Durham University

Durham E-Theses

Reconfigurations of Combinatorial Problems: Graph Colouring and Hamiltonian Cycle

LIGNOS, IOANNIS

How to cite:

LIGNOS, IOANNIS (2017) *Reconfigurations of Combinatorial Problems: Graph Colouring and Hamiltonian Cycle*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/12098/

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

المنارة للاستشارات

www.manaraa.com

Academic Support Office, Durham University, University Office, Old Elvet, Durham DH1 3HP
e-mail: e-theses.admin@dur.ac.uk Tel: +44 0191 334 6107
http://etheses.dur.ac.uk

المنارة للاستشارات
www.manaraa.com

Reconfigurations of Combinatorial Problems

Graph Colouring and Hamiltonian Cycle

Ioannis Lignos

A Thesis presented for the degree of

Doctor of Philosophy

School of Engineering and Computing Sciences
University of Durham
England

July 2016

المنارة للاستشارات

www.manaraa.com

Dedicated to my parents, Niki and Michalis.

المنارة للاستشارات

www.manaraa.com

Reconfiguration of Combinatorial Problems

Graph Colouring and Hamiltonian Cycle

Ioannis Lignos

Abstract

We explore algorithmic aspects of two known combinatorial problems, Graph Colouring and Hamiltonian Cycle, by examining properties of their solution space. One can model the set of solutions of a combinatorial problem P by the solution graph $R(P)$, where vertices are solutions of P and there is an edge between two vertices, when the two corresponding solutions satisfy an adjacency reconfiguration rule. For example, we can define the reconfiguration rule for graph colouring to be that two solutions are adjacent when they differ in colour in exactly one vertex.

The exploration of the properties of the solution graph $R(P)$ can give rise to interesting questions. The connectivity of $R(P)$ is the most prominent question in this research area. This is reasonable, since the main motivation for modelling combinatorial solutions as a graph is to be able to transform one into the other in a stepwise fashion, by following paths between solutions in the graph. Connectivity questions can be made binary, that is expressed as decision problems which accept a 'yes' or 'no' answer. For example, given two specific solutions, is there a path between them? Is the graph of solutions $R(P)$ connected?

In this thesis, we first show that the diameter of the solution graph $R_\ell(G)$ of vertex ℓ-colourings of k-colourable chordal and chordal bipartite graphs G is $\mathcal{O}(n^2)$, where $\ell \geq k+1$ and n is the number of vertices of G. Then, we formulate a decision problem on the connectivity of the graph colouring solution graph, where we allow extra colours to be used in order to enforce a path between two colourings with no path between them. We give some results for general instances and we also explore what kind of graphs pose a challenge to determine the complexity of the problem for general instances. Finally, we give a linear algorithm which decides whether there is a path between two solutions of the Hamiltonian Cycle Problem for graphs of maximum degree five, and thus providing insights towards the complexity classification of the decision problem.

للاستشارات المنارة

www.manaraa.com

المنارة للاستشارات
www.manaraa.com

Declaration

The work in this thesis is based on research carried out at the Algorithms and Complexity at Durham (ACiD) research group, School of Engineering and Computing Sciences, University of Durham, England.

Chapter 4 is a result of joint research which took place during my studies, and has resulted in the corresponding publications [6, 7], also mentioned in the chapter. No other part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Copyright © 2016 by Ioannis Lignos.

"The copyright of this thesis rests with the author. No quotations from it should be published without the author's prior written consent and information derived from it should be acknowledged".

للاستشارات المنارة

www.manaraa.com

www.manaraa.com

Acknowledgements

The utmost gratitude goes to Dr. Matthew Johnson, my PhD supervisor, who introduced me into the fascinating world of reconfiguration problems, as part of an EPSRC-funded project. I am glad that Matthew gave me enough freedom to choose research questions which I thought suited my interests best, and urged me to become an independent researcher from the first instance. At the same time, he introduced me to other researchers and encouraged me to hone the very important skill of working collaboratively in research teams. On personal matters, Matthew showed a lot of understanding and has always been positive and encouraging during my research studies and for any future career plans.

Durham felt like home, where I met my supervisor Matthew and the rest of the ACiD group, who were always keen to discuss new graph theory problems and to invite other researchers from all over the world. This is where I met colleagues who are also now friends, like Viresh Patel – by the way my 'Patel number' is 1!

A bit more back in time in Liverpool, many thanks go to Prof. Leszek Gasieniec, my MSc thesis supervisor and co-author in a paper. Leszek convinced me to take a graph theory project for my dissertation, as he believed that my background fitted his project well. He did not have to try hard, as anyway becoming a software engineer instead seemed to me perhaps only a way to survive and not as fascinating as delving into the theory side of things.

Becoming a researcher has been an unimaginable experience in many aspects. I will never forget the moment when in my first working meeting, I faced something which I had considered an exaggerated rumour; that a professor could stare at a piece of paper while having no answers

www.manaraa.com

to the question at the top of the page, sometimes asking me, his student, whether I had any idea what the answer might be! It was not an academic joke towards the un-initiated student. It was a serious question. Suddenly, although I was at a cafe, it felt as if we had reached the boundaries of the known universe and were trying to push hard to extend them, devising a strategy using ‘pen and paper’, surrounded by people still grounded and having no idea what it might be that is going on. It was my normality from that moment and on.

Many thanks to all the co-authors for the fruitful discussions both at Liverpool and Durham, and to the EPSRC for the financial support on the project linked to my PhD.

Last but not least, thanks to my family for their support, and to my partner for her continuous support in pursuing my dreams.

للاستشارات المنارة

www.manaraa.com

Contents

المنارة للاستشارات
www.manaraa.com

www.manaraa.com

المنارة للاستشارات

www.manaraa.com

المنارة للاستشارات
www.manaraa.com

List of Figures

المنارة للاستشارات

www.manaraa.com

www.manaraa.com

Chapter 1

Introduction

We are interested in the area of Reconfigurations of Combinatorial Problems, focussing particularly on two well-known problems from graph theory: *Graph Colouring* and *Hamiltonian Cycle*. Some of these questions are expressed as decision problems, that is they accept a 'yes' or 'no' answer. Our aim is to explore algorithmic and computational aspects of such decision problems and of other properties of reconfigurations.

In order to define the *reconfiguration* version of a combinatorial problem P, it is necessary to define an adjacency relation between solutions of P. This relation is called the *reconfiguration rule* and is chosen to be a minimal difference between two solutions. That is why, we call two solutions *adjacent* when we can obtain one from the other by applying the reconfiguration rule once. The single application of the rule is called *reconfiguration step*. For example, if problem P is Graph Colouring, then the minimal reconfiguration rule is for two colourings to differ in colour on exactly one vertex. For other problems, or even other variants of Graph Colouring, the reconfiguration rule is not (naturally) unique, as there is more than one way to define a minimal symmetric difference between two solutions. Thus, the same problem P may have more than one reconfiguration version corresponding to respective reconfiguration rules.

A *reconfiguration sequence* is a sequence $s_1, s_2, ..., s_l$ of solutions of P, where s_i and s_{i+1} are adjacent for every $i < l$.

المنارة للاستشارات

www.manaraa.com

The main question for a reconfiguration problem is the following:

- P-PATH (or P Reconfiguration)
  - Instance: A combinatorial problem P and two of its solutions s and s'.
  - Question: Is there a a reconfiguration sequence starting with s and ending with s'?

The reconfiguration graph $R(P)$

Definition 1.0.1. $R(P)$ is a graph whose vertices are solutions of P and there is an edge between every pair of adjacent solutions.

Thus, P Reconfiguration can also be defined in terms of the solution graph $R(P)$ of P. For example, in the case of Graph Colouring, the solution graph is the set of all colourings of a graph G and any two colourings are adjacent in the solution graph when they differ in colour on exactly one vertex. As mentioned above, changing the reconfiguration rule creates a new reconfiguration version of P, and accordingly a new edge set for $R(P)$.

Since we can treat solutions of a problem P as vertices of a graph, we can also give an alternative definition of the reconfiguration sequence between two solutions.

- P-PATH (or P Reconfiguration)
  - Instance: A combinatorial problem P and two of its solutions s and s'.
  - Question: Is there a path between s and s' in $R(P)$?

We can also call a path between two solutions s and s' in $R(P)$ as a *reconfiguration* of s to s'.

1.1 Reconfiguration Questions and their Decision Problem

While P-PATH is naturally the fundamental question of a reconfiguration problem, there are additional interesting questions to ask, when one explores properties of the solution graph $R(P)$.

Reconfiguration questions on a combinatorial problem P can be defined as above, but also in terms of the properties of the solution graph $R(P)$. For example, the P-PATH question is named by

المنارة للاستشارات
www.manaraa.com

the respective property of $R(P)$. That is, the existence of a reconfiguration between two solutions is equivalent to *finding a path* between the two solutions in $R(P)$.

Other important questions regarding feasible solutions of a problem P and its reconfiguration graph are the following:

- P-CONN
  - Instance: A combinatorial problem P with a reconfiguration rule between adjacent solutions.
  - Question: Is there a path between *any* two solutions s and s' in $R(P)$?

- P-DIAM
  - Question: What is the longest shortest path between any two solutions of P?

In other words, P-CONN asks whether $R(P)$ is connected, and P-DIAM asks what is the diameter of $R(P)$. Note that $R(P)$ can be exponential in size and thus its diameter. We will give more details on the properties of $R(P)$ in the Chapters to follow.

1.1.1 Computational Complexity

Computational complexity studies how hard a problem is to solve in terms of time and space resources. Since the complexity classification of a reconfiguration problem is one of the main tasks of this area, we present briefly the complexity classes that we will come across. The complexity class P contains all problems which can be solved in polynomial time. A problem is in the class NP, when it can be computed in non-deterministic polynomial time. That is, given a solution to a problem in NP we can verify that it is a valid solution in polynomial time. Since problems in P are solved in polynomial time, then they are also in NP, as finding a solution is at the same time a verification that it is indeed a solution.

In this thesis, we say that a decision problem P_1 can be *reduced* to a decision problem P_2 or that there is a *reduction* from P_1 to P_2, when there is a polynomial time algorithm which maps (all the) 'yes' (resp. 'no') instances of P_1 to 'yes' (resp. 'no') instances of P_2.

المنارة للاستشارات

www.manaraa.com

Problems in NP which are such that any other problem in NP can be reduced to them are called NP-complete. Moreover, a problem is NP-hard, if to compute the solution to its instance is at least as hard as doing so for an NP-complete problem.

Similarly to classifying problems in terms of polynomial time, we can ask whether polynomial space is enough to compute a solution. A problem is in the class PSPACE (resp. NPSPACE), when the space needed in order to compute (resp. verify) a solution to the problem is of polynomial size. PSPACE is a class of importance for reconfiguration problems, as both the P-PATH and P-CONN problems are in NPSPACE, and by Savitch's theorem in [70] which proves that PSPACE = NPSPACE, they are in PSPACE.

It is easy to see why they are in NPSPACE. An instance of P-PATH of a combinatorial problem P, consisting of the problem P and two of its solutions, can be described in polynomial space in relation to the size of the original problem P. Then, given a sequence of feasible solutions of P which describes a path between the two given solutions, we can verify that each of them is a solution using polynomial space – note that we do not have to verify all the solutions at once or store them, as P-PATH is a decision problem and thus does not necessarily describe the path between the two given solutions, which can be of exponential size.

In relation to completeness and polynomial space, problems that are in the PSPACE-complete class are problems in PSPACE such that we can reduce to them any other problem in PSPACE.

For more details on computational complexity, see the books from Garey and Johnson [30] or Papadimitriou [67].

1.1.2 Graph Theory

We introduce some of the necessary graph-theoretical terminology and notation that is used throughout the thesis, while some more specialised definitions will be given in each chapter. For other basic set-theoretic and graph theory terminology not found here, see Diestel [22].

We consider undirected finite graphs that have no loops and no multiple edges. Given a graph $G = (V, E)$, we denote the set of its vertices by V and the set of its edges by E. For a subset

www.manaraa.com

$S \subseteq V$, the graph $G[S]$ denotes the subgraph of G *induced by* S, i.e., the graph with vertex set S and edge set $\{uv \in E \mid u, v \in S\}$. We write $G - S = G[V \setminus S]$.

For a subgraph $G' \subseteq G$, not necessarily induced, we will denote the set of its vertices by $V(G')$ and the set of its edges by $E(G')$, except otherwise defined. The set of *neighbours* of a vertex u in a subgraph $G' \subseteq G$ is denoted by $N_{G'}(u) = \{v \mid uv \in E(G')\}$. Often, we may omit the subscript for the subgraph, if there is no ambiguity.

If u has no neighbours, then we say that u is an *isolated* vertex. If u and v are adjacent and have no other neighbours, then the edge uv is called an *isolated* edge.

A *(vertex) colouring* of a graph G is a mapping $c : V \to \{1, 2, \ldots\}$ such that $c(u) \neq c(v)$ whenever $uv \in E$. Here, $c(u)$ is referred to as the *colour* of u. We write $c(U) = \{c(u) \mid u \in U\}$ for $U \subseteq V$. Then a k*-colouring* of G is a colouring c of G with $c(V) \subseteq \{1, \ldots, k\}$. If G has a k-colouring, then G is called k*-colourable*. The *chromatic number* of G denoted by χ_G is the smallest value of k for which G is k-colourable. If G is 2-colourable, then G is also called *bipartite*. We denote the reconfiguration graph of the k-colourings of a graph G by $R_k(G)$. The *colouring number* $col(G)$ of a graph G (or *degeneracy of* G) is the maximum minimum degree of any subgraph of G, i.e. $col(G) = \max\{\delta(H) | H \subseteq G\}$.

The number of vertices of a graph G is $|V| = n$, except if stated otherwise. Thus, when we calculate time and space complexities in relation to the number of vertices of G, then we do that in relation to n.

The x-vertex *path* is the graph with vertices $v_1, \ldots, v_x$ and edges $v_i v_{i+1}$ for $i = 1, \ldots, x-1$. If $v_x v_1$ is also an edge, then we obtain the x-vertex *cycle*. The *length* of a path or a cycle is the number of its edges. A graph is called *connected* if, for every pair of distinct vertices v and w, there exists a path connecting v and w. A graph G is k*-connected*, $k \in \mathbb{N}$, if $G - X$ is connected for every set $X \subseteq V$ with $|X| \leq k$.

A *hamiltonian cycle* is an n-vertex cycle of G, where $|V| = n$.

المنارة للاستشارات
www.manaraa.com

1.2 Motivation

Reconfiguration problems have diverse motivations. The most obvious one is obtained straight from their definition, that is to investigate how one can transform a given solution, configuration or setting s_1 of a (combinatorial) structure S to another (desired) setting s_2, while abiding to the adjacency or reconfiguration rule. This rule sets a constraint on how extensive the change between two settings can be, and thus all other invalid shortcuts are excluded. Compared to the original problem P of finding a desired setting of the structure, such as the ones given as an input, the search of the reconfiguration version of P is for a valid transformation between *verified* settings of C. Thus, reconfiguration problems are interesting on their own merit, as search problems within the realm of solutions of an existing known problem P.

The fact that the input comprises of existing settings of a structure C makes reconfiguration problems useful in modelling any situation, where one needs to gradually migrate from an old to a new setting of a structure across any discipline, and where the option of constructing or re-setting the structure directly to the new setting is not an option. For example, in the Power Supply problem [44], where there is an allocation of suppliers to customers, two customers cannot change a supplier at the same time, and also we cannot disconnect many customers completely, that is disconnected from all suppliers.

Looking for a plethora of situations where the reconfiguration version of a problem P can be applied, one could simply consider situations that are modelled by P, and for which the search for a gradual transformation to a new solution is useful. For example, it is well known that Graph Colouring can model resource allocation or scheduling (e.g. airline timetables). Given any two graph colourings modelling those problems, the reconfiguration problem searches for a path between the two colourings and thus models a way to move from one configuration of our situation to a desired one passing through only feasible solutions. On the other hand, the motivation could stem directly from the situation we wish to model. For example, the evolution of a genotype where only single mutations can occur and all genotypes must be above a certain fitness threshold is naturally a reconfiguration problem with a specified transformation rule, which maintains the given threshold.

المنارة للاستشارات
www.manaraa.com

Finally, whereas reconfiguration problems work on existing solutions of a problem P, an understanding of the geometry of the solution (or reconfiguration) space of a problem P can provide insight into the performance of algorithms and heuristics [2] in solving P. Having a minimal transformation between solutions defining the nature of the reconfiguration of P, provides a new way of investigating how different solutions of P interact with or connect to each other.

1.3 Outline of the Thesis

In Chapters 2 and 3, we give an overview of related work done on reconfiguration problems. We can separate these results in two categories. On one level, in Chapter 3, we look at the work done on reconfiguration problems in general. We made an effort to include most of the existing results which are defined similarly to the definition of the general problem in Chapter 1. On another level, in Chapter 2, we look at existing work closest to our results, that is on *Graph Colouring Reconfiguration*. In so doing, we describe the main methods used by the respective authors accompanied by key points in their proofs, as this may give a better understanding of our own work later. Note that there are no results on *Hamiltonian Cycle Reconfiguration* of which we are aware.

In Chapters 4 and 5, we present our results on Graph Colouring Reconfiguration. We show that, under certain conditions, the reconfiguration graph of vertex colourings is connected for the chordal and chordal bipartite classes of graphs and that its diameter is $\mathcal{O}(n^2)$, in Chapter 4. This chapter is based on the corresponding publications [6, 7]. In Chapter 5, we give some more results on Graph Colouring Reconfiguration, attempting to answer a question posed by Cereceda [15] on the number of extra colours one needs to enforce a path between two colourings which are not connected.

Another piece of our research appears in Chapter 6, on Hamiltonian Cycle Reconfiguration. We state the problem and provide definitions and observations on general instances. The main result in the chapter is a linear algorithm for graphs of maximum degree five, as well as a simplified version for graphs of maximum degree four.

Finally, in Chapter 7, we summarise and discuss our results, and set showcase some open ques-

المنارة للاستشارات

www.manaraa.com

tions and future directions.

www.manaraa.com

Chapter 2

Graph Colouring Reconfiguration - A Review

In this chapter, we present results on Graph Colouring Reconfiguration which precede our work. We give enough detail and often key points of proofs in [16–19] as a means of introducing the reader to techniques used in reconfigurations of graph colourings.

2.1 The Reconfiguration Graph of Vertex Colourings

The reconfiguration graph of vertex colourings was introduced in [17] and [15], where the authors call it the *colour graph*. We will use the term *reconfiguration graph* for any reconfiguration problem we refer to, and specify what its vertices are if it is not implied by the context, e.g. colourings, hamiltonian cycles etc. Otherwise and where it is helpful, we also use terminology from the respective published work.

Definition 2.1.1. Let G be a k-colourable graph. Then $R_k(G)$ is a graph whose vertices are all the k-colourings of G and there is an edge between two k-colourings if they differ in colour on exactly one vertex of G.

A k-colouring of G which corresponds to an isolated vertex in $R_k(G)$ is called *frozen*. The term

www.manaraa.com

is intuitively appropriate, as for such a k-colouring, there is no vertex in G such that it can be given a colour different than its current. For example, given a K_3, a graph which is a triangle, then all of its 3-colourings are frozen, since each vertex receives one of the three colours and for every vertex v in the triangle all three colours are used either to colour v or one of its neighbours.

2.1.1 The Decision Problems

The decision problems studied for Colouring Reconfiguration are not other than the reconfiguration questions P-PATH and P-CONN, defined in Section 1. When problem P is Colouring Reconfiguration, the solutions are k-colourings of a k-colourable graph G. We give precise definitions for completeness.

- k-COLOUR-PATH (or Colouring Reconfiguration)
  - Instance: A k-colourable graph G and two k-colourings of G, α and β.
  - Question: Is there a path between α and β in $R_k(G)$?

If $R_k(G)$ is connected, then we can say that G is k-*mixing* [15] (or just *mixing* if the number of k colours is implied). This alternative definition derives from an application of colourings in their solution space, which requires a graph to be mixing; see Section 3.4.2 on Glauber Dynamics and the rapid mixing of Markov Chains.

- k-MIXING
  - Instance: A k-colourable graph G.
  - Question: Is G k-MIXING? That is, is $R_k(G)$ connected?

Cereceda, van den Heuvel and Johnson [19] give the very first results on the connectedness of $R_k(G)$. First, they explore the values of k for which $R_k(G)$ is guaranteed to be connected. Then, by looking at the case $k = \chi(G)$, they show that $R_k(G)$ is not connected for k being 2 or 3, while for $k \geq 4$ there are graphs for which $R_k(G)$ is connected and graphs for which it is not.

In [18], they look at the computational complexity of deciding whether a graph is 3-mixing. Because of the results in [19] for 3-chromatic graphs, the decision (problem) is narrowed down

www.manaraa.com

to the case of bipartite graphs and they show that it is coNP-complete for bipartite graphs and in P for planar bipartite graphs.

The combined results [10] of Cereceda, van den Heuvel, and Johnson[19] and Bonsma and Cereceda [13] give a dichotomy for Graph Colouring Reconfiguration. For a k-colourable graph G and two of its k-colourings k-COLOUR PATH is in P for $k \leq 3$, and PSPACE-complete for $k \geq 4$.

We will have a closer look at how these results were obtained. Note that we only sketch any proofs presented.

2.2 Connectedness of $R_k(G)$

In [17], Cereceda et al. first look for sufficient conditions such that a graph G is k-mixing.

A colour c is *available* to a vertex u in a colouring γ, when there is no neighbour x of u such that $\gamma(x) = c$.

There is no bound depending on the chromatic number $\chi(G)$ which can give a guarantee that a graph is k-mixing. For example, we can consider a graph which is the complete bipartite graph $K_{m,m}$ missing the edges of a perfect matching. Such a graph is k-mixing for every $k \geq 3$, except for $k = m$, as $R_m(G)$ contains a *frozen* colouring, that is a colouring which is isolated in $R_m(G)$. To colour $K_{m,m,}$ with this colouring, colour the m vertices which belong to the same independent set with different colours (from 1 to m). Then, there is one only way left to colour the rest m vertices, as each one of them has only one available colour.

On the other hand, for the colouring number $col(G)$ (or degeneracy) of G, the authors give the following result:

Theorem 2.2.1. *For any graph G and integer $k \geq col(G) + 2$, $R_k(G)$ is connected.*

This improves the lower bound guaranteeing a graph to be k-mixing in terms of the maximum degree of G as given in [50], which was $\Delta(G) + 2$, because $col(G) \leq \Delta(G)$.

المنارة للاستشارات
www.manaraa.com

Connected bipartite graphs have chromatic number 2 and exactly two 2-colourings, therefore $R_2(G)$ is disconnected with two isolated vertices – for example K_2 is the smallest such graph. This implies that 2-MIXING answers 'NO' for every instance.

Thus, the first interesting case for k-MIXING is when $k = 3$.

A Weighting System for 3-Colourings

A weighting system can be defined on the edges and cycles of a graph G, which has been given a 3-colouring α, with colours $1, 2$ and 3 [17].

Given an orientation for an edge uv, say from u to v, then the weight of uv is $+1$, when uv is coloured 12, 23, 31, and -1 when it is coloured 13, 21 or 32. The weight of a cycle C with a colouring α is denoted by $W(\overrightarrow{C}, \alpha)$ and is the sum of the weights of all of its edges. To calculate $W(\overrightarrow{C}, \alpha)$, one has to choose the same orientation for the edges of C, e.g. clock-wise. Likewise, for the weight of a path, one has to choose the same orientation for all of its edges.

2.3 3-MIXING

Lemma 2.3.1. *Given a graph G coloured with a colouring α, let $\overrightarrow{C}$ be an oriented cycle. If $W(\overrightarrow{C}, \alpha) \neq 0$, then G is not 3-mixing.*

If we consider the weights of the cycle in two adjacent colourings in $R_3(G)$, then their weight should be the same, as changing the colour in only one vertex changes the sign of the weight of both incident edges, that is from $+1$ to -1 and vice versa. For the given colouring α and cycle C of G, we can choose two specific colours and swap their occurrence on the vertices on which they appear receiving a different colouring β with $W(\overrightarrow{C}, \beta) = -W(\overrightarrow{C}, \alpha)$. Since they have different weights, we know that they are in different components of $R_3(G)$. Thus, G is not 3-mixing.

Given that a 3-chromatic graph contains at least one cycle of odd length (and thus having non-zero weight), the last lemma immediately implies that 3-chromatic graphs are not 3-mixing.

المنارة للاستشارات
www.manaraa.com

2.3.1 Mixing 3-Colourings in Bipartite Graphs

The remaining case for 3-colourable graphs is when they are bipartite, which is essentially when the decision problem 3-MIXING is not trivial to answer, as its instances cannot be classified as mixing or non-mixing only. This is what is studied by Cereceda van den Heuvel, and Johnson [18], proving that 3-mixing is coNP-complete for bipartite graphs, but in P for planar bipartite. We briefly look at how these results were obtained.

The following theorem states the main result in [18].

Theorem 2.3.2 (Cereceda et al. [18])**.** *The decision problem* 3*-MIXING is* coNP*-complete.*

A *pinch* on two vertices u and w of a graph G, which are at distance two, is the identification of u and w and the removal of any double edges produced. Accordingly, a graph G is *pinchable* to a graph H, if there exists a sequence of pinches that transforms G into H.

The following theorem gives a characterisation for bipartite graphs which are not 3-mixing. We give a sketch of the original proof.

Theorem 2.3.3 (Cereceda et al. [18])**.** *Let G be a connected bipartite graph. The following are equivalent:*

(i) The graph G is not 3*-mixing.*

(ii) There exists a cycle C in G and a 3*-colouring α of G with $W(\overrightarrow{C}) \neq 0$.*

(iii) The graph G is pinchable to the 6*-cycle C_6.*

Proof. If G is not 3-mixing, then $R_3(G)$ has at least two disconnected components. Taking two colourings, α and β on different components means that there is no path between the two colourings in $R_3(G)$. And thus (i) implies (ii).

G does not contain odd cycles and C_4 has always zero weight, so C must be some cycle with even length and more than 4. If $G = C$, then we can pinch G to C_6. If G is C with some additional chords, then by Lemma 2.3.1 and some thought we can show that there is smaller cycle C' with non-zero weight. With these observations in mind, we can carefully fold G to a cycle C making sure that a cycle of non-zero weight is maintained and then pinch that cycle to a 6-cycle.

www.manaraa.com

Finally, C_6 is the smallest bipartite cycle which is not 3-mixing. Assuming (iii) and taking two 3-colourings of C_6, α and β in different components of $R_3(G)$, it is possible to construct two 3-colourings of G, α' and β', such that α' is obtained from α and β' is obtained from β, by using the reverse sequence of folding G to C_6. The construction can be done such that the colouring of G has the same weight with the respective colouring of C_6. Thus, the new colourings are not connected in $\mathcal{R}_3(G)$ which implies (i). $\square$

Since Theorem 2.3.3 gives a characterisation of when a bipartite graph is 3-mixing, then how hard it is to decide this depends on the hardness of the equivalent characterisations.

The decision problem of whether a bipartite graph is pinchable to C_6 is NP-complete [18] by a reduction from the problem of whether the same graph is retractable to C_6 [77]. Therefore, deciding whether G is not 3-mixing is also NP-complete by Theorem 2.3.3 and thus 3-MIXING is coNP-complete.

3-MIXING for Planar Bipartite Graphs

Theorem 2.3.4 (Cereceda et al. [18]). *Restricted to planar bipartite graphs, the decision problem 3-MIXING is in P.*

The authors show this by giving a polynomial algorithm which decides the problem. The algorithm is given in a series of claims which can be applied repeatedly, if needed, and using Theorem 2.3.3.

Before we list the claims used, we need the following definitions.

A *drawing* of a graph G on a surface S is a graphical representation of G on S, with each vertex assigned a distinct point on S, and each edge assigned a curve which joins points which correspond to the vertices the edge connects. The drawing is an *embedding* if no pair of curves intersect in points of the surface which do not correspond to vertices. G is *embeddable* on S, if there exists an embedding of G on S. A *planar embedding* of a graph is its embedding on the plane.

We assume that G has a planar embedding. Given a cycle D in G, let $Int(D)$ and $Ext(D)$ be

المنارة للاستشارات
www.manaraa.com

the sets of vertices inside and outside of D, respectively. If both $Int(D)$ and $Ext(D)$ are non-empty, D is said to be *separating*. For D, a separating cycle in G, let $G_{Int}(D) = G - Ext(D)$ and $G_{Ext}(D) = G - Int(D)$. Let a face of G with k edges in its boundary be a k-*face*, and a face with at least k edges in its boundary a $\geq k$-*face*.

Suppose G has a cut-vertex v. Let H_1 be a component of $G - \{v\}$. Then, G_1 is the induced subgraph $G[V(H_1) \cup \{v\}]$ and G_2 is the induced subgraph $G - H_1$. A *block* of G is a maximal connected subgraph of G with no cut-vertex.

The following claims below decide whether specific induced subgraphs of G are 3-mixing by using Theorem 2.3.3. That is, the decision is made based on whether each subgraph has a cycle C with non-zero weight or whether it is pinchable to C_6.

1. If G has a cut-vertex v, then G is 3-mixing if and only if both G_1 and G_2 are 3-mixing.

2. If G is 2-connected and has a planar embedding with a separating 4-cycle D, then G is 3-mixing if and only if $G_{Int}(D)$ and $G_{Ext}(D)$ are both 3-mixing.

3. If G is 2-connected, has a planar embedding with no separating 4-cycle, and every internal face of the embedding is a 4-face, then G is 3-mixing.

4. If G is 2-connected and has a planar embedding with: no separating 4-cycle, an internal $\geq$6-face, and has an $\geq$6-face outer face, then G is not 3-mixing.

2.4 Finding Paths between k-Colourings

2.4.1 3-COLOUR PATH

Cereceda et al. [19] give a polynomial algorithm for 3-COLOUR PATH, which also finds the shortest path between two 3-colourings, if it exists, but only for some instances. Also, the diameter of $R_3(G)$ is $\mathcal{O}(n^2)$.

Thus the main results of this work are the following.

Theorem 2.4.1 ([19]). *The decision problem 3-COLOUR PATH is in P.*

www.manaraa.com

Theorem 2.4.2 ([19]). *Let G be a 3-colourable graph with n vertices. Then the diameter of any component of $R_3(G)$ is $\mathcal{O}(n^2)$.*

We give some of the ideas on how they prove their results.

Let G be a graph with two of its colourings, α and β. An *obstruction* is any structure in G which forbids the existence of a path between the colourings in $R(G)$. A *fixed vertex* v in α of G is a vertex which cannot be recoloured following any sequence of recolourings beginning from α. A *fixed cycle* is a cycle of fixed vertices with respect to a specific colouring. A *fixed path* is a path whose endvertices are fixed. The set of all fixed vertices in a 3-colouring α is denoted by F_α.

Theorem 2.4.3 ([19]). *Let G be a graph on n vertices. Two 3-colourings α and β of G belong to the same component of $R_3(G)$ if and only if*

(i) $F_\alpha = F_\beta$ *and* $\alpha(v) = \beta(v)$ *for each* $v \in F_\alpha$*,*

(ii) for every cycle C in G, $W(\overrightarrow{C}, \alpha) = W(\overrightarrow{C}, \beta)$*,*

(iii) and for every path P between fixed vertices, $W(\overrightarrow{P}, \alpha) = W(\overrightarrow{P}, \beta)$*.*

Thus, for two 3-colourings α and β to belong to the same component of $R_3(G)$, the set of fixed vertices has to be the same in both colourings, and all cycles and fixed paths in G must have the same weight in relation to the two colourings. The algorithm shows either a path between α and β or an obstruction according to the three necessary conditions above.

The algorithm first decides (i) of Theorem 2.4.3 and, if successful, recolours vertices until the target colouring is reached or a cycle or path is found which does not satisfy (ii) and (iii), respectively, of the same Theorem. When the algorithm finds a path, this is done in $\mathcal{O}(n^2)$ number of steps, which proves Theorem 2.4.1. Moreover, the path found is the shortest possible [52].

Since all the steps taken by the algorithm are used to recolour vertices, we immediately know that the diameter of $R_3(G)$ is $\mathcal{O}(n^2)$. And as there is an example when an exhibited shortest path has quadratic length, this implies that the diameter is actually $\mathcal{O}(n^2)$.

المنارة للاستشارات
www.manaraa.com

2.4.2 k-COLOUR PATH, $k \geq 4$

Bonsma and Cereceda [13], prove that for every $k \geq 4$, the k-COLOUR PATH problem is PSPACE-complete. Moreover, there is an infinite class of graphs such that each graph has two colourings which are at super-polynomial distance in $R_k(G)$. We give some more details on how the completeness result is obtained.

Theorem 2.4.4 ([13]). *For every $k \geq 4$, the decision problem k-COLOUR PATH is* PSPACE-*complete. Moreover, it remains* PSPACE-*complete for the following restricted instances:*

(i) bipartite graphs and any fixed $k \geq 4$,

(ii) planar graphs and any fixed $4 \leq k \leq 6$,

(iii) bipartite planar graphs and $k = 4$.

The proof is by a reduction from k-LIST-COLOUR PATH and SLIDING TOKENS. k-COLOUR PATH is reduced to LIST-COLOUR PATH and the latter to SLIDING TOKENS [40]. We now define these problems.

A *token configuration* of a graph G is a set of vertices on which tokens are placed, in such a way that no two tokens are adjacent. In SLIDING TOKENS instances, the vertices of G are separated into *token triangles* (copies of K_3) and *token edges* (copies of K_2), and all these groups of vertices are connected by *link edges* (normal edges). Exactly one token is placed on one of the vertices of each token triangle or edge and can slide towards any other of their vertices. However, the token is not allowed to slide along a link edge, thus always staying on its token triangle or edge.

- SLIDING TOKENS
  - Instance: A graph G and two token configurations of G, T_A and T_B.
  - Question: Is there a sequence of moves transforming T_A into T_B?

Theorem 2.4.5 ([40]). *SLIDING TOKENS is* PSPACE-*complete.*

The problem LIST-COLOUR PATH is only different to k-COLOUR PATH in that there is a *colour list* for each vertex, a list of available colours from which to choose in order to colour (or

المنارة للاستشارات
www.manaraa.com

recolour) the vertex. Therefore, the reconfiguration graph of list-colourings $R_k^L(G)$ contains proper list-colourings of G.

- LIST-COLOUR PATH
  - Instance: A graph G, colour lists $L(v) \subseteq \{1, 2, ...k\}$ for all $v \in V(G)$, and two k-colourings of G, α and β.
  - Question: Is there a path between α and β in $R_k^L(G)$?

The following lemma shows that a LIST-COLOUR PATH instance can be transformed into a k-COLOUR PATH instance, while maintaining the planarity and/or bipartiteness.

Lemma 2.4.6 ([13])**.** *For any $k \geq 4$, a LIST-COLOUR PATH instance (G, L, α, β) with lists $L(v) \subseteq \{1, 2, 3, 4\}$ can be transformed into a k-COLOUR PATH instance (G', α', β') such that the distance between α and β in $R(G, L)$ (possibly infinite) is the same as the distance between α' and β' in $R_k(G')$. Moreover,*

(i) if G is bipartite, this can be done so that G' is also bipartite, for all $k \geq 4$,

(ii) if G is planar, this can be done so that G' is also planar, when $4 \leq k \leq 6$,

(iii) if G is planar and bipartite, this can be done so that G' is also planar and bipartite, when $k = 4$.

Ideas from the Proof of Theorem 2.4.4:

k-COLOUR PATH is in NPSPACE, as given an instance of the problem together with a certificate (a sequence of recolourings that shows the path between the two colourings in $R_k(G)$), then its validity can be checked in polynomial space. And because NPSPACE = PSPACE [70], k-COLOUR PATH is in PSPACE.

SLIDING TOKENS is reduced to LIST-COLOUR PATH and the instances of the latter can then be transformed to k-COLOUR PATH instances, according to Lemma 2.4.6.

SLIDING TOKENS is PSPACE-complete [40] with the reduction given for restricted classes of graphs. Thus, also the reduction from SLIDING TOKENS to LIST-COLOUR PATH in [13] is proven for restricted classes of graphs. We will outline how this reduction can be done and for

www.manaraa.com

which restricted classes.

Given an instance of SLIDING TOKENS, (G, T_A, T_B), we can construct an instance (G', L, α, β) of LIST-COLOUR PATH, such that token configurations correspond to list-colourings and sliding a token in G to a sequence of recolourings. Moreover, the construction can be done such that graph G' is planar and bipartite. The addition of subgraphs with pairs of vertices which cannot receive a specific colouring (these are called *forbidding paths* in [18]) helps build a bijection between the restricted movement of the tokens in graph G and the vertex recolourings between colourings α and β in graph G'.

Recall that the colouring number of a planar graph is at most 5 and of a bipartite planar graph is at most 3. Therefore, from Theorems 2.2.1, 2.4.1, and 2.4.4 we have the following more generalised conclusion:

Theorem 2.4.7. *k-COLOUR PATH is:*

- PSPACE*-complete, for planar graphs and* $4 \leq k \leq 6$*,*
- *in* P*, for planar graphs and* $k \leq 3$ *or* $k \geq 7$*,*
- PSPACE*-complete, for bipartite planar graphs and* $k = 4$*,*
- *in* P*, for bipartite planar graphs and* $k \neq 4$*.*

2.4.3 Connectedness of $R_k(G)$ on Specific Graph Classes and Other Properties

Choo and MacGillivray [20] explore a very interesting property of the reconfiguration graph of vertex-colourings. They define the *Gray code* of k-colourings of a graph, as the number k_0 such that if $k \geq k_0$, then there is a Hamiltonian cycle in $R_k(G)$. They prove that for every graph there is a Gray code which is smallest possible and also give the Gray code of complete graphs, trees and cycles.

