



HAL
open science

Algorithms for Promise Coloring Problems on Tournaments and Graphs

Felix Klingelhofer

► **To cite this version:**

Felix Klingelhofer. Algorithms for Promise Coloring Problems on Tournaments and Graphs. Data Structures and Algorithms [cs.DS]. Université Grenoble Alpes [2020-..], 2023. English. NNT : 2023GRALM073 . tel-04612259

HAL Id: tel-04612259

<https://theses.hal.science/tel-04612259v1>

Submitted on 14 Jun 2024

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique

Spécialité : Informatique

Unité de recherche : Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble

Algorithmes pour des Problèmes de Coloration avec Promesse sur les Tournois et les Graphes

Algorithms for Promise Coloring Problems on Tournaments and Graphs

Présentée par :

Felix KLINGELHOEFER

Direction de thèse :

Louis ESPERET

DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES

Directeur de thèse

Alantha NEWMAN

CHARGÉE DE RECHERCHE, CNRS DELEGATION ALPES

Co-encadrante de thèse

Rapporteurs :

PIERRE CHARBIT

MAÎTRE DE CONFÉRENCES HDR, UNIVERSITÉ PARIS CITE

SANJEEV KHANNA

PROFESSEUR, UNIVERSITY OF PENNSYLVANIA

Thèse soutenue publiquement le **4 décembre 2023**, devant le jury composé de :

PIERRE CHARBIT

MAÎTRE DE CONFÉRENCES HDR, UNIVERSITÉ PARIS CITE

Rapporteur

SANJEEV KHANNA

PROFESSEUR, UNIVERSITY OF PENNSYLVANIA

Rapporteur

CARLA GROENLAND

ASSISTANT PROFESSOR, DELFT UNIVERSITY OF TECHNOLOGY

Examinatrice

ZOLTAN SZIGETI

PROFESSEUR DES UNIVERSITÉS, GRENOBLE INP

Président

STEPHAN THOMASSE

PROFESSEUR DES UNIVERSITÉS, ENS DE LYON

Examineur

Invités :

LOUIS ESPERET

DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES

ALANTHA NEWMAN

CHARGÉE DE RECHERCHE, CNRS DELEGATION ALPES



Acknowledgements

This thesis would not have been possible without the help of many people who provided guidance, support, or contributed in various different ways, so I would like to thank all of them.

First, I want to thank my supervisors Alantha Newman, Louis Esperet and during my first year Moritz Mühlenthaler. Alantha your rigorous scientific approach and attention to detail taught me a lot. You were also always there to work together and push me forwards when I needed it, and I am deeply grateful for your dedication. Moritz it was a pleasure working with you during my first year. Louis, thank you for helping me through the administrative hurdles I faced, and for providing support throughout my thesis.

I thank Frédéric Meunier for his guidance in the monitoring committee. I also thank all the people with whom I have collaborated for my thesis, or as a part of the workshops and events I attended.

A big thank you to the jury: Zoltan Szigeti, Carla Groenland, Stéphan Thomassé and the referees Pierre Charbit and Sanjeev Khanna. Your questions were helpful for opening up new perspectives, and your comments helped improve the final version of the manuscript.

I would like to thank all the members of the laboratory for providing a very nice work environment. In particular, my officemates Ernest, Cléopée and Ugo, thank you for putting up with me. Thanks to all the card players of the lunch break for many fun rounds.

Thank you to all my friends who made my years in Grenoble a great time. Thanks Manu for making me discover the mountains in a new way through many great cross-country skiing outings. Thanks Lucas for always being cheerful and up for everything. Thanks Jean-Fé and Coco, you are both great fun to be around and I really enjoyed spending evenings at the bar with you. Thank you Ugo for (passionately) teaching me all about coinche and rugby.

Thanks to my family, my parents Marc and Frauke and my sister Lisa, for bringing me to this point in the first place. You always encouraged me, and I would not have come this far without you.

Last, but far from least, I thank my girlfriend Aline. You were an amazing support and I could always rely on you whenever I needed it. Thank you for making my life brighter!

Abstract

The first part of this thesis is on the subject of coloring tournaments, from an algorithmic, complexity and structural perspective. A k -coloring of a directed graph, and in particular a tournament, is a partition of its vertices into k acyclic sets. The chromatic number of a directed graph or a tournament is then the minimum k such that it is k -colorable. Deciding if a tournament is 2-colorable is already NP-hard. A natural problem, akin to that of coloring a 3-colorable graph with few colors, is to color a 2-colorable tournament with few colors. This problem does not seem to have been addressed before, although it is a special case of coloring a 2-colorable 3-uniform hypergraph with few colors, which is a well-studied problem with super-constant lower bounds.

We present an efficient decomposition lemma for tournaments and show that it can be used to design polynomial-time algorithms to color various classes of tournaments with few colors, including an algorithm to color a 2-colorable tournament with ten colors. For the classes of tournaments considered, we complement our upper bounds with strengthened lower bounds, painting a comprehensive picture of the algorithmic and complexity aspects of coloring tournaments. We then extend our results to different classes of tournaments and digraphs.

The neighborhood of an arc uv in a tournament T is the set of vertices that form a directed triangle with arc uv . By using our decomposition lemma, we show that if the neighborhood of every arc in a tournament has bounded chromatic number, then the whole tournament has bounded chromatic number. This holds more generally for oriented graphs with bounded independence number, and we extend our proof from tournaments to this class of dense digraphs. As an application, we prove the equivalence of a conjecture of El-Zahar and Erdős and a recent conjecture of Nguyen, Scott and Seymour relating the structure of graphs and tournaments with high chromatic number.

In the second part of this thesis, we focus on the problem of finding maximum stable sets in the class of CYCLE-PLUS-TRIANGLES graphs. A CYCLE-PLUS-TRIANGLES graph is the disjoint union of t triangles and a Hamilton cycle on the $3t$ vertices. It is 3-colorable, and we give an overview of the different proofs of its 3-colorability. There is, however, no known efficient algorithm to find a 3-coloring or even to find a maximum stable set (i.e., a stable set of size t).

We present a simple randomized algorithm that outputs a maximum stable set upon termination. We conjecture that for any `CYCLE-PLUS-TRIANGLES` instance, our algorithm terminates in expected polynomial time. In an (unsuccessful) effort to find hard instances for our algorithm, we discuss structure and properties of `CYCLE-PLUS-TRIANGLES` instances and methods for generating them. Finally we examine the behavior of these algorithms on related problems, such as 3-coloring, or finding maximum independent sets in a more general class of graphs.

Résumé

La première partie de cette thèse porte sur le sujet de la coloration de tournois, sous l'angle de l'algorithmie, de la complexité et de la structure. Une k -coloration d'un graphe orienté, et en particulier d'un tournoi, est une partition de ses sommets en k ensembles acycliques. Le nombre chromatique d'un graphe orienté ou d'un tournoi est alors le plus petit k tel que le graphe puisse être k -coloré. Décider si un tournoi peut être 2-coloré est déjà NP-difficile. Un problème naturel, similaire à celui de la coloration d'un graphe 3-colorable avec peu de couleurs, est de colorer un tournoi 2-colorable avec peu de couleurs. Ce problème ne semble pas avoir été abordé auparavant, bien qu'il s'agisse d'un cas particulier de la coloration d'un hypergraphe 3-uniforme 2-colorable avec peu de couleurs, problème bien étudié avec des bornes inférieures super-constantes.

Nous présentons un lemme de décomposition efficace pour les tournois et montrons qu'il peut être utilisé pour concevoir des algorithmes en temps polynomial pour colorer différentes classes de tournois avec peu de couleurs, notamment un algorithme pour colorer un tournoi 2-colorable avec dix couleurs. Pour les classes de tournois considérées, nous complétons nos bornes supérieures par des bornes inférieures renforcées, offrant ainsi une vision complète des aspects algorithmiques et de complexité de la coloration des tournois. Nous étendons ensuite nos résultats à différentes classes de tournois et de graphes orientés.

Le voisinage d'un arc uv dans un tournoi T est l'ensemble de sommets qui forment un triangle orienté avec l'arc uv . En utilisant notre lemme de décomposition, nous montrons que si le voisinage de chaque arc dans un tournoi a un nombre chromatique borné, alors tout le tournoi a un nombre chromatique borné. Ceci est également vrai de manière plus générale pour les graphes orientés avec un nombre d'indépendance borné, et nous étendons notre preuve pour les tournois à cette classe de graphes orientés denses. En tant qu'application, nous démontrons l'équivalence d'une conjecture d'El-Zahar et Erdős et d'une conjecture récente de Nguyen, Scott et Seymour concernant la structure des graphes et des tournois avec un grand nombre chromatique.

Dans la deuxième partie de cette thèse, nous nous concentrons sur le problème de la recherche de stables maximums dans la classe des graphes Cycle-plus-Triangles. Un

graphe Cycle-plus-Triangles est l'union disjointe de t triangles et d'un cycle Hamiltonien sur les $3t$ sommets. Il est 3-colorable, et nous présentons un aperçu des différentes preuves de sa 3-colorabilité. Cependant, il n'existe pas d'algorithme efficace connu pour trouver une 3-coloration ou même pour trouver un ensemble stable maximum (c'est-à-dire un stable de taille t).

Nous présentons un algorithme aléatoire simple qui produit un ensemble stable maximum lorsqu'il termine. Nous émettons l'hypothèse que pour toute instance de Cycle-plus-Triangles, notre algorithme s'achève en temps polynomial attendu. Dans une démarche (non fructueuse) visant à trouver des instances difficiles pour notre algorithme, nous discutons de la structure et des propriétés des instances de Cycle-plus-Triangles et des méthodes pour les générer. Enfin, nous examinons le comportement de ces algorithmes sur des problèmes connexes, tels que la 3-coloration ou la recherche d'ensembles indépendants maximums dans une classe plus générale de graphes.

Contents

Main Introduction	5
I Coloring Tournaments	9
1 Introduction	10
1.1 Coloring digraphs and tournaments	10
1.2 Relationship between tournaments and graphs	12
1.2.1 Tournaments and the Erdős-Hajnal conjecture	13
1.2.2 Forbidden structures in tournaments and graphs	14
1.3 Complexity of coloring promise problems	16
1.4 Organization and main results of Part I	17
1.5 Notation	20
2 Algorithms for coloring tournaments	21
2.1 Efficient decomposition for coloring	21
2.2 2-colorable tournaments	24
2.3 3-colorable tournaments	26
2.4 Certificates of non-2-colorability	28
3 Complexity of coloring tournaments	29
3.1 NP-hardness of deciding 2-colorability	29

3.2	NP-Hardness of approximate coloring of k -colorable tournaments . . .	31
3.3	Reduction from coloring graphs to coloring tournaments	36
3.4	Hardness of approximation for general tournaments	39
4	Extensions	42
4.1	Light tournaments	42
4.2	Other decompositions for light tournaments	46
4.3	H_k -free tournaments	51
4.4	Extensions to dense digraphs	54
4.4.1	Decomposition for digraphs	55
4.4.2	Algorithm for 2-colorable digraphs	57
4.4.3	C_3 -free digraph with $\alpha = 2$	58
5	Arc local-to-global	61
5.1	Overview of our results	61
5.2	Arc local-to-global for tournaments	63
5.2.1	Jewels and other tools for coloring t -arc-bounded tournaments	64
5.2.2	Proof of Theorem 5.1.3	66
5.3	Arc local-to-global for dense digraphs	67
5.4	Equivalence of Conjectures 5.1.1 and 5.1.2	74
5.4.1	Proof of Conjecture 5.1.2, assuming Conjecture 5.1.1	76
5.4.2	Proof of Conjecture 5.1.1, assuming Conjecture 5.1.2	78
6	Conclusion of Part I	80
II	Cycle-plus-Triangles Graphs	82
7	Introduction	83

7.1	Organization of Part II	85
7.2	Notation	86
7.3	Proof of 3-colorability by Fleischner and Stiebitz	87
7.4	Sachs' proof of 3-colorability	88
7.4.1	Reduction for graphs with low span	88
7.4.2	Transformation towards graphs of lower span	89
7.4.3	Proof by induction	90
7.5	Alon's proof of independence number $n/3$	95
8	Algorithms for finding stable sets in cycle-plus-triangles graphs	96
8.1	Correspondence between instances and permutations	96
8.2	Algorithms for finding a maximum stable set	99
8.2.1	Locally minimal configurations	102
8.2.2	Deterministic token-sliding algorithm	102
8.3	Intuition for efficient termination	104
8.4	Efficient termination on specific instances	106
8.4.1	Chain-of-twins	106
8.4.2	Chain-of-links	107
9	Empirical evaluation of the algorithms	116
9.1	Structure and generation of problem instances	116
9.1.1	Generating instances uniformly at random	117
9.1.2	Generating instances with high span	120
9.2	Experimental results	125
9.2.1	Performance of deterministic algorithm	127
9.3	Lock-in cycles for RR-TS-Alg	127
9.3.1	Instance with high probability of entering a lock-in cycle	129

10 Extensions	132
10.1 Extensions to multiple stable sets and fair representation	132
10.1.1 Two and three maximum stable sets	132
10.1.2 Fair representation	133
10.2 Triangles-plus-two-factors graphs	135
10.2.1 Triangles-plus-triangles graphs	136
10.2.2 Hardness results	137
10.2.3 Why intuition fails	140
10.2.4 Experimental results for triangles-plus-two-factors	140
11 Conclusion of Part II	142

Main Introduction

A graph $G = (V, E)$ is a mathematical object defined by a set of vertices V and a set of edges E , that may be oriented or not. Vertices can be thought of as representing objects (e.g., countries), whereas edges model the relationship between these objects (e.g., two countries sharing a border). More precisely, E is a set of pairs of vertices, and we say there is an edge between a pair of vertices u and v if and only if $(u, v) \in E$. Graphs where every edge is oriented are called *directed graphs*. Complete directed graphs (where there is an edge between every pair of vertices) are referred to as *tournaments*, and will be our main focus in the first part of this thesis.

There are many algorithmic questions on graphs, for example finding a set of vertices S such that no pair of vertices of S has an edge between them. Such a set S is often referred to as an *independent* or *stable set*. The maximum size of such a set in a graph G is called the *independence number*, represented by $\alpha(G)$. The complement of an independent set, called a *clique*, is a set of vertices S where each pair of vertices in S has an edge between them. The size of the largest clique in a graph G is called the *clique number*, represented by $\omega(G)$. The problems of determining whether there exists a stable set or a clique of a given size are classical NP-complete problems [Kar72]. NP is a complexity class that contains all decision problems (i.e., problems whose answer is either YES or NO), which can be solved in non-deterministic polynomial time [Coo71]. An alternative definition of NP is that it contains all the decision problems whose YES instances have a certificate of polynomial size. The class of problems whose NO instances have a certificate of polynomial size is coNP. A problem is said to be complete for a complexity class if an algorithm that could solve it within the prescribed constraints of the respective complexity class could then be used to solve all other problems in this complexity class. In the example of determining whether there exists a stable set of a given size, a certificate can be a stable set of the required size, since it is easy to check whether a given set of vertices is actually a stable set, and to compute its size.

In this thesis, we focus on problems related to coloring graphs. A coloring of a graph is a partition of its vertex set into stable sets. It can also be defined as assigning a color to each vertex such that each edge has different colors on its endpoints. The minimum number of colors needed to color a graph G this way is defined to be the *chromatic number*, represented by $\chi(G)$. Deciding whether a graph has chromatic number at most k is also an NP-complete problem [Kar72], even when k is fixed for $k \geq 3$. Graphs that are 2-colorable, also called *bipartite*, can be colored in polynomial time simply by coloring a vertex with one color, then all its neighbors with the other, and continuing until the entire (connected) graph is colored. Bipartite graphs can also be characterized as the class of graphs that do not contain a cycle of odd length.

In theoretical computer science, we are often interested in two types of problems: decision problems that always have YES or NO as an answer, and search problems, where the solution is a certificate for a related decision problem. Such problems have their own complexity classes; the analogue of NP for search problems is FNP, which stands for *Functional Non-deterministic Polynomial-time solvable*. For example, the problem of determining whether a graph has chromatic number at most k is a decision problem, whereas the problem of coloring a graph with a given number of colors is a search problem. Every search problem similarly has a corresponding decision problem, which is “easier” in the sense that solving the search problem always solves the associated decision problem. The converse may or may not be true, depending on the specific problem. For instance, an algorithm that can solve the stable set decision problem can be used to find a maximum stable set. To see this, consider the following algorithm that uses an oracle to solve the stable set decision problem. Given a graph G with $\alpha(G) \geq k$, take any vertex v , and let G' be the graph obtained from G by removing v and all its neighbors. Use the oracle to determine if $\alpha(G') \geq k - 1$: if it is the case, add v to a set S and continue the algorithm on G' , and if not go back to G and take a different vertex. Doing this will iteratively build a stable set of size k .

This argument can be generalized to show that all search problems derived from NP-complete decision problems can be reduced to their decision version [AB09], but some problems that are not known to be NP-complete, such as integer factorization, are not known to have this property. It is then natural to ask whether knowing if a solution exists actually helps in finding one. Search problems for which the existence of a solution is guaranteed are called *total* search problems. They also have their own complexity classes; for example, TFNP is the complexity class of total search problems that can be solved in non-deterministic polynomial time. While TFNP is often considered to contain intractable problems, showing NP-hardness for any of them would lead to a complexity class collapse, $\text{NP} = \text{coNP}$ [GP18]. This is

generally conjectured to be false, but is still a major open problem in the field of complexity. TFNP contains subclasses that use the same argument to prove the totality of the search problem [Pap94]. For example, given n positive integers with sum less than $2^n - 1$, the pigeonhole principle guarantees the existence of two distinct subsets with the same sum. The problem of finding these two subsets, the equal sums problem, therefore lies in the PPP class, which stands for *Polynomial-time Pigeonhole Principle*. Another famous complexity subclass of TFNP is PPA, for *Polynomial-time Parity Argument*, which contains problems whose proof of totality relies on a parity argument (e.g., a graph with a vertex of odd degree must contain another vertex of odd degree).

The guarantee of a solution can also stem from a *promise* rather than as innate property of the inputs. Promise problems are a means of restricting the input space (e.g., to the set of YES instances of the associated decision problem) and thus a way to study “easier” instances. Promises often guarantee the existence of a solution, but might not always be useful. For example, if the problem is to find a stable set of size $n/3$ in a graph on n vertices, then the promise of being 3-colorable guarantees the existence of the desired solution. If the problem is to find a 3-coloring of a graph with independence number at least $n/3$, then the promise does not guarantee the existence of the desired solution. In the case of graph coloring problems, one of the most widely-studied promise problem is that of coloring 3-colorable graphs. Such graphs are still NP-hard to color with three colors. Depending on the promise, it can be possible to efficiently determine whether a given instance belongs to the promise class or not. For example, given the promise of having no clique of size at least 3, it is easy to check whether a graph belongs to the promised input space by looking at each group of three vertices and checking whether it forms a clique. On the other hand, if the promise is that a graph is 3-colorable, it is NP-hard to decide if a given graph actually belongs to the promised input space.

In the area of graph theory, we are usually concerned with proving that some class of graphs has a specific property, rather than with designing algorithms for deciding if a given graph has this property. In that sense, the focus is not on search problems, though algorithms may come as a side-product. Proofs of existence are said to be *constructive* when they explicitly build a certificate for the associated problem. An example of a constructive proof that a graph has chromatic number three is one that produces a 3-coloring. In contrast, a *non-constructive* proof of existence does not build the desired object, but simply proves that one must exist. These proofs are often based on counting or parity arguments. For example proving that the number of 3-colorings of a graph is odd immediately implies that the graph has chromatic

number at most 3, but does not yield 3-coloring.

An algorithmic paradigm, which we follow in this thesis, is to study hard search problems by considering promises that guarantee the existence of a solution in order to consider instances where solutions can be found more easily. In the first part of this thesis, we have explicit promises that are crucially used in the design of the algorithms to color tournaments, a notion that we define shortly. As a byproduct of our efficient algorithms, we also develop new tools that we can in turn use to prove existential results for a problem in graph theory. In the second part of this thesis, we study a total search problem which is finding a stable set of size $n/3$ in a graph class that is promised to be 3-colorable. This promise of 3-colorability stems from proofs that are both non-constructive and non-obvious. The challenge in this case is to find a constructive proof leading to an efficient algorithm.

Part I

Coloring Tournaments

Chapter 1

Introduction

1.1 Coloring digraphs and tournaments

A digraph $D = (V, A)$ is a graph defined by a set of vertices V and a set of arcs A , each of which has an orientation. For each pair of vertices u and v , there can be an arc from u to v , an arc from v to u , no arc between u and v , or arcs in both directions. The subgraph consisting of the arcs $uv \in A$ and $vu \in A$ is referred to as a *digon*. Digon-free digraphs are called *oriented* graphs. An oriented graph can equivalently be defined as an *orientation* of a simple undirected graph, where an orientation of a graph is obtained by assigning an orientation to each one of its edges.

A subset of vertices $S \subseteq V$ induces the *subdigraph* $D[S]$. If this subdigraph contains no directed cycles, then it is said to be *acyclic*. The problem of *coloring a digraph* D is that of partitioning the vertices into the minimum number of acyclic sets, sometimes referred to as the *dichromatic number of a digraph*, $\vec{\chi}(D)$. This notion was introduced by Neumann-Lara [Neu82].

This is a generalization of the problem of coloring graphs, since an instance of graph coloring can easily be reduced to an instance of digraph coloring by replacing each edge with a digon. The *dichromatic number of an undirected graph* G , $\vec{\chi}(G)$, is defined as the maximum dichromatic number of a digraph over all orientations of G [Erd79, Neu82]. The relationship between the chromatic number and the dichromatic number is an interesting open problem. A famous conjecture states that the chromatic number of an undirected graph is upper bounded by a function of its dichromatic number [Erd79]. While it is easy to prove that any graph with dichro-

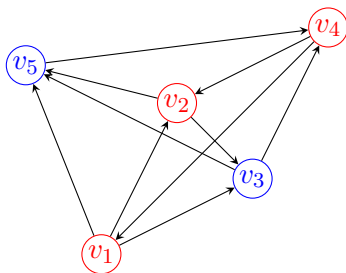


Figure 1.1: A 2-colored tournament.

matic number 1 has chromatic number at most 2 because it cannot have any cycles, this conjecture is open even for dichromatic number 2.

Conjecture 1.1.1. *There exists a function f such that for any undirected graph G , $\chi(G) \leq f(\vec{\chi}(G))$.*

[Erd79] proved that the complete graph on n vertices (i.e., a simple graph with an edge between every pair of vertices) has dichromatic number $\Omega(n/\log n)$. Complete graphs have chromatic number equal to their number of vertices. Thus, we can see that Conjecture 1.1.1 holds for complete graphs; in other words, there is always an orientation of a complete graph with dichromatic number $\Omega(n/\log n)$.

An orientation of a complete graph is also referred to as a *tournament*. In a tournament $T = (V, A)$, for each pair of vertices $i, j \in V$, there is either an arc from i to j or an arc from j to i (but not both). Since tournaments are a subclass of digraphs, the problem of coloring tournaments is defined the same way as the problem of coloring digraphs. A tournament contains a directed cycle if and only if it contains a directed triangle. Therefore the problem of coloring a tournament is equivalent to partitioning the vertices into the minimum number of sets so that each set does not contain a directed triangle. Each set will then be acyclic since it cannot contain a directed cycle, but it is also *transitive* since there is an arc between any pair of vertices. An example of a colored tournament is shown in Figure 1.1. A digraph, and in particular a tournament, is said to be *strongly connected* if and only if there is a directed path between any two pairs of vertices in each direction. When a digraph is not strongly connected, we can color each strong component (i.e., strongly connected subdigraph) separately, and the resulting coloring will be proper even if we use the same color palette on each strong component. Therefore, when we consider algorithms to color a digraph or tournament, we usually assume that they are strongly connected.

We will also study digraphs that are close to tournaments, in the sense that there must be arcs between many pairs of vertices (i.e., they are *dense*). We will measure this by the *independence number* of a digraph, which is the maximum size of a set of vertices I such that there is no arc in A between any two vertices of I . In particular we will study digraphs whose independence number is bounded by some constant.

1.2 Relationship between tournaments and graphs

Digraphs are more general objects than graphs since, as noted previously, any graph can be represented as a digraph by replacing every edge with a digon. As discussed with respect to Conjecture 1.1.1, the relationship between chromatic number of graphs and orientations of simple undirected graphs, when we forbid digons, is far from clear. Since tournaments are a very specific class of oriented graphs, it would seem reasonable to assume that they might not capture the complexity of the class of simple undirected graphs with respect to problems such as coloring. So a natural question is how graphs relate to tournaments, or more specifically, how do problems regarding chromatic number on graphs relate to those on tournaments? Does any problem involving the chromatic number of a graph have an equivalent formulation in terms of tournaments?

This question has been investigated and some tools have been developed to relate chromatic number of graphs and tournaments. A standard technique is to assign an ordering to the vertices of a tournament, and look at the graph on the same vertex set, where uv is an edge of the graph iff its orientation is decreasing with respect to the chosen ordering [BCC⁺13, CSSS24]. Such a graph is called the *backedge graph* of a tournament for a given ordering. Interestingly, this allows many coloring problems on graphs to be stated equivalently as problems on tournaments. As is standard, we use $\chi(G)$ and $\vec{\chi}(T)$ to denote the chromatic number of a graph G and a tournament T , respectively. We use $\omega(G)$ to denote the size of a maximum clique of G . Specifically, the chromatic number of a tournament T and its backedge graph G can be related as follows:

$$\vec{\chi}(T) \leq \chi(G) \leq \omega(G)\vec{\chi}(T).$$

This will be a useful tool in Part I and its proof is provided later on in the thesis.

1.2.1 Tournaments and the Erdős-Hajnal conjecture

A good example of the relationship between graphs and tournaments is a famous conjecture by Erdős and Hajnal [EH89], which has an equivalent statement in terms of tournaments. This conjecture is about graphs in which we forbid a specific graph H as a subgraph, which are called H -free graphs.

Conjecture 1.2.1. *For every graph H , there exists a positive constant $\epsilon(H)$ such that every H -free graph on n vertices contains a clique or an independent set of size $\Omega(n^{\epsilon(H)})$.*

We will refer to this conjecture as the EH conjecture, and say that a graph H verifying this conjecture has the *EH property*. This conjecture has been verified on some classes of graphs, notably all graphs obtained by *replacement* [APS01]. A replacement in a graph G consists of replacing a vertex v by a different graph G' , such that all vertices in $V(G')$ have the same adjacencies as v with vertices of $V(G) \setminus \{v\}$. This operation is sometimes referred to as substitution [Chu14]. Then if a graph G is obtained by taking a graph with the EH property, and substituting one of its vertices with another graph that has the EH property, G will also have the EH property.

[APS01] proved that the EH conjecture is equivalent to the following statement on tournaments.

Conjecture 1.2.2. *For every tournament T , there exists a positive constant $\epsilon(H)$ such that every H -free tournament on n vertices contains a transitive set of size $\Omega(n^{\epsilon(H)})$.*

This reformulation of the EH conjecture motivated the study of the existence of large transitive sets, and also coloring, in tournaments. [BCC⁺13] completely characterized the class of tournaments H such that every H -free tournament has a transitive set of size $\Omega(n/k)$ for some constant k . They call such a tournament H a *celebrity* and prove that the class of celebrities is the same as the class of *heroes*, which are the tournaments H such that every H -free tournament has bounded dichromatic number. The main theorem of their paper is a recursive characterization of heroes. T_i is defined to be the transitive tournament on i vertices, and if X , Y and Z are tournaments, $\Delta(X, Y, Z)$ is the tournament obtained by the union of X , Y and Z where all arcs from vertices of X go towards vertices of Y , from Y to Z and from Z to X .

Theorem 1.2.1. *A tournament is a hero iff all its strongly connected components are heroes. A strongly connected tournament is a hero iff it equals $\Delta(H, T_k, T_1)$ or $\Delta(H, T_1, T_k)$ where H is another hero, and $k \geq 1$ is a constant.*

The main idea behind the proof of this theorem for a strongly connected hero is a decomposition of a hero-free tournaments into a *jewel chain*, where a jewel is a subset of fixed size such that any partition contains two fixed tournaments, one in each partition. The authors then establish a number of structural results on the chains, notably the absence of long backward arcs between different jewels, which is then used to bound the dichromatic number of the tournament. Though not explicitly algorithmic, this type of decomposition can be made into an efficient algorithm for coloring some types of tournaments, which we discuss later on in this thesis.

To prove that a tournament is a hero only if all its strong components are heroes, the authors argue that every subtournament of a hero is a hero. The other direction is equivalent to saying that if A and B are heroes, then the tournament H obtained by the union of A and B where all arcs go from vertices of A to those of B must also be a hero. The proof of this relies on a family of tournaments they call *r-mountains*, which have a specific structure which gives them high chromatic number and bounded size. The proof then relies on an induction on the existence of an *r*-mountain in the H -free tournament, bounding the chromatic number of the H -free tournament in both cases.

1.2.2 Forbidden structures in tournaments and graphs

After looking at classes of graphs or tournaments in which a specific graph or tournament is forbidden, it is natural to consider forbidding not a single graph or tournament, but a set of them. This will be the object of the following subsection, though we will start by presenting a forbidden class of subtournaments under the light of neighborhoods of vertices.

The *local dichromatic number* of a tournament is defined to be the maximum dichromatic number over the outneighborhoods of the vertices. The class of tournaments with bounded local dichromatic number were conjectured to also have bounded dichromatic number [BCC⁺13]. This was proven by Haratyunyan et al. [HLTW19] resulting in the following theorem.

Theorem 1.2.2. *There is a function f , such that every tournament with local dichromatic number c has dichromatic number at most $f(c)$.*

The proof of this theorem relies on structural analysis of tournaments in order to prove that for every constant k , either a tournament has a dominating set of bounded size, or it has a subset of bounded size with dichromatic number at least k . This lemma can then be used for tournaments with local dichromatic number bounded by a constant c , to show that they must have a dominating set of bounded size. Indeed, if $k > c$, a subset with dichromatic number at least k must also be a dominating set (else it would be a subset of some outneighborhood, and the local dichromatic number would be at least k). The upper bound they obtain on the function f is however a tower function, and the best known lower bound is 2 (for example by taking a directed triangle), which lead to the following conjecture [NSS23b].

Conjecture 1.2.3. *Every tournament with local dichromatic number c has dichromatic number at most $2c$.*

While the authors state that this conjecture is likely to be false, finding a better (linear?) upper bound on the function f is an interesting open problem.

Another way of seeing this result is that forbidding the class of tournaments formed by a vertex that is complete to a tournament with high dichromatic number as subtournaments bounds the dichromatic number of a tournament. Thus, in a sense the class of tournaments formed by a vertex that is complete to a tournament with high dichromatic number acts as a sort of “hero”. In fact, [NSS23b] prove a stronger theorem.

Theorem 1.2.3. *Let $k, \ell \geq 1$, and $C_{k,\ell}$ be the class of tournaments that can be decomposed as $\Delta(B, T_k, T_1)$, where B is a tournament such that $\vec{\chi}_c(B) \geq \ell$. There is a function f , such that every tournament that has no subtournament in $C_{k,\ell}$ has dichromatic number at most $f(k, \ell)$.*

This shows that the family of tournaments with high dichromatic number seems to act in a similar way to heroes (see Theorem 1.2.1), so the following conjecture, from [NSS23a] seems natural, as it holds by replacing heroes with tournaments with high dichromatic number.

Conjecture 1.2.4 ([NSS23a]). *For all $c \geq 0$, there exists $d \geq 0$ such that if T is a tournament with $\vec{\chi}(T) \geq d$, there are two sets $A, B \subseteq V(T)$ such that $\vec{\chi}(T[A]), \vec{\chi}(T[B]) \geq c$ and all arcs between A and B go from vertices of A to vertices of B .*

The authors also show that this implies a prior conjecture on undirected graphs that was posed by El-Zahar and Erdős [EE85].

Conjecture 1.2.5 ([EE85]). *For all integers $t, c \geq 1$, there exists $d \geq 1$, such that if a graph G satisfies $\chi(G) \geq d$, and has no clique with t vertices (i.e., $\omega(G) < t$), then there are subsets $A, B \subseteq V(G)$ with $\chi(G[A]), \chi(G[B]) \geq c$, such that there are no edges between A and B .*

This conjecture on graphs can be seen as part of a larger body of work that looks at statements of the form:

“Every graph with sufficiently large chromatic number contains either a complete subgraph on t vertices or an induced ***.”

We can look at what can be put in place of the asterisks [Sco22]. Here, replacing the asterisks with “two anticomplete sets with high chromatic number” yields the conjecture of El-Zahar and Erdős. This is also related to the Erdős-Hajnal conjecture, as any single graph G that could be placed in the asterisks would then have the EH property, since every graph that does not contain G must either have low chromatic number, and thus a large independent set, or a large clique.

1.3 Complexity of coloring promise problems

The algorithmic problem of coloring tournaments can also be compared to the problem of coloring undirected graphs. For the latter, deciding if a graph is 2-colorable (i.e., bipartite) is easy, but it is NP-hard to decide if a graph is 3-colorable. A well-studied promise problem is that we are given a graph promised to be 3-colorable and the goal is to color it (in polynomial time) with few colors [Wig83, Blu94, KMS98, KT17]. For tournaments, it is easy to decide whether or not a tournament is 1-colorable (i.e., transitive), since this is exactly when the tournament is acyclic. However, deciding if a tournament is 2-colorable is already NP-hard [CHZ07].

This suggests the following promise problem: Given a tournament promised to be 2-colorable, what is the fewest number of colors with which it can be colored in polynomial time? This question is the starting point for this part of the thesis and naturally leads to related problems of determining upper and lower bounds for coloring various classes of tournaments. For comparison, the complexity landscape of graph coloring is well studied and we have a general understanding of what it looks like. We present an overview of the state-of-the-art for these problems in Table 1.1, where the lower bound should be read as, “It is hard to color a 3-colorable graph

Graph Type	Lower Bound	Upper Bound
3-Colorable graphs	5 [BKO19], $O(1)^*$ [GS20b]	$\tilde{O}(n^{0.19996})$ [KT17]
k -Colorable graphs, $k \geq 3$	$2k - 1$ [BKO19], $O(1)^*$ [GS20b]	$O(n^{1-\frac{3}{k+1}})$ [KMS98]
General graphs	$n^{1-\epsilon}$ [Has99, Zuc06]	$O(n(\log \log n)^2(\log n)^{-3})$ [Hal93]
3-Uniform 2-colorable hypergraphs	$O(1)$ [DRS05]	$\tilde{O}(n^{\frac{1}{5}})$ [KNS01]

Table 1.1: Best known lower and upper bounds for various graph coloring problems. All inapproximability results are under the assumption $P \neq NP$ except those denoted by *, which are under the d -To-1 Conjecture [Kho02].

with 5 colors,” and the upper bound as, “A 3-colorable graph can be (efficiently) colored with $\tilde{O}(n^{0.19996})$ colors.” In contrast, the problem of coloring tournaments has been studied very little from the algorithmic or complexity perspective. In this thesis, we make an effort to address this disparity.

The problem of coloring a 2-colorable tournament with few colors is a special case of coloring a 2-colorable 3-uniform hypergraph with few colors. Deciding if a 3-uniform hypergraph is 2-colorable is NP-hard [Lov73] and more recently it was proved to be NP-hard to color with any constant number of colors [DRS05]. On the positive side, a 2-colorable 3-uniform hypergraph can be colored in polynomial time with $\tilde{O}(n^{1/5})$ colors [AKMR96, CF96, KNS01], a result which uses tools from and is analogous to that of [KMS98] for 3-colorable graphs. Thus, $\tilde{O}(n^{1/5})$ is the best-known upper bound on the number of colors needed to efficiently color a 2-colorable tournament. Deciding if a tournament is 2-colorable is NP-hard [CHZ07] and furthermore, deciding if a tournament is k -colorable for any $k \geq 2$ is NP-hard [FGSY19]. It is consistent with these results that we can, say, efficiently color a 2-colorable tournament with three colors.

1.4 Organization and main results of Part I

In Chapters 2 and 3 of Part I, we consider some basic algorithmic and computational complexity questions on the subject of coloring tournaments. Our main algorithmic tool, presented in Section 2.1, is a decomposition lemma which can be used to obtain efficient algorithms for coloring tournaments in various cases when certain conditions are met. On a high level, it bears some resemblance to decompositions previously used to prove bounded dichromatic number in tournaments and in dense digraphs

Tournament Type	Lower Bound	Upper Bound
2-Colorable tournaments	2[CHZ07], 3	10
3-Colorable tournaments	5, $O(1)$ *	$\tilde{O}(n^{0.19996})$
k -Colorable tournaments, $k \geq 2$	$2k - 1$, $O(1)$ *	$5 \cdot f(k - 1) \cdot g(k)$
2-Colorable light tournaments	in P?	5
Light tournaments	in P?	8
General tournaments	$n^{\frac{1}{2} - \epsilon}$ †	$n / \log n$ [EM64]

Table 1.2: Best known polynomial time inapproximability results and approximation algorithms for various tournament coloring problems. Lower bounds are under the assumption $P \neq NP$ except those marked with a *, which hold under the d -To-1 Conjecture [Kho02]. The function $g(k)$ denotes the number of colors needed to efficiently color a k -colorable graph, while $f(k)$ is the number of colors needed to efficiently color a k -colorable tournament. The entry indicated by † is a hardness of approximation result.

with forbidden subgraphs [BCC⁺13, HLNT19]. To apply our decomposition lemma to 2-colorable tournaments, we use an observation used by [AKMR96, CF96, KNS01] which states that there is an efficient algorithm to partition a 2-colorable tournament into two tournaments that are each light. A *light tournament* is one in which for each arc uv , the set of vertices $N(uv) = \{w \mid uvw \text{ forms a directed triangle}\}$ is transitive. (Let C_3 denote a directed triangle. A light tournament is H -free where H is the hero $\Delta(C_3, T_1, T_1)$.)

In fact, due to this observation and the fact that [BCC⁺13] showed that light tournaments have constant dichromatic number, it cannot be NP-hard (unless NP=co-NP) to color a 2-colorable tournament with $O(1)$ colors. This does not however immediately imply that there is an efficient algorithm, since there are many search problems that are believed to be intractable even though their decision variant is easy (e.g., those in the class TFNP).

Like some other lemmas which show that the dichromatic number of a tournament is bounded (i.e., constant) if the out-neighborhoods of vertices have bounded dichromatic number [HLTW19], our decomposition lemma also has a local-to-global flavor: If the sets $N(uv)$ can be efficiently colored with few colors for all arcs uv and if there are two vertices s and t such that the out-neighborhood of s and the in-neighborhood of t can be efficiently colored with few colors, then our decomposition lemma yields an efficient algorithm to color the whole tournament with few colors.

We present and give applications of our algorithmic decomposition lemma in Chapter 2. Specifically, we show that 2-colorable tournaments can be efficiently colored with ten colors. We then use our toolbox to study 3-colorable tournaments. Here we show that the problem of coloring a 3-colorable tournament has a constant-factor reduction to the problem of coloring 3-colorable graphs.

Next, we strengthen the lower bounds by showing in Chapter 3 that it is NP-hard to color a 2-colorable tournament with three colors. We then give a reduction from coloring graphs to coloring tournaments, which implies, for example, that it is hard to color 3-colorable tournaments with $O(1)$ colors under the d -To-1 Conjecture of Khot [Kho02]. Then we show that it is NP-hard to approximate the number of colors required for a general tournament to within a factor of $O(n^{1/2-\epsilon})$ for any $\epsilon > 0$.

Thereafter, we extend our algorithm to different classes of tournaments and digraphs in Chapter 4. We start by proving that light tournaments can be efficiently colored with eight colors, and then extend this result to a class of tournaments that generalizes light tournaments. These are the H -free tournaments where H belongs to a class recursively constructed as $\Delta(B, T_1, T_1)$ where B is a tournament in the class, starting from a single vertex tournament. We also show that our algorithm can be adapted to efficiently color 2-colorable digraphs with bounded independence number, and triangle-free digraphs with independence number 2.

We provide a summary of the best algorithmic and complexity bounds we find for different classes of tournaments in Table 1.2. All the results are established in this Thesis, except for those indicated by a citation.

Finally, in Chapter 5, we use the decomposition lemma to derive structural results about tournaments and digraphs with bounded independence number. We show that if the sets $N(uv)$ can be efficiently colored with a constant number of colors for all arcs uv , then the entire tournament or digraph can be colored with a bounded number of colors. This result then allows us to prove the equivalence between Conjecture 1.2.4, and Conjecture 1.2.5. Specifically, we prove that Conjecture 1.2.5 implies Conjecture 1.2.4, and for completeness provide the proof of the other direction by [NSS23b]. The equivalence between these conjectures relates the structure of graphs and tournaments with high chromatic number.

This part is joint work with Alantha Newman. The work in Chapters 2 and 3 is from a paper published in the proceedings of the 2023 edition of the European Symposium on Algorithms (ESA) [KN23b]. The work in Chapter 5 is a paper under submission [KN23a].

