Link Scheduling Directed Graphs Using Undirected Edge Colours

by

Kyle d’Oliveira

B. Sc., Vancouver Island University, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

May 2012

© Kyle d’Oliveira, 2012
Abstract

Communication in a wireless network is typically modeled as either a directed or an undirected communication graph. Link scheduling is the problem of assigning time slots to each edge so that communication on every edge at its assigned time may occur without interference. Interference is also modeled as a graph, typically the same as the communication graph. However, interference may occur even without the desire or possibility of communication and warrants its own graph.

When link scheduling directed graphs, Packet Radio Network (PRN)-colouring is used to create a link schedule. When modeled as an undirected graph, strong edge colouring is typically used to create a link schedule. Both PRN and strong edge colouring have been shown to be NP-complete. In this thesis, two new types of undirected edge colourings are analyzed: Acyclic Colour Induced (ACI) and Even Cycle Parity Colour Induced (ECPCI) edge colouring. Both ACI and ECPCI edge colouring are less restrictive than strong edge colouring in the sense that a strong edge colouring provides both an ACI and an ECPCI edge colouring; in fact, it is possible to use significantly fewer colours. Further, there is a translation that will take an ACI or an ECPCI edge colouring of an undirected graph and transform it into a directed PRN-colouring of the corresponding symmetric directed graph.
These two new types of edge colourings are shown to be NP-complete.

Lower bounds for ACI and ECPCI edge colouring planar graphs are analyzed. It is shown that there exists planar graphs with maximum degree $\Delta$ that require \(\frac{33\Delta}{16}\) (respectively $2\Delta + 1$) colours in an ACI (respectively ECPCI) edge colouring. Also, algorithms are given for strong, ACI, and ECPCI edge colouring that take into account a communication graph that is a subgraph of the interference graph. Using these algorithms, a link schedule using PRN-colouring on directed communication and interference graphs can be created that is an $O(1)$ and $O(\sqrt{\gamma})$ approximation for planar and general graphs with genus $\gamma \geq 1$ respectively. The approximation algorithm given improves the best known upper bound for link scheduling planar graphs.
Table of Contents

Abstract ................................................................. ii

Table of Contents ................................................... iv

List of Tables ........................................................ vi

List of Figures ........................................................ vii

Glossary ................................................................. viii

Acknowledgments ...................................................... ix

Dedication ............................................................. x

1 Introduction ......................................................... 1

  1.1 Interference Models ........................................... 3

2 Preliminaries ....................................................... 5

3 Previous Work ....................................................... 15

4 NP-hardness of ACI and ECPCI Edge Colouring ............... 18
4.1 ECPCI Edge Colouring Reduction .................. 18
4.2 ACI Edge Colouring Reduction .................... 21

5 Link Scheduling ........................................... 24
5.1 Lower Bounds ........................................... 24
5.2 Upper Bounds .......................................... 33

6 Conclusions and Future Work ......................... 45

Bibliography ............................................. 48
List of Tables

Table 5.1  Algorithm 1 summary on an undirected graphs $I$ and $G \subseteq I$ . . 44
Table 5.2  Algorithm 2 summary on an undirected graphs $I$ and $G \subseteq I$ . . 44
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A sample network</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Some colour induced graphs</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Examples of different edge colouring types</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>A modified 4 cycle</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>A gadget for ECPCI edge coloring</td>
<td>20</td>
</tr>
<tr>
<td>4.3</td>
<td>A partial gadget for ACI edge colouring</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>A planar graph that requires $4\Delta - 6$ colours to strong edge colour</td>
<td>26</td>
</tr>
<tr>
<td>5.2</td>
<td>The labeled vertices of the graph $G$</td>
<td>27</td>
</tr>
<tr>
<td>5.3</td>
<td>The graph $G^*$</td>
<td>30</td>
</tr>
<tr>
<td>5.4</td>
<td>A planar graph requiring $2\Delta + 1$ colours to ECPCI edge colour.</td>
<td>33</td>
</tr>
<tr>
<td>5.5</td>
<td>Matched edges that can cause a multi edge</td>
<td>40</td>
</tr>
</tbody>
</table>
Glossary

TDMA  Time Division Multiple Access

CDMA  Code Division Multiple Access

FDMA  Frequency Division Multiple Access

PRN   Packet Radio Network

SINR  Signal to Interference and Noise Ratio

RTS   Request To Send

CTS   Clear to Send

ACI   Acyclic Colour Induced

ECPCI Even Cycle Parity Colour Induced
Acknowledgments

I’d like to thank my supervisor, Dr. David Kirkpatrick, for his continuous support and insight. Without his intuition and pushing in research areas, I would have been lost. Further, his assistance in writing the thesis was invaluable.

I am grateful for my undergraduate thesis supervisor, Gara Pruesse. I would not have been able to write this thesis without the skills she taught me.

I’d also like to thank my good friend Jake Ackeral for helping me proof read and organize my arguments. His interest and knowledge helped clarify my thoughts and proofs.

Finally, I’m especially grateful for my loving fiancé Rose and family for constant support and encouragement. They have given me the motivation to constantly pushing forward. It is to them that I dedicate this thesis.
To Rose and my parents
Chapter 1

Introduction

A wireless network is composed of a collection of stations that each have a transmitter and a receiver. The location of these stations can be either static and unmoving, or the location can be changing dynamically. Each station has a power supply that will determine which other stations it can transmit data to. Being within transmission range is a necessary but not sufficient condition for successful communication. Not all the stations can be active and transmitting simultaneously. The transmission signal sent from a station can cause collisions with data at an unintended receiving station; these collisions can cause data interference and force the need to resend data. Link scheduling is a problem that aims to create an interference free schedule in order to allow every station to communicate with their respective neighbours.

The power source associated with the transmitters is often limited, thus, having a set schedule to determine when a station can communicate with a neighbour is very useful. This schedule would allow stations to lower, or even turn off, the
power supply to the transmitter or receiver when it is known that they are not in use. Such a scheme will also help reduce wasted energy in the network. Further, because the power source is limited, messages need to be conveyed between neighbours rather than sent directly to a centralized receiver or to another station out of direct communication range. If a message is bound for a specific location in the network, it will need to make multiple 'hops' as it travels from station to station along a path through the network.

In wired networks, collisions can be immediately detected. Protocols exist that force the transmitter to 'back off' for a period of time when a collision occurs. However, in a wireless network, collisions are not readily detectable. In this sense, it is better to use a protocol that divides the communication channel up to prevent collisions. There already exist wireless protocols such as Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), and Frequency Division Multiple Access (FDMA) that divide the communication channel up into multiple channels [33].

In TDMA, a time frame is created that consists of multiple time slots. Links are scheduled to communicate within a time slot, and the time frame is repeated ad infinitum to create a schedule. A smaller time frame will increase the throughput in a network. The smaller frame does not save time in CDMA or FDMA, instead it reduces the number of different channels required. For the purposes of creating link schedules, as long as the protocol is slotted and synchronous they are effectively equivalent.

Each of the wireless stations is assumed to only be able to broadcast data for a single recipient at any given time. It is also assumed that the stations can only receive data while not transmitting and they cannot receive multiple data streams.
simultaneously. The last constraint does not necessarily mean that a station cannot be in range of multiple data streams simultaneously but that the stations cannot process multiple transmissions at once. A station being in range of multiple transmissions simultaneously, however, can create interference.

1.1 Interference Models

*Protocol Interference Model* In this model, for a given communication link between two stations, the stations that can interfere with it can be viewed as a property of the network itself. The idea is simply that an active station can only receive one transmission at a time. Interference is created if multiple transmissions are received—the intended transmission is lost or corrupted due to the second, unintended, transmission and will need to be retransmitted. Watanabe et al. [36] gave a generalized version of protocol interference where stations are able to receive $k$ transmissions before interference occurs. The model can also be extended to allow stations to have a power supply for the transmitter that can dynamically alter the power levels. If a station can cause interference when transmitting at one power level then it could scale down its transmitter power level and no longer cause interference. However, the power level would still need to be sufficient for the data to reach its intended destination.

*RTS/CTS Model:* As mentioned in Wang et al. [35], this model does not describe interference but it describes a protocol for how stations transmit data. In order for a station to communicate with one of its neighbours, it first must broadcast a Request To Send (RTS). If the receiver is permitted to receive such a transmission it
will broadcast back a Clear to Send (CTS). The data will then be transmitted along the communication link along with acknowledgments from the receiving station. Before and during the data transmission, both stations are actively transmitting, although one is merely acknowledging. Under the protocol interference model, this adds the additional restriction that if a station is scheduled to transmit (respectively receive), no station, other than the intended receiver (respectively transmitter), can be active within its interference range.