2.4.4 Kempe-Equivalence of Colourings

Given a graph G, a k-colouring of G, and colours c_1 and c_2 (chosen from the k colours), $G(c_1, c_2)$ is the subgraph of G induced by vertices coloured c_1 or c_2. A *Kempe change* is the operation of

المنارة للاستشارات
www.manaraa.com

switching colours c_1 and c_2 on any of the connected components of $G(c_1, c_2)$. Adopting the Kempe change as the reconfiguration rule, then we can ask the standard reconfiguration questions; is there a path between two colourings in $R(G)$ using one Kempe change at a time? Is $R(G)$ connected?

Mohar [63] introduced this variant of reconfiguration before the usual terminology became a standard. In this paper, two specific colourings are called *Kempe-equivalent* if there is a path between them in $R(G)$, and moreover all colourings of the same connected component are called the same.

Most of the work on Kempe-equivalence precedes the work done for the standard reconfiguration version. All 4-colourings of an Eulerian triangulation of the plane [28], all 5-colourings of any planar graph [62], all 5-colourings of any graph containing no K_5 minor [56], and all k-colourings of a planar graph with chromatic number less than k [63] are all Kempe-equivalent.

2.4.5 Reconfiguration on Other Variants of Graph Colouring

Edge Colouring

Ito, Kaminski, and Demaine [46] study the List-edge Colouring Reconfiguration problem where $R(G)$ contains all list edge-colourings of G, given a list of permitted colours for each edge. They show that this problem is PSPACE-complete, even for planar graphs of maximum degree 3 and lists chosen from at most six colours. They also give conditions under which $R(G)$ is connected when G is a tree and an algorithm which finds a path between two list-edge colourings in a quadratic number of steps, which is also best possible.

McDonald, Mohar, and Scheide [60] study similar questions for the Kempe-equivalence version of edge-colouring. They show that $R_k(G)$, where k is the number of colours used in the colourings, is connected when $\Delta(G) \leq 3$ and $k = 4$, and when $\Delta(G) \leq 4$ and $k = \Delta(G) + 2$. Very recently, Belcastro and Haas [3] showed that if G is a 2-connected planar bipartite cubic graph then $R_3(G)$ is connected.

المنارة للاستشارات
www.manaraa.com

$L(2,1)$ Labelling

The $L(2,1)$-Labelling problem can be considered a graph colouring variant, as vertices receive labels instead of colours, and there is the additional restriction that the labels of vertices at distance one have to differ by at least two, and the labels of vertices at distance two have to differ by at least one.

Ito et al. [48] study the list L(2,1)-Labelling Reconfiguration problem, where the reconfiguration rule allows to change the label of exactly one vertex at a time. They show that this problem is PSPACE-complete, even for bipartite planar graphs and $k \geq 6$. They also show that the problem can be solved in linear time for general instances if $k \leq 4$, and that when G is a tree there is a sufficient condition such that $R(G)$ is connected.

المنارة للاستشارات
www.manaraa.com

www.manaraa.com

Chapter 3

Other Reconfiguration Problems

After presenting research done on Graph Colouring Reconfiguration and some variants, which is the most relevant to our work, we now follow results on other reconfiguration problems studying the same questions, P-PATH and P-CONN, regarding the solution space of a combinatorial problem P. We start with Boolean Satisfiability (SAT) Reconfiguration, which together with Graph Colouring was the first work in this context. If we cannot claim that the work on SAT or Graph Colouring initiated the research on reconfiguration problems, we can certainly refer to them as work which precedes a widespread interest in the research community. For example, the term 'reconfiguration' became a standard in one of the papers following the work published on SAT or Graph Colouring – we will cite most of the results in that paper [43] later.

As it is not within the scope of this thesis to survey all the results on reconfigurations or closely related problems, we give preference to results on the reconfigurations of well-known graph theory problems, as our work is within that area. There are a lot of interesting results on token graphs [41], puzzles and games [21], for which it is straightforward to express the P-PATH and P-CONN questions. Towards the end of the chapter we refer to some work which inspired even the very first work on SAT and Graph Colouring and some applications of reconfigurations.

For each problem that we present in relatively more detail, we give necessary definitions, including the statement of the original problem, and then we express the two main decision problems,

www.manaraa.com

P-PATH and P-CONN in terms of the specific problem, by replacing 'P' with a short name representing the specific problem, similarly to 'k-COLOUR' for Graph Colouring with k colours, and stating what is the reconfiguration rule.

3.1 Boolean Satisfiability

Given a Boolean formula ϕ of n variables, which can be evaluated as either 'True' or 'False', then an assignment $(x_1, x_2, ..., x_n)$, where for every $i = 0, 1..., n$, $x_i = True$ or $False$, is satisfying when it evaluates ϕ to 'True'. If the evaluation of ϕ to 'True' is a solution, then the solution graph contains all satisfying assignments. The satisfiability problem SAT, expressed as a decision problem, accepts a Boolean formula ϕ as an input and answers whether a satisfying assignment exists for ϕ or not.

We assume that ϕ has at least two satisfying assignments. Given two satisfying assignments s_1 and s_2, SAT-PATH asks whether there is a path between the two assignments in the solution graph $R(\phi)$ and SAT-CONN asks whether $R(\phi)$ is connected. The reconfiguration rule is the flipping (from 'True' to 'False' and vice versa) of the value of one of the n variables in the ϕ.

We can assume that ϕ is in conjunctive normal form (CNF). Then, the solution graph contains satisfying assignments of k-CNF formulae, where a k-CNF formula is a CNF-formula $C_1 \wedge ... \wedge C_k$, where k is fixed and each clause C_i in the CNF-formula is built using relations from a finite set $\mathcal{S}$. There is an edge between two satisfying assignments, when they differ in the value of exactly one variable.

We express the two reconfiguration questions according to the above definition:

- SAT-PATH
  - Instance: A CNF-formula ϕ, and two of its satisfying assignments s_1 and s_2.
  - Question: Is there a path between s_1 and s_2 in $R(\phi)$?

- SAT-CONN
  - Instance: A CNF-formula ϕ.

المنارة للاستشارات
www.manaraa.com

– Question: Is $R(\phi)$ connected?

3.1.1 Complexity Classifications

Gopalan et al. first in [34] and then in [35] prove that both of these decision problems are PSPACE-complete, but also looked for a dichotomy similar to the one Schaefer [71] showed for the original problem. A set L of logical relations is *Schaefer* if all relations in L are exactly one of bijunctive, Horn, dual-Horn, or affine. Schaefer proves that if the finite set of relations L is Schaefer, then SAT is in P, otherwise it is NP-complete.

Gopalan et al. showed a similar dichotomy for the reconfiguration version of SAT [35]. They define a set of relations L as *tight*, where L properly contains the Schaefer classes. They proved that if L is tight, then SAT-PATH is in P, otherwise it is PSPACE-complete. Also if L is tight, SAT-CONN is in coNP, if L is tight but not Schaefer, it is coNP-complete, otherwise it is PSPACE-complete. They also studied the diameter of $R(G)$; if L is tight, then the diameter of $R(G)$ is linear, otherwise it can be exponential.

In [34], the authors conjectured a trichotomy for SAT-CONN, if Schaefer relations were in P. This was disproved in [58], where a set of specific Horn relations is presented such that SAT-CONN is coNP-complete. In [35], which is the journal version of [34], Gopalan et al. refined their evidence of their conjectured trichotomy. They show that for bijunctive and affine relations, SAT-CONN is in P and they specify new conditions for Horn relations such that SAT-CONN is also in P.

Thus, to complete the complexity trichotomy for SAT-CONN in general, it suffices to establish a dichotomy within Horn relations, i.e. which exactly are in P and which are coNP-complete. Very recently, Schwerdtfeger [72] claims to have completed the trichotomy and specifically to have found an ommision which affects both the trichotomy for SAT-CONN and the SAT-PATH dichotomy. Briefly, he suggests that "Gopalan et al. [35] did not consider repeated occurrences of variables in constraint applications", and shows that these repeated occurrences can make the problems harder and the diameter exponential in cases, while it was thought otherwise. This small

المنارة للاستشارات
www.manaraa.com

shift of the complexity boundaries is represented by defining the set of *safely tight* relations for which both problems are not PSPACE-complete [72], while the tight relations which are not safely tight have moved to the PSPACE-complete side for both decision problems.

In addition, Schwerdtfeger [73] studied two variants of SAT Reconfiguration, considering CNF formulae without constants and partially quantified. Although none of the two versions has a complete classification yet, the author presents specific sets of relations which suggest, similarly to the trichotomy of Gopalan et al., a dichotomy for SAT-PATH and a trichotomy for SAT-CONN for both variants.

3.1.2 Other Results on SAT-CONN

Perhaps, it is worth mentioning that already Ekin et al. [27] had studied connectivity properties of certain Boolean formulae in disjunctive normal form (DNF) and some hardness results are proven. Finally, Makino et al. [59] present an exact algorithm for the answer of the k-SAT-CONN question which runs in time $\mathcal{O}((2-\epsilon_k)^n)$ for some constant $\epsilon_k > 0$, where ϵ_k depends only on k, and not on n.

3.2 On the Complexity of Reconfiguration Problems

The pattern of NP-complete problems giving rise to PSPACE-complete versions, especially for the P-PATH was clearly established with the work of Ito et al. [43] with a journal version appearing later [44], although already the work on SAT and Graph Colouring had suggested that was the case. The authors look at a plethora of NP-complete problems and prove that the complexity of their reconfiguration version is PSPACE-complete, but they also look at problems in P with their reconfiguration version remaining in P. Recall that the latter was also suggested by the result on SAT on tight relations [35].

At this point, it should be clear to the reader how the P-PATH and P-CONN questions can be formulated, given the definition of the original problem and a reconfiguration rule. So, for any

المنارة للاستشارات
www.manaraa.com

results presented, we will only give this necessary information.

We proceed with surveying reconfiguration problems, starting from the work done in [44] and onwards.

3.2.1 Power Supply and Subset Sum

The Power Supply problem is an application of the maximum partitioning problem [45]. Given a bipartite graph G with vertex partitions U and V, then U is the set of *supply* vertices, V is the set of *demand* vertices, $sup(u)$ is the *supply* of u and positive integer, and $dem(v)$ is the *demand* of v, also a positive integer. If a supply forest $T = \bigcup T(u)$, for every $u \in U$, is a partitioning of G into subtrees, where each subtree $T(u)$ contains exactly one vertex u from U and one or more vertices from V, then a supply forest is a solution if the sum of demands of its vertices in V is covered by the supply vertex u.

Thus, the Power Supply reconfiguration graph $R(G)$ is the set of all supply forests which satisfy the demands of all vertices in U and two supply forests are adjacent in $R(G)$, when there is exactly a pair of demand vertices which have swapped their supply vertex. A rather straightforward reduction is given from Boolean Satisfiability Reconfiguration, proving that Power Supply Reconfiguration is PSPACE-complete [44].

We refer the reader to [42] for the work of Ito and Demaine on the Subset Sum Reconfiguration problem.

3.2.2 Shortest Path

Given a graph G and two of its vertices s and t, the Shortest Path problem finds a path between s and t such that the distance between s and t is the smallest possible.

The reconfiguration graph contains all shortest paths between s and t. It is easy to see that the reconfiguration rule can apply a minimal change to a shortest path between s and t by replacing a vertex different from s and t, which results in replacing two edges of the vertex to be replaced with two edges from the newly added vertex such that there is a new shortest path between s and t.

المنارة للاستشارات
www.manaraa.com

Bonsma [8] proved that Shortest Path Reconfiguration (SP-PATH) is PSPACE-complete. For claw-free and chordal graphs it is in P and the diameter of the graph of shortest paths is linear. For the same graphs SP-CONN is also in P. Bonsma proves the PSPACE-completeness of general instances of the problem via a reduction from 4-COLOUR-PATH [13] and he describes polynomial algorithms for claw-free and chordal graphs.

Shortest Path Reconfiguration was first introduced by Kaminski et al. [53] where the existence of paths in the reconfiguration graph of shortest paths of exponential length was shown. This was evidence that either Shortest Path Reconfiguration is PSPACE-complete or that even though remaining in P, the diameter is super-polynomial. The authors also gave a reduction from SAT showing that finding the shortest path between two shortest paths in the reconfiguration graph is NP-hard.

3.2.3 Independent Set

It is not so straightforward to define the solution graph of solutions for an optimisation problem, as a configuration of the input may not be optimal enough to be a solution. That is why, for these problems, there is a threshold given as part of the input, usually a fixed integer.

This is the case with the Independent Set problem: Given a graph G and a positive integer k, is there an independent set of vertices of G of size at least k?

The reconfiguration rule has to take into account the threshold in the input so that the reconfiguration questions for Independent Set, IS-PATH and IS-CONN, define the reconfiguration version of the original problem well. Thus, we need to specify when two independent sets are adjacent in the solution graph, or else what is the exact operation which can change the content of an independent set and produce an adjacent one.

The vertices of the graph of solutions $R(G)$ are the independent sets of G of size at least k, and there is an edge between two independent sets when they differ in exactly one vertex, that is they contain $k-1$ vertices which are exactly same. There is not a unique natural way to define the operation which changes the content of an independent set I to an adjacent I' of G. There have

المنارة للاستشارات

www.manaraa.com

been three different such operations defined using a *token* to mark a vertex, when it belongs to the independent set: *token sliding* (TS), *token jumping* (TJ) and *token addition and removal* (TAR), appearing first in [40], [54], and [44] respectively.

Obviously, a single token can be placed on exactly one vertex, and thus a set of tokens marks an independent set in the graph G. Token Sliding is the 'local' reconfiguration rule, since a token can only slide along an edge of G, producing a new independent set in the solution graph of $TS(G)$, where in Token Jumping a token moves to any other vertex. Thus, two independent sets I and I' adjacent in the solution graph of $TS(G)$ or $TJ(G)$ differ in exactly one vertex such that $I \backslash \{u\} = I' \backslash \{v\}$, where $uv \in G$ for TS only. In TAR, a token can be added to or removed from a vertex as long as the size of the resulting independent set is equal or more than the given threshold.

Token Sliding is PSPACE-complete even for planar graphs of maximum degree 3 [40], where the reduction is to a setting of the *Non-Deterministic Constraint Logic Machine (NCL machine)* presented in the same paper by Hearn and Demaine. The NCL machine proved to be very useful in providing a number of reductions to the PSPACE-complete class as one can see in Hearn's thesis [40] and the resulted publications. Token Addition/Removal is PSPACE-complete, and this can be done by using a reduction from SAT-PATH [44].

All TS, TAR, and TJ remain PSPACE-complete for perfect graphs. This is shown by Kaminski et al. [54] via a reduction from Shortest Path Reconfiguration [8]. More recently, it was shown that IS-PATH is in P for claw-free graphs under the TS and TJ reconfiguration rules [11], and cographs [9] for the TAR rule.

3.2.4 Vertex Cover and Clique

It is briefly stated in [44] that due to the direct relation of Independent Set to Vertex Cover and Clique, Vertex Cover Reconfiguration and Clique Reconfiguration are also PSPACE-complete. The authors also mention that since Set Cover Reconfiguration is a generalisation of Vertex Cover Reconfiguration and Integer Programming of Clique, then these problems are also PSPACE-complete.

المنارة للاستشارات
www.manaraa.com

Very recently, Mouawad et al. [64] showed that Vertex Cover Reconfiguration remains NP-hard for graphs of bounded degree, but it is in P for cactus graphs.

3.2.5 Dominating Set

Given a graph G, the Dominating k-Set problem asks whether there is a dominating set of size k in G. As another optimisation problem for which an accepted solution can be determined on whether it satisfies a certain threshold k, the reconfiguration rule for the Dominating Set problem is not unique and can be defined respectively to the three rules studied for the Independent Set problem. Thus, if we placed tokens on each of the vertices that belong to a dominating set, then we would get the three following case studies:

- *Token Jumping (TJ)*: $R(G)$ contains dominating sets of size k. There is an edge between two dominating sets D_1 and D_2, when they differ in exactly one vertex.
- *Token Sliding (TS)*: $R(G)$ contains dominating sets of size k. There is an edge between two dominating sets D_1 and D_2 of $R(G)$, when they differ in exactly one vertex such that if $u \in D_1 \backslash D_2$ and $v \in D_2 \backslash D_1$, then $uv \in E$.
- *Token Addition-Removal (TAR)*: $R(G)$ contains dominating sets of size at most k. There is an edge between two dominating sets D_1 and D_2 of $R(G)$, when they differ in exactly one vertex.

Using this terminology, it is now easier to refer to the results obtained so far in [29], [36], [74], and [65].

The TJ rule has been considered by Subramanian and Sridharan with $k = \gamma(G)$, where $\gamma(G)$ is the domination number of G, as cited in [36]. Fricke et al. show that $R(G)$ is connected for trees under the TS rule with $k = \gamma(G)$ as well [29].

Haas and Seyffarth [36] study the TAR version of Dominating Set Reconfiguration. Note that according to the TAR rule, we are allowed to add or delete a vertex each time, so for each dominating set $D \in R(G)$, it is $k \geq |D| \geq \gamma(G)$. They show that $R(G)$ is connected for $k = n - 1$, where G has at least two independent edges and n is the number of vertices of G. They also prove

المنارة للاستشارات
www.manaraa.com

that for bipartite and chordal graphs $R(G)$ is connected when $k = \Gamma(G) + 1$, where $\Gamma(G)$ is the maximum cardinality of a minimal dominating set of G.

Very recently, Suzuki, Mouawad, and Nishimura [74] extended the results for the connectivity of $R(G)$ under the TAR rule showing that $R(G)$ is connected for $k = n - m$, when G has at least $m+1$ independent edges. They also give counterexamples and thus giving an answer to a question in [36] on whether $R(G)$ is connected for any graph, when $k = \Gamma(G) + 1$. The examples are planar, multi-partite and of bounded treewidth graphs. Finally, they demonstrate an infinite family of graphs of exponential diameter, when $k = \gamma(G) + 1$, which is the minimum value for k for the TAR model – if $k = \gamma(G)$, then we cannot delete vertices, but only swap, which is possible only under the TS and TJ rules.

3.2.6 Problems Remaining in P

Shortest Path for general instances, 4-Colouring for bipartite and planar graphs and SAT for tight relations are all polynomially solvable, but their reconfiguration version is PSPACE-complete, as seen in Sections 3.2.2, 2.4.2 and 3.1.1, respectively.

Perhaps the result on Shortest Path was the most surprising of all, as it is the only known reconfiguration problem which is PSPACE-complete on general instances while the original version is in P. For example, Minimum Spanning Tree Reconfiguration and Matching Reconfiguration are in P both originally and as a reconfiguration problem [44]. Actually, the authors prove that Matroid Reconfiguration is in P, which generalises the result for Minimum Spanning Tree.

3.3 Parameterized Complexity and Reconfiguration

Lately, there has been an increasing interest to examine the tractability of reconfiguration problems through a different perspective. Since a lot of problems are PSPACE-complete, it seems reasonable to look at their parameterized complexity [24] or how useful it is to approximate their solutions.

المنارة للاستشارات
www.manaraa.com

3.3.1 Classes

Some problems accept an algorithm which requires time polynomial on the input size (e.g. the number of vertices), but can be exponential for some parameter k of the problem. If this parameter can be fixed, i.e. its size does not depend on the input size n of the problem, then the problem belongs in the complexity class FPT (Fixed Parameter Tractable) or we say that the problem is FPT. Other problems remain intractable even when one or more of their parameters are fixed. Those problems belong to the $\mathsf{W}[t], t = 1, 2, \ldots$ complexity classes, with $\mathsf{FPT} = \mathsf{W}[0]$. These classes form the W-hierarchy and they are such that $\mathsf{W}[i] \subseteq \mathsf{W}[j]$, for all $1 \leq i \leq j$. If a problem is $\mathsf{W}[i]$-complete, then there is an FPT-time reduction to other problems which are $\mathsf{W}[i]$-complete. For example, a problem is $\mathsf{W}[1]$-complete if it can reduce to CLIQUE or INDEPENDENT SET in FPT-time and a problem is $\mathsf{W}[2]$-complete if it can reduce to DOMINATING SET using an FPT algorithm. The $\mathsf{W}[i]$ hierarchy can also be formally defined in relation to combinatorial circuits of weft i. For more details and precise definitions the reader should refer to one of the parameterized complexity textbooks available, for example, see [24].

3.3.2 Bounding Solutions and Reconfiguration Sequences

Mouawad et al. [65] first suggested two straightforward parameterizations of reconfiguration problems; to bound the number of solutions k and/or the length ℓ of the reconfiguration sequence between two solutions in $R(G)$. They adapt or extend methods used in the area of parameterized complexity in order to obtain polynomial *reconfiguration kernels* in bounding the number of solutions, and they manage to do this for the Feedback Vertex Set Reconfiguration and Bounded Hitting Set Reconfiguration problems. On the contrary, they show that Unbounded Hitting Set Reconfiguration and Dominating Set Reconfiguration are $\mathsf{W}[2]$-hard, when parameterized by $k + \ell$.

Independent Set, Vertex Cover, Dominating Set, and Graph Colouring Reconfiguration

Mouawad et al. [65] also give a general approach on reconfiguration versions of problems with hereditary properties, classifying them as $\mathsf{W}[1]$-hard, for example Independent Set parameterized

www.manaraa.com

by $k+\ell$ and Vertex Cover parameterized by ℓ. They also show that Dominating Set Reconfiguration parameterised by $k+\ell$ is $\mathsf{W}[2]$-hard, where ℓ is an upper bound on the length of the reconfiguration sequence.

For the latter, there has been more work disseminated very recently, aiming to find restricted instances for which the two problems become fixed-parameter tractable (FPT). Mouawad et al. [64] show that Vertex Cover Reconfiguration remains $\mathsf{W}[1]$-hard for bipartite graphs, which is important in the sense that the original problem is in P for the same class, and FPT for graphs of bounded degree. And finally for the Independent Set Reconfiguration, also very recently, Ito et al. [47] show that the problem under the TJ rule is $\mathsf{W}[1]$-hard, when parameterised by the size of the independent sets, but FPT, when parameterized by both the size of the independent sets and the maximum degree. Even more recently [66] both problems were shown to be FPT for planar graphs.

Finally, Johnson et al. [52] and also Bonsma and Mouawad [12] independently showed that k-Colouring Reconfiguration is FPT for $k \geq 3$, when parameterized by the length of the reconfiguration sequence.

Reconfiguration on Graphs of Bounded Tree-width, Band-width and Tree-depth

Mouawad et al. [66] examine several reconfiguration problems for graphs of bounded tree-width t and they prove that most of them remain PSPACE-complete: e.g. Independent Set, Vertex Cover, Feedback Vertex Set. However, they also show that they are FPT, when parameterized by t. They manage to show this by introducing a technique which defines reconfiguration problems in monadic second order logic.

Wrochna [78] show that k-Colouring, Independent Set, and Shortest Path reconfiguration problems remain PSPACE-complete even for graphs of bounded bandwidth, which restricts instances of the problem more than tree-width and path-width do.

المنارة للاستشارات
www.manaraa.com

3.4 Applications

Frequency Assignment Problems (FAPs) are closely related to Graph Colouring Reconfiguration, a problem in wireless communication networks, where radio frequencies have to be (re)assigned. Perhaps more surprising is an application in the natural realms of the physical world, as in the zero temperature case of the anti-ferromagnetic Potts model, where particles can be seen as vertices and their spins as colourings.

3.4.1 Radio Frequency Assignment

Frequency Assignment Problems and Graph Colouring

In FAPs, there can be different scenarios of varying settings and constraints, each one requiring different parameters or a different model. Aardal, van Hoesel, Koster, Mannino, and Sassano [1] give a survey of the settings of FAPs that may appear in practice, and the models and methods that have appeared in response to the latter. This seems to be the most up to date survey, and it also refers to older surveys of similar content. The same authors maintain a related website with an updated bibliography [55].

Metzger's presentation [61] in 1970 is cited [1] as the first work usually receiving the credit for associating FAPs and graph colouring, and thus optimisation techniques. Since then, radio frequencies have had a fast increasing use, especially after the evolution of the digital cellular phone standard GSM, but also in other fast-developing sectors like the military industry and TV broadcasting, and not too recently wireless internet. One can observe the development of the area and the new problems arising together with new technologies, from the 1980s and Hale's survey [37] until the late 1990s; for example see Eisenblatter's thesis [26] for a discussion on problems and models on GSM networks.

Since the common task in FAPs is to find a balance between the minimisation of the interference between users and the range of frequencies in use, graph colouring methods seem appropriate. Hale [37] refers to models of FAPs in the 1980s, also associating graph colouring as a modelling

المنارة للاستشارات
www.manaraa.com

tool to frequency assignment, introducing some graph colourings variants, mainly t-colourings. For a survey on results specifically on t-colourings, see Roberts [69]. Graph $L(k, h)$-labeling is a generalisation of graph colourings and can also be used in modelling FAPs [76]; for a recent survey on this problem, see [14]. For both graph colouring and labelling techniques with application to FAPs, the reader can refer to [49] and for a survey on a variety of methods and algorithms on FAPs in general to the book of Leese and Hurley [57].

The Colour Graph and Frequency Re-Assignment

When using graph colouring, we usually properly model available frequencies as colours and transmitting points as vertices of a graph, while transmitters that cannot broadcast in the same frequency are connected with an edge in the graph. Additional more realistic constraints produce more complex graph models. For example, such constraints could derive from the decay of radio waves with distance or more sophisticated incompatibilities between transmitters.

The group of FAPs that can be modelled using graph colouring that is most related to the problems on which our research focusses is the one that involves frequency *re*-assignments. This occurs when the constraints designate what is called a Dynamic Channel Assignment (DCA) problem, where the demand of frequencies varies with time, as opposed to Fixed Channel Assignment (FCA) [1]. Resetting the whole network in order to reassign the frequencies from the beginning would not be preferable, as this could mean wide disruption for an unreasonable amount of time. On the contrary, it may be more preferable to disrupt smaller parts of the network for less time each time, until the desired assignment is gradually reached. The research published on frequency reassignment up to now is limited [4, 39], especially when compared to the work on FAPs in general. Graph Colouring Reconfiguration is, apparently, the simplest case of a FAP problem, where transmitters cannot use the same frequency only when they are at most at unit distance from each other.

www.manaraa.com

3.4.2 Relation to Statistical Physics (Glauber Dynamics)

The connectedness of the graph of vertex colourings has been given some attention by statistical physicians when studying the Glauber dynamics of an anti-ferromagnetic Potts model at zero temperature.

Almost uniform sampling enables us to approximately count structures of exponential size in polynomial time [51]. Often the sampling is applied by simulating an appropriate Markov chain. A *rapidly mixing* Markov chain is one that, in simple terms, converges to a very close approximation of the stationary distribution in polynomial time [25]. Such a Markov chain, which is used for sampling k-colourings of a graph, is known as *Glauber dynamics*.

The *Potts model* is a statistical model used to study the mechanics of the particles in a crystalline lattice. Studying the interaction of spins of the particles in this model offers a theoretical basis for describing ferromagnetism and other phenomena related to the physics of solids. In the ferromagnetic case, same spins of neighbouring particles are caused by a form of reduction of the total energy of the system which in its turn is caused by the existence of neighbouring pairs of particles with the same spin. In the anti-ferromagnetic case, neighbouring particles are urged to have different spins. In both cases, the temperature of the system is a measure of the tension of different spins to appear. As the temperature gets lower, the energy of the system is reduced more than the existence of neighbouring pairs of particles with same/different spins. At zero temperature the described causal relation becomes even more evident in the anti-ferromagnetic case.

The zero temperature anti-ferromagnetic k-state(spin) Potts model can be modelled as a k-colouring of a graph G, where the graph is the crystalline lattice (the vertices are the particles) and colours represent the possible spins. Neighbouring particles have different spins under the specific conditions, thus neighbouring vertices have different colours. Thus, the rapidly mixing Glauber dynamics Markov chain of the above model, describes the transition states of the spins of the system. The 'rapidly mixing' part of this model and transition state system is the closest related to our research. One of the conditions for a Glauber dynamics Markov chain to be rapidly mixing, is that the graph model has to be k-mixing. Of course, in this case the graphs are of a

المنارة للاستشارات
www.manaraa.com

very specific class (lattices), and the number of colours is large enough in order to guarantee the k-mixing property (See Theorem 2.2.1).

For some more details on Markov Chains in this context and mixing times of combinatorial objects, see Jerrum's book [51].

www.manaraa.com

www.manaraa.com

Chapter 4

Recolouring Chordal and Chordal Bipartite Graphs

In this chapter, we introduce a class of k-colourable graphs, which we call *k-colour-dense* and we show that the reconfiguration graph $R_\ell(G)$ of vertex colourings of a k-colour-dense G on n vertices is connected, when $\ell \geq k + 1$. We show that this graph class contains the k-colourable chordal graphs and that it contains all chordal bipartite graphs when $k = 2$. Moreover, we prove that for each $k \geq 2$ there is a k-colourable chordal graph G whose reconfiguration graph of the $(k + 1)$-colourings has diameter $\Theta(n^2)$.

Recall that the reconfiguration graph of the k-colourings of a graph G contains as its vertex set the k-colourings of G, and two colourings are joined by an edge in the reconfiguration graph if they differ in colour on just one vertex of G.

Apart from the fundamental problem of characterising the relationship between the complexity of reconfiguration problems and their original version, it is also of interest to find *shortest* paths between solutions. The diameter of the reconfiguration graph provides an upper bound. This is also related to the complexity of finding paths in the reconfiguration graph between given solutions since paths of polynomial length in the reconfiguration graph are certificates for the problem being in NP.

المنارة للاستشارات
www.manaraa.com

For any graph G on n vertices, the diameter of $R_k(G)$, the reconfiguration graph of k-colourings of G, has been shown to be $\mathcal{O}(n^2)$, if $k = 3$ and $R_3(G)$ is connected [17]. Although there are cases where $R_k(G)$ is not connected but contains components of super-polynomial diameter [13], there is no known example of a family of graphs for which $R_k(G)$ is connected but does not have $\mathcal{O}(n^2)$ diameter.

A good place to start when thinking about the above question is to consider graphs of bounded degeneracy. A graph G of degeneracy k is such that for every subgraph $H \subset G$, H has a vertex of degree k. It is well known that graphs of degeneracy k are $(k+1)$-colourable. Bonsma and Cereceda [13] showed that if G is a graph of degeneracy k, then $R_{k+2}(G)$, the reconfiguration graph of $(k+2)$-colourings of G, is connected. In light of what is already known, we are naturally led to ask whether $R_{k+2}(G)$ has quadratic diameter; indeed it is conjectured [13] that $R_{k+2}(G)$ has cubic diameter, although this is modified to quadratic [15]. Our work includes an important class of k-degenerate graphs, namely $(k+1)$-colourable chordal graphs, for which we show the conjecture to be true.

4.1 Preliminaries

In this section we give some basic terminology and notation in addition to what is defined in Section 1.1.2.

The *disjoint union* of two vertex-disjoint graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, which we denote by $G_1 \cup G_2$, is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.

A maximal connected subgraph D of a graph is called a *connected component* (or just *component*) of G; we shall often abuse notation by denoting both the connected component and its vertex set by D. A *separator* of a graph $G = (V, E)$ is a set $S \subset V$ such that $G - S$ has more connected components than G; if two vertices u and v that belong to the same connected component in G are in two different connected components of $G - S$, then we say that S *separates* u and v. We say that we *identify* two vertices u and v if we replace them by a new vertex adjacent to all neighbours of u and v.

المنارة للاستشارات
www.manaraa.com

A *tree* is a connected graph with no cycles. A *clique* is a graph where every pair of vertices is joined by an edge. The size of a largest clique in G is denoted by ω_G. A *perfect* graph is a graph in which $\chi_{G'} = \omega_{G'}$ for every induced subgraph $G' \subseteq G$.

4.2 Sufficient Conditions for Quadratic Diameter

In this section, we introduce the class of *k-colour-dense* graphs, and we show by induction in Theorem 4.2.2 that, for every k-colour-dense graph G, the diameter of $R_\ell(G)$ is at most quadratic in the size of G for all $\ell \geq k+1$. Indeed, the definition of k-colour-dense graphs is recursive and has been formulated in order to facilitate our inductive method. For this reason, it is difficult to establish precisely which graphs are k-colour-dense; however, in the next section, we will show that, for example, k-colourable chordal graphs are k-colour-dense.

For a fixed positive integer k, we say that a k-colourable graph G on n vertices is *k-colour-dense* if either

(i) G is the disjoint union of cliques, each of which has at most k vertices, or

(ii) G has a separator S, and $G - S$ has components D and D' with vertices $u \in D$ and $v \in D'$ such that

  (a) $|D| = 1$ or $|D \cup S| \leq k$,

  (b) $S \subseteq N(v)$, and

  (c) identifying u and v in G results in a k-colour-dense graph G'.

We show the following proposition for use in Section 4.3.1.

Proposition 4.2.1. *If G_1 and G_2 are k-colour-dense graphs, then $G_1 \cup G_2$, the disjoint union of G_1 and G_2, is k-colour-dense.*

Proof. We will show this by induction on the total number of vertices in G_1 and G_2. If G_1 and G_2 are both the disjoint union of cliques, then the claim holds trivially, so assume that G_1 is not the disjoint union of cliques. Thus G_1 has a separator S, components D_1 and D_2, and vertices u and v as in part (ii) of the definition of k-colour-dense graphs; in particular, G_1', the graph obtained from

المنارة للاستشارات

www.manaraa.com

G_1 by identifying u and v, is k-colour dense. Thus, by induction, the disjoint union of $G_1' \cup G_2$ is k-colour dense. Thus S, D, D', u, v also fulfills part (ii)(c) of the definition of k-colour-dense graphs when applied to $G_1 \cup G_2$ (and they obviously still satisfy (ii)(a) and (ii)(b)). □

We define the *ℓ-colour diameter* of a graph G to be the diameter of $R_\ell(G)$.

Theorem 4.2.2. *For an integer $k \geq 1$, let G be a k-colour-dense graph on n vertices. Then, for all $\ell \geq k+1$, the ℓ-colour diameter of G is at most $2n^2$.*

Note that $\ell \geq k+1$ is necessary in the above theorem because, for example, the reconfiguration graph of the k-colourings of a clique on k vertices consists of $k!$ isolated vertices.

Proof. Let $k \geq 1$ be an integer and let G be a k-colour-dense graph on n vertices. We assume $\ell = k+1$; the proof for $\ell > k+1$ is similar. We prove the following claim which immediately implies the theorem.

Claim 1. Let α and β be two $(k+1)$-colourings of G. Then we can transform α to β by recolouring every vertex of G at most $2n$ times.

There are two cases to consider corresponding to the two conditions in the definition of k-colour dense graphs.

We first suppose that G is a disjoint union of cliques and describe how to recolour from α to β. We recolour the disjoint cliques one at a time. Given a clique of G with vertices $v_1, \ldots, v_r$, ($r < k+1$), we consider the vertices in order; once we have $v_1, \ldots, v_{i-1}$ coloured with colours $\beta(v_1), \ldots, \beta(v_{i-1})$ respectively, we try to recolour v_i with $\beta(v_i)$. We are only prevented from doing this directly if there is a vertex v_j with $j > i$ that is presently coloured with $\beta(v_i)$. In this case we first recolour v_j with an unused colour (such a colour exists since $r < k+1$) and then colour v_i with $\beta(v_i)$. When the whole clique is coloured with β each v_j has been recoloured at most $j \leq r \leq 2n$ times.

We now consider the case where G is not a disjoint union of cliques but satisfies condition (ii) of the definition of k-colour dense. We use induction on the number of vertices. Let S, D, D',

www.manaraa.com

$u \in D$ and $v \in D'$ be as in condition (ii). We first show how to transform α into some $(k+1)$-colouring α' satisfying $\alpha'(u) = \alpha'(v)$, by recolouring each vertex of G at most once. Suppose that $\alpha(u) \neq \alpha(v)$. If we can immediately recolour u with $\alpha(v)$, then we do this to obtain the desired colouring α'. If not, then

$$W = \{w \in N_G(u) \mid \alpha(w) = \alpha(v)\} \subseteq N_G(u)$$

must be non-empty. Since $u \in D$, and D is a component of $G - S$, we have $W \subseteq N_G(u) \subseteq D \cup S$. However, every vertex of W is coloured $\alpha(v)$ and no vertex of S is coloured $\alpha(v)$ (since every vertex of S is adjacent to v by condition (ii)(b)), so $W \subseteq D$. Now, for each $w \in W \subseteq D$, we have $N_G(w) \subseteq D \cup S$; thus $|N_G(w)| \leq |D \cup S| \leq k$ by condition (ii)(a) (note that $|D| \neq 1$ since D contains u and the non-empty set $W \subseteq N(u)$). Hence, each vertex of W can be successively recoloured with some colour not used in its neighbourhood. After this we recolour u with $\alpha(v)$ and we do not recolour any other vertices of G. Thus we have recoloured each vertex of G at most once and transformed α to a new $(k+1)$-colouring α' where $\alpha'(u) = \alpha'(v)$. By the same argument, we can transform β to a $(k+1)$-colouring β' with $\beta'(u) = \beta'(v)$. Changing α to α' and β to β' together require that each vertex of G is recoloured at most twice.

We now identify u and v. This leads to a new vertex u' and a graph G' that is k-colour-dense by condition (ii)(c). We can consider α' and β' to be colourings of G' by defining $\alpha'(u') = \alpha'(u) = \alpha'(v)$ and $\beta'(u') = \beta'(u) = \beta'(v)$, respectively. We can transform α' into β' on G' using at most $2(n-1)$ recolourings for each vertex (by application of either the induction hypothesis or the previous case depending on whether G' satisfies the first or second condition of the definition of k-colour dense). Thus we can transform α' into β' on G by simulating each recolouring of u' by a recolouring of u and v in G, that is, every time we recolour u' in G' we apply the same recolouring to u and then immediately to v in G. Thus transforming α' to β' in G requires that each vertex of G is recoloured at most $2(n-1)$ times, and transforming α to α' and β' to β requires at most two additional recolourings of each vertex, resulting in a total of at most $2(n-1)+2 = 2n$ recolourings of each vertex, as required. This completes the proof of the claim and of Theorem 4.2.2. $\square$

المنارة للاستشارات
www.manaraa.com

4.3 Graph Classes

In this section, we show that k-colourable chordal graphs are k-colour-dense for every fixed integer $k \geq 1$ and that chordal bipartite graphs are 2-colour-dense. Hence, these graphs satisfy the necessary condition in Theorem 4.2.2 and consequently have an at most quadratic ℓ-colour diameter, for $\ell \geq k+1$ and $\ell = 3$ respectively.

