ is Eulerian if and only if $\phi(v) = 0$ for all v. Notice that $\sum_{v\in V} \phi(v) = 0$ holds since the total of out-degrees is equal to the total of in-degrees. We also remark that $\phi(v)$ is a multiple of p for each $v \in V$, since $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod p$ implies that both of out-degree ($\sum_{a\in\delta^+(v)} x(a)$) and in-degree ($\sum_{a\in\delta^-(v)} x(a)$) are $r(v)$ modulo p. Then we apply the following Procedure 1' to $\boldsymbol{x}$:

Procedure 1'.

1. Find $u, v \in V$ such that $\phi(u) < 0$ and $\phi(v) > 0$.
2. Find a directed path P from u to v (P always exists since D is strongly connected).
3. $x(a) := x(a) + p$ for each $a \in A(P)$.

Procedure 1' preserves the condition $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod p$, and decreases the value of $\sum_{v\in V} |\phi(v)|$ (by $2p$). By recursively applying Procedure 1' until $\sum_{v\in V} |\phi(v)|$ is zero, we obtain a desired closed walk $\boldsymbol{x}'$.

If $\boldsymbol{x}'$ suggests a connected walk, we obtain a closed walk which satisfies $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod{p}$. Suppose that $\boldsymbol{x}'$ is not connected. Then we apply the following Procedure 2' to $\boldsymbol{x}'$:

Procedure 2'.

1. Find $u, v \in V$ which are in distinct connected components.
2. Find directed paths P from u to v and P' from v to u.
3. $x'(a) := x'(a) + p$ for each $a \in A(P) \cup A(P')$.

Procedure 2' preserves the condition $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod{p}$ and the Eulerian condition, and decreases the number of connected components. By recursively applying Procedure 2', we obtain a connected closed walk which satisfies $M^+\tilde{\boldsymbol{x}} \equiv M^-\tilde{\boldsymbol{x}} \equiv \boldsymbol{r} \pmod{p}$. □

If p is prime or power of a prime, the linear system $M^+\boldsymbol{x} \equiv M^-\boldsymbol{x} \equiv \boldsymbol{r} \pmod{p}$ is solved over GF(p), and we obtain a desired closed walk in polynomial time. Otherwise GF(p) is not a field, and we need an extra observation to solve the equation efficiently.

3.4 Open problem: The PHC orientation

In Chapter 2 we have investigated the PHC problem in undirected graphs. Now it is natural to ask how the problem in undirected graphs and directed graphs are related. The *PHC orientation problem* is a problem to decide if a given undirected graph has an orientation such that the resulted directed graph admits a PHC. Figure. 3.1 shows an example of an undirected graph which has a PHC, but does not admit a PHC orientation. It is open if the PHC orientation problem is solved in polynomial time.

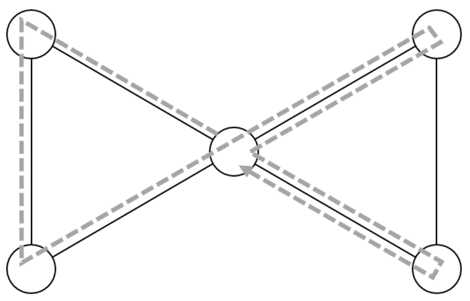


Figure: 3.1: An undirected graph which has a PHC but does not have a PHC orientation. The dotted line indicates an undirected PHC.

Chapter 4

The Hamiltonicity of Covering Graphs

In this chapter we study the HC problem in *covering graphs*. After and introducing some notations and fundamental properties of covering graphs in Section 4.1, we mention our main results of this chapter in Sections 4.2 and 4.3.

4.1 Preliminaries

We use the same notations as preceding chapters for the fundamental graph terminologies. A *voltage graph* is a triple (Γ, G, σ) where Γ is a graph, G is a group, and $\sigma\colon E(\Gamma) \to G$ is a mapping which assigns an element of G to each edge of Γ[1]. We call Γ the *base graph* and $\sigma(e)$ the *label* of e. We will assume Γ is a connected directed graph, and if $e = (u, v) \in E(\Gamma)$, then the inverse edge $e^{-1} = (v, u)$ is also in $E(\Gamma)$. In fact, we will assume the underlying graph of Γ is a tree. Because we want the voltage graph to be undirected, we require, $\sigma(e^{-1}) = \sigma(e)^{-1}$ for every edge $e \in E(\Gamma)$. We also allow Γ to have self-loops, and if $e = (u, u)$ is a loop, we sometimes say $\sigma(e)$ is the label on u.

The *covering graph* of a voltage graph (Γ, G, σ) is a graph with

- the vertex set $V(\Gamma) \times G$, and
- the edge set $E(\Gamma) \times G$. If $e = (u, v) \in E(\Gamma)$ and $a \in G$, (e, a) is the edge which leaves the vertex (u, a) and enters (v, ag) where $g = \sigma(e)$. Because $\sigma(e^{-1}) = \sigma(e)^{-1}$, the covering graph also has the edge leaving (v, ag) and entering (u, a).

We write Γ^σ to denote the covering graph generated from (Γ, G, σ). Instead of writing (v, a) and (e, a), we often use short-hand notations v_a and e_a, respectively. For a vertex v_a, we call a the *level* of v_a. As a simple example, suppose Γ is a path of length two with the vertex set $\{u, v\}$ and each vertex having a self-loop. Let $G = \mathbb{Z}_5$, $\sigma(u, u) = 1, \sigma(v, v) = 2$

[1]Note that we use the symbol G for representing *groups*, not for graphs. This notation is used throughout this chapter.

and $\sigma(u,v) = 0$. Then, the covering graph of this voltage graph is isomorphic to the Petersen graph.

Every vertex $v \in V(\Gamma)$ has $|G|$ copies in the covering graph. A set of copies of a vertex v, $\bigcup_{g\in G}\{v_g\}$ is called a *fiber over* v. Sometimes by the fiber over v, we actually will understand the subgraph of Γ^σ induced by vertices in the fiber over v. Similarly, a set of copies of an edge e, $\bigcup_{g\in G}\{e_g\}$ is called a *fiber over* e.

Throughout this paper, $G = \mathbb{Z}_p$, the cyclic group of order p. We represent the elements of $\mathbb{Z}_p$ by $0, 1, \ldots, p-1$, and the operator of the group by "+" and "-", respectively.

We use the following proposition, whose proof is obvious.

Proposition 4.1 (Invariance under the label shift)**.** *Let $(\Gamma, \mathbb{Z}_p, \sigma)$ be a voltage graph. Let $F \subseteq E(\Gamma)$ be a minimal edge cut of Γ. Let U and V be the vertex sets of the two components of $\Gamma - F$, and $\{F_+, F_-\}$ be the partition of F such that $F_+ = \{(u,v) \in F \mid u \in U, v \in V\}$ and $F_- = \{(v,u) \in F \mid u \in U, v \in V\}$. For $a \in \mathbb{Z}_p$, define a voltage assignment σ_a as*

$$\sigma_a(e) = \begin{cases} \sigma(e) + a & e \in F_+, \\ \sigma(e) - a & e \in F_-, \\ \sigma(e) & e \notin F. \end{cases} \tag{4.1}$$

Then $\Gamma^\sigma \simeq \Gamma^{\sigma_a}$ for any $a \in \mathbb{Z}_p$.

By the previous proposition, if e is a bridge in Γ, we can assume without loss of generality that $\sigma(e) = 0$. Since in this paper the underlying graph of Γ is a bi-directed tree, we assume $\sigma(e) = 0$ for every $e = (u,v)$ when $u \neq v$.

The following is also an useful observation.

Proposition 4.2 (Invariance under multiplication for cyclic groups)**.** *Let $(\Gamma, \mathbb{Z}_p, \sigma)$ be a voltage graph. For an integer d define a voltage assignment σ' as*

$$\sigma'(e) = d \cdot \sigma(e) \bmod p, e \in E(\Gamma). \tag{4.2}$$

If d is coprime to p, then $\Gamma^\sigma \simeq \Gamma^{\sigma'}$.

The result from [4] about the Hamiltonicity of the cartesian product of a cycle and a tree can be restated in the following form.

Theorem 4.3 ([4])**.** *Let Γ be a bi-directed tree with a self-loop at each vertex and let L be the set of self-loops. Let $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$ be defined by*

$$\sigma(e) = \begin{cases} 1 & e \in L, \\ 0 & e \notin L. \end{cases} \tag{4.3}$$

Then Γ^σ is Hamiltonian if and only if $p \geq \Delta$, where Δ is the maximum degree of $\Gamma - L$.

4.2 The First Extension : The Same Label at Both Ends

In this section we give our first extension of Theorem 4.3.

Theorem 4.4. *Let Γ be a bi-directed tree with a self-loop at each vertex and let L be the set of self-loops. Let $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$. Suppose the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ satisfies the following conditions:*

- *There exists a system of paths $P_1, P_2, \ldots, P_k$ such that $\{E(P_1), E(P_2), \ldots, E(P_k)\}$ is a partition of $E(\Gamma) \setminus L$, P_i and P_j are internally vertex disjoint for $i \neq j$, and the self-loops of the two end-vertices of P_i have the same label for each i,*
- *$\sigma(v, v)$ is coprime to p for every $v \in V(\Gamma)$.*

Then the covering graph Γ^σ is Hamiltonian if and only if $p \geq \Delta$, where Δ is the maximum degree of $\Gamma - L$.

Figure 4.1 shows an example of a graph which satisfies the conditions in Theorem 4.4. One can see that Theorem 4.4 is an extension of Theorem 4.3 by restricting σ to be the all-one label, since it is trivial to cover a tree with a system of pairwise internally vertex disjoint paths.

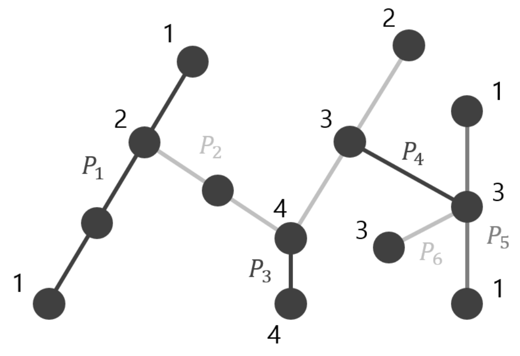


Figure: 4.1: A graph which satisfies the conditions in Theorem 4.4. The number at a vertex denotes the label of its self-loop (the self-loops are not drawn). If a vertex has no number next to it, it means its self-loop can have any label coprime to p. Then, $P_1, P_2, P_3, P_4, P_5, P_6$ is a system of paths satisfying the conditions in Theorem 4.4.