**Physical Interference Model**  This model is significantly different from the protocol interference model. A station that is trying to receive data can tolerate some transmissions from other stations but what is important is that the intended receiving signal is stronger than outside interference. In order to have a successful communication along a link, the transmission signal must have the Signal to Interference and Noise Ratio ($\text{SINR}$) exceed some fixed value. All communication links that are set to be active in a time slot must be successful in order for the time slot to be interference free. As in protocol interference, dynamic power levels can be used within this model.
Chapter 2

Preliminaries

Consider the small wireless sensor network that consists of 8 stations represented by the graph in figure 2.1a. This directed graph consists of directed edges \((x \rightarrow y)\) only if station \(x\) can communicate with station \(y\). Each directed edge represents a communication link between two stations. Due to the broadcast nature of wireless networks, however, some stations may interfere with communication outside of those in figure 2.1a. If station \(a\) is transmitting then stations \(b, d\) and \(e\) will receive it as shown in figure 2.1b. This will, under the fixed power protocol interference model, create interference if station \(e\) was trying to receive a transmission from station \(h\) while station \(a\) is active and transmitting. Finding a schedule to satisfy all of the communication links while being aware of interference is called link scheduling. Link scheduling will be formally defined later in this chapter. Effectively, the communication links need to be partitioned into sets where all of the links within a set can be active simultaneously.

Note the difference between scheduling the communication links to be active
and scheduling the stations to be active. Scheduling when stations are set to be active is called \textit{broadcast scheduling}. In broadcast scheduling, each station intends to broadcast data to all of its neighbours at once. The major distinction from link scheduling is that, although each station still broadcasts data, the message is intended for a specific neighbour only.

It is natural to model networks as directed graphs. Each vertex in the graph represents a station and a directed edge exists between two vertices $i$ and $j$ if station $i$ can transmit data to station $j$. Recall, a station can cause interference with another station even if they cannot communicate with each other. A second directed graph may be used to model the interference network. The directed graphs may have uni-directional edges which would mean that station $x$ can communicate (or interfere) to station $y$ but the reverse is not true.

The edges of the communication graph need to be partitioned so that each set obeys certain independence constraints. This can be viewed as the edges of the communication graph being coloured with all edges of a certain colour obeying the constraints. Each colour will be associated with a time slot and together they will create a time frame. When an edge $(x \to y)$ is given a particular colour, this will give station $x$ broadcast rights and station $y$ receiving rights within the associated
time slot. A legal time frame has every edge of the communication graph coloured and, when all the transmitting and receiving stations of a given colour are active, there is no interference. The goal of link scheduling is to optimize the size of the time frame which requires the number of time slots to be minimized. Since each time slot is associated with a single colour, it is equivalent to minimize the number of colours used to colour the edges of the communication graph.

The interference range for a transmission is typically larger than the communication range [35]. In practice, the interference range is roughly between two and four times the communication range. It is then reasonable to assume that the communication graph is a proper subgraph of the interference graph. Nevertheless, although much attention has been given to working with protocol level interference [4, 22, 23, 27], it has, for simplicity, been assumed the communication graph is the same as the interference graph.

When creating a link schedule for a directed graph, we can apply the same link schedule that is produced on the symmetric version of the same graph. If there is a communication link \((a \rightarrow b)\) in the symmetric graph but, in fact, \(a\) cannot transmit to \(b\), then this link can simply be ignored. At the worst, the size of this link schedule will be twice as big as it needs to be, thus, it is reasonable to assume the communication and interference graphs are symmetric. Communication graphs have also often been viewed as undirected graphs [5, 7, 16, 20, 22, 32].

**Notation.** Let \(G\) be a symmetric directed graph, and \(H\) be the undirected version of \(G\). The *edge* (respectively *vertex*) *set* of a graph \(G\) will be denoted as \(E(G)\) (respectively \(V(G)\)). The *degree* of a vertex is the number of edges (including both incoming and outgoing edges in the directed case) that are incident to it. The
maximum degree of $G$ (respectively $H$) will be denoted $\rho_G$ (respectively $\Delta_H$). Note that since $G$ is symmetric, $2\Delta_H = \rho_G$.

The *genus* of a graph $G$ is the minimum number of handles that must be added to the plane to embed $G$ without any crossings and will be denoted by $\gamma_G$. The *thickness* of a graph $G$, denoted $\theta_G$, is the smallest number of planar graphs the edges of $G$ can be partitioned into. A graph $G$ is *c inductive* (also known as *c degenerate*) if $G$ has at most $c$ vertices or there exists a vertex $u$ with degree at most $c$ such that $G - u$ is also $c$ inductive. It is immediate from Euler’s formula that planar graphs are 5-inductive. More generally, it is known that every graph with thickness $\theta$ is $6\theta - 1$ inductive [28]. The *vertex* (also known as *point*) *arboricity* of a graph is the smallest number of sets its vertices can be partitioned into such that each vertex set induces an acyclic graph. The *edge arboricity* of a graph is the smallest number of sets its edges can be partitioned into such that each edge set induces an acyclic graph. The *girth* of a graph $G$ is the length of the smallest cycle in $G$. If it is clear which graph is being referred to then the subscripts will be dropped. For additional graph notation and terminology, see *Graph Theory with Applications* [9].

**Definition.** Let $I$ be a directed symmetric interference graph and $G$ be a directed symmetric communication graph that is a subgraph of the interference graph, denoted $G \subseteq I$. A *link schedule* of size $k$ under protocol level interference is a partition of $E(G)$ into $k$ sets (also known as time slots) $\{S_1, \ldots, S_k\}$, such that if $(a \rightarrow b)$ and $(c \rightarrow d) \in S_i$, then $a, b, c, d$ must all be distinct and the edges $(a \rightarrow d)$ and $(c \rightarrow b)$ must not be in $E(I)$. An optimal link schedule is a schedule with minimum $k$.

It is interesting to note that this definition still uses both of the communica-
tion and interference graphs rather than assuming that they coincide. In the link schedule partitioning of edges, each set of edges can be viewed as edges of a fixed colour class. The associated directed edge colouring is known as Packet Radio Network (PRN)-colouring [27][32]. It was shown by Gandham et al. as part of theorem 4.6 [16] that undirected edge colouring can also be used to model PRN-colouring, although they do not label it as such. We will find PRN-colourings by edge colouring the undirected version of the graphs then translating the undirected edge colouring into a PRN-colouring.

Finding a link schedule that minimizes the number of time slots using PRN-colouring is NP-hard [27]. This remains the case even if the interference graph (and thus the communication graph) is planar. The translations to PRN-colouring from the undirected edge colourings (which will be given later) will only increase the number of colours, and thus time slots, by a constant factor (and in fact only by a factor of 2). Thus, undirected edge colouring is still useful for finding approximation algorithms for the optimal link schedule. In this thesis, three types of undirected edge colourings will be analyzed: strong edge colouring, ACI edge colouring and ECPCI edge colouring. This is also the first time ACI edge colouring and ECPCI edge colouring have been examined outside of the work of Gandham et al. [16].

Definition. Edge Colouring

Let G be an undirected graph. An edge colouring with k colours is a partition of $E(G)$ into sets (also known as colour classes) such that no two edges in the same colour class are incident to the same vertex.
**Definition. Vertex Colouring**

Let $G$ be an undirected graph. A *vertex colouring* with $k$ colours is a partition of $V(G)$ into colour classes such that no two vertices in the same colour class are adjacent.

**Definition. Colour Induced Graph**

Let $G$ be a graph with its edges coloured in some fashion. Let $S$ be a set of edges in $G$ of a given colour class. Let $V_S$ be all the vertices incident to some edge in $S$. The graph induced on $G$ by the vertex set $V_S$ is the *$S$ colour induced graph* and will be denoted as $G[V_S]$. Note that, although $S$ is a set of edges, the $S$ colour induced graph is not induced on the edges but rather on the vertices that are incident to those edges.

Figure 2.2 is a collection of 4 examples of $S$ colour induced graphs where the edges in the set $S$ are the bold edges.

![Figure 2.2: Some colour induced graphs](image-url)
Definition. Strong Edge Colouring

Let $G$ be an undirected graph. A strong edge colouring (also known as a distance 2 edge colouring) with $k$ colours is a partition of $E(G)$ into colour classes $\{S_1, \ldots, S_k\}$ such that each colour class is an edge colouring and if $(a, b)$ and $(c, d)$ are the same colour then edges $(a, c), (a, d), (b, c), (b, d) \notin E(G)$. In other words, $G[V_{S_i}]$ is an induced matching or $G[V_{S_i}]$ has a maximum degree of one for $1 \leq i \leq k$.

Strong edge colouring is a natural way to move from the directed PRN-colouring to an undirected edge colouring; it even allows the corresponding network to follow the RTS/CTS protocol. Link scheduling using strong edge colouring has been previously studied [5, 7, 32] and will be discussed in the next chapter.

The translation from a strong edge colouring to a PRN-colouring is quite simple. For a given colour class $S_i$ from a strong edge colouring, two PRN-colour classes can be created $S_0^i, S_1^i$. If $(a, b) \in S_i$ then $(a \to b) \in S_0^i$ and $(b \to a) \in S_1^i$. It is clear that both $S_0^i$ and $S_1^i$ satisfy the PRN-colouring constraints since all of the edges came from a strong edge colouring.

Definition. Acyclic Colour Induced (ACI) Edge Colouring

Let $G$ be an undirected graph. An ACI edge colouring with $k$ colours is a partition of $E(G)$ into colour classes $\{S_1, \ldots, S_k\}$ such that each colour class is an edge colouring and that $G[V_{S_i}]$ is acyclic for $1 \leq i \leq k$.