4.3.1 Chordal Graphs

A *chordal* graph is a graph with no induced cycle of length more than 3. Let $G = (V, E)$ be a graph, let $\mathcal{K}$ be the set of maximal cliques of G, and for $v \in V$, let $\mathcal{K}_v$ be the set of maximal cliques of G containing v. A *clique tree* $\mathcal{T}$ of a (connected) graph G is a tree whose vertex set is $\mathcal{K}$ and whose edges are such that $\mathcal{T}[\mathcal{K}_v]$ is connected (i.e. forms a subtree) for all $v \in V$. In this context, the maximal cliques of G are also called *bags* of $\mathcal{T}$.

The next lemma is well known.

Lemma 4.3.1 ([32])**.** *A connected graph is chordal if and only if it has a (not necessarily unique) clique tree.*

The next lemma is also well known (see e.g. [33]).

Lemma 4.3.2. *If G is a chordal graph then $\omega_G = \chi_G$.*

Next we prove some properties of chordal graphs and clique trees that we shall require. The first property (i) is well known [23], and the second one (ii) has probably been used before, but we give proofs for completeness.

Lemma 4.3.3. *Let G be a connected chordal graph that has a clique tree $\mathcal{T}$, where $\mathcal{T}$ has at least two vertices. Let K be a leaf of $\mathcal{T}$ and let K' be the unique neighbour of K in $\mathcal{T}$. We have the following properties.*

(i) $S := K \cap K'$ is a separator of G, and $D := K \backslash S$ is non-empty and a connected component of $G - S$.

www.manaraa.com

(ii) *There exists* $u \in K \backslash K' = K \backslash S = D$ *and* $v \in K' \backslash K$ *such that, if* G' *is obtained from* G *by identifying* u *and* v*, then* G' *is chordal and* $\omega_{G'} \leq \omega_G$ *(so* $\chi_{G'} \leq \chi_G$ *by Lemma 4.3.2).*

We remark that the above lemma holds more generally even if K is not a leaf of $\mathcal{T}$, but the proof in our case is slightly simpler.

Proof. (i) Fix any $u \in D := K \backslash S = K \backslash K'$; such a vertex exists since otherwise $K \subseteq K'$, contradicting the maximality of K. Fix any $z \in G - K$. Let P be a path of G from u to z with vertices $u = a_0, a_1, \ldots, a_r, a_{r+1} = z$ in order. Let $a_i a_{i+1}$ be the first edge of P not in K. Then $a_i a_{i+1}$ is an edge of some maximal clique $K^* \neq K$. Furthermore $a_i \in K$ since either $a_i = u$ or $a_{i-1} a_i$ is an edge of K. We deduce that $K, K^* \in \mathcal{K}_{a_i}$. Since $\mathcal{T}[\mathcal{K}_{a_i}]$ is connected and the only neighbour of K is K', we have $K' \in \mathcal{K}_{a_i}$. Thus $a_i \in K \cap K' = S$ and so P passes through S. So every path from $u \in K \backslash S$ to any vertex $z \notin K$ passes through S. Hence S is a separator of G, and $K \backslash S =: D$ (which is a clique) is a connected component of $G - S$.

(ii) Fix any $u \in K \backslash S = K \backslash K'$ and $v \in K' \backslash K$; such vertices exist by the maximality of K and K'. Let G' be the graph obtained by identifying u and v, and let u' be the new vertex of G' that results. Suppose for a contradiction that G' is not chordal. Then G' has an induced k-cycle for some $k \geq 4$; this cycle necessarily contains u' since otherwise G would contain an induced k-cycle. Therefore in G' there is a path with vertices $u, b_1, \ldots, b_{k-1}, v$ (in order) such that identifying u and v gives an induced cycle. Thus the path can have no chords except possibly $u b_{k-1}$ or $b_1 v$. However both of those chords would give an induced k-cycle in G, so we can assume that P is an induced path (of length $k \geq 4$). But, since S separates u and v (by part (i) of the lemma), P must pass through S, and since every vertex of $S = K \cap K'$ is adjacent to both u and v, P cannot be an induced path.

Finally, suppose that G' has a $(k+1)$-clique. The clique necessarily contains u'; otherwise it would also be a $(k+1)$-clique of G. Thus in G, there is a k-clique L such that $L \subseteq N(u) \cup N(v)$. Fix vertices $a \in L \backslash N(u)$ and $b \in L \backslash N(v)$ (a, b exist, because otherwise we have a $(k+1)$-clique of G). We know that $S \subseteq N(u) \cap N(v)$, so that $a, b \notin S$. We also know S separates u and v, and yet u, b, a, v is a path from u to v in $G - S$, a contradiction. Hence, G' does not have a

المنارة للاستشارات
www.manaraa.com

$(k+1) - clique.$ □

We use Lemma 4.3.3 in the proof of the following result.

Theorem 4.3.4. *For each fixed integer $k \geq 1$, every k-colourable chordal graph is k-colour-dense.*

Proof. Let $G = (V, E)$ be a k-colourable chordal graph on n vertices. We show by induction on n that G is k-colour-dense. We may assume that G is connected since otherwise, each component of G is k-colour-dense (by induction), and so G is k-colour dense by Proposition 4.2.1. We may also assume that G is not a clique, since then it is trivially k-colour-dense.

By Lemma 4.3.1, G has a clique tree $\mathcal{T}$. Since G is not a clique, G has at least two maximal cliques, so $\mathcal{T}$ has at least two vertices. Let K be a leaf of $\mathcal{T}$, and let K' be the unique neighbour of K. By Lemma 4.3.3, $S := K \cap K'$ is a separator of G, $D := K \backslash S$ is a connected component of $G - S$, and there exist two vertices $u \in D$ and $v \in K' \backslash K \subseteq V \backslash (D \cup S)$ such that identifying u and v gives a graph G' that is chordal and $\chi_{G'} \leq \chi_G \leq k$. Set D' to be the connected component of $G - S$ containing v.

Now, for G, it is easy to check that S, D, D', u, v satisfy conditions (ii) in the definition of k-colour-dense graphs. Condition (ii)(a) is satisfied because $D \cup S = K$ and so $|D \cup S| \leq |K| \leq k$. Condition (ii)(b) is satisfied because $v \in K'$ and $S \subseteq K'$, so that $S \subseteq N(v)$. Condition (ii)(c) is satisfied because identifying u and v in G gives a k-colourable chordal graph G', which is k-colour-dense by the induction hypothesis. □

4.3.2 Chordal Bipartite Graphs

A *chordal bipartite* graph is a bipartite graph with no induced cycle of length more than 4. It is a misnomer since chordal bipartite graphs are only chordal if they are trees. We show that chordal bipartite graphs are 3-colour-dense by proving that a more general class of graphs is 3-colour-dense. Let us call a graph *semi-false* if it can be constructed from a set of one or more isolated vertices by a sequence of the following two operations, namely adding a pendant vertex and adding a semi-false twin. Here, a *pendant* vertex in a graph is a vertex of degree one, and a vertex u is

المنارة للاستشارات
www.manaraa.com

a *semi-false* twin of another vertex v if $N(u) \subseteq N(v)$. Note that adding a pendant vertex u is a special case of adding a semi-false twin, unless u is added as the neighbour of an isolated vertex.

In order to show that every chordal bipartite graph is semi-false we need the following terminology. A vertex u in a bipartite graph G is *weakly simplicial* if its neighbours can be labelled $v_1, \ldots, v_t$ such that $N(v_i) \subseteq N(v_{i+1})$ for $i = 1, \ldots, t-1$. Uehara [75] showed the following, which also follows from results of Hammer et al. [38]; see Pelsmajer et al. [68].

Lemma 4.3.5 ([38, 75])**.** *A bipartite graph G is chordal bipartite if and only if every induced subgraph of G has a weakly simplicial vertex.*

We use Lemma 4.3.5 in the proof of the following theorem.

Theorem 4.3.6. *The class of semi-false graphs is a proper superclass of the class of chordal bipartite graphs.*

Proof. We first give an example of a semi-false graph G^* that is not chordal bipartite. Start with a vertex u_1 and add three pendant vertices u_2, u_3, u_4, each with (unique) neighbour u_1. Then add two semi-false twins u_5 and u_6 of u_1 with neighbours u_2, u_3 and u_3, u_4, respectively. Finally add a semi-false twin u_7 of u_3 with neighbours u_5 and u_6. Because $u_1, u_2, u_4, u_5, u_6, u_7$ induce a 6-vertex cycle in G^*, we find that G^* is not chordal bipartite.

We now show by induction on n that every chordal bipartite graph G on n vertices is semi-false. The case $n = 1$ is trivial. Let $n \geq 2$, let G be a chordal bipartite graph on n vertices, and assume that every chordal bipartite graph with $n - 1$ vertices is semi-false. If we can show that G can be obtained from a semi-false graph G' by adding a pendant vertex or a semi-false twin the theorem will follow. Note that any graph obtained from G by removing a vertex is chordal bipartite and so, by the induction hypothesis, semi-false.

As a graph containing only isolated vertices is semi-false, we assume that G has a component D containing at least 2 vertices. Lemma 4.3.5 tells us that D has a weakly simplicial vertex u, the neighbours of which can be labelled $v_1, \ldots, v_t$, $t \geq 1$, such that $N(v_i) \subseteq N(v_{i+1})$ for $i = 1, \ldots, t-1$.

المنارة للاستشارات
www.manaraa.com

First suppose that $t = 1$. Then let $G' = G - u$. Thus G is obtained from G' by adding u as a pendant vertex.

Now suppose that $t \geq 2$. Then let $G' = G - v_1$. Therefore G is obtained from G' by adding v_1 as a semi-false twin of v_2. □

We note that the class of semi-false graphs does not contain the class of chordal graphs; this can be seen by taking any clique on 3 or more vertices.

We now show that semi-false graphs are bipartite.

Proposition 4.3.7. *Every semi-false graph G is 2-colourable.*

Proof. If G contains only isolated vertices the proposition is true. Otherwise G can be obtained from a graph G' by adding a vertex u that is either pendant or a semi-false twin. Using induction, we can assume that G' has a 2-colouring. We show how to extend it to G by colouring u. If u is pendant, we colour it with the colour that is not used on its unique neighbour. If u is a semi-false twin, then all its neighbours have a common neighbour v. We can therefore colour u with the colour used on v. □

We conclude this section by showing that every semi-false graph G is 2-colour-dense.

Theorem 4.3.8. *Every semi-false graph is 2-colour-dense.*

Proof. We prove by induction on n that if $G = (V, E)$ is a semi-false graph on n vertices then it is 2-colour-dense. The claim is trivially true if $n = 1$.

If G is a semi-false graph on n vertices, then we know by Proposition 4.3.7 that G is 2-colourable. Recall that G is constructed from a set U of isolated vertices by a sequence of pendant-vertex and semi-false-twin operations. Let u be the last vertex added to G either as a pendant vertex or a semi-false twin (if there is no such vertex, then we have $G = (U, \emptyset)$, which is trivially 2-colour dense). If u is a pendant vertex, we may assume that u is an end vertex of an isolated edge $e = uu'$ of G (since otherwise we can consider u to be a semi-false twin of another vertex). Then

المنارة للاستشارات

www.manaraa.com

$G[\{u, u'\}] = (\{u, u'\}, \{e\})$ is 2-colour-dense, $G[V \setminus \{u, u'\}]$ is 2-colour-dense by induction, so G is 2-colour dense by Proposition 4.2.1.

Thus we may assume u is a semi-false twin of some other vertex v of G. We take $S = N(u)$, $D = \{u\}$ and we let D' be the component of $G - S$ containing v. Then S is a separator of G (separating u from v) and $|D| = 1$; hence, condition (ii)(a) in the definition of 2-colour-dense is satisfied. Because $S = N(u) \subseteq N(v)$, condition (ii)(b) is satisfied. Finally, identifying u and v in G to form G' is equivalent to deleting u from G. Thus G' is a semi-false graph (obtained from U by performing the same operations as used for G, except the last). Since G' is 2-colour-dense (by induction) we see that condition (ii)(c) is satisfied. This completes the proof of Theorem 4.3.8. $\square$

4.4 Lower Bounds

We prove that the bound in Theorem 4.2.2 is asymptotically sharp up to a constant factor for every k. To be more precise, for $k = 2$, we show that the 3-colour diameter of a path on n vertices is $\Theta(n^2)$. (Note that a path is chordal bipartite, and as such it is 2-colour-dense due to Theorems 4.3.6 and 4.3.8.) Apart from one subtlety, our result employs very similar techniques to [19], where it is shown that a path on n vertices with an appended triangle has two 3-colourings with quadratic separation. Note however that this example has a disconnected reconfiguration graph and hence infinite diameter.

For each fixed $k \geq 3$ and every $n \geq k$, we give an example of an n-vertex, k-colour-dense graph $G_k(n)$ with $(k+1)$-colour diameter $\Theta(n^2)$. We believe that these are the first examples of graphs with quadratic $k+1$-colour diameter. These examples are easily derived from the path.

Theorem 4.4.1. *The 3-colour diameter of a path on n vertices is $\Theta(n^2)$.*

Proof. We have already seen that the 3-colour diameter of a path on n vertices is at most $2n^2$ by Theorem 4.2.2 and recalled that a path is 2-colour-dense. It remains only to show a quadratic lower bound.

Let P be a path on n vertices $v_1, \ldots, v_n$ for some integer $n \geq 2$. Let the $n-1$ edges of

المنارة للاستشارات
www.manaraa.com

P be $e_1, \dots, e_{n-1}$, where $e_i = v_i v_{i+1}$ for $i = 1, \dots, n-1$. We define edge weights $w(e_i) = \min(i, n-i)$ for $i = 1, \dots, n-1$. Given a 3-colouring c of P and an edge $e_i = v_i v_{i+1}$, we define

$$z_c(e_i) = \begin{cases} 1 & \text{if } (c(v_i), c(v_{i+1})) = (1,2), (2,3), \text{ or } (3,1); \\ -1 & \text{otherwise.} \end{cases}$$

We define the *value* of a 3-colouring c as

$$\phi(c) = \sum_{i=1}^{n-1} w(e_i) z_c(e_i).$$

We claim that $|\phi(c_1) - \phi(c_2)| \leq 2$ for any two 3-colourings c_1 and c_2 of P that are adjacent in the graph R_P^3, i.e., that differ on one vertex of P. This is easy to check, but we give the details for completeness.

Note first that $z(e)$ changes sign if we change the colour of exactly one end vertex of e or if we exchange the colours of e. Let v_k be the (unique) vertex on which c_1 and c_2 differ, and suppose $c_1(v_k) = x$ and $c_2(v_k) = y \neq x$. If z is the unique colour that is not x or y, then the vertices v_{k-1}, v_k, v_{k+1} (when they exist) are coloured z, x, z by c_1 and z, y, z by c_2. From this we deduce that

$$z_{c_1}(e_{k-1}) = -z_{c_2}(e_{k-1}) = -z_{c_1}(e_k) = z_{c_2}(e_k), \tag{4.1}$$

ignoring any terms that are not defined. If $k \neq 1, n$ then

$$\begin{aligned} \phi(c_1) - \phi(c_2) &= \sum_{i=k-1}^{k} w(e_i)(z_{c_1}(e_i) - z_{c_2}(e_i)) \\ &= 2z_{c_1}(e_{k-1})\big(w(e_{k-1}) - w(e_k)\big), \end{aligned}$$

where the last line follows from (4.1). Taking the absolute value of both sides (and noting that $|w(e_{k-1}) - w(e_k)| \leq 1$) proves the claim. If $k = 1, n$, then excluding the appropriate terms from the above calculation (and noting that $w(e_1) = w(e_{n-1}) = 1$) also yields $|\phi(c_1) - \phi(c_2)| \leq 2$.

We now let c_1 be the 3-colouring that colours $v_1, v_2, v_3, v_4, \dots$ by colours $1, 2, 3, 1, \dots$, respectively, and we let c_2 be the 3-colouring that colours $v_1, v_2, v_3, v_4, \dots$ by colours $3, 2, 1, 3, \dots$, respectively. Then

$$\phi(c_1) = -\phi(c_2) = \sum_{i=1}^{n-1} w(e_i) = \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil \geq \frac{1}{4}(n^2 - 1).$$

المنارة للاستشارات

www.manaraa.com

In order to get from c_1 to c_2, the value of the colouring must necessarily change by $|\phi(c_1) - \phi(c_2)| \geq \frac{1}{2}(n^2 - 1)$. Hence, the number of recolourings required is at least $\frac{1}{4}(n^2 - 1) = \Theta(n^2)$ because each recolouring changes the value by at most 2. This completes the proof of Theorem 4.4.1. □

We now generalise Theorem 4.4.1. Recall that every k-colourable chordal graph is k-colour dense by Theorem 4.3.4.

Theorem 4.4.2. *For each fixed $k \geq 2$ and each $n \geq k$, there is a k-colourable chordal (hence k-colour-dense) graph $G_k(n)$ on n vertices that has $(k+1)$-colour diameter $\Theta(n^2)$.*

Proof. The case $k = 2$ follows from Theorem 4.4.1. Assume that $k \geq 3$ and set $n' = n-k+2 \geq 2$. Let $G_k(n)$ be the graph obtained from a path P on n' vertices $v_1, \ldots, v_{n'}$ by adding a clique on $k-2$ new vertices $w_1, \ldots, w_{k-2}$ and inserting an edge between each v_i and each w_j. Because we can obtain $G_k(n)$ by repeatedly adding vertices adjacent to all existing vertices, $G_k(n)$ is chordal. Clearly $G_k(n)$ is k-colourable. We now show that the k-colour diameter of $G_k(n)$ is $\Theta(n'^2) = \Theta(n^2)$.

Let c_1 be a colouring of $G_k(n)$ in which the colours 1, 2 and 3 cycle on the vertices of P. Let c_2 be the colouring closest to c_1 in $R_k(G_k(n))$ in which only 2 colours are used on P. To recolour from c_1 to c_2 only involves recolouring vertices on P since as long as there are 3 colours used on the path, we cannot recolour any vertex not in the path. Moreover only the colours 1, 2 and 3 are available to use on the path. So we can forget about the clique and think only about the distance between the restriction to P of c_1 and c_2 in $R_3(G)$. Using the ideas of the proof of Theorem 4.4.1, we note again that the value of c_1 is $\Theta(n'^2) = \Theta(n^2)$ and see that if P has an even number of edges the value of c_2 is 0 (else consider instead $P - v_1v_2$). As again each recolouring changes the value by at most 2, the proof is complete. □

المنارة للاستشارات

www.manaraa.com

www.manaraa.com

Chapter 5

Recolouring with Extra Colours

In this chapter we give some first results on a decision problem related to the reconfiguration graph of vertex colourings, studied in Chapter 4.

It is of interest to examine how a NO-instance of k-COLOUR PATH can be turned into a YES-instance by relaxing the conditions under which a path exists. That is, given a pair of k-colourings α and β, how many extra colours e are needed such that there is a path from α to β in the reconfiguration graph of $k + e$ colourings?

That is, given a NO-instance of k-COLOUR PATH and now setting $t = k + e$, we pose the following optimisation problem:

- k-EXTRA-COLOUR PATH (optimisation problem)
  - Instance: A graph G and two of its k-colourings α and β, which are in different components of $R_k(G)$.
  - Question: What is the smallest integer $t > k$ such that there is a path between α and β in $R_t(G)$?

We can also express the above question as a decision problem:

- k-EXTRA-COLOUR PATH
  - Instance: A graph G and two of its k-colourings α and β, which are in different com-

المنارة للاستشارات
www.manaraa.com

ponents of $R_k(G)$.

– Question: Given an integer t with $t > k$, is there a path from α to β in $R_t(G)$

We immediately obtain $t \leq col(G)+2$ for any graph G, by Theorem 2.2.1, where $col(G)$ is the colouring number of G. That is, t cannot be larger than the number of colours which guarantees that a graph G is mixing [17]. And since our instances are no-instances of k-COLOUR PATH, then $k < t \leq col(G) + 2$.

Our Results

We first show that we can recolour any k-EXTRA COLOUR PATH instance using $k-1$ extra colours in time that is linear in the number of vertices, and then we describe a property exhibited by some instances which allows them to be recoloured using at most $k-2$ extra colours. Next, we show that there are instances which require $k-1$ extra colours, thus motivating the first result. These instances are constructed based on the cartesian product of a complete graph with itself. Focussing on the cartesian product of the triangle (9 vertices), we show how adding a vertex and adding/removing edges produces instances which may require either 1 or 2 extra colours. Finally, we examine instances of simple 3-chromatic graph classes, which can be recoloured using 1 extra colour.

5.1 Preliminaries

Here we give or recall some necessary definitions related to the colour graph and the related reconfiguration problems. For any definition not included in this chapter, we refer the reader to Sections 1.1.2, 2.1 and 4.1.

Definition 5.1.1. Given an instance of k-EXTRA-COLOUR PATH (G, α, β), we define the *colour sets* $V^{\beta}_{i,j}$, $1 \leq i \leq 2k-1, 1 \leq j \leq k$ of G *in relation to the target colouring* β, and such that $V(G) = \bigcup\limits_{i,j} V^{\beta}_{i,j}$. We also define the *colour classes* V_i of G, where V_i is the union of colour sets $V_{i,j}$, for all j. The assignment of the vertices to the colour sets is equivalent to colouring the vertices

المنارة للاستشارات
www.manaraa.com

with some colour within the range $\{1, 2, ..., 2k-1\}$.

Thus, if a vertex u is coloured with colour i, then $u \in V^{\beta}_{i,j} \subset V_i$, where $j = \beta(u)$. Also, given two different colourings of G, there is at least one vertex which belongs to different colour sets (resp. classes) in the two colourings. When the $2k-1$-colouring of G is proper, then every colour class and colour set are independents sets. We call the colour class V_i *initial*, when $V^{\beta}_i = \{u \mid \alpha(u) = i\}$ and *target* or T^{β}_j, when $V^{\beta}_j = T^{\beta}_j = \{u \mid \beta(u) = j\}$. Recolouring a vertex u from colour i to colour i' is equivalent to *moving* u from colour set $V^{\beta}_{i,j}$ (resp. colour class V^{β}_i) to colour set $V^{\beta}_{i',j}$ (resp. colour class $V^{\beta}_{i'}$). Obviously, *recolouring a colour set* $V^{\beta}_{i,j}$ with colour $i' \neq i$, is equivalent to moving all the vertices of $V^{\beta}_{i,j}$ to colour set $V^{\beta}_{i',j}$.

A pair of colour sets $V^{\beta}_{i,j}$ and $V^{\beta}_{i',j'}$, $i \neq i', j \neq j'$ is *disconnected*, when $V^{\beta}_{i,j} \cup V^{\beta}_{i',j'}$ is an independent set. The two colour sets are 'disconnected' in that they could potentially be 'connected' by having an edge between any vertices of the two colour sets, since this would not violate the colouring constraint. Yet, they are 'disconnected', that is there are no edges between any pairs of such vertices.

For what follows, if the target colouring β is clear from the context, then we denote colour set $V^{\beta}_{i,j}$ as $V_{i,j}$ and colour class V^{β}_i as V_i.

5.2 Recolouring in k-EXTRA-COLOUR PATH

Recall that t is the smallest integer such that $t > k$ and colourings α and β are connected in $R_t(G)$. Thus, the least number of extra colours required such that α and β are connected in $R_t(G)$ is $e(G, \alpha, \beta) = t - k$.

In this section we first describe how to recolour any instance $G(\alpha, \beta)$ of k-EXTRA-COLOUR PATH quickly with $k-1$ extra colours. Then in Section 5.2.3, we show that there are instances such that $e(G, \alpha, \beta) = k - 1$. This implies that $k-1$ is not an arbitrarily chosen number of extra colours guaranteeing the recolouring of every instance, but the lower bound of all such numbers. In addition, we show that the trivial lower bound of $2n$ recolourings remains the same when we use

المنارة للاستشارات
www.manaraa.com

$k-1$ extra colours when using the algorithm described in Theorem 5.2.1. The lower bound can be achieved when we use an extra colour for each vertex and then recolour to the target colouring. More specifically, with an unlimited number of available colours, we can recolour all vertices to colours different than the initial k colours in at most n steps, and then recolour each vertex to its target colour in at most n steps.

In between these results and in Section 5.2.2, we show that $k-2$ extra colours are enough to find a path between any two k-colourings for instances with a pair of disconnected colour sets.

5.2.1 Recolouring General Instances with $k-1$ Extra Colours in $\mathcal{O}(n)$ time

We now give a simple polynomial algorithm to find a path between two k-colourings α and β in G, showing that $k-1$ extra colours are enough to recolour any instance (G, α, β).

Proposition 5.2.1. *Given a k-colourable graph G on n vertices and two of its k-colourings α and β, there is always a path of length $\mathcal{O}(n)$ between α and β using $k-1$ extra colours.*

Proof. In each round i, $i = 2, ..., k$, we recolour target colour class T_i with colour $k+i-1$. Then, in k rounds we recolour each target colour class T_i to its target colour i.

It is easy to see that in the first part of the algorithm all colourings are proper, since we recolour vertices of the same target colour class using an extra colour; different extra colours for different colour classes. In the second part, we can only start recolouring to β from target class T_1, as any other class $T_i, i \neq 1$ has neighbours in T_1, which is coloured with colours from α, and thus with colours appearing in β. After we recolour T_1 with colour $\beta(T_1) = 1$, then any other vertex can move to its colour target class, as its neighbours are coloured with either their target colour in β or a colour not appearing in β. Since each vertex is recoloured at most twice, at most once in the first round and once in the second round, the overall time is at most $2n$. □

المنارة للاستشارات
www.manaraa.com

5.2.2 Instances with a Pair of Disconnected Colour Sets

The following theorem provides an upper bound for $e_k(G, \alpha, \beta)$ in k-EXTRA COLOUR PATH, for instances with at least one pair of disconnected colour sets.

Theorem 5.2.2. *Let (G, α, β) be an instance of k-EXTRA COLOUR PATH. If (G, α, β) has a disconnected pair of colour sets, then $e_k(G, \alpha, \beta) \leq k - 2$.*

Proof. Suppose that there is a disconnected pair of colour sets, defined in relation to the target colouring β. Let these colour sets be V_{i_1,t_1} and V_{i_2,t_2} with $i_1 < i_2$, and thus by definition $i_1 \neq i_2, j_1 \neq j_2$. We apply a procedure similar to the one using $k-1$ extra colours in Proposition 5.2.1, but now using $k-2$ extra colours, in order to find a path from α to β.

The basic idea is to recolour all target colour classes $T_j, j = 1, 2, ..., k$ with extra colours apart from the two which contain the disconnected pair of colour sets, V_{i_1,t_1} and V_{i_2,t_2}. Then, recolour both V_{i_1,t_1} and V_{i_2,t_2} with the same colour. After, there is always a target colour class with an available colour, so we recolour target colour classes to their target colour either directly or using available colours until we reach the target colouring β.

- Recolour each of the target colour classes T_j apart from t_1 and t_2 to a new extra colour.
- Since i_1 appears only in colour sets V_{i_1,t_1}, V_{i_1,t_2}, which are disconnected to colour set V_{i_2,t_2}, we recolour V_{i_2,t_2} with i_1.
- Now, i_2 appears only on colour set V_{i_2,t_1}. Recolour target class t_1 with i_2.
- Now, t_2 appears only in target colour class t_2 so we can recolour the latter to t_2.
- Finally, t_1 does not appear anywhere, so we can recolour the respective target colour class.

The recolourings are such that colour sets which are connected never receive the same colour, so all intermediate colourings resulting from colour set or colour class recolouring are proper. □

When there is no pair of disconnected colour sets, then there are instances (G, α, β) for which $e_k(G, \alpha, \beta) = k-1$, but also instances that require fewer than $k-1$ extra colours. For example, in the next section we give an example of an instance (G, α, β) with no pair of disconnected colour sets and yet $e_k(G, \alpha, \beta) = k - 1$. However, if we consider a 3-EXTRA COLOUR PATH instance

المنارة للاستشارات
www.manaraa.com

(G, α, β), where each colour set contains two vertices and between any two target colour classes there are edges such that there is exactly a perfect matching between vertices of each pair of target colour classes, then we get a forest of paths but no disconnected pair of colour sets. For this instance, $e_3(G, \alpha, \beta) < 2 = k - 1$.

5.2.3 Instances with $e_k(G, \alpha, \beta) = k - 1$

In this section we prove that the upper bound for $e_k(G, \alpha, \beta)$ provided by Proposition 5.2.1 is tight. While that proposition shows that $k - 1$ extra colours are enough for any instance, this does not mean that such a number of colours is necessary. However, we give below an example of an instance such that $e_3(G, \alpha, \beta) = 2$, that is $e_k(G, \alpha, \beta) = k - 1$ for $k = 3$.

First, we need to recall the following definition.

The *cartesian product* $G \times H$ of two graphs G and H has vertex set $V(G \times H) = \{u_v : u \in V(G), v \in V(H)\}$ and edge set $E(G \times H) = \{u_v u'_{v'} : uu' \in E(G)$ and $v = v'$, or $u = u'$ and $vv' \in E(H)\}$.

Informally, the cartesian product of two graphs G and H, on n and m vertices respectively, is a graph which is comprised of m copies of G with some edges added between vertices in different copies of G. If we assign each of the m copies of G to a different vertex in H, then given two copies G' and G'' of G, they are assigned to two distinct vertices v' and v'' of H. If $v'v'' \in H$, then we add an edge between vertices $u'_i \in G'$ and $u''_i \in G''$, where $1 \le i \le n$ is a labelling of the vertices of each copy of G.

Now, we are ready to define a graph $Z_{k,k}$ and prove that there are instances $(Z_{k,k}, \alpha, \beta)$ with $e_k(Z_{k,k}, \alpha, \beta) = k - 1$. We will also use $Z_{3,3}$ to define finite graphs 10_{XY} with instances such that $e_3(10_{XY}, \alpha, \beta)$ can be both $k - 2$ and $k - 1$.

Definition 5.2.3. $Z_{k,k}$ is the cartesian product of the complete graph on k vertices, $K_k \times K_k$.

المنارة للاستشارات
www.manaraa.com

Labelling $Z_{k,k}$

For ease of reference we will use a specific construction (drawing) of $K_k \times K_k$ for when we refer to it as $Z_{k,k}$. Recalling the definition of a colour set $V_{i,j}$ (Definition 5.1.1), we can call (i,j) *the colour set coordinates*.

To aid with the construction of $Z_{k,k}$, we look for an instance $(K_k \times K_k, \alpha, \beta)$ such that every colour set $V_{i,j}$, $i,j \leq k$ is non-empty when $K_k \times K_k$ is coloured with α, that is it contains at least one vertex. $K_k \times K_k$ has exactly k^2 vertices, as many as the colour sets with $i,j \leq k$. Thus, exactly one vertex of $K_k \times K_k$ should be in each colour set. Since $K_k \times K_k$ is a disjoint union of K_k copies, we can obtain α and β by ordering the k-colours on each different copy of K_k such that every vertex falls into a different colour set. We consider one of the possible disjoint unions of copies of K_k of the graph and we number the copies from 1 to k. We colour each i copy of K_k with two colours: j and $j+i$. Now observe that each vertex is in a different colour set. We label each vertex using its colour set coordinates, such that vertex $u_{i,j}$ is in colour set $V_{i,j}$.

Note that this labelling does not mean that $Z_{k,k}$ depends on specific colourings, although it is obvious that its labelling corresponds to the specific instance from which it was assigned. If necessary, we will mention when we refer to the label of a vertex in $Z_{k,k}$ or its two colours in an instance $(Z_{k,k}, \alpha, \beta)$.

Similarly, we can use colour set coordinates to label a graph G in the same way that we did for $Z_{k,k}$ and again mention, if not clear, whether the labelling corresponds to actual (two) colourings of an instance of the problem.

Constructing 10_{XY}

We will construct a graph 10_{XY}, where X and Y are colour set coordinates and $X \neq Y$. Consider graph $Z_{3,3}$ and let $X = (i,j)$. We add a new vertex $v_{i,j}$ to $V(Z_{3,3})$. We want both $u_{i,j}$ and $v_{i,j}$ to have three neighbours, not all common. We remove edge $u_{i,j}u_{i',j'}, i \neq i', j \neq j'$, and we add $v_{i,j}u_{i',j'}$. We also add two more edges between $v_{i,j}$ and neighbours of $u_{i,j}$. Now there is one neighbour of $u_{i,j}$ with coordinate Y, which is not a neighbour of $v_{i,j}$.

www.manaraa.com

According to this construction, there are four possible coordinates for X and Y:

- $A = (i + 1, j - 1)$
- $B = (i + 1, j + 1)$
- $C = (i - 1, j - 1)$
- $D = (i - 1, j + 1)$, where addition is modulo 3.

Given an instance (G, α, β) of k-EXTRA-COLOUR-PATH with G being either $Z_{3,3}$ or 10_{XY}, we can assume without loss of generality that the coordinate of a vertex corresponds to its initial and target colours α and β – in this exact order. Such colourings exist, since two of them are used to construct the labellings of those graphs. For 10_{XY} in particular, depending on the two input colourings, the values of X and Y, thus producing any of the possible six combinations of coordinates.

Given a graph G and two of its k-colourings α and β, let $M(G)$ be the set of all maximal independent sets of G. The following proposition specifies the size and contents of $H \in M(G)$, where G is either $Z_{3,3}$ or 10_{XY}.

Proposition 5.2.4. *If $G = Z_{3,3}$ or $G = 10_{XY}$, then for every $H \in M(G)$, we have $3 \leq |H| \leq 4$ and specifically exactly one of the following is true:*

- *$|H| = 4$, when $H = V_{i,j}$*
- *$|H| = 3$, when $H = V_r$, $V_{i,j} \nsubseteq V_r$*
- *$|H| = 3$, when $G = 10_{XY}$ and $H = \{u, v, w : u \in V_{i,j}, v \in V_{rc} \subset X \cup Y, w \in V_{rc'} \cup V_{r'c}\}$, where V_r is a colour class and $V_{i,j}$ is the colour set which contains two vertices in 10_{XY}.*

Proof. By definition of 10_{XY}, if $V_{i,j} \subseteq H$, then $|H| = 4$. If $H = V_r$ and $V_r \neq V_{i,j}$, then $|H| = 3$. In both cases there is no other vertex that we can add to H, because all vertices in $G \setminus H$ have at least one neighbour in H.

Assume that $G = 10_{XY}$ and $H \nsubseteq V_r$, for any colour class V_r. Then, not all vertices in H are from the same colour class. Suppose that $u, v \in H$, such that u and v are from different colour classes. Then, they also belong to connected colour sets. Since $uv \notin E(G)$, then it must be that

المنارة للاستشارات
www.manaraa.com

$u \in V_{ij}$ and $v \in V_X$ or $v \in V_Y$. Now, observe that if there is a third vertex $w \in H$, then it must share the same colour class with one of u, v, otherwise it would be connected with one of them and thus H would not be an independent set. So, if $v \in V_{rc}$, then $w \in V_{rc'}$ for some $c' \neq c$ or $w \in V_{r'c}$ for some $r' \neq r$. With a similar argument it is easy to see that H is maximal, as there is no other vertex in $G \setminus H$ that is not a neighbour of one of the three existing vertices. □

Proposition 5.2.5. *Let G be either $Z_{3,3}$ or $G = 10_{XY}$, and α and β two of its 3-colourings. If for every $H \in M(G)$, $G \setminus H$ contains a fixed 6-cycle C^*, then there is no path between α and β.*

Proof. Let $H \in M(G)$. Then, $G \setminus H$ contains a fixed 6-cycle. We reach a 4-colouring γ, where $\gamma(H) = 4$ and $\gamma(G \setminus H) = \alpha(G \setminus H)$. Since H is maximal and α is a frozen colouring of G, there is no way to recolour vertices of the fixed cycle, unless we recolour at least one vertex from H back to its initial colour.

We will prove that whatever is the content of H, according to Lemma 5.2.4 above, attempting to find a different path always involves the recolouring of a maximal independent set.

Suppose we recolour one or more vertices of H back to their initial colour in colouring α. Then, we reach a colouring γ', where $\gamma(H') = 4$ and $\gamma'(G \setminus H') = \alpha(G \setminus H')$, $H' \subset H$. If $|H'| = 3$, then $|H| = 4$. By Lemma 5.2.4, there is only one vertex to recolour with an extra colour, and that is the unique vertex in $H \setminus H'$, so we would go back to colouring γ. Thus, we assume that $|H| = 3$. Then, $1 \leq |H'| \leq 2$. Since any recolouring would start with the recolouring of one vertex with an extra colour, it suffices to look at the case when $|H'| = 2$.

Case 1: Suppose that H is a colour class.

Then H' has both of its vertices on the same colour class. Obviously $H \cap C^* = \emptyset$, so C^* remains fixed, unless we recolour one more vertex x, extending H'. If x is in the same colour class with both of these vertices, then we end up with the original H. If x is in a different colour class with y, one of the two vertices of H', then without loss of generality $x \in V_{ij}$ and $y \in V_X$. By Lemma 5.2.4, $H'' = H' \cup \{x\}$ is maximal, and thus there is a cycle in $G \setminus H''$ which is fixed.

Case 2: H is not a colour class.