The first condition in Theorem 4.4, requires the two ends of each path to have the same label on the self-loops. This condition is necessary as there are voltage graphs which do not satisfy this condition and their covering graph is not Hamiltonian; for example, consider the example giving the Petersen graph.

Before we prove Theorem 4.4, we prove its special case which will be the base case in the main proof.

Lemma 4.5. *Let Γ be a bi-directed path with a self-loop at each vertex. Suppose $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$ satisfies the following:*

- *$\sigma(u,u) = \sigma(v,v)$ where u and v are the leaves of Γ,*
- *$\sigma(w,w)$ is coprime to p for every $w \in V(\Gamma)$.*

Then Γ^σ is Hamiltonian if and only if $p \geq 2$.

Proof. It is obvious that Γ^σ cannot be Hamiltonian when $p = 1$. This proves the necessity. In the remaining part we prove the sufficiency.

We may assume $\sigma(u,u) = \sigma(v,v) = 1$, by Proposition 4.2. Just for the convenience of explanation, let us suppose Γ is drawn horizontally, u lies on the left-hand side, and v lies on the right-hand side. Our strategy to construct a Hamiltonian cycle of Γ^σ is as follows. We call it the *billiard* strategy (see Figure 4.2): Start by considering the u_0-u_1 Hamiltonian path of the fiber over u, leaving its two ends u_0 and u_1 open. Extend the path to the next fiber on the right from these ends to their corresponding vertices in this fiber. Now include all remaining vertices of this fiber onto the constructed path by adding them in clockwise (or counter-clockwise) order from these starting vertices. This process will create new ends in this fiber, which are extended to next fiber to the right. Repeat this process until we get to the fiber over v. In this fiber we close the path into a Hamiltonian cycle of Γ^σ. We show this is always possible. Let $w^{(1)}, w^{(2)}, \dots, w^{(k)}$ be the vertices of $\Gamma - \{u,v\}$ named in the order as they appear on Γ starting from u. We first construct a u_0-u_1 Hamiltonian path $u_0 u_{p-1} u_{p-2} \dots u_2 u_1$. Then add edges $u_0 w_0^{(1)}$ and $u_1 w_1^{(1)}$. Suppose $\sigma(w^{(1)}, w^{(1)}) = a$. Since we have assumed that a is coprime to p, the fiber over $w^{(1)}$ (considered as a graph) is isomorphic to a simple cycle. Extend the constructed path by adding a path from $w_0^{(1)}$ to $w_{1-a}^{(1)}$ and from $w_1^{(1)}$ to $w_{-a}^{(1)}$ in this fiber, respectively. It is easy to see that this is always possible in such a way that all the vertices in the fiber over $w^{(1)}$ are on the so constructed path. Next, add edges $w_{1-a}^{(1)} w_{1-a}^{(2)}$ and $w_{-a}^{(1)} w_{-a}^{(2)}$. Suppose $\sigma(w^{(2)}, w^{(2)}) = b$. Repeat the same as in the fiber before, that is, extend the path by adding a path from $w_{1-a}^{(2)}$ to $w_{-a-b}^{(2)}$ and from $w_{-a}^{(2)}$ to $w_{1-a-b}^{(2)}$ in this fiber. An important fact here is that the two new ends have the difference of their levels equal to 1. This difference is preserved in the fiber over v as well. Now, since $\sigma(v,v) = 1$, the last two ends in this fiber can be joined by a Hamiltonian path (in this fiber), hence completing the whole path into a Hamiltonian cycle of Γ^σ. □

To prove Theorem 4.4, we also need the following proposition.

Proposition 4.6. *Suppose Γ is a bi-directed tree with a self-loop at every vertex and $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$. If the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the*

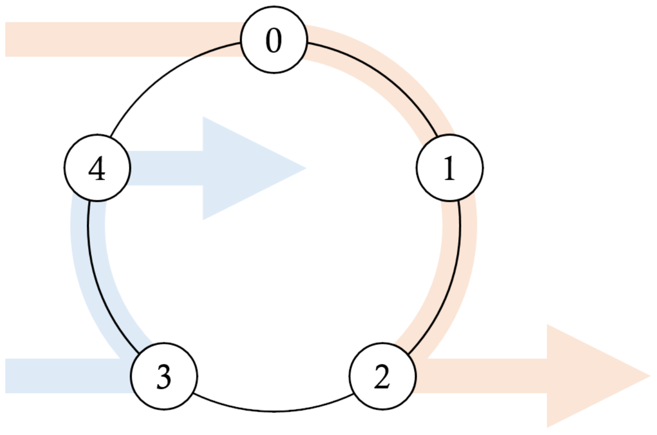


Figure: 4.2: The billiard strategy. The simple cycle is a fiber of some vertex. One end comes to 0 and the other end to 1. Then, each goes along the cycle as long as possible, and stops just before it would collide with an already included vertex. The difference of the two levels at the beginning is preserved at the end. The new ends can now extended to new ends in next fiber.

conditions in Theorem 4.4, then there exists a path P in the system which has the same label at its two ends, and satisfies exactly one of the following:

1. *One end of P is a leaf of Γ, and the other end is its nearest branching vertex of Γ or a leaf of Γ if Γ is a path. We call it* type 1*, or*

2. *Both ends of P are leaves of Γ, and P contains exactly one branching vertex of Γ of degree three. We call it* type 2.

Furthermore, if Γ is not a path and we remove P from Γ except for the vertex of attachment (that is either the branching vertex if P is of type 2 or the end vertex of P if P is of type 1), the new graph will have a system of paths satisfying the conditions of Theorem 4.4.

Proof. The proof is trivial if Γ is a path, so we assume Γ is not a path. Let $P_1, P_2, \dots, P_k$ be paths satisfying the conditions in Theorem 4.4. Since Γ is connected, we can sort these k paths so that they satisfy the following property,

$$\bigcup_{i=1}^{\ell} P_i \text{ constitutes a connected graph for every } \ell \ (1 \leq \ell \leq k), \tag{4.4}$$

and the following constraint: when more than one path can be added to $\bigcup_{i=1}^{\ell-1} P_i$ as P_ℓ, one that becomes of type 2 has always a preference. Then, the last path P_k is obviously of type 1 or type 2. This is the desired path P. We only note the fact that in case 2, the internal branching vertex must be of degree 3 follows from the fact that all paths are internally vertex disjoint.

The second part of the proposition is easy to see. □

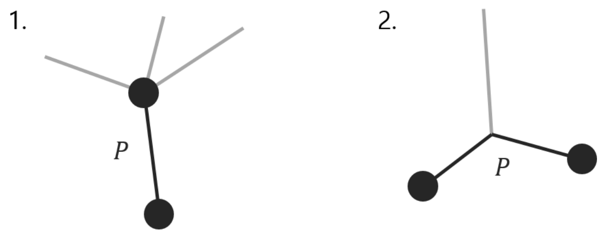


Figure: 4.3: A path of type 1 (left) and type 2 (right).

We have the following corollary.

Proposition 4.7. *Let $(\Gamma, \mathbb{Z}_p, \sigma)$ be a voltage graph. Suppose we have a system of paths $P_k, P_{k-1}, \ldots, P_1$ of Γ which is obtained by recursively removing the paths of type 1 or 2 in Proposition 4.6 until Γ becomes empty. Then $P_1, P_2, \ldots, P_k$ is a system of paths satisfying the conditions of Theorem 4.4.*

Now we proceed to the proof of Theorem 4.4.

Proof of Theorem 4.4. We first prove the necessity. Suppose $p < \Delta$. Let v be a vertex of degree Δ, and let F_v be the fiber over v. Since Γ is a tree, the removal of F_v from Γ^σ creates Δ components. Any spanning cycle of Γ must visit each of these Δ components and so it must visit F_v at least Δ times. However, any closed cycle in Γ^σ can go through the vertices in F_v at most $p < \Delta$ times, a contradiction.

Now we show the sufficiency. The proof is by induction on the parameter $k = 1 + \sum_{v \in V(\Gamma)\,:\, d_{\Gamma-L}(v) \geq 2} (d_{\Gamma-L}(v) - 2)$, i.e., the number of branches of Γ. For the consistency of the induction, we enforce the Hamiltonian cycle to have the following stronger property:

(A) For each loop $(v, v) \in L$, the Hamiltonian cycle uses exactly $p - d_{\Gamma-L}(v)$ edges in the fiber over (v, v).

Lemma 4.5 implies the base case $k = 1$, since **(A)** is satisfied for the constructed Hamiltonian cycle.

For the inductive step, suppose the statement is true for every Γ that has $k > 1$ branches. Suppose Γ has $k+1$ branches. By Proposition 4.6, Γ has a path of either type 1 or 2. We first deal with the former case, i.e., P is of type 1. Let Γ' be the graph obtained by removing the branch P (except for the vertex of attachment) from Γ. Thus, Γ' has k branches and still satisfies the requirements of Theorem 4.4 by Proposition 4.6. Let

σ' be the restriction of σ to $E(\Gamma')$. By the induction hypothesis, $\Gamma'^{\sigma'}$ has a Hamiltonian cycle which satisfies **(A)**, say C'. Let $V(P) = \{v_0, v_1, \ldots, v_\ell\}$, where v_0 is the common vertex of Γ' and P (the vertex of attachment), v_ℓ is the other leaf of P, and v_{i-1} is adjacent to v_i for every i $(1 \le i \le \ell)$. Note that $d_{\Gamma'-L}(v_0) \le \Delta - 1$. By the first condition, $\sigma(v_0, v_0) = \sigma(v_\ell, v_\ell)$. Furthermore, since C' satisfies the property **(A)**, and $p - d_{\Gamma'-L}(v_0) \ge p - (\Delta - 1) \ge 1$, at least one edge in the fiber over the loop $(v_0, v_0) \in L$, say e, is used by C'. To construct a Hamiltonian cycle in Γ^σ, remove e from C', then connect the two end-vertices of e to the vertices in the fiber over v_1 of the same levels, respectively. By using the billiard strategy on P, starting with the two end-vertices, we can extend the current path to a Hamiltonian cycle C of Γ^σ. Let us check C satisfies the property **(A)**. For any vertex different from v_0, C obviously satisfies **(A)**. For v_0, since C' uses $p - d_{\Gamma'-L}(v_0)$ edges in the fiber over v_0, C uses $p - d_{\Gamma'-L}(v_0) - 1 = p - d_{\Gamma-L}(v_0)$ out of them, which ensures **(A)** is satisfied.