ACI edge colouring is not a trivial extension of edge colouring. since each colour induced graph is induced on the vertices of an edge colour class, it is possible to have cycles as in figure 2.2. Thus, the graphs are not trivially acyclic. The name is given because each colour induced graph is acyclic, but this should not be confused with acyclic edge colouring. Acyclic edge colouring [2] is the restriction
where no cycle in the graph is bichromatic. If no cycle is bichromatic, the graph induced on any edges of any two colours will be acyclic.

ACI edge colouring is a relaxation of strong edge colouring such that better schedules are possible. The translation to a PRN-colouring, however, is not as straightforward. Given an ACI edge colouring of $G$, the vertices of the colour induced graphs, for each colour class, need to be given a binary assignment which is then used to create a PRN-colour class.

Essentially, each vertex of an $S$ colour induced graph, $H = G[V_S]$, must be marked either as transmitting or receiving. If a vertex is marked as transmitting, its intended receiver must be marked as receiving and all of the stations it can cause interference with must be marked as transmitting. Let $F$ be a simple binary function $F : V(H) \to \{0, 1\}$. To create a mapping, one first assigns an arbitrary vertex of each component of $H$ to 0. Afterwards, the mapping propagates through the entire component following the rule: if $\exists (a, b) \in E(H), F(a) = F(b)$ if and only if $(a, b) \notin S$. Since $H$ is acyclic, such a mapping will always exist. Due to theorem 4.5 from Gandham et al. [16], $S$ can be split into two PRN-colouring sets $S^0$ and $S^1$ by taking every undirected edge $(a, b) \in S$, and putting the directed edge $(a \to b)$ into $S^{F(a)}$ and the directed edge $(b \to a)$ into $S^{F(b)}$.

**Definition.** $S$-parity Let $G$ be a graph with its edges coloured in some fashion. Let $S$ be a set of edges in $G$ of a given colour class. A cycle in a $G[V_S]$ induced graph has even $S$-parity if and only if the number of edges of $S$ in the cycle is even.

**Definition.** Even Cycle Parity Colour Induced (ECPCI) Edge Colouring

Let $G$ be an undirected graph. An ECPCI edge colouring with $k$ colours is a partition of $E(G)$ into colour classes $\{S_1, \ldots, S_k\}$ such that each colour class is an
edge colouring and every cycle in $G[V_S]$ has even $S_i$-parity for $1 \leq i \leq k$.

ECPCI edge colouring was studied but not named by Gandham et al. [16]. An $S$ colour induced graph from an ECPCI edge colouring can still contain cycles, but cannot contain cycles with an odd number of edges in $S$. Thus, an ACI edge colouring is an ECPCI edge colouring but not vice versa.

To help illustrate what exactly is meant with ECPCI edge colouring consider the bold edge colour induced graphs in figure 2.2. Graph (a) contains no cycles, so the bold edges are a valid ECPCI edge colour class. Graph (b) has no cycle with odd parity, even though it contains two cycles: 234 and 4568. The cycle 234 is a cycle with odd length, but it contains an even number of bold edges (zero) and the cycle 4568 also has an even number of bold edges. The bold edges of graph (c) constitute a valid ECPCI edge colour class because it has only one cycle 13784 and that cycle has two bold edges. In graph (d), the bold edges do not form valid ECPCI colour class. The graph contains two cycles that have odd parity. Firstly, the odd length cycle 256 contains only one bold edge. Secondly, the even length cycle 124873 contains three bold edges.

The translation from ECPCI edge colouring to PRN-colouring is exactly the same as the translation from ACI edge colouring. From theorem 4.3 Gandham et al. [16], which is restated below to match the notation, such a mapping will always exist.

**Theorem** (4.3 Gandham et al.). Let $S$ be a colour class from an edge colouring of a graph $G$. Let $H = G[V_S]$ and let $F$ be a mapping function $F : V(H) \to \{0, 1\}$. $H$ contains no cycle with odd $S$-parity if and only if exists a valid mapping satisfying: if $\exists (a, b) \in E(H)$ then $F(a) = F(b)$ if and only if $(a, b) \not\in S$. 

13
Figure 2.3 is a collection of examples of different types of undirected edge colouring. As previously noted, there is a clear hierarchy of the edge colouring types. Strong edge colouring being the most restrictive, then ACI, ECPCI and finally normal edge colouring. This means that every strong edge colouring is an ACI edge colouring, every ACI edge colouring is an ECPCI edge colouring, and every ECPCI edge colouring is an edge colouring. Therefore, any upper bound established applies to any less restrictive edge colouring type. Similarly, any lower bound established applies to any more restrictive edge colouring type.

**Figure 2.3: Examples of different edge colouring types**
Chapter 3

Previous Work

Lloyd and Ramanathan [23, 27] showed that link scheduling, under protocol interference, using PRN-colouring is NP-complete on general graphs. They also showed that it remains NP-complete on planar graphs. Any graph $G$ will require at least $\rho_G$ colours to PRN-colour. Using that lower bound, an approximation algorithm was given with approximation ratio $\theta_G$ (thickness) for PRN-colouring when run on a graph $G$. The algorithm always produces a PRN-colouring with at most \((4(6\theta - 1))\rho_G - 2(6\theta - 1)^2 + 1\) colours. When the directed graph is symmetric and planar ($\theta = 1$) the bound on the number of colours produced improves to $11\rho_G - 49$. The approach taken here was an inductive approach relying on the fact that $G$ is $6\theta - 1$ inductive.

It was shown by Mahdian [25] that strong edge colouring a graph is NP-complete and remains NP-complete for bipartite graphs of any girth. Barret et al. [7] examined link scheduling on undirected graphs using strong edge colouring to create a schedule. For $c$-inductive graphs with maximum degree $\Delta$, they gave
a polynomial time approximation algorithm that produces \((4c - 3)\Delta - 2c + 3\) time slots. The inductive approach used was very similar to the approach used by Lloyd and Ramanathan. They also gave a centralized and a distributed constant approximation algorithm for unit disk graphs using at most \(8OPT + 1\) colours where \(OPT\) is the number of colours in an optimum strong edge colouring.

Gandham et al. [16] also looked at link scheduling undirected graphs under protocol interference. They introduced the concept of what we have called ECPCI edge colouring and the translation to take an edge colouring of a undirected graph and make it a PRN-colouring of a directed graph. They gave a distributed algorithm to edge colour a graph with at most \(\Delta + 1\) colours and assign a binary mapping to the vertices of each colour induced graph. Their algorithm will always give a \(\Delta + 1\) sized link schedule for graphs that are acyclic. For the equivalent directed symmetric graph, their algorithm would produce a link schedule of size \(\rho + 2\). An edge colouring on a directed graph will always require at least \(\rho\) colours so this is not far from the optimal link schedule size.

Sen and Huson [32] examined link scheduling using strong edge colouring on generic disk graphs constructed on a two dimensional plane (planar point graphs). They showed that the problem remains NP-complete. They also gave an optimal \(O(n \log n)\) time solution when all of the \(n\) stations are located on a line.

In a strong edge colouring each colour class is a distance 2 independent edge set. Rather than trying to find a complete schedule for all of the links, Balakrishnan et al. [5] looked at the related problem of finding the largest distance 2 independent edge set for finding schedules. They modeled the graphs as generic disk graphs. They show that the problem remains NP-complete even on graphs that are both unit disk and planar. They also gave a constant approximation algorithm for general
disk graphs and for unit disk graphs, however, they did not explicitly give how large the constant is.

Wang et al. [35] point out that some links in the network need to be active more often than others. They proposed constant approximation algorithms, both centralized and distributed, for a weighted link scheduling problem on general graphs under the fixed power protocol interference model. The weighted link scheduling problem is a generalized version of link scheduling where, within a time frame, every communication link needs to be active a number of times corresponding to the weight of that communication link. The unweighted case is when all the weights on the communication links are one.

Kumar et al. examine a problem similar to link scheduling. Rather than needing to satisfy the demands of every communication link, a set of packets and their routes through the network are given. The problem is to schedule the packet routes in the smallest schedule. They give an approximation algorithm of $O(\Delta \log^2 n)$ on graphs with $n$ vertices and show that it is NP-hard to approximate within a factor of $\Delta^{1-\epsilon}$ for any $\epsilon > 0$. They also give a constant centralized approximation algorithm and a $O(\log n)$ distributed approximation algorithm for unit disk graphs.

Lv et al. [24] focused on dealing with link scheduling under physical interference. Recall, this means the interference is not automatically present if a station receives a transmission not meant for itself, but rather, a SINR must be satisfied to receive any transmission. They show that the link scheduling problem, even under this model, remains NP-complete.
Chapter 4

NP-hardness of ACI and ECPCI Edge Colouring

ECPCI edge colouring was introduced by Gandham et al. [16] as a method to create a link schedule on undirected graphs. The colouring constraint is specifically tailored to be able to translate the colouring into a directed PRN-colouring. A natural way to relax the colouring constraint is to, instead, have every colour induced graph be acyclic. This gives rise to the concept of ACI edge colouring and it is first studied in this thesis. We show that both ACI and ECPCI edge colouring are NP-complete. It is unknown, however, if it is NP-hard to approximate either ACI or ECPCI edge colouring within a certain threshold.