1.5 Notation

Let $T = (V, A)$ be a tournament with vertex set V and arc set A . Sometimes, we use $V(T)$ to denote its vertex set and $A(T)$ to denote its arc set. For $S \subset V$, we use $T[S]$ to denote the subtournament induced on vertex set S , although we sometimes abuse notation and refer to the subtournament itself as S . We define $uv \in A$ to be an arc directed from u to v . We define $N^+(v)$ to be all $w \in V$ such that arc $vw \in A$ and $N^-(v)$ to be all $w \in V$ such that arc $wv \in A$. We let $N^+[v] = N^+(v) \cup \{v\}$ and $N^-[v] = N^-(v) \cup \{v\}$. For $S \subset V$, we define $N^+(S) = \bigcup_{v \in S} N^+(v)$, and we define $N^-(S), N^+[S], N^-[S]$ analogously. $N^\pm(S)$ will be used to refer to the mixed neighborhood of S , which is the set of vertices that have at least one in-neighbor and one out-neighbor in S . We use $N^o(S)$ to denote vertices in $V \setminus S$ that have at least one in-neighbor and at least one out-neighbor in S . Sometimes we refer to $N^o(S)$ of a set as its *non-neighborhood*.

For $S, U \subset V$ such that $S \cap U = \emptyset$, we use $S \Rightarrow U$ to indicate that all arcs between S and U are directed from S to U . We say $S \leftrightarrow U$ if there exists at least one arc from S to U and at least one arc from U to S (i.e., it is neither the case that $S \Rightarrow U$ nor the case that $U \Rightarrow S$). Let C_3 denote a directed triangle; usually, we refer to this simply as a triangle. Define $N(uv) \subset V$ to contain all vertices w such that uvw forms the directed triangle consisting of arcs uv, vw and wu . In other words, $N(uv) = N^-(u) \cap N^+(v)$. For three tournaments T_1, T_2 and T_3 , we use $\Delta(T_1, T_2, T_3)$ to denote the tournament resulting from adding all arcs from T_1 to T_2 , all arcs from T_2 to T_3 and all arcs from T_3 to T_1 .

A tournament $T = (V, A)$ is *k-colorable* if there is a partition of V into k vertex-disjoint sets, V_1, V_2, \dots, V_k , such that $T[V_i]$ is transitive for all $i \in \{1, \dots, k\}$. We use $\vec{\chi}(T)$ to denote the *dichromatic number* of T (i.e., the minimum number of transitive subtournaments into which $V(T)$ can be partitioned). Computing the value $\vec{\chi}(T)$ is in general NP-hard [CHZ07]. We therefore use $\vec{\chi}_c(T)$ to denote the number of colors by which T can be efficiently colored. Our goal is to find upper and lower bounds on $\vec{\chi}_c(T)$.

We remark that we will always assume that a tournament T which we want to color is strongly connected; if this were not the case, we can color each strongly connected component separately. Therefore, each vertex has an out-neighborhood containing at least one vertex.

Chapter 2

Algorithms for coloring tournaments

In this chapter, we will present efficient approximation algorithms for coloring 2 and 3-colorable tournaments (and extend them to k -colorable tournaments). They both rely on the same tournament decomposition, which we present separately in the following section.

2.1 Efficient decomposition for coloring

We present a decomposition for a tournament that can be computed in polynomial time in certain cases and yields efficient methods to color tournaments with few colors in such cases.

Definition 2.1.1. *We define a c -vertex chain $(v_i)_{0 \leq i \leq k}$ of a tournament T the following way: Let v_0 and v_k be a pair of vertices such that $\vec{\chi}_c(N^+(v_0) \cup N^-(v_k)) \leq c$, and let $(v_i)_{0 \leq i \leq k}$ be the vertices in the shortest directed path from v_0 to v_k .*

Additionally, we define an *arc chain* $(e_i)_{1 \leq i \leq k}$ corresponding to a vertex chain, where e_i is the arc from v_{i-1} to v_i . The main idea behind this decomposition is to build zones that can be efficiently colored, and such that all arcs between zones at distance more than four (i.e., *long arcs*) go backwards.

Definition 2.1.2. *Given a c -vertex chain, a path decomposition of a tournament T is defined as:*

- $D_0 = N^+(v_0)$.
- For $1 \leq i \leq k$, $D_i = N(e_i) \setminus (\cup_{0 \leq j \leq i-1} D_j)$.
- $D_{k+1} = N^-(v_k) \setminus (\cup_{0 \leq j \leq k} D_j)$.

First we prove that this is indeed a decomposition of T .

Claim 2.1.1. *Let $T = (V, A)$ be a tournament, and D be a path decomposition of T . Then $V = \cup_{0 \leq i \leq k} D_i$.*

Proof. We will prove this claim by contradiction: Suppose there is a vertex $w \in V$ that does not belong to any D_i . Assume that w does not belong to the vertex chain. Since w is neither in D_0 nor in D_{k+1} , then $w \in N^-(v_0)$ and $w \in N^+(v_k)$. Take the smallest integer i such that $w \in N^+(v_i)$. There must be one since $w \in N^+(v_k)$. Notice that $i \geq 1$ since $w \notin N^+(v_0)$, so e_i belongs to the arc chain and $w \in N(e_i)$. Therefore, $w \in D_i$, which is a contradiction.

Now consider the case in which w is in the vertex chain. An arc with both endpoints in the vertex chain that is not in the arc chain is backwards. Thus, $v_i \in N(e_{i+2})$ for all $0 \leq i \leq k-2$. Notice that v_{k-1} can belong to D_{k+1} (if it does not belong to D_j for some $j < k+1$). Finally, $v_k \in N(e_{k-1})$. \square

We remark that, for the sake of simplicity and to more easily visualize the decomposition, it might be easier to not include the vertices in the vertex chain in the path decomposition. In this case, these vertices can be colored with two extra colors: Since all arcs not in the arc chain with both endpoints in the vertex chain go backwards (with respect to the arc chain), we can use two colors so that all forwards arcs (those in the arc chain) are bicolored.

Claim 2.1.2. *Let $0 \leq i, j \leq k+1$ and let $j \geq i+5$. For $u \in D_i$ and $w \in D_j$, we have $u \in N^+(w)$.*

Proof. We will prove this by contradiction. Suppose $j \geq i+5$ and $u \in N^-(w)$. Then there is a path of three arcs from v_i to v_{j-1} , namely (v_i, u, w, v_{j-1}) . (By definition of the decomposition, $u \in D_i$ implies $u \in N^+(v_i)$ and $w \in D_j$ implies $w \in N^-(v_{j-1})$.) This is not possible since by the definition of the vertex chain as the shortest path, there can be no path between v_i and v_{j-1} with fewer than four arcs (since $(j-1) - i \geq (i+5-1) - i = 4$). \square

Lemma 2.1.3. *If T has a c -vertex chain that can be found in polynomial time and if $\vec{\chi}_c(N(e)) \leq c$ for each arc e , then $\vec{\chi}_c(T) \leq 5c$.*

Proof. Given a c -vertex chain, we construct a path decomposition. We make five palettes of c colors each with labels from 0 to 4. We color each D_i using the color palette with label $i \bmod 5$. Let us show that we can do this in polynomial time. First, note that the set of colors used is of size c for every D_i . Then, let us consider D_0 : $N^+(v_0)$ can be colored efficiently with c colors by definition of a vertex chain. Similarly, D_{k+1} is a subset of $N^-(v_k)$ and can thus also be efficiently colored with c colors. Finally, for every $1 \leq i \leq k$, D_i is a subset of $N(e_i)$, which can be colored efficiently with c colors by the condition of the lemma.

Our goal is now to prove that this is a proper coloring of T . We will do this by showing that all forward arcs between different D_i are bicolored. By Claim 2.1.2, there are no forward arcs between D_i and D_j when $j \geq i + 5$. Furthermore, by the definition of the coloring, no vertex in D_i and D_j can share a color for $i + 1 \leq j \leq i + 4$. Thus all forward arcs from D_i to D_j will be bicolored. Since every D_i is properly colored, and all forward arcs between different D_i are bicolored, T is properly colored. \square

The next lemma has essentially the same proof as Lemma 2.1.3.

Lemma 2.1.4. *If T has a c -vertex chain that can be found in polynomial time and if $\vec{\chi}_c(N(e)) \leq d$ for each arc e and if $c > d$, then $\vec{\chi}_c(T) \leq c + 4d$.*

Proof. We find the path decomposition using the c -vertex chain. We can color the set $S = D_0 \cup D_{k+1}$ with c colors and the remaining sets D_i for $1 \leq i \leq k$ with d colors each. For the last $c - d$ of the colors used for S , we can remove these vertices from S since these colors will not be used again and call the remaining vertices in S (colored with the first d colors) S' . For the remaining vertices in S , we decompose them into $D_0 := D_0 \cap S'$ and $D_{k+1} := D_{k+1} \cap S'$. Now we have sets D_0, D_1, \dots, D_{k+1} each colored with d colors. We color these sets using five color palettes of d colors each and use the palette $i \bmod 5$ for set D_i . By Claim 2.1.2, this does not create any monochromatic forward arcs. Thus, the total number of colors used is $(c - d) + 5d = c + 4d$. \square

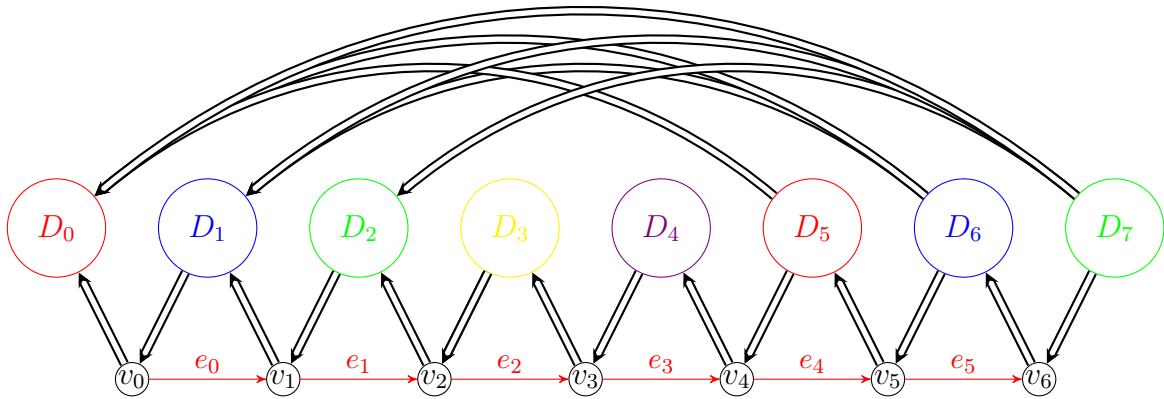


Figure 2.1: Path decomposition of T . The red arcs (e_i) form a shortest path from v_0 to v_k , thus all the arcs not depicted between the v_i 's go backward. All the vertices in a given D_i are colored from the color palette indicated by the color of the D_i . Notice that because there are no long forward arcs between the D_i 's, all arcs between D_i 's that share a color palette are backwards.

2.2 2-colorable tournaments

A tournament $T = (V, A)$ is *2-colorable* if $\vec{\chi}(T) = 2$, and a 2-coloring of tournament T is a partition of V into two vertex sets, V_1 and V_2 , such that $T[V_1]$ and $T[V_2]$ are each transitive. In this section, our goal is to prove Theorem 2.2.1.

Theorem 2.2.1. *Let T be a 2-colorable tournament. Then $\vec{\chi}_c(T) \leq 10$.*

We say an arc uv in A is *heavy* if there exist three vertices $a, b, c \in N(uv)$ which form a triangle abc . If a tournament contains no heavy arcs, then it is *light*. We will use the following observation.

Observation 2.2.2. *Let T be a 2-colorable tournament. Then T can be partitioned into two light subtournaments T_1 and T_2 such that $\vec{\chi}_c(T) \leq \vec{\chi}_c(T_1) + \vec{\chi}_c(T_2)$.*

This observation appears in [AKMR96, CF96, KNS01] where it is stated more generally for 2-colorable 3-uniform hypergraphs. We include a proof here for completeness.

Claim 2.2.3. *In a 2-coloring of a tournament T , each heavy arc must be 2-colored.*

Proof. If u and v are both, say, blue, then each vertex in $N(uv)$ would be red, forcing a triangle in $N(uv)$ to be all red (i.e., monochromatic), which is not possible in a 2-coloring. \square

Corollary 2.2.4. *In a 2-colorable tournament, the heavy arcs form a bipartite graph.*

Now we can prove Observation 2.2.2.

Proof of Observation 2.2.2. All heavy arcs can be easily detected. By Corollary 2.2.4, the set of heavy arcs forms a bipartite graph. The vertex set of this bipartite graph can be colored with two colors (red and blue), such that the tournament induced by each color does not contain a heavy arc. Then we partition the vertices into two sets one containing all the blue vertices and the other containing all the red vertices. The uncolored vertices can go in either set. Since neither of these sets contains any heavy arcs, we can partition the vertices of a 2-colorable tournament into two light subtournaments. \square

Theorem 2.2.1 will follow from Observation 2.2.2 and the following theorem.

Theorem 2.2.5. *Let T be a 2-colorable light tournament. Then $\vec{\chi}_c(T) \leq 5$.*

Our goal is to use Lemma 2.1.3 to prove Theorem 2.2.5. In other words, we want to show that a 2-colorable light tournament has a 1-vertex chain. We first prove a useful claim.

Claim 2.2.6. *Let T be a k -colorable tournament. Then there exist vertices u and w such that $N^+(u) \cup N^-(w)$ is $(k - 1)$ -colorable.*

Proof. Since $T = (V, A)$ is k -colorable, there exist k transitive sets X_1, \dots, X_k such that $V = \cup_{i=1}^k X_i$. Then take u to be the vertex in X_1 that has only incoming arcs from other vertices in X_1 (i.e., the sink vertex for X_1). Similarly, take w to be the vertex in X_1 that has only outgoing arcs to other vertices in X_1 (i.e., the source vertex for X_1). The out-neighborhood of u and the in-neighborhood of w are both subsets of $V \setminus X_1$, and thus so is their union, which is therefore $(k - 1)$ -colorable. \square

Now we are ready to prove that we can find a 1-vertex chain.

Claim 2.2.7. *Let T be a 2-colorable, light tournament. Then T contains a 1-vertex chain that can be found in polynomial time.*

Proof. By Claim 2.2.6, there exist u and w such that $N^+(u) \cup N^-(w)$ is transitive. To find them, we can test the transitivity of $N^+(u) \cup N^-(w)$ for every pair of vertices in T . Then we simply need to find a shortest path from u to w , which can be done in polynomial time. Let k denote the length of the path, and define $v_0 = u$, $v_k = w$, and $(v_i)_{1 \leq i \leq k-1}$ the rest of the vertices in the path. \square

The proof of Theorem 2.2.5 follows from Claim 2.2.7, Lemma 2.1.3 and the fact that $\vec{\chi}_c(N(e)) \leq 1$ for every arc e in a light tournament.

2.3 3-colorable tournaments

Coloring 3-colorable tournaments turns out to be closely related to coloring 3-colorable graphs. This seems surprising since the techniques for 3-colorable graphs were applied to coloring 2-colorable 3-uniform hypergraphs, which are a generalization of 2-colorable tournaments.

We will first show that we can adapt ideas of [Wig83] and [Blu94] to the problem of coloring 3-colorable tournaments by using our algorithm for coloring 2-colorable tournaments with ten colors as a subroutine.

Lemma 2.3.1. *A 3-colorable tournament can be colored with $O(\sqrt{n})$ colors in polynomial time.*

Proof. Let $T = (V, A)$ be a 3-colorable tournament. Notice that T has at least three vertices each of whose out-neighborhoods is 2-colorable. To see this, consider any proper 3-coloring of T . Each color spans a transitive subtournament and each transitive subtournament has a sink vertex that has outgoing arcs only towards the other two colors.

For any vertex, if its out-neighborhood is 2-colorable, we can color its out-neighborhood with 10 colors by Theorem 2.2.1. So we can try to run the algorithm for the out-neighborhood of every vertex, and the algorithm will successfully produce a 10-coloring of the out-neighborhood of at least three vertices.

Therefore, if the minimum outdegree is at least \sqrt{n} , we find a transitive set of size at least $\sqrt{n}/10$. On the other hand, if the minimum outdegree is smaller than \sqrt{n} , we will make progress another way. In this case, let u be a vertex with outdegree smaller than \sqrt{n} . Then, we add u to a set S , and continue the algorithm on the subtournament of T induced on $V \setminus N^+[u]$. We continue this until we find a transitive

subtournament of size at least $\sqrt{n}/20$ or until we have removed half the vertices. In the first case, we will have found a transitive set of size $\Omega(\sqrt{n})$, and in the second case, the set S will be transitive, and also of size $\Omega(\sqrt{n})$.

In conclusion, since we can find a transitive set of size $\Omega(\sqrt{n})$ in polynomial size, we can repeat the procedure recursively to find a coloring with $O(\sqrt{n})$ colors in polynomial time (see [Blu94] for example). \square

We can also use the decomposition of Section 2.1 to get a coloring with fewer colors based on a reduction to coloring 3-colorable graphs.

Theorem 2.3.2. *If we can efficiently color a 3-colorable graph G with k colors, then we can efficiently color a 3-colorable tournament with $50k$ colors.*

Proof. Let $T = (V, A)$ be a 3-colorable tournament. For every arc $e \in A$, try coloring $N(e)$ with 10 colors using Theorem 2.2.1. If the algorithm fails, the neighborhood of the edge is not 2-colorable, and thus the edge is not monochromatic in any 3-coloring. Let $F \subset E$ denote the set of arcs whose neighborhoods cannot be colored with 10 colors using our algorithm. Ignore the direction of the arcs in F and consider the graph $G = (V, F)$. This graph must be 3-colorable, since no arc in F is monochromatic in any 3-coloring of T .

Now let us show that from a coloring of G with k colors, we can obtain a coloring of T with $50k$ colors. Consider a coloring of the graph $G = (V, F)$ and let V_i be the vertices colored with color i in this coloring. Consider the induced subtournament $T' = T[V_i]$; it has no arc in F and thus the neighborhood of every arc in this tournament can be colored efficiently with 10 colors. Furthermore, by Claim 2.2.6 and Theorem 2.2.1, there are vertices u and v in T' such that $N_{T'}^+(u) \cup N_{T'}^-(v)$ is efficiently 10-colorable. So by Lemma 2.1.3, we can efficiently color T' with 50 colors. We can do this for the subtournament $T[V_i]$ for each of the i colors used to color G .

\square

Combining this Lemma with approximation algorithm [KT17], which colors a 3-colorable graph with fewer than $n^{\frac{1}{5}}$ colors, we obtain the same asymptotic bound for 3-colorable tournaments.

Corollary 2.3.3. *Let T be a 3-colorable tournament on n vertices. Then, $\vec{\chi}_c(T) \leq O(n^{.19996})$.*

We can extend Theorem 2.3.2 to a more general case.

Lemma 2.3.4. *Let f and g be functions such that we can efficiently color k -colorable graphs (respectively, k -colorable tournaments) with $g(k)$ (respectively, $f(k)$) colors. Then $f(k) \leq 5 \cdot f(k-1) \cdot g(k)$.*

Proof. We use the same reduction as in the proof of Theorem 2.3.2, but now F is the set of arcs whose neighborhoods cannot be efficiently $f(k-1)$ -colored. Then each V_i in G is colored with $5 \cdot f(k-1)$ colors. So we need a total of $5 \cdot f(k-1) \cdot g(k)$ colors. \square

2.4 Certificates of non-2-colorability

In Section 2.2, we presented an algorithm to color a 2-colorable tournament with ten colors. Suppose we run this algorithm on a tournament T that is *not* 2-colorable. Then it will either color T with ten colors or it will produce at least one certificate that T is not 2-colorable. A certificate will have the following form: either a) there is an odd cycle of heavy arcs in T , or b) for every ordered pair of vertices (u, v) , the subtournament $T[N^+(u) \cup N^-(v)]$ is not transitive. In particular, an 11-chromatic tournament must contain such a certificate.

Chapter 3

Complexity of coloring tournaments

In this chapter, we examine the hardness of approximate coloring of tournaments. [CHZ07] showed that deciding if a tournament can be 2-colored is NP-hard. For completeness, we provide a simplified (though similar) proof of this result in 3.1. Later, [FGSY19] proved that for any k , it is NP-hard to decide if a tournament is k -colorable.

3.1 NP-hardness of deciding 2-colorability

For completeness, we start by providing a proof of the NP-hardness of coloring 2-colorable tournaments with two colors. This proof is strongly inspired by the proof of [CHZ07].

Lemma 3.1.1. *It is NP-hard to decide if a tournament has chromatic number two.*

Proof. We will reduce this problem to the problem of deciding 2-colorability of 3-uniform hypergraphs, which is known to be NP-hard [DRS05].

Let $H = (V_H, E_H)$ be a 3-uniform hypergraph. We now build a tournament $T = (V, A)$ such that T is 2-colorable iff H is 2-colorable.

We will start by defining a subtournament $T_1 = (V_1, A_1)$ of T . Given an enumeration of the hyperedges of H , $e_i = (v_a, v_b, v_c)$, we will add three vertices $v_{a,i}$, $v_{b,i}$ and

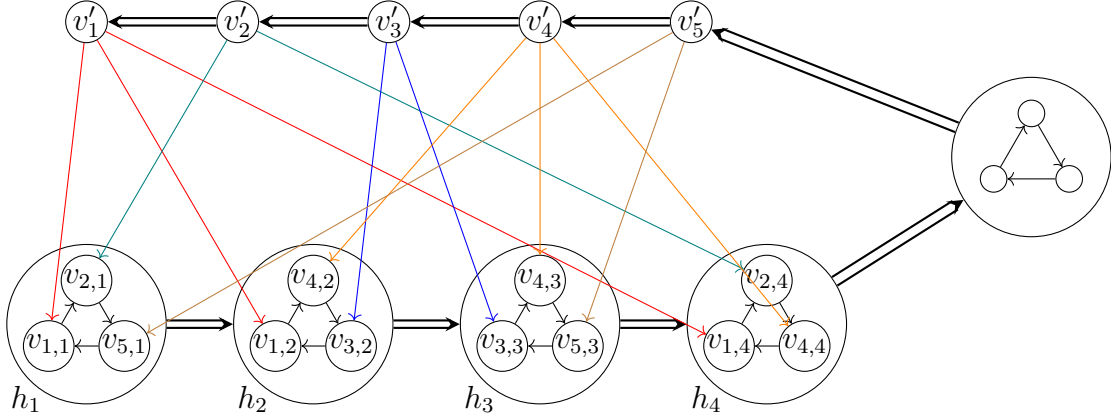


Figure 3.1: Construction of T from a 3-uniform hypergraph H . There is a downward arc between v'_b and all vertices $v_{b,i}$ for every b, i . These are the colored arcs in the figure. All remaining arcs all go from the vertices $v_{a,i}$ towards the vertices v'_b for $a \neq b$ (they go up).

$v_{c,i}$ to V_1 , and add to A_1 the arcs $(v_{a,i}, v_{b,i})$, $(v_{b,i}, v_{c,i})$ and $(v_{c,i}, v_{a,i})$ such that these three vertices form a directed triangle. We then add the arcs from all the vertices $v_{a,i}$ towards all the vertices $v_{b,j}$ for any a, b, i, j with $i < j$. We now define a new subtournament $T_2 = (V_2, A_2)$ made up of three vertices that form a directed triangle. Finally, we define a last subtournament $T_3 = (V_3, A_3)$: $V_3 := V_H$, and T_3 forms a transitive set on its vertex set.

Then add T_1 , T_2 and T_3 to T . Orient all arcs from vertices in V_1 towards vertices of V_2 and all arcs from vertices of V_2 towards vertices of V_3 . The only arcs we still need to orient are those between V_1 and V_3 . For this, we look at the vertices of H from which the vertices of V_1 and V_3 are derived; for $v_{a,i} \in V_1$ and $v'_b \in V_3$, we add an arc from v'_b to $v_{a,i}$ iff $a = b$ (i.e., if they are derived from the same vertex of H), and we add an arc from $v_{a,i}$ to v'_b otherwise. This completes the definition of T . Figure 3.1 gives an example of this construction for a hypergraph with five vertices and four hyperedges.

We will now establish that if H is 2-colorable, so is T . Given a 2-coloring of H , give all the vertices of V_1 the same color as the vertex of H they are derived from, and those in V_3 the opposite color of the vertex of H they are derived from. Finally color T_2 properly with the same 2 colors. Then, any arc that goes from V_3 to V_1 will be 2-colored, and since all arcs are oriented from V_1 towards V_2 and from V_2 towards V_3 , there can only be monochromatic triangles inside V_1 , V_2 or V_3 . However, V_2 is a

bicolored triangle, and, every triangle in V_1 and V_3 represents a hyperedge of H and must therefore contain two vertices of different colors.

It remains to show that if T has a proper 2-coloring C , then we can construct a proper 2-coloring of H .

We define a coloring C_H of H by assigning to every vertex $v_a \in V_H$ the same color as its corresponding vertex $v'_a \in V_3$ has in C . Let us show that C_H is a proper 2-coloring of H . Notice that in a proper 2-coloring of H , $v_{a,i} \in V_1$ must have the opposite color of $v'_a \in V_3$, for any a, i . If it were not the case they would form a directed triangle with the vertex in V_2 of the same color, since T_2 is a directed triangle and must therefore be bicolored. Now suppose some hyperedge $e_i = (v_a, v_b, v_c)$ is monochromatic under C_H . Then $v'_a, v'_b, v'_c \in V_3$ all have the same color. Then, there is a triangle $(v_{a,i}, v_{b,i}, v_{c,i})$ in T_1 by definition, and all its vertices must have the same color (the opposite of that used for v'_a, v'_b, v'_c). This is a contradiction, thus all hyperedges of H are bicolored, and C_H is proper. \square

3.2 NP-Hardness of approximate coloring of k -colorable tournaments

We will first improve upon these NP-hardness results and then show hardness of coloring k -colorable tournaments for $k \geq 3$ with $O(1)$ colors under the d -To-1 conjecture. The d -To-1 conjecture was first introduced by Khot alongside the famous Unique Games conjecture [Kho02], and has since been used to show hardness of coloring 3-colorable graphs with $O(1)$ colors [GS20b].

First notice that the search problem must be at least as hard as its decision version.

Observation 3.2.1. *Let $k < \ell$ be any two constants. If we can color k -colorable tournaments with ℓ colors, then we can distinguish k -colorable tournaments from tournaments with chromatic number at least $\ell + 1$.*

This comes immediately from the fact that if we could ℓ -color all k -colorable tournaments, then we could see that they do not have chromatic number $\ell + 1$ or greater. The hardness of distinguishing between chromatic number k and greater or equal to $\ell + 1$ is therefore commonly established as a way of implying the hardness of coloring k -colorable graphs with ℓ colors (see for example [BKO19]).

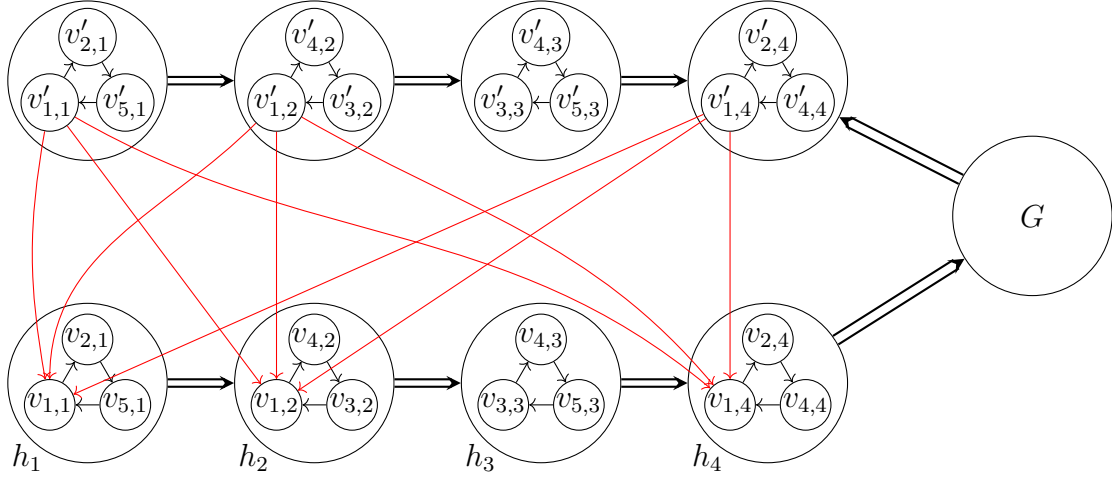


Figure 3.2: Construction of T from a 3-uniform hypergraph H . The edges in red (going down) were represented only for vertex v_1 , but there is an arc from any vertex $v'_{a,i}$ towards all vertices $v_{a,j}$ for any j . The remaining arcs all go from the vertices $v_{a,i}$ towards the vertices $v_{b,j}$ for $a \neq b$ (they go up).

We now prove that it is NP-hard to 3-color a 2-colorable tournament.

Theorem 3.2.2. *For a tournament T , it is NP-hard to distinguish between the case in which $\bar{\chi}(T) = 2$ and the case in which $\bar{\chi}(T) \geq 4$.*

Proof. Let H be a 3-uniform hypergraph. In [FGSY19] and [CHZ07], they show how to construct a tournament G such that G is 2-colorable iff H is 2-colorable. We will build a new tournament $T = (V, A)$ such that if H is 2-colorable, T is also 2-colorable, and if H has chromatic number at least 7, then T has chromatic number at least 4. (Notice that it is NP-hard to color a 3-uniform 2-colorable hypergraph with c colors for any constant c [DRS05].)

We will start by defining a subtournament $T_1 = (V_1, A_1)$ of T . Given an enumeration of the hyperedges of H , $e_i = (v_a, v_b, v_c)$, we will add three vertices $v_{a,i}$, $v_{b,i}$ and $v_{c,i}$ to V_1 , and add to A_1 the arcs $(v_{a,i}, v_{b,i})$, $(v_{b,i}, v_{c,i})$ and $(v_{c,i}, v_{a,i})$ such that these three vertices form a directed triangle. We then add the arcs from all the vertices $v_{a,i}$ towards all the vertices $v_{b,j}$ for any a, b, i, j with $i < j$. We make a copy of T_1 , that we call $T_2 = (V_2, A_2)$, and add both to T . We then add the tournament G , and orient all arcs from vertices in V_1 towards vertices of G , and all arcs from vertices of G towards vertices in V_2 . The only arcs we still need to orient are those between

V_1 and V_2 . For this, we look at the vertices of H from which the vertices of V_2 are derived; for $v_{a,i} \in V_1$ and $v'_{b,j} \in V_2$, we add an arc from $v'_{b,j}$ to $v_{a,i}$ iff $a = b$ (i.e., if they are derived from the same vertex of H), and we add an arc from $v_{a,i}$ to $v'_{b,j}$ otherwise. This completes the definition of T .

We will now establish that if H is 2-colorable, so is T . Given a 2-coloring of H , give all the vertices of V_1 the same color as the vertex of H they are derived from, and those in V_2 the opposite color of the vertex of H they are derived from. Finally color G with the same 2 colors. Then, any arc that goes from V_2 to V_1 will be 2-colored, and since all arcs are oriented from V_1 towards G and from G towards V_2 , there can only be monochromatic triangles inside V_1 , V_2 or G . However, G is properly 2-colored and thus does not have any monochromatic triangles. Furthermore, every triangle in V_1 and V_2 represents a hyperedge of H and must therefore contain two vertices of different colors.

It remains to show that if H has chromatic number at least 7, T has chromatic number at least 4. We will establish this by contradiction: We show that if T has a proper 3-coloring C , then we can construct a proper 6-coloring of H .

For every vertex v_a of H , consider the set of vertices $S_a = \{v_{a,i} \mid \forall i, v_{a,i} \in V_1\}$ and $Q_a = \{v_{a,i} \mid \forall i, v_{a,i} \in V_2\}$. A key property of our construction is that if H is not 2-colorable, then in any proper 3-coloring of T , either the set S_a or the set Q_a must be monochromatic. To see this, notice that if any vertex of S_a has the same color as any vertex of Q_a , then they will form a monochromatic triangle with a third vertex from G that has the same color (since G is colored with at least 3 colors). So if S_a and Q_a each use at least 2 out of 3 colors, then at least one color appears in both S_a and Q_a resulting in a monochromatic triangle.

Next we define a coloring C_H of H as follows, by relating each color to an integer. If S_a is monochromatic, then set $C_H(v_a) = C(S_a)$. Otherwise, if Q_a is monochromatic, then set $C_H(v_a) = C(Q_a) + 3$. Now take any hyperedge (v_a, v_b, v_c) of H ; if the three sets S_a , S_b and S_c are monochromatic, then since there is a directed triangle $(v_{a,j}, v_{b,j}, v_{c,j})$ in T_1 for some j , the three vertices cannot have the same color in C , so they also do not all have the same color in C_H . If none of the three sets S_a , S_b and S_c are monochromatic, then the sets Q_a , Q_b and Q_c are each monochromatic, so the same argument applies. Finally, without loss of generality we can suppose S_a is monochromatic but not S_b . Then v_a and v_b do not have the same color in C_H by definition. Therefore, by case analysis, no hyperedge of H can be monochromatic, and thus C_H is a proper 6-coloring of H . \square

Our goal is now to extend this hardness result to k -colorable tournaments. To do this, we will use an iterative construction presented in the following claims.

Claim 3.2.3. *Let a, b, c, d, e, ℓ be positive integers such that $e < c + d$. Let H be a 3-uniform hypergraph, and let R_1, R_2 and R_3 be three tournaments such that if $\chi(H) = 2$, then $\vec{\chi}(R_1) = a$, $\vec{\chi}(R_2) = b$ and $\vec{\chi}(R_3) = a + b$, and if $\chi(H) \geq \ell$, then $\vec{\chi}(R_1) \geq c$, $\vec{\chi}(R_2) \geq d$ and $\vec{\chi}(R_3) \geq e$.*

Then we can construct a tournament R' with chromatic number $\vec{\chi}(R') = a + b$ if $\chi(H) = 2$, and $\vec{\chi}(R') \geq e + 1$ if $\chi(H) \geq \ell$.

Proof. Let H be a hypergraph and let R_1, R_2 and R_3 be three tournaments that satisfy the conditions. Let $R' = \Delta(R_1, R_2, R_3)$. Now we want to show that if $\chi(H) = 2$, $\vec{\chi}(R') = a + b$. Simply color R_1 with a colors, R_2 with a new set of b colors, and R_3 with the same set of $a + b$ colors. These dicolorings will be proper since $\chi(H) = 2$. The dicoloring of R' is proper since there is no monochromatic triangle inside R_1, R_2 or R_3 , and any triangle containing vertices from R_1 and R_2 will have at least two different colors.

Next we want to show that if $\chi(H) \geq \ell$, $\vec{\chi}(R') \geq e + 1$. Suppose R' has a coloring with e colors. Then, since $c + d > e$, R_1 and R_2 must share at least one color. Furthermore, all e colors are used in R_3 by assumption. So there must be a monochromatic triangle since every triplet (u, v, w) with $u \in R_1, v \in R_2, w \in R_3$ forms a directed triangle. Thus, $\vec{\chi}(R') \geq e + 1$. \square

Claim 3.2.4. *Let a, b, c, d, ℓ be positive integers. Let H be a 3-uniform hypergraph, and let R_1 and R_2 be two tournaments such that if $\chi(H) = 2$, then $\vec{\chi}(R_1) = a$ and $\vec{\chi}(R_2) = b$, and if $\chi(H) \geq \ell$, then $\vec{\chi}(R_1) \geq c$, $\vec{\chi}(R_2) \geq d$.*

Then there exists a tournament R' with $\vec{\chi}(R') = a + b$ if $\chi(H) = 2$ and $\vec{\chi}(R') \geq c + d$ if $\chi(H) \geq \ell$.

Proof. We will prove by induction on k with $a + b \leq k \leq c + d$, that there exists a tournament R'_k with $\vec{\chi}(R'_k) = a + b$ if $\chi(H) = 2$, and $\vec{\chi}(R'_k) \geq k$ if $\chi(H) \geq \ell$.

Initialization: For $k = a + b$, define R'_{a+b} to be any tournament with chromatic number $a + b$.

Induction: Suppose that for a fixed $k < c + d$, there is a tournament R'_k verifying the conditions, then let us show that there is a tournament R'_{k+1} that verifies these same conditions for $k + 1$. We apply Claim 3.2.3 where R_1 and R_2 from both claims

are the same, and R_3 is R'_k . This proves the existence of a tournament R'_{k+1} such that $\vec{\chi}(R'_{k+1}) = a + b$ if $\chi(H) = 2$, and $\vec{\chi}(R'_k) \geq k + 1$ if $\chi(H) \geq \ell$.

Now we simply define $R' = R'_{c+d}$.

We remark that as every iteration of the construction can be done in polynomial time, and there are at most $c + d$ iterations, R' can be constructed in polynomial time and has size $|V(R')| \leq (c + d) \cdot (|V(R_1)| + |V(R_2)|) + |V(R_{a+b})|$. \square

This gadget R' will allow us to prove that it is NP-hard to color a k -colorable tournament with $2k - 1$ colors.

Theorem 3.2.5. *For a tournament T , it is NP-hard to distinguish between the case in which $\vec{\chi}(T) = k$ and the case in which $\vec{\chi}(T) \geq 2k$.*

Proof. Given a 3-uniform hypergraph H , we will prove by strong induction on k that for every k , there exists a tournament T_k of size polynomial in $|V(H)|$ such that if $\chi(H) = 2$ then $\vec{\chi}(T_k) = k$, and if $\chi(H) \geq 7$ then $\vec{\chi}(T_k) \geq 2k$.

Initialization: For $k = 2$, we refer to the tournament constructed in the proof of Theorem 3.2.2.

For $k = 3$, let $T_3 = \Delta(T_2, T_2, T_2)$. If $\chi(H) = 2$, coloring the first copy with colors 1, 2, the second with colors 2 and 3, and the third with colors 3 and 1 yields a 3-coloring. This tournament is also not 2-colorable since in any 2-coloring, all copies of T_2 must use the same two colors, and thus there would be a monochromatic directed triangle.

If $\chi(H) \geq 7$, T_2 has chromatic number at least 4. Therefore, in any 5-coloring, two colors must be used in every copy of T_2 , which would lead to a monochromatic directed triangle. Therefore, $\vec{\chi}(T_3) \geq 6$.

Induction hypothesis: For every $m \leq k$, there exists a tournament T_m of size polynomial in $|V(H)|$ such that if $\chi(H) = 2$, $\vec{\chi}(T_m) = m$, and if $\chi(H) \geq 7$, $\vec{\chi}(T_m) \geq 2m$.

Induction: Let us show that there exists a tournament T_{k+1} of size polynomial in $|V(H)|$ such that if $\chi(H) = 2$, $\vec{\chi}(T) = k + 1$, and if $\chi(H) \geq 7$, $\vec{\chi}(G) \geq 2(k + 1)$.

Take the two tournaments $T_{\lfloor \frac{k+1}{2} \rfloor}$, $T_{\lceil \frac{k+1}{2} \rceil}$ obtained from the 3-uniform hypergraph H . These obey the conditions of Claim 3.2.4, thus there exists a tournament, that we call T_{k+1} , such that if $\chi(H) = 2$, $\vec{\chi}(T_{k+1}) = k + 1$, and if $\chi(H) \geq 7$, $\vec{\chi}(T_{k+1}) \geq 2(\lceil \frac{k+1}{2} \rceil + \lfloor \frac{k+1}{2} \rfloor) = 2(k + 1)$.

This concludes the induction. It immediately follows that it is NP-hard to distinguish between tournaments with chromatic number k and tournaments with chromatic number $2k$, as being able to do so would allow us to distinguish between 3-uniform hypergraphs with chromatic number 2, and 3-uniform hypergraphs with chromatic number at least 7, which is NP-hard [DRS05, Bha18]. \square

3.3 Reduction from coloring graphs to coloring tournaments

In Section 2.3, we showed that if we can color a 3-colorable graph with k colors, then we can color a 3-colorable tournament with $50k$ colors. In this section, we give a reduction in the other direction, we show that the problem of coloring a k -colorable graph with ℓ colors is reducible to the problem of coloring a k -colorable tournament with ℓ colors. A corollary of this reduction is hardness of coloring tournaments under the d -To-1 Conjecture of Khot [Kho02]; [GS20b] showed that assuming the d -To-1 Conjecture, it is hard to color 3-colorable graphs with $O(1)$ colors, and using our reduction, we can extend this hardness to tournaments.

Theorem 3.3.1. *Given any two constants $k, \ell \geq 3$, if we can efficiently distinguish k -colorable tournaments and tournaments with chromatic number at least ℓ , then we can efficiently distinguish k -colorable graphs and graphs with chromatic number at least ℓ .*

We start by proving the following lemma that presents the building block of the reduction.