المنارة للاستشارات

www.manaraa.com

Then by Lemma 5.2.4, it contains two vertices which belong to different colour classes. This means that H has the form of the set H'' in the first case above. If we follow the construction of H'' in reverse order by setting $H = H''$, it easy to see that we need to reach to a different maximal independent set to recolour a vertex of C^*.

Thus, there is no path starting from colouring α and reaching a colouring γ which does not result in a fixed cycle in G. □

Corollary 5.2.6. $e_3(Z_{3,3}) = 2$.

Proof. Every maximal independent set $H \in Z_{3,3}$ is either an initial or target colour class and for every such set H, $G \backslash H$ contains a fixed 6-cycle. By Proposition 5.2.5, if $(Z_{3,3}, \alpha, \beta)$ is an instance of k-EXTRA COLOUR PATH, then there is no path between α and β. Thus, by Proposition 5.2.1, any instance of $Z_{3,3}$ would require two extra colours, $e(Z_{3,3}) = 2$. □

Recall the definition of 10_{XY} and the possible four states (coordinates) of each of X and Y. Also, we denote $V_{r,s}$ as the colour set containing two vertices. Now, we can prove the following proposition, which gives a necessary and sufficient condition such that $e(10_{XY}, \alpha, \beta) = 1$.

Proposition 5.2.7. $e_3(10_{XY}) = 1$*, unless* $XY = AD$ *or* BC*.*

Proof. Consider the case where XY is neither AD nor BC. For these instances, there is a colour class H which does not contain $V_{r,s}$ and none of V_X or V_Y. Thus, $G \setminus H$ is a path of six vertices. As all the neighbours of the vertices of this path are coloured with the extra colour, we can recolour $G \setminus H$ as if we would recolour a graph which is a path. It is easy to do this using no extra colour on the path itself, by starting with a vertex which one of the three original colours available. Then, we recolour H to β_H.

(For example, if XY is AB and $(r, s) = (2, 3)$, then $V_{r,s}$ is V_{23}, exactly one vertex in V_{23} is connected to $u_{31} \in V_{31}$ and exactly one vertex in V_{23} is connected to $u_{32} \in V_{32}$. Thus, $H = I_1$, the initial colour class, with $\alpha(I_1) = 1$).

Now consider when XY is either AD or BC. Observe that for every maximal independent set

المنارة للاستشارات
www.manaraa.com

H, $G \setminus H$ contains a fixed 6-cycle. By Proposition 5.2.5, there is no path using one extra colour. By Proposition 5.2.1, $e_3(10_{XY}, \alpha, \beta) = 2$. □

A consequence of Proposition 5.2.7 is that we have found examples of instances (G, α, β) which do not have a disconnected pair of colour sets and yet $e_3(G, \alpha, \beta) = 1$. Thus the opposite of Theorem 5.2.2 above is not true. That is, there are examples of instances with no disconnected pair of colour sets, and yet $e_k(G, \alpha, \beta) = k - 2$.

Theorem 5.2.8. *For any two k-colourings α, β of a graph G, $e(G, \alpha, \beta) \leq k - 1$ is tight.*

Proof. Proposition 5.2.1 immediately implies that a path is guaranteed using $k-1$ colours. Finally, Corollary 5.2.6 provides an example where $k - 1$ extra colours are necessary. □

Corollary 5.2.9. *For any graph G of less than k^2 vertices and two k-colourings α and β of G, $e_k(G, \alpha, \beta) < k - 1$.*

Proof. Consider graph $Z_{k,k}$. Since α and β are k-colourings of G, then there are exactly k^2 non-empty colour sets, and thus $Z_{k,k}$ has k^2 vertices. Suppose that there is a graph G with fewer vertices than $Z_{k,k}$, and for which $k - 2$ extra colours are not enough in order to transform α colouring to β. Then by Theorem 5.2.2, there is no pair of disconnected colour sets and thus there is no empty colour set – otherwise there would be more than one pair of disconnected colour sets. This implies that each colour set has at least one vertex. By the pigeonhole principle, G has at least k^2 vertices; a contradiction. □

5.3 3-EXTRA-COLOUR PATH on Some Graph Classes

In this section we attempt to explore the k-extra-COLOUR PATH problem by looking at the smallest value of k for which the problem is not trivial, that is $k = 3$. Note that applying Theorem 5.2.8 for $k = 3$ we get that $e_3(G, \alpha, \beta) \leq 2$ for any graph G and any two of its 3-colourings. Thus in the case of two colourings which are in different components of the $R_3(G)$, $e(G, \alpha, \beta)$ is either 1 or

المنارة للاستشارات
www.manaraa.com

2. Consequently, the optimisation problem of computing $e_3(G, \alpha, \beta)$ is equivalent to the decision problem of whether $e_3(G, \alpha, \beta) = 1$.

- 3-EXTRA COLOUR PATH
  - Instance: A graph G and two of its k-colourings α and β, which are in different components of $R_3(G)$.
  - Question: Is there a path between α and β in $R_4(G)$?

An equivalent question is obviously: Is $e_3(G, \alpha, \beta) = 1$?

In exploring the computational hardness of the above question, we present some classes of instances for which $e_3(G, \alpha, \beta) = 1$. In these cases, computing $e_3(G, \alpha, \beta)$ takes as long as recognising that the input graph G belongs to a specific graph class or that the instance (G, α, β) has some specific property.

By Theorem 5.2.2, $e_k(G, \alpha, \beta) = 1$, when there is a disconnected pair of colour sets. This recognition requires $\mathcal{O}(n^2)$ time.

5.3.1 Bipartite Graphs

Let G be a bipartite graph and any two of its 3-colourings.

Proposition 5.3.1. $e_3(G, \alpha, \beta) = 1$, *when* G *is bipartite.*

Proof. Let $V = A \cup B$ be the partition of G. We can find A and B by taking a walk on the graph starting from any vertex and then include all vertices of parity 1 in A and of parity 2 in B. To recolour G, we first recolour A with colour 4. Since every vertex in B has neighbours only in A, then vertices in B can be recoloured as in β. Since every vertex v in A has neighbours coloured as in β, then $\beta(v)$ is available to v, and so we recolour vertices in B as in β. $\square$

5.3.2 Some 3-Chromatic Graphs

In this section, G is a 3-chromatic graph and α and β are two of its 3-colourings.

المنارة للاستشارات

www.manaraa.com

Definition 5.3.2. Let $L_{m,r} = P_m \times P_r$ be a lattice with m rows and r columns of vertices. Let $u_{i,j}$ be the vertex in row i and column j. Then, $E(L_{m,r}) = \{u_{i,j}u_{i+1,j}, u_{i,j}u_{i,j+1}, 1 \leq i \leq m-1, 1 \leq j \leq r\}$.

Also, let $L^t_{m,r} = C_m \times C_r$. Then $L^t_{m,r}$ is as $L_{m,r}$ above with some edges added; $u_{i1}u_{ir}$, $u_{1j}u_{mj}$, for all $i, j \geq 1$.

Theorem 5.3.3. *Let G be a 3-chromatic graph. For the following cases, $e_3(G, \alpha, \beta) = 1$:*

(a) G is a cycle

(b) G is a cycle with a chord

(c) G is a theta graph

(d) $G = L_{m,r}$

(e) $G = L^t_{m,r}$

Proof. (a). Let G be an odd cycle with n vertices, and two of its 3-colourings, α and β.

Pick a vertex v. Since $G \setminus v$ is bipartite, we can partition it into two independent sets A and B, using the parity property of bipartite graphs. Observe that A cannot contain both neighbours of v, as they have different parity. Let u be the neighbour of v which is not in A. We recolour A to 4. We can recolour all vertices in $B \setminus \{u, v\}$ to their target colour, as all of their neighbours are in A. Since u and v are incident to vertices coloured 4, then u and v have an available colour. We can recolour u and v to their target colour, using the available colour to one of them, if needed. Finally, we can recolour vertices in A to their target colour.

(b). Let G be a cycle C with a chord, and two of its 3-colourings α and β. Let x,y be the vertices that induce the chord. Let $P_r \equiv x_1...x_r$ and $P_s \equiv y_1...y_s$ be the two distinct paths in G between x and y, with $r, s \geq 1$.

As in (a), we recolour an independent set A to colour 4. We set A to contain all vertices with an odd index from paths P_r and P_s. In this case $B = G \setminus A$ contains vertices of the two paths with even parity plus edge uv. We recolour A to 4. We can now recolour all vertices on the two paths with even parity to their colour in β, except if their neighbour is either x or y. Then, apart from vertices in A, the only vertices which are not set to their target colour induce a path P of length

المنارة للاستشارات
www.manaraa.com

at most 4, inclusive of x and y. The endvertices of P have exactly one neighbour coloured 4. If the path has length four, then we add one of the two internal vertices to A by recolouring it to 4, and then recolour the vertex which is between two vertices in A to its target colour. Now the only vertices in B not in their target colour are the vertices of an edge in path P. Since any neighbours of u and v are in A, then each of u and v has an available colour. If none of u and v can be set to their target colour immediately, we can use an available colour first.

(c). Let G be a theta graph, C be the largest cycle of G, P the path between vertices v_i and v_j of C, excluding those vertices, $P = G \setminus C$. Let α and β two 3-colourings of G.

We assume that P has length more than one (or else G is a graph with a chord). We follow exactly the same procedure as in (b) with the only difference that A is determined excluding all vertices in path $P(x, y)$ – and not only vertices x and y. After A is coloured to 4, the remaining path P in $B = G \setminus A$ has length three or more. With a further inclusion of vertices to set A, as done in (b), the only vertices in B not set to their target colour induce an edge, but they also have an available colour. Using the available colour, if needed, we can set those vertices to their target colour. Finally we set the vertices of A to their target colour.

(d). Let $L_{m,r}$ be a lattice graph with m vertices on each row and r vertices on each column. We choose an independent set A which we colour with 4 and then attempt to recolour the vertices $B = G \setminus A$ to their target colour. Let $u_{i,j}$ be the vertex in row i and column j. We set $A = \{u_{i,j}, u_{i+1,j+1}$, where $i, j \geq 1$ and odd $\}$. Observe that A and B contain only isolated vertices. Thus, we can recolour A to 4, recolour the vertices in B to their target colour and then the vertices in A also to their target colour.

(e). Let $L^t_{m,r}$ be a lattice and furthermore add edges $u_{i1}u_{ir}$, $u_{1j}u_{mj}$, for all $i, j \geq 1$. Considering the same set A as defined in (d) above, A is not an independent set in G because some of the added edges have both of their vertices in A. We remove the vertex with the highest row or column index from A so that A is an independent set again. Since G contains at least one odd cycle, then one of the dimensions is odd. In that case, if we recolour as in (d) above, then B contains isolated vertices and an even cycle. It is not hard to exchange vertices between A and B such that B is a path plus isolated vertices (ie avoid having a cycle in B). And since we can recolour a path with no

www.manaraa.com

fixed endvertices without using an extra colour, we can recolour vertices in B to their target colour, and then the same for A. Thus, $e(G, \alpha, \beta) = 1$, as we only used one extra colour to recolour A in the meantime. $\square$

المنارة للاستشارات
www.manaraa.com

www.manaraa.com

Chapter 6

Reconfiguration of Hamiltonian Cycles in Graphs of Bounded Degree

6.1 Introduction

In this chapter we look at reconfigurations of the *HAMILTONIAN CYCLE* problem:

Let G denote a simple undirected graph. A hamiltonian cycle of G is a cycle that contains every vertex of G. Consider the following problem.

- Hamiltonian Cycle
  - Instance: A graph G.
  - Question: Does G contain a hamiltonian cycle?

For an instance G of the *HAMILTONIAN CYCLE* problem, we define the *reconfiguration graph* $H(G)$: the vertices correspond to its solutions (that is, each vertex of $H(G)$ is a hamiltonian cycle of G) and a pair of hamiltonian cycles C_1 and C_2 is joined by an edge in $H(G)$ if there are vertices t, u, v and w in G such that $C_2 = C_1 \setminus \{tu, vw\} \cup \{tv, uw\}$. We call the operation of obtaining C_2 from C_1, a *switch* (of u and v). One way to think about the difference between C_1 and C_2 is that after we remove edges tu and vw from C_1, we are left with two disjoint paths which are $t...w$ and

المنارة للاستشارات
www.manaraa.com

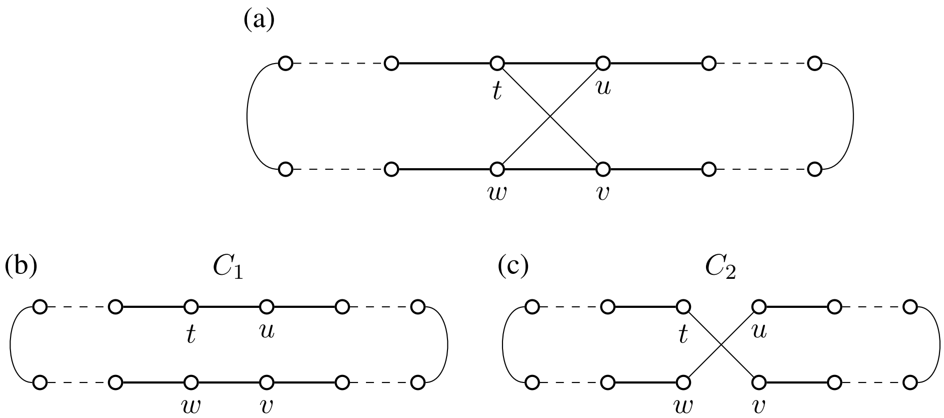

Figure 6.1: A switch on vertices t, u, v, and w. In (b), cycle C_1 with edges tu and wv is adjacent to the cycle C_2 with edges tv and wu in (c).

$u...v$, without loss of generality. Then, we use edges tv and uw to join the two paths and create C_2. Such a switch is shown in Figure 6.1 in (a), and where (b) and (c) are the two states of the switch in cycle C_1 and C_2, respectively.

We are concerned with the problem of whether there is a sequence of hamiltonian cycles $C^i = C_0, C_1, \ldots, C_k = C^t$ such that pairs that are adjacent in the sequence are adjacent in $H(G)$ and so each can be obtained from the other by switching a pair of vertices. We call this a *switching* sequence.

Minimality of the Switch

In general, reconfiguration rules of combinatorial problems are minimal in the sense that there is minimal difference between two adjacent solutions of the solution graph.

The switch of $H(G)$, as defined earlier, is a minimal reconfiguration rule because it removes the minimal number of edges, two edges from the initial cycle C_1, and adds two new edges to create the new cycle C_2 adjacent to C_1 in $H(G)$.

However, for the rest of the chapter we add extra 'minimality' to the switch of $H(G)$ by minimising the distance in C_1 between the two edges to be removed from C_1. It is easy to see that the minimum distance between these two edges cannot be zero. That is, the two edges cannot share

المنارة للاستشارات
www.manaraa.com

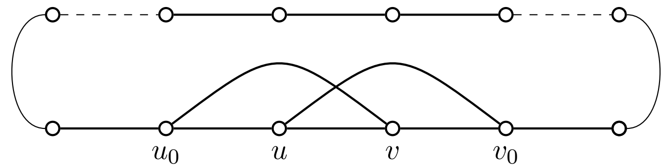

Figure 6.2: A switch on vertices u_0, u, v, and v_0 as it is defined specifically for the HC-PATH problem, where the vertices of the switch appear in consecutive order on both of the two adjacent cycles C_1 and C_2, with u and v swapping positions in C_2.

a vertex, otherwise it is not enough to add two new edges to obtain the new cycle C_2. Thus, the minimum distance in C_1 between the removed edges is one.

From now on, we refer to a *switch* according to the following refined definition, with an illustration found in Figure 6.2.

Definition 6.1.1. Let u_0, u, v, and v_0 be vertices in G, and C_1 and C_2 a pair of hamiltonian cycles of G, where u_0uvv_0 is a path on cycle C_1, u_0vuv_0 is a path on cycle C_2, and $C_2 = C_1 \setminus \{u_0u, vv_0\} \cup \{u_0v, uv_0\}$. Then, the *switch of uv* is the operation of obtaining C_2 from C_1. When considering cycle C_1, we call uv, u, and v *switching* and u_0 and v_0 *supporting vertices* of uv.

For ease of reference to the application of this reconfiguration rule, we will use equivalent expressions, as appropriately. Thus, to obtain C_2 from C_1 is to *switch (edge) uv* or to *switch (vertices) u and v*. Alternatively, we can say that *two vertices u and v switch* or *an edge uv switches*.

Our Results

Hamiltonian Cycle is a well-known NP-complete problem, which remains NP-complete even for cubic graphs [31]. On the contrary, we prove that Hamiltonian Cycle Reconfiguration, defined below as HC-PATH and equipped with the reconfiguration rule as defined in Definition 6.1.1, can be decided in linear time for graphs of maximum degree 5.

Thus we can now define our reconfiguration problem.

المنارة للاستشارات
www.manaraa.com

- HC-PATH
  - Instance: A graph G and two of its hamiltonian cycles C^i and C^t.
  - Question: Is there a reconfiguration path between C^i and C^t?

In the next section we introduce some basic definitions and lemmas useful to further study HC-PATH both for graphs of bounded degree and in general.

6.1.1 Definitions

Orientation, Paths and Arcs of a Hamiltonian Cycle C

Throughout this chapter and when discussing the HC-PATH problem for graphs of degree at most k, a hamiltonian cycle C of a graph G is given a left-to-right orientation by means of listing its vertices consecutively, starting with any vertex x and ending in the same vertex x, where any two consecutive vertices in the ordering are adjacent in C.

Let C be a hamiltonian cycle of a graph G.

- A *path* $P(u, v)$ on cycle C with *endvertices* u and v is the subgraph of C induced by u, v and the vertices between them. All vertices except u and v, are called the *internal* vertices of path $P(u, v)$. When all internal vertices are mentioned, then a path is also denoted by the sequence of the vertices in the order of appearance. For example, $uwxyzv$ is a path $P(u, v)$.
- The *distance* $d(u, v)$ between u and v on C is the number of edges of the shortest path between u and v on C.
- An *arc* uv of length k is an edge in G with $k = d(u, v) > 2$ on C.

Alignment of Edges in Relation to a Pair of Cycles

Given two hamiltonian cycles C and C^t of a graph G, we can partition the edges of G into two sets A and M, according to the ordering of their vertices in C and C^t.

- An edge uv in G is *aligned* (in relation to C and C^t), when the vertices of uv appear in the

www.manaraa.com

same order both in C and C^t. A is the set of all aligned edges in C.

- An edge uv is *misaligned* (in relation to C and C^t), when uv is not aligned. M is the set of all misaligned edges.

Note: Since the target cycle remains unchanged, from now on we define aligned and misaligned edges referring to the current cycle C only, as the target cycle C^t is implied. And when C is also implied, then we refer to aligned and misaligned edges without mentioning C.

When deciding whether an edge uv is aligned or misaligned, we assume that n is much larger than k, and thus the shortest path between u and v along each of the two cycles defines the orientation of uv in each cycle – and for example, different orientations suggest that uv is misaligned.

An edge in C is *ready* when it is in a switch, otherwise it is *unready*. That is an edge uv is ready if it is possible to immediately switch u and v and obtain a new hamiltonian cycle in which their order is reversed.

We can further partition misaligned edges:

- U is the set of all misaligned and unready edges.
- R is the set of all misaligned and ready edges.

For ease of reference and as we will be using sets U and R more often, we refer to edges in U as just 'unready' and edges in R as just 'ready'. On the other hand, for edges which are aligned and unready or aligned and ready there will be an explicit reference.

We further partition the set U, as shown in Figure 6.3. Let u_0uvv_0 be a switch and $uv \in U$:

- if $uv_0 \notin E$, then $uv \in U^+$
- if $u_0v \notin E$, then $uv \in U^-$
- if $u_0v, uv_0 \notin E$, then $uv \in U_0$

Similarly, we define $A^-, A^+, A_0 \subseteq A$. Moreover, if uv is aligned and ready, as shown in (c) of Figure 6.3, then $uv \in \bar{A} \subseteq A \cap C$.

المنارة للاستشارات
www.manaraa.com

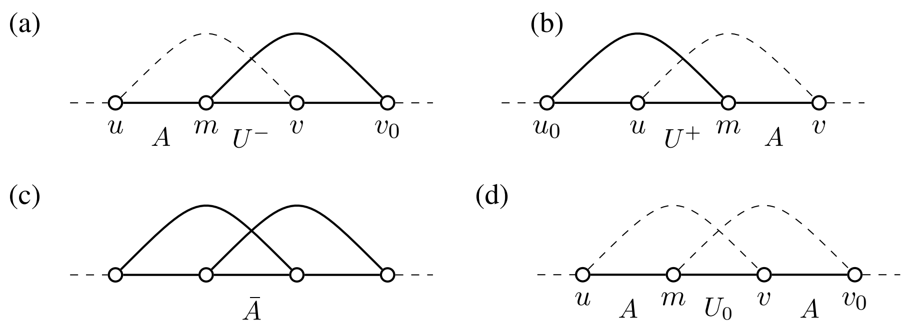

Figure 6.3: (a) A d-arc $u(m)v$ with an unready edge $mv \in U^-$. (b) A d-arc $u(m)v$ with an unready edge $mv \in U^-$. (c) An aligned and ready edge in $\bar{A}$. (d) A U_0 edge.

6.1.2 Deriving the Alignment of an Edge

We define the *alignment* of an edge uv, or two vertices u and v, as their state as aligned or misaligned in a cycle C (in relation to the target cycle). Also, we say that we *align* (resp. *misalign*) an edge uv (or two vertices u and v) in relation to C, when we switch uv and uv is misaligned (resp. aligned) in C.

Observation 6.1.2. *Let u and v be two vertices in a graph G, and let C^i and C^t be two of its hamiltonian cycles which are connected in $H(G)$. If $uv \in M$, then $uv \in E$, and if $uv \notin E$, then $uv \in A$.*

Proof. If $uv \in M$ and $uv \notin E$, then uv cannot switch, and thus two cycles are not connected; a contradiction. Therefore, $uv \in E$. Also, the contrapositive is true, that when $uv \notin E$, then $uv \in A$. □

Given a hamiltonian cycle C and two vertices u and v in a graph G, the degree of u and v and their relative position in C provide requirements such that u and v can be adjacent in some cycle C' in $H(G)$.

Lemma 6.1.3. *Let a, b and c be three vertices in a graph G, and let C^i and C^t be two of its hamiltonian cycles which are connected in $H(G)$. If a is the leftmost and c the rightmost vertex of the three in C, then the following are true:*

المنارة للاستشارات
www.manaraa.com

- *(i) If both ab, $bc \in M$, then $ac \in M$.*
- *(ii) If both ab, $bc \in A$, then $ac \in A$.*

Proof. (i) If there is a path between C and C', assume that bc switches first, without loss of generality. Then, it remains that ab must switch. For this to happen, ac has to switch first. Now, all vertices appear in reverse order compared to cycle C. Observe that there is no way to switch ac back again, without misaligning the rest of the edges. Hence, it must be that $ac \in M$. (ii) Suppose that $ac \in M$. Then to align ac requires to switch bc. But, then $bc \in M$. □

Corollary 6.1.4. *Let a, b and c be two vertices in a graph G, and let C^i and C^t be two of its hamiltonian cycles which are connected in $H(G)$. If a is the leftmost and c the rightmost vertex of the three in C. Then if the alignment of ac is different from one of ab and bc, then it is the same as the other.*

Observation 6.1.5. *Let u and v be two vertices not adjacent in a hamiltonian cycle C of a graph G. Then every internal vertex of $P(u, v)$ is connected to at least one of u, v.*

Proof. For uv to switch, every internal vertex in $P(u, v)$ has to first switch with either u or v. □

6.2 Maximum Degree 4

In this section, we consider HC-PATH on the first non-trivial case of bounded degree graphs, i.e. when the maximum degree is $\Delta(G) = 4$. We will prove that HC-PATH is in P for this restricted class of graphs. Although this result is a special case of the result on graphs with maximum degree 5, found in Section 6.3, we think it is worth highlighting as it is much simpler.

The problem HC-PATH for graphs of maximum degree 4 is defined as follows:

- HC-PATH with $\Delta = 4$
  - Instance: A graph G of maximum degree 4 and two of its hamiltonian cycles C^i and C^t.

المنارة للاستشارات

www.manaraa.com

– Question: Is there a reconfiguration path between C^i and C^t?

We will prove the following.

Theorem 6.2.1. *HC-PATH with $\Delta = 4$ can be decided in linear time.*

The proof is based on a number of lemmas. Before we proceed with the lemmas we provide some intuition on how the problem is decided in linear time. The algorithm is described by procedure *Pr*, stated in the proof of the Theorem later. We briefly state the procedure:

Starting from cycle C^i we switch ready edges successively on every newly obtained cycle until there are no ready edges left.

The lemmas to follow help us show that such a procedure decides the problem correctly. That is, if there are no ready edges left to switch, then either there is no path between the two cycles or cycle C^t has been reached.

Definition 6.2.2. A *supporter* s of an edge uv is a common neighbour of u and v which is either the left-neighbour of u or the right-neighbour of v, when $uv \in R$ in some cycle C'. We also say that s *supports* uv in C'.

Lemma 6.2.3. *Let C^i and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C^i and C^t are connected in $H(G)$, then each vertex in C^i is misaligned with at most two vertices.*

Proof. If a vertex is misaligned with three vertices to the right and aligned to any other vertex, then it has five distinct neighbours; its initial left neighbour, the three misaligned vertices and its final right-neighbour. If a vertex u is misaligned with two vertices to the right and vertex v to the left, then v is misaligned with three vertices to its right. □

In the next lemma we prove that we can choose any ready edge in C to switch. This derives from the fact that it is not necessary for any vertex in a ready edge to support any other switch in subsequent cycles, and also switching the ready edge does not harm the generality of finding a path.

المنارة للاستشارات
www.manaraa.com

Lemma 6.2.4. *Let C and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C and C^t are connected in $H(G)$, and, moreover, it is possible to reach C^t from C by only switching ready edges, then the order in which ready edges switch does not matter.*

Proof. Consider a ready edge uv. We will prove that, without loss of generality, vertex u is not necessary to support any other switch at distance 1 or 2 (beyond that distance, maximum degree 4 rules out such a case).

Case. Distance 2

Suppose the path $uvwx$ is on the current cycle C, and that vertex v supports $wx \in M$. We will prove that wx does not require v to support it. If $ux \notin E$, then before both uv and wx switch, there has to be some vertex y between u and x.

– Suppose y is on the right of x. Observe that all vertices mentioned have degree 4 (also vertex x, as it has to have two supporting edges for its switch with y). The right neighbour of w is always a vertex in $N(w) = \{u, v, x, y\}$. Suppose we first switch uv. Then after we can either switch y first until it reaches u, or wx. Both cases require that the right neighbour of w is not in $N(w)$. Suppose we do not switch uv first, then y will not be able to switch with w, as $yv \notin E$. At any case, one of the vertices mentioned has to have degree at least 5 in order to switch both misaligned edges, and thus ux must be an edge. This also implies that u, instead of v, can support wx after uv switches. It remains to prove that vertex u is does not need to support any other switch, before it switches with v. Vertex u has maximum degree and can only support edge vw in the next step, if needed. Edge vw is aligned, otherwise u would have more than two misaligned neighbours, contradicting Lemma 6.2.3.

–Suppose y is on the left of u. Similarly, we can prove that at least one of the vertices involved has to have degree more than four. Therefore, vertices in ready edges do not have to support any switch at distance two.

Case: Distance 1

In this case, there are two consecutive misaligned edges in C on a path $auxwb$. Suppose $ux, xw \in R$. Then the structure induced by the vertices in G is unique, when there is a path from

المنارة للاستشارات
www.manaraa.com

C to C^t. Apart from the degree imposed by the ready edges, also vertex u and w have to have vertices b and a, respectively, as final neighbours. We illustrate this for b. Vertices u, x, and w have all four neighbours, including their final. If the final neighbour f of u is not b, then f would have to reach w before u does. But the only neighbour of w on its right is b, and so $f = b$. Notice that both $auwx$ and $uwxb$ have to be 4-cliques and that following any order of switching ready edges on these vertices, leads to their target position in C^t, as appearing in the path $axwub$ on C^t.

Suppose $uw \in R$ and $wx \in U$ in path $auxwb$. Can we switch uw first, without obstructing the path to the target cycle? Since b cannot support wx, there has to be some other vertex y, supporting wx on its right. But, $N(w) = \{w', u, x, b\}$, where x' is the final left-neighbour of w, on the left u. Therefore, y has to be on the left of w. Observe that y must be u, and thus ux must switch before xw. □

Lemma 6.2.5. *Let C and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C and C^t are connected in $H(G)$, then C^t can be reached from C without switching any aligned edges.*

Proof. Suppose $uv \in A$ in C. Also, suppose $uv \in \bar{A}$, i.e. in a switch, and that one of its vertices can support another edge at a valid distance C. The proof of Lemma 6.2.4 implies that if uv is ready, and thus in a switch, then none of its vertices is needed to support another edge, before uv possibly switches. Thus, also uv, being aligned but also ready, does not need to support any vertex and can remain aligned. □

It follows from Lemma 6.2.3, that we cannot have three consecutive edges in U. In fact, we can say something stronger:

Lemma 6.2.6. *Let C and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C and C^t are connected in $H(G)$, then there cannot be a pair of adjacent edges in C which are both in U.*

Proof. Suppose cycle C contains the path $axyzb$, where $xy, yz \in U$. As $xz \in M$, then $xz \in E$, see Observation 6.1.2. As $xy, yz \in U$, then $ay, yb \notin E$. It is $N(x) = \{a, z, y, x'\}$, where x' is

المنارة للاستشارات
www.manaraa.com

the final right-neighbour of x. As x and y must have two supporting vertices in order to switch, yx' must be an edge, and $x' \neq b$. Similarly, the final left-neighbour of z cannot be a. But then this implies that x and z have only one neighbour in common, and thus xz cannot switch. □

Lemma 6.2.7. *Let C and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C and C^t are connected in $H(G)$, then for a given edge uv in C which is in U, no vertex to the right of v is misaligned to v (and no vertex to the left of u is misaligned to u).*

Proof. The two parts are symmetric, so we just prove the first. Let w be the right-neighbour of v. By Lemma 6.2.6 $vw \in A$. If any vertex to the right of w is misaligned to v, then it is also misaligned with u and w, because $uv \in U$ and $vw \in A$. This contradicts Lemma 6.2.3. □

Lemma 6.2.8. *Let C and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C and C^t are connected in $H(G)$, then for a given edge $uv \in U$ in C, we can assume that vertices u and v have a common neighbour initially to the right of v and a common neighbour initially to the left of u and these vertices are at distance at most two from the nearest of the pair.*

Proof. We assume that u and v have exactly two neighbours in common since they must switch (observe that if they have three neighbours in common, then $uv \notin U$). We can assume that a common neighbour s of u and v is initially on the right of v; if s is at distance more than 2 from v then it is not adjacent to the right-neighbour of v, so s could only support a switch if this right-neighbour switched left, in which case it also neighbours u and so there is a neighbour at distance at most 2. If the vertex that supports the switch of u and v from the left is a vertex initially to the left of u we are done.

Otherwise a vertex x from the right of v must switch with u. Vertex x and all of its neighbours are of maximum degree 4, therefore the neighbours of x can be specified; $N(x) = \{u, v, u_0, x_0\}$, where u_0 is the initial left-neighbour of u, and x_0 is the initial right-neighbour of x. This implies that xv is initially in C. Observe that once uv switches with the support of vertex x on the left and, say, vertex y on the right, the edge xv is now in U. This is because uv is initially in U, and by Lemma 6.2.6, vx is in A. Therefore since vx switches once, it has to switch back in the next

www.manaraa.com

steps. The common neighbours of v and x are exactly two and these are u, supporting from the left, and x_0, both currently on the right of xv. Thus $uv \in A$ if and only if $vx \in M$. Therefore the two cycles are not connected in $H(G)$, a contradiction. □

For the following two lemmas, we assume that set $R = \emptyset$, i.e. there are no ready edges in the current cycle C. Also, recall the assumption that every misaligned edge is involved in a switch.

Lemma 6.2.9. *Let C^* and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If C^i and C^t are connected in $H(G)$, and there are no ready edges to switch, then there cannot be a pair of adjacent edges in A, unless every edge is in A.*

Proof. Let cycle C^* contain $fabcxyz$ where xy and yz are in A and cx is in U. By Lemma 6.2.6, bc is in A. As cx is in U, either cy or bx is not an edge.

Case 1. Suppose $cy \notin E$.

By Lemma 6.2.7, the final right-neighbour of c is initially to its right. By Lemma 6.2.8, c and x are both neighbours of z. But as z is aligned to the right of y, z cannot be the final right-neighbour of c and if c is misaligned with z, then it is misaligned with x,y and z and we apply Lemma 6.2.3. Thus c,x,y are all to the left of z and are all aligned to its left so its initial right-neighbour is its final right-neighbour. So the final right-neighbour of c is at least two to the right of z and misaligned with at least three vertices.

Case 2. Suppose $bx \notin E$.

If $ab \in A$, this is the same as the first case. By Lemma 6.2.8, a and b have a common neighbour to the right within distance 2. As $bx \notin E$, both then it must be $ac, bc \in E$. Similarly c and x have a neighbour in common to the left which must be a. So, $N(a) = \{f, b, c, x\}$. But as $ab \in U$, then $fb \notin E$. But a and b must have a neighbour in common to their left. If this is not f, then a has a fifth neighbour. □

We have proved that adjacent edges cannot belong to the same set. So the whole cycle C^* has edges alternating between A and U. We show this is a contradiction.

المنارة للاستشارات
www.manaraa.com

Lemma 6.2.10. *Let C^* be the hamiltonian cycle and C^t be two hamiltonian cycles of a graph G of maximum degree 4. If there is a path between the two cycles, and there are no ready edges to switch, then it cannot be the case that all edges of C^* alternate between sets A and U.*

Proof. Suppose cycle C^* contains path $abcdefg$. If some edge in the cycle is not aligned, then we can assume we have $ab, cd, ef \in A$ and $bc, de \in U$. Without loss of generality, we assume that $bd \notin E$. So, by Lemma 6.2.8, $be, ce \in E$. Thus $df \notin E$, as $de \in U$. By Lemma 6.2.8 again, $dg, eg \in E$, so then $N(e) = \{b, c, d, f, g\}$; a contradiction. □

Proof of Theorem 6.2.1:

We consider what happens when the following procedure is applied to C^i:

Procedure Pr:

- While R is nonempty, switch $uv \in R$ in the current cycle C, where uv is an edge in the current cycle C.

Claim 6.2.11. *Procedure Pr terminates in at most linear time.*

Proof. Lemmas 6.2.4 and 6.2.5 imply that switching any available ready edge leads to an adjacent cycle which is on some path leading to C^t in the solution graph, and no edge is required to be switched more than once. Therefore, procedure Pr terminates in at most $|E|$ steps, where E is the set of edges of graph G. □

Then, we suppose we have the case that there is a path from C^i to C^t, and that procedure Pr has been applied to C^i. Let C^* be the output of the procedure. The next claim 6.2.12 proves that $C^* = C_2$.

Claim 6.2.12. *When procedure Pr terminates, if there is a path from C^i to C^t, then the cycle C^* obtained by the procedure is C^t.*

Proof. After Pr terminates, there are no ready edges left in cycle C^*. This means all edges in C^* are either in set A or U. By Lemmas 6.2.6, 6.2.9 and 6.2.10, there are no two adjacent edges both

المنارة للاستشارات
www.manaraa.com

in either A or U, and it cannot be that the membership of adjacent edges of C^* alternates between A and U. Thus, the only remaining case is that every edge is in A, which means that $C^* = C^t$. The only case left is that Pr is not sufficient to find C^t from C^i since aligned vertices must be switched. Lemma 6.2.5 disproves this case, and thus the procedure decides the problem correctly. □

6.3 Maximum Degree 5

We consider graphs of maximum degree 5, as the first case where the procedure in Theorem 6.2.1 cannot decide the problem correctly. For example, given two cycles C^i and C^t, suppose that five consecutive edges in cycle C^i are labelled as A, R, R, U, A. It is left to the reader to observe that the order of switching ready edges in R in the next steps, while trying to obtain the target setting, *does* matter. Therefore, applying the procedure Pr of Theorem 6.2.1 will not decide whether there is a path to C^t correctly.

For the case of graphs of maximum degree 5, we provide an algorithm which applies local manipulations whenever we reach a cycle with no ready edges, in order to reach a new cycle, again with no ready edges, but with fewer misaligned edges overall. The algorithm decides whether there is a path by considering the order of switching edges (both in R and $\bar{A}$). It does so by processing cycle paths of constant length, as imposed by the degree constraints of the graph (for example, see Observation 6.1.5).

Section 6.3 is comprised of three main parts. First, we give some necessary definitions (Section 6.3.1), then we describe the algorithm (Sections 6.3.2 and 6.3.3), and finally we prove its correctness (Sections 6.3.4 to 6.3.6).

6.3.1 Definitions

What follows is a list of definitions used throughout Chapter 6. Let G be a graph and C one of its hamiltonian cycles. $\bar{G}$ is the complement of G.

المنارة للاستشارات
www.manaraa.com

Arcs and Disconnected Arcs

Let V and $\bar{V}$ be the set of vertices of G and $\bar{G}$, respectively, where $\bar{G}$ is the complement graph of G.

Recall the definition of an arc uv in Section 6.1.1. We define the *vertices and edges of an arc* uv as all the vertices and edges, respectively, of path $P(u, v)$ on C. The endvertices of the arc uv are the endvertices of $P(u, v)$. We categorise a pair of arcs according to the distance or relative location on the cycle. We say that two arcs:

- *cross*, when exactly one endvertex of one arc is an internal vertex of the path of the other arc.
- are *disjoint*, when they do not share any endvertices.
- are *consecutive*, when they share an endvertex.
- are *at distance* k in C, when they are disjoint, do not cross, and the shortest path between two of their endvertices is k.