4.1 ECPCI Edge Colouring Reduction

It is known that vertex colouring 4 regular planar graphs with 3 colours is an NP-complete problem [13]. This problem will be reduced to the problem of ECPCI
edge colouring a graph with 3 colours.

![Graph Image]

**Figure 4.1:** A modified 4 cycle

**Lemma 1.** Let \( G \) be the graph in figure 4.1. Any ECPCI edge colouring of \( G \) with 3 colours must have the edges \((a,e),(b,f),(c,g),(d,h)\) the same colour.

**Proof.** Let \( x,y,z \) be the three colours used in the ECPCI edge colouring. Suppose, without loss of generality, the edges \((a,e)\) and \((b,f)\) are coloured different colours. Assume that \((a,e)\) is coloured \( x \) and \((b,f)\) is coloured \( y \). Then the edges \((a,b)\), \((b,c)\), and \((a,d)\) must be coloured \( z \), \( x \), and \( y \) respectively. This implies that \((c,d)\), \((d,h)\) and \((c,d)\) must be coloured \( z \), \( x \) and \( y \) respectively. This however is a contradiction since there is now a cycle \( abcd \) in the \( x \) colour induced graph with only one \( x \) coloured edge.

Therefore, all of the edges \((a,e),(b,f),(c,g),(d,h)\) must be coloured the same colour. \( \square \)

**Theorem 1.** ECPCI edge colouring a planar graph with maximum degree of 3 using 3 colours is NP-Complete

**Proof.** Given a 4 regular planar graph \( H \), a new graph \( G \) will be created. For each vertex \( v \) in \( H \), add a copy of the gadget \( v' \) shown in figure 4.2 to \( G \). In a gadget \( v' \), call the vertices \( O_i \) the *out vertices* of \( v' \), and call the edges incident to any \( O_i \) the
out edges of $v'$. If two vertices $u$ and $v$ are adjacent in $H$, then, in $G$, identify an out vertex of $u'$ and an out vertex of $v'$.

![Figure 4.2: A gadget for ECPCI edge coloring](image)

Suppose that there exists an ECPCI edge colouring of $G$ with 3 colours. Note, the gadget is made up of many modified 4 cycles that are shown in figure 4.1. By lemma 1, all of the out edges of a gadget must be the same colour. For every vertex $v$ in $H$, it can be coloured with the colour of the out edges of $v'$ from $G$. If two vertices of $H$ are adjacent to one another, then, in $G$, there is an out vertex shared between the two corresponding gadgets. Since the vertex is shared, the out edges of the two adjacent gadgets must be different. Thus, there is a vertex colouring of $H$ with 3 colours.

Suppose that there is a vertex colouring of $H$ with 3 colours. If vertex $v$ of $H$ is coloured $x$, with $y$ and $z$ being the two remaining colours, colour the edges of $v'$
as shown in figure 4.2. In the gadget itself, there are no cycles that contain an odd number of edges of any colour. Further, every path between two out vertices of the gadget passes through an even number of $x$ edges, an even number of $y$ edges and an even number of $z$ edges. This implies that in any $c$ colour induced graph, every cycle has an even $c$-parity. Thus, there is an ECPCI edge colouring of $G$ with 3 colours.

There exists a vertex colouring of $H$ with 3 colours if and only if there exist an ECPCI edge colouring of $G$ with 3 colours. This means that ECPCI edge colouring is NP-hard. It is clear that ECPCI edge colouring is in NP, thus, ECPCI edge colouring is NP-complete.

\[\square\]

### 4.2 ACI Edge Colouring Reduction

Vertex colouring a graph with a fixed number of colours $k$ is NP-complete [18] and a reduction to ACI edge colouring a graph with $k$ colours will be shown. Given some integer $k \geq 15$, a gadget $G^k$ will be needed for the reduction. Let $G^k$ be the graph in figure 4.3 with $k - 6$ additional vertices adjacent to each $P_i$. Denote the vertices $O_1, O_2, O_3$ as the out vertices and the edges $(O_1, I_1), (O_2, I_2), (O_3, I_3)$ as the out edges.

**Lemma 2.** In any ACI edge colouring of $G^k$ with $k$ colours, the out edges must all be coloured the same colour.

**Proof.** Firstly, any ACI edge colouring $G^k$ must use at least 15 distinct colours since every pair of edges $(w, x), (y, z)$ in $G^k$, where $w, x, y, z \in \{I_1, I_2, I_3, P_1, P_2, P_3, O_3\}$, either share an endpoint or the vertices of the edges will induce a cycle. Also, note
that all the vertices $P_i$ have degree $k - 1$; there is exactly 1 colour that is free at each of those vertices.

Any colour on an out edge must be free on at least two of $P_1, P_2$ or $P_3$. Since each of $P_1, P_2$ and $P_3$ has exactly one free colour, there cannot be more than one colour used on the out edges.

\[ \square \]

**Theorem 2.** AC1 edge colouring a graph with $k$ colours, where $k \geq 15$ is NP-Complete

**Proof.** It is clear to see that AC1 edge colouring is in NP.

Given a graph $H$ and an integer $k \geq 15$, a new graph $G$ will be created. For each
vertex \( v \) with degree \( d \) in \( H \), add a new gadget \( v' \) to \( G \). \( v' \) consists of \( d - 2 \) copies \( G_1^k, \ldots, G_{d-2}^k \) of the gadget \( G^k \) described in lemma 2 with an out edge of \( G_i^k \) being an out edge in \( G_{i+1}^k \) for \( 1 \leq i \leq d - 2 \). Note, there will be \( d \) out edges incident to an out vertex of degree one in gadget \( v' \). If two vertices \( u, v \in V(H) \) are adjacent then identify a degree one out vertex of \( u' \) with a degree one out vertex of \( v' \) in \( G \).

Suppose there is a ACI edge colouring of \( G \) with \( k \) colours. From lemma 2, the out edges of a gadget \( v' \) must all be coloured the same colour. Colour all vertices \( v \) in \( H \) the same colour as the out edges of the corresponding gadget \( v' \) in \( G \). If two vertices \( u, v \) are adjacent in \( H \), then the out edges of \( u' \) and \( v' \) must be coloured differently since they share an out vertex. Therefore, there is a vertex colouring of \( H \) with \( k \) colours.

Suppose there is a vertex colouring of \( H \) with \( k \) colours. If a vertex \( v \) is coloured \( c \), then the out edges of corresponding gadget \( v' \) can also be coloured \( c \). If vertices \( u \) and \( v \) are adjacent in \( H \) with the joined out edges of \( u' \) and \( v' \) being \((x, O)\) and \((O, y)\), then the four edges incident to \( y \) cannot be the same colour as the edge \((x, O)\). Any one of the edges \((p_1, p_2), (p_2, p_3)\) or \((p_1, p_3)\) of \( v' \) can be coloured the same colour as \((x, O)\). The remaining edges in \( v' \) can be ACI edge coloured using the remaining \( k - 1 \) colours in a greedy fashion. In any colour induced graph, there are no cycles contained inside any gadget and any two gadgets will be in distinct components of the induced graph. Thus, there is an ACI edge colouring of \( G \) with \( k \) colours.

Since there exists a vertex colouring of \( H \) with \( k \) colours if and only if there exist ACI edge colouring of \( G \) with \( k \) colours, ACI edge colouring a graph with \( k \) colours is NP-complete. This can also be phrased as ACI edge colouring a graph with \( \Delta + 1 \) colours is NP-complete for \( \Delta \geq 14 \).
Chapter 5

Link Scheduling

5.1 Lower Bounds

Strong Edge Colouring

Recall, in strong edge colouring, if both \((a \rightarrow b)\) and \((c \rightarrow d)\) are the same colour then none of the of the edges \((a, c),(a, d),(b, c)\) or \((b, d)\) will appear in the interference graph. Suppose that there are two communication links \((a \rightarrow b)\) and \((c \rightarrow d)\) both set to be active in a given time slot. If there was an edge, such as \((b, c)\), in the interference graph then we know that if \(c\) is transmitting with the intended recipient of \(d\), that \(b\) must not be receiving. This imposes a level of coordination between the coloured edges on which will send and which will receive. Each node must be aware of when it is allowed to transmit and when it is allowed to receive. However, if there is no edge that joins \((a, b)\) and \((c, d)\) in the interference graph then those active nodes are free to send or receive at any point in the time slot.

The freedom, given by using strong edge colouring, could be useful if the
communication channels can have simultaneous bidirectional communication and would effectively reduce the number of time slots used by half. Another benefit is that the RTS/CTS protocol model can be followed. Allowing acknowledgments to be sent at any time instead of waiting for the designated time slot will help ensure that the data arrives correctly.