Lemma 3.3.2. *Let c be a positive integer, $G = (V_G, E_G)$ a graph and $T = (V_T, A_T)$ a tournament such that $\vec{\chi}(T) = k$ when $\chi(G) = k$, and $\vec{\chi}(T) \geq \min(\chi(G), c)$ when $\chi(G) > k$. We can build a new tournament $U = (V_U, A_U)$ such that $\vec{\chi}(U) = k$ when $\chi(G) = k$, and $\vec{\chi}(U) \geq \min(\chi(G), c + 1)$ otherwise.*

Proof. Let $n_G = |V_G|$ and let $(T_i)_{1 \leq i \leq n_G - 1}$ be copies of T . Let $T_i = (V_i, A_i)$. Then $V_U := (\cup_{1 \leq i \leq n_G - 1} V_i) \cup V_G$. Fix an arbitrary ordering of the vertices in V_G . To build A_U , add the arc from v_j to v_i if $(v_i, v_j) \in E_G$, and the arc from v_i to v_j otherwise (i.e., if $(v_i, v_j) \notin E_G$). The resulting tournament induced on the vertices of V_G is said to have G as a *backedge graph*. Next we add all the arcs from v_i to all vertices of T_j

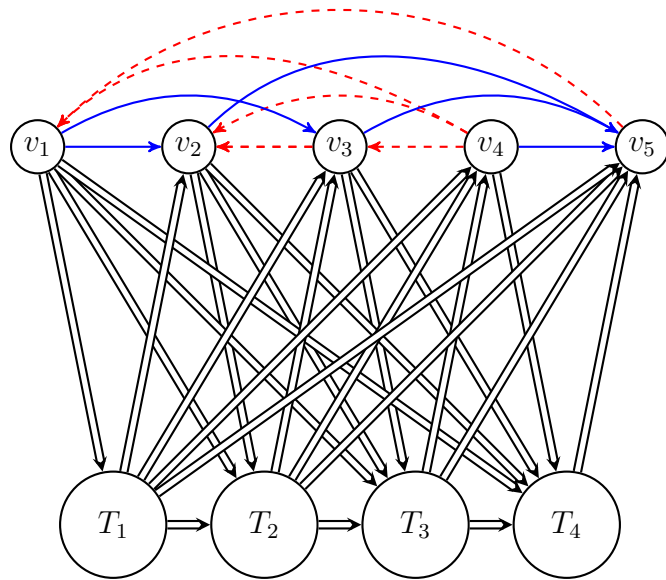


Figure 3.3: Construction of the tournament U from a graph G on five vertices. The dashed red edges are those present in G and all go backwards, whereas the remaining edges are blue and go forwards.

for every $i \leq j$, and the arcs from every vertex of T_i to v_j for all $i < j$. Finally, we add the arcs from any vertex of T_i to any vertex of T_j for every $i < j$. This concludes the construction of U , which is depicted in Figure 3.3.

Suppose $\chi(G) = k$. Then let us show that $\vec{\chi}(U) = k$. In this case, $\vec{\chi}(T) = k$ by assumption. We take a k -coloring of G and a k -coloring of T and color the vertices in U (i.e., in V_G and in V_i for all $1 \leq i \leq n_G - 1$) according to these colorings. Notice that all arcs that are backwards with respect to the order $v_1 \rightarrow T_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_i \rightarrow T_i \rightarrow \dots \rightarrow T_{|V_G|-1} \rightarrow v_{|V_G|}$ are bicolored. To see this, observe that arcs from v_j to v_i for $j > i$ belong to E_G and are therefore bicolored, and by construction, there are no arcs from v_j to T_i nor from T_j to T_i for $j > i$. Thus, there can only possibly be monochromatic triangles within T_i , but these sets are properly colored. Therefore, this is a proper dicoloring of the tournament U and $\vec{\chi}(U) = k$.

Let us now prove that when $\chi(G) > k$, we have $\vec{\chi}(U) \geq \min\{\chi(G), c + 1\}$. By assumption, we have $\vec{\chi}(T) \geq \min\{\chi(G), c\}$ in this case. Thus, if $c = \chi(G)$, then the claim is true, since T is a subtournament of U . So let us consider the case in which $c < \chi(G)$. Then given a coloring of U with c colors, there must be a monochromatic edge (v_i, v_j) in G . Assuming without loss of generality that $i < j$, there is a monochromatic arc from v_j to v_i in U . Furthermore, since $\vec{\chi}(T) \geq c$, there must be some vertex of T_i that has the same color as v_i and v_j . Since all vertices in T_i form a directed triangle with v_i and v_j , this means that there is a monochromatic triangle in U , which is a contradiction. \square

We can then prove Theorem 3.3.1 by a simple induction.

Proof of Theorem 3.3.1. Let $G = (V_G, E_G)$ be a graph and let $\ell \geq 3$ be a constant. For all $c \geq k$, let us build a tournament $T_c = (V_{T_c}, A_{T_c})$ by induction such that if $\chi(G) = k$, then $\vec{\chi}(T_c) = k$, and otherwise if $\chi(G) \geq \ell$, then $\vec{\chi}(T_c) \geq \min\{\chi(G), c\}$.

Initialization: For $c = k$, any k -colorable tournament satisfies the conditions.

Induction: Suppose there is a tournament T_c satisfying the conditions for a constant c . Let us show that there is a tournament T_{c+1} that satisfies these same conditions for $c + 1$. This follows from Lemma 3.3.2 where T_c is T , and T_{c+1} is U .

Consider the tournament T_ℓ ; it is of size $|V_{T_\ell}| = O(|V_G|^{2^\ell})$, which is polynomial for fixed ℓ . Furthermore, if $\chi(G) = k$ then $\vec{\chi}(T_\ell) = k$, and otherwise $\vec{\chi}(T_\ell) \geq \min\{\chi(G), \ell\}$, thus if we can efficiently decide if T_ℓ has chromatic number k or at least ℓ , we can also efficiently decide if G has chromatic number k or at least ℓ . \square

Using the hardness of coloring 3-colorable graphs with a constant number of colors under the d -To-1 conjecture [GS20b], we get equivalent hardness for coloring 3-colorable tournaments, and thus k -colorable tournaments for $k \geq 3$ (since any 3-colorable tournament is also k -colorable when $k \geq 3$).

Corollary 3.3.3. *Let $3 \leq k < \ell$ be any two constants. Then if the d -To-1 conjecture is true, we cannot distinguish between tournaments with chromatic number k and tournaments with chromatic number at least ℓ .*

Notice that if stronger hardness (for example constant hardness under the $P \neq NP$ assumption) were established for approximate coloring of 3-colorable graphs, then this reduction would provide stronger hardness results for 3-colorable tournaments (and thereby also for k -colorable tournaments). This would hold up to constant hardness, after which the blowup of the size of the tournament in the construction would be more than polynomial.

3.4 Hardness of approximation for general tournaments

In this section, our goal is to prove hardness of approximation for tournaments with no promise on the chromatic number. Our proof parallels the proof of hardness of approximate coloring of digon-free digraphs of [FHS19]; we extend their approach to tournaments and show that it can be used to obtain hardness of approximation.

Lemma 3.4.1. *Let ϵ be a constant such that $0 < \epsilon < 1$. There exists a tournament $T = (V, A)$ where $V = X + Y$ with $|X| = |Y| = n$ such that with probability going to 1 as n goes to infinity, for every two subsets $S_X \subseteq X$, $S_Y \subseteq Y$ having $|S_X| \geq n^\epsilon$, $|S_Y| \geq n^\epsilon$, the tournament induced by $S_X \cup S_Y$ contains a triangle.*

Proof. Define $T = (V, A)$ with $V = X \cup Y$ such that X and Y each form transitive tournaments on n vertices. Then orient all the remaining arcs randomly; so for $u \in X$, $v \in Y$, the arc goes from u to v with probability $1/2$.

Given $0 < \epsilon < 1$, take any $S_X \subseteq X$, $S_Y \subseteq Y$ with $|S_X| \geq n^\epsilon$ and $|S_Y| \geq n^\epsilon$. Take $u, v \in S_X$, and $w \in S_Y$. Then the probability of (u, v, w) forming a directed triangle in T is $1/4$, thus the probability of u and v forming no triangle with any vertex in S_Y is at most $(3/4)^{n^\epsilon}$ since the arcs between u and w and between v and

w are oriented independently for every $w \in S_Y$. This probability tends to zero as n goes to infinity. \square

The previous lemma can be derandomized using an explicit construction for bipartite Ramsey graphs [BRSW12].

Lemma 3.4.2. *Let ϵ be a constant such that $0 < \epsilon < 1$. There exists a tournament $T = (V, E)$ where $V = X + Y$ with $|X| = |Y| = n$ such that for sufficiently large n , for every two subsets $S_X \subseteq X$, $S_Y \subseteq Y$ having $|S_X| \geq n^\epsilon$, $|S_Y| \geq n^\epsilon$, the tournament induced by $S_X \cup S_Y$ contains a triangle.*

Proof. Take a sufficiently large n . Then, from Theorem 1.3 in [BRSW12] there exists an explicit construction of a bipartite $o(n)$ -Ramsey graph over n vertices. Let $B_1 = (X_1, Y_1, E_1)$ be such a graph. Then, define the tournament $T = (V, E)$ with $V = X + Y$ in the following way:

- $X = X_1$ and $Y = Y_1$
- Orient the arcs inside X and Y such that they both induce transitive tournaments.
- For every $u \in X$, $v \in Y$, orient the arc from u to v if $(u, v) \in E_1$ and from v to u otherwise.

Given $0 < \epsilon < 1$, take any $S_X \subseteq X$, $S_Y \subseteq Y$ with $|S_X| \geq n^\epsilon$ and $|S_Y| \geq n^\epsilon$. Let $x \in S_X$ and $y \in S_Y$ be the middle vertices of S_X and S_Y (i.e. x has roughly equal in and out-degree in S_X , and y in S_Y). Without loss of generality, suppose that the arc between u and v is oriented from u to v . Then, the graph induced on B_1 by $S_X[N^-(u)]$ and $S_Y[N^+(v)]$ is of size at least $n^\epsilon - 2$, thus it is neither complete nor empty for sufficiently large n (since B_1 is a $o(n)$ -Ramsey graph). This implies that there is an arc from a vertex $y \in S_Y[N^+(v)]$ to a vertex $x \in S_X[N^-(u)]$. Thus, there is a directed cycle (u, v, y, x) in $S_X \cup S_Y$, and since it is a tournament, there is some directed triangle. \square

Theorem 3.4.3. *It is NP-hard to find an acyclic induced subgraph of size greater than $n^{1/2+\epsilon}$ in an n^ϵ -colorable tournament, for every $0 < \epsilon < \frac{1}{4}$.*

Proof. For any $\epsilon > 0$, let $G = (V, E)$ be a graph on n_G vertices, colorable with n_G^ϵ colors. Feige and Kilian [FK96] proved that it is NP-hard to find an independent set of size greater than n_G^ϵ in such graphs.

For each vertex $v_i \in V$, define a new transitive tournament T_i on n_G vertices. For each edge $(v_i, v_j) \in E$, join T_i and T_j such that they form the tournament of Lemma 3.4.2, with T_i being X and T_j being Y . For all remaining $v_i, v_j \in V$ with $i < j$ (such that $(v_i, v_j) \notin E$), orient all arcs from each vertex of T_i to each vertex of T_j . This defines a new tournament T on $n = n_G^2$ vertices. T has an acyclic k -coloring with $k \leq n^{\epsilon/2}$ by coloring each T_i with the color of v_i in a $n_G^\epsilon = n^{\epsilon/2}$ -coloring of G . Indeed, the only arcs that are not bicolored are inside a T_i for some i , or from a vertex of T_i to a vertex of T_j for $i < j$, and can thus never form a triangle. Let S be an acyclic induced subtournament of T . Notice that from Lemma 3.4.2, if S intersects every $(T_i)_{i \in I}$ on more than n^ϵ vertices, then $(v_i)_{i \in I}$ forms an independent set of G . Therefore, if $|S| > 2n^{\frac{1+\epsilon}{2}}$, there must be at least n^ϵ tournaments that intersect S on at least n^ϵ vertices, which then leads to an independent set of size at least n^ϵ in G . \square

The hardness of approximating a coloring in tournaments then comes as a corollary, as it immediately follows that it is NP-hard to distinguish a n^ϵ -colorable tournament from a k -colorable tournament with $k \geq n^{1/2-\epsilon}$, for every $0 < \epsilon < \frac{1}{4}$.

Corollary 3.4.4. *Given any arbitrarily small constant $\epsilon > 0$, it is NP-hard to approximate the chromatic number of tournaments within a factor of $n^{1/2-\epsilon}$.*

Chapter 4

Extensions

In the following chapter, we will start by showing bounds for our algorithm on light tournaments. For completeness we also present an algorithmic version of the decompositions of [BCC⁺13], which gives weaker results for the class of light tournaments. Thereafter, we extend our algorithm to the setting of 2-colorable dense digraphs, and triangle-free digraphs with independence number 2.

4.1 Light tournaments

Light tournaments are exactly those which do not contain the hero $\Delta(1, 1, C_3)$. [BCC⁺13] proved that light tournaments have constant chromatic number, but they did not state a precise constant, and their proof is not algorithmic. A careful modification of their approach can be used to give an algorithmic proof that this constant is around 35. Details are provided in Section 4.2, since they could be useful in finding algorithms for tournaments with other forbidden heroes.

In this section, our goal is to prove the following theorem.

Theorem 4.1.1. *Let T be a light tournament. Then $\vec{\chi}_c(T) \leq 8$.*

Lemma 4.1.2. *Let T be a light tournament. Then we can find u, v such that:*

- (i) $\vec{\chi}_c(N^+(u)) \leq 3$,
- (ii) $\vec{\chi}_c(N^-(v)) \leq 3$, and

(iii) $\vec{\chi}_c(N^-(v) \cup N^+(u)) \leq 5$.

Assuming Lemma 4.1.2, we can prove Theorem 4.1.1.

Proof of Theorem 4.1.1. If the shortest path from u to v has length at least four, then notice that all arcs between $N^+(u)$ and $N^-(v)$ go from $N^-(v)$ to $N^+(u)$. Then by items (i) and (ii) from Lemma 4.1.2, we have $\vec{\chi}_c(N^-(v) \cup N^+(u)) \leq 3$, so T has a 3-vertex chain. By Lemma 2.1.4, we can color T with seven colors.

Next, we consider the case in which the shortest path from u to v has length at most three. Then by item (iii) from Lemma 4.1.2, we have $\vec{\chi}_c(N^-(v) \cup N^+(u)) \leq 5$. Moreover each remaining vertex is in $N(e)$ for some edge e on the shortest path. So in total, we use at most eight colors. \square

Now it remains to prove Lemma 4.1.2. We will start by establishing some structural claims about light tournaments which are adapted from [BCC⁺13]. For the rest of this section, $T = (V, A)$ will denote a light tournament. Note that we do not assume that T is necessarily 2-colorable. Recall that a C_3 is a directed triangle.

Definition 4.1.1. Define a C_3 -chain of length ℓ in T to be a set of ℓ vertex disjoint C_3 's, $X = (X_1, X_2, X_3, \dots, X_\ell)$, such that for each $i \in \{1, \dots, \ell - 1\}$, $X_i \Rightarrow X_{i+1}$.

A backwards arc in a C_3 -chain is an arc uv with $u \in X_i$ and $v \in X_j$ for $j < i$.

Lemma 4.1.3. A C_3 -chain has no backwards arcs.

This follows from the following claim.

Claim 4.1.4. If $X = (X_1, X_2, \dots, X_\ell)$ is a C_3 -chain of length ℓ , then $X_i \Rightarrow X_j$ for $i < j$, where $1 \leq i < j \leq \ell$.

Proof. Notice that there are no arcs from X_{i+1} to X_i , since by definition of a C_3 -chain, we have all arcs from X_i to X_{i+1} . Moreover, there is no arc uv from X_{i+2} to X_i since otherwise triangle X_{i+1} would appear in the neighborhood $N(uv)$, meaning that uv is heavy, which is a contradiction. This implies that all arcs go from X_i to X_{i+2} (since T is a tournament). Now suppose $j > i + 2$. If there is a back arc uv from $u \in X_j$ to $v \in X_i$, then uv is a heavy arc, because X_{j-1} would be in $N(uv)$ since by induction we have all arcs from X_i to X_{j-1} and from X_{j-1} to X_j . \diamond

Let us fix $X = (X_1, X_2, \dots, X_\ell)$ to be a C_3 -chain in T , and let $W = V(T) \setminus V(X)$. Initially, X can be of any length $\ell \geq 1$.

Claim 4.1.5. For $w \in W$:

1. If $w \Rightarrow X_i$, then $w \Rightarrow X_j$ for all $j \geq i$.
2. If $X_i \Rightarrow w$, then $X_j \Rightarrow w$ for all $j \leq i$.

Proof. Suppose $w \Rightarrow X_i$ and there is an arc uw with $u \in X_j$ for $j > i$. Then uw is a heavy arc. Similarly, suppose $X_i \Rightarrow w$ and there is an arc wu with $u \in X_j$ for $j < i$, then wu is a heavy arc. \diamond

We partition the vertices in W into zones $(Z_0, Z_1, \dots, Z_\ell)$ using the following criteria. For $w \in W$, if i is the highest index such that $X_i \Rightarrow w$, then w is assigned to zone Z_i . If there is no such X_i , then w is assigned to zone Z_0 .

Say a vertex $w \in W$ is *clear* if $w \Rightarrow X_i$ or $X_i \Rightarrow w$ for all X_i in X . Let $C \subseteq W$ be the set of clear vertices.

Claim 4.1.6. If C is not transitive, we can extend X .

Proof. If the set $Z_i \cap C$ contains a triangle, then we can extend X by adding a new triangle to the chain between X_i and X_{i+1} .

If there is no i such that $Z_i \cap C$ contains a triangle, then we claim that C is transitive. This follows from the observation that there are no backwards arcs from $Z_j \cap C$ to $Z_i \cap C$ for $i < j$. Indeed, should such an arc uv from $Z_j \cap C$ to $Z_i \cap C$ exist, then $X_{i+1} \subset N(uv)$, so uv would be heavy. \diamond

We say that X is a *maximal C_3 -chain* if C is transitive. Let us also now define the *unclear* vertices U , where $U = W \setminus C$. In a maximal C_3 -chain $X = (X_1, \dots, X_\ell)$, notice that for a vertex $a \in X_1$, we have $N^-(a) \cap U \subseteq N^\pm(X_1)$. (This is because if a vertex $u \in N^-(a)$ has $u \Rightarrow X_i$ then u would be a clear vertex.)

Claim 4.1.7. We can efficiently find two directed triangles $X_1 = abc$ and $X_\ell = xyz$ such that the set $S = \{v \mid v \Rightarrow X_1 \text{ or } X_\ell \Rightarrow v\}$ is transitive.

Proof. Find a maximal C_3 -chain X and let ℓ be the length of this chain. Let $abc = X_1$ and $xyz = X_\ell$. The set of vertices $\{v \mid v \Rightarrow X_1 \text{ or } X_\ell \Rightarrow v\}$ is a subset of C and is therefore transitive. \diamond

Claim 4.1.8. Let xyz be a directed triangle. Then $\vec{\chi}_c(N^\pm(\{x, y, z\})) \leq 3$.

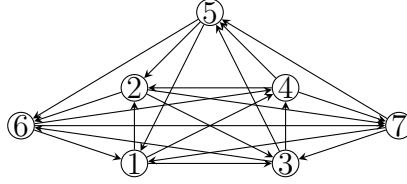


Figure 4.1: 3-chromatic light tournament.

Proof. Each vertex $v \in N^\pm(\{x, y, z\})$ belongs to $N(xy)$, $N(yz)$ or $N(zx)$. Since each of these sets is transitive, we conclude that $N^\pm(\{x, y, z\})$ can be colored with three colors. \diamond

We can now prove Lemma 4.1.2.

Proof of Lemma 4.1.2. Recall that for a vertex $a \in X_1$, we have $N^-(a) \cap U \subseteq N^\pm(X_1)$. If $X_1 = abc$, notice that for $v \in N^-(a) \cap U$, $v \notin N(ca)$. Thus, $N^-(a) \cap U \subseteq N(ab) \cup N(bc)$, which is efficiently 2-colorable. Making an analogous argument for $X_\ell = xyz$ and $N^+(z) \cap U$, we conclude that $(N^+(z) \cup N^-(a)) \cap U$ is efficiently 4-colorable. The rest of the vertices in $N^+(z) \cup N^-(a)$ belong to the set S defined in Claim 4.1.7 and can be colored with one color. Therefore $\vec{\chi}_c(N^+(z) \cup N^-(a)) \leq 5$. Moreover, we have $\vec{\chi}_c(N^+(z)) \leq 3$ and $\vec{\chi}_c(N^-(a)) \leq 3$. \square

The approach in this section can be extended to bound the chromatic number of a more general subclass of heroes. See Section 4.3 for details.

It is a natural question to determine upper and lower bounds on the chromatic number of light tournaments (e.g., see Problem 1 in [MW11]). Theorem 4.1.1 gives an upper bound on the chromatic number of a light tournament. On the other hand, there exist light tournaments that are not 2-colorable. An example is the Paley tournament P_7 , one of the four 3-chromatic tournaments on seven vertices [NL94]. This tournament is represented in Figure 4.1. We have not found any light tournament with chromatic number at least four. The Paley tournament P_{11} is the unique 4-chromatic tournament on 11 vertices [NL94]. A light 4-chromatic tournament would have to have at least 13 vertices as [BBKP23] proved that any 4-chromatic tournament on 12 vertices must contain an induced copy of P_{11} and P_{11} is not light.

Regarding the complexity of coloring a light tournament, notice that if we could show that it is hard to color a 2-colorable tournament with four colors (rather than three as per Theorem 3.2.2), this would imply hardness of coloring a 2-colorable light tournament with two colors by Observation 2.2.2. Indeed, we have no hardness

results for coloring light tournaments. Any upper bound of c on their chromatic number would imply that it cannot be NP-hard to color them with c colors, because the property of being light is checkable in polynomial time (unlike the property of being, say, 2-colorable).

4.2 Other decompositions for light tournaments

For the sake of completeness, we show that two other approaches from the literature can be adapted to obtain efficient algorithms for coloring light tournaments. We prove the following lemma, which is weaker than what we proved in Section 4.1.

Lemma 4.2.1. *Let T be a light tournament. Then $\vec{\chi}_c(T) \leq 35$.*

Algorithm I for Coloring Light Tournaments

Since a light tournament forbids a heavy edge, and a heavy edge is a hero (i.e., it is $\Delta(C_3, 1, 1)$), we show that the decomposition approach of [BCC⁺13] for bounding the dichromatic number of tournaments without a fixed hero can be turned into an efficient algorithm to color a light tournament with approximately 35 colors. Throughout this section $T = (V, A)$ will denote a light tournament.

In Section 4.1, we already presented many of the necessary definitions. We consider a C_3 -chain (Definition 4.1.1). Since there are no backwards arcs in a C_3 -chain, we have the following corollary.

Corollary 4.2.2. *A C_3 -chain can be efficiently 2-colored.*

Let us fix $X = (X_1, X_2, \dots, X_\ell)$ to be a maximal C_3 -chain in T , and let $W = V(T) \setminus V(X)$. Recall that W is further partitioned into *clear* and *unclear* vertex sets denoted by C and U , respectively. A vertex v belongs to C if for every X_i , we have either $v \Rightarrow X_i$ or $X_i \Rightarrow v$. If C is transitive, then X is defined to be maximal. Notice that such a maximal C_3 -chain can be found in polynomial time, while in the proof of [BCC⁺13], they used a maximum length C_3 -chain (or more generally *jewel-chain*); it is not clear that a C_3 -chain of maximum length can be found efficiently.

Now let us consider the unclear vertices U . Notice that if a vertex $u \in U$ belongs to zone Z_i for $i \in \{1, \dots, \ell - 1\}$, then there is at least one arc from u to a vertex in X_{i+1} .

Claim 4.2.3. $\vec{\chi}_c(Z_i \cap U) \leq 3$.

Proof. If $z \in Z_i \cap U$, then $X_i \Rightarrow z$. However, we have $z \leftrightarrow X_{i+1}$. This means that z belongs to $N(uv)$ for some arc $uv \in X_{i+1}$. In other words, we can partition the vertices in $Z_i \cap U$ into three sets according to the three arcs in X_{i+1} . Since there are no heavy arcs, each of these three sets is transitive and we can color $Z_i \cap U$ with three colors. \square

Claim 4.2.4. $\vec{\chi}_c(X_i \cup (Z_i \cap U)) \leq 5$.

Proof. We use two colors for X_i (which is a triangle) and three colors for $Z_i \cap U$. \square

For simplicity, let us now assume that every vertex in $V(T)$ belongs to X or to U . Thus, we assume that $Z_i = Z_i \cap U$. (We only need one extra color for C since it is transitive.) Let $Y_0 = Z_0$ and for $i \in \{1, \dots, \ell\}$, let $Y_i = X_i \cup Z_i$. Define $Y_i^L = \bigcup_{j=0}^{i-1} Y_j$ and $Y_i^R = \bigcup_{j=i+1}^{\ell} Y_j$.

Claim 4.2.5. *Let $v \in Y_i$. Then:*

(i) $\vec{\chi}_c(N^+(v) \cap Y_i^L) \leq 3$, and

(ii) $\vec{\chi}_c(N^-(v) \cap Y_i^R) \leq 3$.

Proof. We consider several cases. The first case is $v \in X_i$. Notice that $N^+(v) \cap X_j = \emptyset$ for $j < i$, and $N^-(v) \cap X_j = \emptyset$ for $j > i$. We also note that $N^-(v) \cap (Z_{i+1} \cup \dots \cup Z_\ell) = \emptyset$. Now observe $\vec{\chi}_c(N^+(v) \cap (Z_0 \cup Z_1 \cup \dots \cup Z_{i-1})) \leq 3$. This is because each $u \in N^+(v) \cap (Z_0 \cup Z_1 \cup \dots \cup Z_{i-1})$ belongs to $N(xy)$ for some arc $xy \in X_i$.

Now consider the case where $v \in Z_i$. There are four subcases to consider.

1. $N^+(v) \cap (X_1 \cup \dots \cup X_{i-1}) = \emptyset$.

2. $N^+(v) \cap (Z_0 \cup Z_1 \cup \dots \cup Z_{i-1})$

Consider $u \in N^+(v) \cap (Z_0 \cup Z_1 \cup \dots \cup Z_{i-1})$. If $u \Rightarrow X_i$, then vu is a heavy arc (since $X_i \subseteq N(vu)$). If $X_i \Rightarrow u$, then u would be in Z_i (at least). Thus, u belongs to $N(xy)$ for some arc $xy \in X_i$. Hence, $\vec{\chi}_c(N^+(v) \cap (Z_0 \cup Z_1 \cup \dots \cup Z_{i-1})) \leq 3$.

3. $N^-(v) \cap (X_{i+1} \cup \dots \cup X_\ell)$.

Since $\vec{\chi}_c(X) \leq 2$, we have $\vec{\chi}_c(N^-(v) \cap (X_{i+1} \cup \dots \cup X_\ell)) \leq 2$.

4. $N^-(v) \cap (Z_{i+1} \cup \dots \cup Z_\ell)$.

Consider $x \in X_{i+1}$ such that arc vx is an arc. (Such an x exists, because it is not the case that $X_{i+1} \Rightarrow v$.) Now consider $u \in N^-(v) \cap (Z_{i+1} \cup \dots \cup Z_\ell)$. We claim that $u \in N(vx)$. We conclude that $\vec{\chi}_c(N^-(v) \cap (Z_{i+1} \cup \dots \cup Z_\ell)) \leq 1$.

1. and 2. together imply (i) in the statement of the claim, and 3. and 4. imply (ii). □

Lemma 4.2.6. *Let $(Y_0, Y_1, \dots, Y_\ell)$ be a partition of $V(T)$ such that $\vec{\chi}_c(Y_i) \leq c_1$ and for each $v \in Y_i$:*

- $\vec{\chi}_c(N^+(v) \cap (Y_0 \cup Y_1 \cup \dots \cup Y_{i-1})) \leq c_2$, and
- $\vec{\chi}_c(N^-(v) \cap (Y_{i+1} \cup \dots \cup Y_\ell)) \leq c_2$.

Then $\vec{\chi}_c(T) \leq 2(2c_1 + 2c_2 + 1)$.

Proof. Let $B \subset A(T)$ be the set of backwards arcs ($uv \in B$ if $u \in Y_i$ and $v \in Y_j$ for $j < i$). If there are no backwards arcs, then $\vec{\chi}_c(T) \leq \max_{i \in \{0, 1, \dots, \ell\}} \{\vec{\chi}_c(Y_i)\} \leq c_1$. Now we consider only a subset of backwards arcs chosen as follows: Choose the longest backwards arc u_1v_1 where $u_1 \in Y_\ell$. Suppose $v_1 \in Y_j$ for $j < \ell$. Let $T_1 = \bigcup_{i=j}^\ell Y_i$. Then choose the next backwards arc u_2v_2 with $u_2 \in V(T_1)$ and v_2 in Y_k for the smallest value of k possible, etc. Let $T_2 = \bigcup_{i=k}^{j-1} Y_i$, etc. Notice that if we consider the union of all T_i with odd i , there are no backwards arcs between them, and the same for T_i with even i . Suppose there are h such T_i 's.

Then, $\vec{\chi}_c(T) \leq 2 \cdot \max_{i \in \{1, \dots, h\}} \vec{\chi}_c(T_i)$. Now we claim that $\vec{\chi}_c(T_i) \leq 2c_1 + 2c_2 + 1$. Consider the backwards arc uv , where $u \in Y_k$ and $v \in Y_j$ for $j < k$. We can color $N^-(v) \cap (T_i \setminus Y_j)$ and $N^+(u) \cap (T_i \setminus Y_k)$ each with c_2 colors. We can color Y_j and Y_k each with c_1 colors. Finally, we consider all vertices in $P = T_i \setminus \{Y_j \cup Y_k \cup N^+(u) \cup N^-(v)\}$. All vertices in P belong to $N(uv)$ and thus form a transitive tournament requiring one more color. □

So the algorithm to color a light tournament T is to find a maximal C_3 -chain X . Next, color the clear vertices C with one color and remove C from T . Now consider the induced tournament on the remaining vertices and construct the partition $(Y_0, Y_1, \dots, Y_\ell)$ based on X . Now follow the procedure in Lemma 4.2.6.

Notice that $c_1 = 5$ and $c_2 = 3$. So Lemma 4.2.6 uses 34 colors, and we add one more color to color C .

Algorithm II for Coloring Light Tournaments

[HLNT19] gave an algorithm to color a triangle-free dense digraph (with bounded independence number). We show how this approach can be adapted to give another algorithm to color a light tournament with at most 29 colors.

In this section, T is always a light tournament unless otherwise noted.

Definition 4.2.1. *A set of vertices $B \subseteq V$ is a bag of T if for every triangle xyz in $V \setminus B$, there is some vertex $b \in B$ such that $\{x, y, z\} \Rightarrow b$ or $b \Rightarrow \{x, y, z\}$. Moreover, a bag must contain a directed triangle (i.e., it cannot be transitive).*

Observe that if B is not a bag of T , then we can color B with three colors. If B is a bag of T , then any S such that $B \subset S \subset V$ is also a bag of T . Also, note that V itself is a bag of T .

Claim 4.2.7. *If $B \subset V$ is not a bag of T , then $\vec{\chi}_c(T[B]) \leq 3$.*

Proof. If B is not a bag of T because it does not contain a triangle, then it is transitive. If it contains a triangle and is not a bag of T , then there is some triangle xyz such that $\{x, y, z\} \subset V \setminus B$ and for every $b \in B$, $b \in N^o(\{x, y, z\})$. Thus, we can apply Claim 4.1.8. \square

We say a bag B is *poor* if there is some triangle $xyz \in B$ such that $N^+(\{x, y, z\})$ or $N^-(\{x, y, z\})$ is not a bag. We want to show that poor bags can also be colored with a constant number of colors.

Claim 4.2.8. *If $B \subseteq V$ is a poor bag, then $\vec{\chi}_c(T[B]) \leq 18$.*

Proof. Consider a poor bag B . Consider all triangles in B . For each triangle, either its in-neighborhood or its out-neighborhood is not a bag of T . If there is a triangle xyz in B such that both its in-neighborhood and its out-neighborhood are not bags of T , then we can color B with at most 11 colors: three for in-neighborhood, three for the out-neighborhood, two for the triangle and three for $N^o(\{x, y, z\})$.

So suppose for each triangle in B , its in-neighborhood is not a bag and its out-neighborhood is a bag, or vice-versa. According to these two possibilities, partition these triangles into L and R and consider the respective vertex sets (which can overlap). Consider R . These are triangles whose out-neighborhood is not a bag. Since there are no backwards arcs in a chain of C_3 's, there must be some triangle

xyz in R such that $N^-(\{x, y, z\}) \cap R$ does not contain a triangle. Thus, we can color its in-neighborhood with one color, its out-neighborhood with three colors, its non-neighborhood with three colors, and the triangle itself with two colors, for a total of nine colors. We repeat the argument for L , so the maximum number of colors required is 18. \square

Following [HLNT19], our plan is to find a chain of poor bags, put the remaining vertices into zones and show that there are no long backwards arcs between the zones.

We define $B = (B_1, B_2, \dots, B_\ell)$ to be a *bag chain* of length ℓ if each B_i is a bag of T and $B_i \Rightarrow B_{i+1}$ for all $i \in \{1, 2, \dots, \ell - 1\}$. Let $W = V(T) \setminus V(B)$. Assign $w \in W$ to zone Z_i if i is the highest index such that $B_i \Rightarrow w$.

Claim 4.2.9. *Let $B = (B_1, B_2, \dots, B_\ell)$ be a bag chain for a light tournament T . Let $(Z_0, Z_1, \dots, Z_\ell)$ be a partition of $V(T) \setminus V(B)$ zones. The following properties hold:*

1. $B_i \Rightarrow B_{i+r}$ for every $r \geq 1$,
2. $B_i \Rightarrow Z_{i+r}$ for every $r \geq 0$,
3. $Z_i \Rightarrow B_{i+r}$ for every $r \geq 3$,
4. $Z_i \Rightarrow Z_{i+r}$ for every $r \geq 2$.

Proof. The first property holds for $r = 1$ by definition of a chain of bags. Now let $r \geq 2$. Suppose there is a backwards arc uv with $u \in B_{i+r}$ and $v \in B_i$. Since B_{i+1} contains a triangle, the arc uv is heavy, which is a contradiction.

By the partitioning criteria of vertices in $V(T) \setminus V(B)$ into zones, we have $B_i \Rightarrow Z_i$. If there is some arc uv with $u \in Z_i$ and $v \in B_j$ for $j < i$, then arc uv is heavy. Thus, $B_j \Rightarrow Z_i$ for all $j < i$.

To prove the third property, suppose there is an arc uv with $u \in B_{i+3}$ and $v \in Z_i$. Then there is some arc vx for $x \in B_{i+1}$ (otherwise, v would be in Z_{i+1}). Then uvx is a triangle. Since B_{i+2} is a bag of T , there is some vertex $y \in B_{i+2}$ such that $y \Rightarrow \{u, v, x\}$ or $\{u, v, x\} \Rightarrow y$. But this is not possible since $x \Rightarrow B_{i+2}$ and $B_{i+2} \Rightarrow u$. Thus, there is no such arc uv and we have $Z_i \Rightarrow B_{i+3}$. Now replace 3 with r .

For the last property, suppose that there is an arc uv with $u \in Z_{i+2}$ and $v \in Z_i$. Consider some $x \in B_{i+1}$ such that uxv is a triangle. Now since B_{i+2} is a bag of T , there is some $y \in B_{i+2}$ such that $y \Rightarrow \{u, x, v\}$ or $\{u, x, v\} \Rightarrow y$, which is a contradiction since $B_{i+1} \Rightarrow B_{i+2}$ and $B_{i+2} \Rightarrow Z_{i+2}$. Now replace 2 with $r \geq 2$. \square

Now we need to show that we can color $B_i \cup Z_i$ efficiently with a constant number of colors. For this, we need the following observations.

Claim 4.2.10. *A zone Z_i does not contain a bag chain of length at least five.*

Proof. If so, we can extend the principal bag chain B . □

Claim 4.2.11. *A tournament without a bag chain of length five can be colored with c colors.*

Proof. Let $S \subset V$ be a set of vertices such that S does not contain a bag chain of length five for T . Either S itself is not a bag of T or S is a poor bag of T , in which case, we are done. Otherwise, we find a triangle in S and partition the remaining vertices according to the in- and out-neighborhoods of this triangle, and perhaps repeat this procedure to produce a bag chain of length at most four. Each vertex that is not in this bag chain is in the non-neighborhood of some triangle. There are at most three “pivot” triangles used. So in the end, the vertices of S are decomposed into a bag chain of at most four poor bags, a chain of C_3 's, and five zones, each of which can be colored with three colors. So the number of colors required is at most $18 + 2 + 3 \cdot 3 = 29$. □

4.3 H_k -free tournaments

Definition 4.3.1. *Let $(H_k)_{0 \leq k}$ be the family of tournaments defined recursively with H_0 being a single vertex, and $\forall k \geq 1, H_{k+1} = \Delta(H_k, 1, 1)$.*

Notice that $H_1 = C_3$, and in general H_k is a *hero*. Define f to be the function such that for any $k \geq 1, f(k) = \vec{\chi}_c(T)$ where T is an H_k -free tournament. (In particular, $f(1) = 1$ and $f(2) \leq 8$, as shown in Section 4.1.)

Lemma 4.3.1. *For an integer $k \geq 0$, the number of colors needed to color an H_k -free tournament is $f(k) \leq \prod_{i=1}^{k-1} (4i + 5)$.*

The proof of this Lemma will follow the proof of Theorem 4.1.1: we will build a chain of H_k 's in order to bound the in-neighborhood of some vertex v and the out-neighborhood of another vertex u , which we will then use as the endpoints of a vertex chain.

We start by extending the notion of heavy arcs to this setting.

Definition 4.3.2. We say an arc is k -heavy if its neighborhood contains an H_k .

Notice that T is H_{k+1} -free iff it does not contain a k -heavy arc.

For the rest of the section, let T be an H_{k+1} -free tournament. Our goal will be to partition T into H_k -free sets, which will then allow us to color this type of tournament by induction.

Definition 4.3.3. Define a H_k -chain of length ℓ in T to be a set of ℓ vertex disjoint H_k 's, $X = (X_1, X_2, X_3, \dots, X_\ell)$, such that for each $i \in \{1, \dots, \ell - 1\}$, $X_i \Rightarrow X_{i+1}$.

A backwards arc in a H_k -chain is an arc uv with $u \in X_i$ and $v \in X_j$ for $j < i$.

Lemma 4.3.2. A H_k -chain has no backwards arcs.

This follows from the following claim.

Claim 4.3.3. Let T be a H_{k+1} -free tournament. If $X = (X_1, X_2, \dots, X_\ell)$ is a H_k -chain of length ℓ , then $X_i \Rightarrow X_j$ for $i < j$, where $1 \leq i < j \leq \ell$.

Proof. Notice that there are no arcs from X_{i+1} to X_i , since by definition of a H_k -chain, we have all arcs from X_i to X_{i+1} . Moreover, there is no arc uv from X_{i+2} to X_i since otherwise X_{i+1} would appear in the neighborhood $N(uv)$, meaning that $\{u\} \cup \{v\} \cup X_{i+1}$ forms an H_{k+1} , which is a contradiction. This implies that all arcs go from X_i to X_{i+2} (since T is a tournament). Now suppose $j > i + 2$. If there is a back arc uv from $u \in X_j$ to $v \in X_i$, then uv is a k -heavy arc, because X_{j-1} would be in $N(uv)$ since by induction we have all arcs from X_i to X_{j-1} and from X_{j-1} to X_j . \square

Let us fix $X = (X_1, X_2, \dots, X_\ell)$ to be an H_k -chain in T , and let $W = V(T) \setminus V(X)$. Initially, X can be of any length $\ell \geq 1$.

Claim 4.3.4. For $w \in W$:

1. If $w \Rightarrow X_i$, then $w \Rightarrow X_j$ for all $j \geq i$.
2. If $X_i \Rightarrow w$, then $X_j \Rightarrow w$ for all $j \leq i$.

Proof. Suppose $w \Rightarrow X_i$ and there is an arc uw with $u \in X_j$ for $j > i$. Then uw is a k -heavy arc. Similarly, suppose $X_i \Rightarrow w$ and there is an arc wu with $u \in X_j$ for $j < i$, then wu is a k -heavy arc. \square

We partition the vertices in W into zones $(Z_0, Z_1, \dots, Z_\ell)$ using the following criteria. For $w \in W$, if i is the highest index such that $X_i \Rightarrow w$, then w is assigned to zone Z_i . If there is no such X_i , then w is assigned to zone Z_0 .

Say a vertex $w \in W$ is *clear* if $w \Rightarrow X_i$ or $X_i \Rightarrow w$ for all X_i in H . Let $C \subseteq W$ be the set of clear vertices.

Claim 4.3.5. *If C is not H_k -free, we can extend X .*

Proof. If the set $Z_i \cap C$ contains an H_k , then we can extend X by adding a new H_k to the chain between X_i and X_{i+1} .

If there is no i such that $Z_i \cap C$ contains an H_k , then we claim that C is H_k -free. This follows from the observation that there are no backwards arcs from $Z_j \cap C$ to $Z_i \cap C$ for $i < j$. Indeed, should such an arc uv from $Z_j \cap C$ to $Z_i \cap C$ exist, then $X_{i+1} \subset N(uv)$, so uv would be k -heavy. \square

We say that X is a *maximal H_k -chain* if C is H_k -free. Let us also now define the *unclear* vertices U , where $U = W \setminus C$. In a maximal H_k -chain $X = (X_1, \dots, X_\ell)$, notice that for a vertex $a \in X_1$, we have $N^-(a) \cap U \subseteq N^o(X_1)$.