An arc is called *disconnected*, if uv is in $\bar{E}$.

Definition 6.3.1. Let C^t be the target cycle of an instance of HC-PATH. If umv is a path on C, then arc uv (of length 2) is called a *d-arc* with *middle vertex* m. We will denote the d-arc by uv, or when referring to its middle vertex explicitly, by $u(m)v$. The vertices in C^t between u and v are called *final middle vertices* of d-arc uv. If d-arc uv has only one final middle vertex, it is called a *single-middle* d-arc; otherwise it is called *multi-middle*.

Note that at most one of the two edges of a d-arc can be unready (Corollary 6.1.4). Whether the d-arc contains an unready edge or not, it will be clear from the context.

Sequences and Supporters

Recall that a switching sequence is a sequence of hamiltonian cycles, which are pairwise adjacent in $H(G)$. For every pair of adjacent cycles, there is a corresponding switch. This fact leads in to the following definition:

Definition 6.3.2. A *sequence* Q is a sequence of switches between pairs of cycles forming a switch-

المنارة للاستشارات
www.manaraa.com

ing sequence. A *subsequence* $Q' \subset Q$ is a sequence of switches of which pairs of cycles do not necessarily form a switching sequence.

It follows by the definition that a subsequence is not necessarily a path in $H(G)$, but a set of paths in $H(G)$ (which could form the path induced by Q in $H(G)$ if zero or more edges are added).

A *possible supporter* s of an unready edge mv in cycle C is a common neighbour of m and v in G and satisfies Observation 6.1.5. A vertex x *replaces* the middle vertex m in d-arc $u(m)v$ in cycle C', when there is a sequence from a cycle C which contains path umv to a cycle C' which contains path uxv.

Definition 6.3.3. A possible supporter s of a d-arc $u(m)v$ in cycle C is *vicious* to uv, if for every sequence Q_s from C during which s replaces m in uv, exactly one of the two is true:

– uv remains a d-arc with a middle vertex $x \neq m$, or

– m replaces s in $u(s)v$.

In other words, a possible supporter s is vicious to a d-arc $u(m)v$, when there is no sequence of switches which can align mv with s as a supporting vertex.

Below, we refine the definition of a supporter, as described in Section 6.2.

Definition 6.3.4. A *supporter* s of an edge mv of a d-arc $u(m)v$ is a possible supporter of mv which is not vicious to $u(m)v$. If s will be the left supporting vertex on the left (resp. right) of mv, then s is called a *left supporter* (resp. right supporter) of mv.

Settings and d-arc Settings

Let S_P be the induced subgraph induced by a path P of a cycle C in G minus all arcs of length more than two.

Definition 6.3.5. A *setting* S on a path P of a cycle C in G is such that $S_P \subseteq S \subseteq G[P]$, where the alignment of the edges in P is known and G[P] is the induced subgraph of P in G.

للاستشارات المنارة
www.manaraa.com

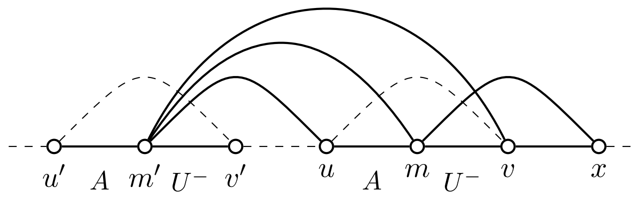

Figure 6.4: A d-arc setting is a setting which contains a pair of related d-arcs. The middle vertex m' of d-arc $u'v'$ on the current cycle is the final middle vertex of d-arc uv in the target cycle.

A setting is described by assigning labels to the edges of a path on C, depending on which of the sets A, U, R the edge belongs. Given the setting of a path $x_1x_2...x_d$ of length d, we let $l_1l_2...l_{d-1}$ be labels where l_i is the label for the edge x_ix_{i+1} and is: 'A', 'R', or 'U', for x_ix_{i+1} being aligned, ready, or unready, respectively.

Two d-arcs $u'(m')v'$ and $u(m)v$ are *related*, when m' is a final middle vertex of $u(m)v$, or m is a final middle vertex of $u'v'$.

A setting is *aligned*, when all of its edges are aligned. The sequence of switches which aligns one or more of the misaligned edges of a d-arc setting is called *aligning*.

Definition 6.3.6. A *d-arc setting* is a setting with its path containing exactly two related d-arcs.

An example of a (generic) d-arc setting with both unready edges in U^- is shown in Figure 6.4. Below, we categorise d-arc settings according to the relative position of their two d-arcs on the cycle, also shown in Figure 6.5. Let $u(m)v$ be a d-arc with middle vertex m.

– If mm' is a d-arc with middle vertex either u or v then the setting on path $m'umv$ (or $umvm'$) a *d-crossing*.

– If $u'(m')v'$ is a d-arc related to $u(m)v$, m' is a final middle vertex of $u(m)v$ and the least distance in C between uv and $u'v'$ is k, then we distinguish two cases. When m is the final middle vertex of $u'(m')v'$, then the setting on path $P(u',v)$ is a *k-exchange*, otherwise it is a *k-sub*. Note that $k = 0$ is possible and then $v = u'$.

المنارة للاستشارات
www.manaraa.com

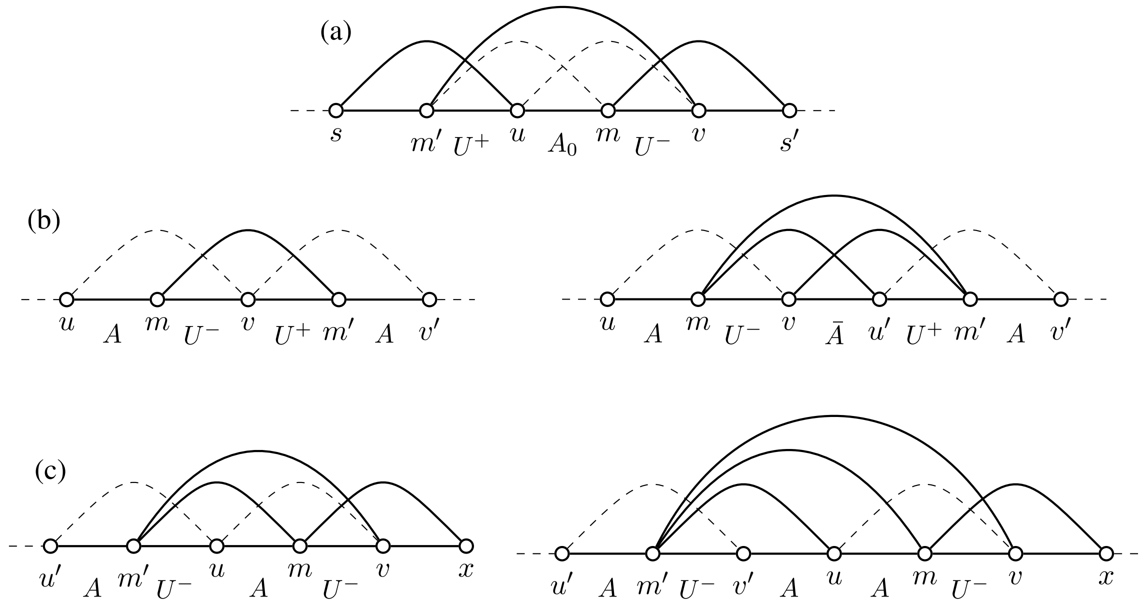

Figure 6.5: (a) A d-crossing exchange setting. (b) A zero-exchange setting, on the left, and a one-exchange setting on the right. Observe that it must be $mm' \in M$ in both cases. (c) A zero-sub setting, on the left, and a one-sub setting, on the right. Only edges and non-edges in G required by definition are illustrated.

6.3.2 Outline of Algorithm $\mathcal{A}$ and Basic Routines

Let C be a hamiltonian cycle of G. We give some definitions of subsets of the sets of ready edges R and unready edges U of C, which are useful when the algorithm picks a d-arc setting in C and attempts to align some or all of its edges. Edges in R_0 and R_p are ready but not allowed to switch, until other specific ready edges switch first. Edges in R_0 switch after specific ready edges switch, whereas edges in R_p switch when certain unready edges switch first. Aligned edges, previously in M and which can remain aligned until we reach C^t without loss of generality, move to set $Z \subset A$. Unready edges which have to be preceded by the switching of other unready edges move to set $U_p \subset U$.

www.manaraa.com

Main idea of the Algorithm

Algorithm $\mathcal{A}$ attempts to find a path between the two cycles C^i and C^t in $H(G)$. In every iteration the current cycle C has no edges in $R\backslash(R_0 \cup R_p)$ and $\mathcal{A}$ picks a d-arc $u(m)v$ with unready edge mv, recognises the d-arc setting S and applies the respective sequence of switches Q. Q attempts to align one or more misaligned edges of S. This is done without loss of generality, what is referred to as "Property N" in Section 6.3.4. With every main iteration, aligned and misaligned edges are appropriately moved to specific subsets, if needed, that is to Z, U_p, and R_p, and $\mathcal{A}$ outputs a new cycle C'. The new cycle is either closer to C^t in $H(G)$ or U_p has obtained a new edge, or identifies a setting or misaligned edges which cannot be aligned. If appropriate, the algorithm re-iterates by choosing a d-arc from the new input cycle C'.

We present the algorithm and the sequences it uses in Section 6.3.3. In Section 6.3.5, we prove that the algorithmic procedures determining the aligning sequences for the settings are correct, and finally in Section 6.3.6, we show that algorithm $\mathcal{A}$ is correct. Before these, we describe basic routines used by all the aligning sequences and $\mathcal{A}$.

Important Conventions

(1) Recall that given a cycle C we assign a left-to-right (anti-clockwise) orientation. However, when the algorithm picks a d-arc setting from C, the orientation of the vertices may be reversed, such that the unready of the d-arc is in U^-. This can happen without loss of generality, as in the next step the current cycle will be processed with all of its edges re-labeled as appropriate.

(2) The observations in Section 6.1.2 of deriving the alignment of an edge will be considered 'common knowledge' throughout the chapter, if not explicitly stated. Deriving the alignment of an edge by looking at two other edges is due to Corollary 6.1.4, and deriving required edges between an endvertex of a path and its internal vertices is done by Observation 6.1.5.

المنارة للاستشارات
www.manaraa.com

Basic Routines

We introduce the basic routines *Support*, *Switch-R*, and *Setting-R*, which $\mathcal{A}$ utilises in order to construct an aligning sequence for each of the different d-arc settings, according to the structure of the subgraph induced by the vertices of the d-arc setting.

Schematically, the basic routines which we describe thereafter in detail, are the following:

- *Support* p-switches a supporter of an unready edge to the respective d-arc and outputs a cycle where the unready edge is ready.
- *Setting-R* switches ready edges within a d-arc setting, according to the membership of the edges in subsets of R.
- *Switch-R* switches ready edges in $R \backslash R_p$ arbitrarily.

Definition 6.3.7. A vertex x on a cycle C is *p-connected* to vertex y, if $xw \in E$ for every $w \in P(x, y) \backslash \{x\}$. A vertex x is *p-connected to d-arc* $u(m)v$, if x is on the left of u and p-connected to m or on the right of v and p-connected to u. A vertex x *p-switches to vertex* y, when x is p-connected to y and switches with every internal vertex in $P(u, v)$.

Definition 6.3.8. Let x, y be two non-adjacent vertices on a cycle C, where x is on the left of y in C. Vertex h is the left (resp. right) host for xy, if h is the left (resp. right) supporter of xy after y p-switches to the right (resp. left) of x on a cycle C'.

Note: We suggest the reader perhaps to skip the study of the basic routines and follow them by means of studying one or more of the aligning sequences in the next section, Section 6.3.3.

المنارة للاستشارات
www.manaraa.com

Procedure Support(s, D)

Input: A supporter s supporting d-arc $D = u(m)v$ with unready edge mv in the current cycle C.

Output: A cycle C', where s is the left supporting vertex of mv.

IF s is on the right, THEN p-switch s to the host h of ms

ELSE p-switch s to m

FOR x in $\{u, m, v\}$:

  IF xs is in R and was aligned in C, THEN:

    Put xs in R_0.

Procedure Switch-R(H)

Input: A set of edges $H \subset E$, which are ready in the current cycle C.

Output: A cycle C', where the edges in H are not in $R\backslash(R_0 \cup R_p)$.

$R(H) := R \cap H$, $R_{0p} := R_0 \cup R_p$.

WHILE $R(H)\backslash R_{0p} \neq \emptyset$ DO:

  Switch e in $R(H)\backslash R_{0p}$.

Procedure Setting-R(S)

Input: Setting S on path P in the current cycle C with $R \neq \emptyset$.

Output: A cycle C' with $R\backslash R_p = \emptyset$.

$H =$ the set of edges induced by S in G.

$R_{0p} := R_0 \cup R_p$.

$C'' :=$ *Switch-R*(H).

WHILE $R_0 \cap H \neq \emptyset$ DO:

  FOR e_r in $R_0 \cap P$

    IF e_r is at distance more than one from any $e_m \in U_p$ THEN:

      Switch e_r, reaching a cycle C'''

      $C' :=$ *Switch-R*(H).

    ELSE put e_r in R_p, reaching a cycle C'.

المنارة للاستشارات

www.manaraa.com

6.3.3 Aligning Sequences and Algorithm $\mathcal{A}$

An aligning sequence is called *single-middle*, when the chosen d-arc is single-middle, otherwise it is called *multi-middle*. We proceed in listing the procedures required by the algorithm in order to produce appropriate aligning sequences for each setting. We can distinguish these sequences according to the location of the initial and final middle vertex/vertices of the chosen d-arc.

Note that from now on we may refer to both the aligning sequences and the procedures determining them as 'aligning sequences', especially when it is clear from the context.

Minimal Structure of the d-arc Settings

For each aligning sequence below, we will give an input d-arc setting, the edges of which are either given explicitly or can be derived by listing the d-arcs and the unready edges in U^+ or U^- it contains. E.g. for an unready edge U^+ in path u_0uvv_0 we know already that edge u_0v exists, while uv_0 does not. Recall that a setting gives information for all arcs of length 2 in relation to the input cycle C.

Two vertices u and v are (or edge uv is) *related in* C', when $uv \in C'$. If $C' = C^t$, then they are called just *related*.

The d-crossing exchange Sequence

The d-arc setting *d-crossing* is a $U^+A_0U^-$ setting on the path $sm'umvs'$, which is also in the path $u_0sm'umvs'v_0$. The d-arc $u(m)v$ is single-middle with middle vertex m and final middle vertex m', $mm' \notin E$, and m' is at distance 2 from m. If a vertex s, its left neighbour and the two vertices immediately to its right form a clique, then s is called *right-complete*. *Left-complete* is defined similarly.

المنارة للاستشارات
www.manaraa.com

d-crossing exchange(P):

Input: Path $P(u_0, v_0)$, as defined above, on the current cycle C.

Output: A cycle C' where one or more misaligned edges of the setting are aligned or $U_p = U$.

$S =$ the setting on path P.

$H =$ the set of edges induced by P in G.

IF $ss' \notin E$ THEN:

IF s is right-complete, p-connected to v and $sm' \in \bar{A} \backslash Z$ THEN $s := s$.

IF s' is left-complete, p-connected to u, $vs' \in \bar{A} \backslash Z$ THEN $s := s'$.

IF s is p-connected to v , $sm' \in \bar{A} \backslash Z$ and $m'(u)m$ is multi-middle THEN $s := s$.

IF s' is p-connected to u, $vs' \in \bar{A} \backslash Z$ and uv is multi-middle THEN $s := s'$.

C^0 :=*Support*$(s, u(m)v)$

C' :=*Setting-R*(H).

ELSE-IF $s's \in E$ and $sm', vs' \in \bar{A} \backslash Z$ THEN:

IF s is right-complete, and sm, $s'm \in E$ THEN:

$s_1 := s$, $D_1 := m'(u)m$

$s_2 := s'$, $D_2 := s(m)v$.

IF s' is left-complete and $s'u$, $sv \in E$ THEN:

$s_1 := s'$, $D_1 := u(m)v$

$s_2 := s$, $D_2 := m'(u)s'$.

C^0 :=*Support*(s_1, D_1)

C^* :=*Support*(s_2, D_2)

C' :=*Setting-R*(H).

ELSE $U_p := U$.

المنارة للاستشارات
www.manaraa.com

The zero-exchange Sequence

The d-arc setting *zero-exchange* is a $AU^{-}U^{+}A$ setting on the path $umvm'v'$ which is also in the path $xtsumvm'v's't'x'$. The d-arc $u(m)v$ is single-middle, m' is its final middle vertex, arc mm' is misaligned and of length two.

zero-exchange(P):

Input: Path $P(x, x')$, as defined above, on the current cycle C.

Output: A cycle C' with at least one of the three misaligned edges of the setting aligned or $U_p = U$.

$S =$ the setting on path P.

$H =$ the set of edges induced by P in G.

IF vv' is single-middle, THEN:

  IF s' is p-connected to v and $v's' \in \bar{A}\backslash Z$, THEN:

    $s := s'$, $D := v(m')v'$.

  ELSE-IF s is p-connected to m' and $su \in \bar{A}\backslash Z$, THEN:

    $s := s$, $D := u(m)v$.

  ELSE-IF t' is p-connected to v, $s't' \in Z\backslash\bar{A}$, $s'm, s'v \in E$ and $v'm, v'x \in E$, THEN:

    $s := t'$, $D := v(m')v'$.

  ELSE-IF t is p-connected to v, $ts \in Z\backslash\bar{A}$, $s'm', s'v, ux' \in E$, THEN:

    $s := t$, $D := u(m)v$.

ELSE-IF vv' is multi-middle, THEN:

  mv is in U_p (until vm' is aligned).

ELSE $U_p := U$.

$C^* :=$ *Support*(s, D).

$C' :=$ *Setting-R*(H).

المنارة للاستشارات
www.manaraa.com

The one-exchange Sequence

The d-arc setting *one-exchange* is a $AU^-\bar{A}U^+A$ setting on the path $umvu'm'v'$. The d-arc $u(m)v$ is single-middle, m' is its final middle vertex, $m'v$ is related, and arc mm' is misaligned and of length three.

one-exchange(P):

Input: Path $P(u, v')$, as defined above, on the current cycle C.

Output: A cycle C' with at least one of the misaligned edges of the setting aligned or $U_p := U$.

$S =$ the setting on path P.

$H =$ the set of edges induced by P in G.

IF $u'v'$ is single-middle, THEN:

  $s := v$, $D := u'(m')v'$, if $vv' \in E$

  $s := u'$. $D := u(m)v$, if $uu' \in E$.

ELSE-IF $u'v'$ is multi-middle, THEN:

  $s := u'$, $D := u(m)v$, if $uu' \in E$.

ELSE $U_p := U$.

$C^* :=$ *Support*(s, D).

$C' :=$ *Setting-R*(H).

The sub Sequences

Given a single-middle d-arc $u(m)v$ with final middle vertex m', where $m'm$ is aligned, there can be three different d-arc settings depending on the distance of m' to m. Specifically, then $u(m)v$ is in:

- *zero-sub* setting AU^-AU^- on path $u'm'umv$, when m' is at distance two from m and $m'm \in E$

المنارة للاستشارات

www.manaraa.com

- *one-sub* setting AU^-AAU^- on path $u'm'umv$, when m' is at distance three from m and $m'm \in E$
- *dis*-1-*sub* setting AU^-AAU^-A on path $u'm'v'umvv_0$, when m' is at distance three from m and $m'm \notin E$.

Below, we provide the aligning sequences for each of these three settings.

zero-sub(P):

Input: Path $P(u', v)$, as defined above, on a cycle C.

Output: A cycle C' where $mv \in U_p$.

Move mv to U_p.

---

dis-one(P):

Input: Path $P(u', v_0)$, as defined above, on a cycle C.

Output: A cycle C' where mv is aligned or $U_p := U$.

$S =$ the setting on path P.

$H =$ the set of edges induced by P in G.

IF v' is p-connected to m, THEN:

  C^0 :=*Support*(v', $u(m)v$)

  $C := C^0$.

  IF $v'v \notin E$ and v_0 is p-connected to m, THEN:

    C^* :=*Support*($v_0, v'(m)v$)

    $C := C^*$.

  C' :=*Setting-R*(H).

ELSE $U_p := U$.

المنارة للاستشارات
www.manaraa.com

The multi-middle Sequence

multi-middle(P):

Input: Path $P(x, y)$ induced by a pair of multi-middle d-arcs $u(m)v$ and $u'(m')v'$, where m' is a final middle vertex of $u(m)v$, x is the leftmost and y is the rightmost vertex of the respective setting, by definition.

Output: A cycle C', where one or more misaligned edges of the d-arc setting formed by uv and $u'v'$ or at least one unready edge moves to U_p or $U_p := U$.

S = the setting induced by path P in G.

H = the set of edges induced by P in G.

IF S is a zero-exchange THEN:

  IF u is the only left-neighbour of m, THEN:

    Move mv to U_p.

  ELSE-IF v' is the only right neighbour of m', THEN:

    Move vm' to U_p.

  ELSE $U_p := U$.

ELSE-IF S is a one-exchange THEN:

  IF both $uu', vv' \in E$, THEN:

    $C^* :=$ *Support*(u',$u(m)v$)

    $C^* :=$ Switch vm'

    $C' :=$ Setting-R(C^*).

  ELSE $U_p := U$.

ELSE-IF S is a d-crossing THEN:

  $C' :=$ d-crossing(P).

ELSE $U_p := U$.

www.manaraa.com

The algorithm $\mathcal{A}$

Input: Cycles C^i and C^t.

C :=*Switch-R*(E)

Move misaligned edges to U and aligned edges to A.

Initialise sets Z, U_p, R_0, and R_p as empty.

Move all edges in U_0 to U_p.

WHILE $U \neq \emptyset$ DO:

Choose a d-arc $u(m)v$ with its unready edge in $U \backslash U_p$ in the following order of preference:

- $u(m)v$ is single-middle
- $u(m)v$ is multi-middle and in a k-exchange setting
- $u(m)v$ is multi-middle and in a d-crossing exchange setting

IF the d-arc setting S formed by d-arcs $u(m)v$ and $u'(m')v'$ (where m' is a final middle vertex of uv) is such that there is an aligning sequence Q, THEN:

Apply Q on S and output cycle C'.

C'' := *Switch-R*(E).

Move any edges in U_0 to U_p.

IF any misaligned edge e with both vertices in S aligned, THEN move e to Z.

Set the current cycle to C.

ELSE $U_p := U$.

IF $U_p \neq \emptyset$, THEN return 'NO'

ELSE return 'YES'.

المنارة للاستشارات
www.manaraa.com

6.3.4 Property N of a Sequence

In this section, we provide a series of lemmas useful in proving the correctness of the (aligning) sequences described in Section 6.3.5 and algorithm $\mathcal{A}$ later in Section 6.3.6. Notably, when any sequence Q satisfies what is defined below as Property N, then Q can be applied at the time of the algorithmic call without loss of generality. That is, applying Q does not cause $\mathcal{A}$ to decide HC-PATH wrongly.

Recall that a sequence (of switches) Q, starts from a cycle C and switches edges in the order given by the sequence, and outputs the resulting cycle C'. Also, a subsequence of switches is a strict subset of switches of a sequence Q and not necessarily a sequence.

Definition 6.3.9. $\mathcal{S}_C$ is the set of all minimal sequences corresponding to all paths from cycle C to the target cycle C^t.

That is, every switch in each $Q \in \mathcal{S}_C$ is necessary such that the misaligned edges in C align. Another way to think about it is that we do not allow the path between C and C^t to contain cycles in $H(G)$. Clearly, if there is a path between C and C^t, then $\mathcal{S}_C \neq \emptyset$.

Given a path rst and a vertex u on a cycle C, if u is on the right (resp. left) of rst, then the u-*internal* edge of s is st (resp. rs).

Definition 6.3.10. A supporter s of an edge e is *direct* in cycle C, when there is a sequence Q_s such that s is a supporting vertex of a ready edge in some cycle C' and there is no unready edge in C which is aligned in C'.

That is, it is not required to align any unready edge, for Q_s to move vertex s next to edge e. And thus, e is the first unready edge to be in R between cycles C and C'.

Lemmas 6.3.12 to 6.3.11 provide both requirements for and implications of a vertex s being the supporter of a d-arc $u(m)v$. The lemma below shows that $d(s, m) \leq 3$, where s is the left direct supporter $mv \in U^-$ of d-arc $u(m)v$ in cycle C and if the m-internal edge of s is aligned, then it is also in $\bar{A}$.

المنارة للاستشارات
www.manaraa.com

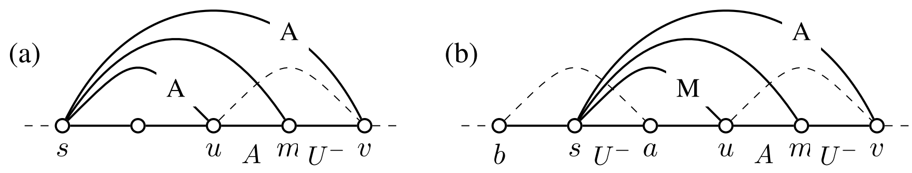

Figure 6.6: (a) Vertex s is a direct supporter of the unready edge mv and not a final middle vertex of $u(m)v$. According to Lemma 6.3.12, s must be p-connected to m and v. (b) According to Lemma 6.3.13, the left direct supporter s of d-arc mv is a final middle vertex of $u(m)v$, as $e_s = sa \in U$.

Lemma 6.3.11. *Let s be the left direct supporter of the unready edge mv of d-arc $u(m)v$ in cycle C, where e_s is the m-internal edge of s. The following are true:*

(i) $d(s,m) \leq 3$

(ii) If $e_s \in A$, then $e_s \in \bar{A}$.

Proof. (i) Vertex s is connected to u, m, v and its initial neighbour is not in e_s. Thus, there can be at most one vertex between s and u, in which case the neighbour of s in e_s is not u and $d(s,m) \leq 3$.

(ii) Suppose that e_s is aligned, but not ready, that is $e_s \in A \backslash \bar{A}$. We will show that at any case, either s is not direct or s is not a supporter of the unready edge mv. By Lemma 6.3.12, s is p-connected to $u(m)v$, and so $e_s \in A^-$. Thus, e_s requires a left supporter y such that it can be in $\bar{A}$.

CASE A. s is between $u(m)v$ and y.

Then also the d-arc for which s is the middle vertex is between y and $u(m)v$ in C.

– Suppose that s is on the left of u. Since $ys \notin C$, let $z \neq y$ be the left-neighbour of s. Due to $N(s) = \{y, z, u, m, v\}$, path $zsumv$ must be on C. Suppose that y is able to p-switch to and replace s in $z(s)u$ and then s replaces m in $u(m)v$. We reach a cycle with path $zyusvm$, where $us \in U^+$. Edge us has only two possible supporters, m and y. Thus, s is vicious to uv, as m must replace s in $u(s)v$ in order to align us, and so uv remains a d-arc.

– Suppose that s is on the right of v. Similarly to above, y is on the right of s and path $umvsz$

www.manaraa.com

must be on C. Suppose that y replaces s in the d-arc $v(s)z$ and s p-switches to u. If $my \notin E$, then my cannot switch, so y will remain on the right of m. Thus, vertex s is vicious to uv, as m is the only left supporter for $u(s)v$, and y is the only right one, so m must replace s in $u(s)v$ in order to align sv. If $my \in E$, the conclusion is similar.

CASE B. y and $u(m)v$ are on the same side of s.

Case 1. $d(s,m) = 2$.

– Suppose that s is on the left of u. Since $d(s,m) = 2$, then $zsumvx$ is a path on C and y is on the right of u. Also, $uv \notin E$, then $y \neq v$. If y is on the right of v, then $\deg(y) > 5$, considering that $N(y) = \{z, s, u, m, v, x\}$. The only case left is that $y = m$. Edge sm needs a left host and, without loss of generality, that is z. So, $mz \in E$. Since $um \notin \bar{A}$, and all the neighbours of m appear on the path $zsumvx$, then x is the only possible right supporter of um.

Let Q be a sequence starting from C such that vx is ready, x replaces m in $u(m)v$ and m p-switches to z, then we reach a new cycle C' with path $zmsuxv$. Since m is the left supporter of su, su must switch before mu, which is misaligned in C' with only possible supporters s and x. Then, x should be the left supporter of mu in some subsequent cycle, as s has to be on the right of m. This requires $sx \in E$. Without loss of generality, x is neighbours with all vertices in path $zmsuxv$, and its initial right-neighbour in C, that is it has degree six. Hence, such Q does not exist.

– Suppose that s is on the right of v. Since $d(s,m) = 2$, $umvsz$ is a path on C and the right supporter y of vs is on the left of v. With s the left supporter of mv, mv cannot switch before s, so $y \neq m$. In every other case, $\deg(y) > 5$.

Case 2. $d(s,m) = 3$.

– Suppose that s is on the left of u. Since $d(s,m) = 3$, then $zstumv$ is a path on C, which also contains all neighbours of s. Also, $zt \notin E$. Thus, y is on the right of t and one of u, m, v.

If $y = u$, then $tu \in A$, otherwise s would not be a direct supporter of mv. $N(u) = \{z, s, t, m, m'\}$, where $s \neq m'$ since $su \in A$. Thus, the required right supporter of tu such that the latter is ready in a subsequent cycle must be m, that is $tm \in E$. Let Q be the sequence

المنارة للاستشارات
www.manaraa.com

starting from C and in which u replaces s in $z(s)t$, and mv switches, resulting in a cycle C' with the path $zutsvmx$. Observe that s and m are the only possible supporters for $ut \in U^-$, with s being the only possible left supporter. Thus, s and m must support ut before we reach cycle C'. Therefore, for any sequence Q, s is vicious to $u(m)v$, since uv remains a d-arc.

Observe that if y is m or v, then y cannot be the left supporter of st, unless $\deg(y) > 5$. For example, if $y = m$, then m must be p-connected to z.

– Suppose that s is on the right of v. Since $d(s, m) = 3$, then $umvtsz$ is a path on C, which contains all neighbours of s. Thus, y, on the left of t, must be one of u, v – it cannot be m, as this would mean that mv can switch without s as the left supporter.

If $y = v$, then $vz \in E$. Let Q be a sequence starting from C, where v p-switches to z and replaces s in $t(s)z$, and s p-switches to u, reaching path $usmtvz$ on C'. The only supporters for $tv \in U^+$ are s and m. If s is the left supporter of tv then $u(s)v$ is a d-arc with m the only left supporter. And if s is the right supporter of tv, then Q reaches back to cycle C. Thus, at any case, s is vicious to $u(m)v$. It is easy to see that also $y \neq u$.

Thus, if s is a left direct supporter of $u(m)v$, then it can only be $e_s \in \bar{A}$. □

The next lemma shows that if s is a direct supporter, then it is p-connected to the unready edge, apart from one case. Figure 6.6 (a) illustrates one such case.

Lemma 6.3.12. *Let s be a direct supporter of the unready edge mv of a d-arc $u(m)v$ in cycle C. If s is not a final middle vertex of $u(m)v$, then s is p-connected to m and v, apart from the following case:*

- *s is the right supporter of mv and the left supporter s' of mv is between v and s.*

Proof. We suppose that s is not p-connected to mv, and reach a contradiction. Then there is some vertex x between s and the d-arc $u(m)v$ such that $xs \notin E$. Then x has to switch with all vertices in $u(m)v$, before s is able to be a supporting vertex of mv in some cycle C'. Let e_x be the m-internal edge of x in C. Since s is a direct supporter, e_x must have the same alignment both in C and C', therefore $e_x \in A$.

المنارة للاستشارات
www.manaraa.com

Consider that s can be either the left or the right supporter of $mv \in U^-$.

CASE 1. s is the left supporter of mv.

– Suppose that s is on the left of u. Thus, x is between s and u. If x can switch with u and v, then there is a right host (see Definition 6.3.8) x' of xv. Since x has four neighbours on its right starting from u, and it also has an initial left-neighbour, then the path $txumv$ is on C. If $xu \notin \bar{A}$, then one of its neighbours is its left supporter. Due to its degree, x' cannot be the left supporter of xu and $uv \notin E$. Thus, t is the left supporter of xu and s is on the left of t. Since $xu \in \bar{A}$, we p-switch x to x' to reach (a cycle with) path $tumvxx'$. Then s p-switches to and replaces m in uv, reaching a cycle C_s with path $tusvxmx'$. Notice that when on C, since $xu \in A$, then $su \in A$, and so $us \in U^+$. The only possible right supporter for us is m, and thus s is vicious to $u(m)v$, as $us \in A$ requires $mv \in U$. Thus, x does not exist.

– Suppose that s is on the right of v. Similarly to the above case, if x is between v and s and we switch xv and xu, then s is vicious to $u(m)v$, while we try to align us after s replaces m in $u(m)v$.

CASE 2. s is the right supporter of mv.

– Suppose that s is on the left of u. If t is the left-neighbour of s, then $N(s) = |\{t, u, m, v, h\}$, where h is the left host of sv. Thus, s is p-connected to $u(m)v$ in the path $tsumv$ on C.

–Suppose that s is on the right of v. Since there is some vertex x between v and s such that $xs \notin E$, then x has to switch with v, m, and y, which is the left supporter of mv, before mv switches – by assumption $x \neq y$. Thus $N(x) = \{y, u, m, v, w\}$, where w is a vertex between x and s. The only available host for xy is u. It is easy to see that the only right supporter, required to switch vx, is w (mv cannot switch by assumption, $uv \notin E$, and $yw \notin E$. Also, $vx \in A$, otherwise s is not direct. Thus, $vx \in \bar{A}$. It is $N(v) = \{y, m, x, w, s\}$ and $N(y) = \{z, u, m, v, w\}$, where z is the left-neighbour of y, and so path $yumvxws$ is on C. At least one final middle vertex m' is such that $m' \in N(v)$. In fact, $m' \in N(v)\backslash\{x, s, m\} = \{y, w\}$, because vx is aligned and $s \neq m'$ by assumption. Since, $m'm \notin E$, m' is on the left of u, and so $m' = y$ and $yu \in U^-$. Observe that there is no left supporter for yu. Thus, this setting is not possible, when $\mathcal{S}_C \neq \emptyset$. □

المنارة للاستشارات
www.manaraa.com

Lemma 6.3.13. *Let s be the left and direct supporter of the unready edge mv of d-arc $u(m)v$ in cycle C, where e_s is the m-internal edge of s in C. If e_s is unready, then s is one of the final middle vertices of uv in the target cycle C^t.*

Proof. We will assume the opposite of the statement, an illustration of which is in (b) of Figure 6.6, and reach a contradiction.

Suppose that e_s is unready and s is not one of the final middle vertices of uv. Then s is either on the left of u or on the right of v in C^t.

CASE A. s is on the left of u in C^i.

Let s be in the path $zstumv$. Since all of the neighbours of s are on this path, one of these vertices has to be the left supporter of st.

Case 1. s is on the left of u in C^t.

Then, $su \in A$. Also, $t \neq u$, otherwise su would be both aligned and misaligned. Edge $tu \in A$, because $su \in A$ and by Lemma 6.1.4. Only u can be the left supporter for st. According to (ii), $tu \in \bar{A}$ and u is p-connected to z. The final right neighbour of u and one of the final middle vertices of uv, say u_r, must be on the right of u, otherwise $u_r s$ would be misaligned, and thus $u_r s \in E$. But s cannot have more neighbours. Since mv is unready then $u_r m$ must be misaligned, and thus $u_r m \in E$. But then none of the neighbours of m can be its final right neighbour. Therefore, s cannot be on the left of u in C^t.

Case 2. s is on the right of v in C^t.

If $d(s,u) = 1$, then s is in the path $zsumv$, where su, $mv \in U^-$. By (i) and (ii) of Lemma 6.3.11, the left supporter y of su must be on the left of z. Once y replaces s in $z(s)u$ and s replaces m in $u(m)v$, we reach the path $zyusvm$. If y can replace s in the d-arc uv, then it must be the initial left neighbour of z. If yz is not in $\bar{A}$, then yz cannot have a left supporter while on C, due to degree. So, $yz \in \bar{A}$. Once y replaces s in d-arc uv, then s is the only right supporter for uy. Thus, s is vicious to uv.

If $d(s,u) = 2$, then s is in the path $zstumv$, where st, $mv \in U^-$. Since all the neighbours of

المنارة للاستشارات
www.manaraa.com

s are on this path and only tu can be in $\bar{A}$, by (ii) of Lemma 6.3.11 u must be the left supporter of st. It is also required that $uz \in E$, so that z is the host for su. Once u replaces s in $z(s)t$, then s replaces m in uv, reaching the path $zutsvmx$. Since s is the only left supporter for ut, we cannot align both ut and sv.

CASE B. s is on the right of v in C^i.

Case 1. s is on the right of v in C^t.

Since $vs \in A$, but $e_s \in U$, then s must be in the path $umvtsz$, where t is the left-neighbour of s and $e_s = ts \in U^+$. All the neighbours of s are in this path, so the right supporter of ts is one of u, m, v. Since only vt can be in $\bar{A}$, by (ii) of Lemma 6.3.11 v is the only possible right supporter for ts. As such, also $vz \in E$, as only z can be the host for vs. Once v replaces s in $t(s)z$, then s p-switches to u, reaching path $usmtvz$, where $tv \in U^+$. If the right supporter for tv is only s, then s is vicious to $u(m)v$, because $u(m)v$ remains a d-arc, when s reaches back to z to support tv. Thus, the right supporter of tv must be $x \neq s$. It is easy to see that x can only be on the right of z. Assuming that x can replace v in $t(v)z$, we can switch mv, and now $u(s)v$ is a d-arc. If m replaces s in $u(s)v$, then s will be vicious to $u(m)v$. If t replaces s in $u(s)v$, then we reach a cycle where $u(t)v$ is a d-arc. Since the only possible right supporters for ut are s and m, then s is vicious to $u(t)v$, when s replaces t, and s is vicious to $u(m)v$, when m replaces t.