A trivial lower bound on strong edge colouring for undirected planar graphs with maximum degree $\Delta$ is $4\Delta - 6$. This is easily achieved by looking at a complete graph on 4 vertices with each vertex having $\Delta - 3$ additional neighbours. Figure 5.1 illustrates such a graph with $\Delta = 6$. Every pair of edges is either incident to the same vertex or has an edge that joins them.

It is claimed by Mahdian [25], there are planar graphs that require $4\Delta - 4$ colours to strong edge colour. It is unknown if there are planar graphs that require more than $4\Delta - 4$ colours. For general graphs, and, in particular any complete graph on $n$ vertices, it is clear that a strong edge colouring requires $\frac{n\Delta}{2}$ colours (1 for every edge). Further, if a similar construction to figure 5.1 is used, a general graph containing an $n$ clique can be constructed that, when strong edge coloured, will require $n\Delta - \frac{(n)(n-1)}{2}$ colours. It is interesting to note that any graph containing an $n$ clique will also have thickness $\theta$ at least $\left\lfloor \frac{n+7}{6} \right\rfloor$ [26] and genus $\gamma$ of at least $\left\lceil \frac{(n-3)(n-4)}{12} \right\rceil$ [29].

**ACI Edge Colouring**

A graph with $m$ edges, in the worst case can require as many as $m$ colours to ACI edge colour. For example, in a complete graph, the vertices of any pair of disjoint edges will induce a cycle which means that each edge must be coloured a distinct
**Figure 5.1:** A planar graph that requires $4\Delta - 6$ colours to strong edge colour

colour. We will show that a lower bound on the number of colours required to AC1 edge colour a graph can be established based on the sum of degrees of any group of vertices that form a clique.

**Lemma 3.** Given any graph $H$, if $v_1, \ldots, v_k \in V(H)$ and $v_1, \ldots, v_k$ form a clique then $H$ requires at least $\sum_{i=1}^{k} \frac{\deg(v_i)}{2}$ colours to AC1 edge colour.

**Proof.** $v_i$ must appear in $\deg(v_i)$ different coloured induced graphs of $H$. The sum of the appearances of those $k$ vertices over all of the coloured induced graphs is exactly $\sum_{i=1}^{k} \deg(v_i)$. If any three of those vertices appeared in the same colour induced graph then there would be a cycle since $v_1, \ldots, v_k$ form a clique. Therefore, any colour class can have at most two of the vertices in its coloured induced graph and at least $\sum_{i=1}^{k} \frac{\deg(v_i)}{2}$ colours are required to AC1 edge colour. \qed

This lemma means, in the worst case, a complete graph requires the same number of colours to AC1 edge colour as it does to strong edge colour. The lemma also implies a lower bound of $2\Delta$ colours that can be required for some planar graphs. As demonstrated below, it is possible to find planar graphs that require more than $2\Delta$ colours, but it is not trivial to prove the lower bound.
Consider the graph in figure 5.2 with the change in that the graph has an arbitrary maximum degree $\Delta$. In this new graph, add $\Delta - \deg(v)$ additional edges to each vertex $v$. All of the vertices shown and labeled in figure 5.2 have degree $\Delta$ and all those not shown have degree one. The graph $G$ will, in this section, always refer to the graph defined above.

**Lemma 4.** Suppose that $G$ has been $AC1$ edge coloured. Let $H$ be any colour induced subgraph of $G$ that does not contain both vertices $a$ and $b$. Then $|V(H)| \leq 5$.

**Proof.** Suppose that $|V(H)| = 6$

For any cycle in $G$, there must be at least one vertex of the cycle not present in $H$.

Case 1: Suppose that $a \in V(H)$.

Five of the vertices in $\{c, \ldots, j\}$ must be present in $V(H)$. In other words there are exactly $3$ vertices that are not in $V(H)$. Consider the $2$ sets of
3 disjoint cycles \( \{ae, fd, ij\} \) and \( \{fh, ai, cd\} \). Because the cycles, within each set, are disjoint there must be exactly one vertex from each cycle that is not in \( V(H) \).

Since \( a \in V(H) \), from the cycles \( ai, j \) and \( aec \) in \( G \), one of \( \{i, j\} \) and one of \( \{e, c\} \) are not in \( V(H) \). The cycles \( ijh \) and \( cde \) are also in \( G \), which implies that \( h \) and \( d \) both must be in \( V(H) \). This means that one of \( \{c, e\} \) must be in \( V(H) \) and similarly for \( \{f, g\} \) and \( \{i, j\} \). This implies that in \( H \) there is a path from \( a \) to \( d \), a path from \( d \) to \( h \), and a path from \( h \) to \( a \) that are disjoint except at the starting and ending points. This is a contradiction since \( H \) is acyclic.

**Case 2:** Suppose that \( b \in V(H) \).

As in case 1, there are exactly 3 vertices from the set not in \( V(H) \).

Since \( b \in V(H) \) and the cycles \( bcd \) and \( bhj \) are in \( G \), one of \( \{c, d\} \) and one of \( \{h, j\} \) are not in \( V(H) \). The cycle \( eif \) being in \( G \) implies that that \( g \) must be present in \( V(H) \). Consequently, this means that both \( d \) and \( h \) must be the two vertices not present in \( V(H) \). Because two of \( \{e, f, i\} \) must be in \( V(H) \), one of the cycles \( \{bcefg, bcej, bgfi, j\} \) must also be present. This is a contradiction as \( H \) is acyclic.

**Case 3:** Suppose that both \( a \) and \( b \) are not in \( V(H) \).

There may be only two vertices from the set \( \{c, \ldots, l\} \) that are missing from \( V(H) \). The cycles \( dfg \) and \( ijh \) contain both of the missing vertices, so \( c \) and \( e \) must both must be present in \( V(H) \). Due to the cycle \( cde \), \( d \) must absent in \( H \), which implies that both \( f \) and \( g \) are present. Finally, since only one of
\{i, h\} is in \(V(H)\), there is either cycle \(fgh\) or \(efi\) in \(H\). This is contradiction.

Every case ends in contradiction so the result follows. \(\square\)

**Lemma 5.** Suppose that \(G\) has been AC1 edge coloured. Let \(H\) be any colour induced subgraph of \(G\). If \(a\) and \(b\) are both in \(V(H)\), then \(|V(H)| \leq 4\).

**Proof.** Suppose that both \(a\) and \(b\) are in \(V(H)\). Neither \(c\) nor \(j\) can be in \(V(H)\) without creating a cycle in \(H\). Only one of \(e\) and \(i\) can be in \(V(H)\) without causing a cycle at \(a\). Only one of \(\{d, g\}\) and \(\{g, h\}\) can be in \(V(H)\) without creating a cycle with \(b\). If both \(d\) and \(h\) are in \(V(H)\) then neither \(e\) nor \(i\) can be in \(V(H)\). \(f\) can be in \(V(H)\) only if at most one of \(\{e, i, d, g, h\}\) is in \(V(H)\). This means that \(|V(H)| \leq 4\). \(\square\)

**Lemma 6.** Suppose that \(G\) has been AC1 edge coloured. If \(b\) and \(a\) have \(l\) colours in common then at least \(2\Delta + \frac{1}{5} l\) colours required.

**Proof.** There are 10 vertices labeled in \(G\), and each of these vertices must appear in exactly \(\Delta\) different colour induced graphs. In other words, the sum of the labeled vertices in all of the coloured induced graphs must be exactly \(10\Delta\).

There are \(2(\Delta - l)\) colours that are not both incident on \(a\) and \(b\). From lemma 4 in each of these coloured induced graphs, the number of labeled vertices in each is less than or equal to 5. These colours can only contribute \(10(\Delta - l)\) to the sum of labeled vertices.

There are \(l\) colours incident on both of \(a\) and \(b\) so from lemma 5 that accounts for at most \(4l\) labeled vertices in those colour induced graphs. This means, over all the colours incident at either \(a\) or \(b\), that the sum of labeled vertices in the colour
induced graphs is $10(\Delta - l) + 4l$. Since the sum must be exactly $10\Delta$ there are at least $6l$ vertices missing.

There must be additional colours to account for these $6l$ vertices. From lemma 4 each colour can only add at most $5$ more labeled vertices to the sum. Thus, there must be at least $\frac{6l}{5}$ colours that are not on either $a$ or $b$. This implies that the number of colours that must be used is at least $2(\Delta - l)$ from the distinct colours on $a$ and $b$, plus $l$ from the shared colours on $a$ and $b$ plus $\frac{6l}{5}$ additional colours. The total number of colours that must be used is then is at least $2\Delta + \frac{l}{5}$.

\[ \Box \]

**Theorem 3.** There exist planar graphs with maximum degree $\Delta \geq 11$ that requires at least $\frac{33\Delta}{16}$ colours to $AC1$ edge colour

Proof. Consider a graph $G^*$ consisting of three isomorphic copies $G_0, G_1, G_2$ of $G$ such that a triangle $xyz$ is formed with $x = a_0 = b_1$, $y = b_0 = a_2$ and $z = a_1 = b_2$ as
In figure 5.3, add additional edges to the labeled vertices of each $G_i$ so that every vertex of $G^*$ either has degree $\Delta$ or degree one.