Claim 4.3.6. *We can efficiently find two H_k 's X_1 and X_ℓ such that the set $S = \{v \mid v \Rightarrow X_1 \text{ or } X_\ell \Rightarrow v\}$ is H_k -free.*

Proof. Find a maximal H_k -chain X and let ℓ be the length of this chain. The set of vertices $\{v \mid v \Rightarrow X_1 \text{ or } X_\ell \Rightarrow v\}$ is a subset of C and is therefore H_k -free. \square

Claim 4.3.7. *Let Y be an H_k . Then $\vec{\chi}_c(N^o(Y)) \leq (2k + 1) \cdot f(k)$.*

Proof. Take a Hamilton cycle $(e_i)_{1 \leq i \leq 2k+1}$ of Y . Each vertex $v \in N^o(Y)$ belongs to $N(e_i)$, for some i . Since each of these sets is H_k -free, we conclude that $N^o(Y)$ can be colored with $(2k + 1) \cdot f(k)$ colors. \square

We can now easily prove that H_{k+1} -free tournaments have bounded chromatic number by finding a $(4k + 1) \cdot f(k)$ -vertex chain.

Theorem 4.3.8. *Let T be an H_{k+1} -free tournament. Then $\vec{\chi}_c(T) \leq (4k + 5) \cdot f(k)$.*

Proof. We just need to show that we can find a $(4k + 1) \cdot f(k)$ -vertex chain. Recall that for a vertex $a \in X_1$, we have $N^-(a) \cap U \subseteq N^o(X_1)$. If $a \in X_1$, and $(e_i)_{1 \leq i \leq 2k+1}$ is a Hamilton cycle of X_1 with $e_1 = ua$ and $e_2 = av$ for some vertices u and v , then

notice that for $w \in N^-(a) \cap U$, $w \notin N(e_1)$. Thus, $N^-(a) \cap U \subseteq \cup_{2 \leq i \leq 2k+1} N^o(e_i)$, which is efficiently colorable with $2k \cdot f(k)$ colors, since it can be decomposed into $2k$ sets which are H_k -free and thus efficiently colorable with $f(k)$ colors. Making an analogous argument for $N^+(z) \cap U$, we conclude that $(N^+(z) \cup N^-(a)) \cap U$ is efficiently $4k \cdot f(k)$ -colorable. The rest of the vertices in $N^+(z) \cup N^-(a)$ belong to the set S defined in Claim 4.3.6 and can be colored with $f(k)$ colors. Therefore $\vec{\chi}_c(N^+(z) \cup N^-(a)) \leq (4k + 1) \cdot f(k)$, so we can use z and a as the endpoints of a $(4k + 1) \cdot f(k)$ -vertex chain. Finally, it is clear that the neighborhood of an edge in an H_{k+1} -free tournament is H_k -free, and can thus be colored efficiently with $f(k)$ colors. Then we can apply Lemma 2.1.3 to prove that $\vec{\chi}_c(T) \leq (4k + 5) \cdot f(k)$. \square

As an immediate corollary, we can bound the function f .

Corollary 4.3.9. *For all integers k , the number of colors needed to color an H_k -free tournament $f(k) \leq \prod_{i=1}^{k-1} (4i + 5)$.*

4.4 Extensions to dense digraphs

In this section, we extend our tools from tournaments to digraphs with bounded independence number. In this section, a digraph will always be an *oriented* graph (with no directed 2-cycles) unless indicated otherwise. Most of our notation extends easily from tournaments to digraphs.

We refer to the set of vertices that have no arc from or towards v as its *non-neighborhood* $N^o(v) = V \setminus \{N^+(v) \cup N^-(v)\}$. We say there is a *non-edge* between u and v if there is no arc in A from u to v or from v to u . If we are referring to a non-edge with a fixed direction (i.e., a non-edge $e = uv$), then we refer to it as a *non-arc*. For a non-arc $e = uv$, we use $N(e)$ to denote the vertices in $N^+(v) \cap N^-(u)$.

A digraph $D = (V, A)$ is *k -colorable* if there is a partition of V into k vertex-disjoint sets, V_1, V_2, \dots, V_k , such that $T[V_i]$ is acyclic for all $i \in \{1, \dots, k\}$. In this case, we say that T is *k -colorable* and we use $\vec{\chi}(D)$ to denote the *dichromatic number* of D (which we often refer to as *chromatic number* when the context is clear). Computing the value $\vec{\chi}(D)$ is in general NP-hard [BFJ⁺04]. We therefore use $\vec{\chi}_c(D)$ to denote the number of colors by which D can be efficiently colored. Our goal is to find upper and lower bounds on $\vec{\chi}_c(D)$.

4.4.1 Decomposition for digraphs

The goal of this section is to extend our techniques from tournaments to digraphs with bounded independence number. We adapt the decomposition given for tournaments in Section 2.1 by taking a shortest path—not just of arcs—but of arcs and non-arcs (in other words, non-edges can be included in the shortest path to be traversed in either direction).

Definition 4.4.1. *We define a vertex chain $(v_i)_{0 \leq i \leq k}$ of a digraph D as follows: Let v_0 and v_k be a pair of vertices and let $(v_i)_{0 \leq i \leq k}$ be the vertices in the shortest path from v_0 to v_k , where the path may consist of both forward arcs and non-arcs (but no backward arcs). If in addition, $\vec{\chi}_c(N^+(v_0)) \leq b$ and $\vec{\chi}_c(N^-(v_k)) \leq b$, then we call it a b -vertex chain.*

Additionally, we define an *edge chain* $(e_i)_{1 \leq i \leq k}$ corresponding to a vertex chain, where e_i is the arc or non-arc from v_{i-1} to v_i . We build zones that can be efficiently colored, and such that arcs between zones at distance more than four (i.e., *long arcs*) go backwards.

Definition 4.4.2. *Given a vertex chain, a path decomposition of a digraph D is defined as:*

- $D_0 = N^+(v_0)$.
- For $1 \leq i \leq k$, $D_i = N(e_i) \setminus (\cup_{0 \leq j \leq i-1} D_j)$.
- $D_{k+1} = N^-(v_k) \setminus (\cup_{0 \leq j \leq k} D_j)$.
- For $0 \leq i \leq k$, $N_i = N^o(v_i)$.

First we prove that this is indeed a decomposition of D .

Claim 4.4.1. *Let $D = (V, A)$ be a digraph, and $(D_0, \dots, D_{k+1}, N_0, \dots, N_k)$ be a path decomposition of D . Then $V = \cup_{0 \leq i \leq k+1} D_i \cup_{0 \leq i \leq k} N_i$.*

Proof. We will prove this claim by contradiction. Suppose there is a vertex $w \in V$ that does not belong to any D_i or N_i . Since w does not belong to D_0, D_{k+1} (nor to N_0 or N_k), then $w \in N^-(v_0)$ and $w \in N^+(v_k)$. Take the smallest integer i such that $w \in N^+(v_i)$. There must be one since $w \in N^+(v_k)$. Notice that $i \geq 1$ since $w \in N^-(v_0)$. Since $w \notin N^o(v_{i-1})$, then $w \in N(e_i)$. Therefore, $w \in D_i$, which is a contradiction. \square

We now show that long arcs between N_i s or between D_i s go backwards. Since the N_i s and D_i s will be colored with different color palettes, we don't need to worry about arcs between an N_i and D_j .

Claim 4.4.2. *Let $0 \leq i, j \leq k + 1$ and let $j \geq i + 5$. For $u \in D_i$ and $w \in D_j$, we have $u \in N^+(w)$. Furthermore, if $u \in N_i$ and $w \in N_{j-1}$, we have $u \in N^+(w)$.*

Proof. We will prove this by contradiction. Suppose $j \geq i + 5$ and $u \in N^-(w)$. Then there is a path of three arcs from v_i to v_{j-1} , namely (v_i, u, w, v_{j-1}) . (By definition of the decomposition, $u \in D_i$ implies $u \in N^+(v_i)$ and $w \in D_j$ implies $w \in N^-(v_{j-1})$.) This is not possible since by the definition of the vertex chain as the shortest path, there can be no path between v_i and v_{j-1} with fewer than four arcs (since $(j - 1) - i \geq (i + 5 - 1) - i = 4$). Finally, if $u \in N_i$ and $w \in N_{j-1}$, (v_i, u, w, v_{j-1}) is also a path (of forward arcs and non-arcs) of length three from v_i to v_{j-1} , which as previously is a contradiction. \square

Lemma 4.4.3. *If D has a b -vertex chain that can be found in polynomial time and if $\vec{\chi}_c(N(e)) \leq c$ for each arc and non-arc e , and $\vec{\chi}_c(N^\circ(v)) \leq d$ for every vertex v , then $\vec{\chi}_c(D) \leq 5c + 4d + 2(b - c)$ if $b > c$ and $\vec{\chi}_c(D) \leq 5c + 4d$ if $b \leq c$. Furthermore, if the shortest path from v_0 to v_k in the vertex chain has length at least four, then $\vec{\chi}_c(D) \leq 4c + 4d + b$.*

Proof. Given a b -vertex chain, we construct a path decomposition as per Definition 4.4.1. We make five palettes of c colors each with labels from 0 to 4. We color each D_i using the color palette with label $i \bmod 5$. By the assumptions of the lemma, we know that each D_i can be efficiently colored with c colors. Moreover, for every $1 \leq i \leq k$, D_i is a subset of $N(e_i)$, which can be colored efficiently with c colors by the condition of the lemma.

Next, we make four different palettes of d colors each with labels from 5 to 8. We color each N_i using the color palette with label $5 + (i \bmod 4)$. The set of colors used is of size d for every N_i , thus we can efficiently color each set by the assumption of the lemma.

Our goal is now to prove that this is a proper coloring of the digraph D . We will do this by showing that all forward arcs between different D_i or N_i are bicolored. By Claim 4.4.2, there are no forwards arcs between D_i and D_j when $j \geq i + 5$, or N_i and N_j with $j \geq i + 4$. Furthermore, by the definition of the coloring, no vertex in D_i and D_j can share a color for $i + 1 \leq j \leq i + 4$, and the same goes for vertices in N_i and N_j with $i + 1 \leq j \leq i + 3$. Thus all forward arcs from D_i to D_j or N_i

to N_j will be bicolored. Since every D_i and N_i is properly colored, and all forward arcs between different D_i are bicolored, the unions of D_i 's and the union of N_i 's are both properly colored. Finally, since these two unions use disjoint color palettes, and because every vertex is in some D_i or N_i (by Claim 4.4.1), the whole digraph D is properly colored.

If the shortest path from v_0 to v_k has length at least four, then we can save $b - c$ colors since we have all arcs from $N^-(v_k)$ to $V^+(v_0)$. \square

4.4.2 Algorithm for 2-colorable digraphs

In this section, our goal is to prove Theorem 4.4.4.

Theorem 4.4.4. *Let D be a 2-colorable digraph with independence number α . Then $\vec{\chi}_c(D) \leq \frac{10}{3}(4^\alpha - 1)$.*

In order to prove this theorem, we will define $f(\alpha)$ to be a function such that $f(\alpha) \geq \vec{\chi}_c(D)$ for every 2-colorable digraph D with independence number α . Our goal is to find an upper bound on $f(\alpha)$.

We say an arc e in D is *heavy* if $N(e)$ contains a directed cycle. Moreover, we say a non-edge between u and v is *heavy* if $N^+(u) \cap N^-(v)$ or if $N^+(v) \cap N^-(u)$ contains a directed cycle. If a digraph contains no heavy arcs and no heavy non-edges, then it is *light*. In the case of 2-colorable tournaments, we can partition the vertex set into two digraphs such that no arc is heavy, but in the case of 2-colorable digraphs we still need to account for the non-arcs. To do this, we will add a step before this partition.

Claim 4.4.5. *A 2-colorable digraph D can be transformed into a 2-colorable digraph D' with no heavy non-arc by adding arcs.*

Proof. Let e be a heavy non-arc between u and v . Without loss of generality, suppose that there is a directed cycle in $N^+(u) \cap N^-(v)$. This cycle must be colored with two colors, thus u , v and any $z \in N^-(u) \cap N^+(v)$ cannot all be colored the same color, else there would be a monochromatic 4-cycle. Therefore, we can add the arc from u to v , since it will not lead to any monochromatic cycle in a 2-coloring of D . So we can construct D' by starting from D , and while there are still heavy non-arcs, we add an arc between two vertices in way we just described. \square

After applying Claim 4.4.5, we obtain a digraph in which every non-arc is light. (Of course, we have not increased the independence number, since we have not removed any arcs.) Now we can observe that the set of heavy arcs forms a bipartite graph and we can therefore partition the set of vertices in order to split all heavy arcs. This results in the following corollary.

Corollary 4.4.6. *Let D be a 2-colorable digraph. Then D can be partitioned into two light 2-colorable digraphs D_1 and D_2 such that $\vec{\chi}_c(D) \leq \vec{\chi}_c(D_1) + \vec{\chi}_c(D_2)$.*

The next claim is similar to the case of tournaments.

Claim 4.4.7. *Let D be a k -colorable digraph. Then there exist vertices u and w such that $N^+(u) \cup N^-(w)$ is $(k - 1)$ -colorable.*

Proof. Since $D = (V, A)$ is k -colorable, there exist k transitive sets X_1, \dots, X_k such that $V = \cup_{i=1}^k X_i$. Then take u to be the vertex in X_1 that has only incoming arcs (or non-arcs) from other vertices in X_1 (i.e., the sink vertex for X_1). Similarly, take w to be the vertex in X_1 that has only outgoing arcs (or non-arcs) to other vertices in X_1 (i.e., the source vertex for X_1). The out-neighborhood of u and the in-neighborhood of w are both subsets of $V \setminus X_1$, and thus so is their union, which is therefore $(k - 1)$ -colorable. \square

We can then find a $(k - 1)$ -vertex chain in any k -colorable digraph (by guessing all pairs of endpoints). With the following observation, we can prove the main theorem by applying Lemma 4.4.3.

Observation 4.4.8. *Let $D = (V, A)$ be a digraph with independence number α . Then for any vertex $v \in V$, $N^o(v)$ has independence number $\alpha - 1$.*

This gives us $f_l(\alpha) = 5 + 4 \cdot f_l(\alpha - 1)$. We now define $f_l(\alpha)$ to be a function such that $f_l(\alpha) \geq \vec{\chi}_c(D)$ for every 2-colorable light tournament. By Corollary 4.4.6 we have $f(\alpha) \leq 2 \cdot f_l(\alpha)$, and thus $f(\alpha) \leq 2(5 + 4 \cdot f_l(\alpha - 1))$.

A simple calculation then leads to the statement of Theorem 4.4.4.

4.4.3 C_3 -free digraph with $\alpha = 2$.

In this section, $D = (V, A)$ is a C_3 -free oriented graph with independence number $\alpha(D) = 2$. Let C_k denote a cyclicly oriented k -cycle. We will assume that D is

maximally triangle-free, which means that for any pair of vertices u, v in V with no arc between them, both arcs uv and vu would each create a C_3 in D . In other words, if there is no arc between u and v , then there is a C_4 in D containing u and v . Let $g(\alpha)$ denote the dichromatic number of a C_3 -free digraph with independence number at most α .

The goal of this section is to prove the following lemma.

Lemma 4.4.9. *Let D be a C_3 -free digraph with $\alpha(D) = 2$. Then $\chi(D) \leq 16$.*

In [HLNT19], a bound for around 25 colors was claimed, so our bound here is an improvement, although the best lower bound is 2 (for a directed 4- or 5-cycle).

Claim 4.4.10. *If e is a non-edge between u and v in D , then $\chi(N(uv)) \leq g(\alpha - 1)$.*

Proof. Let a be a vertex in $N(uv)$. Since there is no arc between u and v , it must be the case that adding arc vu causes a C_3 , say with w . So then $N(uv) \subseteq N^o(w)$. If this were not the case, then we would have either triangle wva or wau , which is a contradiction. \square

Claim 4.4.11. *If D is C_k -free for $k \leq 5$ and $\alpha(D) = 2$, then D is acyclic.*

Proof. Assume that D contains some directed cycle and let C be the shortest such cycle on k vertices. Then $k \geq 6$. However, since $\alpha(D) = 2$, then C must contain a chord, implying that D contains a shorter cycle, which is a contradiction. \square

Let $X = (X_1, X_2, \dots, X_\ell)$ be a chain of C_4 's and C_5 's of maximum length.

Claim 4.4.12. *Let us fix some X_i in X . Let $S \subseteq V \setminus X_i$ be a subset of vertices. For every $s \in S$, there is some $x \in X_i$ such that there is an arc between s and x .*

Proof. If there is no such arc, then s is independent of every vertex in X_i resulting in a stable set of size three. \square

Claim 4.4.13. *All arcs in X are forward.*

Proof. We want to show that all arcs between X_i and X_j for $i \neq j$ go from X_i to X_j . We prove this by induction. By definition all arcs exist between X_i and X_{i+1} and go forwards. Now consider $x \in X_{i+2}$ and $y \in X_i$. By Claim 4.4.12, for each vertex in X_i , there is an arc to some vertex in X_{i+2} . This arc must be forwards, otherwise

there is a triangle. Now we assume that all arcs of length k are forwards and we can prove it for all arcs of length $k + 1$. Suppose we have an arc from $x \in X_i$ to $y \in X_{i+k+1}$. By Claim 4.4.12, we have an arc from x to some vertex $z \in X_{i+k}$. Since we have all arcs from X_{i+k} to X_{i+k+1} , we would have a triangle if x is not going to y . \square

Claim 4.4.14. *D has two vertices $u, v \in V$ such that $\chi(N^+(u)) \leq 5$ and $\chi(N^-(v)) \leq 5$.*

Proof. Fix a vertex $v \in X_1$. We want to analyze $\chi(N^-(v))$. A vertex in $N^-(v)$ is either in $N^-(X_1)$ or $N^\pm(X_1)$. The set of vertices that have arcs to all the vertices in X_1 is acyclic by Claim 4.4.11. The remaining vertices are either in $N^\pm(X_1)$ or are in the non-neighborhood of at least one vertex in X_1 . Since each vertex in $N^\pm(X_1)$ is also in the non-neighborhood of at least one vertex in X_1 , we can conclude that the $\chi(N^-(X_1)) \leq 6$. However, we have $\chi(N^-(v)) \leq 5$ since we can ignore $N^o(v)$. The same argument can be made for the out-neighborhood of some vertex $u \in X_\ell$. \square

Combining Claim 4.4.14 and Lemma 4.4.3, we conclude that we can color D with 16 colors. In the case where the shortest path from v_0 to v_k has length more than three, we get 13 colors. If the shortest path from v_0 to v_k has length at most three, then we get 3 colors for the edge neighborhoods and 4 colors for the vertex neighborhoods. To color $N^-(v_0)$ and $N^+(v_k)$, let A denote the set of vertices with complete arcs to X_1 (i.e. a vertex in A has arcs to each vertex in X_1) and let B denote the set of vertices with complete arcs from X_ℓ . Then we can color $(N^-(v_0) \cup N^+(v_k)) \setminus (A \cup B)$ with 8 colors. Then we claim that we can color $A \cup B$ with a single color. By some repeated applications of Claim 4.4.12, we can show that any arc between A and B must be from A to B . Thus, we can use at most 16 colors.

Chapter 5

Arc local-to-global

5.1 Overview of our results

We would like to thank the anonymous referees for helpful comments on the work in the following section.

The chromatic number of a graph is the minimum integer k required to partition its vertex set into k independent sets. The chromatic number of a tournament (and more generally, a directed graph) is the minimum integer k required to partition its vertex set into k acyclic sets. Exploring the similarities and differences between the two notions is a well-studied area [EH89, APS01].

For example, if a graph has a large clique, it must have high chromatic number. However, the converse is far from true. In fact, a graph can be triangle-free, implying that the neighborhood of each vertex is an independent set, and yet still have high chromatic number [Des54]. In [BCC⁺13], it was conjectured that this phenomenon does not occur in tournaments. Specifically, [BCC⁺13] conjectured that in a tournament T , if each vertex $v \in V(T)$ has an out-neighborhood $N^+(v)$ that induces a subtournament $T[N^+(v)]$ with bounded chromatic number, then T itself should have bounded chromatic number. This was proved by [HLTW19] with the following theorem.

Theorem 5.1.1 ([HLTW19]). *There is a function f such that if for all $v \in V(T)$, $\vec{\chi}(T[N^+(v)]) \leq t$, then $\vec{\chi}(T) \leq f(t)$.*

We say a tournament T has vertex set $V(T)$ and arc set $A(T)$. For an arc $e =$

$uv \in A(T)$, we define the *neighborhood of arc e* to contain all vertices w in $V(T)$ such that w forms a directed triangle with uv . Formally, we define $N(e) = N^+(v) \cap N^-(u)$. A stronger theorem, analogous to Theorem 5.1.1, but with vertex out-neighborhoods replaced by arc neighborhoods, is the following.

Theorem 5.1.2. *There is a function f such that for any tournament T , if for all $e \in A(T)$, $\vec{\chi}(T[N(e)]) \leq t$, then $\vec{\chi}(T) \leq f(t)$.*

This theorem is a special case of 13.3 in [NSS23b]. We give a different proof, obtained independently, which we subsequently extend to prove our main theorem. Notice that the assumption that $\vec{\chi}(T[N^+(v)]) \leq t$ for every vertex $v \in V(T)$ is stronger than the assumption that $\vec{\chi}(T[N(e)]) \leq t$ for every arc $e \in A(T)$. However, our proof of Theorem 5.1.2 uses a theorem from [HLTW19], which they used to prove Theorem 5.1.1. Thus, we do not give a new proof of Theorem 5.1.1. We say a tournament T is *t -arc-bounded* if for every arc $e \in A(T)$, $\vec{\chi}(T[N(e)]) \leq t$. We can now restate Theorem 5.1.2 as follows.

Theorem 5.1.3. *There is a function f such that for every t -arc-bounded tournament T , we have $\vec{\chi}(T) \leq f(t)$.*

We prove Theorem 5.1.3 in Section 5.2, where in addition to the aforementioned theorem of [HLTW19], we use ideas from Chapter 2. In Section 5.3, we extend our proof of Theorem 5.1.3 to oriented graphs with bounded independence number and prove our main theorem. For the sake of simplicity, we often refer to oriented graphs as digraphs, but in this Chapter, a digraph never contains a directed 2-cycle or “digon”. Recall that the *independence number* of a digraph is the maximum size of a vertex set that contains no arcs. We say a digraph D has vertex set $V(D)$ and arc set $A(D)$. As we did for tournaments, for an arc $e = uv$, we define $N(e) = N^+(v) \cap N^-(u)$.

Theorem 5.1.4. *There is a function h such that for any digraph D with independence number α , if $\vec{\chi}(D[N(e)]) \leq t$ for every arc $e \in A(D)$, then $\vec{\chi}(D) \leq h(t, \alpha)$.*

As an application of Theorem 5.1.4, we prove the equivalence of two conjectures, one on graphs with high chromatic number and one on tournaments with high chromatic number. The first one, concerning graphs, was originally posed by [EE85] in the form of an open problem, which asks if the following conjecture is true.

Conjecture 5.1.1 ([EE85]). *For all integers $t, c \geq 1$, there exists $d \geq 1$, such that if a graph G satisfies $\chi(G) \geq d$, and has no clique with t vertices (i.e., $\omega(G) < t$),*

then there are subsets $A, B \subseteq V(G)$ with $\chi(G[A]), \chi(G[B]) \geq c$, such that there are no edges between A and B .

The second conjecture, concerning tournaments, was recently stated by [NSS23a].

Conjecture 5.1.2 ([NSS23a]). *For all $c \geq 0$, there exists $d \geq 0$ such that if T is a tournament with $\bar{\chi}(T) \geq d$, there are two sets $A, B \subseteq V(T)$ such that $\bar{\chi}(T[A]), \bar{\chi}(T[B]) \geq c$ and all arcs between A and B go from vertices of A to vertices of B .*

[NSS23b] show that Conjecture 5.1.2 implies Conjecture 5.1.1. They explore the possibility of the converse being true, but they do not prove it and write that Conjecture 5.1.2 seems to be strictly stronger than Conjecture 5.1.1. In Section 5.4, we prove that Conjecture 5.1.1 does in fact imply Conjecture 5.1.2, showing that the two conjectures are equivalent.

5.2 Arc local-to-global for tournaments

In this section, we prove Theorem 5.1.3. Since our goal is to color a tournament T , we can assume that T is strongly connected; otherwise T can be partitioned into strongly connected parts, and each one can be colored separately. A *dominating set* (respectively, *absorbing set*) in T is a set of vertices $S \subset V$ such that for every $v \in V \setminus S$, there is a $u \in S$ such that uv (respectively, vu) is an arc in T . By *domination number*, we mean the minimum size of a dominating set. We will use the following theorem from [HLTW19].

Theorem 5.2.1 ([HLTW19]). *For every constant k , there exist constants K and ℓ such that every tournament T with domination number at least K has a subset of size ℓ that induces a tournament with chromatic number at least k .*

Following the notation in [AAC22], we define a (k, ℓ) -*cluster* to be a set of vertices S such that $\bar{\chi}(T[S]) \geq k$, $|S| \leq \ell$ and $T[S]$ is strongly connected. This notion is directly related to the previous theorem, which can be restated for our purposes as follows.

Corollary 5.2.2. *There exist functions K and ℓ such that for every integer $t \geq 1$, every tournament T contains either i) a dominating set and an absorbing set, each of size at most $K(t)$, or ii) a $(t, \ell(t))$ -cluster.*

Proof. Let t be a constant. By Theorem 5.2.1, there exist constants $K(t)$ and $\ell(t)$ such that one can find either a dominating set of size at most $K(t)$, or a subset of size $\ell(t)$ with chromatic number t . If this subset is not strongly connected, we can find a strongly connected subset with chromatic number t . Then take the tournament obtained by reversing all the arcs in T and repeat the previous argument. A dominating set in this tournament is an absorbing set in T , while a subset with high chromatic number would also have high chromatic number in T , as reversing all the arcs preserves the chromatic number. \square

Now let us fix a constant t . Using the function ℓ from Corollary 5.2.2, we define a *jewel* to be a $(t + 1, \ell(t + 1))$ -cluster. We will use the fact that in a t -arc-bounded tournament, for any arc e , the vertex set $N(e)$ does not contain a jewel.¹ We now present some useful tools for coloring t -arc-bounded tournaments.

5.2.1 Jewels and other tools for coloring t -arc-bounded tournaments

We begin with a decomposition lemma for tournaments.

Lemma 5.2.3. *Let T be a tournament, and let $P = (v_0, v_1, \dots, v_k)$ be a shortest path in T from v_0 to v_k with arcs $e_i = v_{i-1}v_i$ for $i : 1 \leq i \leq k$. Then we have the following properties.*

1. *Each vertex in $N^-(v_0) \cap N^+(v_k)$ belongs to $N(e_i)$ for some $i : 1 \leq i \leq k$.*
2. *If $k \geq 3$, then each vertex in $V(P)$ belongs to $N(e_i)$ for some $i : 1 \leq i \leq k$.*
3. *If $k = 2$, then v_0 belongs to $N(e_2)$ and v_2 belongs to $N(e_1)$.*

Proof. First consider a vertex v in $N^-(v_0) \cap N^+(v_k)$ and let i be the maximum index such that $v \in N^-(v_i)$. Some v_i must exist, since $v \in N^-(v_0)$. Then $v \in N(e_{i+1})$. Next, consider a vertex $v \in V(P)$. Notice that all arcs between vertices $V(P)$ that are not adjacent in P must go backward. It follows that $v_i \in N(e_{i+2})$ and $v_i \in N(e_{i-1})$. When $k \geq 3$, we can conclude that each v_i belongs to $N(e_j)$ for some j such that $1 \leq j \leq k$. When $k = 2$, the same argument applies, except now v_1 belongs neither to $N(e_1)$ nor to $N(e_2)$. \square

¹We will redefine a jewel in Section 5.3, but it will have the same purpose.

Lemma 5.2.4. *Let T be a t -arc-bounded tournament. Suppose that $P = (v_0, v_1, \dots, v_k)$ is a shortest path from v_0 to v_k , and let $S = (N^-(v_0) \cap N^+(v_k)) \cup V(P)$. Then $T[S]$ can be colored with at most $5t$ colors.*

Proof. If $k \geq 3$, then each vertex in S belongs to $N(e_i)$ for some $i : 1 \leq i \leq k$. There are no arcs from $N(e_i)$ to $N(e_{i+5})$, since this would give a shorter path from v_0 to v_k . Thus, we can color all the vertices in S using five color palettes of t colors each, using one color palette for each $N(e_i)$ assigned modulo 5. Since all forward arcs have length at most four, each cycle with vertices belonging to different $N(e_i)$'s has at least two different colors. Finally, if $k = 2$, then $T[S]$ can be colored with $2t + 1$ colors, and if $k = 1$, then $T[S]$ can be colored with $t + 2$ colors. \square

Lemma 5.2.4 can be used to prove the following two lemmas, which we will apply shortly to prove Theorem 5.1.3.

Lemma 5.2.5. *Let T be a t -arc-bounded tournament, containing two vertices u and v such that $\bar{\chi}(T[N^+(u)]) \leq g(t)$ and $\bar{\chi}(T[N^-(v)]) \leq g(t)$ for some function g . Then $\bar{\chi}(T) \leq 2 \cdot g(t) + 5t$.*

Proof. Since T is strongly connected, we can set $v_0 = u$ and $v_k = v$, find a shortest path from v_0 to v_k , and apply Lemma 5.2.4 to color the subtournament $T[S]$. Any remaining vertex belongs to at least one of the sets $N^+(v_0)$ and $N^-(v_k)$, which can each be colored with $g(t)$ colors. \square

If T has small dominating and absorbing sets, then the following lemma provides a bound on $\bar{\chi}(T)$.

Lemma 5.2.6. *Let T be a t -arc-bounded tournament. Suppose T has a dominating set $\gamma^+(T)$ and an absorbing set $\gamma^-(T)$. Then $\bar{\chi}(T) \leq 5t \cdot |\gamma^-(T)| \cdot |\gamma^+(T)|$.*

Proof. We may assume that T is strongly connected. Let $q = |\gamma^-(T)| \cdot |\gamma^+(T)|$. Let $\mathcal{P} = \{P_1, P_2, \dots, P_q\}$ be a set of $|\gamma^-(T)| \cdot |\gamma^+(T)|$ shortest paths from each $u \in \gamma^-(T)$ to each $w \in \gamma^+(T)$. Then for each $v \in V$, there is some path $P_j \in \mathcal{P}$ from some u to some w such that $v \in (N^-(u) \cap N^+(w)) \cup V(P_j)$. So we can apply Lemma 5.2.4, which implies the lemma. \square

To prove Theorem 5.1.3, we need one more lemma. First, we give some notation and a definition. For two disjoint vertex sets, $X, Y \subset V$, we say $X \Rightarrow Y$ if all arcs between X and Y go from X to Y . For a set $S \subset V$, we define the set $N^\pm(S)$ to be

all vertices v in $V \setminus S$ such that there exist vertices $u, w \in S$ and arcs uv and vw in T .

Definition 5.2.1. We define a jewel-chain of length p in a tournament to be an ordered set $X = (X_i)_{1 \leq i \leq p}$ such that each X_i induces a jewel, all X_i 's are disjoint, and $X_i \Rightarrow X_{i+1}$ for all i such that $1 \leq i \leq p - 1$.

For a jewel-chain X , we say that an arc uv is *forward* if $u \in X_i$ and $v \in X_j$ and $i < j$. If $j < i$, then uv is *backward*. Jewel-chains are useful because they contain no backward arcs.

Observation 5.2.7. A jewel-chain X contains no backward arcs.

Proof. Consider a backward arc $e = uv$, with $u \in X_j$ and $v \in X_i$ for $j > i$. It must be the case that $j > i + 1$, since all arcs between X_i and X_{i+1} are forward by definition. Then $X_{i+1} \subseteq N(e)$, and since X_{i+1} has chromatic number at least $t + 1$, we have $\vec{\chi}(T[N(e)]) \geq t + 1$, which contradicts T being t -arc-bounded. Thus, all arcs with endpoints in distinct X_i 's must be forward. \square

Lemma 5.2.8. Let T be a t -arc-bounded tournament that contains a jewel. Then there exists a function g such that there are two vertices u, v such that $\vec{\chi}(T[N^+(u)]) \leq g(t)$, and $\vec{\chi}(T[N^-(v)]) \leq g(t)$.

Proof. Let X be a jewel-chain in T of maximum length, say p . Consider X_1 . Let Y be the set of vertices such that $Y \Rightarrow X_1$. Then Y does not contain a jewel (otherwise X would not have maximum length). By Corollary 5.2.2, Y must have a small dominating set and a small absorbing set, each of size at most $K(t + 1)$. So we can apply Lemma 5.2.6 to bound the chromatic number of Y by $5t \cdot (K(t + 1))^2$. Moreover, the set $N^\pm(X_1)$ has chromatic number at most $\ell(t + 1) \cdot t$, since X_1 contains a Hamilton cycle with at most $\ell(t + 1)$ arcs and each vertex in $N^\pm(X_1)$ belongs to $N(e)$ for some e in the Hamilton cycle. Finally, a vertex v in X_1 can have in-neighbors in X_1 itself, but this set has chromatic number at most $|X_1| \leq \ell(t + 1)$.

Set $g(t) = 2\ell(t + 1) \cdot t + 5t \cdot (K(t + 1))^2$. Then each vertex $u \in X_1$ has $\vec{\chi}(T[N^-(u)]) \leq g(t)$. By the same argument, each vertex $v \in X_p$ has $\vec{\chi}(T[N^+(v)]) \leq g(t)$. This proves Lemma 5.2.8. \square

5.2.2 Proof of Theorem 5.1.3

We are now ready to prove Theorem 5.1.3.

Theorem 5.1.3. *There is a function f such that for every t -arc-bounded tournament T , we have $\bar{\chi}(T) \leq f(t)$.*

Proof. Let $f(t) = 2 \cdot g(t) + 5t$, where g is defined as in the end of the Proof of Lemma 5.2.8. If T does not contain a jewel, then by Corollary 5.2.2, it contains a dominating and an absorbing set each of size at most $K(t+1)$. In this case, we can apply Lemma 5.2.6 to show that T can be colored with at most $5t \cdot (K(t+1))^2$ colors. If T contains a jewel, then we can apply Lemma 5.2.8 and Lemma 5.2.5 to prove the theorem. \square

5.3 Arc local-to-global for dense digraphs

In this section, we extend Theorem 5.1.3 from tournaments to oriented graphs with bounded independence number. Since our goal is to color a digraph D , we can assume that D is strongly connected; otherwise D can be partitioned into strongly connected parts, and each one can be colored separately. An important tool for the proof is Theorem 5.3.1, which extends Theorem 5.2.1 from tournaments to digraphs with bounded independence number.² A (k, ℓ) -cluster in a digraph D is a set of vertices S in $V(D)$ such that $\bar{\chi}(D[S]) \geq k$, $|S| \leq \ell$ and $D[S]$ is strongly connected.

Let D be a digraph with independence number α , and let $X, Y \subseteq V(D)$. Then the following inequalities are straightforward.

$$\begin{aligned} \gamma(D[N^+[X]]) &\leq |X|, \\ \gamma(D[Y]) &\leq \gamma(D[X]) + \gamma(D[Y \setminus X]). \end{aligned} \tag{5.3.1}$$

We now provide a proof of Theorem 5.3.1 that is very similar to the proof of Theorem 5.2.1 from [HLTW19].

Theorem 5.3.1. *There exist functions K and ℓ such that for every pair of integers $k, \alpha \geq 1$, every digraph D with independence number α and dominating number at least $K(\alpha, k)$ contains a $(k, \ell(\alpha, k))$ -cluster.*

Proof. Let $P(\alpha, k)$ denote the statement of the theorem for α and k . Our goal is to prove $P(\alpha, k)$ for all integers $\alpha, k \geq 1$. Let us assume that $P(\alpha - 1, k)$ holds for all $k \geq 1$. The base case for this is $P(1, k)$, which is proved in [HLTW19]. Now we

²Theorem 5.1.1 was extended to digraphs with bounded independence number by [HLNT19], but they did not provide an extension of Theorem 5.2.1.

fix α and we want to prove $P(\alpha, k)$, which we will do by induction on k . The base case for this is $P(\alpha, 1)$, which is true since any digraph with independence number α and domination number at least 1 contains at least one vertex, which serves as a $(1, 1)$ -cluster. To build intuition, we can also consider the next case, which is $P(\alpha, 2)$. This is true since any digraph with independence number α and domination number at least $\alpha + 1$ contains a directed cycle of length at most $\ell(\alpha, 2) \leq 2\alpha + 1$, and this cycle requires two colors.³ Now we assume $P(\alpha, k - 1)$ (as well as $P(\alpha - 1, k)$) and we want to prove $P(\alpha, k)$.

We will follow the proof of Theorem 5 from [HLTW19]. Let us first prove a useful claim. Recall that $N^o(v)$ is the set of vertices that form non-edges with v .

Claim 5.3.2. *If D does not contain a $(k, \ell(\alpha - 1, k))$ -cluster, then for any vertex $v \in V(D)$,*

$$\gamma(D[N^o(v)]) \leq K(\alpha - 1, k).$$

Proof. The digraph $D' = D[N^o(v)]$ has independence number $\alpha - 1$. By the inductive hypothesis on α , either D' has a $(k, \ell(\alpha - 1, k))$ -cluster or D' has domination number at most $K(\alpha - 1, k)$. Thus, $\gamma(D[N^o(v)]) \leq K(\alpha - 1, k)$. \diamond

Let $D = (V, E)$ be a digraph with independence number α such that $\gamma(D) \geq K(\alpha, k)$, and let B be a minimum dominating set of D . We will assume that D does not contain a $(k, \ell(\alpha - 1, k))$ -cluster, since otherwise, we would be done. Fix

$$K(\alpha, k) = k(K(\alpha - 1, k) + 1)(K(\alpha, k - 1) + \ell(\alpha, k - 1) \cdot (K(\alpha - 1, k) + 1) + 1) + K(\alpha, k - 1).$$

Consider a subset W of B , where

$$|W| = k(K(\alpha, k - 1) + \ell(\alpha, k - 1) \cdot (K(\alpha - 1, k) + 1) + 1).$$

From (5.3.1) and Claim 5.3.2, we have

$$\begin{aligned} \gamma(D[V \setminus (N^+[W] \cup N^o(W))]) &\geq \gamma(D) - \gamma(D[N^+[W]]) - \gamma(D[N^o(W)]) \\ &\geq \gamma(D) - |W| - |W|(K(\alpha - 1, k)) \\ &\geq K(\alpha, k) - |W|(K(\alpha - 1, k) + 1) \\ &\geq K(\alpha, k - 1). \end{aligned}$$

³This follows from the well-known classical theorem that an acyclic digraph has an independent dominating set. See [Bon03].

By applying the induction hypothesis, the digraph $D[V \setminus (N^+[W] \cup N^o(W))]$ contains a $(k-1, \ell(\alpha, k-1))$ -cluster. Call this vertex set A . Note that by construction, $A \cap W = \emptyset$ and A is complete towards W . Now consider a subset S of W where

$$|S| = K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1.$$

We claim that

$$\gamma(D[N^+(S)]) \geq K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1). \quad (5.3.2)$$

If not, we can choose a dominating set S' of $N^+(S)$, where

$$|S'| \leq K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) - 1.$$

Note that x dominates S for any $x \in A$, and so $S' \cup \{x\}$ dominates $N^+[S]$. Hence $(B \setminus S) \cup S' \cup \{x\}$ would be a dominating set of D of size less than $|B|$ which contradicts the minimality of B . We therefore conclude that Inequality (5.3.2) holds.

Let $N' = N^+(S) \setminus (N^+(A) \cup N^o(A))$. From Claims 5.3.1 and 5.3.2 we have

$$\begin{aligned} \gamma(D[N']) &\geq \gamma(D[N^+(S)]) - \gamma(D[N^+(A)]) - \gamma(D[N^o(A)]) \\ &\geq K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) - |A|(K(\alpha-1, k) + 1) \\ &= K(\alpha, k-1). \end{aligned}$$

Thus, by the induction hypothesis on k , there is a subset $A_S \subseteq N'$ that forms a $(k-1, \ell(\alpha, k-1))$ -cluster. By construction, $A_S \cap A = \emptyset$ and A_S is complete towards A .

We now construct our subdigraph of D with chromatic number at least k . We consider the set of vertices $A \cup W$ to which we add the collection A_S , for all subsets $S \subseteq W$ of size $K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1$. Call A' this new vertex set and observe that its size is at most

$$|A'| \leq |A| + |W| + |A_S| \binom{|W|}{|S|}.$$

So we have

$$\begin{aligned} \ell(\alpha, k) &= \ell(\alpha, k-1) + k(K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1) \\ &\quad + \ell(\alpha, k-1) \binom{k(K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1)}{K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1}. \end{aligned}$$

To conclude, it is sufficient to show that $\chi(A') \geq k$. Suppose not, and for contradiction, take a $(k-1)$ -coloring of A' . Since $|W| = k(K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1)$ there is a monochromatic set S in W of size $K(\alpha, k-1) + \ell(\alpha, k-1) \cdot (K(\alpha-1, k) + 1) + 1$ (say, colored 1). Recall that A_S is complete to A , and A is complete to S , and note that since $\chi(A) \geq k-1$ and $\chi(A_S) \geq k-1$, both A and A_S have a vertex of each of the $k-1$ colors. Hence there are $u \in A$ and $w \in A_S$ colored 1. Since $A_S \subseteq N^+(S)$, there is $v \in S$ such that (v, w) is an arc of D . We then obtain the monochromatic triangle (u, v, w) of color 1, a contradiction. Thus, $\vec{\chi}(D[A']) \geq k$ implying that A' is a $(k, \ell(\alpha, k))$ -cluster in D completing the induction on k .