Case 2. s is on the left of u in C^t.

Since s is misaligned to u,m, and v, then ℓ, the final left-neighbour of s, is not one of them. Thus, $N(s) = \{u, m, v, z, \ell\}$, where z is the initial right-neighbour of s. It is easy to see that $\ell \neq z$, and so ℓ is initially on the left of s. In fact ℓ is initially on the left of u, as if it is the initial left-neighbour of s, it would have to be an unready edge, but $\ell s \in A$. Thus, s must be exactly in the path $\ell umvsz$, where $mv \in U^-$ and $sv \in U^+$. Observe that none of the neighbours of s can be the right supporter of vs. The described setting of this case is not possible. We have reached a contradiction, and thus s must be a final middle vertex of $u(m)v$. □

Next, we prove that a vertex x cannot be misaligned to both vertices u and v of a d-arc $u(m)v$, except for the case where $u(m)v$ is in a d-crossing.

المنارة للاستشارات
www.manaraa.com

Lemma 6.3.14. *Let $u(m)v$ be a d-arc with $mv \in U^-$ on a cycle C, and x is a vertex not in $u(m)v$. Then x cannot be misaligned to both u and v, unless uv is in a d-crossing.*

Proof. Assume x is misaligned to both u and v.

Case A. Suppose x is on the left of u in C.

Case: $d(x, u) = 1$.

Let x be in the path x_0xumvv_0. Since $xu \in U$ and $xumv$ is not in a d-crossing, then $xu \in U^-$.

We will show that the left supporter y of xu is a fifth neighbour of x. By Lemma 6.3.11, $y \neq m$, as $um \in A\backslash\bar{A}$, and $y \neq v$ because $vu \notin E$. Finally, y cannot be on the right of v, because then $\deg(y) > 5$. So y must be on the left of u. $y \neq x_0$, because $x_0u \notin E$. Thus, $N(x) = \{y, x_0, u, m, v\}$.

Now, we will show that only one of x, m can be misaligned to v. Let Q be a sequence starting from C, which switches xu and xv. From all the neighbours of x, m is the only possible final right-neighbour of x, so mv has to precede xv in Q. At the same time, when xv switches, then the middle vertex in uv is a neighbour of x. Now, there can be one of the following: either xv is unready, and some neighbour of x supports xv, or x switches with one or more existing vertices between u and v. At any of the two cases, x cannot be misaligned to both u and v, unless $\deg(x) > 5$.

Case: $d(x, u) = 2$.

Let x be in the path $x_0xaumvv_0$. We determine the alignment of edges between x_0 and u. If $au \in M$, then it can only be $au \in U^-$, otherwise d-arc uv is in a d-crossing. But $xu \in E$, so then $au \in R$. We assume no ready edges on input cycles, thus $au \in A$ and $xa \in U^-$, and so $x_0a \notin E$. This also implies that $x_0u \in A$.

We next locate m', the final middle vertex of $u(m)v$. $N(x) = \{x_0, a, u, v, x_r\}$, where x_r is the final right neighbour of x. Then, $m'x \notin E$. Obviously, $m'x \in A$, and given that $xv \in M$, m' is on the left of x_0. Observe that m' must be the left-neighbour of x_0, that is $wm'x_0xaumv$ is on C, where w is the left-neighbour of m'. Also, any other final middle vertex of uv would have to be on the left of m' with degree more than five. So, m' is unique and $N(m') = \{w, x_0, a, u, v\}$.

المنارة للاستشارات
www.manaraa.com

Let path $x_1m'x_0xaumvv_0$ be on C, and Q a sequence starting from C, which first switches xu and xv before $m'u$. Now observe that one of the neighbours of m' has to be the middle vertex of x_0a, while Q switches $m'x_0$ and then $m'a$, and this can only be u. By now, $N(u) = \{m', x_0, x, a, m\}$. Also, $m'x_0 \in U$, because $x_0a \in A$ and $m'a \in M$, and since $m'x \notin E$. Thus, $m'x_0$ needs a right supporter and this can only be u; not possible due to degree.

Case B. Suppose x is on the right of v in C.

First of all, x is misaligned to u, m and v, has a final left-neighbour x', and an initial right-neighbour w. Thus, $umvxw$ is a path on C due to the degree of x.

Let m' be a final middle vertex of $u(m)v$. $m'x \notin E$, and thus $m'x \in A$ and m' is on the right of w.

By Lemmas 6.3.13 and 6.3.11, m is the only possible right supporter for vx. So mv and mx have to switch before vx. The left supporter s of mv must be on the left of u, otherwise $xs \in E$, since s has to p-switch to $u(m)v$ before vx switches. Since now m has degree five, $mw \notin E$. But, w is the only possible right host for mx, thus mx has no host. □

Definition 6.3.15. Given two sequences Q_1 and Q_2, then their concatenation is sequence Q_1+Q_2, which applies the switches of Q_1 and continues with the switches of Q_2, in the exact order the switches appear in Q_1 and Q_2.

Recall that $\mathcal{S}_C$ is the set of all minimal sequences starting from cycle C and resulting in the target cycle C^t. Also, an aligning sequence Q starts from a cycle C and results in a cycle C', which is closer to C^t.

In order to prove the correctness of algorithm $\mathcal{A}$, we will need to show that all aligning sequences, which are used by $\mathcal{A}$, satisfy the following property:

Property N: Let Q be a sequence starting from a cycle C and resulting in a cycle C'

(i) $Q \subset Q_C$ for some $Q_C \in \mathcal{S}_C$,

(ii) There is a sequence $Q_{C'} \in \mathcal{S}_{C'}$ such that $Q + Q_{C'} \in \mathcal{S}_C$.

Practically, if a sequence Q satisfies Property N, then by (i) all the switches it contains are

المنارة للاستشارات
www.manaraa.com

required by some sequence Q_C in $\mathcal{S}_C$ and by (ii) Q can be applied on the current cycle C without loss of generality.

In showing that some aligning sequence Q_A satisfies Property N, we will often do so for its for each of its subsequences Q_s and Q_m, which are such that $Q_s + Q_m \subset Q_A$. Recall that once Q_A identifies a supporter s of the unready edge of a d-arc $u(m)v$ on a cycle C, s p-switches to $u(m)v$. We denote the sequence of the switches of this action by Q_s. After we apply Q_s we reach a cycle C', where sm is in C and mv is ready to switch. We denote the sequence of switches starting from the output cycle of Q_s until the switch of mv, as Q_m.

If an aligning sequence Q does not satisfy (ii) of Property N, then must be some misaligned edge e which cannot align once some or all of the switches of Q are applied. We will show that for Q_s (Lemma 6.3.16) and Q_m (Lemma 6.3.17) satisfy (ii) of Property N by examining whether such an edge e exists.

Without loss of generality, from now on we assume that the supporter s which p-switches to d-arc $u(m)v$ in Q_s is the left supporter of $mv \in U^-$. By symmetry we can apply the same lemma for the right supporter of an unready edge in U^+.

Lemma 6.3.16. *Let s be the left supporter of the unready edge mv of and p-connected to d-arc $u(m)v$ on a cycle C. Moreover, let Q_s be the sequence starting from C which p-switches s to $u(m)v$ and replaces m with s in $u(m)v$ reaching a cycle C', and only under the following conditions:*

- *s is not a final middle vertex of uv*
- *s is not a supporter of an edge containing a final middle vertex of uv and a vertex not in $u(m)v$.*

If Q_s satisfies (I) of Property N, then it also satisfies (II).

Proof. Let $Q_C \in \mathcal{S}_C$ such that $Q_s \subset Q_C$, that is Q_s satisfies (I) of Property N. We will show that every switch in Q_s can precede any other switch in $Q_C \backslash Q_s$. In other words, any $e \in M$ which can precede Q_s can also switch after Q_s.

www.manaraa.com

We note that by assumption and Lemma 6.3.11, the m-internal edge of s is aligned and ready. Let m' be any final middle vertex of uv, and by assumption $m' \neq s$.

Case A. m' and s are on different sides of $u(m)v$ in C.

– Suppose that m' is on the left of u and s is on the right of v in C. We first show that $d(s, m) \neq 3$. We apply Q_s on cycle C with path $umvast$ and reach a cycle C' with path $hsmvat$, where h is the left host of ms, and so $N(s) = \{h, m, v, a, t\}$. Without loss of generality, $h = u$. Then a is the only possible left supporter for d-arc $u(s)v$, and so a must be p-connected to u. Thus, $N(a) = \{u, m, v, s, t\}$. Observe that in this structure exactly one of a, s, m will be the middle vertex of d-arc uv, for any cycle where s is the supporter of mv. Thus, s is vicious to $u(m)v$.

The only case left is that vs is in C, since $d(s, m) \leq 3$. We are looking for $e \in M$ such that it has to precede Q_s. If $e = vt$, then $st \in U^+$. We apply Q_s on cycle C with path $umvst$ and we reach a cycle C' with path $usmvt$. Then, $N(s) = \{u, m, v, t, s'\}$, where s' is the left supporter of sv, when mv switches in a subsequent cycle. By Lemma 6.3.14, t cannot be misaligned to both u and v, and thus t is a final middle of uv. By assumption, s does not support vt. Suppose that $e = s'u$ or $e = s'v$, where s' is not a final middle of uv. Then $N(s) = \{u, m, v, t, s'\}$, and s' is misaligned to both u and v. By Lemma 6.3.14, this can happen only if $s'umv$ is a d-crossing. So, s' must be in path $t's'umvst$ on C, and $s'u \in U^+$. If s p-switches to u, then s can replace m in $u(m)v$ and u in $s'(u)m$ in any order. Therefore, $e = s'u$ does not have to precede mv. If $e = s'm$, then s' is misaligned to all u, m, v. Since $s'umv$ is not in a d-crossing, this case is not possible, by Lemma 6.3.14.

– Suppose that s is on the left of u and m' is on the right of v in C. If $d(s, m) = 3$ with s in the path $bsaumvx$, then $m's \notin E$ due to the degree of s, and $N(s) = \{b, a, u, m, v\}$. Vertex s is not in a d-crossing and not a final middle vertex of uv, therefore $su \in A$, and by Lemma 6.3.11 $su \in \bar{A}$. Edge $ba \in A$, otherwise s has no final left neighbour. If s is the supporter of some misaligned edge, then observe that the only candidate is au, but then e is not in a d-crossing by assumption.

If s is in the path $basumvx$ and $e = ba \in M$, then we apply $Q_s \subset Q_C$. With a similar argument we conclude that there will be a cycle after Q_s with a path $basu$, where s can support ba.

المنارة للاستشارات
www.manaraa.com

Case B. s and m' are on the same side of $u(m)v$.

We will explore $N(s)$. Since $su, sv \in A$, then after we apply Q_s and path $usmv$ is on cycle C', s is misaligned to the one vertex from u, v to which also m' is misaligned. Thus, it must be $m's \in M$, and so $m's \in E$. We deduce that $N(s) = \{t, m', u, m, v\}$, where t is the final neighbour of s that is further from d-arc $u(m)v$ in C. Observe that there is no misaligned e supported by s with the properties defined in the assumption.

In conclusion, all switches of Q_s can be applied on C and there is $Q_{C'} \in \mathcal{S}_{C'}$ such that $Q_s + Q_{C'} \in \mathcal{S}_C$. □

The following lemma shows that we can switch a misaligned edge mv, which has become ready in cycle C under certain conditions, without loss of generality, that is satisfying Property N.

Lemma 6.3.17. *Let $u(m)v$ be a d-arc with unready edge $mv \in U^-$ in a cycle C with vertices m' and s, the final middle vertex and the left supporter of $u(m)v$, respectively. Let Q_m be the sequence which switches mv from ready to aligned, when the following conditions are met:*

- *m' is connected to both u and v*
- *when $s \neq m'$, then s is direct*
- *if mv is in a AU^-U^+A setting on path $umvm'v'$, and vm' precedes mv, then both d-arcs are single-middle.*

If Q_m satisfies (I) of Property N, then it also satisfies (II).

Proof. Let Q_m be the sequence which switches mv in a cycle C'. We will show that every switch in $Q_m \subset Q_{C'} \subset \mathcal{S}_{C'}$ can precede any switch in $Q_{C'} \backslash Q_m$. If this precedence is not possible, then it must be because there is some misaligned edge $e \in Q_{C'}$ such that e must precede Q_m.

We look at every possible edge $e \in M$ which requires one of the vertices of mv as a supporter while $mv \in M$, and specifically at the cases:

(i) both vertices of e are on the left of mv, and m or u is a possible supporter of e

المنارة للاستشارات

www.manaraa.com

(ii) both vertices of e are on the right of mv and v is a possible supporter of e

(iii) each vertex of e is on a different side of mv and any of u, m or v is a possible supporter of e.

Note that e does not contain any final middle vertices of $u(m)v$, otherwise Q_m satisfies Property N by assumption.

(i) Suppose that the vertices of e are on the left of m and one or more of u, m are supporters of e. Let path $cbaumvx$ be on C.

Case: $e = ba$

First we show that ba must be in U^- and that m' and s must be on the right of v. Since, by assumption, b is not a final middle of $u(m)v$ and uv is not in a d-crossing, then by Lemma 6.3.14 b is aligned to u and v. Thus, u is the final right-neighbour of b in C^t, and so $bu \in A$. This implies that $ba \in U^-$. It is $ca \in A$ and so $cu \in A$. Any vertex on the left of c which is a final middle of uv would have degree six, as its neighbours would be all vertices in path $cbau$, its initial left-neighbour and its final right-neighbour (either v or some other final middle). Thus, m' is on the right of v.

Next, we locate s. If $s = a$, then $N(s) = \{r, b, u, m, v\}$, where r is the final left-neighbour of s. After we apply Q_s and switch mv, we reach a cycle C' with path $usvm$. Observe that m is the only possible right supporter for d-arc $u(s)v$ in cycle C', and thus s is vicious to $u(m)v$. By Lemma 6.3.11, the left supporter of $u(m)v$ cannot be on the left of b, and by Lemma 6.3.13, $s \neq b$, or else b would be a final middle of uv. Hence, s is on the right of v.

Now, we prove that neither u or m are supporters of ba. If m is a possible supporter of ba, then $N(m) = \{b, a, u, v, m'\}$, and so m does not have a final right neighbour. Since u is the right-neighbour of ba in C, it can be its right supporter after mv switches. Suppose that u is the left supporter of ba, then $N(u) = \{h, a, b, m, m'\}$, where h is the host of bu. Observe that since $su \in E$, then $s = m'$. By Lemma 6.3.11, $u \in \bar{A}$. We are looking for a right supporter for ba different than u. It is $N(m) = \{u, v, m', a, r\}$, where $r \neq b$ is its final right neighbour. Since $mb \notin E$, m is not a supporter of ba. Also, m' is not a possible supporter of ba due to degree. Therefore, u is not the left supporter of ba.

المنارة للاستشارات
www.manaraa.com

Case: $e = zu$, where z is on the left of a

If $z = b$, then $bu \in M$ and b is not a final middle vertex of uv. This means that $bv \in M$. By Lemma 6.3.14, $bumv$ is a d-crossing. None of u or m is a possible supporter of bu.

If z is on the left of b, then $zu \in M$. By Lemma 6.3.14, since zu is not in a d-crossing with uv, z is a final middle vertex of uv; but e does not contain final middle vertices of uv.

Thus, $e \neq zu$, for any z on the left of a.

Case: $e = pq$, where p and q are on the left of u and $pq \neq ba$

We will show that m is not a supporter of pq in these cases. The neighbours of m are $N(m) = \{u, v, s, x\}$. If m is a supporter of pq, then one of p, q must be an existing neighbour of m and on the left of u, for example $q = s$. Then, $N(s) = \{u, m, v, p\}$. q is a direct supporter, and thus $d(q, m) \leq 3$. This means q is one of b, a. Since $pq \neq ba$, and given all the conditions above, either $bqumvx$ is a path on C and p on the left of b, or $pqaumvx$ is a path on C.

When $pqaumvx$ is on C, then $pq \in U^-$. Since $q = s$, $N(q) = \{p, a, u, m, v\}$. Observe that there is no final left-neighbour for q. It remains that $bqumvx$ is on C and p is on the left of b. At least one of pb, bq is unready, as $pq \in M$. Then, at least one of pb, bq is misaligned. Since b is the only vertex that can be the final left-neighbour of q, then $bq \in A$ and so $pb \in M$. Also, p is not a final middle of uv. It remains that p is the final left-neighbour of u. Therefore, $dpbqumvx$ is on C, and none of the neighbours of p can be the left supporter of $pb \in U^-$. Thus, the assumption that m is a supporter of pq is false.

(ii) The vertices of e are on the right of v and v is a supporter of e. Let path $umvxab$ be on C.

Case: $e = vx$

Since x is not a final middle vertex of $u(m)v$, it has to be misaligned to all vertices in path umv. By Lemma 6.3.11, x is misaligned to both vertices of d-arc uv and x is not in a d-crossing. Thus vx is in U^+ and so $va \notin E$. $N(x) = \{h, u, m, v, a\}$, where h is the left host of ux. Thus, the right supporter of vx must be on its left and by Lemma 6.3.11 this can only be m. Then mv precedes vx.

المنارة للاستشارات

www.manaraa.com

Case: $e = xy$, where y is on the right of x in C.

We assume that the switch of xy precedes that of mv, and we will reach a contradiction.

– Suppose that m is the left and v is the right supporter of xy. Path $umvx$ is on C, and $N(v) = \{m', m, x, y, w\}$, where w is the right host of yw. It is obvious that $umvxy$ is a path on C, with $xy \in U^+$. If y was further away from x, then v would be of degree more than 5. $mw \notin E$, otherwise m is a possible right supporter of xy, and then mv can precede xy, violating the initial assumption. Thus, v is the only right supporter for xy. If we p-switch v to w, and v replaces y in $x(y)w$, then we reach a cycle C' with path $umyxvw$. The only supporters for d-arc $x(v)w$ are m and y, which means that v is vicious $x(y)w$, and thus v is not the right supporter of xy.

– Suppose that v is the left supporter of xy. First, we look at the connectivity of vertices in the path in consideration.

If xy precedes mv, then $my \notin E$; otherwise, m can also be the left supporter of xy, and thus mv can precede xy. Since $my \notin E$, there must be at least one vertex between m and y in C^t. Let z be the right-neighbour of m in C^t. Since $xy \in M$ and z is on the right of x in C, then $xz \in M$. So, $N(x) = \{m, v, y, z, x_r\}$, where x_r is the final right-neighbour of x, as the other neighbours are on its left in C^t. Moreover, $N(m) = \{u, v, s, z, x\}$, where s is the left supporter of mv – not necessarily distinct from m'. Moreover, s is on the left of u, otherwise sx would be misaligned, and sx an edge. If vz is not an edge, then mv has exactly two possible supporters; s is the left and x is the right. In this case v cannot support xy; the switch of mv must precede xy, because x has to remain on the left of y and when mv aligns, v cannot be the left supporting vertex of xy. Thus, $vz \in E$ and $N(v) = \{m', m, x, y, z\}$. Also, observe that $s = m'$ and recall that s, and thus m' must be in the left of u.

The following claim will assist us in the rest of the proof, while we consider different paths on C, starting from and on the right of v.

Claim 1: Let P_v the path of length five starting from vertex v and extending to its right. In addition to edge vx, which is in P_v at any case, if the rest of the vertices P_v are a permutation of x_r, y and

المنارة للاستشارات
www.manaraa.com

z, then P_v cannot be on C.

Proof. We prove the claim, using the information on the neighbourhoods of the relevant vertices above. If $vxyz$ is on C, then xy is ready. If vxx_r is on C, then x_r is misaligned to y, z, and so x_r and one of y, z, are in a ready edge. Finally, if $vxwx_r$ is on C, where w is one of y, z, then xw is ready. In all cases, there is a ready edge in C, while C has only unready or aligned edges. □

Observe that path $m'vmzyxx_r$ is on the target cycle C^t, since this is the only ordering of the vertices on this path which abides to all the above constraints – to see this easily, consider $N(x)$ first. Thus, $yz \in E$.

Given that $vxyz$ or $vxzy$ contains a ready edge, then there must be some vertex b between x and y or y and z, and also by Claim 1, $b \neq x_r$. Since x has already degree five, then b is between y and z.

Due to the degree of x, $bx \notin E$, and thus by or bz is misaligned. Without loss of generality, by is misaligned. Then, $N(y) = \{v, x, z, b, y_r\}$, where y_r is the right-neighbour of y, and so $vxzbyy_r$ is a path on C. Then, it is $by \in U^+$. By Lemma 6.3.11, by has no right supporter.

– Suppose that v is the right supporter of xy. Since v is the right supporter of xy, then the left supporter of xy must be on the left and it cannot be the right supporter. This can only be m. Since xy is unready, then in some cycle C^*, v can p-switch to $x(y)q$, where q is the host of vy. After v replaces y in $x(y)q$, then the only possible supporter for xv is y. That is, v is vicious to $x(y)q$. Therefore, v cannot be the right supporter of xy.

Case: $e = wt$, where x is on the left of wt

Given that v is necessarily one of the two supporters for wt, then either in C or some subsequent cycle, mv is in C and wt is unready. If not, then v is not a supporter of wt. Therefore, without loss of generality, we assume that wt is in C, and thus $wt \in U$.

– Suppose that $x \neq m'$ and $N(v) = \{m, m', x, w, t\}$. That is, vertex $m' = s$ is both the left supporter of mv and final middle vertex of $u(m)v$. Also, v can only be the left supporter of wt, otherwise v has six neighbours, including a right host for vt.

المنارة للاستشارات
www.manaraa.com

Suppose xw is in C. If xt is not an edge, then m must be the final left-neighbour of t, thus $mt \in E$. Observe that mv must precede xv. Thus, $xt \in E$. Since both vw and xt are edges, then $xw \in \bar{A}$ – it could also be in R, but C does not have ready edges. Observe that x is both a left and right possible supporter of wt. If x is the right supporter of wt, with h the right host of xt, we p-switch x to $w(t)h$ and x replaces t reaching a cycle C' with path $umvtwxh$. Now, the only possible right supporter for $w(x)c$ is t – consider that $N(x) = \{m, v, w, t, h\}$. Thus, x is vicious to $w(t)c$ and x can be the left supporter of wt without loss of generality, and so wt does not have to precede mv.

Suppose xw is not in C. There is some vertex q between x and w. Due to the degree of v, vq is not an edge. So q has to switch with at least w and t, before v supports wt. If this is possible, then there is a right host for qt and q is also a right supporter of wt. We p-switch q to c and reach a cycle C' with path $umvxwtqc$. Now, if xt is an edge and $N(v) = \{m, m', x, w, t\}$, then the conclusion is similar to the previous case – $xw \in C$ – and mv must precede wt.

– Suppose that $x = m'$. First, we note the difference with the first case; v has now a spare neighbour, if it exists. It is $vx = vm' \in U^+$, and so q, the right-neighbour of x, is such that $vq \notin E$, and this imposes that $q \neq w$, as $vw \in E$. Since q has to switch with w and t before v can support wt, we infer that $qt \in E$.

For v to support wt, there is a subsequent cycle with path $vwtqc$. Since m', due to degree, cannot be on the right of t, then vm' is aligned in C'. Thus, if vm' precedes wt, then it also precedes mv. Since vm' precedes mv, then the right supporter s' of vm' must be one of the neighbours of v on the right of m' – otherwise s' is on the left of u and requires degree at least six. In fact, for the same reason s' cannot be on the right of t, and $s' \neq t$ by assumption. Therefore, $s' = w$. Then, $um'mvs'qwtc$ is a path on C. Thus, at the same time that q p-switches to c, also w becomes the right-neighbour of m', and we reach a cycle with path $umvm'wtqc$, and then we switch some ready edges, and reach cycle C' with path $um'mvwtqc$. Observe that $mt \notin E$, otherwise m could substitute v as the left supporter of wt, and thus mv would be able to precede wt. So, we switch wt and reach a cycle with path $um'mvtwqc$. Now, q requires a right supporter r, which is easy to see that it must be on the right of c, so $N(q) = \{m', w, t, c, r\}$. Finally, mv

www.manaraa.com

needs a right supporter in d-arc $m(v)t$, and this can only be q, thus $qm \in E$. Once wq is ready, we p-switch q to v, we switch all ready edges, and then we reach a cycle with path $um'vmqtwc$. Observe that d-arc vq has been imposed to have more than one final middle vertex, which violates one of the assumptions.

In conclusion, for all cases which agree with the assumptions, mv can precede e, without loss of generality. □

A consequence of the two last lemmas above is that once an unready edge is aligned by some sequence Q satisfying Property N, it does not have to be misaligned again. We describe this, formally, in the following corollary.

Corollary 6.3.18. *Let $u(m)v$ be a d-arc with unready edge mv in some cycle C and Q be a sequence which satisfies Property N. If after applying Q on C, mv is aligned, then we can move mv to Z.*

Proof. By the definition of Property N, the alignment of mv can precede the alignment of any other misaligned edge not aligned by Q. Thus, we can move mv to Z. □

6.3.5 Correctness of the Aligning Sequences

In this section, we show that each of the aligning sequences in Section 6.3.3 attempt to align one or more misaligned edges of the respective d-arc setting, while outputting a new cycle C' such that if $\mathcal{S}_C \neq \emptyset$, then $\mathcal{S}_{C'} \neq \emptyset$. We achieve this by showing that each aligning sequence can be broken down into smaller sequences, each of them satisfying Property N.

d-crossing exchange

Recall that a d-crossing exchange is a setting $U^+A_0U^-$ on a path $P_d \equiv m'umv$, where $m'(u)m$ and $u(m)v$ are d-arcs – see Section 6.3.1. We call the right supporter of $m'u$ and the left supporter of mv the *in-support of the d-crossing*.

المنارة للاستشارات

www.manaraa.com

In this section, we show that the d-crossing exchange sequence Q_d, presented in Section 6.3.3, is such that if the d-crossing is in the current cycle C, then Q_d satisfies Property N and thus we can apply Q_d on C without loss of generality.

In Lemma 6.3.19 we identify the possible support for the d-crossing exchange by defining requirements on its neighbourhood in E. In Lemma 6.3.20, we prove that Q_d chooses the supporter(s) of the d-crossing exchange such that Property N is satisfied.

Lemma 6.3.19. *Let a d-crossing exchange setting $U^+A_0U^-$ be on path $m'umv$ on a cycle C with no edges in R. The possible in-support of the d-crossing are vertices s and s', where $sm'umvs'$ is on C. More specifically, if the in-support of the d-crossing is:*

(a) exactly one vertex, then this is s or s'.

(b) two vertices, then s is the right supporter for $m'u$ and s' is the left supporter for mv, and:

(i) $ss' \in E$

(ii) if s is p-connected to $m'(u)m$, then s' is p-connected to m and connected to m'

(iii) if s' is p-connected to $u(m)v$, then s is p-connected to u and connected to v

Proof. Consider the induced subgraph of path $sm'umvs'$ in G.

(a) Let x be the in-support of the d-crossing. Since x supports both $m'u$ and mv, then Observe that $N(x) = \{x_0, m', u, m, v\}$, where x_0 is an initial neighbour. Thus, xu or xv is in C, and so $x = s$ or $x = s'$.

(b) The in-support of the d-crossing has two vertices.

Suppose these are x and y respectively, $x \neq y$, so x is on the left of m' and y is on the right of v. (We will prove that $x = s$ and $y = s'$.)

Vertex x is neighbours with its initial left-neighbour x_0, m', u, and the right supporter x' of $m'x$, when $m'x$ is in a cycle C' with path $um'xh_x$, where h_x is the right host of xu, also a neighbour of x. These neighbours are not necessarily distinct. Nevertheless, x must be p-connected at least to u, otherwise there is some vertex q between x and u with $xq \notin E$. Then, q must be p-connected to m such that x can also p-switch to h_x. But then without loss of generality $x = q$. Thus x is

المنارة للاستشارات

www.manaraa.com

p-connected to u. Moreover, if $x \neq s$ and $sx \notin E$, then s must be on the left of x, otherwise from the previous argument, $x = s$. By assumption, s is the left-neighbour of m', and x is on the left of m', thus $x = s$ (at any case).

Similarly, y is p-connected to m, and $y = s'$. By Lemma 6.3.11 the m-internal edges of s and s' are in $\bar{A}$, thus $xu, xs' \in \bar{A}$. Also, $N(x) = \{x_0, m', u, h_x, x'\}$, and $N(y) = \{y_0, v, m, h_y, y'\}$.

Claim. Let t be the left-neighbour of s in path $wtsm'umvs'$ on C, and t is p-connected to m. Also, s is a possible right supporter of $m'u$. Then, t is vicious to $m'(u)m$, and it cannot be the right host of xu or the left host for ms' or the right supporter for $m'x$, once $m'u$ is aligned.

Proof. We p-switch t to m, and then x to m. This imposes that w is the left supporter of $m'u$. We then do all ready edges, with one of t, x being the right supporter of $m'u$, and we reach a cycle C' with path $wsum'tmv$, without loss of generality, where $m'u$ is aligned. Observe that $m'tsmv$ is a d-crossing exchange in cycle C', and the only right supporter for $m't$ is u. Thus, t is vicious to d-arc $m'(u)m$ in C (both as a host of su and right supporter of $m's$). Further note that t has five neighbours, not including s'. $\square$.

We continue with the main proof. The Claim above immediately implies for cycle C that h_x is either m or y, and x' is on the right of m. By symmetry, we assume the same for y, that h_y is either u or x, and y' is on the left of u.

Finally, we claim that when $h_x = m$, then $h_y = x$, and vice versa. Suppose that $h_x = m$. Then, x p-switches to m and now y p-switches to x. After we switch the ready edges $m'u$ and mv, since $xv \notin E$, yv cannot switch before $m'x$. And for $m'x$ to switch it must be $x' = y$, that is y is the right supporter of $m'x$, and so $ym' \in E$.

As $x = s$ and $y = s'$, we have proven (ii). $\square$

Recall that the algorithmic procedure in Section 6.3.3 which determines the d-crossing exchange sequence Q_d, looks at the neighbourhoods of the vertices in the d-crossing exchange setting and chooses the in-support of the d-crossing, the order in which supporters p-switch, and switches ready edges from the setting, if appropriate.

المنارة للاستشارات
www.manaraa.com

With the next lemma, we show that the d-crossing exchange sequence Q_d satisfies Property N.

Lemma 6.3.20. *Given a cycle C with no ready edges in $R \backslash R_p$ and a d-crossing exchange setting $U^+A_0U^-$ on the path $m'umv$ on a cycle C, the d-crossing exchange sequence Q_d satisfies Property N.*

Proof. We can express the output sequence Q_d as the concatenation of two sequences. It is $Q_d = Q_s + Q_m$, where Q_s is the p-switching of the supporter(s) to one or both of the d-arcs in the d-crossing and Q_m is the switching of any of $m'u$, mv which are ready on the output cycle of Q_s. We will prove that Q_d satisfies Property N by showing this for Q_s and Q_m.

By Lemma 6.3.19 the supporter(s) of the d-crossing can be exactly one of two options, as described in (i) and (ii), respectively, of the same lemma. Option (i) or (ii) depends exactly on whether $ss' \in E$. If $ss' \notin E$, Q_d has to identify which of s and s' is the in-support of the d-crossing. When $ss' \in E$, Q_s has to determine the order of p-switching s and s'.

First, we show that $Q_s \subset Q_C$ for some $Q_C \in \mathcal{S}_C$ for each of the two options (i) and (ii) of Lemma 6.3.19, Cases (A) and (B) respectively, below.

CASE A. s and/or s' are possible in-support of the d-crossing.

Q_d has to choose the in-support of the d-crossing between s and s' such that if $\mathcal{S}_C \neq \emptyset$, then $\mathcal{S}_{C'} \neq \emptyset$, where C' is the output cycle of Q_d. In this case, either s or s' is p-connected to both d-arcs. That is, if s (resp. s') is the in-support of the d-crossing, then s is p-connected to v (resp. m').

We can choose as the in-support of the d-crossing any of s or s' which has the '-complete' property, in order to align both unready edges such that $\mathcal{S}_{C'} \neq \emptyset$. Without loss of generality, if s is right-complete, and s' is not left-complete, then we choose s as the in-support – and eventually s' will remain as the right supporter for mv.

If s is not right-complete and s' is not left-complete, then one of s, s' is not a possible in-support of the d-crossing (always given that $\mathcal{S}_C \neq \emptyset$). If both of them are possible in-support of the d-crossing, then observe that none of the two unready edges can have both a left and a right

www.manaraa.com

supporter. To illustrate this, if s is chosen as the in-support of the d-crossing, then $m'u$ does not have a left supporter, because the possible supporters of $m'u$ are s and s'. By (i) of Lemma 6.3.19, in this case we cannot use s' as the right supporter of $m'u$, because then s' would be the in-support of the d-crossing. Thus, we choose as the in-support exactly the one vertex from s, s' which is p-connected to the d-crossing exchange.

If none of the above can be satisfied, then $\mathcal{S}_C = \emptyset$.

CASE B. s and s' are (jointly) the in-support of the d-crossing.

By (ii) of Lemma 6.3.19, $ss' \in E$ and we can have exactly one of the two subgraphs of (a) or (b) in (ii) of the same lemma.

Depending on its neighbourhood, if s must p-switch first to $u(m)v$, then s' must p-switch to $s(m)v$ afterwards. Similarly, if s' must p-switch first (to $m'(u)m$). And since both of the s and s' are the in-support of the d-crossing, there is no further choice to be made by the sequence.

Furthermore, if s must p-switch first and s is not right-complete or s' must p-switch first and s' is not left-complete, then $\mathcal{S}_C = \emptyset$. To illustrate this without loss of generality, suppose that s and s' have the neighbourhood in (ii)(a) of Lemma 6.3.19, and so s must p-switch first to $u(m)v$. Suppose that some vertex x is the final right-neighbour of m and that we manage to switch mv using s' and x as its supporters. Then, we reach a subsequent cycle with the path $bm'uss'vmxc$. Edge $us \in U_p$, as s has to remain as the right supporter of $m'u$, and $vm \in Z$ by Corollary 6.3.18. If we choose to align $s'v \in U^-$, then by Lemma 6.3.11, there is no possible left supporter available. So, also $s'v \in U_p$. Observe that edge $m'u$ must have its left supporter on its right. All possible supporters are either in edges in U_p or in Z. So, $m'u$ cannot have a left supporter.

We have proven that for every possible neighbourhood of the vertices on the d-crossing exchange setting, $Q_s \subset Q_C$ for some $Q_C \in \mathcal{S}_C$. Now, we apply Lemma 6.3.19 to Q_s. For Case (A) above, we apply the lemma for the supporter s or s' and d-arcs $m'(u)m$ and $u(m)v$. For Case (B) we apply the lemma for both s, s' and the d-arc to which each supporter p-switches. Thus, Q_s satisfies Property N. Let C' be the cycle such that $m'u$ and/or mv is ready, as a result of applying Q_s on C. By the Property N of Q_s, there is $Q_{C'} \in \mathcal{S}'_C$ such that $Q_C = Q_s + Q_{C'}$. By definition,

المنارة للاستشارات
www.manaraa.com

$Q_m \subset Q_{C'}$. By Lemma 6.3.17, Q_m satisfies Property N.

Since both Q_s and Q_m satisfy Property N and Q_s precedes Q_m, we deduce that $Q_d = Q_s + Q_m$ satisfies Property N, since there is some $Q \in \mathcal{S}_C$ such that $Q = Q_d + Q \backslash Q_d$. □

k-exchange

Recall that a zero-exchange setting is a setting AU^-U^+A on a path $umvm'v'$ and an one-exchange setting is a setting $AU^-\bar{A}U^+A$ on a path $umvu'm'v'$, where the single-middle d-arc uv exchanges its middle vertex m with the one of the other d-arc; vv' or $u'v'$, whichever applies for the specific setting. See Section 6.3.1 for the formal definition of k-exchange.

In this section we will show that a k-exchange setting can be aligned only when $k \leq 1$, using the zero-exchange Q_{0x} and one-exchange Q_{1x} sequences. We show that both of them satisfy Property N. More specifically, for each of the two sequences, we look at the neighbourhoods of the vertices of the respective setting and assign supporters to at least one of the misaligned edges of the setting such that the resulting sequence satisfies Property N. This is shown in Lemmas 6.3.21 and 6.3.25. And finally Lemma 6.3.26 shows that for $k > 1$, a k-exchange setting cannot be aligned, and thus its existence immediately suggests that there is no path between the two input cycles.

Lemma 6.3.21. *Given a cycle C with no ready edges in $R \backslash R_p$ and a zero-exchange setting AU^-U^+A on the path $umvm'v'$ on C with d-arc uv being single-middle, the zero-exchange sequence Q_{0x} satisfies Property N.*

Proof. The algorithmic procedure which determines sequence Q_{0x} looks at the possible neighbourhoods of the vertices of the three misaligned edges in the zero-exchange setting and chooses their supporters, if they exist. Q_{0x} can be broken into two sequences such that $Q_{0x} = Q_s + Q_m$. We will prove that Q_s and Q_m satisfy Property N, which immediately implies the same for Q_{0x}.

First, we prove three claims.

The first one determines the alignment of $v's'$, where $abtsumvm'v's't'a'b'$ is a path on C with a zero-exchange setting as stated.

المنارة للاستشارات
www.manaraa.com

Claim 6.3.22. *The following are true:*

(i) If $m's'$ is an edge, then $v's'$ is aligned.

(ii) If s' is p-connected to v and s' is a supporter of vm', then $v's'$ is in $\bar{A}$, and m' and m do not support any unready edge apart from edges in the setting. Similarly, for s and mv.

Proof. (i) Suppose $m's'$ is an edge and that $v's'$ is unready. In cycle C^t, $um'v$ is a path and m is on the right of v.