Suppose that $G^*$ can be AC1 edge coloured in less than $\frac{33\Delta}{16}$ colours. Let $C_{x,y}$ be the number of colours in common between $x$ and $y$. Also let $I_x$ be the number of colours on $x$ that are not shared on either $y$ or $z$. Define $C_{y,z}, C_{x,z}, I_y$ and $I_z$ similarly.

Without loss of generality, assume that the maximum of $C_{x,y}, C_{x,z}$, and $C_{x,z}$ is $C_{x,y}$. Clearly, no colour can be counted in any two of $C_{x,y}, C_{y,z}, C_{x,z}$ without causing a cycle. Thus,

$C_{x,y} + C_{y,z} + C_{x,z} + I_x + I_y + I_z < \frac{33\Delta}{16}$ (5.1)

$C_{x,y} + C_{x,z} + I_x = \Delta$ (5.2)

$C_{y,z} + C_{x,y} + I_y = \Delta$ (5.3)

$C_{x,z} + C_{y,z} + I_z = \Delta$ (5.4)

$2\Delta + \frac{C_{x,y}}{5} < \frac{33\Delta}{16}$ (5.5)

(Equation (5.5) comes from lemma 6) Consider (5.2)+(5.3)+(5.4)-(5.1).

$C_{x,y} + C_{y,z} + C_{x,z} = 3\Delta - (C_{x,y} + C_{y,z} + C_{x,z} + I_x + I_y + I_z) > 3\Delta - \frac{33\Delta}{16} = \frac{15\Delta}{16}$

Also, rearranging (5.5) $C_{x,y} < \frac{5\Delta}{16}$. So $C_{x,y} + C_{y,z} + C_{x,z} \leq 3C_{x,y} < \frac{15\Delta}{16}$

This is a contradiction, so $G^*$ must use at least $\frac{33\Delta}{16}$ in any AC1 edge colouring.

$\blacksquare$
ECPCI Edge Colouring

Recall that ECPCI edge colouring is another degree of relaxation that can apply to the types of edge colouring used to create link schedules. However, since any colour induced graph can be cyclic, and in fact may even contain cycles with an odd length, it is not trivial to determine if the induced graph violates the condition. This is the first time ECPCI edge colouring is formally defined and analyzed so nothing is currently known about the lower bound on how many colours must be used in the worst case for ECPCI edge colouring planar graphs. We will construct a family of planar graphs that require $2\Delta + 1$ colours.

**Lemma 7.** There exist planar graphs with maximum degree $\Delta$ that requires $2\Delta + 1$ colours to ECPCI edge colour.

**Proof.** Consider a complete bipartite graph $K_{X,Y}$ with $X = \{a, b\}$ and $Y = \{y_1, \ldots, y_{\Delta-1}\}$. Augment $K_{X,Y}$ by adding the edges $(a, b)$ and $(y_i, y_{i+1})$ for $1 \leq i \leq \Delta - 2$. Figure 5.4 illustrates when $\Delta = 7$.

All edges incident to $a$ must have distinct colours which will account for $\Delta$ colours. If any edge $(u, v)$ along the path joining the vertices $Y$ has the same colour $t$ as an edge incident to either $b$ or $a$ then there exists a cycle $uvb$ or $uva$ that contains an odd number of edges that are coloured $t$. The path along the vertices of $Y$ requires at least two colours. Lastly, if for any $u, v$ an edge $(a, u)$ and an edge $(b, v)$ are both coloured $t$ then there exists a cycles $aub$ and $bva$ that both contain an odd number of $t$ coloured edges. Since the edge $(a, b)$ is already accounted for, this means that there is $\Delta - 1$ distinct colours incident to $b$. Therefore, $\Delta + 2 + \Delta - 1 = 2\Delta + 1$ colours are required to ECPCI edge colour the graph. \qed
Figure 5.4: A planar graph requiring $2\Delta + 1$ colours to ECPCI edge colour.

There is no indication if this means that it is possible to ECPCI edge colour all of the edges of a planar graph with less than $2\Delta + c$ colours where $c$ is some fixed constant. Also, unlike AC1 edge colouring, the argument for cliques cannot be used in a similar fashion. Though, just like strong edge colouring, it is clear that a complete graph will force every edge to be coloured differently.

5.2 Upper Bounds

Strong Edge Colouring

If an undirected interference graph $I$ is planar then there is a strong bound on the number of colours required to create an interference free link schedule using strong edge colouring. It is known that any planar graph with maximum degree $\Delta$ can be strong edge coloured with at most $4\Delta + 4$ colours [14]. We give an algorithm that will produce $4\Delta + 4$ colours on a planar communication graph and will meet the strong edge colouring condition with respect to the planar interference graph.
Algorithm 1: An algorithm to strong edge colour a planar graph

Require: A planar interference graph $I$ and a communication graph $G \subseteq I$

$EC \leftarrow edgeColour(G)$ // Edge colour the graph $G$. $EC$ is a collection of the edge
colour classes in $G$.

$StrongColours \leftarrow \{\}$

for all Colour class $C \in EC$ do

$H \leftarrow I[V_C]$ // $H$ is the colour induced graph for the colour class $C$

$H' \leftarrow$ Contract all edges in $C$ on $H$

$VC \leftarrow vertexColour(H')$ // Vertex colour the resulting graph.

for all Colour class $V \in VC$ do

$E \leftarrow$ the edges in $H$ corresponding to the vertices in $V$.

$StrongColours \leftarrow StrongColours \cup \{E\}$

end for

end for

return $StrongColours$

Lemma 8. Algorithm 1 will produce a strong edge colouring with at most $4\Delta + 4$
colours when run on a planar graph $I$ and a planar graph $G \subseteq I$. Further, if $\Delta_G \geq 7$
it will produce a strong edge colouring with at most $4\Delta$ colours.

Proof. Firstly, the algorithm edges colours the graph $G$ with at most $\Delta + 1$ edge
colours [34]. If $\Delta_G \geq 7$ then it is always possible to edge colour $G$ with $\Delta$ colours.

When looking at each colour class $C$ individually, it creates a new graph $H$ which
is the $C$ colour induced graph on $I$. It then contracts each coloured edge to form
a graph $H'$, and vertex colours this graph. If two edges of the same initial edge
colour are joined by an edge in $H$ then in $H'$ they will be adjacent and vertex
coloured different colours.

Because contracting an edge does not remove the planarity of $H, H'$ can be ver-
tex coloured with at most 4 colours [3]. Every edge will have two sets of colours,
one from the initial edge colouring and another from the vertex colouring. Thus,
this will define at most $4\Delta + 4$ colours. If $\Delta_G \geq 7$ then this will define at most $4\Delta$
Algorithm 1 can be run on general graphs, as well, to get a strong edge colouring for any graph.

**Lemma 9.** Algorithm 1 will produce a strong edge-colouring with at most 
\[ \left\lfloor \frac{7 + \sqrt{1 + 48\gamma}}{2} \right\rfloor (\Delta_G + 1) \] colours when run on any graph \( I \) and any graph \( G \subseteq I \).

**Proof.** This is derived from the fact that contracting an edge of a graph does not increase the genus. Further, it is known that vertex colouring a graph with genus \( \gamma \) is always possible in \( \left\lfloor \frac{7 + \sqrt{1 + 48\gamma}}{2} \right\rfloor \) colours [29].

**Theorem 4.** For a symmetric interference graph \( I \) and a symmetric communication graph \( G \subseteq I \), a link schedule can be created with at most 
\[ \left\lfloor \frac{7 + \sqrt{1 + 48\gamma}}{2} \right\rfloor (\rho_G + 2) \] time slots.

**Proof.** From lemma 8, a strong edge colouring of the undirected form of \( G \), with respect to the undirected interference graph \( I \), can be produced with at most 
\[ \left\lfloor \frac{7 + \sqrt{1 + 48\gamma}}{2} \right\rfloor (\Delta_G + 1) \] colours. The translation to PRN-colouring will double that value giving a link schedule with 
\[ \left\lfloor \frac{7 + \sqrt{1 + 48\gamma}}{2} \right\rfloor (\rho_G + 2) \] time slots.

**Corollary.** For a symmetric planar interference graph \( I \) and a symmetric communication graph \( G \subseteq I \), a link schedule can be created with at most \( 4(\rho_G + 2) \) time slots.

For planar graphs, algorithm 1 is only off by an additive constant of the lower bound, but there still is a gap between the upper bound and a lower bound. It is unknown if less than \( 4\Delta \) colours are always sufficient to strong edge colour a planar graph with maximum degree \( \Delta \geq 7 \).
It is also interesting to note that the complete graph on \( n \) vertices has a genus \( \gamma \) equal to \( \left\lfloor \frac{(n-3)(n-4)}{12} \right\rfloor \) [29]. By lemma 9, algorithm 1 would colour the edges of this graph and produce at most \( n(\Delta + 1) \) colours. However, there are only \( \frac{n(n-1)}{2} = \frac{n\Delta}{2} \) edges in a complete graph with \( n \) vertices. When edges are contracted, the genus will not increase, however, it may decrease. The number of colours produced by algorithm 1 may be significantly lower than the upper bound presented in lemma 9. It is unknown under what conditions the number of colours produced will be smaller. It would be interesting to discover if there are any graphs of large genus that require a relatively small number of colours for a strong edge colouring.