Since this induction proves the statement $P(\alpha, k)$ holds for any k , it proves the inductive hypothesis for α . Then, by induction on α we have proven that the theorem is true for any pair of integers α, k . \square

Corollary 5.3.3. *There exist functions K and ℓ such that for every pair of integers $k, \alpha \geq 1$, every digraph D with independence number α contains either i) a dominating and an absorbing set, each of size at most $K(\alpha, k)$, or ii) a $(k, \ell(\alpha, k))$ -cluster.*

Proof. Let k and α be constants, and D a digraph with independence number α . By Theorem 5.3.1, there exist constants $K(\alpha, k)$ and $\ell(\alpha, k)$ such that one can find either a dominating set of size at most $K(\alpha, k)$, or a subset of size $\ell(\alpha, k)$ with chromatic number at least k . Take the digraph obtained by reversing all the arcs in D and repeat the previous argument. A dominating set in this digraph is an absorbing set in D , while a subset with high chromatic number would also have high chromatic number in D , as reversing all the arcs preserves the chromatic number. \square

In a digraph $D = (V, A)$, there are some pairs of vertices that do not have arcs between them. A pair u, v is a *non-edge* in D if neither arc uv nor arc vu belongs to A . The proof of the next theorem involves adding arcs to a digraph D to obtain a tournament. Since there are two sets of arcs, A and B , we use, for example, $N_A^+(u)$ (rather than the more standard $N_D^+(u)$) and $N_B^+(u)$ to denote the set of vertices adjacent from u via arcs in A or arcs in B , respectively. We define $N_A^o(u)$ to be all vertices in V that form non-edges with u in D . The goal in this section is to prove our main theorem.

Theorem 5.1.4. *There is a function h such that for any digraph D with independence number α , if $\vec{\chi}(D[N(e)]) \leq t$ for every arc $e \in A(D)$, then $\vec{\chi}(D) \leq h(t, \alpha)$.*

Proof. We prove this theorem by induction on α . For the base case, Theorem 5.1.2 proves the statement for $\alpha = 1$, by setting $h(t, 1) = f(t)$. For the induction hy-

pothesis, we assume that for any digraph $D = (V, A)$ with independence number $\alpha - 1$, if for all $e \in A$, $\vec{\chi}(D[N(e)]) \leq t$, then $\vec{\chi}(D) \leq h(t, \alpha - 1)$. Now our goal is to prove that for any digraph $D = (V, A)$ with independence number α , if for all $e \in A$, $\vec{\chi}(D[N(e)]) \leq t$, then $\vec{\chi}(D) \leq h(t, \alpha)$.

Consider a digraph $D = (V, A)$ with independence number α . We construct a tournament $T = (V, A \cup B)$ where each arc in B is a non-edge in D . Recall that we use $N_A^+(u)$ to denote the set of vertices adjacent from u via arcs from A . Now we assign directions as follows. For each non-edge u, v in D , if $N_A^+(v) \cap N_A^-(u)$ and $N_A^+(u) \cap N_A^-(v)$ are both empty (i.e., contain no vertices) or are both non-empty, we direct the arc arbitrarily. Otherwise either $N_A^+(u) \cap N_A^-(v) = \emptyset$, and we direct the arc from v to u , or $N_A^+(v) \cap N_A^-(u) = \emptyset$ and we direct the arc from u to v . Thus, we have the following property for each arc uv in B : Either $N_A^+(v) \cap N_A^-(u)$ contains no vertices or $N_A^+(v) \cap N_A^-(u)$ and $N_A^+(u) \cap N_A^-(v)$ both contain at least one vertex.

Now our goal is to color the tournament T such that each color class induces an acyclic set of arcs from A . This will in turn bound the chromatic number of D . We use the notation $D[N_T(e)]$ to denote the subgraph of D (i.e., arcs from A) in the neighborhood of arc e in T .

Claim 5.3.4. $\forall e \in A \cup B, \vec{\chi}(D[N_T(e)]) \leq 3 \cdot h(t, \alpha - 1) + 2t$.

Proof. Consider an arc $e = uv \in A$. We partition $N_T(e)$ into three subsets of vertices.

- (i) $S_1 = N_A^-(u) \cap N_A^+(v)$. By the condition of the theorem, $\vec{\chi}(D[S_1]) = \vec{\chi}(D[N_A(e)]) \leq t$.
- (ii) $S_2 = N_B^-(u)$. Then $D[S_2]$ has independence number at most $\alpha - 1$. Thus, by the induction hypothesis, $\vec{\chi}(D[S_2]) \leq h(t, \alpha - 1)$.
- (iii) $S_3 = N_B^+(v)$. Then $D[S_3]$ has independence number at most $\alpha - 1$. Thus, by the induction hypothesis $\vec{\chi}(D[S_3]) \leq h(t, \alpha - 1)$.

Therefore, for an arc $e \in A$, we have $\vec{\chi}(D[N_T(e)]) \leq 2 \cdot h(t, \alpha - 1) + t$. Next, we consider an arc $e = uv \in B$. We partition $N_T(e)$ into three subsets of vertices.

- (i) $S_1 = N_A^-(u) \cap N_A^+(v)$. Then either S_1 is empty, in which case $\vec{\chi}(D[S_1]) = 0$, or S_1 is non-empty. In this case, take any vertex $w \in N_A^+(u) \cap N_A^-(v)$. Notice that $S_1 \subseteq N_A(uw) \cup N_A(wv) \cup N_A^o(w)$. By the condition of the theorem, $\vec{\chi}(D[N_A(uw)]) \leq t$ and $\vec{\chi}(D[N_A(wv)]) \leq t$. Finally, $N_A^o(w)$ has independence

number at most $\alpha - 1$. Thus by the induction hypothesis, $\vec{\chi}(D[N_A^o(w)]) \leq h(t, \alpha - 1)$. Therefore, $\vec{\chi}(D[S_1]) \leq 2t + h(t, \alpha - 1)$.

- (ii) $S_2 = N_B^-(u)$. Then $D[S_2]$ has independence number at most $\alpha - 1$. Thus, by the induction hypothesis $\vec{\chi}(D[S_2]) \leq h(t, \alpha - 1)$.
- (iii) $S_3 = N_B^+(v)$. Then $D[S_3]$ has independence number at most $\alpha - 1$. Thus, by the induction hypothesis $\vec{\chi}(D[S_3]) \leq h(t, \alpha - 1)$.

Therefore, $\vec{\chi}(D[N_T(e)]) \leq 3 \cdot h(t, \alpha - 1) + 2t$. \diamond

Claim 5.3.5. *For any pair of vertices u, v in V , $\vec{\chi}(D[N_T^-(u) \cap N_T^+(v)]) \leq 15 \cdot h(t, \alpha - 1) + 10t$.*

Proof. Recall that D , and therefore T , is strongly connected. For any pair of vertices u, v , take the shortest path $(e_i)_{1 \leq i \leq k}$ from u to v in T . Any vertex in $N_T^-(u) \cap N_T^+(v)$ must be in the neighborhood $N_T(e_i)$ of some arc e_i of the shortest path. An arc from a vertex in $N_T(e_i)$ to a vertex in $N_T(e_j)$ is *forward* if $i < j$ and *backward* if $j < i$. There can be no arc in A from a vertex in $N_T(e_i)$ to a vertex in $N_T(e_j)$ for $j \geq i + 5$, or else there would be a shorter path from u to v . Thus, we can use five color palettes of $3 \cdot h(t, \alpha - 1) + 2t$ colors each, and color $N_T(e_i)$ with the color palette $i \bmod 5$. By Claim 5.3.4, each neighborhood $N_T(e_i)$ does not contain a monochromatic directed cycle of arcs from A . Because all forward arcs from A between different neighborhoods are bicolored, this results in a coloring with no monochromatic directed cycle of arcs from A . In total, this coloring uses $15 \cdot h(t, \alpha - 1) + 10t$ colors. \diamond

If we can find a pair of vertices u, v such that $\vec{\chi}(D[N_T^+(u) \cup N_T^-(v)])$ is small (i.e., bounded by a function of t and α), then we can use Claim 5.3.5 to bound $\vec{\chi}(D)$ and prove the theorem. To do this, we need a few more tools.

Claim 5.3.6. *If the tournament $T = (V, A \cup B)$ has a dominating set $\gamma^+(T)$ and an absorbing set $\gamma^-(T)$, then $\vec{\chi}(D) \leq |\gamma^+(T)| \cdot |\gamma^-(T)| \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2)$.*

Proof. We now define a coloring C of D . For each pair of vertices $u \in \gamma^-(T), v \in \gamma^+(T)$, we can color the set $N_T^-(u) \cap N_T^+(v)$ using a different palette of $15 \cdot h(t, \alpha - 1) + 10t$ colors by Claim 5.3.5. Each vertex w of $V \setminus (\gamma^-(T) \cup \gamma^+(T))$ can be colored this way; indeed for each such vertex w , there is some pair of vertices $u \in \gamma^-(T), v \in \gamma^+(T)$ such that $w \in N_T^-(u) \cap N_T^+(v)$. Moreover, each vertex in $\gamma^+(T) \cup \gamma^-(T)$ can be colored with its own color. If a vertex is assigned more than one color, simply use the first color it is given. This coloring uses a total of at most $|\gamma^+(T)| \cdot |\gamma^-(T)| \cdot (15 \cdot$

$h(t, \alpha - 1) + 10t) + |\gamma^+(T)| + |\gamma^-(T)| \leq |\gamma^+(T)| \cdot |\gamma^-(T)| \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2)$ colors. \diamond

Set $d = 3 \cdot h(t, \alpha - 1) + 2t$. Notice that $T = (V, A \cup B)$ is not necessarily d -arc-bounded, since $\bar{\chi}(T[N_T(e)]) \geq \bar{\chi}(D[N_T(e)])$. We now modify the definition of a jewel (defined in the previous section) for our current setting: A *jewel* is a subset $J \subset V$ such that J is a $(d + 1, \ell(\alpha, d + 1))$ -cluster in D , so $\bar{\chi}(D[J]) \geq d + 1$.

Definition 5.3.1. We define a jewel-chain in T of length p to be an ordered set $X = (X_i)_{1 \leq i \leq p}$ such that each X_i induces a jewel in D (i.e., $D[X_i]$ is a jewel), all X_i 's are disjoint, and $X_i \Rightarrow X_{i+1}$ for all $1 \leq i \leq p - 1$ (i.e., X_i is complete to X_{i+1} in T).

As in the previous section, we say that for a jewel chain X , an arc uv is *forward* if $u \in X_i$ and $v \in X_j$ and $i < j$. If $j < i$, then arc uv is *backward*. The next claim is similar, but not identical, to Observation 5.2.7. The subtle difference stems from the fact that we care about the chromatic number of jewel with respect to D rather than T .

Claim 5.3.7. A jewel-chain X contains no backward arcs in T .

Proof. Consider a backward arc $e = uv$, with $u \in X_j$ and $v \in X_i$ for $j > i$. It must be the case that $j > i + 1$, since all arcs between X_i and X_{i+1} are forward by definition. Then $X_{i+1} \subseteq N_T(e)$, and since $\bar{\chi}(D[N_T(e)]) \geq d + 1$, this contradicts Claim 5.3.4. Thus, all arcs with endpoints in distinct X_i 's must be forward. \diamond

Let X be a jewel-chain in T of maximum length, say p . Define Y to be the vertex set such that $Y \Rightarrow X_1$ in T . Then $D[Y]$ does not contain a jewel by assumption (otherwise, we could make the jewel-chain longer). By Corollary 5.3.3, since $D[Y]$ does not contain a $(d + 1, \ell(\alpha, d + 1))$ -cluster, $D[Y]$ contains a dominating set and an absorbing set, each of size at most $K(d + 1, \alpha)$. Notice that a dominating (absorbing) set in $D[Y]$ is also a dominating (absorbing) set in $T[Y]$. So we can apply Claim 5.3.6 to bound the chromatic number of $D[Y]$ by $(K(d + 1, \alpha))^2 \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2)$.

Moreover, the set $N_T^\pm(X_1)$ has chromatic number at most $\bar{\chi}(D[N_T^\pm(X_1)]) \leq \ell(d + 1, \alpha) \cdot d$. Finally, $v \in X_1$ can have in-neighbors in X_1 itself, but these can have chromatic number at most $|X_1| \leq \ell(d + 1, \alpha)$.

So for each vertex $v \in X_1$, we have

$$\bar{\chi}(D[N_T^-(v)]) \leq (K(d + 1, \alpha))^2 \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2) + \ell(d + 1, \alpha) \cdot (d + 1).$$

By the same argument, each vertex $u \in X_p$ has the same bound on $\vec{\chi}(D[N_T^+(u)])$. So we have

$$\vec{\chi}(D[N_T^+(u) \cup N_T^-(v)]) \leq 2((K(d+1, \alpha))^2 \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2) + \ell(d+1, \alpha) \cdot (d+1)).$$

By Claim 5.3.5, we have

$$\vec{\chi}[D] \leq 2((1 + (K(d+1, \alpha))^2) \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2) + \ell(d+1, \alpha) \cdot (d+1)).$$

Since $d = 3 \cdot h(t, \alpha - 1) + 2t$, we can define the function h as follows.

$$h(t, \alpha) = 2((1 + (K(3 \cdot h(t, \alpha - 1) + 2t + 1, \alpha))^2) \cdot (15 \cdot h(t, \alpha - 1) + 10t + 2) + \ell(3 \cdot h(t, \alpha - 1) + 2t + 1, \alpha) \cdot (3 \cdot h(t, \alpha - 1) + 2t + 1)).$$

So we have $\vec{\chi}[D] \leq h(t, \alpha)$, concluding the proof of the theorem. \square

5.4 Equivalence of Conjectures 5.1.1 and 5.1.2

[NSS23b] show that Conjecture 5.1.2 implies Conjecture 5.1.1. In this section, we prove that Conjecture 5.1.1 implies Conjecture 5.1.2, showing they are equivalent. Our main tool is Theorem 5.1.4.

Let s be a function such that $s(x) \geq x^2 \cdot s(x-1) + x$ and let T be a tournament. Recall that a $(t, s(t))$ -cluster is a subset S of V of size $s(t)$ such that $\vec{\chi}(T[S]) \geq t$. For brevity, we use t -cluster to denote a $(t, s(t))$ -cluster in this section.

Definition 5.4.1. *Define a t -heavy arc $e \in A(T)$ to be an arc such that $T[N(e)]$ contains a $(t-1)$ -cluster, and a t -light arc to be an arc that is not t -heavy.*

We start by including a proof from [NSS23b] of a standard inequality that relates the chromatic number of a tournament T to the chromatic number of its *backedge graph* G . We say G is the backedge graph of T under a given ordering of its vertex set $V(T) = \{v_1, \dots, v_n\}$ when for every $i < j$, $v_i v_j \in E(G)$ iff v_i is adjacent from v_j in T .

Claim 5.4.1. *Let T be a tournament, and G its backedge graph under a given ordering. Then $\vec{\chi}(T) \leq \chi(G) \leq \vec{\chi}(T)\omega(G)$, where $\omega(G)$ is the size of the largest clique in G .*

Proof. The first part of the inequality results from every stable set in G being a transitive set in T . Now take a transitive set X in T , and let $<_p$ be the partial order on the elements of X where for $v_i, v_j \in X$, $v_i <_p v_j$ iff $i < j$ and $v_j v_i \in E(T)$. This is a partially ordered set, since $v_i <_p v_j$ and $v_j <_p v_k$ implies $i < j < k$, $v_i, v_j, v_k \in X$ and $v_j v_i, v_k, v_j \in E(T)$, therefore $v_k v_i \in E(T)$ since X is transitive, and thus $v_i <_p v_k$. Then, every totally ordered subset of X is a clique of G , and is therefore of size at most $\omega(G)$. By the dual of Dilworth's theorem, X can be partitioned into $\omega(G)$ subsets, each an antichain of the poset and hence a stable set of G . Thus $\chi(G[X]) \leq \omega(G)$. It follows by using the same argument for every transitive set of T that $\chi(G) \leq \chi(T)\omega(G)$. \square

Let us now prove a lemma that will allow us to restate Conjecture 5.1.2. The proof is reminiscent of the proof of 3.7 in [BCC⁺13] and essentially the same as the proof of Lemma 3.4 in [AAC22]. Let clust be a function such that $\text{clust}(x) = x \cdot 2^{s(2x)} + s(2x) + 1$.

Lemma 5.4.2. *For all $c \geq 0$, in any tournament T with $\bar{\chi}(T) \geq \text{clust}(c)$ that has a $2c$ -cluster, there are two sets $A, B \subseteq V(T)$ such that $\bar{\chi}(T[A]), \bar{\chi}(T[B]) \geq c$ and all arcs between A and B go from vertices of A to vertices of B .*

Proof. Let $C \subset V(T)$ be a $2c$ -cluster. By the definition of a cluster, $|C| \leq s(2c)$. So there are at most $2^{s(2c)}$ ways of partitioning C . Consider any vertex $v \in V(T) \setminus C$. Then $(N^+(v) \cap C, N^-(v) \cap C)$ forms a partition of C . Thus, we can partition $V(T) \setminus C$ into at most $2^{s(2c)}$ subsets $(S_i)_{1 \leq i \leq 2^{s(2c)}}$ such that all vertices in a subset S_i have the same in-neighborhood and out-neighborhood in C (i.e., each vertex in S_i partitions C in the same way). If every S_i can be colored with at most c colors, T can be colored with at most $c \cdot 2^{s(2c)} + s(2c)$ colors. Therefore, since $\bar{\chi}(T) \geq \text{clust}(c) = c \cdot 2^{s(2c)} + s(2c) + 1$ by the condition of the lemma, there must exist some subset S_i with $\bar{\chi}(T[S_i]) \geq c$. Consider the partition $(N^+(v) \cap C, N^-(v) \cap C)$ of C for a vertex $v \in S_i$. Either $\bar{\chi}(T[N^+(v) \cap C]) \geq c$ or $\bar{\chi}(T[N^-(v) \cap C]) \geq c$, since $\chi(C) \geq 2c$. By definition, S_i is complete to $N^+(v) \cap C$ and complete from $N^-(v) \cap C$. Thus by setting $A = N^-(v) \cap C$ and $B = S_i$ if $\bar{\chi}(T[N^-(v) \cap C]) \geq c$, and $A = S_i$, $B = N^+(v) \cap C$ if $\bar{\chi}(T[N^+(v) \cap C]) \geq c$, we have found A and B with A complete to B and $\bar{\chi}(T[A]), \bar{\chi}(T[B]) \geq c$. \square

Let us restate Conjectures 5.1.1 and 5.1.2.

Conjecture 5.4.2 (Restatement of Conjecture 5.1.1). *There exists a function ee such that for every pair of integers $t, c \geq 1$, if a graph G satisfies $\chi(G) \geq \text{ee}(t, c)$ and*

$\omega(G) < t$, then there are subsets $A, B \subseteq V(G)$ with $\chi(G[A]), \chi(G[B]) \geq c$, such that there are no edges between A and B .

Conjecture 5.4.3. *There exists a function nss such that for every pair of integers $t, c \geq 1$, if a tournament T satisfies $\vec{\chi}(T) \geq \text{nss}(t, c)$ and T contains no t -cluster, then there are subsets $A, B \subseteq V(T)$ such that $\vec{\chi}(T[A]), \vec{\chi}(T[B]) \geq c$ and all arcs between A and B go from vertices in A to vertices in B .*

Conjecture 5.4.3 may seem weaker than Conjecture 5.1.2, but is in fact equivalent. This is a direct consequence of Lemma 5.4.2. Indeed, for any c , if a tournament T has no sets A and B with A complete to B and $\vec{\chi}(T[A]), \vec{\chi}(T[B]) \geq c$, then by the contrapositive of Lemma 5.4.2 it has no $2c$ -cluster or it has chromatic number less than $\text{clust}(c)$. Therefore, Conjecture 5.4.3 will imply that T has chromatic number strictly less than $d = \max(\text{nss}(2c, c), \text{clust}(c))$, which is some constant since c is fixed. This is exactly the contrapositive of Conjecture 5.1.2. We now state the contrapositive of Conjecture 5.4.3, which is also equivalent to Conjecture 5.1.2.

Conjecture 5.4.4 (Restatement of Conjecture 5.4.3). *There exists a function nss such that for every pair of integers $t, c \geq 1$, if a tournament T contains no t -cluster and T does not contain subsets $A, B \subseteq V(T)$ such that $\vec{\chi}(T[A]), \vec{\chi}(T[B]) \geq c$ with all arcs between A and B going from vertices in A to vertices in B , then $\vec{\chi}(T) \leq \text{nss}(t, c)$.*

5.4.1 Proof of Conjecture 5.1.2, assuming Conjecture 5.1.1

Proof of Conjecture 5.4.4, assuming Conjecture 5.4.2. For $t = 2$, a tournament T with no 2-cluster does not contain a directed triangle and therefore has $\vec{\chi}(T) = 1$. Thus, we have $\text{nss}(2, c) = 1$. Now we assume that $\text{nss}(t - 1, c)$ exists. We will prove that $\text{nss}(t, c)$ exists.

We consider a tournament T , which by assumption does not contain a t -cluster. Since t is now fixed for the rest of this proof, we simply use heavy and light in place of t -heavy and t -light. Let L be the set of light arcs and H the set of heavy arcs. Notice that every arc in T must be either in L or in H . Let $D_H = (V, H)$ and $D_L = (V, L)$ be digraphs containing the heavy and light arcs, respectively. Let $G_H = (V, H)$ denote the undirected graph of heavy edges and let $G_L = (V, L)$ denote the undirected graph of light edges. (Notice that we are abusing notation by using H and L to refer to both directed and undirected edge sets.)

Our first claim is that the graph G_H has no large clique, and consequently, the graph G_L has bounded independence number.

Claim 5.4.3. $\omega(G_H) \leq t - 1$.

Proof. Suppose that G_H contains a K_t (i.e., a clique on t vertices) and let S be the set obtained by including the t vertices of the clique in addition to the vertices in the $(t - 1)$ -cluster in the neighborhood of each arc corresponding to an edge in the clique. Then S has at most $t + t^2 \cdot s(t - 1)$ vertices. Moreover, $T[S]$ cannot be colored with $t - 1$ colors since every arc is heavy and the endpoints of a heavy arc cannot have the same color in any coloring using only $t - 1$ colors. Since S contains a clique, we have that $\chi(S) \geq t$. Thus, T contains a t -cluster, which is a contradiction. \diamond

Claim 5.4.4. $\alpha(G_L) \leq t - 1$.

Proof. L and H are complementary edge sets (i.e., every edge not in L belongs to H and vice versa). If G_L has an independent set of size t , then G_H would have a clique on those same t vertices, which would contradict Claim 5.4.3. \diamond

Claim 5.4.5. For every arc $e \in L$, $\vec{\chi}(T[N(e)]) \leq \text{nss}(t - 1, c)$.

Proof. By definition, the neighborhood of any light arc contains no $(t - 1)$ -cluster. Thus by the induction hypothesis it can be colored with $\text{nss}(t - 1, c)$ colors. \diamond

It follows immediately that the neighborhood of every arc in D_L has chromatic number at most $\text{nss}(t - 1, c)$. We can then use Theorem 5.1.4 to show that D_L can be colored with $h(\text{nss}(t - 1, c))$ of colors.

Fix such a coloring of D_L . Each color induces a tournament that has a vertex ordering in which each backward arc belongs to H (since all monochromatic arcs with the same color from L form an acyclic digraph). Consider the subtournament T_i induced on vertices with the i^{th} color, let n denote the number of vertices in this subtournament and fix a vertex ordering $\{v_1, \dots, v_n\}$ in which all arcs in D_L are forward. Let G_i be the undirected graph on this vertex set whose edge set corresponds to the backward arcs of T_i with respect to the fixed vertex ordering. Notice that G_i is a subgraph of G_H , which is K_t -free by Claim 5.4.3.

Now let us apply Conjecture 5.4.2 to the graph G_i . Let $c_2 = 2tc$. Either each G_i has chromatic number at most $d = \text{ee}(t, c_2)$ or G_i contains two sets S_1 and S_2 with $\chi(G[S_1]), \chi(G[S_2]) \geq c_2$ and with no edges in G_i between S_1 and S_2 . In the latter case, let a be the smallest index such that $\chi(G[\{v_1, \dots, v_a\} \cap S_1]) \geq tc$, and let b be the smallest index such that $\chi(G[\{v_1, \dots, v_b\} \cap S_2]) \geq tc$. Without loss of generality, assume that $a < b$. Now let $A' = \{v_1, \dots, v_a\} \cap S_1$ and $B' = \{v_{b+1}, \dots, v_n\} \cap S_2$. Observe that since S_1 and S_2 have no arcs between them in G_i , which corresponds

to the backedge graph of T_i , then all arcs between A' and B' in T_i must go from A' to B' . Moreover, we have $\bar{\chi}(T_i[A']) \leq \chi(G_i[A']) \leq \omega(G_i[A'])\bar{\chi}(T_i[A'])$ from Claim 5.4.1. Since $\chi(G_i[A']) \geq tc$ and $\omega(G_i[A']) \leq \omega(G_i) \leq t$, we have $\bar{\chi}(T_i[A']) \geq c$. Using the same argument, we also have $\bar{\chi}(T_i[B']) \geq c$. However, by assumption, such sets A' and B' do not exist in T . So we conclude that we are in the first case, in which $\bar{\chi}(T_i) \leq \chi(G_i) \leq ee(t, c_2)$.

Thus, we can color the subtournament induced by each color class of D_L with $ee(t, 2tc)$ colors, resulting in a coloring of T with $nss(t, c) = ee(t, 2tc) \cdot h(nss(t-1, c))$ colors. \square

5.4.2 Proof of Conjecture 5.1.1, assuming Conjecture 5.1.2

For the sake of completeness, we include the proof from [NSS23b] that Conjecture 5.1.2 implies Conjecture 5.1.1. Combined with our proof that Conjecture 5.1.1 implies Conjecture 5.1.2, this shows the equivalence of both conjectures.

Proof of Conjecture 5.1.1, assuming Conjecture 5.1.2. We will prove the statement by induction on t . The statement is trivial for $t = 2$, since a graph with no clique of size 2 has chromatic number 1, so we can simply choose $d = 2$ for any c .

Then suppose for any graph G with $\omega(G) < t - 1$, and $\chi(G) \geq d$, there exist two anticomplete subsets A and B of $V(G)$, both with chromatic number at least c . Let $c_T = \max(2d, 2c)$, and let d_T be the constant obtained from the assumed truth of Conjecture 5.1.2 such that any tournament T with chromatic number at least d_T has a complete pair two subsets A and B with A complete to B , and $\chi(A), \chi(B) \geq c_T$.

Let $d' = t \cdot d_T$, and let G be a graph with $\chi(G) \geq d'$ and $\omega(G) < t$. Our goal is to show that there are two anticomplete subsets A and B of $V(G)$ both with chromatic number at least c .

Let $V(G) = \{v_1, \dots, v_n\}$ and let T be the tournament such that G is its backedge graph under the numbering v_1, \dots, v_n . From Claim 5.4.1, we get $\chi(T) \geq d'/t = d_T$. By the assumed Conjecture 5.1.2, there are two disjoint subtournaments A_T, B_T of T with A_T complete to B_T , and $\chi(A_T), \chi(B_T) \geq c_T$. Then choose the smallest integer i such that either $\{v_1, \dots, v_i\} \cap A_T$ or $\{v_1, \dots, v_i\} \cap B_T$ induces a tournament with chromatic number at least $c_T/2$.

If $\{v_1, \dots, v_i\} \cap A_T$ induces a tournament with chromatic number at least $c_T/2$, define $A = \{v_1, \dots, v_i\} \cap A_T$ and $B = \{v_{i+1}, \dots, v_n\} \cap B_T$. By the minimality of i ,

$\chi(T[B]) \geq c_T - \chi(T \setminus B) \geq c_T/2 \geq c$. Since A is complete to B , and all arcs from A to B are increasing in the ordering $\{v_1, \dots, v_n\}$, A and B are anticomplete in G . Moreover, from Claim 5.4.1, $\chi(G[A]) \geq \chi(T[A]) \geq c$ and $\chi(G[B]) \geq \chi(T[B]) \geq c$, thus A and B are the two desired subsets of G .

Else, $\{v_1, \dots, v_i\} \cap B_T$ induces a tournament with chromatic number at least $c_T/2$. Then define $A = \{v_{i+1}, \dots, v_n\} \cap A_T$ and $B = \{v_1, \dots, v_i\} \cap B_T$. Since A is complete to B , and all arcs from A to B are decreasing in the ordering $\{v_1, \dots, v_n\}$, A and B are complete in G . By the minimality of i , $\chi(T[B]) \geq c_T - \chi(T \setminus B) \geq c_T/2 \geq d$. As previously, Claim 5.4.1 implies $\chi(G[B]) \geq \chi(T[B]) \geq d$ and since B is contained in the outneighborhood of any vertex in A (which cannot be empty since i is minimal), $\omega(G[B]) \leq t - 1$. Then by the induction hypothesis, there are anticomplete subsets A' and B' of B such that $\chi(G[A']), \chi(G[B']) \geq c$.

□

Chapter 6

Conclusion of Part I

As we have seen in Chapter 2 coloring tournaments is strongly related to coloring graphs, in the sense that if k -colorable graphs could be colored with a constant number of colors, so could k -colorable tournaments. 3-colorable tournaments have an even stronger relation to 3-colorable graphs: they can be colored with at most 50 times the number of colors needed for 3-colorable graphs. While these results may be improved, the case of 2-coloring is interesting since for graphs, it is easy to efficiently color 2-colorable graphs in polynomial time, while it is **NP**-hard to color 2-colorable tournaments with 2 colors [CHZ07]. This implies that there can be no general reduction from coloring tournaments to coloring graphs that preserves the chromatic number of the instance.

Another observation is the relation of coloring tournaments and the feedback vertex set (FVS) problem on tournaments. There is an elegant 2-approximation for this problem [LMM⁺21]. Notice that Theorem 2.2.1 implies that in a 2-colorable tournament, we can efficiently find a FVS of size at most $9n/10$. In contrast, the algorithm in [LMM⁺21] could just return the whole vertex set if the two transitive sets were of roughly equal size. Finally, we mention that, analogous to a well-studied question for general graphs [DKPS10, KS14], one can ask what is the largest transitive induced subtournament that one can efficiently find in a 2-colorable tournament? Is it larger than $n/10$?

We also remark that an implication of Theorem 2.3.2 is that proving any hardness of coloring 3-colorable tournaments would then provide hardness of coloring 3-colorable graphs with 50 times fewer colors. Since it has taken around 20 years to go from proving **NP**-hardness of coloring a 3-colorable graph with four colors [KLS00,

GK00, GK04] to NP-hardness of coloring a 3-colorable graph with five colors [BKO19], it would be interesting to see if we can prove hardness of coloring 3-colorable tournaments for a constant larger than five (at least five is shown in Theorem 3.2.5), or perhaps show that the two problems are actually equivalent.

Our work proving the equivalence of Conjectures 5.1.2 and 5.1.1 leaves open the question of whether these conjectures are true. Proving or disproving one of them would be very interesting as it would also provide results for the other setting. In fact, a weaker formulation of the conjecture on graphs, replacing cliques with clusters is also equivalent to them.

Finally, the graph theoretic results we obtained in Chapter 5 are based on the algorithmic tools developed in the previous chapters. This is interesting, as structural results are often the basis for algorithms, and in this case the reverse was also true, highlighting the strong link between the fields of graph theory and algorithmics.

Part II

Cycle-plus-Triangles Graphs

Chapter 7

Introduction

Many modern combinatorial techniques supply existence proofs but do not yield efficient algorithms for the respective problems [Alo90, Alo93]. This is the case for the 3-colorability of a CYCLE-PLUS-TRIANGLES graph [Alo02, Alo22]. A CYCLE-PLUS-TRIANGLES graph is the disjoint union of t vertex disjoint triangles and a Hamilton cycle on the $n = 3t$ vertices. (Notice that a CYCLE-PLUS-TRIANGLES graph can be a multigraph since an edge can belong to both a triangle and to the Hamilton cycle.) Such a graph can be recognized in polynomial time, as shown by [BK17], who also gave an efficient algorithm for finding a decomposition into a Hamilton cycle and a set of vertex disjoint triangles. It was conjectured by Du, Hsu and Hwang that a CYCLE-PLUS-TRIANGLES graph has a stable set ¹ of size $n/3$ [DHH93]. This was extended by Erdős who conjectured that a CYCLE-PLUS-TRIANGLES graph is 3-colorable, ² which was proved by Fleischner and Stiebitz [FS92]. Interestingly, Fellows observed that the conjecture of Erdős is equivalent to a conjecture of Schur which states that for any partition of the integers into triples, there is a 3-coloring of the integers such that each color class contains a member of each triple but no pair of consecutive integers [Fel90].

The proof of Fleischner and Stiebitz is based on an application of an algebraic tool of Alon and Tarsi and is nonconstructive [AT92]. Subsequently, Sachs gave an “elementary” proof of this theorem [Sac94], which yields an algorithm, but the algorithm is not guaranteed to run in polynomial time. In fact, currently, we do not even know how to find a stable set of size $n/3$ in polynomial time, although due to

¹A stable set is a subset of vertices that induce a subgraph with no edges.

²A graph is 3-colorable if its vertex set can be partitioned into three stable sets.

previously cited results, it is guaranteed to exist. (Notice that a maximum stable set in a CYCLE-PLUS-TRIANGLES graph has size $n/3$.) Alon et al. gave another (also nonconstructive) proof based on topological methods for proving that a CYCLE-PLUS-TRIANGLES graph contains two disjoint stable sets each of size $n/3$ [AAB⁺17]. Topological tools were also used to study the case in which the Hamilton cycle is replaced with a disjoint union of cycles, in which case a stable set of size $n/3$ is known to exist if at most two of the cycles have length equal to $1 \pmod 3$ [AHHS15].

In this thesis, we present a new randomized algorithm for finding a maximum stable set in a CYCLE-PLUS-TRIANGLES graph. Our algorithm is simple and easy to implement, and our approach is not based on any of the previously mentioned (non-constructive) methods. Rather, it is loosely inspired by constructive algorithms for the Lovász Local Lemma (LLL) [MT10]. The general framework of these algorithms are to find a “flaw” and resample the relevant set of variables. If certain conditions are met, then it can be shown that such an algorithm terminates quickly. It is not known how to use the LLL to prove the existence of a stable set of size $n/3$ in a CYCLE-PLUS-TRIANGLES graph, and therefore this line of research has not led to efficient algorithms for this problem. Nevertheless, our algorithm, which can be viewed as resampling flaws, runs quickly in practice and outputs a maximum stable set upon termination. The study of what parameters are actually required to guarantee the efficiency of resampling-based algorithms is an area of ongoing study [CCS⁺17, HLS23]. In any case, we have not been able to find instances on which our algorithm fails (i.e., on which it does not terminate after a number of steps that is roughly linear in the instance size). While we are not able to prove that it is efficient, we conjecture that it runs in expected polynomial time. We also implemented a deterministic version of our algorithm based on the rotor-router model, which seems to run faster than the randomized algorithm most of the time. However, we found instances in which this derandomized version does not terminate.

To understand the limitations of our framework, we considered the problem of finding two or three maximum stable sets in a CYCLE-PLUS-TRIANGLES graph. Additionally, we considered other generalizations such as FAIR-REPRESENTATION. In the latter problem, we are given a Hamilton cycle $H = (V, E)$ and a partition of the vertex set $V = \{V_1, V_2, \dots, V_\ell\}$. The goal is to find a stable set of the cycle $I \subset V$ such that $|I \cap V_i| \geq |V_i|/2 - 1$ for all $i \in \{1, \dots, \ell\}$. This problem has a nonconstructive proof [AAB⁺17] and was shown to be PPA-complete [Hav21], which suggests that finding an efficient algorithm for it is unlikely. Our algorithm easily extends to these problems, but does not perform well. This leads us to believe that finding a maximum stable set in a CYCLE-PLUS-TRIANGLES graph is an easier problem than

any of these generalizations.

7.1 Organization of Part II

In the rest of this chapter, we start by presenting basic notation in Section 7.2. We then give an overview of the different proofs of 3-colorability and the proof of existence of a stable set of size $n/3$. We present in detail the “elementary” proof of 3-colorability by Sachs [Sac94], and give a brief synopsis of the proof of 3-colorability of Fleischner and Stiebitz [FS92], and the proof of independence number $n/3$ by [AAB⁺17].

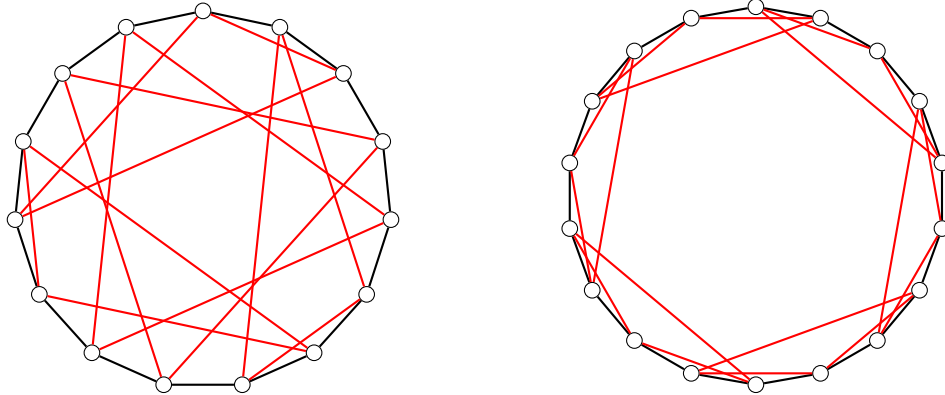
In Chapter 8, we first discuss the relation between instances of the CYCLE-PLUS-TRIANGLES problem and random permutations in Section 8.1. We then present our algorithms, both randomized and deterministic, for finding a maximum stable set in a CYCLE-PLUS-TRIANGLES graph in Section 8.2. In Section 8.3, we give some intuition as to why our algorithm seems to always terminate in a polynomial number of steps, and prove efficient termination on some families of instances in Section 8.4.

Chapter 9 focuses on an empirical evaluation of the algorithms. In Section 9.1, we present methods to generate instances of CYCLE-PLUS-TRIANGLES graphs. In Section 9.2, we present experimental results of running our algorithm on various sets of instances, giving evidence of its efficiency. Then in Section 9.3, we discuss the limitations of the deterministic algorithm.

Finally, Chapter 10 contains results about extensions of our algorithms, either to problems more general than finding a maximum independent set in a CYCLE-PLUS-TRIANGLES graph, or to graphs that are not CYCLE-PLUS-TRIANGLES graphs but closely related. In Section 10.1, we discuss extensions of our algorithms to coloring and fair representation. In Section 10.2 we examine the behavior of our algorithms on TRIANGLES-PLUS-TWO-FACTOR graphs, which are a superset of CYCLE-PLUS-TRIANGLES graphs.

Our code can be found at: <https://anonymous.4open.science/r/CT-ss-alg-7C9C>.

This part of the thesis is joint work with Duc-Anh Do, Moritz Mühlenthaler, Alantha Newman and Heiko Röglin. We also thank John Sylvester for encouraging us to look at potential function arguments.



(a) An instance for the CYCLE-PLUS-TRIANGLES problem. (b) A particular instance for the CYCLE-PLUS-TRIANGLES problem, the chain-of-links.

Figure 7.1: CYCLE-PLUS-TRIANGLES graphs. The Hamilton cycle edges are in black and the triangle edges in red.

7.2 Notation

Let \mathcal{G} denote the class of all CYCLE-PLUS-TRIANGLES graphs and let \mathcal{G}_n denote the class of all CYCLE-PLUS-TRIANGLES graphs on n vertices. We say that $G \in \mathcal{G}_n$ is a CYCLE-PLUS-TRIANGLES *instance* (or just *instance*) where $G = (V, E_T \cup E_H)$ and E_T denotes the edge set of the $t = n/3$ triangles and E_H denotes the edge set of the Hamilton cycle of length n . An edge can belong to both E_H and to E_T , in which case the instance is a multigraph.

We use $\text{dist}_H(i, j)$ to denote the length of the shortest path between vertices i and j on the Hamilton cycle H . The *span* of a triangle in G is the length of its shortest side, where length is computed with respect to $\text{dist}_H(\cdot)$. The span of the graph G is the minimum span of all triangles in G . We use $\sigma(G)$ to denote the span of G . For example, in Figure 9.1b, the instance has span three, and in Figure 9.1a, the instance has span four. If an instance has span at least two, then it is not a multigraph.

Note that an instance is unique up to isomorphism; that is, two graphs represent the same instance if and only if they are isomorphic. In fact, if $\sigma(G) \geq 3$, then there is a unique decomposition of the edges of G into E_H and E_T . (This is formally stated and proved in Lemma 9.1.1.)

7.3 Proof of 3-colorability by Fleischner and Stiebitz

The first proof of 3-colorability of CYCLE-PLUS-TRIANGLES graphs is due to Fleischner and Stiebitz [FS92]. The proof is based on counting the number of arc subsets of an orientation of the graph that induce an Eulerian subdigraph. A directed graph is said to be Eulerian iff each vertex has equal in-degree and out-degree. They introduce the parameter $e(D)$ for a digraph $D = (V, A)$, defined as $e(D) = |\{E \subseteq A : D[E] \text{ is Eulerian}\}|$, where $D[E]$ denotes the digraph induced by the arc set E on the vertex set V . They then use the following corollary of theorems of Alon and Tarsi based on the polynomial method [AT92].