Now we deal with the case P is of type 2. Let Γ' be the graph obtained by removing P (except for the vertex of attachment) from Γ. As before, Γ' has $k-1$ branches and still satisfies the requirements of Theorem 4.4 by Proposition 4.6. Let σ' be the restriction of σ to $E(\Gamma')$. By the induction hypothesis, $\Gamma'^{\sigma'}$ has a Hamiltonian cycle which satisfies **(A)**, say C'. Let v_j be the unique vertex in $V(\Gamma') \cap V(P)$ (the vertex of attachment), and let $\sigma(v_j, v_j) = a$. Since v_j is a leaf of Γ', by the property **(A)**, C' uses $p-1$ edges in the fiber over the loop (v_j, v_j). Remove all these edges from C', and let P' be the resulting path having two open ends the difference of whose levels is a. By applying the construction from proof of Lemma 4.5 to P, we have a Hamiltonian cycle C_P of P which satisfies **(A)**. Now we explain how to combine P' and C_P. Without loss of generality, suppose the levels of two open ends of P' are 0 and a, respectively. If C_P uses the edge joining $(v_j, 0)$ and (v_j, a), remove the edge and combine the resulting Hamiltonian path to P', to form a Hamiltonian cycle of Γ^σ. Otherwise *shift* C_P; that is, for every edge (u_0, v_g) in C_P, replace it with (u_h, v_{g+h}) for some $h \in \mathbb{Z}_p$. There exists $h \in \mathbb{Z}_p$ such that the shifted Hamiltonian cycle uses the edge joining $(v_j, 0)$ and (v_j, a) and we can proceed as in the previous case. Thus, by this shifting operation we can always combine the two paths to form a Hamiltonian cycle C of Γ^σ. Let us check C satisfies the property **(A)**. For any vertex different from v_j, C obviously satisfies **(A)**. For v_j, C_P uses $p-2$ edges in the fiber over v_j. Thus, C uses $p - 2 - 1 = p - 3 = p - d_{\Gamma-L}(v_j)$ edges in the fiber over (v_j, v_j), which ensures **(A)** is satisfied. This completes the proof. □

4.2.1 Linear Time Recognition

Since the conditions in Theorem 4.4 are non-trivial to check, we consider the following question:

Question : Can we decide in a polynomial time whether there is a system of paths in $(\Gamma, \mathbb{Z}_p, \sigma)$ which satisfies the conditions in Theorem 4.4?

To answer this question, we again use Proposition 4.6.

Theorem 4.8. *Suppose Γ is a bi-directed tree with a self-loop at every vertex and $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$. There is a linear time algorithm to decide whether $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.4.*

Proof. We say the vertices u and v are *path-adjacent* if u is reachable from v without passing through a branching vertex[2]. To describe the algorithm, we define the notion *safe*. A branching vertex v is safe if at most two of the path-adjacent leaves of v have different labels from v's, and if there are two such leaves, they have the same label.

The algorithm is as follows.

1. If Γ is a path, compare the labels on its two ends. If the labels are same return YES, otherwise NO.
2. If Γ is not a path, find a branching vertex such that all but at most one of its path-adjacent vertices are leaves. Collect all such branching vertices.
3. If one of the collected branching vertices is not safe, return NO.
4. Suppose all of the collected branching vertices are safe. For each such branching vertex v, remove paths of type 1 and 2 in Proposition 4.6 from Γ (except for v) until the degree of v becomes at most two. (These branching vertices are no more branching vertices in the new tree.)
5. Go back to Step 1.

We implement the algorithm in the following way. Put a pointer on each leaf at the beginning of the algorithm. In Step 2, move it until they reach a branching vertex. Each pointer carries the label on its starting leaf. If there is a branching vertex of degree d on which at least $d-1$ pointers are, then it is the desired (by Step 2) branching vertex. Keep the positions of all pointers, and unify the pointers into a single pointer on a vertex if it is processed in Step 4. When unifying the pointers put on v, its label is determined as follows: if exactly one of the pointers (before unifying) has a label different from v's, then the new label is set to be the same as its label. Otherwise the new label is same as v's.

We first verify this algorithm runs in linear time. In Step 1 the algorithm checks whether Γ is a path or not. This is done by just counting the number of pointers; if there are at most two pointers then Γ is a path, otherwise it is not. It only takes constant time for each iteration, and since the number of iteration is at most linear, Step 1 takes linear time in total. In Step 2, the algorithm finds a set of branching vertex satisfying

[2]A branching vertex is a vertex of degree at least three.

the condition on the number of path-adjacent leaves. By the implementation, one can see that every edge of Γ is traversed at most once (by moving pointers) and every vertex is checked at most as much as its degree with this implementation. Thus, Step 2 takes linear time through the run of the algorithm. In Step 3 and 4, the algorithm compares the label on a branching vertex with the labels on its leaves which is carried by pointers, which costs a time proportional to its degree. Hence it takes $\mathcal{O}(|E(\Gamma)|)$ time in total. In Step 4, the algorithm also removes paths, which obviously takes linear time, as it is done by unifying the pointers on currently collected branching vertices. Since every step runs in linear time, the running time of the algorithm is linear.

Next we verify the correctness of the algorithm.

Proposition 4.9. *Let Γ_i be the graph at the beginning of i-th iteration of the algorithm. Then $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.4 if and only if for every i, Γ_i is either a path whose labels on its two ends are same, or Γ_i is not a path and all of the branching vertices the algorithm collects in iteration i are safe.*

Proof. ($\Rightarrow$) Suppose $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.4. If Γ_i is a path, since the algorithm has removed paths of type 1 and 2 iteratively, the labels on its two ends must be same by Proposition 4.6. Suppose Γ_i is not a path. Suppose $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.4 and the algorithm finds a branching vertex v that is not safe. Let L be the multi-set of the labels on the path-adjacent leaves of v, and a be the label on v. Then L contains two distinct labels each of which is different from a, or three same labels different from a. Consider the former case. Let x, y be the leaves having different labels from a. Since Γ has a system of paths satisfying the conditions in Theorem 4.4, x and y are respectively ends of some paths in that system. Let x' (resp. y') be the other end of the path of x (resp. y). Since both x and y have different labels from a, $x' \neq v$ and $y' \neq v$. This implies the x-x' path and the y-y' path intersect at v[3], which contradicts to the condition that P_i and P_j do not share the inner vertex for any $i \neq j$.

For the latter case, suppose x, y, z be the leaves having different labels from a. If these three leaves belong different paths in the system, the three paths intersect at v, which is the contradiction. If two of the vertices (say x, y) belong the same path, then the x-y path and the z-z' path intersect at v, where z' is the other end of the path of z, which is the contradiction again.

($\Leftarrow$) Suppose Γ_i is not a path. Suppose all branching vertices the algorithm finds in each iteration are safe. Let v be a branching vertex for which the algorithm is processing. Let L be the multi-set of the labels on the path-adjacent leaves of v, and a be the label on v. Since v is safe, L contains at most one label different from a, or two same labels

[3]We assume here that any of two paths in the system do not share a vertex of degree two; in that case we can merge the two paths and obtain a new single path, which does not affect the existence of the path system.

different from a. In the former case, all but at most one of the path-adjacent leaves are ends of paths of type 1, the other end being v. The algorithm formally removes these paths until the degree of v becomes two. In the latter case, all but two of the path-adjacent leaves are ends of paths of type 1, the other end being v. The algorithm removes these paths, then the remaining two leaves are the ends of a path of type 2. The algorithm removes the path and the degree of v becomes one. Thus all but one edges incident to v are (formally) removed from Γ_i, v is not a branching vertex any more. Thus in each iteration, paths of type 1 or 2 are removed. And since the number of branching vertices decreases, Γ is finally reduced to a path. If Γ_i is a path, Γ_i is a path of type 1. Since in each iteration only paths removed are of type 1 and 2, by Proposition 4.7, $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.4. □

□

In Theorem 4.8 we described a linear time algorithm which recognizes voltage trees which have a system of paths satisfying Theorem 4.4. The following observation suggests a simple characterization of graphs that have a system of paths satisfying the conditions of Theorem 4.4 in case of cubic tree[4].

Lemma 4.10. *Suppose Γ is a cubic tree with a self-loop at each vertex and L' is the set of self-loops attached to leaves and branching vertices. Suppose $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$ satisfies that $\sigma(L') \in \{a, b\}$, where $a, b \in \mathbb{Z}_p$. Then, the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ contains a system of paths satisfying the conditions in Theorem 4.4 if and only if both $|\{e \in L' : \sigma(e) = a\}|$ and $|\{e \in L' : \sigma(e) = b\}|$ are even.*

Proof. The necessity is easy, we prove the sufficiency. Suppose that both $|\{e \in L' : \sigma(e) = a\}|$ and $|\{e \in L' : \sigma(e) = b\}|$ are even. If Γ is a path, the labels on its two ends are either both a or both b, $E(\Gamma)$ is a path satisfying the conditions in Theorem 4.4. Suppose Γ is not a path. Since Γ is a cubic tree, Γ contains a branching vertex u having two path-adjacent leaves v, w. Since there are at most two distinct labels on u, v, w, we may assume that two of these three vertices have the same label. If they are v and w, v-u-w is a path of type 2 in Proposition 4.6. If one of them is u, the path which starts at u and goes to the leaf having the same label as u is a path of type 1 in Proposition 4.6. Let P_1 be the path of either type. Remove P_1 from Γ except for u. Then the resulting graph is a cubic tree, and satisfies the conditions of the lemma. By recursively finding a branching vertex and removing a path P_i, Γ finally becomes a path. Since the labels on its two ends are both a or both b, by the argument above, Γ itself is a path of type 1. This way, we obtain a system of paths $P_1, P_2, \ldots, P_k$, which satisfies the conditions of Theorem 4.4 by Proposition 4.7. □

[4]A cubic tree is a tree such that every non-leaf vertex has degree at most three.

4.3 The Second Extension : The Same Label at Two Consecutive Vertices

In this section we give another extension of Theorem 4.3.

Theorem 4.11. *Let Γ be a bi-directed tree with a self-loop at each vertex and let L be the set of self-loops. Let $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$. Suppose the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ satisfies the following conditions:*

(a) *There exists a system of paths $P_1, P_2, \ldots, P_k$ of Γ such that $\{E(P_1), E(P_2), \ldots, E(P_k)\}$ is a partition of $E(\Gamma) \setminus L$, the paths P_i and P_j are internally vertex disjoint for any $i \neq j$, and for all i $(1 \leq i \leq k)$ there are two adjacent vertices of P_i, say u_i, v_i, such that their self-loops have the same label,*

(b) *Both u_i and v_i have degree at most two in $\Gamma - L$ for every i $(1 \leq i \leq k)$,*

(c) *$\sigma(w, w)$ is coprime to p for every $w \in V(\Gamma)$.*

Then the covering graph Γ^σ is Hamiltonian if and only if $p \geq \Delta$, where Δ is the maximum degree of $\Gamma - L$.