If $ms' \notin E$, then there must be at least one vertex between m and s' in C^t. If this is a', right neighbour of t' in C^i, then since $a's' \in M$, also $a't' \in M$, therefore $t'a' \in U$ (or else C^i would have ready edges). Since $ms' \notin E$, then a' is also the final right-neighbour of m. One of m, a' is misaligned to v', because m and a' are an edge in C^t. By Lemma 6.3.14, $a'v' \notin M$, so $mv' \in M$. s' cannot be the host of mv', so this must be a', thus $a'v' \in E$. Observe that $t'a'$ have only one possible supporter, s', and so $t'a'$ cannot align.

If $ms' \in E$, then either $s'm$ or $s'v$ is an edge in C^t, because by Lemma 6.3.14, s' is not a final middle of d-arc uv. If vs' is an edge in C^t, then the possible supporters of $v's'$ are m' and m, given that $mv' \in E$. Neither $m's'$ nor ms' has a host such that one of them can be the right supporter of $v's'$. Thus, $vs' \notin E$ and $s'm$ must be an edge in C^t. Now observe that vm' has only one supporter, which is m.

Our assumption that $v's'$ is unready arrived to a contradiction and thus $v's' \in A$.

(ii) By (i), if s' is p-connected to v, then $v's' \in A$, and by Lemma 6.3.11, $v's' \in \bar{A}$. Moreover, m' has maximum degree and it is not in any misaligned edges other than those in the setting. If s is p-connected to v, then $su \in A$ because s is not a final middle vertex of uv, and by Lemma 6.3.14, it cannot be $sv \notin M$. Then, similarly, $su \in \bar{A}$. In both cases, due to degree it is obvious that m' and m cannot support any misaligned edges outside the setting. □

The three misaligned edges of the setting, namely mv, mm' and vm' can be aligned in two possible orders, according to the order of the edges in sequences $Q_0^1 = (mv, mm', vm')$ or its reverse, $Q_0^2 = (vm', mm', mv)$. Thus, every sequence $Q_C \in \mathcal{S}_C$ contains either Q_0^1 or Q_0^2. In fact,

www.manaraa.com

exactly one of Q_0^1, fQ_0^2 is in every $Q_C \in \mathcal{S}_C$.

Claim 6.3.23. *Given a cycle C with no ready edges in $R \backslash R_p$ and a zero-exchange setting on the path $umvm'v'$ on C, then exactly one of $Q_{0x}^i, i = 1, 2$ is such that $Q_{0x}^i \subset Q_C$ for every $Q_C \in \mathcal{S}_C$.*

Proof. Both m and m' have three neighbours towards the side of their target d-arc; m' is connected to m, v and its final left-neighbour m'_ℓ (either another final middle vertex of uv or u). Similarly for m, let m_r be its final right-neighbour.

We assume that that any of mv and vm' can switch first, and we will reach a contradiction.

Suppose that s is the left supporter of mv, when mv switches first, and s' the right supporter of vm', when vm' switches first. Therefore, $s \neq m'$, since m' is the right supporter of mv, and $s' \neq m$, since m is the left supporter of vm'. If $s = m'_\ell$, then $su \in U$ and in C with $N(s) = \{s_\ell, u, m, v, m'\}$, where s_ℓ is the left-neighbour of s. Observe that su has only one supporter. So $s \neq m'_\ell$. Similarly, $s' \neq m_r$. When mv switches first, edge mm' requires a right host, which must be a common neighbour of m and m', so this can only be v' and thus $m_r = v'$. Similarly, when vm' switches first, $m_\ell = u$ is the only left host for mm'. Without loss of generality, mv switches first and we reach the path $usvm'mv$ on a cycle C^*, where $usvm'$ is a d-crossing exchange setting. The only right supporter for us is m, but then s is vicious to $u(m)v$; a contradiction, as s is a supporter of mv.

Thus, exactly one of mv, vm' can switch first (which directly implies the lemma). □

Claim 6.3.24. *If s is p-connected to m' or s' is p-connected to m, then vv' is also single-middle.*

Proof. Suppose that s' is p-connected to m.

By Claim 6.3.22, $v's' \in A$. If $mv' \notin E$, then $ms' \in A$, and so the final right-neighbour of m is misaligned to s' and v'. The only vertex with this property is t'. But then, s' does not have a final right-neighbour, as again t' is the only candidate. Thus, $mv' \in E$, and m has already five neighbours. Therefore, vv' can have only one final middle vertex, as m' cannot have any extra neighbours.

www.manaraa.com

Suppose that s is p-connected to m' and that vv' is not single-middle. Then $mv' \notin E$. The supporters of vm' are s and m; s is the left supporter, so m must be the right. Since m is on the left of vm', a right host for $m'm$ is required. From the five neighbours of m', only v' can be the required right host, thus $mv' \in E$, and, similarly to the above, vv' must be single-middle. □

Now, we continue with the main proof.

We define Q_s and Q_m and prove that they satisfy Property N. Q_s is the sequence p-switching the chosen supporter to one of the d-arcs of the setting which is either $u(m)v$ or $v(m')v'$. Q_m is the sequence switching misaligned edges in the setting. Recall that the procedure determining Q_{0x}, and here specifically Q_s, looks at the possible neighbourhoods of the vertices in the setting, and assigns supporters to one or more of the misaligned edges of the setting.

Let s_m be the left supporter of mv and $s_{m'}$ be the right supporter of vm'. By the proof of Claim 6.3.23, s_m of mv and $s_{m'}$ of vm' must be on the same side of d-arc $u(m)v$. Without loss of generality, we suppose that both of them are on the right of v, and thus $Q^2_{0x} \subset Q_C$. And so, Q_s is the p-switching of $s_{m'}$ to d-arc $v(m')v'$. Now, we specify Q_s by looking at the possible right supporters of vm', that is which vertex $s_{m'}$ can be, for different induced neighbourhoods of the vertices of the setting in G.

CASE A. vv' is single-middle.

Observe that due to the degree of m', $s_{m'}$ is a unique vertex, found on the right of m'. Specifically, it is either s' or t'.

Case 1. Suppose that $s_{m'} = s'$.

Since vv' is single-middle, s' is not a final middle vertex of vv' and thus $v's' \in A$. It is obvious that s' is p-connected to v and a direct supporter of vv'. By Lemma 6.3.11, $v's' \in \bar{A}$. *We p-switch s' to v and do all available ready edges which are in the setting*, reaching a cycle C' with the path $um'mvs'v't'$. Edges vm', $m'm$ are now aligned. If ms' is not an edge, there is a d-crossing on $mvs'v'$. m' cannot support the d-crossing, or else m' is vicious to $v(s')v'$. If ms' is an edge, then we keep switching ready edges and we reach a cycle where all three misaligned edges of the setting are aligned.

المنارة للاستشارات
www.manaraa.com

Case 2. Suppose that $s_{m'} = t'$.

Since all edges from on the path between m' and s' are aligned, then no unready edge has to align before t' p-switches to m'. Thus, t' is a direct supporter of vm'. By Lemmas 6.3.11 and 6.3.13, since t' is not a final middle vertex of vv', then $s't' \in \bar{A}$. *We p-switch t' to m' and switch ready edges*, reaching a cycle C' with the path $um'mvtv's'a'b'$, where $mvtv'$ is a d-crossing.

Clearly for both cases 1 and 2 above, the only left supporter for vm' is m and $s_{m'}$ is the only right, both p-connected to m'. Thus, $Q_s \subset Q_C$ with $s = s_{m'}$ and d-arc $v(m')v'$. By Lemma 6.3.16, Q_s satisfies Property N. Since $Q^2_{0x} \subset Q_C$, any misaligned edges with switches in Q^2_{0x} are in Q_C. So, for each of these misaligned edges Q_m satisfies Property N by Lemma 6.3.17. In conclusion, $Q_s + Q_m$ satisfies Property N, and thus Q_{0x} also does.

CASE B. vv' is multi-middle.

Vertex m needs a final right neighbour. Since mv' is not an edge in C^t, then the final right-neighbour of m must be on the right of v'.

If ms' is an edge in C^t, then $v's' \in U$, and by Claim 6.3.22 $m's' \notin E$. Observe that now the only supporters for mm' are u and v, so vm' has to switch before mv to bring m and m' between u and v. By Claim 6.3.23, $Q^2_{0x} \subset Q_C$, for every $Q_C \in \mathcal{S}_C$. Therefore, *mv is in U_p until vm' is aligned*, and there is a d-crossing exchange setting on path $vm'v's'$.

Let mt' is an edge in C^t. By the proof of Claim 6.3.23, exactly one of the following can be true: either the left supporter of mv is on the left of mv, or only the right supporter of vm' is on its right. Suppose the first, that the left supporter s_m of mv is on its left. The five neighbours of m are $N(m) = \{s_m, u, v, m', t'\}$. Thus, edges $s't', v't' \in M$, as $s'm, v'm \notin E$. Since m' is the right supporter of mv, mv switches before vm', so mm' requires a right host. None of the common neighbours of m and m' can have that property. Thus, both the right supporter of vm' and the left supporter of mv are on the right of v, and so $Q^2_0 \subset Q_C$. Therefore, *mv must be in U_p until vm' is aligned*. Hence, Q_{0x} satisfies Property N. □

Lemma 6.3.25. *Let C be a cycle with no ready edges in $R \backslash R_p$. Given a one-exchange setting $AU^-\bar{A}U^+A$ on the path $umvu'm'v'$ where uv is a single-middle d-arc, then the one-exchange*

المنارة للاستشارات

www.manaraa.com

sequence Q_{1x} satisfies Property N.

Proof. The algorithmic procedure which determines sequence Q_{1x}, first looks at possible neighbourhoods of the vertices on the path of the setting, and chooses the supporters for each of the misaligned edges of the setting.

By assumption, uv is single-middle, so $N(m') = \{u', v', u, v, m\}$. The supporters of $u'm' \in U$. Without loss of generality, the supporters of $u'm'$ are m and v (For example, u can be too. For u to be the left supporter of $u'm'$, v has to p-switch to v'. Then $u'm'$ is ready with m the left supporting vertex. After we switch ready edges, all the misaligned edges are aligned without using u as a supporter). Obviously, there two cases; either m is the left and v is the right supporter of $u'm'$, or vice versa.

Claim 1. If $u'v'$ is multi-middle, then m is the left supporter and v is the right supporter of $u'm'$. Moreover, when v is the right supporter of $u'm'$, then m' is the left supporter of mv.

Proof. Since d-arc $u'v'$ is multi-middle, $N(m) = \{u, v, u', m, x\}$, where x is its final right-neighbour. For the first part of the claim, if $x \neq v'$, then $xm' \notin E$, due to the degree of m'. So x cannot be the right host of mm'. Thus, m can only be the left supporter of $u'm'$. For the second part of the claim, if v is the right supporter of $u'm'$, then once v p-switches to v' and we switch all ready edges, all edges of the setting are aligned, with m' being the left supporter of mv. □

Given the above, we follow the algorithmic procedure determining Q_{1x} step by step (Section 6.3.3), and show that Q_{1x} satisfies Property N. We express Q_{1x} as $Q_{1x} = Q_s + Q_m$, where Q_s is the p-switching of supporter s to d-arc D and Q_m are switches of misaligned edges of the setting – after Q_s is applied.

Case. $u'v'$ is single-middle.

If vv' is an edge, then since vu' is aligned and ready, Q_{1x} is: *v p-switches to and replaces m' in d-arc $u'v'$. We switch all ready edges on the setting*. If uu' is an edge, then since vu' is aligned and

المنارة للاستشارات
www.manaraa.com

ready, Q_{1x} is:*u p-switches to u and replaces m in d-arc uv. We switch all ready edges available on the setting*. It is easy to verify that all the edges of the setting are aligned in the output cycle. If $vv', u'u \notin E$, then $\mathcal{S}_C = \emptyset$.

By Lemma 6.3.16, Q_s satisfies Property N, where s is either u' or v. By Lemma 6.3.17 on mv (resp. $u'm'$), Q_m satisfies Property N.

Case. $u'v'$ is multi-middle.

If $vv' \in E$, by Claim 1 above, v is the right supporter of $u'm'$ and m' is the left supporter of mv. Thus the existence of uu' does not matter and so $Q_{1x} \subset Q_C$ for some $Q_C \in \mathcal{S}_C$. If $vv' \notin E$, then it must be $mv' \in E$, otherwise $u'm'$ has no right supporter. Thus, $u'v'$ is single-middle and we refer to the case above. Note that in every other case, one of the unready edges cannot be aligned, so $\mathcal{S}_C = \emptyset$. Thus, Q_{1x} satisfies Property N. □

The following lemma shows that k-exchange can be performed only for $k < 2$.

Lemma 6.3.26. *k-exchange is not possible for $k = 2$.*

Proof. Let the setting AU^-XXU^+A be a 2-exchange setting on path $u'm'v'abumv$, where $u'(m')v'$, $u(m)v$ are the two d-arcs exchanging their middle vertices. Observe that one of the two X edges must be aligned. Without loss of generality, assume that $v'a \in A$. Also observe that m must be p-connected to m'. Let x be a vertex such that xm is an edge in C^t. If x is another final middle vertex of $u'v'$, then this cannot be on the right of m', since $v'a \in A$. If x is not a final middle vertex of $u'v'$, then $x = u'$. At any case, x must be on the left of m', but then $\deg(m) = 6$. □

disconnected- and connected-sub

We recall the definitions of the settings that we examine in this section. A disconnected-1-sub setting – *dis-one* in short – is a setting U^-AAU^- on a path $u'm'v'umv$, where m' is the final middle vertex of uv and $m'm$ is not an edge. A k-sub setting is a setting where d-arcs $u'(m')v'$ and $u(m)v$ are at distance k from each other, and m' is the final middle vertex of and p-connected to $u(m)v$.

المنارة للاستشارات
www.manaraa.com

In this section, we will show that a dis-one setting and a k-sub setting for $k \leq 1$ (see Section 6.3.3) can be aligned by the respective aligning sequence, in Lemmas 6.3.27 and 6.3.28 respectively. The algorithmic procedures which determine the aligning sequences examine all possible neighbourhoods of the vertices in each setting and find their supporters. Specifically for these two settings, we show that there are unique left and right supporters for the unready edge mv, and that these are assigned correctly, if they exist, and thus Property N is satisfied.

Lemma 6.3.27. *Given a cycle C with no ready edges in $R \backslash R_p$ and a dis-one setting AU^-AAU^-A on the path $u'm'v'umvx$ with d-arc $u(m)v$, which is single-middle and mv is its unready edge, the dis-one sequence Q_{d1} satisfies Property N.*

Proof. First we show that the only possible left supporters for mv are v' and x. Due to degree, the left supporter of mv cannot be on the left of u'. We check whether u' is a supporter. Since u' is not p-connected to m and is not a final middle vertex of uv, then by Lemma 6.3.12 u' is not a direct supporter of uv or it is not a supporter. If u' is a supporter of $u(m)v$, but not direct, then there must be at least one misaligned edge with a vertex between u' and m which has to switch before u' can p-switch to mv. The only vertex which obstructs u' from p-switching to v is v', because $u'v' \notin E$. But v' is not in a misaligned edge with any of its neighbours, except for m'. Switching $m'v'$ will not remove v' from the path between u' and m. Thus, u' is not a supporter of mv, and the only possible supporters are v' and x.

Let x be the right-neighbour of v in C.

Claim 1. If x is a left supporter of mv, then $v'x \in E$.

Proof. Assume the opposite, that $v'x \notin E$. Vertex m' is connected to a middle vertex w of d-arc uv in some cycle. Since $N(m') = \{u', v', u, v, w\}$. Since x will replace m in $u(m)v$, then it must be $x = w$ and thus $m'x \in E$. Since the left host of mx cannot be v', then it must be u or m'. If it is not u, then it is m'. In that case, m' must p-switch to m first, not possible because $m'm \notin E$. So, u is the left host of xm, and thus $N(x) = \{m', u, m, v, x_0\}$, where x_0 is the initial right-neighbour of x. As now x is p-connected to u, x p-switches to u and replaces m in $u(m)v$, reaching the path

www.manaraa.com

$u'm'v'uxvm$ on a cycle C^*. For m' to replace x in $u(x)v$, $m'v'$ must be ready. The only possible supporter of $m'v'$ is u, but $v'u$ is not ready (as required). Thus, it must be that $v'x \in E$, and v' is the left host of xm.

Now, we follow the algorithmic procedure which determines Q_{d1} in Section 6.3.3 and prove that Q_{d1} is correct and satisfies Property N in the following two cases.

Case 1. If v' is p-connected to v, then v' is the only left supporter for mv and v' is the only left host for mx.

Let $Q_{v'}$ be the sequence which *p-switches* v' *to and replaces* m *in* $u(m)v$*, and switches* mv*, reaching the path* $u'm'uv'vmx$. Let m_1 be the final middle vertex of $u'v'$. The neighbourhood of v' is $N(v') = \{u, m, v, m', m_1\}$, and so $v'x \notin E$. By Claim 1, x is not the left supporter of mv. Thus, v' is the only left supporter for mv. Moreover, since $Q_{v'}$ precedes mv and mv precedes $m'u$, then $Q_{v'}$ precedes $m'u$. By this fact and Lemmas 6.3.16 and 6.3.17, $Q_{v'}$ satisfies Property N, and $Q_{d1} = Q_{v'}$.

Case 2. If v' is not p-connected to v, then x is the only left supporter for mv, v' is the only left host for mx, and m' is the only left supporter for xv.

Since v' is not p-connected to v, then either mv' of vv' is not an edge, and so v' is not a supporter for mv. It is $N(x) = \{m', v, m, h_x, x_r\}$, where h_x is the left host of mx and x_r is the right-neighbour of x in C. As the only possible left supporter is x, $v'x \in E$ by Claim 1. Recall that $N(m') = \{u', v', u, v, h'\}$, where h' is the right host for $m'u$. Then it must be $h_x = v'$. This implies $v'm \in E$, and thus $v'v \notin E$, as v' is not p-connected to v. Since x is a direct supporter of mv, then by Lemma 6.3.11 $vx \in \bar{A}$, and so $vx_r \in E$. Now, we look for the left and right supporters of xv, to ensure that m is not vicious to x. Let s_{xv} be the right supporter of xv. When x is in d-arc $u(x)v$ and mv is aligned, path $uxvms_{xv}$ is on the current cycle C^*. The right-neighbour of m is s_{xv}, given that it is the only neighbour of m which is on its right. And this is true until xv can align. Thus, s_{xv} cannot be the left or right supporter of xv. So, if y is the left supporter of xv, then $y \neq s_v$. In this case, $my \notin E$, as $N(m) = \{x_r, x, v, u, v'\}$. As with s_v, y cannot reach x when vm is aligned, as y would have to be on the right of s_v and thus on the right of path $uxvms_v$.

المنارة للاستشارات
www.manaraa.com

Therefore, the only supporters for xv are m' and m, and therefore $N(m') = \{u', v', u, v, x\}$.

Considering the initial cycle C again, u is the only possible left supporter for $m'v'$ and u' is the only host for $m'u$, as u p-switches to u' on the left of m'. Thus $u'u \in E$, without loss of generality. Let $Q_{u'}$ be the sequence which *p-switches u to u'* and Q_x the sequence which *p-switches x to v'*. If we apply $Q_{u'} + Q_x$, then we reach the path $u'um'v'xmv$. Edge mv is either ready or in U^+. If mv is ready then we switch mv. Conditionally on whether mv is ready or not, xm is either aligned or in R_p, until the right supporter of mv is adjacent to v in C, respectively.

Since u and x are the only left supporters for unready edges $m'v'$ and mv, respectively, then $Q_u + Q_x \subset Q_C$ for every Q_C in $\mathcal{S}_C$, which proves (i) of Property N. By applying Lemma 6.3.16 on $Q_u + Q_x$ and Lemma 6.3.17 on the supporters we p-switch and their respective d-arcs – $u'(m')v'$ and $v'(m)v$ – we get that $Q_{d1} = Q_u + Q_x + Q_m$, where Q_m is the switching of mv, satisfies Property N. □

Lemma 6.3.28. *Given a cycle C with no ready edges in $R \backslash R_p$ and a k-sub setting with d-arcs $u'(m')v'$ and $u(m)v$, where uv is single-middle, m' is its final middle vertex and mv its unready edge, the aligning sequence Q_{sub} applied on the k-sub setting satisfies Property N. If there is a k-sub setting in C with $k > 1$, then $\mathcal{S}_C = \emptyset$.*

Proof. Let path $umvx$ be on cycle C, for $k \leq 1$. Since uv is single-middle, x is not a final middle vertex of uv and uv is not in a d-crossing, so by Lemma 6.3.14, vx is aligned.

For $k = 0$, we get the setting AU^-AU^- on the path $u'm'umvx$ and for $k = 1$ the setting AU^-AAU^- on the path $u'm'v'umvx$. Without loss of generality, we choose x to be the right supporter of mv, since m' can p-switch and replace m in $u(m)v$, once $m'u$ aligns. That is, m' is the left supporter. Thus, Q_{0s} and Q_{1s} *move edge mv to U_p*, and by Lemma 6.3.17 and $s = m'$, the two sequences satisfy Property N. □

المنارة للاستشارات

www.manaraa.com

d-separation

A *d-separation* is a setting AU_0A on a path $umvv'$, where uv and mv' are d-arcs sharing the same unready edge mv, which is an alternative definition for an unready edge in U_0. Note that this is not a d-arc setting, as the two d-arcs cannot be related. And that is why, there is no special aligning sequence for d-separation settings in Section 6.3.3, given that each aligning sequence attempts to align a setting which contains a pair of related d-arcs, a d-arc setting.

In this section we examine which settings containing edges in U_0 can exist in a cycle C, such that $\mathcal{S}_C \neq \emptyset$ or else d-arc settings containing U_0 edges. Then, we show that moving any U_0 edge to U_p satisfies an equivalent form of Property N.

We first prove some useful claims. Claims 1 and 3 give information about the possible location of the supporters and the final middle vertices of d-arcs whose unready edge is in U_0. Claim 2 shows how close two unready edges in U_0 can be in the cycle and that if two vertices are incident to a different U_0 edge, then they cannot be related.

Claim 1. Given a AU_0A setting on the path $umvw$, the final middle vertex m' of uv is on the left of uv (and the final middle vertex of mw is on its right).

Proof. Let m' be the final middle vertex of $u(m)v$. Suppose that m' is on the right of m. $N(m') = \{f, m, v, w, a\}$, where f is the final left neighbour of m' and a is the initial right neighbour of m' – both mm' and vm' are misaligned and thus in E. So, m' is p-connected to m with $wm' \in C$. It is $wm' \in U^+$, and thus $wa \notin E$ and wm' requires a right supporter. By Lemma 6.3.11, none of the rest of the neighbours of m' is a possible right supporter of wm'. Thus, m' is on the left of u. Because of the symmetry of the d-separation setting around the U_0 edge, we deduce that the final middle vertex of mw is on its right. $\square$

Claim 2. Let $u'm'v'w'$ and $umvw$ be two d-crossing separation settings in C with U_0 unready edges $m'v'$ and mv respectively. If $\mathcal{S}_C \neq \emptyset$, then edges $m'v'$ and mv must be at distance at least three from each other, and in general m' is not a final middle vertex of uv and v is not a final middle vertex of $u'v'$. That is, the vertices of the two unready edges U_0 are not related.

www.manaraa.com

Proof. Suppose that $\mathcal{S}_C \neq \emptyset$.

– Suppose that the two unready edges $m'v', mv \in U_0$ are at distance one from each other. That is, we examine the setting $AU_0A_0U_0A$ on the path $u'm'v'mvw$ on C. Let x be a possible supporter of $m'v'$. Vertex x cannot be one of m, v, w, as then either x is not a supporter of $m'v'$ or it is not possible to reach a cycle C' where xv' is in C'. For the same reason, if x is on the right of w in C, then x must be p-connected to v', as the vertices in between cannot switch towards the left. But, then $\deg(x) > 5$. By symmetry we infer the same for vertices on the left of $m'v'$.

– Suppose that the two unready edges $m'v', mv \in U_0$ are at distance two from each other. Let the setting AU_0AAU_0A be on path $u'm'v'umvw$ with $m'v'$, $mv \in U_0$. Observe that $m'u, v'v \in A$ and so m' (resp. v) is not a final middle vertex of uv (resp. $m'u$). The final middle vertex m_1 of $m'u$, and related to u, can only be on the left of u', and so it has to be $m_1u', m_1v', m_1m' \in M$, and connected to u and its initial left neighbour u_0. Then, $m_1u' \in U^-$. Observe in $N(m_1) = \{u', v', m', u, u_0\}$ that m_1u' does not have a left supporter.

Thus, two unready edges in U_0 must be at distance at least three from each other. Let such a setting AU_0AXAU_0A be on the path $u'm'v'aumvw$. Suppose that m' is a final middle vertex of uv. Then, m' is misaligned to v', u, m, and v. So $\deg(m') > 5$, since it is also connected to its initial and final neighbours. This shows that the vertices of two unready edges in U_0 are not related in any setting. □

Claim 3. The right (resp. left) supporter s of mv is on its right (resp. left).

Proof. Suppose that s is the right supporter of mv and that it is on its left in C. Then $N(s) = \{u, m, v, h, a\}$, where h is the right host of sv and a the initial left-neighbour of s. This means path $basumvw$ must be on C. By Claim 1, s is not a final middle vertex of mw, but it could be a final middle vertex of uv. If that is the case, then $su \in U^-$ and observe that none of the neighbours of s can be the left supporter of su. Thus, s is not a final middle vertex of uv and $su \in A$. As only a can be the left supporter of su, $su \in \bar{A}$. By Claim 1, the final middle vertex m' of uv is on the left. This implies that $m'u \in M$, and so $sm' \in E$. Then, $m' = a$, and $bm'sumvw$ is a path on C.

المنارة للاستشارات
www.manaraa.com

Suppose that $m's \in U^-$. We p-switch s to w and reach path $bm'umvsw$. Because $N(m') = \{b, u, m, v, s\}$, then b must be the left supporter of $m'u$, thus $bu \in E$, and so $m'u \in R$. If we switch $m'u$, then we can switch mv. Now, we get to the path $bum'vmsw$. Observe that only m can replace s in d-arc $m(s)w$, thus s is vicious to uv. Therefore, s is not a supporter of mv; a contradiction. In conclusion, the right supporter of mv is on its right. By the symmetry of the d-separation setting AU_0A, also the left supporter of mv is on its left. □

Remember that anything shown so far in relation to Property N has not involved edges in U_0. Therefore we now show that how $\mathcal{A}$ deals with edges in U_0 satisfies Property N.

Lemma 6.3.29. *Given a cycle C with no ready edges in $R \backslash R_p$ and a U_0-setting S_{uv} with d-arcs $u'(m')v'$ and $u(m)v$ such that there is a d-separation setting on path $umvw$, the sequence Q_0 applied on the U_0-setting satisfies Property N.*

Proof. Given two unready edges in U_0, Claim 2 implies that there is at least one unready edge not in U_0 on both of its sides. Claim 3 implies that we can p-switch each supporter of the unready edge mv independently of when we p-switch the other. Thus, given a d-separation setting AU_0A on the path $umvw$, we can choose to align any of the following two U_0-settings independently: S_{uv} contains d-arc uv and S_{mw} contains d-arc mw. By Claim 2, S_{uv} extends to the left including the d-arc which contains the final middle vertex of uv, and S_{mw} extends to the right including the d-arc which contains the final middle vertex of mw.

According to the above and by the symmetry of the d-separation setting, it suffices to look at how to align S_{uv}. According to algorithm $\mathcal{A}$, Q_0 *moves* mv *to* U_p, until both S_{uv} and S_{mw} align and provide the two final middle vertices for mv, which can also be its supporters. And this implies that $Q_0 \subset Q_C \in \mathcal{S}_C$. To prove that Property N holds for Q_0, we have to consider which other edges m may need to support (on its left), before mv aligns in some subsequent cycle. By symmetry, one can prove the same for v. Let mv be in the path $baumvw$, and e a misaligned edge with vertices on the left of m in C and of which m is a supporter. $N(m) = \{u, v, r, s, x\}$, where s, r are the left and right supporter of m, respectively, and x is a fifth distinct neighbour of m. Since r is on the right of mv, then the vertices of e can only be two of u, s and x.

المنارة للاستشارات
www.manaraa.com

– Suppose that $e = sx$, and $x \neq u, a$, $s \neq a$. Since $am \notin E$, then x and s would have to switch with a before m can be support xs. But then a and u can support xs.

– Suppose that $e = ya$, $y \in \{x, s\}$. We switch y to the right such that ya is an edge in a subsequent cycle C^* with path $yaumvw$. If yu is an edge, then u can be the right supporter of xa instead of m. If yu is not an edge, then um has to switch first such that y can switch to m. But um does not have a right supporter.

– Suppose that $e = au$. Then, $au \in U$ and m is its right supporter. If uv is single-middle then, neither a nor u are in any other misaligned edges, and so m does not support any other misaligned edge apart from au. Since a is the final middle vertex of $u(m)v$ and the left supporter of mv, then au precedes mv for every $Q_C \in \mathcal{S}_C$. If uv is not single-middle, then let x be another final middle vertex of uv, which by Claim 1 is on the left of u. Vertex m can support either xa or xu. If au switches first, then x reaches u in the path $xuamv$. Since a is a supporter of xu, then we can align mv before xu without loss of generality.

– Finally, suppose that the left supporter s of mv is not a final middle vertex, and that m is the right supporter of $m'u$, where m' is the final middle vertex of uv. By Claim 3, s is on the left of u both in C and C^t and so $m's \in M$. This implies that mv can switch before $m'u$ as s can be the right supporter of $m'u$, and then m' is the right supporter for su.

Hence, at any case when mv is possible to precede e, then mv can do so without loss of generality, and thus (II) of Property N is satisfied. □

multi-middle

A multi-middle d-arc setting is formed by two related multi-middle d-arcs. Its aligning sequence is determined by the algorithmic procedure in Section 6.3.3, which gives priority to the single-middle sequences where possible.

Therefore, in Lemma 6.3.32 we prove that for each multi-middle d-arc setting, defined similarly to respective single-middle settings, the multi-middle sequence satisfies Property N.

Before we look at the individual multi-middle settings and their sequences, we provide condi-

المنارة للاستشارات
www.manaraa.com

tions on the location and alignments of the final middle vertices of a multi-middle d-arc. Specifically, given a path $xumv$, where $u(m)v$ is a d-arc with unready edge mv, the alignment of xu determines the alignment of the u-internal edges of the final middle vertices of uv which are on the left of x in C.

Lemma 6.3.30. *Let x be the left-neighbour of u, where $u(m)v$ is a d-arc in a cycle C, with unready edge mv. If $xu \in A$, then every final middle vertex m' of $u(m)v$ on the left of x is in a d-arc. Moreover, the u-internal edge of m' is unready.*

Proof. We will prove this by induction on the number of final middle vertices m_i of uv which are on the left of x and by increasing distance from x. Let us denote by m_k the k-th closest to x final middle vertex of $u(m)v$, b_k the left-neighbour of m_k and a_k its right-neighbour. By Lemma 6.3.14, any vertex on the left of x and misaligned to u must be a final middle vertex of d-arc uv. So the $a_1u \in A$, since m_1 is the first final middle vertex of uv on the left of x. Given that and $m_1u \in M$, then $b_1m_1 \in U^-$ and $b_1a_1 \notin E$. Assume that $m_ka_k \in U^-$ and that $b_k(m_k)a_k$ is its d-arc. Now consider m_{k+1}. Because $b_ka_k, bu \in A$, then $b_ku \in A$, and thus $m_{k+1} \neq b_k$. Similarly to m_1, $m_{k+1}a_{k+1} \in U^-$, since $a_{k+1}u \in A$. □

An implication of the previous lemma is the following:

Corollary 6.3.31. *Let m' be a final middle vertex of a d-arc $u(m)v$ on the left of and not adjacent to $u(m)v$, in a cycle C. If the u-internal edge of m' is aligned, then every vertex between m' and u is misaligned to u.*

Proof. Suppose that the u-internal edge of m' is aligned and that there is some vertex between m' and u which is not misaligned to u. By Lemma 6.3.30, the u-internal edge of m' is unready; a contradiction. Therefore, no vertex between m' and u is aligned to u. □

Next, we show which multi-middle settings are possible to align, when $\mathcal{S}_C \neq \emptyset$, specifically those used by $\mathcal{A}$.

Lemma 6.3.32. *Let C be a cycle with two multi-middle d-arcs $u'(m')v'$ and $u(m)v$, where m' is a final middle vertex of uv. Furthermore, C does not contain:*

المنارة للاستشارات
www.manaraa.com

- *ready edges in* $R \backslash R_p$
- *single-middle d-arcs with an unready edge in* $U \backslash U_p$

If $\mathcal{S}_C \neq \emptyset$*, then the multi-middle sequence* Q_t *applied on the setting* S *in cycle* C*, satisfies Property* N.

Proof. We look at different settings S of two related multi-middle d-arcs, following the multi-middle aligning sequence in Section 6.3.3. We do so by proving a series of claims, corresponding to different possible settings for S.

Claim 1. If S is a one-zero exchange setting in C, then Q_t satisfies Property N.
Proof: S is on path $umvu'm'v'$ with multi-middle d-arcs uv and vv'. Let $m_\ell \neq m'$ be a final middle vertex of uv and $m_r \neq m$ be a final middle vertex of vv'. Observe that m_ℓ must be on the left of u, and m_r on the right of v'. Looking at the neighbourhoods $N(m) = \{u, v, u', m_r, m'\}$ and $N(m') = \{u', v', v, m_\ell, m\}$, we find that the supporters of mv are u' and m' and the supporters of $u'm'$ are m and v. Since the setting is symmetrical around edge vu', we focus on the alignment of mv and conclude the same for both edges.

If m' is the left supporter of mv, then mm' needs a left host in a subsequent cycle. This host must be u'. But then both u' and m' remain on the left of mv, and so there is no right supporter in place for mv. Thus, it remains that m' is the left supporter of mv, and u' is the right. The host of mu' can only be u. Similarly, the left supporter of $u'm'$ is m, and v is the right and the host for vm' is v'. Thus, $u'u, vv' \in E$.

Let Q_s be the sequence which applies the *p-switching of* u' *to* u *and* v *to* v'. After applying Q_s, we reach the path $uu'mm'vv'$, where $mm' \in R$. Since the supporters of the two unready edges are unique, then $Q_s \subset Q_C$, $Q_C \in \mathcal{S}_C$. Lemma 6.3.16 applies on the supporter u' of d-arc uv and the supporter v of d-arc $u'v'$. Moreover, switching the ready edge mm' is trivially in Q_C, and we can do so whenever mm' is ready, given that m and m' cannot support any other misaligned edge. Hence, the multi-middle sequence satisfies Property N when S is one-exchange.

المنارة للاستشارات
www.manaraa.com

Claim 2. If S is a zero-exchange setting, then Q_t satisfies Property N.

Proof: Recall the two possible sequences Q^1_{0x} and Q^2_{0x} involving the three misaligned edges of a zero-exchange setting, according to the order in which the switches of the three edges appear in some $Q_C \in \mathcal{S}_C$. By Claim 6.3.23, only one of the two sequences appears in $Q_C \in \mathcal{S}_C$. We assume, without loss of generality, that $Q^1_{0x} \subset Q$ and thus $Q^2_{0x} \notin Q$. So in Q_C, mv precedes mm' and mm' precedes vm'. That is why, Q_t *moves the* vm' *to* U_p. We will show that Q_t satisfies Property N.

There is a AU^-U^+A setting on the path $umvm'v'$. Let m_ℓ be the final middle vertex of $u'v'$ and related to m', m_r be the final middle vertex of vv' and related to m, S_ℓ be the setting formed by the d-arc of m_ℓ and uv, and Q_ℓ be the sequence aligning S_ℓ and $v'm_r$ reaching a cycle C'. By the proof of the same claim, we know that $s \neq m_\ell$, where s is the supporter of mv, m the right supporter of vm', m_r is the host of mm', and m' is the right supporter of mv.

First, we show that (i) Q_ℓ precedes the switch of vm' in Q_C. The neighbours of m are in $N(m) = \{u, v, m', s, m_r\}$. Since $s \neq m_\ell$, then $m_\ell m \notin E$, so $m_\ell u$ requires a host. Q_ℓ p-switches s to m before mv switches, so $m_\ell s$ must switch after us. So $m_\ell s \in E$ and without loss of generality s is the host for $m_\ell m$. Observe that m_ℓ is the only supporter of m' and v. So, m_ℓ is the left supporter of vm', and thus the only right supporter of us. Since $m_\ell u$ precedes mv and mv precedes vm', then Q_ℓ precedes vm'. Moreover, $v'm_r$ precedes $m'm$ – m_r is the only host for mm' – and $m'm$ precedes vm'.

Next, we show that (ii) S_ℓ is either a d-crossing exchange or a dis-one setting, uv is single-middle in C', and $vm' \in U_p$ until $v'm_r$ aligns.

Since by (i) above, Q_ℓ and $v'm_r$ precede vm', *we move* vm' *to* U_p. If we run Q_ℓ, then we get to a cycle C' where mv is aligned and uv is single-middle in relation to C' with final middle vertex m_ℓ. Therefore we can apply without loss of generality the single-middle sequence which corresponds to uv in C'. Observe that due to its neighbourhood, m_ℓ can either be at distance one or two from u. Thus, if $m_\ell u$ is in C, then there is a d-crossing exchange on the path $m_\ell umv$. If $m_\ell u$ is not in C, then $am_\ell s$ is in C, where as is a d-arc. So, d-arcs as and uv form a dis-one setting.

المنارة للاستشارات
www.manaraa.com

In conclusion, S can result in a single-middle setting, for which there is an aligning sequence, as shown in (ii). Also neither v is the supporter of any unready edge to its left, nor m is the supporter of any unready edge on its right, as shown in (i). Thus, whenever vm' is ready in some subsequent cycle, it can switch without loss of generality.