**ACI Edge Colouring**

ACI edge colouring is a more relaxed version of strong edge colouring meaning that algorithms will be able to get closer to the optimal link scheduling size. However, because links in a given schedule may have edges joining them in the interference graph, the stations will be unable to follow the RTS/CTS model. When a station is scheduled to be a receiving station, it will be unable to freely acknowledge the reception of data. Although the schedule is interference free, there is a possibility that some of the data could become corrupt and will need to be transmitted again. Without the ability to acknowledge, the receiving schedule must wait until it has broadcast rights along the same communication link to respond that it did not receive the transmitted data properly.

We show that algorithm 1 can be adapted to create an ACI edge colouring by making only a slight alteration. Instead of colouring the vertices when the edges of a particular colour class have been contracted, it will partition the vertices into
sets such that each set induces an acyclic graph. Recall, the minimum number of sets required is defined by the vertex arboricity of the graph. However, the work done on vertex arboricity assumes that the graph is simple (does not contain any multiple edges), so it is important that every edge contracted colour induced graph is simple. If the graph becomes a multi-graph through the contraction of edges, the bound for vertex arboricity increases to the same bound as vertex colouring. This means that the algorithm will only work on graphs with girth at least 5. This does not say that graphs with girth less than 5 can not achieve a similar bound, but rather, algorithm 2 cannot be run on such a graph.

Algorithm 2 A modified algorithm to ACI edge colour graphs

Require: A planar interference graph $I$ and a communication graph $G \subseteq I$

$EC \leftarrow edgeColour(G)$ // Edge colour the graph $G$. $EC$ is a collection of the edge colour classes in $G$.

$ACIColours \leftarrow \{\}$

for all Colour class $C \in EC$ do

$H \leftarrow I[V_C]$ // $H$ is the colour induced graph for the colour class $C$

$H' \leftarrow$ Contract all edges in $C$ on $H$

$AS \leftarrow arboricityPartition(H)$

for all Vertex set $V \in AS$ do

$E \leftarrow$ the edges in $H$ corresponding to the vertices in $V$.

$ACIColours \leftarrow ACIColours \cup \{E\}$

end for

end for

return $StrongColours$

Lemma 10. Algorithm 2 will produce an ACI edge colouring with at most

$$\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor (\Delta_G + 1)$$

colours when run on a graph $I$ with girth at least 5 and graph $G \subseteq I$.

Proof. Let $H$ be any colour induced graph and $H'$ be the fully edge contracted version of $H$ as described in the algorithm. Let $T$ be a set of vertices of $H'$ that
induce an acyclic graph and let $S$ be the corresponding edges in $H$. By creation, the graph induced by $T$ on $H'$ is acyclic. It is clear that if $H[V_S]$ contains a cycle then $T$ must also have created a cycle when induced on $H'$. This means that it will produce a valid AC1 edge colouring.

It was shown by Kronk [19] that the vertex arboricity for any graph with genus $\gamma$ is at most $\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor$. It should be noted that the bound is also tight. Thus, the algorithm returns at most $\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor (\Delta_G + 1)$ colours.

**Corollary.** Algorithm 2 will produce a AC1 edge colouring with at most $(2 + \lfloor \sqrt{\gamma} \rfloor)(\Delta_G + 1)$ colours when run on a graph $I$ with girth at least 7 and graph $G \subseteq I$.

**Proof.** This comes from the fact that when $I$ has girth at least 7, every edge contracted graph $H$ has girth at least 4. Thus, the arboricity of any edge contracted colour induced graph $H'$ is at most $2 + \lfloor \sqrt{\gamma} \rfloor$ [12].

**Theorem 5.** For a symmetric interference graph $I$ with girth at least 5 and a symmetric communication graph $G \subseteq I$, a link schedule can be created with at most $\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor (\rho_G + 2)$ time slots.

**Proof.** When applied to the undirected version of $I$ and $G$, algorithm 2 produces $\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor (\Delta_G + 1)$ colours. The translation from AC1 edge colouring to PRN-colouring will double the number of colours. Thus, $G$ can be link scheduled, with respect to $I$, in at most $\left\lfloor \frac{9 + \sqrt{1 + 48\gamma}}{4} \right\rfloor (\rho_G + 2)$ time slots.
Corollary. For a planar symmetric interference graph $I$ with girth at least 5 and a symmetric communication graph $G \subseteq I$, a link schedule can be created with at most $3(\rho_G + 2)$ time slots.

Proof. This follows from the fact that the genus of any planar graph is 0. □

Corollary. For a planar symmetric interference graph $I$ with girth at least 7 and a symmetric communication graph $G \subseteq I$, a link schedule can be created with at most $2(\rho_G + 2)$ time slots.

Recall, every strong edge colouring is also a $ACI$ edge colouring, so it is known that roughly $4\Delta$ colours are sufficient to $ACI$ edge colour a planar graph. $3\Delta$ has also been shown to be sufficient if the girth is at least 5. It would seem roughly $3\Delta$ colours should be sufficient all the time for planar graphs, however, it is still unknown if there are graphs that even require $3\Delta$ colours to $ACI$ edge colour. The graph $G^*$ from figure 5.3 has the maximum arboricity for planar graphs (which is 3) which may have been the property that forced it to require a large number of $ACI$ edge colours.

ECPCI Edge Colouring

Algorithm 2 can be applied without any modifications to ECPCI edge colour a graph since any $ACI$ edge colouring is an ECPCI edge colouring. Notice though, the girth condition can be lowered from at least 5 to at least 4. The girth condition was imposed to eliminate multi edges when doing edge contractions. There are only two types of conditions that will cause multi edges. In figure 5.5, where the bold edges are the coloured edges, the 3 cycle would be a colouring that violates the ECPCI edge colouring constraint, but the other is valid.
Lemma 11. Algorithm 2 will produce a valid ECPCI edge colouring when run on a graph $I$ with girth at least 4 and graph $G \subseteq I$.

Proof. Let $S$ be some colour class produced by algorithm 2. Suppose that $I[V_S]$ contains a cycle with odd parity. Let $C$ be such a cycle with the smallest number of edges in $S$. Since algorithm produces an ACI edge colouring on graphs with girth at least 5, $C$ must contain two edges $(a,b)$ and $(c,d)$ in $S$ that form a 4 cycle in $I$. Without loss of generality assume that the edges $(a,c)$ and $(b,d)$ exists in $E(I)$. Note that those two edges are not required to be in $C$. Let $P_{u,v}$ denote a path in $I$ from vertex $u$ to vertex $v$.

Case 1: $C$ consists of $\{(a,b), P_{b,c}, (c,d), P_{d,a}\}$.

There exists then a cycle $C'$ that consists of $\{(a,c), P_{c,b}, (b,d), P_{d,a}\}$ that contains two fewer edges from $S$ than $C$ which is a contradiction.

Case 2: $C$ consists of $\{(a,b), P_{b,d}, (d,c), P_{c,a}\}$.

Consider the two potential cycles $P_1 = \{(a,c), P_{c,a}\}$ and $P_2 = \{(b,d), P_{d,b}\}$. They are potential since $P_{c,a}$ could in fact be the edge $(a,c)$ and similarly for $P_{d,b}$. However, at least one of them must be an actual cycle. The number of edges from $S$ in $P_1$ and $P_2$ is two fewer that the number in $C$. This implies that one of $P_1$ or $P_2$ is a cycle with odd parity and fewer a number of edges from $S$ than $C$ which is a contradiction.
Therefore every colour class produced by the algorithm is a valid odd cycle colour.

**Theorem 6.** For a symmetric interference graph $I$ with girth at least 4 and a symmetric communication graph $G \subseteq I$, a link schedule can be created with at most $\left\lfloor \frac{9 + \sqrt{1 + 48G}}{4} \right\rfloor (\rho_G + 2)$ time slots.

**Corollary.** For a planar symmetric interference graph $I$ with girth at least 4 and a symmetric communication graph $G \subseteq I$, a link schedule can be created with at most $3(\rho_G + 2)$ time slots.

**Algorithm Analysis**

The runtime of algorithm is dependent on the algorithms for both edge and vertex colouring. These algorithms can be implemented either in a centralized or a distributed fashion. Further, there are algorithms that have an improved runtime when run on planar graphs.

Vizing’s proof [34] that any graph with $n$ vertices and $m$ edges can be edge coloured with at most $\Delta + 1$ colours describes an algorithm that can produce such a colouring in $O(mn)$ time. An improved algorithm, by Gabow et al., can produce $\Delta + 1$ edge colours and it will run in $O(m\sqrt{n\log n})$ time [15]. For graphs that are planar, Gabow et al. gave a general $O(n\log n)$ time algorithm. Cole and Kowalik [11] presented an $O(n)$ time algorithm to colour the edges when the maximum degree of a planar graph is larger than 9. A distributed edge colouring algorithm, given by Gandham et al. [16], requires $O(n\Delta^2 + n^2m)$ unicast messages to be sent and produces $\Delta + 1$ colours. However, in a distributed setting, multiple messages
can be sent in unison so the analysis of the time complexity of their algorithm is very conservative.