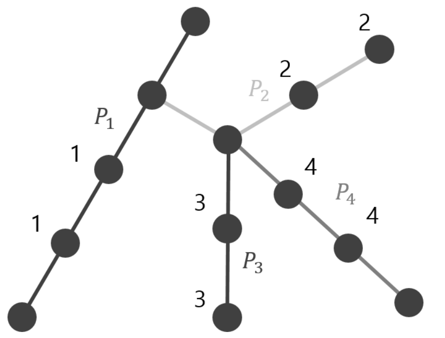


Figure: 4.4: A graph which satisfies the conditions in Theorem 4.11. The number at a vertex denotes the label of its self-loop (self-loops are not drawn). If a vertex has no number next to it, it means its self-loop can have any label coprime to p. P_1, P_2, P_3, P_4 is a system of paths satisfying the conditions in Theorem 4.11.

As in the previous section, we consider the base case first.

Lemma 4.12. *Let Γ be a path with a self-loop at every vertex and let L be the set of self-loops. Suppose $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$ satisfies the following:*

- *$\sigma(u, u) = \sigma(v, v)$ where u and v are some two adjacent vertices on Γ,*

- *$\sigma(w,w)$ is coprime to p for every $w \in V(\Gamma)$.*

Then Γ^σ is Hamiltonian if and only if $p \geq 2$.

Proof. The necessity is easy, so we prove the sufficiency. We may assume $\sigma(u,u) = \sigma(v,v) = 1$ by Proposition 4.2. Let u', v' be the two leaves of Γ such that the u-u' path does not contain v and the v-v' path does not contain u. It is possible that $u = u'$ and/or $v = v'$. Let $\sigma(u',u') = a$ and $\sigma(v',v') = b$. We show that, in the covering graph of the subgraph of Γ induced by $\{u,v\}$, there exists a pair of a vertex-disjoint u_0-v_0 path and a u_a-v_b path that covers all the vertices of this graph. These two paths can then be extended to a Hamiltonian cycle in Γ^σ, by applying the same strategy as in Lemma 4.5 to both the u-u' path starting with ends u_0 and u_a, and the v-v' path starting with ends v_0 and v_b, respectively. It remains to show how to construct the two starting paths u_0-v_0 and u_a-v_b.

We can assume both a and b are odd numbers. If p is odd, this is obvious since a and $p-a$ are same, and one of them must be odd. If p is even, a is odd since a must be coprime to p. The same argument applies to b.

So assume that a and b are both odd. Without loss of generality we can also assume that $a \leq b$, when they are compared as integers. The u_0-v_0 path is constructed in the following way: start from u_0, go straight down to u_{a-1}, hop to v_{a-1}, and go straight up to v_0. The u_a-v_b path is constructed in the following way: start from u_a and go zig-zag until reaching u_b; that is, $u_a, v_a, v_{a+1}, u_{a+1}, u_{a+2}, v_{a+2}, \ldots, v_{b-1}, u_{b-1}, u_b$. Once we get to u_b we can get to v_b by first going straight down to u_{p-1} and hop to v_{p-1}, then going straight up to v_b.

Note that, if u or v is a leaf, i.e. $u = u'$ or $v = v'$ respectively, one can see that the construction above still works. □

To prove Theorem 4.11 we need the following proposition whose proof is similar to that of Proposition 4.6.

Proposition 4.13. *Suppose Γ is a bi-directed tree with a self-loop at every vertex and $\sigma \colon E(\Gamma) \to \mathbb{Z}_p$. If the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ has a system of paths satisfying the conditions in Theorem 4.11, then there exists a path P which contains two consecutive vertices having the same label on their self-loops, has the property:*

- *one end of P is a leaf of Γ, and the other end is its nearest branching vertex of Γ (or a leaf of Γ if Γ is a path).*

Furthermore, if we remove P from Γ except for the vertex of attachment, the new graph will have a system of paths satisfying the conditions of Theorem 4.11.

By using Proposition 4.13, one can obtain a linear time algorithm to decide whether $(\Gamma, \mathbb{Z}_p, \sigma)$ satisfies the conditions in Theorem 4.11, which is similar to the one in Theorem 4.8. We omit the details here.

Now we prove Theorem 4.11.

Proof of Theorem 4.11. To construct a Hamiltonian cycle of Γ^σ, we use the recursive construction similar to the one in the proof of Theorem 4.4. For the consistency of induction, we assume the following condition for the Hamiltonian cycle:

(A) The Hamiltonian cycle uses $p - d_{\Gamma - L}(v)$ edges in the fiber over $(v, v) \in L$ for each $v \in V(\Gamma)$ except possibly for u_i, v_i in conditions (a) and (b).

The base case is proved in Lemma 4.12. Note that the Hamiltonian cycle constructed there satisfies the condition **(A)** since it uses $p - 1$ edges in the fiber over the loop of each end-vertex and $p - 2$ edges in the fiber over the loop of each inner-vertex except for u and v.

For the inductive step, suppose there are k branches in Γ. By Proposition 4.13, there exists a path P of Γ satisfying the property of Proposition 4.13 that can be removed from Γ (except for the vertex of attachment) so that the resulting graph Γ' will still satisfies the conditions of Theorem 4.11. Let v^* be the vertex of attachment, and σ' be the restriction of σ to $E(\Gamma')$. Note that v^* cannot be either u_i or v_i in conditions (a) and (b) since degree of v^* in Γ is at least three. Let C' be the Hamiltonian cycle of Γ' which we assume exists by the inductive hypothesis.

To construct a Hamiltonian cycle C of Γ^σ we do the following: cut one "vertical" edge in the fiber over v^* from C' to make its two ends open, then connect these ends with a Hamiltonian path of the covering graph of the k-th branch. To accomplish this strategy successfully, one must ensure that there are available "vertical" edges in the fiber over v^* to cut off. Since $p \geq \Delta$, the number of edges in the fiber over v^* used in C' is $p - d_{\Gamma' - L}(v^*) \geq \Delta - d_{\Gamma' - L}(v^*) \geq 1$. Hence there must be at least one edge we can make use of, and thus we can obtain a Hamiltonian cycle C of Γ^σ. Now let us see that C satisfies **(A)**. Since C' satisfies **(A)**, the only vertex that can violate the property is v^*. By the construction, C uses $p - d_{\Gamma' - L}(v^*) - 1 = p - d_{\Gamma - L}(v^*)$ edges in the fiber over the loop (v^*, v^*), hence C satisfies **(A)**. □

As in the previous section, one can see that Theorem 4.11 is an extension of Proposition 4.3 by restricting σ to be the all-one label.

Finally, we note that we can merge Theorem 4.4 and Theorem 4.11 to obtain the following.

Corollary 4.14. *Let Γ be a bi-directed tree with a self-loop at each vertex and let L be the set of self-loops. Let $\sigma \colon E(\Gamma) \to \mathbb{Z}_p$. Suppose the voltage graph $(\Gamma, \mathbb{Z}_p, \sigma)$ satisfies the following conditions:*

- *There exists a system of paths $P_1, P_2, \ldots, P_k$ of Γ such that $\{E(P_1), E(P_2), \ldots, E(P_k)\}$ is a partition of $E(\Gamma) \setminus L$, the paths P_i and P_j are internally vertex disjoint any $i \neq j$, and for all i $(1 \leq i \leq k)$ P_i satisfies either of the following:*

  - *the two ends of P_i have the same label, or*
  - *there are two adjacent vertices u_i, v_i having the same label, both u_i and v_i have degree at most two in $\Gamma - L$,*

- *$\sigma(w, w)$ is coprime to p for every $w \in V(\Gamma)$.*

Then the covering graph Γ^σ is Hamiltonian if and only if $p \geq \Delta$, where Δ is the maximum degree of $\Gamma - L$.

4.4 Further Discussion : When Some Labels are Not Coprime

So far we have required that every label has to be coprime to p, the order of the cyclic group. In this section, we investigate the case when some labels are not coprime to p. The following result gives a sufficient condition of Hamiltonicity of a covering graph of a path under some additional conditions.

Theorem 4.15. *Let Γ be the path of length n with a self-loop at every vertex and suppose $V(\Gamma) = \{v_1, v_2, \ldots, v_n\}$, where v_1 and v_n are leaves, and v_{k-1} and v_k are adjacent for all k $(2 \leq k \leq n)$. Let $\sigma\colon E(\Gamma) \to \mathbb{Z}_p$. If n is odd and $\sigma(v_1, v_1) = \sigma(v_n, v_n) = 1$ and $\gcd(p, \sigma(v_k, v_k)) = d$ for every k $(2 \leq k \leq n-1)$ for some odd integer d, then Γ^σ is Hamiltonian if $p \geq 2d$.*

Proof. We first construct $2d$ paths joining vertices in the fiber over v_1 to those in the fiber over v_n. Then we connect these paths appropriately to a Hamiltonian cycle of Γ^σ. Let P_i $(0 \leq i \leq 2d-1)$ be the path having one end at (v_1, i) which will be fixed, and another end at (v_2, i) initially. At each step we extend each P_i by extending its non-fixed end to the next fiber, until it reaches a vertex in the fiber over v_n. We achieve this by applying the billiard strategy simultaneously to all $2d$ paths. Finally, we appropriately close ends of these paths in the first and last fibers to form a Hamiltonian cycle of Γ^σ. For a complete example of the construction, we refer the reader to Fig. 4.5. The following proposition ensures that this strategy will work properly.

For the path P_i, let $f_k(i)$ be the level of the vertex at which P_i leaves the fiber over v_k when we apply the billiard strategy simultaneously to all these paths for $1 \leq k \leq n-1$.

Proposition 4.16.

$$f_k(i) = A_k + i \bmod p \tag{4.5}$$

for odd k, and

$$f_k(i) = \begin{cases} A_k + i \bmod p & (0 \leq i \leq d-1), \\ A_k - 2d + i \bmod p & (d \leq i \leq 2d-1) \end{cases} \tag{4.6}$$

for even k, where A_k is a constant depending on k $(0 \leq A_k \leq p-1)$. Furthermore, for every k, $f_k(i) \equiv j \pmod{d}$ if and only if $i = j$ or $i = j+d$.