Claim 3. There is no k-sub setting with multi-middle d-arcs.

Proof: Let $u'(m')v'$ and $u(m)v$ be the two multi-middle d-arcs of S, where m' is a final middle vertex of uv. Recall that in k-sub settings, m' is non the left of and p-connected to v. In the definition of the single-middle k-sub settings, $m'v$ is an edge. Note that this does not have to be the case here, as m' can be related to a second final middle vertex of uv, instead of v. We explore whether a k-sub setting can be multi-middle, given all the assumptions.

If $k = 0$, then setting AU^-AU^- is non the path $u'm'umvx$. Let $w \neq m'$ be a final middle vertex of uv different from m' and in d-arc $x(w)y$.

– Suppose w is non the right of v. If vw is in C, then $x = v$ and the d-arcs $u(m)v$ and $v(w)y$ in C are in a zero-exchange setting. This is not possible by (the proof of) Claim 2, otherwise $u(m)v$ can be in a d-crossing or dis-one setting with its related d-arc on its left in C. Thus, vw is not in C. If $d(v, w) = 2$, then uv and xy form an one-exchange setting. This, as well, is not possible by Claim 2, since Q_t has already dealt with one-exchange settings.

– Suppose w is on the left of m'. Since w is a final middle vertex of uv, then w must be misaligned to both u' and u. By Lemma 6.3.14, this is only possible if $x(w)y$ is in a d-crossing with $u'(m')u$, where $y = u'$. But then $wm' \notin E$, so uv has at least three final middle vertices and there is some final middle vertex w' which is between w and m' in C^t. As mentioned earlier in the proof, there is no final middle vertex of uv on the right of v, which is in a d-arc, so $vx \in A$ and the v-internal edge of w' must be aligned. Since vx is aligned, then by Lemma 6.3.30 w' must be in a d-arc; a contradiction. Thus, w' is not a final middle vertex of uv, and so $wm' \in E$, and thus xy is not in a d-crossing with $u'v'$. This refutes the assumption that w can be on the left of u. In conclusion, m' is the only final middle vertex of uv which is in a d-arc in C and there is no final middle vertex of uv on its right. Therefore, if uv is multi-middle, then a second final middle

المنارة للاستشارات

www.manaraa.com

vertex w is on the left of u', while the u-internal edge of w is aligned. If that is the case, then by Corollary 6.3.31, all vertices between w and u must be misaligned to u, which is not true. Thus, uv is single-middle.

If $k = 1$, let $u'(m')v'$ and $u(m)v$ form a 1-sub setting without $m'v$ necessarily be an edge. Then, the setting AU^-AAU^- setting is on the path $u'm'v'umv$. With similar arguments we deduce that uv has to be single-middle. For example, a second final middle vertex of uv on its right would impose a zero- or one-exchange setting, which are not possible for the same reasons stated for $k = 0$. And finally, the conditions of Corollary 6.3.31 are not satisfied, just as above.

Claim 4. If S is a d-crossing exchange setting, then Q_t satisfies Property N.

Proof. S is on path $m'umv$ on cycle C, with multi-middle d-arcs $m'(u)m$ and $u(m)v$.

None of the two d-arcs is in a k-exchange setting in C, since Q_t has dealt with those settings. If there is a d-arc $x(w)y$ related to $u(m)v$, on the right of m, then the two d-arcs form a k-sub setting. By definition of the k-sub setting, $x(w)y$ does not provide a final middle vertex to $u(m)v$ – but the opposite. Moreover, any other vertex in a d-arc which is on the right of $x(w)y$ cannot be a final middle vertex of $u(m)v$, by Lemma 6.3.14. By symmetry, the same conclusion is true for d-arc $m'(u)m$. Thus, it only remains that any final middle vertices of d-arcs $m'm, uv$ are incident to aligned u and v-internal edges.

Now, we show that Q_t satisfies Property N. We consider the final middle vertices of d-arc uv. Let w be a final middle vertex of uv. If w is a supporter of the d-crossing on the path $m'umv$, then wm' is in C. If w is not a supporter of the d-crossing, then by Lemma 6.3.30 every vertex between w and m' is a final middle vertex of uv. Thus, we can p-switch the supporter of the d-crossing, and once $m'u$ is ready, u switches with all final middle vertices (found on its left). If Q_s is the p-switching of the supporter(s) of the d-crossing, since the supporter(s) are unique by Lemma 6.3.19, then $Q_s \subset Q_C$ for every $Q_C \in \mathcal{S}_C$. Moreover, by Lemma 6.3.20, a supporter of the d-crossing is not a possible supporter of any other misaligned edge not in the d-crossing, thus Q_s satisfies Property N.

Now it remains to show that switching any of $m'u$, mv when they are ready satisfies Property

المنارة للاستشارات

www.manaraa.com

N. Because of symmetry of the setting, we only consider mv. If s' is the right-neighbour of mv and s' is not a supporter of the d-crossing, then $u(m)v$ satisfies Property N, by Lemma 6.3.17. Let the path $m'umvs'x$ be on C. Suppose now that s' is the left supporter of mv. Since all the final middle vertices of $u(m)v$, apart from m', are on the right of v and their v-internal edges are aligned, then by Lemma 6.3.30 s' is a final middle vertex of uv. By assumption and since $e \notin U_p$, e must be in another multi-middle d-crossing setting, at a distance in C such that s' or v be neighbours with vertices of e.

The series of Claims 1 to 4, concludes the proof. □

6.3.6 Correctness of $\mathcal{A}$

All the d-arc settings defined in Section 6.3.1 are defined based on the distance between the two d-arcs $u'(m')v'$ and $u(m)v$ and on whether the middle vertices m' and m move towards the same or opposite directions. In the next lemma, we show that the aligning sequences in Section 6.3.3 deal with all possible d-arc settings which can occur in any cycle C in a switching sequence starting from a cycle C^i and ending with a cycle C^t.

Proposition 6.3.33. *Let S be a d-arc setting in a cycle C of a switching sequence Q, starting from cycle C^i and ending in cycle C^t. If $\mathcal{S}_C \neq \emptyset$, then there is an aligning sequence which accepts S as an input.*

Proof. In other words, we will show that if S can be aligned, then there is some aligning sequence which will align (or change the state of) one of its unready edges.

There are four categories of d-arc settings of two related d-arcs $u'(m')v'$ and $u(m)v$, where m' is the final middle vertex of $u(m)v$:

- k-sub, where m' is p-connected to m and m is not the final middle vertex of $u'v'$
- k-exchange, where m' is p-connected to m and m is the final middle vertex of $u'v'$
- dis-k-sub, where m' is not p-connected to m and m is not the final middle vertex of $u'v'$
- d-crossing exchange, where the two d-arcs cross in both cycles but they 'exchange' their relative position on the two cycles

المنارة للاستشارات
www.manaraa.com

Finally, in the d-separation settings, two d-arcs uv and mw cross in C^i on path $umvw$ with $mv \in U_0$ and 'separate' in C^t, where they are at distance one or more from each other.

It is clear by Lemma 6.3.29 that we consider every possible distance between d-arcs with an unready edge in U_0. When the unready edge of d-arc $u(m)v$ is either in U^- or in U^+, then the initial middle vertex m of d-arc $u(m)v$ has an orientation in relation to u and v in the target cycle C^t. When the unready edges of $u(m)v$ and $u'(m')v'$ have the same orientation, that is both in U^- or U^+ then the k-sub and dis-one settings cover both of these by symmetry. When the orientation is different, one unready edge is in U^- and one is in U^+. The k-exchange settings cover the case when U^+ edge is non the right of the U^- edge. The remaining case is covered by the d-crossing exchange setting.

We show that for each of the possible orientations, the aligning sequences consider all feasible d-arc settings. To do so, we consider the distance between the two related d-arcs in the cycle C. For the orientations explored by the k-sub and k-exchange settings, Lemmas 6.3.28 and 6.3.26 show that each of these categories of d-arc settings does not exist beyond a certain distance, when $\mathcal{S}_C \neq \emptyset$, and thus are not assigned an aligning sequence.

What remains to prove concerns the d-arc settings with two d-arcs having unready edges of different orientation, and where the U^+ edge is non the left of the U^- edge. Let $u'(m')v'$ be on the left of $u(m)v$ in cycle C and, according to the above, edges $u'm'$ and mv are unready and m' is related to v. When the two d-arcs cross both in C^i and C^t, we get a d-crossing exchange. In this case, $m' = u$. Since m' is between u and v and non the left of u' and v' in C^t, then both u' and v' are misaligned to u. So the two d-arcs cannot be adjacent, that is $v' = u$, otherwise $u'u$ is not an edge. Suppose that the two d-arcs are at distance at least one from each other, that is $d(v', u) \geq 1$. Since m' is related to v, then $m'u \in M$. The neighbours of m' are $N(m') = \{u', v', u, v, a\}$, where a the initial left-neighbour of u', and the neighbours of u $N(u) = \{m, m', u', v', u_\ell\}$, where u_ℓ is its final left-neighbour. Since u is misaligned to all of its left neighbours and aligned to m, then u_ℓ must on the left of u'. Observe that $v'u \in U^+$, and none of the neighbours of u can be the right supporter for $v'u$. This proves that $\mathcal{S}_C = \emptyset$. □

Proposition 6.3.34. *Every aligning sequence satisfies Property N.*

المنارة للاستشارات
www.manaraa.com

Proof. This is a direct implication of Lemmas 6.3.20, 6.3.21, 6.3.25, 6.3.27, 6.3.28, and 6.3.32. □

Theorem 6.3.35. *Algorithm $\mathcal{A}$ decides HC-PATH for a graph G of maximum degree* 5.

Proof. We prove this by showing that $\mathcal{A}$ produces a sequence of switches which satisfies Property N, and which corresponds to a path in $H(G)$.

First of all, given any cycle C, we implicitly consider that $\mathcal{A}$ updates sets A, M, R, and U whenever it outputs a new current cycle C' adjacent to C in $H(G)$. In the same fashion, $\mathcal{A}$ updates particular subsets of the sets above which are defined by edges in G, in relation to the position of their vertices in C' and C^t. (For example, recall that an edge in U in a cycle C can be in exactly one of U^-, U^+, or U_0, depending on the arcs of length 2 induced by the cycle in G).

Given the input cycles C^i and C^t of graph G, $\mathcal{A}$ starts with procedure *Switch-R*, which switches all the available ready edges until $R = \emptyset$, reaching a cycle C with edges only in A and U. It is obvious that if $C \neq C^t$, then it contains at least one unready edge and no ready edges. In fact C contains at least two unready edges, as we show with the next claim.

Claim: There is at least one pair of related d-arcs, and thus a d-arc setting in C, when $C \neq C^t$.

Let $u(m)v$ be the only d-arc in C. Since there is no other unready edge in C, then the final middle vertices of uv are currently incident to edges in C which are aligned. Let m' be one of the final middle vertices of uv and also $m'u \in M$, without loss of generality. Then $m'u$ is not in C, otherwise it would be unready. Since the u-internal edge of m' is aligned, then by Corollary 6.3.31, $xu \in U$, where x is the left-neighbour of u. Thus, C has at least two unready edges.

We continue with the correctness of the algorithm. Next, $\mathcal{A}$ moves all edges in U_0 to U_p, which satisfies Property N, by Lemma 6.3.29. The main body of $\mathcal{A}$ (main loop) can determine correctly whether $\mathcal{S}_C$ is empty or not. Since there is at least one unready edge in $U \backslash U_p$, then by the claim above there is a d-arc setting S. $\mathcal{A}$ determines whether there is an aligning sequence Q for S, according to Proposition 6.3.33. If not, then $\mathcal{S}_C = \emptyset$. We apply the aligning sequence Q to S. By Proposition 6.3.34, Q is such that either satisfies Property N, or sets $U_p = U$, because S does not

www.manaraa.com

satisfy any structural requirement such that $\mathcal{S}_C \neq \emptyset$. Consequently, if $\mathcal{S}_C \neq \emptyset$, then Q outputs a new cycle C' with $\mathcal{S}'_C \neq \emptyset$. As such, every unready or misaligned edge in C which is aligned in C' is now in set Z, by Corollary 6.3.18.

Next, we show that each iteration of $\mathcal{A}$ with input cycle C outputs a cycle C' closer to the target cycle C^t than C is. If $|M(C)|$ is the number of misaligned edges in a cycle C, then we will show that $|M(C')| < |M(C)|$, by assigning at least one misaligned edge to every aligning sequence, which $\mathcal{A}$ applies. Let Q be an aligning sequence applied in C. If Q moves an unready edge e from U to U_p, then e aligns later in some cycle C^*. This will happen as a consequence of aligning another misaligned edge e^* during some aligning sequence Q^*. Q^* will align at least two edges, the misaligned edge which provides a supporter to e and, via Procedure *Supporter*, e, which we assign to Q. If Q does not only move edges from U to U_p, then it aligns at least one edge which was misaligned in C. On the other hand, it may move edges from A to R_p. However, every edge in R_p is associated with some ready or unready edge in that both are aligned by the same sequence. Thus, aligned edges which become misaligned are eventually put back again by sequences already assigned a misaligned edge. Finally, every time an aligning sequence is applied, $\mathcal{A}$ applies procedure *Switch-R* on the output cycle C', which switches ready edges in $R \backslash R_p$. After doing so, we reach a cycle C'' with $R \backslash R_p = \emptyset$. Since this procedure only switches ready edges, it cannot decrease the number of misaligned edges, thus $|M(C'')| \leq |M(C')|$. In conclusion, it is always $|M(C'')| < |M(C)|$ for every iteration of the main loop of $\mathcal{A}$. Thus, $\mathcal{A}$ both terminates and decides the problem correctly. □

Theorem 6.3.36. *$\mathcal{A}$ decides 5-HC-PATH in linear time.*

Proof. We refer to the last part of the proof of Theorem 6.3.35, where we count the number of misaligned edges on each cycle along the sequence of switches (or the path of cycles) which takes us from C^i to C^t. For every application of an aligning sequence or move of an unready edge from U to U_p we can assign at least one misaligned edge which is aligned at some point along the path, if not on the current iteration. Thus, we need $\mathcal{O}(|E|)$ iterations in order to reach C^t from C^i. In each iteration, there is a constant number of operations related to the chosen d-arc setting. Each setting

المنارة للاستشارات
www.manaraa.com

contains a constant number of edges and there is a constant number of unit operations applied on each of them. Thus, the overall running time is still $\mathcal{O}(|E|)$. $\square$

www.manaraa.com

Chapter 7

Conclusions

In this final chapter we briefly discuss the results presented in this thesis. We contextualise our work within the wider research area and we summarise our results. Finally, we pose some open questions deriving from our work and also discuss the possible future directions for reconfiguration problems.

7.1 Graph Colouring Reconfiguration

A great part of our work focussed on questions on Graph Colouring Reconfiguration. In Chapter 4, we provided results on the diameter of the colour graph which also appear in the respective publications [6, 7]. We determined sufficient conditions for the reconfiguration graph to have a diameter at most quadratic in the number of vertices. We gave examples of graph classes, such as chordal graphs and chordal bipartite graphs, that satisfy these conditions and described a class of graphs that show that our quadratic bound is sharp. More specifically, given a k-colourable chordal graph G, $R_\ell\,(G), \ell \geq k+1$ is connected and has diameter $\mathcal{O}(n^2)$.

The outcome coincides with the conjecture that if G is a k-colourable graph, then the diameter of $R_{k+2}(G)$ is $\mathcal{O}(n^2)$ [15].

In [7] we posed open questions on graph classes that generalise our result for chordal graphs:

www.manaraa.com

1. We know that a k-colourable chordal graph has bounded treewidth $t = k - 1$. Is it true that $R_{t+2}(G)$ of a graph G of bounded treewidth t has diameter at most $\mathcal{O}(n^2)$?
2. We know that a chordal graph is also perfect. Is it true that for all k-colourable perfect graphs G, $R_\ell(G), l \leq k+1$ is connected and has diameter at most $\mathcal{O}(n^2)$?

The first question was answered in the affirmative in [5]. The second remains open.

In Chapter 5 we explore the k-EXTRA COLOUR PATH problem, originally posed by Cereceda in his thesis [15] and which results naturally from k-COLOUR PATH. k-EXTRA COLOUR PATH accepts the no-instances of k-COLOUR PATH as input and asks whether a path using t-colourings exists, where $t > k$.

We have given initial results for k-EXTRA COLOUR PATH, as well as specific results for the case $k = 3$. We have shown that using $k-1$ extra colours to find a path between two k-colourings is sufficient for any instance, and also required by some of them. Moreover, we give examples of instances for which $k-1$ extra colour is always sufficient. Another result is based on the definition of a disconnected pair of colour sets. Recall that given an instance (G, α, β) of k-COLOUR PATH, then a vertex v with $\alpha(v) = i$ and $\beta(v) = j$ belongs to colour set $V_{i,j}$. We have given conditions such that $k-2$ extra colours are enough to turn a no-instance of k-COLOUR PATH to a yes-instance of k-EXTRA COLOUR PATH, when there is a pair of colour sets which is an independent set.

According to Cereceda [15], using extra colours to obtain a transformation between colourings has been examined before, but not directly related to or defined as a reconfiguration problem.

7.1.1 Open Questions on Graph Recolouring

It is an open problem whether k-EXTRA COLOUR PATH accepts a polynomial algorithm, for some integer k. We hope that our initial insight in Chapter 5 will help in investigating this further. Of interest would be to discover classes of graphs which persist in polynomial solutions. For example lattices embedded on the torus (or otherwise the cartesian product of two cycles) provide us with both yes and no instances, as shown by Theorems 5.3.3 and 5.2.8. Thus, one could explore graphs which are denser than the latter but somewhat retain the regularity of a lattice and/or torus.

المنارة للاستشارات
www.manaraa.com

Furthermore, the technique used in Theorem 5.3.3 of partitioning the graph into two parts $G \setminus H$ and H, where H is a maximal independent set, could be extended to more general classes of graphs and lead to either more polynomial results or some reduction from some known hard problem.

Perhaps the most prominent open question in Graph Colouring Reconfiguration has been that of determining the complexity of k-MIXING, as the first and only result on this was published in [18]; see Chapter 2 for more details on why 3-mixing is NP-complete. Does the hardness of 3-MIXING [18] imply that k-MIXING is at least as hard for $k \geq 4$? If we assume the latter, then is there an interesting class of graphs for which k-MIXING can be answered in polynomial time? Would it be sensible to look for a class with some geometric property similar to the case of 3-MIXING, where the problem becomes polynomial for planar bipartite graphs? When G is bipartite and not 3-mixing, it contracts to the 6-cycle. Can we define a class of graphs depending on k to which every graph G contracts, when G is not k-mixing?

7.2 Hamiltonian Cycle Reconfiguration

Although the reconfiguration of combinatorial problems has been an area of growing interest, Hamiltonian Cycle Reconfiguration has not been visited at all, to the best knowledge of the author. It has been implicitly posed as an open problem in [44], where authors ask the same question about the Travelling Salesman Problem, which can be considered as a generalised version of Hamiltonian Cycle. Thus, our result in Chapter 7 is the first result on Hamiltonian Cycle Reconfiguration, and specifically presents a class of graphs which accepts a polynomial algorithm.

Whether Hamiltonian Cycle Reconfiguration is hard for graphs of bounded degree remains an open question. If we conjecture that it is computationally hard, then it would be interesting to find more classes of graphs for which the problem can be decided in polynomial time. It would be reasonable to build directly on the work of this thesis and investigate graphs of bounded degree k. Do instances of those graphs accept a polynomial solution when $k > 5$? If not, then is there some exact constant $k > 5$ for which the problem becomes hard?

المنارة للاستشارات

www.manaraa.com

7.3 Epilogue

After the work on SAT Reconfiguration and Graph Colouring Reconfiguration, the research community extended its focus further amongst classic graph theory and combinatorial problems. For an NP-complete problem to become PSPACE-complete in its reconfiguration version is now thought the default pattern, although there have been exceptions from the very first published work, e.g. for 3-COLOURING.

Nevertheless, one can find the study of the reconfiguration version of a problem worth exploring, independently of how high is the expectation of it following the established pattern or of falling into the exceptions. Until proven, each problem maintains its own interest. And especially when the reconfiguration rule is not always naturally imposed by the statement of the original problem, this creates an additional motivation to explore how the defined reconfiguration rule affects the complexity outcome and why. For example, there is more than one 'natural' minimal reconfiguration step for the reconfiguration of independent sets or hamiltonian cycles. Equally interesting is to work on a restricted class of instances of a reconfiguration problem and possibly find a polynomial algorithm or look at specific features of the solution graph such as its diameter. All of these different and inherent motivations have resulted in numerous results within the last decade.

Perhaps an interesting meta-question is to give some form of general justification to the relation between the complexity of an original combinatorial problem P and its reconfiguration version $R(P)$, and specifically why and when exceptions arise. In particular, when an NP-complete problem P is given, then $R(P)$ is PSPACE-complete with very few exceptions discovered to date. What do those exceptions suggest for the problems P and $R(P)$? It seems that in these exceptional cases, the constraints of the original problem coupled with the reconfiguration rule result in an "easy" structure of the reconfiguration graph, such that to decide its properties is easier than to construct its vertices (solutions to the original problem). Can this or some similar observation be rigorously answered and perhaps define reconfiguration (complexity) classes according to some level of constraints and/or reconfiguration rules?

One of those exceptional cases is 3-COLOURING, which is very well-known to be NP-complete,

المنارة للاستشارات
www.manaraa.com

but 3-COLOUR PATH is in P. Intuitively, we could claim that because the non-trivial instances of 3-COLOUR PATH are bipartite graphs, and 2-COLOURING is in P, then in some sense the nature of the reconfiguration rule restricts the expected complexity of the reconfiguration version. Rather than relying on intuition, a meta-theorem is needed to refute or confirm the former. And for example, justify why k-COLOUR PATH is in P for $k = 3$, but PSPACE-complete for any $k > 3$.

المنارة للاستشارات
www.manaraa.com

www.manaraa.com

Bibliography

[1] AARDAL, K., VAN HOESEL, S. P. M., KOSTER, A. M. C. A., MANNINO, C., AND SASSANO, A. Models and solution techniques for frequency assignment problems. *Annals of Operational Research 153*, 1 (2007), 79 – 129.

[2] ACHLIOPTAS, D., COJA-OGHLAN, A., AND RICCI-TERSENGHI, F. On the solution-space geometry of random constraint satisfaction problems. *Random Structures & Algorithms 38*, 3 (2011), 251–268.

[3] BELCASTRO, S.-M., AND HAAS, R. Counting edge-kempe-equivalence classes for 3-edge-colored cubic graphs. *Discrete Mathematics 325* (2014), 77 – 84.

[4] BILLINGHAM, J., LEESE, R., AND RAJANIEMI, H. Frequency reassignment in cellular phone networks. Tech. rep., Smith Institute Study Group, 2005.

[5] BONAMY, M., AND BOUSQUET, N. Recoloring bounded treewidth graphs. *Electronic Notes in Discrete Mathematics 44* (2013), 257–262.

[6] BONAMY, M., JOHNSON, M., LIGNOS, I., PATEL, V., AND PAULUSMA, D. On the diameter of reconfiguration graphs for vertex colourings. *Electronic Notes in Discrete Mathematics 38*, 0 (2011), 161 – 166. The Sixth European Conference on Combinatorics, Graph Theory and Applications, EuroComb 2011.

[7] BONAMY, M., JOHNSON, M., LIGNOS, I., PATEL, V., AND PAULUSMA, D. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *J. Combinatorial Optimization 27*, 1 (2014), 132–143.

www.manaraa.com

[8] Bonsma, P. The complexity of rerouting shortest paths. *Theoretical Computer Science 510* (2013), 1–12.

[9] Bonsma, P. Independent set reconfiguration in cographs. In *Graph-Theoretic Concepts in Computer Science: 40th International Workshop, WG 2014, Nouan-le-Fuzelier, France, June 25-27, 2014. Revised Selected Papers* (2014), D. Kratsch and I. Todinca, Eds., Springer International Publishing, pp. 105–116.

[10] Bonsma, P., Cereceda, L., van den Heuvel, J., and Johnson, M. Finding paths between graph colourings: Computational complexity and possible distances. *Electronic Notes in Discrete Mathematics 29* (2007), 463 – 469. European Conference on Combinatorics, Graph Theory and Applications European Conference on Combinatorics, Graph Theory and Applications.

[11] Bonsma, P., Kamiński, M., and Wrochna, M. Reconfiguring independent sets in claw-free graphs. In *Algorithm Theory – SWAT 2014: 14th Scandinavian Symposium and Workshops, Copenhagen, Denmark, July 2-4, 2014. Proceedings* (2014), R. Ravi and I. L. Gørtz, Eds., Springer International Publishing, pp. 86–97.

[12] Bonsma, P., and Mouawad, A. E. The complexity of bounded length graph recoloring. *CoRR abs/1404.0337* (2014).

[13] Bonsma, P. S., and Cereceda, L. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science 410*, 50 (2009), 5215–5226.

[14] Calamoneri, T. The $L(h, k)$-labelling problem: An updated survey and annotated bibliography. *The Computer Journal 54*, 8 (2011), 1344–1371.

[15] Cereceda, L. *Mixing Graph Colourings*. PhD Thesis, London School of Economics and Political Science, London, 2007.

[16] Cereceda, L., van den Heuvel, J., and Johnson, M. Mixing 3-colourings in bipartite graphs. In *Graph-Theoretic Concepts in Computer Science, 33rd International Workshop,*

المنارة للاستشارات
www.manaraa.com

WG 2007, Dornburg, Germany, June 21-23, 2007. Revised Papers (2007), A. Brandstädt, D. Kratsch, and H. Müller, Eds., vol. 4769 of *Lecture Notes in Computer Science*, Springer, pp. 166–177.

[17] CERECEDA, L., VAN DEN HEUVEL, J., AND JOHNSON, M. Connectedness of the graph of vertex-colourings. *Discrete Mathematics 308*, 5-6 (2008), 913–919.

[18] CERECEDA, L., VAN DEN HEUVEL, J., AND JOHNSON, M. Mixing 3-colourings in bipartite graphs. *European Journal of Combinatorics 30*, 7 (2009), 1593–1606.

[19] CERECEDA, L., VAN DEN HEUVEL, J., AND JOHNSON, M. Finding paths between 3-colorings. *Journal of Graph Theory 67*, 1 (2011), 69–82.

[20] CHOO, K., AND MACGILLIVRAY, G. Gray code numbers for graphs. *Ars Mathematica Comporanea 4*, 5-6 (2011), 125–139.

[21] DEMAINE, E. D. *Playing Games with Algorithms: Algorithmic Combinatorial Game Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 18–33.

[22] DIESTEL, R. *Graph Theory*, 3rd ed. Springer-Verlag, Heidelberg, 2005.

[23] DIRAC, G. On rigid circuit graphs. *Anh. Math. Sem. Univ. Hamburg 25*, 1-2 (1961), 71–76.

[24] DOWNEY, R. G., AND FELLOWS, M. R. *Parameterized Complexity*. Springer-Verlag, 1999.

[25] DYER, M., GOLDBERG, L. A., AND JERRUM, M. Systematic scan for sampling colorings. *The Annals of Applied Probability 16*, 1 (2006), 185–230.

[26] EISENBLATTER, A. *Frequency assignment in GSM networks: Models, heuristics, and lower bounds*. PhD Thesis, Technische Universitt Berlin, Berlin, Germany, 2001.

[27] EKIN, O., HAMMER, P. L., AND KOGAN, A. On connected boolean functions. *Discrete Applied Mathematics 96-97* (1999), 337 – 362.

[28] FISK, S. Geometric coloring theory. *Advances in Mathematics 24*, 3 (1977), 298 – 340.

www.manaraa.com

[29] FRICKE, G., HEDETNIEMI, S. M., HEDETNIEMI, S. T., AND HUTSON, K. R. γ-graphs of graphs. *Discussiones Mathematicae Graph Theory 31*, 3 (2011), 517–531.

[30] GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[31] GAREY, M. R., JOHNSON, D. S., AND TARJAN, R. E. The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing 5*, 4 (1976), 704–714.

[32] GAVRIL, F. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B 16*, 1 (1974), 47 – 56.

[33] GOLUMBIC, M. C. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2004.

[34] GOPALAN, P., KOLAITIS, P. G., MANEVA, E. N., AND PAPADIMITRIOU, C. H. The connectivity of boolean satisfiability: Computational and structural dichotomies. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I* (2006), M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4051 of *Lecture Notes in Computer Science*, Springer, pp. 346–357.

[35] GOPALAN, P., KOLAITIS, P. G., MANEVA, E. N., AND PAPADIMITRIOU, C. H. The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing 38*, 6 (2009), 2330–2355.

[36] HAAS, R., AND SEYFFARTH, K. The k-dominating graph. *Graphs and Combinatorics 30*, 3 (2014), 609–617.

[37] HALE, W. Frequency assignment: theory and applications. vol. 68 of *Proceedings of IEEE*, pp. 1497–1514.

[38] HAMMER, P., MAFFRAY, F., AND PREISSMANN, M. A characterization of chordal bipartite graphs. RUTCOR Research Report 16–89, Rutgers University, New Brunswick NJ, RRR, 1989.

المنارة للاستشارات
www.manaraa.com

[39] HAN, J. Frequency reassignment problem in mobile communication networks. *Computers and Operations Research 34*, 10 (2007), 2939 – 2948.

[40] HEARN, R., AND DEMAINE, E. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science 343*, 1-2 (2005), 72–96.

[41] VAN DEN HEUVEL, J. The complexity of change. *Surveys in Combinatorics, London Mathematical Society Lecture Notes Series 409* (2013).

[42] ITO, T., AND DEMAINE, E. D. Approximability of the subset sum reconfiguration problem. In *Theory and Applications of Models of Computation: 8th Annual Conference, TAMC 2011, Tokyo, Japan, May 23-25, 2011. Proceedings* (Berlin, Heidelberg, 2011), M. Ogihara and J. Tarui, Eds., Springer Berlin Heidelberg, pp. 58–69.

[43] ITO, T., DEMAINE, E. D., HARVEY, N. J. A., PAPADIMITRIOU, C. H., SIDERI, M., UEHARA, R., AND UNO, Y. On the complexity of reconfiguration problems. In *Algorithms and Computation: 19th International Symposium, ISAAC 2008, Gold Coast, Australia, December 15-17, 2008. Proceedings* (Berlin, Heidelberg, 2008), S.-H. Hong, H. Nagamochi, and T. Fukunaga, Eds., Springer Berlin Heidelberg, pp. 28–39.

[44] ITO, T., DEMAINE, E. D., HARVEY, N. J. A., PAPADIMITRIOU, C. H., SIDERI, M., UEHARA, R., AND UNO, Y. On the complexity of reconfiguration problems. *Theoretical Computer Science 412*, 12-14 (2011), 1054–1065.

[45] ITO, T., DEMAINE, E. D., ZHOU, X., AND NISHIZEKI, T. Approximability of partitioning graphs with supply and demand. *Journal of Discrete Algorithms 6*, 4 (2008), 627 – 650. Selected papers from the 1st Algorithms and Complexity in Durham Workshop (ACiD 2005).

[46] ITO, T., KAMINSKI, M., AND DEMAINE, E. D. Reconfiguration of list edge-colorings in a graph. *Discrete Applied Mathematics 160*, 15 (2012), 2199–2207.

[47] ITO, T., KAMINSKI, M., ONO, H., SUZUKI, A., UEHARA, R., AND YAMANAKA, K. On the parameterized complexity for token jumping on graphs. In *Theory and Applications of*

المنارة للاستشارات
www.manaraa.com

Models of Computation: 11th Annual Conference, TAMC 2014, Chennai, India, April 11-13, 2014, Proceedings (2014), T. V. Gopal, M. Agrawal, A. Li, and S. B. Cooper, Eds., Springer International Publishing, pp. 341–351.

[48] ITO, T., KAWAMURA, K., ONO, H., AND ZHOU, X. Reconfiguration of list $L(2,1)$-labelings in a graph. *Theoretical Computer Science 544* (2014), 84 – 97.

[49] JANSSEN, J. Channel assignment and graph labeling. In *Handbook of Wireless Networks and Mobile Computing*. Wiley, 2002, pp. 95–117.

[50] JERRUM, M. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures & Algorithms 7*, 2 (1995), 157–166.

[51] JERRUM, M. *Counting, Sampling and Integrating: Algorithms and Complexity*. Birkhauser Verlag, Basel, 2003.

[52] JOHNSON, M., KRATSCH, D., KRATSCH, S., PATEL, V., AND PAULUSMA, D. Colouring reconfiguration is fixed-parameter tractable. *CoRR abs/1403.6347* (2014).

[53] KAMINSKI, M., MEDVEDEV, P., AND MILANIC, M. Shortest paths between shortest paths. *Theoretical Computer Science 412*, 39 (2011), 5205–5210.

[54] KAMINSKI, M., MEDVEDEV, P., AND MILANIC, M. Complexity of independent set reconfigurability problems. *Theoretical Computer Science 439* (2012), 9–15.

[55] KAREN I. AARDAL, STAN P.M. VAN HOESEL, A. M. K. C. M., AND SASSANO, A. Fap web - a website about frequency assignment problems, 2007. Last accessed: 28 July 2016.

[56] LAS VERGNAS, M., AND MEYNIEL, H. Kempe classes and the hadwiger conjecture. *J. Combinatorial Theory Series B 31* (1981), 95104.

[57] LEESE, R., AND HURLEY, S., Eds. *Methods and Algorithms for Radio Channel Assignment*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, United Kingdom, 2002.

المنارة للاستشارات
www.manaraa.com

[58] MAKINO, K., TAMAKI, S., AND YAMAMOTO, M. On the boolean connectivity problem for horn relations. *Discrete Applied Mathematics 158*, 18 (2010), 2024–2030.

[59] MAKINO, K., TAMAKI, S., AND YAMAMOTO, M. An exact algorithm for the boolean connectivity problem for k-CNF. *Theoretical Computer Science 412*, 35 (2011), 4613 – 4618.

[60] MCDONALD, J., MOHAR, B., AND SCHEIDE, D. Kempe equivalence of edge-colorings. *J. Graph Theory 70*, 2 (May 2012), 226–239.

[61] METZGER, B. H. Spectrum management technique, 1970. Presentation at the 38th National ORSA meeting.

[62] MEYNIEL, H. Les 5-colorations d'un graphe planaire forment une classe de commutation unique. *J. Combinatorial Theory Series B 24*, 3 (1978), 251–257.

[63] MOHAR, B. Kempe equivalence of colorings. In *Graph Theory in Paris, Proceedings of a Conference in Memory of Claude Berge* (Birkhauser, 2006), J.A. Bondy, J. Fonlupt, J.L. Fouquet, J.-C. Fournier, and J. Ramirez Alfonsin (Eds.), pp. 287–297.

[64] MOUAWAD, A. E., NISHIMURA, N., AND RAMAN, V. Vertex cover reconfiguration and beyond. In *Algorithms and Computation: 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings* (2014), H.-K. Ahn and C.-S. Shin, Eds., Springer International Publishing, pp. 452–463.

[65] MOUAWAD, A. E., NISHIMURA, N., RAMAN, V., SIMJOUR, N., AND SUZUKI, A. On the parameterized complexity of reconfiguration problems. In *Parameterized and Exact Computation: 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers* (2013), G. Gutin and S. Szeider, Eds., Springer International Publishing, pp. 281–294.

[66] MOUAWAD, A. E., NISHIMURA, N., RAMAN, V., AND WROCHNA, M. Reconfiguration over tree decompositions. In *Parameterized and Exact Computation: 9th International*

المنارة للاستشارات

www.manaraa.com

Symposium, IPEC 2014, Wroclaw, Poland, September 10-12, 2014. Revised Selected Papers (2014), M. Cygan and P. Heggernes, Eds., Springer International Publishing, pp. 246–257.

[67] PAPADIMITRIOU, C. *Computational Complexity*. Addison-Wesley, Boston, 1994.

[68] PELSMAJER, M. J., TOKAZY, J., AND WEST, D. B. New proofs for strongly chordal graphs and chordal bipartite graphs. Manuscript.

[69] ROBERTS, F. S. T-colorings of graphs: recent results and open problems. *Discrete Mathematics 93*, 2-3 (1991), 229–245.

[70] SAVITCH, W. J. Relationships between nondeterministic and deterministic tape complexities. *J. Computer and System Sciences 4*, 2 (1970), 177–192.

[71] SCHAEFER, T. J. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing* (New York, USA, 1978), STOC '78, ACM, pp. 216–226.

[72] SCHWERDTFEGER, K. W. A computational trichotomy for connectivity of boolean satisfiability. *CoRR abs/1312.4524* (2013).

[73] SCHWERDTFEGER, K. W. The connectivity of boolean satisfiability: No-constants and quantified variants. *CoRR abs/1403.6165* (2014).

[74] SUZUKI, A., MOUAWAD, A. E., AND NISHIMURA, N. Reconfiguration of dominating sets. *CoRR abs/1401.5714* (2014).

[75] UEHARA, R. Linear time algorithms on chordal bipartite and strongly chordal graphs. In *ICALP* (2002), P. Widmayer, F. T. Ruiz, R. M. Bueno, M. Hennessy, S. Eidenbenz, and R. Conejo, Eds., vol. 2380 of *Lecture Notes in Computer Science*, Springer, pp. 993–1004.

[76] VAN DEN HEUVEL, J., LEESE, R. A., AND SHEPHERD, M. A. Graph labeling and radio channel assignment. *Journal of Graph Theory 29*, 4 (1998), 263–283.

[77] VIKAS, N. Computational complexity of compaction to irreflexive cycles. *J. Computer and System Sciences 68*, 3 (2004), 473–496.

www.manaraa.com

[78] WROCHNA, M. Reconfiguration in bounded bandwidth and treedepth. *CoRR abs/1405.0847* (2014).

المنارة للاستشارات

www.manaraa.com