Corollary 7.3.1. *If G is a $2k$ -regular graph on n vertices which has an Eulerian orientation D satisfying $e(D) = 2 \pmod{4}$, and if the number of edges is even, then $\chi(G) \leq k + 1$.*

When G is a CYCLE-PLUS-TRIANGLES graph, it is the case that $k = 2$. Thus, proving that G has an Eulerian orientation D satisfying $e(D) = 2 \pmod{4}$ implies the 3-colorability of G . They prove this statement for the orientation obtained by directing all arcs of the Hamilton cycle such that they form a directed Hamilton cycle, and all arcs of each triangle in a way that they form a directed triangle. The proof is then done by induction on the number of triangles, by showing that this directed graph can be reduced to directed graphs with the same decomposition but with fewer triangles. They then show that $e(D)$ can be written as a function of the parameter e of these smaller digraphs, on which the induction hypothesis can be applied.

This allows them to prove their main theorem, which is the following.

Theorem 7.3.2. *Let G be a CYCLE-PLUS-TRIANGLES graph. Then $\chi(G) = 3$.*

The authors also mention that this theorem can be obtained without using the results of Alon and Tarsi based on the polynomial method, but rather the theory of nowhere-zero \mathbb{Z}_3 -flows and the flow polynomial.

7.4 Sachs' proof of 3-colorability

The following proof by Sachs of the 3-colorability of CYCLE-PLUS-TRIANGLES graphs can be considered elementary because it is based solely on combinatorial arguments, and uses no theorem established outside of this work.

The idea of the proof is to show that the number of 3-colorings of a CYCLE-PLUS-TRIANGLES graph, up to permutation of the colors, is odd, and thus cannot equal zero. To prove that, the author starts by defining the number of normalized 3-colorings of a graph, $\pi(G)$.

Definition 7.4.1. *Given a CYCLE-PLUS-TRIANGLES graph $G = (V, E)$, we will say a 3-coloring $c : v \rightarrow 1, 2, 3$ is normalized if for two fixed adjacent vertices v_1 and v_2 , $c(v_1) = 1$ and $c(v_2) = 2$. It is clear that every 3-coloring can be normalized by a permutation of the colors. Then, let $\pi(G)$ be the number of normalized colorings of G . This is also the number 3-colorings of G that are different up to a permutation of the colors.*

The main theorem that Sachs' proves is the following:

Theorem 7.4.1. *For any CYCLE-PLUS-TRIANGLES graph G , $\pi(G) \equiv 1 \pmod{2}$.*

The proof of this theorem relies on a double induction, first on the number of triangles of the instance, and then on its span. Sachs shows that for a given graph G , the parity of $\pi(G)$ can be deduced from the parity of graphs with strictly lower span (and is odd if they are all odd), and then that graphs of low span can be reduced to graphs with fewer triangles whose number of normalized 3-colorings share the same parity.

7.4.1 Reduction for graphs with low span

Lemma 7.4.2. *Given a CYCLE-PLUS-TRIANGLES graph G on n triangles, with span at most two, there exists a CYCLE-PLUS-TRIANGLES graph G' on $n - 1$ triangles such that every coloring of G' can be extended to a coloring of G . Furthermore, $\pi(G') \equiv \pi(G) \pmod{2}$.*

Though this formulation of the lemma is more restrictive than Sachs' Lemma 1, and only the equivalence on the parity of the number of normalized 3-colorings is

necessary to prove the 3-colorability of CYCLE-PLUS-TRIANGLES graphs, it follows from the same proof, and we find it interesting to point out that instances with span 1 or 2 can in fact be reduced to smaller instances, from which every 3-coloring can be extended.

Proof. There are three different types of reduction that need to be considered in order to prove the lemma. They are all represented in Figure 7.2.

The first is when there exists a triangle t with span one that has two edges of span one. In that case, G' is obtained by removing the triangle and contracting the Hamilton cycle such that the two vertices that had a neighbor in t , y and z , become neighbors on the Hamilton cycle.

The second case is when there is a triangle t with span one, but only one of its edges is of span one. G' is similarly obtained by removing the triangle and contracting all Hamilton cycle edges. For example, y and z the neighbors of the vertices of t that are at distance one of each other, will become Hamilton cycle neighbors.

Finally, in the case of a triangle t of span two, call u and v the vertices of t at distance two on the Hamilton cycle. Then, call w the third vertex in their triangle, and x their common neighbor on the Hamilton cycle. G' is obtained by first swapping the positions of w and x on the Hamilton cycle, and then removing the triangle t . As previously, y and z the two neighbors of v and u respectively, become neighbors on the Hamilton cycle.

In all cases, the proof is done in two parts, by splitting the 3-colorings of G into two sets: those in which y and z have the same color, and those in which they don't. One can then show through case analysis that the first set has an even size, and that the second set has an odd size.

□

7.4.2 Transformation towards graphs of lower span

The author then proves a lemma that allows him to deduce the parity of the number of colorings of a graph from that of graphs obtained by applying two specific types of transformations, which can both be used to make the span smaller.

Definition 7.4.2. *For a CYCLE-PLUS-TRIANGLES graph G , define the interchange of u and v in G (who are adjacent on the Hamilton cycle), $I(G, u, v)$, to be the graph obtained from G by swapping the places of u and v on the Hamilton cycle.*

For a CYCLE-PLUS-TRIANGLES graph G , define the bitransplantation of u , v , s and t in G , $B(G, u, v, s, t)$, to be the graph obtained from G by moving the two vertices that are between u and v on the Hamilton cycle (call them x and y) between the vertices s and t (which are adjacent on the Hamilton cycle), such that (s, x, y, t) becomes a path on the Hamilton cycle. By this procedure, u and v become neighbors on the Hamilton cycle.

Lemma 7.4.3. *Let $G^1 = (V, (E_T, E_H))$ be a CYCLE-PLUS-TRIANGLES graph, and (x, y) , (s, t) a pair of edges in E_H . Call u and v the respective neighbors of x and y in E_H (other than themselves).*

Define $G^2 = I(G^1, x, y)$, $\tilde{G}^1 = B(G^1, u, v, s, t)$ and $\tilde{G}^2 = B(G^2, u, v, s, t) = I(G^1, x, y)$.

Then, $\pi(G^1) + \pi(G^2) \equiv \pi(\tilde{G}^1) + \pi(\tilde{G}^2) \pmod{2}$.

Proof. The proof of this lemma relies on examining the colorings such that u and v , or s and t have the same color, and when they don't. When they do, it is straightforward that any coloring of the graph is also a coloring of the graph where the two vertices between them are interchanged, thus the number of such colorings is even on both sides of the equality. When this is not the case, a case analysis shows that any coloring of either of the first two graphs gives exactly one coloring of either of the two latter. \square

7.4.3 Proof by induction

The goal is to prove by induction, first on the number of triangles, then on the span of the instance, that for any CYCLE-PLUS-TRIANGLES graph G , $\pi(G) \equiv 1 \pmod{2}$.

Induction on the number of triangles t

Initialization (size): For $t = 1$, there is only one CYCLE-PLUS-TRIANGLES graph with a single triangle, and $\pi(G) = 1 \equiv 1 \pmod{2}$.

Induction hypothesis (size): For any CYCLE-PLUS-TRIANGLES graph G on t triangles, $\pi(G) \equiv 1 \pmod{2}$.

Take a CYCLE-PLUS-TRIANGLES graph G on $t + 1$ triangles. Then, we will show by induction on the span of the instance, that $\pi(G) \equiv 1 \pmod{2}$.

Induction on the span σ

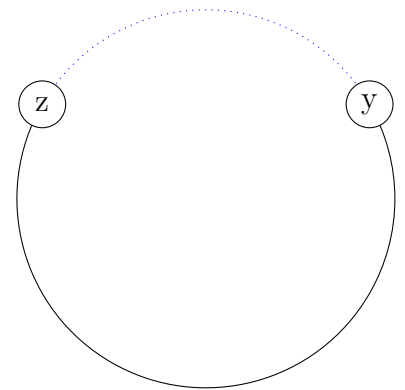
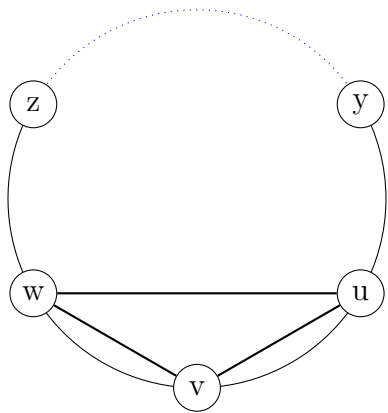
Initialization (span): If $\sigma(G) \leq 2$, we can use a reduction from Lemma 7.4.2, which yields $\pi(G) = \pi(G')$ with G' having only t triangles. Thus, by the induction hypothesis, $\pi(G) = \pi(G') \equiv 1 \pmod{2}$.

Induction hypothesis (span): For any CYCLE-PLUS-TRIANGLES graph G on $t + 1$ triangles with span at most σ , $\pi(G) \equiv 1 \pmod{2}$.

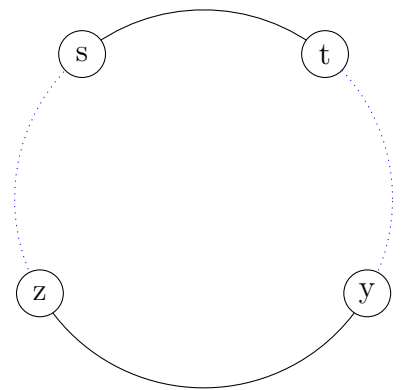
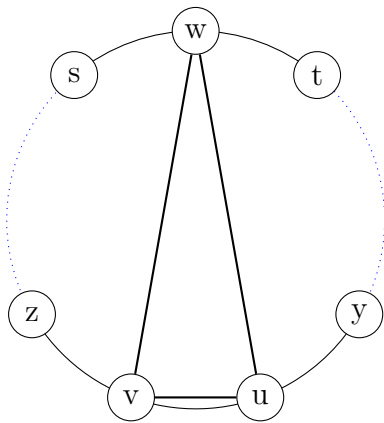
Take a CYCLE-PLUS-TRIANGLES graph G on $t + 1$ triangles with span $\sigma + 1$. Notice that you can choose u, v, x, y, s and t such that $G = G^1$, $G^2 = I(G, x, y)$ has span σ , $\tilde{G}^1 = B(G^1, u, v, s, t)$ has span 2, and $\tilde{G}^2 = B(G^2, u, v, s, t)$ has span 1. This is done by interchanging one of the endpoints of the edge with span $\sigma + 1$ with its neighbor in the direction of the other endpoint of that edge. It is clear that this interchange will reduce the span of that edge. The bitransplantation can then move these two vertices between the third vertex of the triangle with low span, and one of its neighbors on the Hamilton cycle. A representation of such a configuration is given in Figure 7.4.

Then, from lemma 7.4.3, $\pi(G) + \pi(G^2) \equiv \pi(\tilde{G}^1) + \pi(\tilde{G}^2) \pmod{2}$, or equivalently, $\pi(G) \equiv \pi(G^2) + \pi(\tilde{G}^1) + \pi(\tilde{G}^2) \pmod{2}$, and from the induction hypothesis, $\pi(G^2) \equiv \pi(\tilde{G}^1) \equiv \pi(\tilde{G}^2) \equiv 1 \pmod{2}$, implying $\pi(G) \equiv 1 \pmod{2}$.

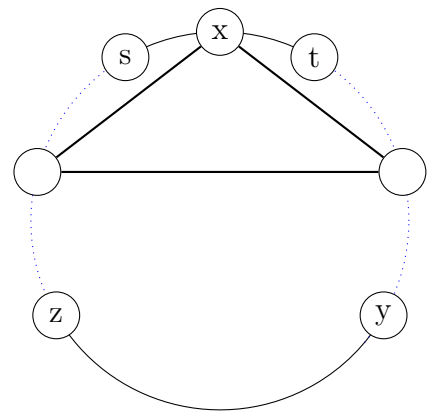
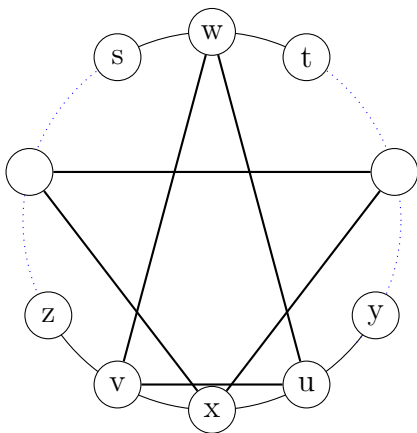
This concludes the induction on the span, thereby also concluding the induction on the size of the instance.



(a) Case **A**

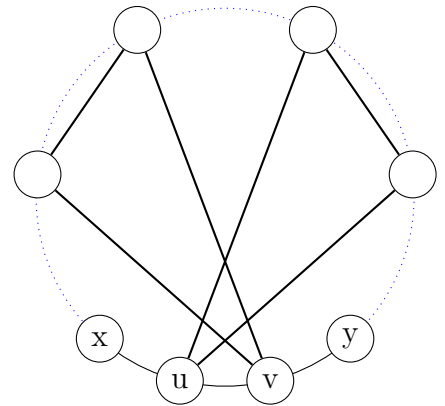
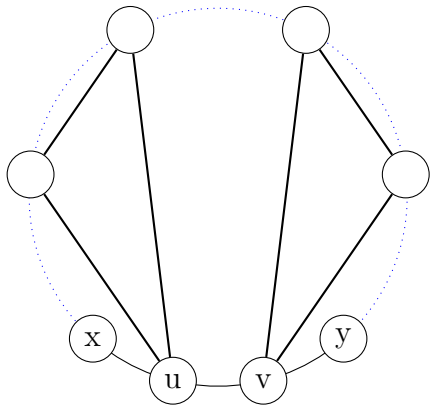


(b) Case **B**

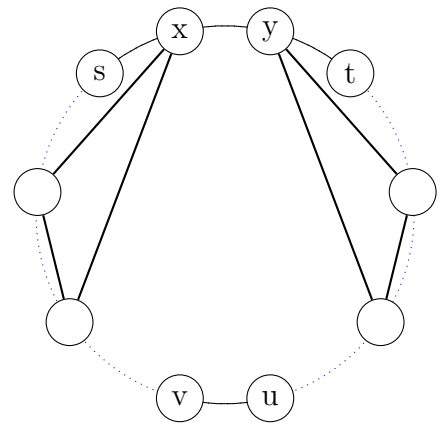
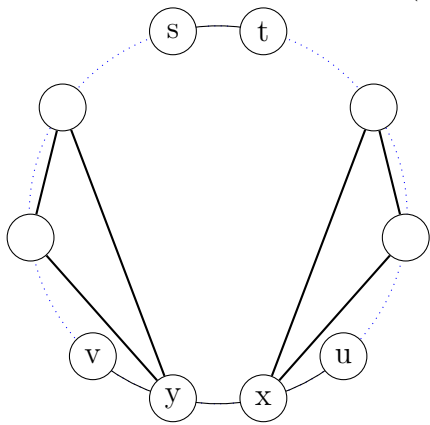


(c) Case **C**

Figure 7.2: The three cases of reductions to lower span instances. A blue dotted line represents a path on the Hamilton cycle \mathcal{H}_2 . Note that in case **C**, the two vertices in the same triangle as x can also both be on the same path of the Hamilton cycle (ie. both between t and y).

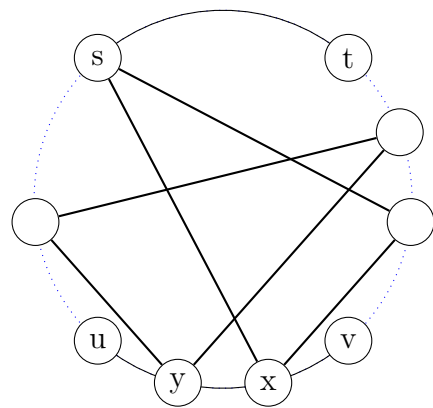
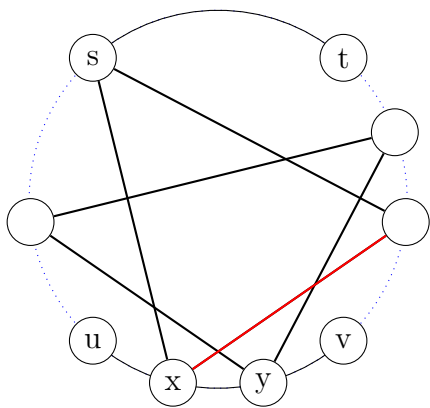


(a) Interchange of u and v

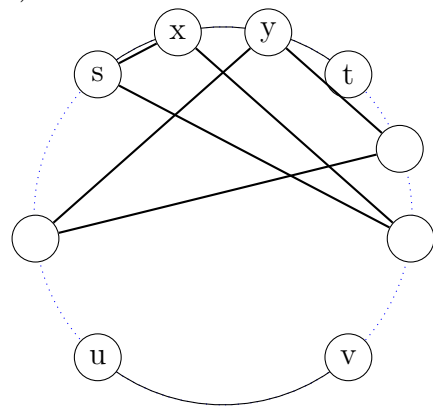
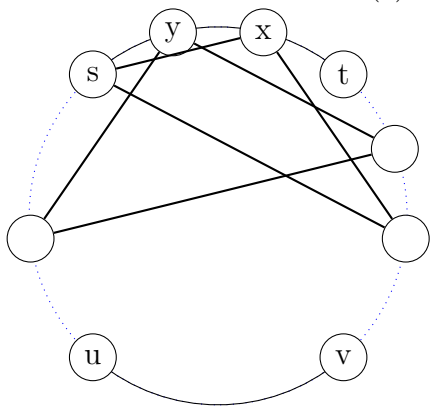


(b) Bitransplantation of u, v, s and t

Figure 7.3: Interchange and Bitransplantation. A blue dotted line represents a path on the Hamilton cycle. The unnamed vertices can be anywhere on either side of the dotted paths.



(a) $G = G^1$ (left) and G^2 (right)



(b) \tilde{G}^1 (left) and \tilde{G}^2 (right)

Figure 7.4: Choice of vertices to form a double pair where all other graphs have lower span than G^1 . The red edge has span $\sigma + 1$.

7.5 Alon's proof of independence number $n/3$

Schrijver proves the following theorem [Sch78]:

Theorem 7.5.1. *The family $I(n, k)$ of independent sets of size k in the cycle C_n cannot be partitioned into fewer than $n - 2k + 2$ intersecting families.*

The general idea behind the proof is to build a Kneser graph from C_n (where each node is an independent set of C_n , and adjacent to another iff they do not intersect). Then, an independent set of the Kneser graph is an intersecting family of C_n . Therefore, proving that the Kneser graph has a given chromatic number will give as a corollary the previous theorem.

To prove the chromatic number of the Kneser graph is $n - 2k + 2$ when $n > 2k$, Schrijver embeds the vertices with unit vectors on a k -dimensional sphere, and uses polynomial representations in order to show that hemispheres enclose subsets of $n/2 - k$ vertices. Then, supposing there is a partition of the nodes into $k + 1$ families, he shows that some families contains two antipodal points of the sphere using Borsuk's theorem, which according to the embedding means they are disjoint sets.

Then, Alon et al. prove the following [AAB⁺17]:

Theorem 7.5.2. *For $G = C_n$, if $\{V_1, \dots, V_n\}$ is a partition of the vertex set, there exists an independent set I of G , such that for each V_i , $|I \cap V_i| = (|V_i| - 1)/2$.*

The proof is done by contradiction: if there is no such set I , then all independent sets of size $\sum_i (|V_i| - 1)/2 < n/2$ must have some i for which $|I \cap V_i| \geq (|V_i| - 1)/2 + 1$. Then he defines a family F_i to be all independent sets such that $|I \cap V_i| \geq (|V_i| - 1)/2 + 1$. Each family F_i is intersecting, and there are fewer than $n - 2k + 2$ of them, which contradicts Theorem 1.

Chapter 8

Algorithms for finding stable sets in cycle-plus-triangles graphs

In this chapter, we start by discussing the correspondence between CYCLE-PLUS-TRIANGLES instances and permutations. We then present a randomized algorithm, and a deterministic version, to find maximum stable sets in CYCLE-PLUS-TRIANGLES graphs. We then discuss some intuition behind the algorithms, and prove termination of a version of the randomized algorithm on some families of instances.

8.1 Correspondence between instances and permutations

A *labeled instance* has a natural correspondence to a permutation. Let $[n] = \{1, \dots, n\}$.

Definition 8.1.1. *Given a permutation $\pi : [n] \rightarrow [n]$ (where $n \equiv 0 \pmod{3}$), we define $I(\pi)$ to be a labeled CYCLE-PLUS-TRIANGLES graph or a labeled instance where*

- *Each vertex in V has a unique label in $\{1, \dots, n\}$.*
- $E_H = \{(i, (i \bmod n) + 1) \mid i \in V\}$.
- $E_T = \{(\pi(3i + 1), \pi(3i + 2)) \cup (\pi(3i + 2), \pi(3i + 3)) \cup (\pi(3i + 3), \pi(3i + 1)) \mid i \in \{0, \dots, n/3 - 1\}\}$.

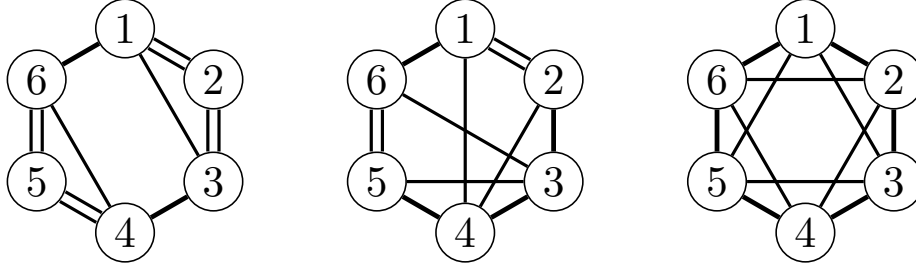


Figure 8.1: The three CYCLE-PLUS-TRIANGLES instances on two triangles

In other words, if $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ is a permutation of $[n]$, then the first three numbers in the list (i.e., $\pi(1), \pi(2), \pi(3)$) correspond to the first triangle, the second triple corresponds to the second triangle, etc. Thus, every permutation on $[n]$ corresponds to a labeled problem instance on n vertices and every (labeled and unlabeled) problem instance corresponds to a set of permutations. However, we note that two nonisomorphic graphs G and G' can map to sets of permutations with different cardinalities. In other words, if we simply generate permutations on $[n]$ uniformly at random, we will generate some instances more than others.

We now discuss this a bit further. We say that two permutations π, ρ on $[n]$ are *label equivalent* if they correspond to the same labeled instances (i.e., instances whose triples are the same sets). For example, if $n = 9$, then consider the permutation $\rho = (1, 3, 2, 5, 4, 6, 7, 8, 9)$ and let π be the identity permutation. Then $I(\pi)$ and $I(\rho)$ are the same labeled graphs. Moreover, if $\rho = (4, 5, 6, 1, 3, 2, 7, 8, 9)$, then $I(\pi)$ and $I(\rho)$ still correspond to the same labeled graphs. Thus, each labeled instance on t triangles corresponds to $6^t \cdot t!$ permutations. If two permutations π and ρ are label equivalent, then $I(\pi)$ and $I(\rho)$ are the same labeled graph.

An instance G of a CYCLE-PLUS-TRIANGLES graph is a unique (up to isomorphism) unlabeled graph. An instance G can correspond to multiple labeled graphs. To quantify how many labeled graphs map to G (and thus, how many permutations map to G), we define $equiv(G)$ to be the set of labeled graphs that map to instance G .

Observation 8.1.1. *For instance G , the number of permutations π on $[n]$ such that $I(\pi) \in equiv(G)$ is $6^t \cdot t! \cdot |equiv(G)|$.*

We will illustrate this by looking at the permutations and instances on two triangles.

There are three (nonisomorphic) instances on three triangles, presented in Figure

8.1. Let us call them G_1 (on triangles (1, 2, 3) and (4, 5, 6)), G_2 (on triangles (1, 2, 4) and (3, 5, 6)) and G_3 (on triangles (1, 3, 5) and (2, 4, 6)), following the left-to-right order of the figure. There are $6! = 720$ permutations of [6]. Let us fix a permutation π of [6]. Any reordering of the first three numbers or of the last three numbers will lead to label-equivalent instances. This is the 6^t factor in the formula of Observation 3, which is 36 in this case. Moreover, any reordering of the two triangles will also lead to label-equivalent instances. This is the factor $t!$, which is 2 in this case. This means that permutation π is label equivalent to 72 other permutations.

Now we wish to compute the size of $equiv(G)$ for each instance G on two triangles. Consider the notation $\{\{a, b, c\}, \{d, e, f\}\}$, which represents a set of label-equivalent permutations or instances. (It does not distinguish the order within each group of three vertices, or between the two groups, so it could also be written $\{\{e, d, f\}, \{c, b, a\}\}$). Then there will be $720/72 = 10$ instances, which are not label equivalent. However, some of these instances are isomorphic to one another, since there are only three nonisomorphic instances on two triangles.

The sets of permutations that generate instances isomorphic to G_1 are:

$$\begin{aligned} & \{\{1, 2, 3\}, \{4, 5, 6\}\}, \\ & \{\{1, 5, 6\}, \{2, 3, 4\}\}, \\ & \{\{1, 2, 6\}, \{3, 4, 5\}\}. \end{aligned}$$

The sets of permutations that generate instance isomorphic to G_2 are:

$$\begin{aligned} & \{\{1, 2, 4\}, \{3, 5, 6\}\}, \\ & \{\{1, 4, 6\}, \{2, 3, 5\}\}, \\ & \{\{1, 2, 5\}, \{3, 4, 6\}\}, \\ & \{\{1, 3, 6\}, \{2, 4, 5\}\}, \\ & \{\{1, 3, 4\}, \{2, 5, 6\}\}, \\ & \{\{1, 4, 5\}, \{2, 3, 6\}\}. \end{aligned}$$

There is only one set of permutations that generates instances isomorphic to G_3 :

$$\{\{1, 3, 5\}, \{2, 4, 6\}\}.$$

This shows the variability of $|equiv(G)|$, the number of label equivalent graphs that correspond to an instance G (i.e., $|equiv(G_1)| = 3$, $|equiv(G_2)| = 6$ and $|equiv(G_3)| = 1$). In general, $|equiv(G)|$ is equal to $2n$ divided by the number of rotational symmetries of G , and then divided by an extra factor 2 if the instance has any axial

symmetry. Here we can see that G_3 has six rotational symmetries and an axial symmetry, whereas G_2 only has an axial symmetry, and G_1 has one rotational and an axial symmetry. This works out since $|equiv(G_1)| = 12/4 = 3$, $|equiv(G_2)| = 12/2 = 6$ and $|equiv(G_3)| = 12/12 = 1$.

We remark that although, technically, as previously mentioned, an instance G of a CYCLE-PLUS-TRIANGLES graph is an unlabeled graph, it is usually necessary when referring to G to have some labeling of the vertices. Thus, throughout this chapter, when we consider an instance $G \in \mathcal{G}_n$, G will always be accompanied by a labeling, which is a function $\ell : V \rightarrow [n]$. When we refer to a vertex $i \in V$, we will often abuse notation and use i to be the label of V . For an edge $e \in E_H$, where $e = (i, (i \bmod n) + 1)$, we refer to i as its *index*. For a triangle $\{i < j < k\}$, we say that j is the *clockwise neighbor* of i , k is the clockwise neighbor of j and i is the clockwise neighbor of k .

Finally, we note that there is another natural way to map permutations to instances. Let the labels of the triangles be fixed (i.e., (1,2,3), (4,5,6), etc.). Then let the Hamilton cycle H be a permutation π on $[n]$. However, we mainly use the previous representation in this thesis.

8.2 Algorithms for finding a maximum stable set

We begin with some terminology necessary for presenting our algorithms. For $G \in \mathcal{G}_n$ and $G = (V, E_H \cup E_T)$, a *token configuration* is an arrangement of tokens on the vertices V , exactly one token per triangle. With respect to a token configuration, we say edge $e \in E_H$ is *monochromatic* if both endpoints of e have tokens. We present an algorithm that begins with an initial token configuration and outputs a token configuration with no monochromatic edges upon termination; such a token configuration corresponds to stable set in G of size $n/3$, which is thus maximum. First, we describe the most general framework.

Token-Sliding-Algorithm Framework

Input: A CYCLE-PLUS-TRIANGLES graph $G = (V, E_T \cup E_H)$.

1. (Initialization) Fix any token configuration.
2. While there is a monochromatic edge in E_H :
 - (a) Choose any monochromatic edge $e \in E_H$ (i.e., e has tokens on both endpoints).
 - (b) Choose an endpoint of e arbitrarily, and slide the token on this vertex to another vertex in its triangle.

It is clear that there is an implementation of this algorithm that does not terminate: If the direction the token slides along the triangle is arbitrary, a token could just continually move from one endpoint of an edge in E_T to the other, and the algorithm would not terminate. So the first choice we make is to require that in Step 2(b), the token is always slid along the triangle in a fixed (e.g., clockwise) direction. Even if we require that the token slides along the triangle in the clockwise direction, then a worst-case choice of endpoint of the monochromatic edge e can also clearly lead to nontermination. Thus, we use randomness or pseudorandomness in an attempt to avoid this worst-case behavior.

Notice that there are four places in which we could use randomness: (i) in the initialization step (i.e., the initial token configuration), (ii) in the choice of monochromatic edge $e \in E_H$, (iii) in the choice of endpoint of e whose token we move, and (iv) the choice of direction (clockwise or counterclockwise) that we slide the token in its respective triangle.

As explained above, we choose not to use randomness in (iv). Moreover, we will not use randomness in (i) or (ii) (since it does not seem useful). In Step (1)–the initialization step—we will use a *minimal token configuration*: For each triangle (i, j, k) with $i < j < k$, we will place the token at i . For (ii), we will always choose the monochromatic edge with the smallest index. This leaves only one point of randomness: (iii) the choice of endpoint of e whose token we move. Lemma 8.2.1 provides some intuition why this is a good place to use randomness. Now we present our main algorithm, which only uses randomness for (iii).¹

¹If we choose the monochromatic edge (i.e., the “flaw”) arbitrarily and “resample” both of its endpoints (by choosing one vertex on each of the two respective triangles uniformly at random), this is perhaps closest to the resampling algorithms for the constructive LLL, but the experimental

Rand-Token-Sliding-Algorithm (Rand-TS-Alg)*Input:* A CYCLE-PLUS-TRIANGLES graph $G = (V, E_T \cup E_H)$.

1. (Initialization) Fix the minimal token configuration.
2. While there is a monochromatic edge in E_H :
 - (a) Choose monochromatic edge $e \in E_H$ with lowest index.
 - (b) Choose endpoint j of e at random.
 - (c) Slide the token on j in the clockwise direction to the next vertex in its triangle.

We conjecture the following.

Conjecture 8.2.1. *For any CYCLE-PLUS-TRIANGLES instance G , **Rand-TS-Alg** terminates in expected polynomial time with respect to the size of G .*

A natural question is whether the number of monochromatic edges can be decreased at each iteration of **Rand-TS-Alg**. The answer to this question is no, since there are token configurations where no move can decrease the number of monochromatic edges (see Section 8.2.1 for more details). Nevertheless, as evidenced in Section 9.2, **Rand-TS-Alg** seems to terminate in a roughly linear number of token-sliding iterations. In fact, for instances of span at least three, this is the case for all instances we have encountered. One of the instances we found that exhibits the worst running time in our experiments has span two and is shown in Figure 7.1b. More discussion can be found in Section 8.4.

The proof of the next lemma is based on the fact that a stable set of size $n/3$ exists, so it does not provide a new proof of existence. Nor does it prove the existence of an efficient algorithm, since the proof shows the algorithm terminates taking (expected) exponential time.

Lemma 8.2.1. ***Rand-TS-Alg** terminates with probability 1.*

Proof. Consider any maximum stable set S of the instance of size $n/3$ (which we know exists by previous results). For a current token configuration occurring in the algorithm, define the distance between it and S to be the minimum number of clockwise token sliding moves necessary to get from the current token configuration

performance of this algorithm is worst than that of **Rand-TS-Alg**.

to the token configuration corresponding to S . This distance can be at most $2n/3$. Then, notice that while the algorithm is still running, for every monochromatic edge there is one token from the edge that is not on the same vertex as in the stable set, since the stable set S has no monochromatic edge. Hence, there is at least a $1/2$ probability that this vertex is moved, decreasing the distance between the next configuration and the stable set S by 1. Therefore, from any configuration, there is at least a probability of $1/4^n$ that it terminates within $2n$ steps. (It does not necessarily terminate by finding the stable set S , but if there is no more monochromatic edges, it must have found some stable set.) Since we have shown that the algorithm has a fixed probability of terminating within a fixed number of steps regardless of the current configuration, it results that repeating this number of steps i times will give a probability $1 - (1 - 1/4^n)^i$ of terminating, which goes to 1 as i goes to infinity. \square

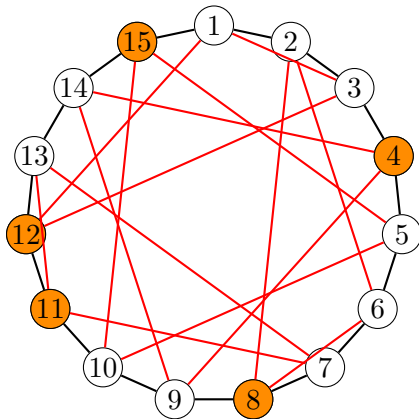
We discuss the intuition behind why we believe the algorithm terminates efficiently in Section 8.3.

8.2.1 Locally minimal configurations

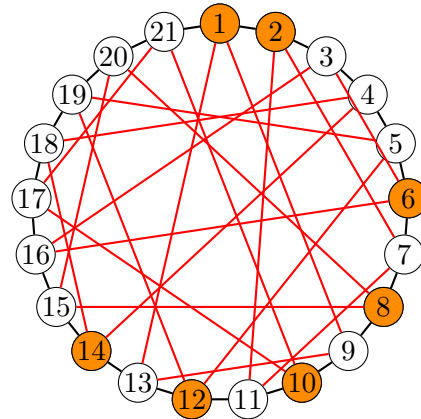
Since all of the algorithms in our framework are based on local moves, it is a natural question to ask whether there exist token configurations where no local move can decrease the number of monochromatic edges (i.e., it is *locally minimal*). By *local move*, we refer to any move that takes a token from a monochromatic edge and places it onto a different vertex of the same triangle. The answer to this question is negative, as illustrated in Figure 8.2. Furthermore, it is possible to construct instances where up to a logarithmic number of local moves that increase the number of monochromatic edges must be made before any can decrease the number of monochromatic edges. Such a construction can be made by making all four unoccupied (containing no token) vertices of the two triangles of the monochromatic edge double-blocked, then making all of their free vertices double-blocked, etc.

8.2.2 Deterministic token-sliding algorithm

There is a deterministic version of **Rand-TS-Alg** that works well in practice. It is inspired by the rotor-router model [PDDK96]. We simply replace the random choice in Step 2(b) of **Rand-TS-Alg** with a pseudorandom choice, which we keep track of with a *rotor-router* or *pointer*. As discussed in Section 9.2.1, we have found instances for which this algorithm does not terminate. However, they seem to be very rare



(a) Token configuration where no local move decreases the number of monochromatic edges.



(b) Token configuration where every local move increases the number of monochromatic edges.

Figure 8.2: CYCLE-PLUS-TRIANGLES graphs with locally minimal token configurations represented in orange.

and difficult to find. Thus, the deterministic algorithm seems to be a good tool for efficiently finding a maximum stable set of a CYCLE-PLUS-TRIANGLES graph in practice.

Rotor-Router-Token-Sliding-Algorithm (RR-TS-Alg)

Input: A CYCLE-PLUS-TRIANGLES graph $G = (V, E_T \cup E_H)$.

1. (Initialization)
 - (a) Fix the minimal token configuration.
 - (b) For each edge $(a, b) = (i, (i \bmod n) + 1)$ in E_H , initialize a pointer from a to b .
2. While there is a monochromatic edge in E_H :
 - (a) Choose monochromatic edge $e \in E_H$ with lowest index.
 - (b) Choose endpoint j as indicated by the pointer.
 - (c) Slide the token on j in the clockwise direction to the next vertex in its triangle.
 - (d) Change the direction of the pointer on edge e .

8.3 Intuition for efficient termination

Let $G = (V, E_T \cup E_H)$ be a CYCLE-PLUS-TRIANGLES instance and let us fix a token configuration (defined in Section 8.2).

Definition 8.3.1. *Define a free vertex to be a vertex that does not contain a token, and is not adjacent (on the Hamilton cycle) to a vertex with a token. Let n_{free} denote the number of free vertices.*

A single-blocked vertex is a vertex with no token, adjacent (on the Hamilton cycle) to exactly one vertex with a token. Let $n_{\text{single-blocked}}$ denote the number of single-blocked vertices.

A double-blocked vertex is a vertex with no token, adjacent (on the Hamilton cycle) to exactly two vertices with a token. Let $n_{\text{double-blocked}}$ denote the number of double-blocked vertices.

Recall that in a token configuration, each triangle has exactly one token. We use n_{mono} to denote the number of monochromatic edges in E_H (with respect to the fixed

token configuration).

Observation 8.3.1.

$$n_{\text{free}} = 2n_{\text{mono}} + n_{\text{double-blocked}}.$$

Proof. Let $S \subset V$ be the subset of vertices that have a token in the fixed token configuration. The number of monochromatic edges in E_H is equal to the number of edges in $G[S]$ (the subgraph of G induced by S). The number of edges in E_H crossing the cut $(S, V \setminus S)$ is equal to $n_{\text{single-blocked}} + 2n_{\text{double-blocked}}$.

By definition, a vertex is free if it does not have a token and it is not single- or double-blocked. Therefore,

$$n_{\text{free}} = 3t - |S| - n_{\text{single-blocked}} - n_{\text{double-blocked}} = 2t - n_{\text{single-blocked}} - n_{\text{double-blocked}}.$$

The number of edges in E_H with at least one endpoint in S equals the number of edges in $G[S]$ plus the number of edges crossing the cut $(S, V \setminus S)$. Moreover, the number of edges in E_H with at least one endpoint in S is also equal to twice the number of vertices in S minus the number of edges in $G[S]$, which is $2t - n_{\text{mono}}$. Therefore,

$$\begin{aligned} 2t - n_{\text{mono}} &= n_{\text{single-blocked}} + 2n_{\text{double-blocked}} + n_{\text{mono}} \\ 2n_{\text{mono}} &= 2t - n_{\text{single-blocked}} - 2n_{\text{double-blocked}} \\ n_{\text{free}} &= 2n_{\text{mono}} + n_{\text{double-blocked}}, \end{aligned}$$

which proves the formula. □

The formula in Observation 8.3.1 implies that if the new position of a token is distributed uniformly at random, then the algorithm will “make progress”. Specifically, notice that moving a token to a free vertex decreases the number of monochromatic edges by at least 1, while moving it to a double-blocked vertex increases it by at most 1. Since there are $2n_{\text{mono}}$ more free vertices than double-blocked, moving a token (from an endpoint of a monochromatic edge) to a position chosen uniformly at random will lead to an expected decrease of size at least $\frac{2n_{\text{mono}}}{n} \geq \frac{2}{n}$ in the number of monochromatic edges. Thus, if every token move were uniformly distributed, we would expect the algorithm to terminate in a quadratic number of iterations. Of course, as we execute the algorithm, the new positions for the tokens are not guaranteed to be distributed uniformly at random. Still, this is one possible explanation for the fast runtime of our algorithm that we observe empirically.

8.4 Efficient termination on specific instances

In this section, our goal is to prove that our algorithm terminates in polynomial time on specific families of instances. More specifically, we will do this for two families of instances where each triangle is adjacent to only two other triangles: chain-of-twins and chain-of-links.

We will focus on proving termination of a slightly modified version of **Rand-TS-Alg** with two points of randomness. We use all types of randomness described in Section 8.2 except (i) and (ii). Specifically, we choose any (i.e., can be arbitrary) initial configuration. Then we choose both the endpoint of the monochromatic edge and the destination for the token (clockwise or counter-clockwise) at random.

Alt-Rand-Token-Sliding-Algorithm (Alternate-Rand-TS-Alg)

Input: A CYCLE-PLUS-TRIANGLES graph $G = (V, E_T \cup E_H)$.

1. (Initialization) Fix an arbitrary token configuration.
2. While there is a monochromatic edge in E_H :
 - (a) Choose monochromatic edge $e \in E_H$ with lowest index.
 - (b) Choose endpoint j of e at random.
 - (c) Move the token on j to another vertex of the same triangle at random.

Both proofs of termination can be adapted to prove that **Rand-TS-Alg** terminates in polynomial time on the two classes of instances we consider, but the proof in the case of chain-of-twins is more complicated, so we choose to present the proofs for termination of **Alternate-Rand-TS-Alg**.

8.4.1 Chain-of-twins

Two triangles (i, j, k) and $(i + 1, j + 1, k + 1)$ (indices computed modulo n) are *twins*. Formally, the triangles in a chain-of-twins instance on t triangles are given by the the triples $\{(i, i + t, i + 2t) \mid i \in [t]\}$. (See Figure 9.1a for the chain-of-twins instance when $t = 4$.)

Lemma 8.4.1. *For a chain-of-twin instances on $t \geq 3$ triangles, Alternate-Rand-*

TS-Alg terminates with probability at least $1/2$ after $4t$ iterations.