Proof. Proof is by induction on k. If $k = 1$ we have $f_k(i) = i$ by the definition of P_i, (4.5) clearly holds with $A_1 = 0$, and the latter statement is clear as well. Suppose $k > 1$ and the proposition is true for $k-1$. We first prove the latter statement. Since $\gcd(\sigma(v_k, v_k), p) = d$, the fiber over v_k (considered as a graph) is a disjoint union of d cycles of length p/d. Let $C_j^{(k)}$ be the cycle in the covering graph which contains the vertex (v_k, j) $(0 \leq j \leq d-1)$. Then, $C_j^{(k)}$ contains all vertices whose levels are equivalent to j modulo d, that is, $(v_k, j), (v_k, j+d), \dots, (v_k, j+p-d)$. By the induction hypothesis, $f_{k-1}(i) \equiv j \pmod{p}$ if and only if $i = j$ or $i = j+d$, hence there are only two paths P_j and P_{j+d} which are passing through $C_j^{(k-1)}$. These two paths enter $C_j^{(k)}$ at vertices having the same levels as vertices via which these paths left $C_j^{(k-1)}$ in the fiber over v_{k-1}, since $\sigma(e) = 0$ for every bridge e of Γ. By the billiard strategy, these two paths visits only the vertices in $C_j^{(k)}$ in the fiber over v_k, and they visit all vertices of levels j modulo d before leaving the fiber, and thus $f_k(i) \equiv j \pmod{d}$ holds for $i = j, j+d$. The latter statement is proved.

Now let us see that (4.5) and (4.6) hold. Suppose k is odd and $f_{k-1}(i)$ satisfies (4.6). Given j, $0 \leq j \leq d-1$, consider the two paths P_j and P_{j+d}, which get into $C_j^{(k)}$ by the latter statement. Let $\sigma(v_k, v_k) = \ell d$ where ℓ is coprime to p. Then, by the behavior of the billiard strategy, we have

$$f_k(j) = f_{k-1}(j+d) - \ell d \bmod p \text{ and } f_k(j+d) = f_{k-1}(j) - \ell d \bmod p. \tag{4.7}$$

By (4.6) and (4.7), we get $f_k(j+d) - f_k(j) = d \bmod p$ for every j $(0 \leq j \leq d-1)$, and $f_k(j+1) - f_k(j) = 1 \bmod p$ for every j $(0 \leq j \leq d-2)$. Thus, $f_k(0), f_k(1), \dots, f_k(2d-1)$ is a monotonic sequence increasing by one modulo p, hence $f_k(i)$ satisfies (4.5).

For even k, suppose $f_{k-1}(i)$ satisfies (4.5). Note that (4.7) also holds for even k. By (4.5) and (4.7) we get $f_k(j+d) - f_k(j) = -d \bmod p$ for every j $(0 \leq j \leq d-1)$, and $f_k(j+1) - f_k(j) = 1 \bmod p$ for every j $(0 \leq j \leq d-2)$. This implies $f_k(d), f_k(d+1), \dots, f_k(2d-1), f_k(0), f_k(1), \dots, f_k(d-1)$ is a monotonic sequence increasing by one modulo p, thus $f_k(i)$ satisfies (4.6). □

Now we will complete the proof of Theorem 4.15. We have that $f_{n-1}(i)$ satisfies (4.6) since n is odd, and thus, end of each P_i meets the vertex of level $f_{n-1}(i)$ in the fiber over v_n. We stop the extensions of paths at this point. To construct a Hamiltonian cycle of Γ^σ, close the $2d$ ends of P_i's in the fiber over v_n in the following way:

- Connect the end of P_i to the end of P_{i+1} for $i = 0, 2, \dots, d-3$,
- Join the end of P_{d-1} to the end of P_d by the path consisting of the vertices lying between them,

- Connect the end of P_i to the end if P_{i+1} for $i = d+1, d+3, \ldots, 2d-2$.

Close the ends of P_i's in the fiber over v_1 in the following way:

- Connect the end of P_i to the end of P_{i+1} for $i = 1, 3, \ldots, 2d-3$,
- Join the end of P_{2d-1} to the end if P_0 by the path consisting of vertices $(v_0, 2d), (v_0, 2d+1), \ldots, (v_0, p-1)$.

One can see that we obtain a single closed cycle that covers all vertices of Γ^σ. Thus we have obtained a Hamiltonian cycle of Γ^σ. □

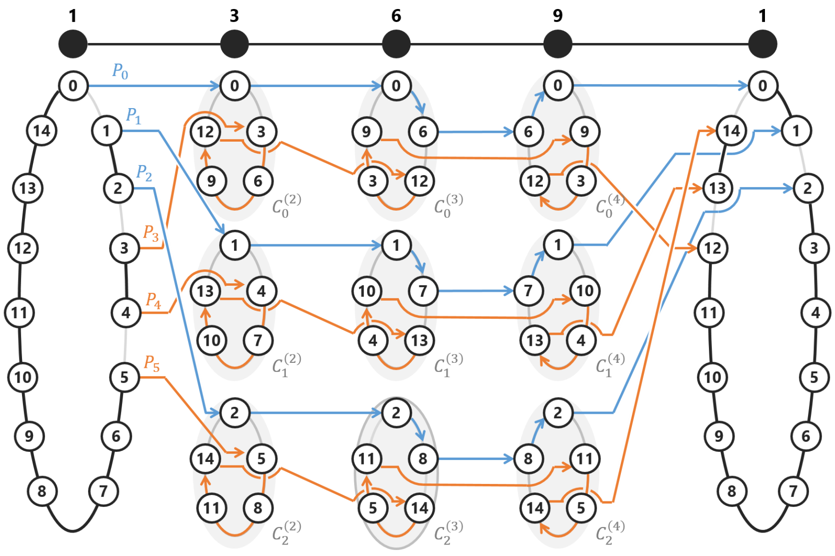


Figure: 4.5: The billiard strategy in Theorem 4.15. The underlying group is $\mathbb{Z}_{15}$. A number beside a vertex of the horizontal path Γ is the label on it. In this example, we suppose the gcd together with 15 is 3. The blue and orange lines denote the paths P_i's, and the black lines in the first and last fibers are the joining edges. Observe that together these lines form a Hamiltonian cycle.

Chapter 5

The Odd Depth Tree Problem

A spanning tree of a graph G is a spanning connected subgraph of G having no cycles. While it is easy to find a spanning tree (of minimum weight) in a graph, there are considerable number of hard variants. For example, finding a spanning tree having minimum or maximum number of leaves, bounded degrees, bounded diameter, and bounded number of hops, are all known to be NP-hard [23, 25]. Notice that the Hamiltonian path problem is equivalent to the problem to find a spanning tree with minimum number of leaves. In this chapter, we consider a variant of the problem to find a spanning tree with bounded diameter relaxing the constraint using parity. An *odd depth tree* is a spanning tree such that every leaf has an odd distance from a prescribed root vertex. The *odd depth tree problem* is to find such a tree in a given undirected graph and a root vertex.

After defining some new notations in Section 5.1, we mention our results in the continuing sections. In Section 5.2 we give a characterization for the odd depth tree problem in bipartite graphs and show it is solved in polynomial time. In Section 5.3 we show the problem in NP-complete. In Section 5.4 we consider the directed version of the problem.

5.1 Preliminaries

For a vertex $v \in V$, $u \in V$ is called a *neighbor* of v if $\{u, v\} \in E$. We denote by $N_G(v)$ the set of neighbors of v. For $X \subseteq V$, we denote by $N_G(X)$ the set of vertices which are not in X and having neighbors in X. By $\delta_G(v)$ we denote the set of edges incident to v in G. The number of edges incident to v is called the *degree* of v and denoted by $d_G(v)$. For $X \subseteq V$, we denote by $\delta_G(X)$ the set of edges having one end in X and the other end in $\overline{X}$. $N_G(X)$ (resp. $\delta_G(X)$) is simply written as $N(X)$ (resp. $\delta(X)$) if the graph we are discussing is obvious. For $F \subseteq E$, we denote by $G[F]$ a subgraph of G induced by F. We simply write $N_{G[F]}(X)$ as $N_F(X)$ (analogously we write $\delta_F(X)$ and $d_F(v)$).

A *forest* of G is a spanning subgraph of G which does not contain a cycle. A *spanning* *tree*, or simply *tree* is a forest consisting of one connected component. A vertex of a tree of degree one is called a *leaf* of G. The *distance* of two vertices u, v in a tree is the number of edges contained in the unique path between u and v. When the root vertex r is specified in a tree, the distance from r to a vertex v is called the *depth* of v. The definition of an *odd depth tree* is given as follows:

Definition 5.1 (Odd Depth Tree)**.** *Let $G = (V, E)$ be an undirected graph and $r \in V$. A tree T of G is an* odd depth tree with respect to r *if every leaf of T (except r if it is a leaf) has an odd depth.*

The *odd depth tree problem* is the problem to find an odd depth tree with respect to a prescribed root r in an undirected graph. We assume the input graph is connected to ensure it has a spanning tree.

5.2 Bipartite Graphs

In this section we study the odd depth tree problem in bipartite graphs. The following is the characterization of bipartite graphs which contains odd depth trees.

Theorem 5.2. *Let $G = (U, V; E)$ be a bipartite graph. Suppose $r \in U$. Then G has an odd depth tree with respect to r if and only if*

$$|N(X)| \geq |X| + 1 \tag{5.1}$$

holds for every $X \subseteq U \setminus \{r\}, X \neq \emptyset$.

Proof. **Necessity.** Suppose there is a non-empty subset $X \subseteq U \setminus \{r\}$ with $|N(X)| \leq |X|$. If $|X| = 1$, the unique member of X has degree one, which is a leaf of an even depth in any tree of G. Suppose $|X| = k \geq 2$ and G has an odd depth tree T with respect to r. Since every vertex in $u \setminus \{r\}$ cannot be leaves in T, $d_T(u) \geq 2$ for any $u \in U \setminus \{r\}$, there are at least $2k$ edges incident to X. On the other hand, since $|N_T(X)|$ is at most k, $|X \cup N_T(X)| \leq 2k$. Hence there are at most $2k$ vertices and at least $2k$ edges in $T[X \cup N_T(X)]$, which implies there is a cycle, which is a contradiction.
Sufficiency. We give an algorithm to construct an odd depth tree with respect to r. The algorithm uses the matroid intersection algorithm for two matroids $\mathcal{M}_i = (E, \mathcal{I}_i)$ $(i = 1, 2)$ on the same set E defined as follows:

- $\mathcal{I}_1$ consists of $F \subseteq E$ such that F does not induce a cycle, which is known to be the graphic matroid, and
- $\mathcal{I}_2$ consists of $F \subseteq E$ such that $d_F(u) \leq 2$ for every $U \setminus \{r\}$, which is known to be the partition matroid.

The solution the matroid intersection algorithm returns, if exists, is a forest in which every vertex in $u \in U \setminus \{r\}$ has degree two. Once we have obtained it, we add edges connecting all of the components of the forest and we obtain an odd depth tree with respect to r. In what follows we show under the assumption $|N(X)| \geq |X| + 1$ for every non-empty $X \subseteq U \setminus \{r\}$, the matroid intersection algorithm always returns such a forest.