For vertex colouring, there are $O(n^2)$ and $O(n)$ time algorithms that can colour a planar graph in 4 and 5 colours respectively [10, 30]. Recall, for general graphs $G$ that are $c$ inductive, there exists a vertex $v$ with degree less than $c$ such that $G-v$ is also $c$ inductive. This means that there is an inductive ordering $v_1, \ldots, v_n$ of the vertices such that, for any $i$, $v_i$ has degree at most $c$ in the graph induced by the vertices $v_i, \ldots, v_n$. The inductive ordering of the vertices can be found in $O(\max(m, n))$ time [8]. Once the ordering is known, an algorithm can iterate over each vertex in reverse order and examine the colours of the vertices already processed. At most $c$ neighbours are already processed at each iteration, thus, $c+1$ colours can be produced. This algorithm would run in $O(\max(m, n) + nc)$.

A graph with genus $\gamma \geq 1$ is $c$-inductive for $c \leq \left\lfloor \frac{5 + \sqrt{48\gamma + 1}}{2} \right\rfloor$ [17, 37]. This means that every edge contracted colour induced graph that algorithm 1 creates will also be $c$-inductive for $c \leq \left\lfloor \frac{5 + \sqrt{48\gamma + 1}}{2} \right\rfloor$. Thus, a vertex colouring, with at most $\left\lfloor \frac{5 + \sqrt{48\gamma + 1}}{2} \right\rfloor + 1 = \left\lfloor \frac{7 + \sqrt{48\gamma + 1}}{2} \right\rfloor$ colours can be created in $O(\max(m, n) + n\sqrt{\gamma})$ time. It is important to note that the genus of the graph does not need to be known in order for this algorithm to run.

Using a distributed algorithm to vertex colour a graph, a randomized algorithm is able to produce $O(\Delta)$ colours in $O(\log n)$ time [1]. A deterministic distributed algorithm can colour the vertices with $O(\Delta^2)$ colours in $O(\log^* n)$ time [21]. This was the best known deterministic algorithm for vertex colouring until Barenboim and Elkin [6] created a deterministic algorithm to produce $O(a^{1+\epsilon})$ colours in $O(\log a \log n)$ time where $\epsilon > 0$ and $a$ is the edge arboricity of the graph. However, it is unknown if the edge arboricity can increase during edge contractions.
Algorithm 2 uses vertex arboricity instead of vertex colouring, but the two problems are related. Any independent set induces an acyclic graph, and every acyclic graph can be split into two independent sets. Similar to the algorithm for vertex colouring, if a graph is $c$ inductive, then an arboricity partition can be created with at most $\frac{c+1}{2}$ vertex sets. For planar graphs, there is an $O(n)$ time algorithm to produce a vertex arboricity partition of size 3 [31].

Table 5.1 and 5.2 give a summary of the results for algorithms 1 and 2. Each algorithm performs one edge colouring and $O(\Delta)$ vertex colourings or vertex arboricity partitions. For simplicity, $n_G$ and $m_G$ will denote the number of vertex and edges, respectively, of a graph $G$ and let $a_G$ be the edge arboricity of $G$. 

43
<table>
<thead>
<tr>
<th>Type</th>
<th>Condition</th>
<th>Runtime</th>
<th>Colours produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>$I$ is planar</td>
<td>$O(n_G \log n_G + \Delta_G n_I)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I$ is planar</td>
<td>$O(n_G \log n_G + \Delta_G n_I^2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I$ is planar and $\Delta_G \geq 9$</td>
<td>$O(n_G + \Delta_G n_I^2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\gamma \geq 1$</td>
<td>$O(m_G \sqrt{n_G \log n_G} + \Delta_G (\max(m_I, n_I) + n_I \sqrt{\gamma_I}))$</td>
<td>$\frac{7 + \sqrt{48\gamma + 1}}{2} (\Delta_G + 1)$</td>
</tr>
<tr>
<td>Distributed</td>
<td>General Graphs</td>
<td>$O(n_G \Delta_G^2 + n_G^2 m_G + \Delta_G \log a_I \log n_I)$</td>
<td>$O(\Delta_G^2)$</td>
</tr>
</tbody>
</table>

**Table 5.1:** Algorithm summary on an undirected graphs $I$ and $G \subseteq I$

<table>
<thead>
<tr>
<th>Type</th>
<th>Condition</th>
<th>Runtime</th>
<th>Colours produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized</td>
<td>$I$ is planar with girth $\geq 4$</td>
<td>$O(n_G \log n_G + \Delta_G n_I)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I$ is planar with girth $\geq 7$</td>
<td>$O(n_G \log n_G + \Delta_G n_I)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I$ is planar and $\Delta_G \geq 9$</td>
<td>$O(n_G + \Delta_G n_I)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\gamma \geq 1$</td>
<td>$O(m_G \sqrt{n_G \log n_G} + \Delta_G (\max(m_I, n_I) + n_I \sqrt{\gamma_I}))$</td>
<td>$\frac{9 + \sqrt{48\gamma + 1}}{4} (\Delta_G + 1)$</td>
</tr>
<tr>
<td>Distributed</td>
<td>General graphs</td>
<td>$O(n_G \Delta_G^2 + n_G^2 m_G + \Delta_G \log a_I \log n_I)$</td>
<td>$O(\Delta_G^2)$</td>
</tr>
</tbody>
</table>

**Table 5.2:** Algorithm summary on an undirected graphs $I$ and $G \subseteq I$
Chapter 6

Conclusions and Future Work

Link scheduling under a protocol level interference model is typically modeled as directed edge colouring of a communication graph. However, the range in which stations can cause interference with one another is typically greater than the range in which stations can communicate. This gives rise to a need for algorithms that can schedule the links in the communication graph that are sensitive to interference with respect to the interference graph. Algorithms 1 and 2 can take a distinct interference graph into account when creating link schedules.

Three undirected edge colouring methods were examined and translations were given to take undirected colourings and translate them into a directed PRN-colouring. Using strong edge colouring would allow a network to follow the more strict RTS/CTS model for communication whereas ACI and ECPCI edge colouring simply deal with fixed protocol level interference. Reductions were given to show that both ACI and ECPCI edge colouring are NP-complete.

For strong edge colouring, an algorithm was given that can link scheduled
the communication graph with maximum degree $\rho$ and an interference graph with genus $\gamma$ in $O(\sqrt{\gamma \rho})$ time slots. For planar graphs this algorithm gave a worst case of $4\rho + 8$ time slots. The trivial bound on the number of time slots required is $\rho$ so these algorithms produce a $O(\sqrt{\gamma})$ and a $O(1)$ approximation algorithm respectively.

In the worst case, because there also exists planar graphs that require $4\Delta - 4$ strong edge colours, any algorithm is going to need at least that many strong edge colours. This means that the algorithm presented cannot be significantly worse in that regard, however, it is unknown if the performance guarantee improves if the required number of colours for the graph is smaller. This also applies to non-planar graphs as well since there exists graphs that require $O(\sqrt{\gamma \Delta})$ colours.

The strong edge colouring algorithm was adapted to be able to produce both ACI and ECPCI edge colourings. Both algorithms on general directed communication graphs produce a performance guarantee of $O(\sqrt{\gamma \rho})$ time slots with a constant that is about half that for strong edge colouring. It also gives a performance guarantee of roughly three time the optimal link schedule size for planar graphs. However, the algorithm is restricted to graphs with girth greater than or equal to 5 for ACI edge colouring or greater than or equal to 4 for ECPCI edge colouring. The girth constraint is established to prevent a graph, when contracting edges, from becoming a multi graph.

For undirected planar graphs, it is unknown if it is always possible to ACI edge colour with $3(\Delta + 1)$ colours. In fact, it is not even known if there exists planar graphs that require $3\Delta$ ACI edge colours. Showing that a particular graph requires even slightly more than $2\Delta$ colours was not trivial. The graph, $G^*$ given in figure 5.3, is a planar graph with vertex arboricity of 3 which may be the contributing
factor to why it required more than $2\Delta$ colours. It may be possible to construct
general graphs with large arboricity that require many colours to ACI edge colour.

Lloyd and Ramanathan [23] gave an algorithm to link schedule a graph with
thickness $\theta$ with $O(\theta \rho)$ colours. For planar graphs ($\theta = 1$), using the undirected
style of edge colourings has given an improvement over the best known constant
for link scheduling by a factor of almost three. However, in general, it is unknown
if the improvement still applies to graphs with higher thickness. It is unknown if the
thickness of a graph can increase during edge contractions. If it cannot increase,
then the performance guarantee algorithms 1 and 2 can be expressed as roughly
$6\theta(\Delta + 1)$ and $3\theta(\Delta + 1)$ respectively.

It would be interesting to examine the link scheduling problem under physical
interference and compare the results to the results under protocol interference. It
has empirically been shown by Badia et al. [4] that the schedules produced un-
der physical interference can be better than those for protocol interference given
specific inputs and algorithms, however, this is unknown in general.
Bibliography


48


49


50