Proof. We will prove this lemma by showing that at each step of the algorithm, the expected number of monochromatic edges goes down.

More formally, let S denote a token configuration and let $\rho(S)$ denote the number of monochromatic edges in S . Suppose that, for token configuration S , there is a monochromatic edge $e = uv$, where u belongs to triangle t_a and v belongs to the (adjacent) triangle t_b . Let the t_c denote the other triangle adjacent to t_a . Besides u , let w and x denote the other two vertices in triangle t_a . Notice that at most one of w and x can be adjacent to another vertex in the Hamilton cycle with a token, which would be a vertex occupied by the token from t_c . Without loss of generality, let us assume that two vertices adjacent to x on the Hamilton cycle do not contain tokens.

Without loss of generality, suppose we move u to an adjacent vertex in triangle t_a and let S' denote the resulting token configuration. With probability $1/2$, we will move the token from u to x and with probability $1/2$, we will move the token from u to w . In the former case, we have $\rho(S) - \rho(S') = 1$ and in the latter case, we have $\rho(S) - \rho(S') \geq 0$. Thus, $\mathbb{E}[\rho(S) - \rho(S')] \geq 1/2$.

Let X be a random variable denoting the number of token moves before the algorithm terminates. Then $\mathbb{E}[X] \leq 2t$, since $\rho(S) \leq t - 1$. By Markov's inequality, we see that $\Pr[X \geq 4t] \leq 2t/(4t) = 1/2$, which proves the lemma. \square

8.4.2 Chain-of-links

A *chain-of-links* instance on t triangles is a graph where the triangles are given by the triples $\{(3i - 2, 3i, 3i + 2 \mid i \in [t])\}$ (with indices computed modulo $3t$). An example of such a graph can be seen in Figure 7.1

In order to prove efficient termination on this instance, we will state a random process on a circular trinary string that is equivalent to **Alternate-Rand-TS-Alg**. Let $n \in \mathbb{Z}_{\geq 2}$ and let $S = (a_1, \dots, a_n) \in \{0, 1, 2\}^n$ be a circular (i.e., $a_{n+1} = a_1$) trinary sequence of length n . We say that (a_i, a_{i+1}) is a *decreasing pair* if $a_i > a_{i+1}$.

Definition 8.4.1. We define the following random process:

1. While there exists a decreasing pair, do the following:
 - (a) Find the decreasing pair (a_k, a_{k+1}) with minimum index k .

(b) Choose a variable of the pair with uniform probability:

$$x = a_k \text{ OR } x = a_{k+1}.$$

(c) Randomly change the value of the chosen variable to another value with equal probability:

$$x \leftarrow x + 1 \pmod{3} \text{ OR } x \leftarrow x + 2 \pmod{3}.$$

2. Return the final sequence S .

This trinary string can be thought of as representing the position of the token on each triangle. In other words, if the i^{th} value of the string is a 0, then the token of the i^{th} triangle is on the vertex with index $3i - 2$. If the value is a 1, the token is on vertex $3i$, and if the value is 2, then the token is on vertex $3i + 2$. Thus, a trinary string of length t corresponds to a token configuration for a chain-of-links instance on t triangles. It is easy to see that monochromatic edges in the token configuration correspond to the decreasing pairs in the trinary string. The random process corresponds exactly to the **Alternate-Rand-TS-Alg** on a chain-of-links instance and it terminates if and only if **Alternate-Rand-TS-Alg** terminates on this instance. Let us now define a potential function to analyze the random process from Definition 8.4.1.

Definition 8.4.2. For $n \in \mathbb{Z}_{\geq 2}$ and $S = (a_1, a_1, \dots, a_n) \in \{0, 1, 2\}^n$, define

$$n_{0,1} := |\{(a_i, a_{i+1}) \mid i \in [n], a_i = 0, a_{i+1} = 1\}|,$$

$$n_{1,1} := |\{(a_i, a_{i+1}) \mid i \in [n], a_i = 1, a_{i+1} = 1\}|,$$

$$n_{1,2} := |\{(a_i, a_{i+1}) \mid i \in [n], a_i = 1, a_{i+1} = 2\}|.$$

Define the potential function ρ as $\rho(S) := n_{0,1} + n_{1,2} + 2n_{1,1}$.

Lemma 8.4.2. Let S be a circular trinary string and let S' be the string obtained after one iteration of the random process on S . Then, $\mathbb{E}[\rho(S')] \geq \rho(S)$.

Proof. To prove this lemma, we do a case analysis in which we examine each possible decreasing pair along with the variable before the pair and the variable after the pair. A single iteration of the algorithm can only change the value of one variable of a pair. Thus, the change in potential function on the four aforementioned variables will reflect the change in potential function on the entire token configuration.

1. Decreasing pair: $(2, 0)$.

(a)

$$(0, 2, 0, 0) \mapsto \begin{cases} (0, 0, 0, 0) \text{ then } \rho(S') = \rho(S) \\ (0, 2, 1, 0) \text{ then } \rho(S') = \rho(S) \\ (0, 1, 0, 0) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(b)

$$(0, 2, 0, 1) \mapsto \begin{cases} (0, 0, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (0, 2, 1, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 1, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (0, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(c)

$$(0, 2, 0, 2) \mapsto \begin{cases} (0, 0, 0, 2) \text{ then } \rho(S') = \rho(S) \\ (0, 2, 1, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 1, 0, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 2, 2, 2) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(d)

$$(1, 2, 0, 0) \mapsto \begin{cases} (1, 0, 0, 0) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 2, 1, 0) \text{ then } \rho(S') = \rho(S) \\ (1, 1, 0, 0) \text{ then } \rho(S') = \rho(S) + 1 \\ (1, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(e)

$$(1, 2, 0, 1) \mapsto \begin{cases} (1, 0, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 2, 1, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (1, 1, 0, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (1, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(f)

$$(1, 2, 0, 2) \mapsto \begin{cases} (1, 0, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 2, 1, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (1, 1, 0, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (1, 2, 2, 2) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(g)

$$(2, 2, 0, 0) \mapsto \begin{cases} (2, 0, 0, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 1, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 0, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(h)

$$(2, 2, 0, 1) \mapsto \begin{cases} (2, 0, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 1, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (2, 1, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(i)

$$(2, 2, 0, 2) \mapsto \begin{cases} (2, 0, 0, 2) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 1, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (2, 1, 0, 2) \text{ then } \rho(S') = \rho(S) \\ (2, 2, 2, 2) \text{ then } \rho(S') = \rho(S) \end{cases}$$

2. Decreasing pair: (2, 1).

(a)

$$(0, 2, 1, 0) \mapsto \begin{cases} (0, 0, 1, 0) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \\ (0, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 3 \\ (0, 2, 0, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(b)

$$(0, 2, 1, 1) \mapsto \begin{cases} (0, 0, 1, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 2 \\ (0, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 3 \\ (0, 2, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(c)

$$(0, 2, 1, 2) \mapsto \begin{cases} (0, 0, 1, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 2, 2, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 3 \\ (0, 2, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(d)

$$(1, 2, 1, 0) \mapsto \begin{cases} (1, 0, 1, 0) \text{ then } \rho(S') = \rho(S) \\ (1, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \\ (1, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 3 \\ (1, 2, 0, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(e)

$$(1, 2, 1, 1) \mapsto \begin{cases} (1, 0, 1, 1) \text{ then } \rho(S') = \rho(S) \\ (1, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 2 \\ (1, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 3 \\ (1, 2, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(f)

$$(1, 2, 1, 2) \mapsto \begin{cases} (1, 0, 1, 2) \text{ then } \rho(S') = \rho(S) \\ (1, 2, 2, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 3 \\ (1, 2, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(g)

$$(2, 2, 1, 0) \mapsto \begin{cases} (2, 0, 1, 0) \text{ then } \rho(S') = \rho(S) + 1 \\ (2, 2, 2, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 2 \\ (2, 2, 0, 0) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(h)

$$(2, 2, 1, 1) \mapsto \begin{cases} (2, 0, 1, 1) \text{ then } \rho(S') = \rho(S) + 1 \\ (2, 2, 2, 1) \text{ then } \rho(S') = \rho(S) - 2 \\ (2, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 2 \\ (2, 2, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

(i)

$$(2, 2, 1, 2) \mapsto \begin{cases} (2, 0, 1, 2) \text{ then } \rho(S') = \rho(S) + 1 \\ (2, 2, 2, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (2, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 2 \\ (2, 2, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \end{cases}$$

3. Decreasing pair: $(1, 0)$.

(a)

$$(0, 1, 0, 0) \mapsto \begin{cases} (0, 2, 0, 0) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 1 \\ (0, 0, 0, 0) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 2, 0) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

(b)

$$(0, 1, 0, 1) \mapsto \begin{cases} (0, 2, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 2 \\ (0, 0, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 2, 1) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(c)

$$(0, 1, 0, 2) \mapsto \begin{cases} (0, 2, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 3 \\ (0, 0, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (0, 1, 2, 2) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

(d)

$$(1, 1, 0, 0) \mapsto \begin{cases} (1, 2, 0, 0) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 2 \\ (1, 0, 0, 0) \text{ then } \rho(S') = \rho(S) - 2 \\ (1, 1, 2, 0) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

(e)

$$(1, 1, 0, 1) \mapsto \begin{cases} (1, 2, 0, 1) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 3 \\ (1, 0, 0, 1) \text{ then } \rho(S') = \rho(S) - 2 \\ (1, 1, 2, 1) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(f)

$$(1, 1, 0, 2) \mapsto \begin{cases} (1, 2, 0, 2) \text{ then } \rho(S') = \rho(S) - 1 \\ (1, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 3 \\ (1, 0, 0, 2) \text{ then } \rho(S') = \rho(S) - 2 \\ (1, 1, 2, 2) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

(g)

$$(2, 1, 0, 0) \mapsto \begin{cases} (2, 2, 0, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 1, 0) \text{ then } \rho(S') = \rho(S) + 2 \\ (2, 0, 0, 0) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 2, 0) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

(h)

$$(2, 1, 0, 1) \mapsto \begin{cases} (2, 2, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 1, 1) \text{ then } \rho(S') = \rho(S) + 3 \\ (2, 0, 0, 1) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 2, 1) \text{ then } \rho(S') = \rho(S) \end{cases}$$

(i)

$$(2, 1, 0, 2) \mapsto \begin{cases} (2, 2, 0, 2) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 1, 2) \text{ then } \rho(S') = \rho(S) + 3 \\ (2, 0, 0, 2) \text{ then } \rho(S') = \rho(S) \\ (2, 1, 2, 2) \text{ then } \rho(S') = \rho(S) + 1 \end{cases}$$

We can see that in each case, since every one of the four outcomes is equally likely, ρ will increase or stay the same on average. \square

Let S_i be the string obtained after i iterations of the random process starting on the circular trinary string S_0 . We say that the i^{th} step is *silent* if $\rho(S_{i-1}) = \rho(S_i)$. Otherwise, the step is *active*. If the i^{th} step considers the decreasing pair (b, c) , then we say that S_{i-1} *focuses on configuration* (a, b, c, d) , where a and d are the neighbors of b and c . Notice that if S_{i-1} focuses on configuration (a, b, c, d) , then the only changes in S_i with respect to S_{i-1} will take place among the variables (b, c) and the next decreasing pair will either involve variables (a, b) , (b, c) or (c, d) .

Lemma 8.4.3. *If the random process runs for k steps, then with probability at least $1/2$, we have at least $k/16$ active steps.*

Proof. Notice that a step has probability at least $1/4$ of being active, unless the string S_i focuses on the configuration $(2, 2, 0, 0)$ (where the decreasing pair is $(2, 0)$ and the neighboring variables have values 2 and 0). In this case, there is at least a $1/2$ probability that the next step leads to a string S_{i+1} which does not focus on the configuration $(2, 2, 0, 0)$. Thus, with probability at least $1/4$, the next step is active. So in expectation, after k steps of the random process, at least $k/8$ of them are active. By Markov's inequality, we conclude that with probability at least $1/2$, the set of active steps is at least $k/16$. \square

Lemma 8.4.4. *Let S be a circular trinary string on n variables. Then the random process applied to S will terminate in expected $O(n^2)$ number of iterations.*

Proof. We will follow Section 21.3 of [Sin22], which proves a more general lemma. Let us consider only the active steps, and let A_i denote the string after the i^{th} active step. Let $A_0 = S$. Define $X_i = \rho(A_i)$ and $D_i = X_i - X_{i-1}$. Notice that X_i is a *submartingale*, since by Lemma 8.4.2, $\mathbb{E}(X_i) \geq \mathbb{E}(X_{i-1})$. Thus, we have

$$\mathbb{E}(D_i \mid X_1, \dots, X_{i-1}) \geq 0.$$

Furthermore, the value of D_i^2 is bounded below.

$$\mathbb{E}(D_i^2 \mid X_1, \dots, X_{i-1}) \geq 1.$$

Then $Y_i = X_i^2 - i$ is also a submartingale with respect to X_1, \dots, X_{i-1} , because

$$\mathbb{E}(Y_i \mid X_1, \dots, X_{i-1}) \geq Y_{i-1}.$$

Define T to be the time step when the process terminates. Since the process only terminates when $X_i = 2n$, it follows that $\mathbb{E}(X_T) = 2n$, and $\mathbb{E}(X_T^2) = 4n^2$. Then we can apply the Optional Stopping Theorem [GS20a] to Y_T in order to bound the expected length of the process $\mathbb{E}(T)$.

$$\begin{aligned}\mathbb{E}(Y_T) &\geq \mathbb{E}(Y_0) \\ \mathbb{E}(X_T^2) - \mathbb{E}(T) &\geq 0 \\ 4n^2 - \mathbb{E}(T) &\geq 0 \\ \mathbb{E}(T) &\leq 4n^2.\end{aligned}$$

It then follows immediately from Lemma 8.4.3, that the expected number of steps for the algorithm to terminate is at most $64n^2$, which concludes the poof. \square

To prove termination of **Rand-TS-Alg** on this family of instances (no matter the initial configuration), one can take as a potential function the sum of the number of 11 pairs and the number of 01 pairs.

Chapter 9

Empirical evaluation of the algorithms

In this chapter, our goal is to present empirical evidence that our algorithm terminates in an expected linear number of iterations. To do so, we start by discussing the structure of `CYCLE-PLUS-TRIANGLES` instances and describing our generation methods. Then, we present experimental results for the different algorithms and generation methods, before discussing some instances we found where the deterministic algorithm never terminates.

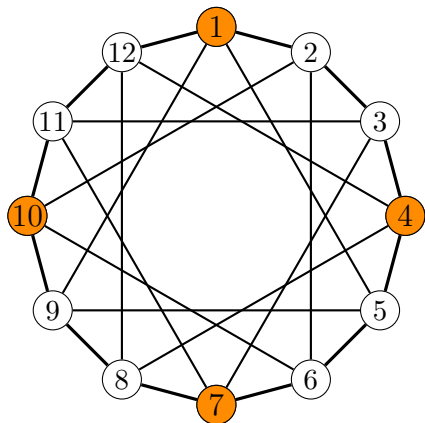
9.1 Structure and generation of problem instances

In order to experimentally evaluate **Rand-TS-Alg**, we attempt to generate a comprehensive set of problem instances. The most natural set of instances to consider are those generated uniformly at random. We show how to generate instances according to the uniform distribution and we use this as our first generation method. Beyond uniform generation, we want to consider different parameters of `CYCLE-PLUS-TRIANGLES` graphs in order to generate a diverse set of instances. One parameter, defined in Section 7.2, is the span.

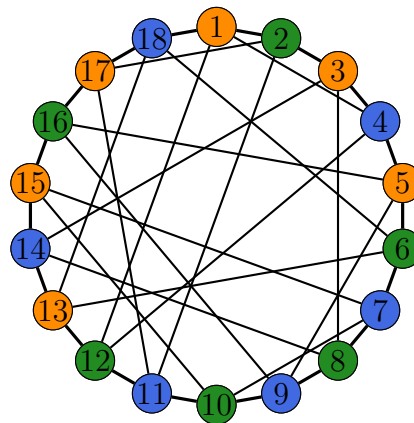
The maximum possible span of an instance on t triangles is t , which is attained uniquely by a set of t “equilateral” triangles, which we call a *chain-of-twins*.¹ (See

¹Two triangles (i, j, k) and $(i + 1, j + 1, k + 1)$ (indices computed modulo n) are *twins*. Formally, the triangles in a chain-of-twins instance on t triangles are given by the the triples $\{(i, i + t, i + 2t) |$

Figure 9.1a for $t = 4$.) By Lemma 7.4.2, it is sufficient to consider instances with span at least three. We show that for sufficiently large n , the number of instances of span at least k decays exponentially as k grows. Hence, when sampling instances uniformly at random, instances of high span occur very rarely. This motivates other methods to efficiently generate instances of at least a given span at the expense of uniformity. We are not aware of any efficient method to uniformly at random generate instances with a given span.



(a) Chain-of-twins on four triangles.



(b) CYCLE-PLUS-TRIANGLES graph with six triangles.

Figure 9.1: Two examples of CYCLE-PLUS-TRIANGLES graphs. In the graphs, one and three stable sets of size t are shown, respectively, where t is the number of triangles.

9.1.1 Generating instances uniformly at random

Our method to generate a graph $G \in \mathcal{G}_n$ with $\sigma(G) \geq 3$ uniformly at random begins with generating a permutation on $[n]$ uniformly at random. As discussed in Section 7.2, each labeled graph corresponds to $6^t \cdot t!$ permutations, where $t = n/3$. Moreover, each instance $G \in \mathcal{G}_n$ corresponds to $|\text{equiv}(G)|$ labeled graphs. In order to generate an instance $G \in \mathcal{G}_n$ uniformly at random, we generate a permutation π uniformly at random, let $G = I(\pi)$, check if $\sigma(G) \geq 3$ and, if so, compute $\text{equiv}(G)$ and output G with probability $1/|\text{equiv}(G)|$.

$i \in [t]$.

Thus, we need to compute the cardinality of $\text{equiv}(G)$, which means determining how many labeled graphs correspond to the same instance G . To do this, we need to understand the automorphism structure of G ; if two labeled graphs (that are not label equivalent) correspond to the same instance G , then the labelings yield an automorphism of G . We show that the possible automorphisms of G are highly restricted.

Lemma 9.1.1. *Let $G = (V, E_T \cup E_H)$ and $G' = (V', E'_T \cup E'_H)$ be two CYCLE-PLUS-TRIANGLES graphs such that $\sigma(G), \sigma(G') \geq 3$. If G and G' are isomorphic with mapping $f : V \rightarrow V'$, then f maps edges in E_T to E'_T and edges in E_H to E'_H .*

Proof. For the sake of contradiction, suppose that f maps an edge in E_H to an edge not in E'_H . Then there is at least one triangle $T \subset E_T$ of G , say on the vertex set $\{u, v, w\}$, such that an edge in $E(T)$ is mapped to an edge in E'_H . Since $\sigma(G) \geq 3$, then G has at least three triangles. We distinguish three cases.

Case 1 $|E'_H \cap E(T)| = 1$. Suppose that $E'_H \cap E(T) = uv$. Then the two other edges in $E(T)$, uw and vw , belong to the same triangle, say T' , of G' . But then $uw \in T'$ so there must be two parallel edges between u and w , one in T' and one in E'_H . Thus, G' must have span one, which is a contradiction.

Case 2 $|E'_H \cap E(T)| = 2$. Suppose that $E'_H \cap E(T) = \{uv, vw\}$. Then the edge uw belongs to some triangle T' of G' . Therefore u and w have common neighbor z in T' , which is also a common neighbor in the Hamilton cycle H . This implies that G has span two, a contradiction.

Case 3 $|E'_H \cap E(T)| = 3$. Then $E'_H = E(T)$, which is a contradiction, since $|E'_H| \geq 9$.

□

By Lemma 9.1.1, when $\sigma(G) \geq 3$, any automorphism of $G = (V, E_H \cup E_T)$ can only be obtained from a mapping of V to V that maps edges in E_H to edges in E_H . Such a mapping corresponds to either i) a labeling of V along the Hamilton cycle in the clockwise order (i.e., a *rotation*) or ii) a labeling of V along the Hamilton cycle in the counterclockwise order (i.e., a *reflection-rotation*).

The exact size of the equivalence class of G can then be computed in polynomial time, by generating the $2n$ labeled graphs obtained from a sequence of rotations and reflection-rotations, and counting the number of label-equivalent sets (i.e., each set contains a maximal subset of the $2n$ permutations that are label equivalent). Notice

that in order to check if two labeled instances are label equivalent, we just need to check if their adjacency matrices are exactly the same (without any relabeling of either graph). The pseudocode for the method to generate a CYCLE-PLUS-TRIANGLES instance G uniformly at random from among all instances on t triangles and span at least k is shown below. Notice that as k increases, Generation 1 takes exponentially more time to output a graph, as shown in Lemma 9.1.3. We have not found an efficient method to generate CYCLE-PLUS-TRIANGLES instances of a given span uniformly at random. This motivates the generation methods presented in the next section.

Computation of $equiv(G)$

Input: A CYCLE-PLUS-TRIANGLES graph G with labeling ℓ and $\sigma(G) \geq 3$.

Output: The set of labeled instances in $equiv(G)$.

1. $equiv(G) = \{\}$.
2. For $s = 0$ to 1 do:
 - (a) For $i = 0$ to n do:
 - i. For v in G do:

$$\ell'(v) = s \cdot ((\ell(v) + i) \bmod n) + (1 - s) \cdot ((-\ell(v) - i) \bmod n).$$
 - ii. If $(G, \ell') \notin equiv(G)$ do:

$$equiv(G) = equiv(G) \cup (G, \ell').$$
3. Return $equiv(G)$.

Generation 1

Input: A number of triangles t and minimum span $k \geq 3$.

Output: A CYCLE-PLUS-TRIANGLES graph G with $\sigma(G) \geq k$.

1. Generate a permutation π on $[3t]$ uniformly at random and let $G = I(\pi)$.
2. If $\sigma(G) < k$, then restart from 1.
3. Compute $|equiv(G)|$.
4. Output G with probability $\frac{1}{|equiv(G)|}$; otherwise restart from 1.

We remark that since the size of $\text{equiv}(G)$ varies by a factor of at most $2n$, the probability of a given instance with span greater or equal to 3 of being generated by Generation 1 is at least $1/2n$ times the probability of it being generated from a random permutation.

9.1.2 Generating instances with high span

We first observe that CYCLE-PLUS-TRIANGLES instances of high span are exponentially rare. Thus, the rejection sampling approach used in Generation 1 is inefficient for generating instances of large span. Table 9.1, shows the number of nonisomorphic CYCLE-PLUS-TRIANGLES graphs on t triangles and span exactly k for $1 \leq k \leq t \leq 4$. (These numbers were obtained using a simple backtracking algorithm followed by isomorphism testing.) If we generate an instance via a random permutation, then for sufficiently large n , there is a large probability ($> .99$) of generating an instance with span 1. The proof of the next lemma follows from a formula derived by Kaplansky in [Kap44] for a problem introduced by Netto.

Lemma 9.1.2. *Let π be a permutation of $[n]$ chosen uniformly at random. Then*

$$\lim_{n \rightarrow \infty} \Pr[\sigma(I(\pi)) = 1] \geq 1 - e^{-6} .$$

Proof. This proof follows from a formula derived by Kaplansky in Section 6 of [Kap44] for a problem introduced by Netto. The setting is that we split the set $[n]$ into n/a subsets of size a each (in our case we have $a = 3$). The problem of Netto asks for the probability that a random permutation of $[n]$ has no two consecutive elements from a same subset. We consider the inverse permutation from the one we use to build our instance, that is, the triangles are numbered $\{1, 2, 3\}, \{4, 5, 6\}, \dots, \{n-2, n-1, n\}$ and the permutation gives the Hamilton cycle. Whenever two consecutive elements in the permutation are from the same subset, i.e., the same triangle, the corresponding instance has span one. Observe that for a permutation π of $[n]$, we have that $\pi(n)$ and $\pi(1)$ are consecutive on the Hamilton cycle, so if π maps 1 and n to the same triangle then the corresponding instance has span span. However, in the problem of Netto, the numbers n and 1 are not considered to be consecutive, so the asymptotic bound obtained for this problem is a lower bound on the probability that a CYCLE-PLUS-TRIANGLES instance generated from a random permutation has span one. Finally, to obtain the claimed bound it remains to compute the value of $[F(3, 2)]^{n/3} \phi_0$, where $[F(a, a-1)]^{n/a} \phi_0$ is defined in Eq. (5) of [Kap44]. \square

More generally, if we generate an instance via a random permutation, then for sufficiently large n , the probability of generating an instance with span at least k is exponentially small in k , as shown in the next lemma.

Notice that the span distribution of instances generated from Generation 1 would be similar to that of instances generated by random permutations. Indeed, as we stated earlier, the probability for generating a given instance via Generation 1 with parameter k and via a random permutation with span at least k differ by at most a factor of $2n$.

Lemma 9.1.3. *Let π be a permutation of $[n]$ chosen uniformly at random. Then*

$$\lim_{n \rightarrow \infty} \Pr[\sigma(I(\pi)) \leq k] \geq 1 - e^{-k/4} .$$

Proof. In order to lower bound the probability of an instance of having span at most k , we will consider the probability that a given triangle has span at most k . This probability can be lower bounded by the probability of the last vertex of the triangle being at distance k or less on the Hamilton cycle to either of the other two vertices, assuming that they are at distance at least $k + 1$ from each other (or the triangle would already be span k). For a given triangle T we have $\Pr[\sigma(T) \leq k] \geq 3k/n$, since there are at least $3k$ positions for the last vertex (from the n total) that are at distance k or less from the two first vertices. The obstacle now is that the probability of each triangle having span at least k is not independent. Therefore, consider the following lower bound: For the first triangle, we use the bound we just obtained. Then, we remove the three vertices in that triangle and consider the resulting permutation. There are now 3 places in the permutation already occupied, whose position is random compared to the positions of the vertices of the next triangle we consider. Therefore, using the same lower bound we get for the second triangle $\Pr[\sigma(T) \leq k] \geq (1 - \frac{3}{n})\frac{3k}{n-3}$. We repeat this reasoning and after having fixed a triangles, there will be $3a$ fixed positions in the permutation so

n	span	1	2	3	4
	1	1			
2	2	2	1		
3	3	21	3	1	
4	4	629	76	6	1

Table 9.1: Number of small nonisomorphic CYCLE-PLUS-TRIANGLES instances by span.

$\Pr[\sigma(T) \leq k] \geq (1 - \frac{3a}{n}) \frac{3k}{n-3a}$. We obtain the following formula.

$$\begin{aligned}
\Pr[\sigma(G) \leq k] &\geq 1 - \prod_{a=0}^{n/3-1} \left(1 - \left(1 - \frac{3a}{n} \right) \frac{3k}{n-3a} \right) \\
&\geq 1 - \prod_{a=0}^{n/6} \left(1 - \left(1 - \frac{3a}{n} \right) \frac{3k}{n} \right) \\
&\geq 1 - \prod_{a=0}^{n/6} \left(1 - \left(1 - \frac{1}{2} \right) \frac{3k}{n} \right) \\
&\geq 1 - \left(1 - \frac{3k}{2n} \right)^{n/6}.
\end{aligned}$$

Finally, an asymptotic evaluation of this probability gives $\lim_{n \rightarrow \infty} \Pr[\sigma(I(\pi)) \leq k] \geq 1 - e^{-k/4}$. \square

We propose three efficient methods for generating instances with at least a given span. These methods are efficient in that they efficiently produce some instance with the specified span. However, none of the methods generate instances uniformly at random from the set of instances with a given span. Moreover, for two of the methods (Generation 3 and Generation 4), there are instances they will not output. The pseudocode for these methods is presented at the end of the section.

The method Generation 2 is based on increasing the success probability of the rejection sampling approach employed by Generation 1. This results in generating larger instances of given span within a reasonable amount of time. This method works as follows: it adds triangles to a cycle of length n , one at a time, such that in each step the new triangle has span at least k . If there is no such triangle, the

algorithm restarts. After successfully adding $n/3$ triangles, the resulting instance is returned and it is guaranteed to have span at least k . In order to add a triangle, we choose at random a vertex u among those that are not yet in a triangle, followed by a vertex v of distance at least k to u on the cycle and finally a vertex w of distance at least k to u and v on the cycle.

The next method, Generation 3, begins with an instance that has the desired span (for example, the chain of twins on k triangles). It then inserts triangles with span at least k into the instance. This can only ever increase the span of other triangles, so the resulting instance is guaranteed to have span at least k , and therefore never leads to rejection. However, not all instances of span k can be generated in such manner; if they do not have the starting graph as an induced subgraph, they will never be generated.

Finally, in Generation 4 we start with a chain of twins on t triangles, and then do random local swaps (adjacent vertices on the Hamilton cycle swap their triangle membership) a given number of times. In order to preserve span at least k , we only apply a swap if it does not decrease the span below k . This also has the drawback of not generating all possible instances of span k , but gives a different distribution on the high span instances than the previous methods.

Generation 2

Input: Number t of triangles and minimum span $k \geq 1$

1. (Initialization) Let $R = [3t]$ and $T := \emptyset$.
2. For $i = 1$ to t do:
 - (a) Pick $a \in R$ uniformly at random and let $S = \{v \in [3t] \mid \text{dist}_H(a, v) \geq k\}$.
 - (b) If $R \cap S \neq \emptyset$: pick $b \in R \cap S$ uniformly at random, otherwise restart from 1.
 - (c) Let $S' = \{v \in [3t] \mid \text{dist}_H(a, v) \geq k \wedge \text{dist}_H(b, v) \geq k\}$.
 - (d) If $R \cap S' \neq \emptyset$: pick $c \in R \cap S'$ uniformly at random, otherwise restart from 1.
 - (e) Add triangle $\{a, b, c\}$ to T .
3. Return the instance defined by T .

Generation 3

Input: Number t of triangles, minimum span $k \geq 1$, and set T of triangles of a CYCLE-PLUS-TRIANGLES instance of span at least k .

1. For $i = |T|$ to t do:
 - (a) Pick a uniformly at random in $[3i]$.
 - (b) Pick (b, c) uniformly at random in the set S given by
$$S = \{(v, w) \in [3i] \times [3i] \mid \text{dist}_H(a, v) \geq k \wedge \text{dist}_H(a, w) \geq k \wedge \text{dist}_H(v, w) \geq k\}.$$
 - (c) Let G be the CYCLE-PLUS-TRIANGLES graph of the instance corresponding to the triangles T .
 - (d) Subdivide the edges of G incident to the vertices a, b, c (going in clockwise direction), respectively, and connect the midpoints to form a new triangle. Let T be the set of triangles of the resulting instance.
2. Return the instance defined by T .

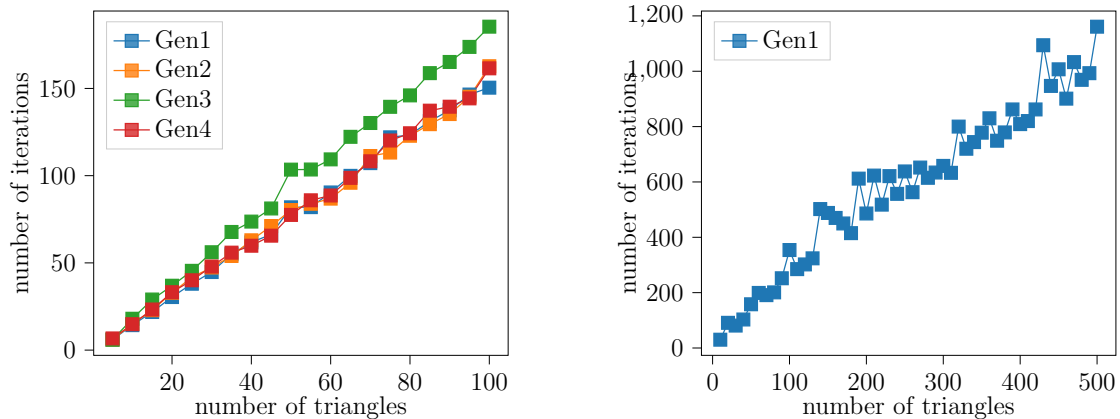
Generation 4

Input: Number t of triangles, minimum span $k \geq 1$, and number p of swaps.

1. (Initialization) Let T be the triangles of a chain-of-twins on t triangles.
2. For $i = 1$ to p :
 - (a) Pick vertex v uniformly at random in $[1, 3t]$.
 - (b) Pick a neighbor w of v on the Hamilton cycle uniformly at random.
 - (c) If after swapping v and w , we have $\sigma((T)) \geq k$, keep the swap.
3. Return the instance defined by T .

9.2 Experimental results

In this section, we present some computational results that support Conjecture 8.2.1. This is only a selection of the tests we performed, but it shows the behavior we generally observe, which is that for any instance, the **Rand-TS-Alg** appears to terminate after a roughly linear (in n) number of iterations of Step 2 of **Rand-TS-Alg** (i.e., token-sliding moves).



- (a) Maximum over 100 instances of the average number of iterations of **Rand-TS-Alg** over 100 runs for each instance.
- (b) Maximum over 100 instances of the number of iterations of **Rand-TS-Alg** on a single run. The instances with span at least three are generated from random permutations.

Figure 9.2: Results on instances with span at least three generated via methods in Section 9.1.

In most of the computational experiments presented in this paper, we compute the maximum over 100 instances of the average number of iterations over 100 runs of **Rand-TS-Alg** on each instance. This gives us an estimate of the worst-case expected runtime of our algorithm for the different types of instances obtained via our various generation methods. The instances were generated independently and in advance of running **Rand-TS-Alg** and saved. Hence if two experiments are run on instances with the same set of parameters (i.e., span, generator, and number of triangles), then they are actually run on the exact same set of instances.

The first computational results are shown in Figure 9.2. Here, we ran **Rand-TS-Alg** on instances with span at least three. We increased the number of triangles

in the instance from 5 to 100 by steps of 5 for each generation method presented in Section 9.1. As seen in Figure 9.2a, for each generation method, we observe an increase in the runtime of **Rand-TS-Alg** that is linearly proportional to the instance size. In Figure 9.2b, we show the maximum number of iterations for one run of the algorithm, over 100 instances generated from a random permutation, up to 500 triangles. Since we wanted to demonstrate the performance of the algorithm on some instances with a large number of triangles, we used random permutations corresponding to instances with span at least three rather than method Generation 1 (i.e., uniform generation), since the latter takes too long. The results are more noisy since we are running **Rand-TS-Alg** only once on each instance, but it still shows that the number of iterations is roughly linear in the size of the instance.

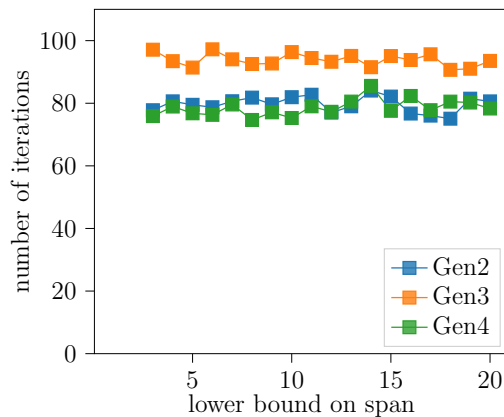
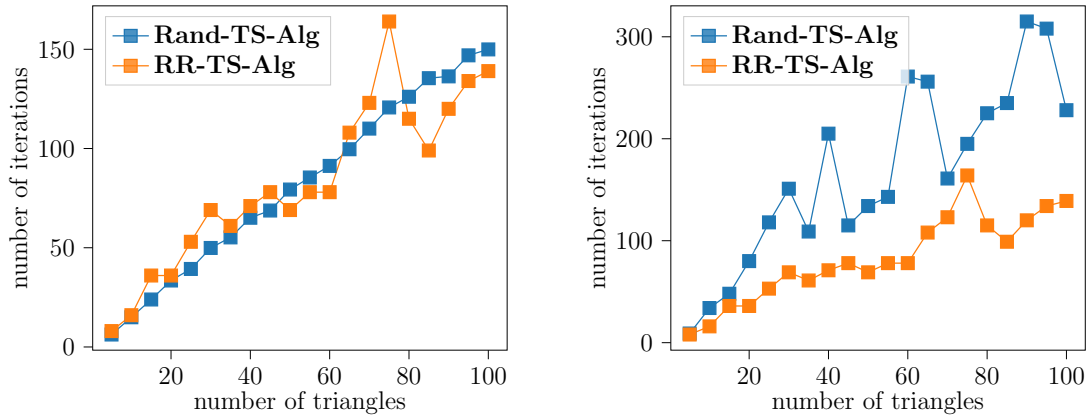


Figure 9.3: Maximum over 100 instances of the average number of iterations over 100 runs of **Rand-TS-Alg**. The instances have 50 triangles and were generated by the different high-span generation methods presented in Section 9.1.

In Section 9.1, we discussed the span as a key property of an instance. We tested whether larger span lead to a worse/slower performance for our algorithm. It seems that, for instances with a fixed number of triangles, larger span had no deleterious effect on the behavior of the algorithm for the instances we used. This is exhibited in Figure 9.3, where we consider instances on 50 triangles with span ranging from three to 20. These instances were obtained via the three high span generators (Generation 2, Generation 3 and Generation 4). In the figure, we plotted the maximum over 100 instances for each span of the average number of iterations of the algorithm over 100 runs.



(a) Maximum over 100 instances of the average number of iterations over 100 runs. The instances were generated by Gen1. (b) Maximum over 100 instances of the number of iterations on a single run. The instances were generated by Gen1.

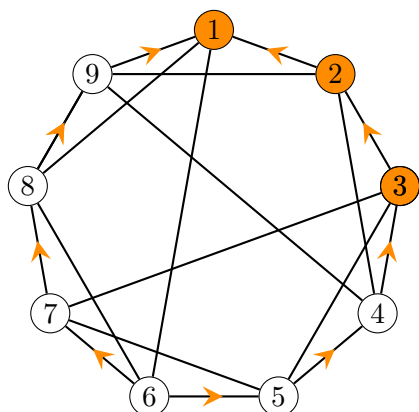
Figure 9.4: Comparison of the two stable set algorithms **Rand-TS-Alg** and **RR-TS-Alg**.

9.2.1 Performance of deterministic algorithm

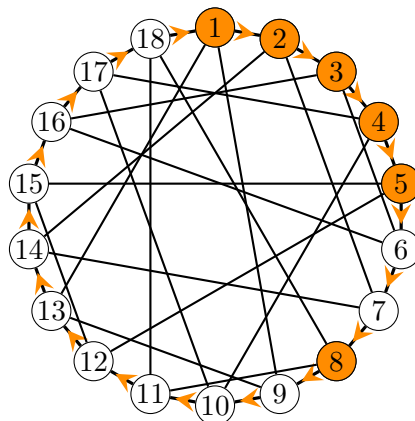
In Figure 9.4, we compare the performance of **Rand-TS-Alg** and its deterministic implementation **RR-TS-Alg**. We ran both algorithms on instances obtained via Generation 1. In this setting, both algorithms have similar worst-case performance, but **RR-TS-Alg** shows higher variance in the number of iterations for different instances. Notice that in Figure 9.4a, we take the number of iterations in a single run for the deterministic algorithm, whereas for the randomized algorithm, we take the average over 100 runs, which reduces the variance for **Rand-TS-Alg**. In practice, it is usually the case that the deterministic algorithm terminates much more quickly (see Figure 9.4b). In fact, it usually terminates in nearly two times fewer iterations. However, there are some instances on which **RR-TS-Alg** does not terminate. This is discussed in Section 9.3.

9.3 Lock-in cycles for RR-TS-Alg

The reason that we are interested in instances on which **RR-TS-Alg** does not terminate is that, since these instances seem to be very rare, perhaps we can uncover some property that is sufficient for efficient termination. In this case, then at least



(a) Graph with an initial configuration that leads to a lock-in cycle of length 12.



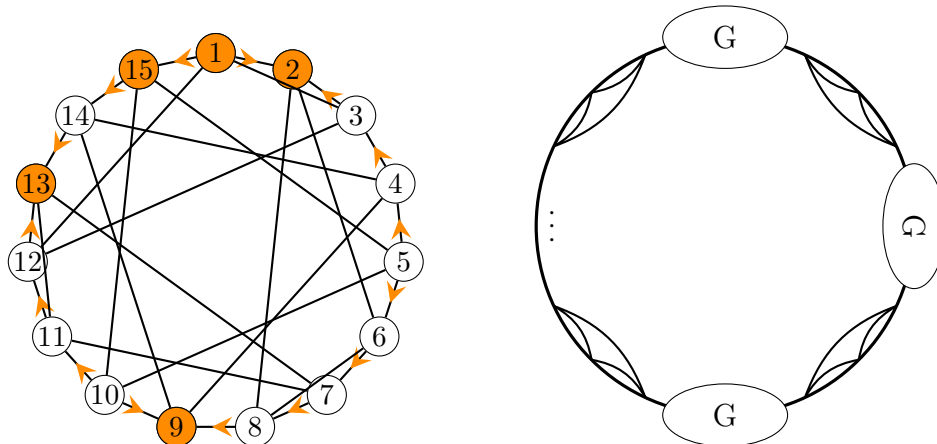
(b) Instance for which **RR-TS-Alg** runs into a lock-in cycle.

Figure 9.5: **CYCLE-PLUS-TRIANGLES** graphs with initial (token and rotor-router) configurations such that **RR-TS-Alg** runs into a lock-in cycle. The token configurations are represented by the orange vertices.

for instances with this property, we could find an alternate, constructive proof that a subclass of **CYCLE-PLUS-TRIANGLES** graphs have stable sets of size $n/3$.