Let $F = \emptyset$. While there exists a vertex $u \in U \setminus \{r\}$ such that $d_F(u) < 2$, the matroid intersection algorithm do the following:

- If there is an edge $e \in \delta_G(u)$ such that $F \cup \{e\}$ does not induce a cycle, add e to F.
- If there is no such edge, find edges $e_0, f_1, e_1, f_2, e_2, \ldots, f_\ell, e_\ell \in F$ such that $e_0 \in \delta_G(u)$ and $F \cup \bigcup_{i=0}^{\ell} e_i \setminus \bigcup_{i=1}^{\ell} f_i \in \mathcal{I}_1 \cap \mathcal{I}_2$.

let X be the set of vertices in $U \setminus \{r\}$ contained in the same connected component as u in $G[F]$. Let $M(X) \subseteq V$ be the set of vertices mated to vertices in X in M. Find a vertex $u' \in X$ which is adjacent to a vertex $v' \in V \setminus M(X)$ in G. Such u' always exists since $|N(X)| \geq |X| + 1$. Note that $d_F(u') = 1$ since u' lies in the same component as $u \neq u'$. Let $e = u'v'$ and $f \in F \cap \delta_G(u')$. Exchange e and f, that is, $F \leftarrow F \cup \{e\} \setminus \{f\}$. The edge set $M \cup F$ the procedure returns induces a forest of G such that every vertex in $U \setminus \{r\}$ has degree two. Finally, by adding edges connecting distinct components in $M \cup F$, we obtain an odd depth tree with respect to r. $\square$

Since the matroid intersection algorithm runs in polynomial time, the algorithm in the sufficiency part also runs in polynomial time, we obtain the following.

Corollary 5.3. *The odd depth tree problem is solved in polynomial time for bipartite graphs.*

Before moving to the next section, we should mention here an interesting relation between our result and Bérczi et al's [6], which studies the DM-irreducible graphs. In their paper, the characterization of DM-irreducible graphs is shown (Lemma 3.1 in [6]), which is exactly same as ours (Theorem 5.2). Moreover, they gave a polynomial time algorithm to find an optimal edge set to add to a DM-redcucible graph to make it DM-irreducible. In their algorithm, the matroid intersection algorithm with two matroids is used, the graphic matroid and the partition matroid. This is again same as what we used in the sufficiency part of the proof of Theorem 5.2. This coincidence was totally unexpected and we don't know how to explain this; anyway, it should imply some relationship between two objects matchings and spanning trees. To clarify the detail is one of our future works on this topic.

5.3 Non-bipartite Graphs

In the previous section we gave a characterization of bipartite graphs having odd depth treed, and showed the problem is solved in polynomial time. Contrary to it, this section shows the problem for non-bipartite graphs is intractable.

Theorem 5.4. *The odd depth tree problem for two-edge-connected non-bipartite graphs is NP-complete.*

Proof. We show a reduction from CNF-SAT. Let ϕ be an input boolean formula of CNF-SAT such that every clause has at least three literals. Let $x_1, \ldots, x_n$ be the variables and $C_1, \ldots, C_m$ be the clauses of ϕ, where each C_j is in the form of $y_{j1} \wedge \ldots \wedge y_{jl}$ where y_{jk} is either x_{jk} or $\overline{x_{jk}}$. Construct an undirected graph G_ϕ as follows: first draw a root vertex r, and a triangle ru_iv_i for each variable x_i $(1 \leq i \leq n)$. We call the triangle ru_iv_i the *variable gadget* of x_i. For each clause C_j, draw l_j squares where l_j is the number of the literals in C_j. Draw a single vertex w_j, and connect each square and w_j by a single edge. The *clause gadget* of C_j is the subgraph consisting of w_j and the l squares connected to it.

We explain how to connect the variable gadgets and the clause gadgets. For each square of the clause gadget of C_j, the *entrance* is the vertex which is the farthest from w_j in the clause gadget. Connect squares and variable gadgets as follows:

- For each square corresponding to the literal x_i, connect its entrance and v_i by a single edge.
- For each square corresponding to the literal $\overline{x_i}$, connect its entrance and v_i by a path of length two.

Now the construction of G_ϕ is completed.

We show that ϕ is satisfiable if and only if G_ϕ has an odd depth tree with respect to r. Suppose ϕ is satisfiable. We show the construction of an odd depth tree T of G_ϕ with respect to r. Let $\boldsymbol{x}^*$ be a satisfying assignment of ϕ. For the variable gadget of x_i, pick up edges for $E(T)$ as follows: ru_i and rv_i if $x_i^* = \mathrm{T}$, or ru_i and u_iv_i if $x_i^* = \mathrm{F}$. Note that v_i is of an odd depth if $x_i = \mathrm{T}$, or of an even depth otherwise. Pick all of the edges connecting the variable gadgets and clause gadgets and add them to $E(T)$. Let a, b, c, d be the distinct vertices of a square, where a is the entrance and c is the vertex at opposite side, that is, the vertex adjacent to w_j. Since at least one literal is assigned T in each clause in $\boldsymbol{x}^*$, there is at least one square having the entrance of even depth for each clause gadget. For l squares of clause gadget of C_j, pick edges for $E(T)$ as follows:

- Choose one square having the entrance of an even depth, pick ab, bc, ad and cw_j.
- For all the other squares having the entrances of even depths, pick ab, cd and cw_j.

- For squares having the entrances of odd depths, pick bc, cd and cw_j.

One can verify that $E(T)$ induces a tree of G_ϕ, and every leaf of T has an odd depth. Thus we have obtained an odd depth tree T with respect to r.

Now we show the converse. Suppose G has an odd depth tree T with respect to r. We first claim the following: for every clause gadget, there exists a square whose entrance is of an even depth and is connected to v_i for some i by a single edge or a path of length two. The proof is by contradiction. Let S be the set of the squares in the clause gadget of C_j having connections to some v_i. Suppose there exists a clause C_j such that the entrances of all the squares in S are of odd depths. Let a, b, c, d be the vertices of a square, where a is the entrance and c is the vertex at opposite side, that is, the vertex adjacent to w_j. If there is a square in S such that a is adjacent to either b or d in T, one of them should be a leaf of even depth, that must not occur. Hence for every square in S, the entrance is adjacent to neither b nor d in T. However, the disconnections imply that T is disconnected, this is a contradiction again. Therefore, there must be at least one entrance of even depth for every clause gadget having connection to some v_i.

Now let us see how to construct a satisfying assignment of ϕ. For each clause gadget, select one square as the representative which has the entrance of an even depth. Let x_i be the variable corresponding to the square. Then, by the construction of G_ϕ, the depth of v_i must be odd when x_i appears in the clause as a positive literal, or even when it appears as a negative literal $\overline{x_i}$. Determine the assignment of each x_i as follows: $x_i = \mathrm{T}$ if v_i is of an odd depth, and $x_i = \mathrm{F}$ if v_i is of an even depth. Note that the assignment does not conflict (that is, it does not happen that a variable x_i assigned to T from a clause, but assigned to F from another clause) since T is a tree, and it is bipartite. For v_i that is not connecting to a representative square, the assignment of the corresponding variable x_i is don't care. Now the assignment is completed. Let us verify the assignment $\boldsymbol{x} = (x_i)_{i=1}^n$ satisfies ϕ. Since we selected an entrance of an even depth for each clause, and by the rule of deciding the assignment of each variable, the corresponding literal x_i or $\overline{x_i}$, must be T. Thus $\boldsymbol{x}$ assigns T for at least one literal for every clause, $\boldsymbol{x}$ satisfies ϕ. □

G_ϕ is two-edge-connected but not two-connected in general. However, G_ϕ is NOT two-connected if and only if ϕ is partitioned into $\phi = \phi_1 \wedge \phi_2$ such that ϕ_1 and ϕ_2 have no common variable. In that case the root vertex r is a (only) cut-vertex of G_ϕ. For CNF formulas that are not separable the proof of Theorem 5.4 holds, the following corollary is derived.

Corollary 5.5. *The odd depth tree problem is NP-complete for two-connected non-bipartite graphs.*

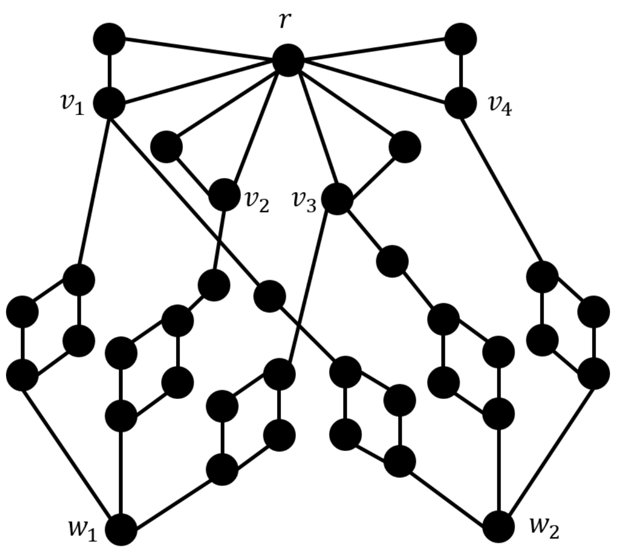


Figure: 5.1: G_ϕ of the boolean formula $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$.

5.4 Directed Graphs : The Odd Depth In-tree Problem

In this section we consider the odd depth tree problem in *directed* graphs. We first introduce some notations for digraphs.

Let $D = (V, A)$ be a digraph. For a vertex v, a vertex u is called an *incoming neighbor* (resp. *outgoing neighbor*) of v if $(u, v) \in A$ (resp. $(v, u) \in A$). We denote by $N^-_D(v)$ (resp. $N^+_D(v)$) the set of incoming (resp. outgoing) neighbors of v. By $\delta^-_D(v)$ (resp. $\delta^+_D(v)$) we denote the set of incoming (resp. outgoing) arcs of v. The number of incoming (resp. outgoing) arcs of v is called the *incoming* (resp. *outgoing*) *degree* of v, and denoted by $d^-_D(v)$ (resp. $d^+_D(v)$). For $X \subseteq V$, we define $N^-_D(X)$ and $N^+_D(X)$ analogously to $N_G(X)$. We simply write $N^-(X)$ or $N^+(X)$ if the digraph we are discussing is obvious.

The followings are the definitions of *in-trees* and *odd depth in-trees*:

Definition 5.6 (In-trees)**.** *Let $D = (V, A)$ be a digraph and $r \in V$. An* in-tree with root *r is a subgraph T of D which satisfies the following three conditions:*

(a) The out-degree of r is 0,

(b) Every vertex except r has out-degree 1, and

(c) r is reachable from every vertex.

Definition 5.7 (Odd Depth In-trees)**.** *Let $D = (V, A)$ be a digraph and $r \in V$. An in-tree T of D is an* odd depth in-tree with respect to *r if every leaf of T except r itself has an odd distance from r.*

The *odd depth in-tree problem* is the problem to find an odd depth in-tree with respect to a prescribed root r in a digraph D. We assume that the root r is reachable from any vertex in D to ensure it has an in-tree.

5.4.1 Directed Bipartite Graphs

As in the undirected version of the problem, we first consider the bipartite case. For $X, Y \subseteq V(D)$ such that $X \cap Y = \emptyset$, an (X, Y)-*perfect matching* is a matching that only consists of arcs from X to Y and that covers all vertices in Y. The following is the main result.

Theorem 5.8. *Let $D = (U, V; A)$ be a bipartite digraph which contains an in-tree with root r. Suppose $r \in U$. Then D has an odd depth in-tree with respect to r if and only if there is a $(V, U \setminus \{r\})$-perfect matching M which satisfies the following:*

(A) *For every non-empty $X \subseteq U \setminus \{r\}$, there exists a vertex $v \in N^+(X) \setminus M(X)$ such that $N_D^+(v) \setminus X \neq \emptyset$,*

where $M(X)$ denotes the set of vertices mated to some vertex in X in M.

Proof. **Necessity.** Suppose D has an odd depth in-tree T with respect to r. Since all vertices in $U \setminus \{r\}$ are not leaves, they must have at least one incoming arcs in T. By the definition of in-tree, every vertex except r has out-degree one, $N_T^-(u_1) \cap N_T^-(u_2) = \emptyset$ for any $u_1, u_2 \in U \setminus \{r\}$ $(u_1 \neq u_2)$. By taking one of the incoming arcs of each $u \in U \setminus \{r\}$ in T, we obtain a $(V, U \setminus \{r\})$-perfect matching M.

Now we prove M must satisfy the condition **(A)**. Suppose that D contains an odd depth tree with respect to r, and let M be a $(V, U \setminus \{r\})$-perfect matching M contained in T. Assume that in M there is a non-empty set $X \subseteq U \setminus \{r\}$ that violates **(A)**. Then either of the following holds: $N_T^+(X) \setminus M(X)$ is empty or every $v \in N_T^+(X) \setminus M(X)$ satisfies $N_T^+(v) \setminus X = \emptyset$. In both cases, any path starting from a vertex of X comes back to some vertex in X after two steps, and it cannot reach r, which contradicts to the fact that T is an odd depth in-tree. Thus we proved the necessity.

Sufficiency. The proof is quite similar to the sufficiency part of the proof of Theorem 5.2. Suppose there is a $(V, U \setminus \{r\})$-perfect matching M satisfying **(A)**. Let $F = \emptyset$. We repeat the following procedure while there exists a vertex $u \in U \setminus \{r\}$ such that $d_{M \cup F}^+(u) = 0$:

- If there is an arc $e \in \delta_D^+(u) \setminus M$ such that $M \cup F \cup \{e\}$ does not induce a cycle[1], add e to F.

[1]Here a cycle is not necessarily directed; it may contain arcs of any direction.

- If there is no such arc, let $X \subseteq U \setminus \{r\}$ be the set of vertices contained in the same connected component as u in $D[M \cup F]$. Find a pair of vertices u' and v' such that $u' \in X$, $v' \in V \setminus M(X)$, and $(u', v') \in A$. Such u', v' always exists since M satisfies the condition **(A)**. Let $e = (u', v')$ and $f \in F \cap \delta_D^+(u')$. Exchange e and f, that is, $F \leftarrow F \cup \{e\} \setminus \{f\}$.

The arc set $M \cup F$ the procedure returns induces a forest of D satisfying the following: every vertex in $U \setminus \{r\}$ has in-degree one and out-degree one, having exactly one vertex of out-degree zero in each connected component. In other words, $M \cup F$ induces a union of disjoint in-forests. To complete the construction of an odd depth in-tree, connect all components of the forest so that r can be reached from every vertex. This is done by the following procedure.

1. Set $W = \{r\}$.
2. While $W \subsetneq U \cup V$ repeat the following.
   (a) Find a vertex $v \in V \setminus W$ such that v has no outgoing arc in $M \cup F$ and has an outgoing arc to some vertex in W in D.
   (b) Add the outgoing arc to F and let $W \leftarrow W \cup C$, where C is the component to which v belongs.
3. If $W = U \cup V$, return $M \cup F$.

Let us verify that this procedure always works correctly. We show the vertex $v \in V \setminus W$ in Step 2(a) always exists. If there exists a vertex $v \in V \setminus W$ that constitutes a component by itself (that is, v has no incident arc in $M \cup F$), take this v. Suppose there is no such vertex. Since M satisfies the condition **(A)**, there exists a vertex $v \in N_D^+(U \setminus W)$ such that $N_D^+(U \setminus W) \setminus (U \setminus W) \neq \emptyset$. The condition implies v has an outgoing arc to W in D, take this v. Thus we showed the v in Step 2(a) always exists. And since W monotonically increases, the procedure terminates in a finite number of repetitions. Thus we obtain an odd depth in-tree with respect to r. □

Theorem 5.8 gives a characterization of odd depth in-trees in bipartite directed graphs, but it is not clear that it is tested in polynomial time. In fact it is unsettled that the odd depth in-tree problem is solved in polynomial time in directed graphs. However, for bipartite *DAG's* (directed acyclic graphs), we have the following result.

Theorem 5.9. *Let $D = (U, V; A)$ be a bipartite DAG which has an in-tree with root r. Suppose $r \in U$. Then D has an odd depth tree with respect to r if and only if*

$$|N^-(X)| \geq |X|, \tag{5.2}$$

holds for any $X \subseteq U \setminus \{r\}, X \neq \emptyset$.

We omit the proof as it is essentially same as in Theorem 5.2. The characterization is exactly same as that of Hall's theorem, the following is obtained.

Corollary 5.10. *The odd depth tree problem is solved in polynomial time for bipartite DAG's.*

5.4.2 Directed Non-bipartite Graphs

In this section we consider the odd depth in-tree problem in non-bipartite graphs. Similar to Section 4, we obtain the following hardness result.

Theorem 5.11. *The odd depth in-tree problem in non-bipartite graphs is NP-complete even for DAGs.*

Proof. The proof is similar to that of Theorem 5.4. We show the reduction from the same problem, CNF-SAT. Let ϕ be the input CNF formula, and $x_1, x_2, \ldots, x_n$ be its variables and $C_1, C_2, \ldots, C_m$ be its clauses. We explain how to construct a DAG $D_\phi = (V, A)$ from ϕ. First draw a single vertex r. For each variable x_i, add three arcs $(v_i, u_i), (u_i, r)$ and (v_i, r). We call this the variable gadget of x_i. Construct the clause gadget of C_j as follows: Draw the complete bipartite graph $K_{k,k}$, where k is the number of literals in C_j. Orient all edges to the same direction; in other words, orient them not to make a directed cycle. Let us call a vertex with no incoming arc *source*, and a vertex with no outgoing arc *sink*. Then each sink of $K_{k,k}$ corresponds to each literal in C_j. Connect the clause gadgets and variable gadgets in the following way: if a sink in the clause gadget of C_j corresponds to the variable x_i, connect the sink to v_i by:

- an arc if x_i appears as the positive literal in C_j, or
- a path of length two if x_i appears as the negative literal in C_j.

By connecting every pair of sinks and v_i, we obtain D_ϕ.

Now let us prove D_ϕ contains an odd depth in-tree with respect to r if and only if ϕ is satisfiable. We first see the "if" part. Suppose ϕ is satisfiable, that is, ϕ has a satisfying assignment $\boldsymbol{x}^*$. We obtain an odd depth in-tree T of D_ϕ as follows: For the variable gadget of x_i, pick up the following two arcs in $E(T)$: (u_i, r) and (v_i, r) if x_i^* is T, or (v_i, u_i) and (u_i, r) if x_i^* is F. Pick up all arcs connecting variable gadgets and clause gadgets for $E(T)$. Then, one can observe that there is at least one sink which is of an even depth for each clause gadget. For each sink of an even depth, pick up one of its incoming arcs for $E(T)$ not to take more than one outgoing arc of the same source. For the other sinks, pick up none of their incoming arcs. Then, pick up the outgoing arcs of the remaining sources that get in sinks of even depth. From the construction one can see that the depth of every leaf is odd, thus we obtain an odd depth in-tree T with respect to r.

Now let us see the converse. Suppose D_ϕ has an odd depth in-tree with respect to r. For each clause gadget, every source in $K_{k,k}$ is a leaf, thus it must be of an odd depth. Then there is at least one sink of an even depth, since it has an incoming arc from the sources. Select one of such sinks as the *representative* of the clause. For each variable x_i, determine its truth value as follows: if v_i is reached by some representative, the value of x_i is in accordance with the parity of the depth of v_i; that is, $x_i = \mathrm{T}$ if v_i is of an odd depth and $x_i = \mathrm{F}$ otherwise. If v_i cannot be reached by any representatives, then its value is don't care. Note that the assignment does not conflict for any x_i since T is an in-tree and thus T is bipartite. By the way of assigning values to the variables, the literal corresponding to the representative is assigned to T, thus every clause has at least one literal assigned to T. Hence ϕ is satisfied by $\boldsymbol{x}$. □

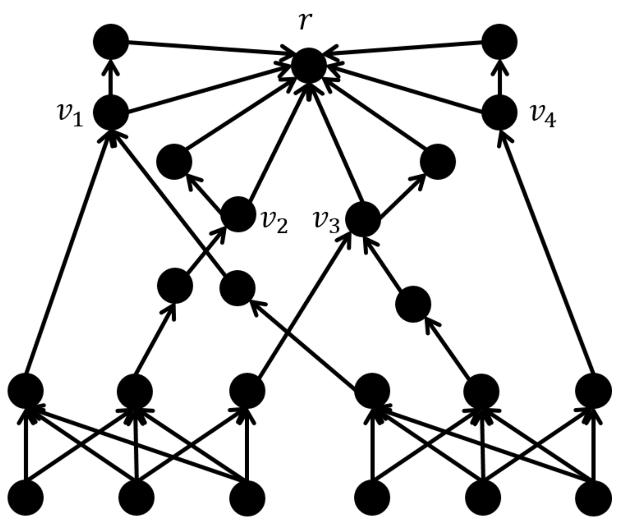


Figure: 5.2: D_ϕ of the boolean formula $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$.