The simplest example of an instance and an initial token configuration for which **Rand-TS-Alg** does not terminate is shown in Figure 9.5a. Notice that **Rand-TS-Alg** as written uses a *minimal token configuration* as its initial token configuration. Using such minimal token configurations as initializations, we ran **Rand-TS-Alg** on all instances containing 6 triangles or fewer and found that **Rand-TS-Alg** terminates on all but one instance shown in Figure 9.5b. Testing other rotor-router initializations for this same instance, it turns out that 5 of them (out of 2^{18}) have minimal token configurations that result in nontermination of the algorithm. While apparently extremely rare, this shows that **RR-TS-Alg** does not always terminate for certain combinations of initial token and rotor-router configurations.

For instances on which **RR-TS-Alg** does not terminate, the execution of **RR-TS-Alg** necessarily repeats configurations (here a *configuration* refers to both token positions and pointer positions). Such a series of repeated configurations is called a *lock-in cycle*. In fact, we are able to construct instances for which the probability of entering a lock-in cycle is arbitrarily close to 1, if we take a random initial configuration.



(a) Graph with an initial configuration that leads to a lock-in cycle for which the last edge (i.e., $(1, 15)$) is never critical. (b) Construction for boosting the probability of running into a lock-in cycle. Between any two copies of G there is a triangle on three consecutive vertices of the Hamilton cycle.

Figure 9.6: Construction of an instance for which **RR-TS-Alg** with random initial configuration runs into a lock-in cycle with high probability. In order to obtain such an instance, we use copies of the graph in 9.6a as graph G in 9.6b.

9.3.1 Instance with high probability of entering a lock-in cycle

In Figure 9.5 we gave examples for which the algorithm **RR-TS-Alg** with a certain initial configuration of the tokens and pointers does not terminate. In this section we show that choosing a random initial configuration is not a cure for this behavior. To this end we show that there is a class of instances for which the algorithm **RR-TS-Alg** with a random initial configuration does not terminate with high probability. Recall that **RR-TS-Alg** considers the monochromatic edge with the lowest index on the Hamilton cycle in order to update the token configuration and the pointer. We say that an edge of the Hamilton cycle is *critical* with respect to a lock-in cycle if it is at some point a monochromatic edge with the lowest index while the algorithm runs through the lock-in cycle.

Notice that if for some lock-in cycle all edges of an instance are critical, then the last edge will be the first monochromatic edge at some step, which gives a stable set of size $n - 1$. However, there are instances and initial configurations for which there

is a lock-in cycle in which some edges are never critical (e.g., see Figure 9.6a). From these instances, we can build a family for which an exponentially small fraction of random initial configurations lead to termination for the **RR-TS-Alg**. This in turn proves that randomness on the initialization does not suffice to guarantee termination with high probability.

Lemma 9.3.1. *Given an instance that has some initial configuration with a lock-in cycle in which the last edge is never critical, we can build a family of instances for which the probability of a random initial configuration leading to a lock-in cycle goes to 1 as the number of triangles goes to infinity.*

Proof. Let G be a CYCLE-PLUS-TRIANGLES graph that has some initial configuration for which **RR-TS-Alg** runs into a lock-in cycle, such that the last edge on the Hamilton cycle is not critical. We now define a family G_i of CYCLE-PLUS-TRIANGLES graphs, such that G_1 is G , where the last edge of the Hamilton cycle (which is not critical) is subdivided three times and a triangle is added on the three new vertices. Let G'_i be a copy of G_i from which we remove the last edge of the Hamilton cycle. We create G_{i+1} by stitching together the two graphs G'_i and G'_1 , identifying the last vertex of the Hamilton path of G'_i and the first vertex of the Hamilton path of G'_1 as well as the last vertex of the Hamilton path of G'_1 and the first vertex of the Hamilton path of G'_i . Figure 9.6b shows a schematic depiction of this construction.

Notice that if the initial configuration on any of the copies of G is randomly chosen to be exactly one of the configurations that leads to a lock-in cycle for which the last edge of the Hamilton cycle of G is never critical, and both adjacent triangles have their token on the center vertex or the one not adjacent to that copy, then the algorithm will run into a lock-in cycle on that copy of G , if it did not run into a lock-in cycle in any of the previous (with respect to the Hamilton cycle of G_i) copies.

Let $p > 0$ be the probability that **RR-TS-Alg** with random initial configuration runs into a lock-in cycle on G . Then, for any $i \geq 1$, the probability that **RR-TS-Alg** with random initial configuration avoids a lock-in cycle on G_i is $(1 - p)^i$. Therefore, **RR-TS-Alg** with random initial configuration runs into a lock-in cycle with probability at least $1 - (1 - p)^i$, which approaches 1 as i goes to infinity. \square

While these instances are built to be bad for the **RR-TS-Alg** (with random initialization), **Rand-TS-Alg** finds a maximum stable set in linear time for them: Indeed, once a stable set is found for one of the copies, it remains fixed because of the triangles inserted between the copies. The number of iterations is proportional to

the number of iterations required for the small instance times the number of copies, which at least in this case, is linear in the size of the instance. We note that one caveat is that this instance has span 1.

Chapter 10

Extensions

In this chapter we will present different types of extensions of our algorithms, to different problems and to different types of graphs, in order to show the limitations of our approach.

10.1 Extensions to multiple stable sets and fair representation

We extended our algorithm to several related problems that are less likely to have efficient algorithms. The first (straightforward) extension is to find two or three disjoint maximum stable sets. The second extension is to the problem **FAIR-REPRESENTATION**, which is known to be PPA-complete [Hav21]. If our algorithm worked well, even for random instances of these harder problems, this could suggest that the instances we are considering are simply too limited to get any meaningful results out of our computational experiments. The fact that our algorithm does not perform well on these generalizations leads us to believe that the stable set problem is actually an easier problem.

10.1.1 Two and three maximum stable sets

We can extend the randomized algorithm **Rand-TS-Alg** in a natural way to find multiple disjoint maximum stable sets. First, we consider the case in which the goal

is to find two or three vertex disjoint maximum stable sets. In this case, we can think of each triangle as having three tokens, one on each vertex. The red token corresponds to the first stable set, the blue token corresponds to the second stable set, and the green token corresponds to the third stable set. In the case where we want to find two disjoint maximum stable sets, a monochromatic edge in E_H is an edge with either two red or two blue tokens. In the case where we want to find three disjoint maximum stable sets, a monochromatic edge in E_H is an edge with either two red, two blue or two green tokens. A minimal token configuration is defined so that each triangle (i, j, k) with $i < j < k$ contains a red token on i , a blue token on j , and a green token on k .

Multi-SS-Token-Sliding-Algorithm

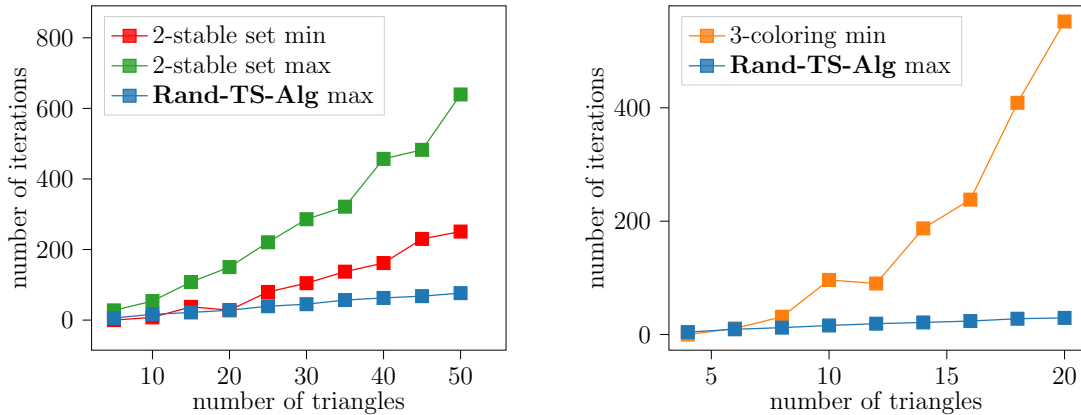
Input: A CYCLE-PLUS-TRIANGLES graph $G = (V, E_T \cup E_H)$.

1. (Initialization) Fix the minimal token configuration.
2. While there is a monochromatic edge in E_H :
 - (a) Choose monochromatic edge $e \in E_H$ with lowest index.
 - (b) Choose endpoint j of e at random and let k denote the clockwise neighbor of j .
 - (c) Swap tokens on j and k .

Thus, upon termination, **Multi-SS-Token-Sliding-Algorithm** outputs one, two or three stable sets depending on the definition of monochromatic edge. We compared the performance of the algorithm in the cases of finding one, two or three stable sets. This is shown in Figures 10.1a and 10.1b. It appears that the number of iterations required to find a 3-coloring evolves in an exponential manner, while the evolution for finding two disjoint stable sets is less clear, though evidently not linear as for the stable set problem.

10.1.2 Fair representation

We also extend our algorithm to the FAIR-REPRESENTATION problem, which is defined as follows. We are given a Hamilton cycle and a partition of the vertex set



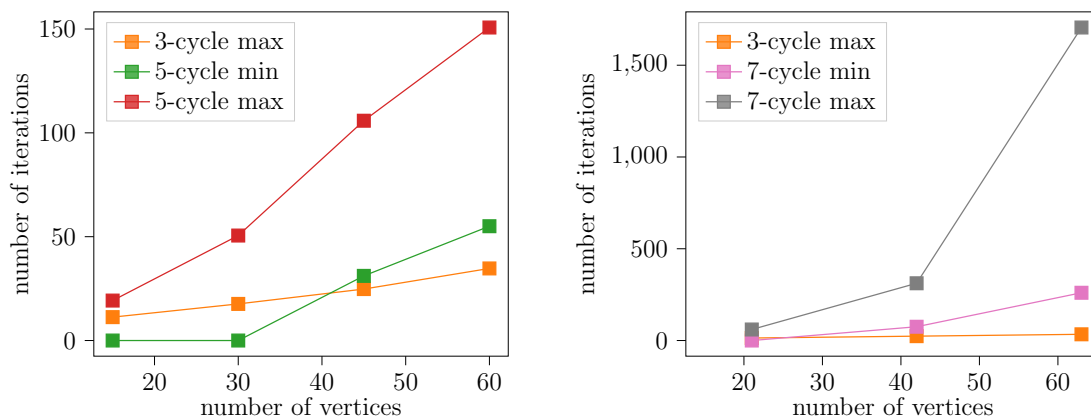
(a) For stable set and 2-stable set, maximum (and minimum number for 2-stable set) of average iterations over 100 instances taken over 100 runs with instances obtained via Generation 1. (b) Maximum for stable set and minimum for 3-stable set over 100 instances of the average taken over 100 runs.

Figure 10.1: Comparison of the 1-, 2- and 3-stable set algorithms for instances obtained via Generation 1.

$V = \{V_1, V_2, \dots, V_\ell\}$. The goal is to find a stable set of the cycle I such that $|I \cap G_i| \geq \lfloor |G_i - 1|/2 \rfloor$ for all $i \in [\ell]$. Notice that this can be viewed as a generalization of the problem of finding a maximum stable set in a **CYCLE-PLUS-TRIANGLES** graph, since choosing the partitions of V to be triples yields an equivalent problem statement. This problem was shown to be PPA-complete [Hav21], which suggests that finding an efficient algorithm for it is unlikely.

The algorithm we implemented is for a subclass of this problem, in which the partition is into a set of odd cycles. Each cycle C_{2k+1} then has k tokens, which are moved the same way as in **Rand-TS-Alg**: If two are adjacent on the Hamiltonian cycle, one is chosen at random and we slide it to the next vertex (in clockwise order) of the C_{2k+1} that does not already have a token.

We ran this implementation for different lengths of cycles: five and seven, and show the results in Figures 10.2a and 10.2b, which shows a clear degradation in performance as the partition changes from a 3- to 5- to 7-cycles. The instances were generated from random permutations, taking the first i elements in the first cycle, the next i in the next cycle, etc. There was no condition on the span of the instances, as this property has not been studied for this more general problem.



(a) Maximum and minimum over 100 instances of the average taken over 100 runs number of iterations of the fair representation algorithm on 3- and 5-cycles. (b) Maximum and minimum over 100 instances of the average taken over 100 runs number of iterations of the fair representation algorithm on 3- and 7-cycles.

Figure 10.2: Comparison of the performance of the extension of **Rand-TS-Alg** to **FAIR-REPRESENTATION** for different cycle lengths for problem instances generated from random permutations.

10.2 Triangles-plus-two-factors graphs

A natural extension of the **CYCLE-PLUS-TRIANGLES** problem is to look at graphs that are the disjoint union on the same vertex set, of a set of t vertex disjoint triangles, and a 2-regular graph (which we will refer to as a 2-factor). Notice that **CYCLE-PLUS-TRIANGLES** graphs are a subclass of **TRIANGLES-PLUS-TWO-FACTOR** graphs, and correspond to the case where the 2-factor is a Hamilton cycle.

Graphs belonging to this class are not always 3-colorable, since they can contain 4-cliques. Their independence number may also be as low as $n/4$: an example of such a **TRIANGLES-PLUS-TWO-FACTOR** graph is found in [VW09], and presented in Figure 10.3. The conditions for these graphs to have independence number $n/3$ were studied in [AHHS15], where the authors prove that length mod 3 of the cycles in the 2-factor is crucial since if there are at most two cycles of length $1 \pmod 3$, the graph has independence number $n/3$. It is interesting to note that for both extremes of the length of the cycles making up the 2-factor, triangles on the one hand and a Hamilton cycle on the other, the graphs are 3-colorable.

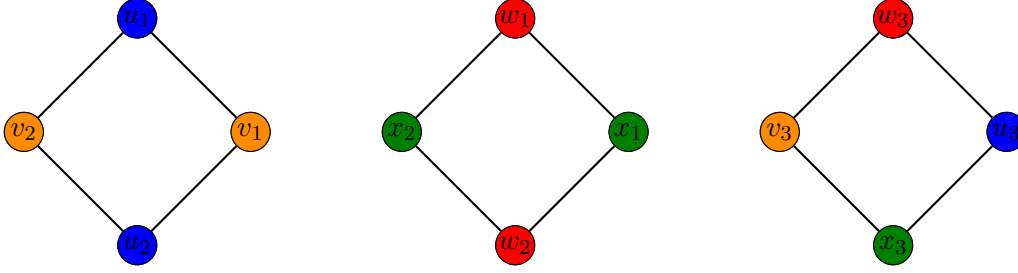


Figure 10.3: TRIANGLES-PLUS-TWO-FACTOR graph on 12 vertices with independence number 3. The triangles are (u_1, u_2, u_3) , (v_1, v_2, v_3) , (w_1, w_2, w_3) and (x_1, x_2, x_3) , also represented by the color of the vertices.

10.2.1 Triangles-plus-triangles graphs

One of the extreme cases of TRIANGLES-PLUS-TWO-FACTOR graphs is when the 2-factor is also a set of vertex disjoint triangles: we will call such graphs TRIANGLES-PLUS-TRIANGLES graphs.

Definition 10.2.1. A TRIANGLES-PLUS-TRIANGLES graph is the disjoint union, on the same vertex set, of two sets of t vertex disjoint triangles.

In [VW09], the authors prove the 3-choosability of TRIANGLES-PLUS-TRIANGLES graphs by using a theorem of Galvin [Gal95]. This proof uses Galvin's theorem, so the corresponding coloring can be found in polynomial time, as pointed out in [Sli96]. We propose an alternative proof that shows the weaker property of 3-colorability with efficient computability using Hall's theorem [Hal87].

Lemma 10.2.1. Let G be a TRIANGLES-PLUS-TRIANGLES graph. G is 3-colorable in polynomial time.

Proof. Let G be a TRIANGLES-PLUS-TRIANGLES graph with T_1 and T_2 the two sets of triangles. Build a new graph B by adding a vertex v_i for every triangle t_i of G . Then for every vertex v of G such that v is in triangles t_i and t_j , add an edge to B between vertices v_i and v_j (that correspond to the triangles t_i and t_j). Note that this graph is bipartite, since triangles from T_1 can only share vertices with triangles from T_2 and vice-versa. Also, since a triangle has three vertices, that are all shared with other triangles, B will be 3-regular (allowing multi-edges).

Now, we can apply Hall's marriage theorem to B in order to find a perfect matching. This matching can be seen as an independent set of G by taking the vertices of

G from which every edge in the matching of B was derived. Notice that every triangle will have exactly one vertex in this set, as its vertex has one edge in the perfect matching. This can be repeated on the graph B' obtained from B by removing all the edges of the first perfect matching, since B' will be bipartite and 2-regular. This gives us another disjoint independent set of G , that can be used as the second color, and final one is obtained from the set of remaining edges.

Thus, we 3-colored G by using Hall's marriage theorem, whose matchings can be found in polynomial time. \square

In the following lemma we will show the reducibility of TRIANGLES-PLUS-TRIANGLES instances with low span.

Lemma 10.2.2. *Let G be a TRIANGLES-PLUS-TRIANGLES graph. If G contains any triangle with two or more vertices from another triangle (i.e., it has span one), it can be reduced in polynomial time to a smaller TRIANGLES-PLUS-TRIANGLES graph where this is not the case (ie. it has span infinity).*

Proof. Take a TRIANGLES-PLUS-TRIANGLES graph G on triangle sets T_1 and T_2 . Suppose this graph is of span 1: there is some triangle $t_a \in T_1$ that contains two vertices of another triangle $t_b \in T_2$. We will call these two vertices v_1 and v_2 , the last vertex of t_a is v_3 , and the last vertex of t_b is v_4 . Finally, we will call t_c the triangle from T_2 that contains v_3 .

Consider the graph G' obtained by removing the triangle t_a (ie. the vertices v_1 , v_2 and v_3). Now add the vertex v_4 to t_c and remove the triangle t_b . We want to show that from any maximum stable set of G' we can derive a maximum stable set of G in polynomial time. Take a stable set I' of G' of size $n/3$. If $v_4 \in I'$, let $I = I' \cup \{v_3\}$, and I is a stable set of G since t_c has no other vertex in I' (or it wouldn't be a stable set). Otherwise, let $I = I' \cup \{v_1\}$ (we could also take v_2), and we also get a maximum stable set of G since v_4 is not in I .

In conclusion, if G has span one it reduces to a smaller instance G' . Therefore, we can do this successively until reaching a graph that does not have span one (at the latest by reaching an empty graph). \square

10.2.2 Hardness results

Lemma 10.2.3. *Let G be a TRIANGLES-PLUS-TWO-FACTOR graph. Deciding whether its independence number is $n/3$ is NP-hard.*

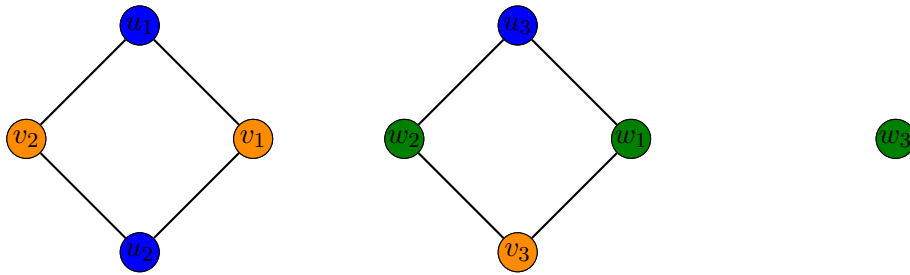


Figure 10.4: TRIANGLES-PLUS-TWO-FACTOR subgraph where w_3 , which can be placed anywhere, is forced to contain a token in any stable set of size $n/3$. The triangles are (u_1, u_2, u_3) , (v_1, v_2, v_3) and (w_1, w_2, w_3) , also represented by the color of the vertices.

Proof. We will abuse notation by referring to vertices belonging to a stable set as having a token in a given stable set. In order to prove this lemma, we will start by showing that it is possible to build a subgraph that forces a given vertex to contain a token in any stable set of size $n/3$. Consider the subgraph in Figure 10.4 and suppose it has a stable set of size $n/3$ where w_3 does not contain a token. Then, w_1 or w_2 must contain a token, which implies that both u_3 and v_3 do not. However, it is impossible for v_1 or v_2 and u_1 or u_2 to simultaneously contain tokens, hence one of the two triangles has no tokens and the stable set is not of size $n/3$.

Then we can use multiple vertices with forced tokens (in stable sets of size $n/3$) in order to block some vertices, and get triangles where only two vertices could contain a token. This is simply done by making cycles of length 2 with one forced vertex adjacent to the third vertex of some new triangle.

We now have all the tools we need in order to reduce the NP-complete problem $2P2N3 - SAT$ to deciding whether a TRIANGLES-PLUS-TWO-FACTOR graph has a stable set of size $n/3$. The $2P2N3 - SAT$ problem is a variant of $3 - SAT$ in which every variable appears as a positive literal twice and as a negative literal twice. This was proven to be NP-complete in [BKS03].

Take an instance of $2P2N3 - SAT$. For every variable, we will create a triangle with one vertex blocked by a forced vertex, such that in any stable set of size $n/3$, it has only two vertices that could contain a token. Arbitrarily designate one of these two vertices as *True* and one as *False* for every variable. Then, for every clause, add a triangle adjacent on the 2-factor, to the *False* vertex of a variable if that variable appears as a positive literal, and to the *True* vertex if it appears as a negative literal. Because the clause is from 3-sat, it contains 3 literals and can be

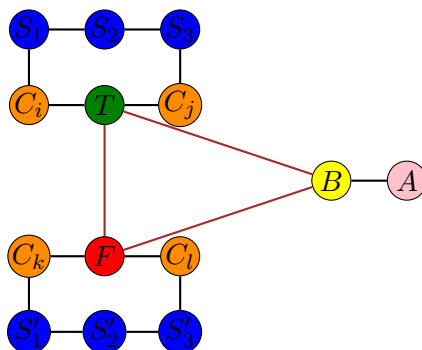


Figure 10.5: Gadget for the reduction from $2P2N3-SAT$. The triangle edges are in brown, and the triangle (T, F, B) represents some variable from the formula. B cannot have a token because A is forced (like w_3 in Figure 10.4), so in a stable set of size $n/3$ either T or F have a token: the corresponding variable is true in the equivalent truth assignment iff the token is on T . The clauses of the 3-SAT formula are represented by the triangles on vertices C_i . Finally, the blue vertices form triangles on consecutive vertices (ie. (S_1, S_2, S_3) is a triangle) used as separators where a token can be placed on the middle vertex (here S_2) without creating monochromatic edges.

represented by a triangle. Since each variable appears twice as a positive and twice as a negative literal, every *True* and *False* vertex will be adjacent to two clause triangles on the 2-factor, which is fine. To complete the graph into a TRIANGLES-PLUS-TWO-FACTOR graph, one can close any paths into cycles by adding a triangle on 3 consecutive vertices between its endpoints. Figure 10.5 shows the gadget derived from one variable of the SAT formula.

It now remains to show that the TRIANGLES-PLUS-TWO-FACTOR graph we constructed has a stable set of size $n/3$ iff the $2P2N3-SAT$ formula is satisfiable. Suppose the formula is satisfiable, then take any satisfying assignment and add to the token set every *True* vertex from a triangle derived from a variable that is *True* in the satisfying assignment, and do the same for the *False* vertices. Also add to the token set tokens of the initial construction that force some vertices that are compatible with a stable set of size $n/3$ (ie. the forced vertices will be in the token set). Finally, there are only the triangles representing the clauses that do not have a token yet. However, since every clause is satisfied, it contains a variable with the satisfying truth assignment, hence its triangle has a vertex adjacent to the vertex of that variable with no token, and some separator triangle, so this vertex can be given a token. All separator triangles can be given a token on their middle vertex so as not to interfere, and we now have a stable set of size $n/3$.

Conversely, if we have a stable set of size $n/3$, every forced vertex must have a token, so every variable triangle has a token either on its *True* vertex or its *False* vertex. Take the truth assignment that gives to every variable the value of the vertex with a token on it. Since every clause triangle has some vertex with a token, its neighboring variable vertex has no token, so the clause is satisfied by that variable in the derived truth assignment. Therefore every clause is satisfied and the $2P2N3$ -SAT formula is True. \square

Note that this hardness proof relies on 2-factors that contain 4-cycles, which is consistent with the results of [AHHS15] since those are the only TRIANGLES-PLUS-TWO-FACTOR graphs whose independence number is not guaranteed to be $n/3$.

A nice question is whether the graphs on triangles and 4-cycles, where the 4-cycles are restricted to have at most one vertex per triangle (ie. infinite span) always have independence number $n/3$. If so, can one make an equivalent hardness proof?

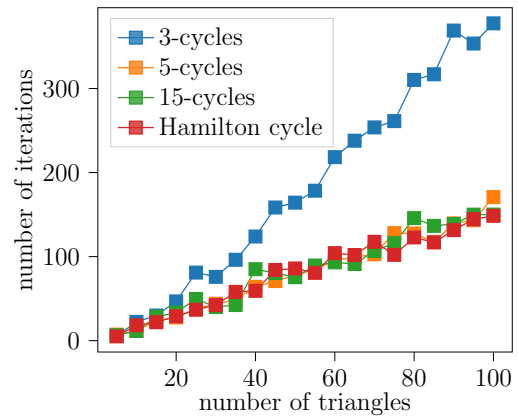
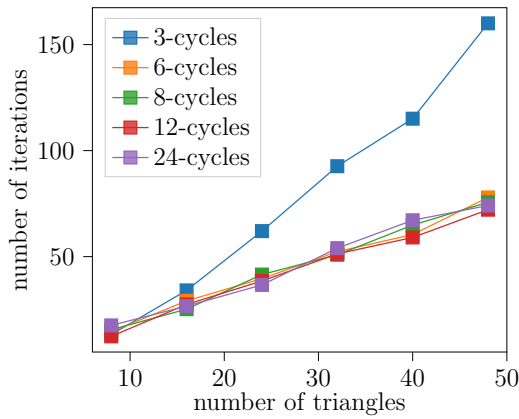
10.2.3 Why intuition fails

The formula for the number of free vertices is the same for TRIANGLES-PLUS-TWO-FACTOR and CYCLE-PLUS-TRIANGLES graphs. Therefore, one can expect the algorithm to have a higher probability to reach a free vertex than a double-blocked vertex, at least starting from a random token configuration.

Examining the execution of the algorithm on the graph from Figure 10.3, it appears that where the intuition fails is on the probability of reaching a free vertex: while 2 of the 12 vertices will be free, they cannot be adjacent to a vertex from the monochromatic edge, hence the algorithm will loop infinitely without ever reaching either a double-blocked vertex (of which there are none) or a free vertex. This seems to be linked to the equivalence of some vertices: for example v_1 and v_2 have the same neighbors, so moving a token from v_1 to v_2 is actually equivalent to doing nothing.

10.2.4 Experimental results for triangles-plus-two-factors

Figures 10.6a and 10.6b compare the performance of our algorithm on graphs that are the union of a set of triangles and a set of cycles of given length. These instances are generated by taking a random permutation of the vertex set to determine the triangles, and considering the first k -cycle to be on the first k vertices, then the next on the following k , etc.



- (a) Maximum over 100 instances on varying number of triangles of the average number of iterations over 100 runs of **Rand-TS-Alg** for two-factor plus triangles, where the two-factor is a disjoint union of cycles of given length.
- (b) Maximum over 10 instances on varying number of triangles of the average number of iterations over 10 runs of **Rand-TS-Alg** for two-factor plus triangles, where the two-factor is a disjoint union of cycles of given length.

Figure 10.6: Comparison of the algorithm on graphs that are the union of triangles and cycles of given length.

Chapter 11

Conclusion of Part II

We conclude with some related open problems. Are there hard/easy instances for finding a maximum stable set in a CYCLE-PLUS-TRIANGLES graph? Call a triangle *bad* when it has all sides with length $0 \pmod 2$ (assuming n is even). If an instance G has no bad triangles, notice that it is easy to find a maximum stable set. Similarly, we can (re)define a bad triangle as one that has at least one side with length $0 \pmod 3$. Then if instance G has no bad triangles, it is easy to 3-color G : simply assign colors along the Hamilton cycle in the clockwise order $B - G - R - B - G - \dots - B - G - R$. Since no edge has length equal to $0 \pmod 3$, there will be no monochromatic edges. Having many bad triangles does not necessarily make the respective problems hard, since a random instance will have a constant fraction of (both types of) bad triangles in expectation, and **Rand-TS-Alg** seems to work well on such instances. Is there an algorithm for finding a maximum stable set or a 3-coloring whose running time is parametrized by the number of bad triangles?

Finally, an interesting related problem is due to Petrov [Pet17]. Given a CYCLE-PLUS-TRIANGLES graph, find a subset $F \subset E_T$ of triangle edges (exactly one per triangle) so that each edge in F is intersected by an even number (possibly zero) of other edges in F . That such a subset F exists was shown by Petrov via a non-constructive proof based on parity and the polynomial method. Notice that this problem is not equivalent to finding 2 or 3 stable sets in a CYCLE-PLUS-TRIANGLES graph. Indeed, the desired subset F directly yields two maximum stable sets, but 3-colorings do not always yield a solution to Petrov's problem.

Bibliography

- [AAB⁺17] Ron Aharoni, Noga Alon, Eli Berger, Maria Chudnovsky, Dani Kotlar, Martin Loeb, and Ran Ziv. Fair representation by independent sets. In *A Journey Through Discrete Mathematics*, pages 31–58. Springer, 2017.
- [AAC22] Pierre Aboulker, Guillaume Aubian, and Pierre Charbit. Heroes in oriented complete multipartite graphs. *arXiv:2202.13306*, 2022.
- [AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
- [AHHS15] Ron Aharoni, Ron Holzman, David Howard, and Philipp Sprüssel. Cooperative colorings and independent systems of representatives. *The Electronic Journal of Combinatorics*, 22(2):P2.27, 2015.
- [AKMR96] Noga Alon, Pierre Kelsen, Sanjeev Mahajan, and Hariharan Ramesh. Coloring 2-colorable hypergraphs with a sublinear number of colors. *Nordic Journal of Computing*, 3:425–439, 1996.
- [Alo90] Noga Alon. Non-constructive proofs in combinatorics. *Proceedings of the International Congress of Mathematicians (ICM 1990)*, pages 1421–1429, 1990.
- [Alo93] Noga Alon. Restricted colorings of graphs. In *Surveys in Combinatorics, Proceedings of 14th British Combinatorial Conference*, volume 187, pages 1–33, 1993.
- [Alo02] Noga Alon. Discrete mathematics: Methods and challenges. *Proceedings of the International Congress of Mathematicians (ICM 2002)*, pages 119–135, 2002.
- [Alo22] Noga Alon. Fair partitions. 2022.

- [APS01] Noga Alon, János Pach, and József Solymosi. Ramsey-type theorems with forbidden subgraphs. *Combinatorica*, 21(2):155–170, 2001.
- [AT92] Noga Alon and Michael Tarsi. Colorings and orientations of graphs. *Combinatorica*, 12(2):125–134, 1992.
- [BBKP23] Thomas Bellitto, Nicolas Bousquet, Adam Kabela, and Théo Pierron. The smallest 5-chromatic tournament. *Mathematics of Computation*, 2023.
- [BCC⁺13] Eli Berger, Krzysztof Choromanski, Maria Chudnovsky, Jacob Fox, Martin Loebl, Alex Scott, Paul Seymour, and Stéphan Thomassé. Tournaments and colouring. *Journal of Combinatorial Theory, Series B*, 103(1):1–20, 2013.
- [BFJ⁺04] Drago Bokal, Gasper Fijavz, Martin Juvan, P. Mark Kayll, and Bojan Mohar. The circular chromatic number of a digraph. *Journal of Graph Theory*, 46(3):227–240, 2004.
- [Bha18] Amey Bhangale. NP-hardness of coloring 2-colorable hypergraph with poly-logarithmically many colors. In *Proceedings of 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 15:1–15:11, 2018.
- [BK17] Kristóf Bérczi and Yusuke Kobayashi. An algorithm for identifying cycle-plus-triangles graphs. *Discrete Applied Mathematics*, 226:10–16, 2017.
- [BKO19] Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, pages 602–613, 2019.
- [BKS03] Piotr Berman, Marek Karpinski, and Alex Scott. Approximation hardness of short symmetric instances of max-3sat. *Electronic Colloquium on Computational Complexity (ECCC)*, 01 2003.
- [Blu94] Avrim Blum. New approximation algorithms for graph coloring. *Journal of the ACM*, 41(3):470–516, 1994.
- [Bon03] J. Adrian Bondy. Short proofs of classical theorems. *Journal of Graph Theory*, 44(3):159–165, 2003.

- [BRSW12] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-Source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, pages 1483–1543, 2012.
- [CCS⁺17] Jan Dean Catarata, Scott Corbett, Harry Stern, Mario Szegedy, Tomas Vyskocil, and Zheng Zhang. The Moser-Tardos resample algorithm: Where is the limit? (an experimental inquiry). In *Proceedings of the 19th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 159–171. SIAM, 2017.
- [CF96] Hui Chen and Alan Frieze. Coloring bipartite hypergraphs. In *Proceedings of the 5th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 345–358, 1996.
- [Chu14] Maria Chudnovsky. The Erdős-Hajnal conjecture—A survey. *Journal of Graph Theory*, 75(2):178–190, 2014.
- [CHZ07] Xujin Chen, Xiaodong Hu, and Wenan Zang. A min-max theorem on tournaments. *SIAM Journal on Computing*, 37(3):923–937, 2007.
- [Coo71] Stephen A. Cook. The complexity of theorem proving procedure. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.
- [CSSS24] Maria Chudnovsky, Alex Scott, Paul Seymour, and Sophie Spirkl. Pure pairs. X. Tournaments and the strong Erdős-Hajnal property. *European Journal of Combinatorics*, 115:103786, 2024.
- [Des54] Blanche Descartes. k -Chromatic graphs without triangles. *American Mathematical Monthly*, 61(5):352–353, 1954.
- [DHH93] D.Z. Du, D.F. Hsu, and F.K. Hwang. The Hamiltonian property of consecutive- d digraphs. *Mathematical and Computer Modelling*, 17(11):61–63, 1993.
- [DKPS10] Irit Dinur, Subhash Khot, Will Perkins, and Muli Safra. Hardness of finding independent sets in almost 3-colorable graphs. In *Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 212–221, 2010.
- [DRS05] Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005.

- [EE85] Mohamed El-Zahar and Paul Erdős. On the existence of two non-neighborly subgraphs in a graph. *Combinatorica*, 5:295–300, 1985.
- [EH89] Paul Erdős and András Hajnal. Ramsey-type theorems. *Discrete Applied Mathematics*, 25(1-2):37–52, 1989.
- [EM64] Paul Erdős and Leo Moser. On the representation of directed graphs as unions of orderings. *Math. Inst. Hung. Acad. Sci.*, 9:125–132, 1964.
- [Erd79] Paul Erdős. Problems and results in number theory and graph theory. In *Proceedings of the 9th Manitoba Conference on Numerical Math and Computing*, pages 3–21, 1979.
- [Fel90] Michael R. Fellows. Transversals of vertex partitions in graphs. *SIAM Journal on Discrete Mathematics*, 3(2):206–215, 1990.
- [FGSY19] Jacob Fox, Lior Gishboliner, Asaf Shapira, and Raphael Yuster. The removal lemma for tournaments. *Journal of Combinatorial Theory, Series B*, 136:110–134, 2019.
- [FHS19] Tomás Feder, Pavol Hell, and Carlos Subi. Complexity of acyclic colorings of graphs and digraphs with degree and girth constraints. *arXiv:1907.00061*, 2019.
- [FK96] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Proceedings 11th Annual IEEE Conference on Computational Complexity (CCC)*, pages 278–287, 1996.
- [FS92] Herbert Fleischner and Michael Stiebitz. A solution to a colouring problem of P. Erdős. *Discrete Mathematics*, 101(1-3):39–48, 1992.
- [Gal95] Fred Galvin. The list chromatic index of a bipartite multigraph. *Journal of Combinatorial Theory, Series B*, 63(1):153–158, 1995.
- [GK00] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. In *Proceedings 15th Annual IEEE Conference on Computational Complexity (CCC)*, pages 188–197, 2000.
- [GK04] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. *SIAM Journal on Discrete Mathematics*, 18(1):30–40, 2004.

- [GP18] Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions. *Journal of Computer and System Sciences*, 94:167–192, 2018.
- [GS20a] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 2020.
- [GS20b] Venkatesan Guruswami and Sai Sandeep. d -to-1 hardness of coloring 3-colorable graphs with $o(1)$ colors. In *Proceedings of 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2020.
- [Hal87] Philip Hall. On representatives of subsets. *Classic Papers in Combinatorics*, pages 58–62, 1987.
- [Hal93] Magnús M Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 1993.
- [Has99] Johan Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
- [Hav21] Ishay Haviv. The complexity of finding fair independent sets in cycles. In *Proceedings of 12th Annual Conference on Innovations in Theoretical Computer Science Conference (ITCS)*, volume 185, pages 4:1–4:14, 2021.
- [HLNT19] Ararat Harutyunyan, Tien-Nam Le, Alantha Newman, and Stéphan Thomassé. Coloring dense digraphs. *Combinatorica*, 39(5):1021–1053, 2019.
- [HLS23] Kun He, Qian Li, and Xiaoming Sun. Moser-Tardos algorithm: Beyond Shearer’s bound. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3362–3387. SIAM, 2023.
- [HLTW19] Ararat Harutyunyan, Tien-Nam Le, Stéphan Thomassé, and Hehui Wu. Coloring tournaments: From local to global. *Journal of Combinatorial Theory, Series B*, 138:166–171, 2019.
- [Kap44] Irving Kaplansky. Symbolic solution of certain problems in permutations. *Bulletin of the American Mathematical Society*, 50(12):906–914, 1944.
- [Kar72] Richard Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

- [Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.
- [KLS00] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.
- [KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.
- [KN23a] Felix Klingelhofer and Alantha Newman. Bounding the chromatic number of dense digraphs by arc neighborhoods. *arXiv:2307.04446*, 2023.
- [KN23b] Felix Klingelhofer and Alantha Newman. Coloring tournaments with few colors: Algorithms and complexity. In *Proceedings of the 31st Annual European Symposium on Algorithms (ESA)*, 2023.
- [KNS01] Michael Krivelevich, Ram Nathaniel, and Benny Sudakov. Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *Journal of Algorithms*, 41(1):99–113, 2001.
- [KS14] Subhash Khot and Rishi Saket. Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1607–1625, 2014.
- [KT17] Ken-ichi Kawarabayashi and Mikkel Thorup. Coloring 3-colorable graphs with less than $n^{1/5}$ colors. *Journal of the ACM*, 64(1):1–23, 2017.
- [LMM⁺21] Daniel Lokshantov, Pranabendu Misra, Joydeep Mukherjee, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. 2-Approximating feedback vertex set in tournaments. *ACM Transactions on Algorithms*, 17(2):1–14, 2021.
- [Lov73] László Lovász. Coverings and colorings of hypergraphs. In *Proceedings of the 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.
- [MT10] Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM*, 57(2):1–15, 2010.

- [MW11] Kevin Milans and Douglas B. West. Chromatic numbers of tournaments, 2011. <https://dwest.web.illinois.edu/regs/chromtourn.html>, Last accessed on 2023-09-11.
- [Neu82] Victor Neumann-Lara. The dichromatic number of a digraph. *Journal of Combinatorial Theory, Series B*, 33(3):265–270, 1982.
- [NL94] Victor Neumann-Lara. The 3 and 4-dichromatic tournaments of minimum order. *Discrete Mathematics*, 135(1-3):233–243, 1994.
- [NSS23a] Tung Nguyen, Alex Scott, and Paul Seymour. On a problem of El-Zahar and Erdős. *arXiv:2303.13449*, 2023.
- [NSS23b] Tung Nguyen, Alex Scott, and Paul Seymour. Some results and problems on tournament structure. *arXiv:2306.02364*, 2023.
- [Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
- [PDDK96] Vyatcheslav B. Priezzhev, Deepak Dhar, Abhishek Dhar, and Supriya Krishnamurthy. Eulerian walkers as a model of self-organized criticality. *Physical Review Letters*, 77(25):5079, 1996.
- [Pet17] Fedor Petrov. General parity result and cycle-plus-triangles graphs. *Journal of Graph Theory*, 85(4):803–807, 2017.
- [Sac94] Horst Sachs. Elementary proof of the cycle-plus-triangles theorem. Technical report, Cahiers du GERAD, 1994.
- [Sch78] Alexander Schrijver. Vertex-critical subgraphs of kneser-graphs. 1978.
- [Sco22] Alex Scott. Graphs of large chromatic number. In *Proceedings of the International Congress of Mathematicians (ICM 2022)*, 2022.
- [Sin22] Alistair Sinclair. Lecture notes: CS271 Randomness & Computation, Fall 2022. <https://people.eecs.berkeley.edu/~sinclair/cs271/n21.pdf>.
- [Sli96] Tomaz Slivnik. Short proof of Galvin’s theorem on the list-chromatic index of a bipartite multigraph. *Combinatorics, Probability and Computing*, 5(1):91–94, 1996.

- [VW09] Jennifer Vandenbussche and Douglas B. West. Independence number of 2-factor-plus-triangles graphs. *The Electronic Journal of Combinatorics*, page R27, 2009.
- [Wig83] Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *Journal of the ACM*, 30(4):729–735, 1983.
- [Zuc06] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 681–690, 2006.