Chapter 6

Conclusion

Motivated by a development of new technologies for relaxation, this thesis has focused on relaxations using *finite groups*. We have presented a new variant of Hamiltonian cycle problem, the parity Hamiltonian cycle problem, for undirected and directed graphs. For the undirected version, we have given a complete characterization of graphs having PHC's when $z \geq 4$, and showed the problem is in P as a corollary. We have also showed that the problem is NP-complete when $z \leq 3$. Then, we are involved in the case $z = 3$, and showed some graph classes for which the problem is in P. It is open if the PHC_3 problem is in P for three-edge-connected graphs. For directed graphs, we gave two different characterizations of graphs having PHC's, and from that, we have derived a linear time algorithm. We have proposed a polynomial time algorithm to construct a PHC in a directed graph. We have further extended the problem to the modulo p version, and showed a characterization similar to the one of PHC's.

For the original Hamiltonian cycle problem, we have also studied the Hamiltonicity of the covering graphs of trees. We have showed the characterization of Batagelj et al. [4] is applicable to two wider classes; both are defined by the path partitions with some extra conditions. For both classes, we proposed a linear time algorithm to test if the input graph satisfies the condition. We have also studied the case when some labels are not coprime to the order of the given cyclic group, and showed a sufficient condition of Hamiltonicity for a path type covering graphs.

We have also introduced the odd depth tree problem by relaxing the constraint on the diameter by parity. For bipartite graphs, we have showed a Hall-type characterization of odd depth trees, and showed the problem is solved in polynomial time. For non-bipartite graphs, we have showed the problem is NP-complete. We have also investigated the variant of the problem for directed graphs, the odd depth in-tree problem. For the directed bipartite graphs we have showed a characterization of graphs having an odd depth in-trees which is similar to the undirected case. For non-bipartite DAG's, we have showed the problem is NP-complete.

Our ultimate goal is to find structures of problems in NP, and our future work is to study how the (simple) characterizations for relaxed variants can be used to characterize the original problems. For this purpose, we will be concerned with variants of NP-hard problems changing the strength of relaxation using finite groups. The author believes the approach of relaxing using finite groups is prospective, and will make progress to derive good characterizations of NP-hard problems.

Reference

[1] O. Aichholzer, T. Hackl, M. Hoffmann, A. Pilz, G. Rote, B. Speckmann, B. Vogtenhuber, Plane graphs with parity constraints, WADS, **9** (2009), 13–24.

[2] B. Alspach, The classification of Hamiltonian generalized Petersen graphs, Journal of Combinatorial Theory B, **34** (1983), 293–312.

[3] J. Bang-Jensen, G. Z. Gutin: Digraphs Theory, Algorithms and Applications, Springer, 2008.

[4] V. Batagelj, T. Pisanski, Hamiltonian cycles in the Cartesian product of a tree and a cycle, Discrete Mathematics, **38** (1982), 311–312.

[5] M. Beck, T. Zaslavsky, The number of nowhere-zero flows on graphs and signed graphs, Journal of Combinatorial Theory B, **96** (2006), 901–908.

[6] K. Bérczi, S. Iwata, J. Kato, Y. Yamaguchi, Making bipartite graphs DM-irreducible, arXiv.org e-print archive abs/1612.08828.

[7] N. Biggs, E. Lloyd, R. Wilson, Graph Theory, 1736-1936, Oxford University Press, 1986.

[8] S. Boyd, R. Sitters, S. van der Ster, L. Stougie, TSP on cubic and subcubic graphs, Lecture Notes in Computer Science, **6655** (2011), 65–77

[9] S. Boyd, S. Iwata, K. Takazawa, Finding 2-factors closer to TSP walks in cubic graphs, SIAM Journal on Discrete Mathematics, **27** (2013), 918–939.

[10] A. Brandstaedt, V.B. Le, J.P. Spinrad, Graph Classes, A Survey, Society for Industrial and Applied Mathematics, 1987.

[11] R.C. Brigham, R.D. Dutton, P.Z. Chinn, F. Harary, Realization of parity visits in walking a graph, The College Mathematics Journal, **16** (1985), 280–282.

[12] D. P. Bunde, K. Milans, D. B. West, H. Wu, Optimal strong parity edge-coloring of complete graphs, Combinatorica, **28** (2008), 625–632.

[13] P.A. Catlin, A reduction method to find spanning Eulerian subgraphs, Journal of Graph Theory, **12** (1988), 29–44.

[14] M. Chudonovsky, J. Geelen, B. Gerards, L. Goddyn, M. Lohman, P. Seymour, Packing non-zero A-paths in group-labelled graphs, Combinatorica, **26** (2006), 521–532.

[15] V. Chvàtal, P. Erdös: A note on Hamiltonian circuits, Discrete Math, **2** (1972), 111–113.

[16] S. J. Curran, J. A. Gallian, Hamiltonian cycles and paths in Cayley graphs and digraphs – A survey, Discrete Mathematics, **156** (1996), 1–18.

[17] J. S. Deogun, G. Steiner, Polynomial algorithms for Hamiltonian cycle in cocomparability graphs, SIAM J. Comput., **23** (1992), 520–552.

[18] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, W. P. Thurston, Word processing in groups, A. K. Peters, Ltd., Natick, MA, USA, 1992.

[19] L. Euler, Solutio problematis ad geometriam situs pertinentis, Commentarii Academiae Scientiarum Imperialis Petropolitanae **8** (1736) 128–140 = Opera Omnia (1) **7** (1911-56), 1–10.

[20] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, R. de Wolf, Exponential lower bounds for polytopes in combinatorial optimization, Journal of the ACM, **62** (2015), 17.

[21] S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, R. de Wolf, Linear vs. semidefinite extended formulations, exponential separation and strong lower bounds, Proc. of STOC 2012, 95–106.

[22] A. Frank, Z. Király, Graph orientations with edge-connection and parity constraints, Combinatorica, **22** (2002), 47–70.

[23] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 2009.

[24] M. R. Garey, D. S. Johnson, R. E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, SIAM Journal on Computing, **5** (1976), 704–714.

[25] L. Gouveia, C. Requejo, A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem, European Journal of Operational Research, **132** (2001), 539–552.

[26] J. L. Gross, Voltage graphs, Discrete Mathematics, **9** (1974), 239–246.

[27] J. L. Gross, T. W. Tucker, Topological graph theory, Courier Corporation, 1987.

[28] D. Gusfield, Connectivity and edge-disjoint spanning trees, Information Processing Letters, **16** (1983), 87–89.

[29] D. Hartvigsen, Extensions of Matching Theory, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1984

[30] D. Hartvigsen, Y. Li, Maximum cardinality simple 2-matchings in subcubic graphs, SIAM J. Optim., **21** (2011), 1027–1045.

[31] P. Hell, H. Nishiyama, L. Stacho, Hamiltonian cycles in covering graphs of trees. In: X. Gao, H. Du, M. Ha (eds) Combinatorial Optimization and Applications. COCOA 2017. Lecture Notes in Computer Science, vol 10628. Springer, Cham

[32] R.-W. Hung, M.-S. Chang, Linear-time algorithms for the Hamiltonian problems on distance-hereditary graphs, Theoretical Computer Science, **341** (2005), 411–440.

[33] B. Jackson and N.C. Wormald, K-walks of graphs, Australas. J. Combin., **2** (1990), 135–146.

[34] M. Joglekar, N. Shah, A. A. Diwan, Balanced group-labeled graphs, Discrete Mathematics, **312** (2012), 1542–1549.

[35] N. Kakimura, K. Kawarabayashi, Fixed-parameter tractability for subset feedback set problems with parity constraints, Theoretical Computer Science, **576** (2015), 61–76.

[36] R.M. Karp, Reducibility among combinatorial problems, Proc. Complexity of Computer Computations 1972, 85–103.

[37] K. Kawarabayashi, B. Reed, Highly parity linked graphs, Combinatorica, **29** (2009), 215–225.

[38] K. Kawarabayashi, P. Wollan, Non-zero disjoint cycles in highly-connected group-labelled graphs, Electric Notes in Discrete Mathemactics, **25** (2002), 271–275.

[39] Y. Kobayashi, S. Toyooka, Finding a shortest non-zero path in group-labeled graphs via permanent computation, Algorithmica, **77** (2017), 1128–1142.

[40] M. Kochol, Polynomials associated with nowhere-zero flows, Journal of Combinatorial Theory B, **84** (2002), 260–269.

[41] B. Korte, J. Vygen, Combinatorial Optimization, Theory and Algorithms, Fifth Edition, Springer-Verlag, 2012.

[42] K. Kutner, R. Marušič, Hamilton cycles and paths in vertex-transitive graphs – Current directions, Discrete Mathematics, **309** (2009), 5491–5500.

[43] S. Lakshmivarahan, J-S Jwo, S.K. Dhall, Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey, Parallel Computing, **19** (1993), 361–407.

[44] D. Lokshtanov, M. S. Ramanujan, Parameterized tractability of multiway cut with parity constraints, ICALP 2012, 750–761.

[45] L. Lovász, The matroid parity problem, Unpublished manuscript, University of Waterloo, Waterloo, Ontario, 1979.

[46] C. St. J. A. Nash-Williams, Edge-disjoint spanning trees of finite graphs, Journal of the London Mathematical Society, **36** (1964), 445–450.

[47] H. Nishiyama, Y. Kobayashi, Y. Yamauchi, S. Kijima, M. Yamashita, The parity Hamiltonian cycle problem, Discrete Mathematics, **341** (2018), 606–626.

[48] H. Nishiyama, Y. Yamauchi, S. Kijima, M. Yamashita, The parity Hamiltonian cycle problem in directed graphs. In: R. Cerulli, S. Fujishige, A. Mahjoub (eds) Combinatorial Optimization. ISCO 2016. Lecture Notes in Computer Science, vol 9849. Springer, Cham

[49] C. H. Papadimitriou, M. Yannakakis, The complexity of restricted spanning tree problems, Journal of Association for Computing Machinery, **29** (1982), 285–309.

[50] T. Pisanski, J. Žerovnik, Hamilton cycles in graph bundles over a cycle with tree as a fibre, Discrete Mathematics, **309** (2009), 5432–5436.

[51] J. Roskind, R. E. Tarjan, A note on finding minimum-cost edge-disjoint spanning trees, Mathematics of Operations Research, **10** (1985), 701–708.

[52] A. Schrijver, Combinatorial Optimization, Springer, 2003.

[53] P. Seymour, C. Thomassen, Characterization of even directed graphs, Journal of Combinatorial Theory B, **42** (1987), 36–45.

[54] S. Tanigawa, Y. Yamaguchi, Packing non-zero A-paths via matroid matching, Discrete Applied Mathematics, **214** (2016), 169–178.

[55] W. Tutte, A theorem on planar graphs, Trans. Am. Math. Soc. **82** (1956), 309–324.

[56] M. Yannakakis, Expressing combinatorial optimization problems by linear programs, Journal of Computer and System Sciences, **43** (1991), 441–466.