5-2015

Some Multicolor Ramsey Numbers Involving Cycles

David E. Narvaez

Some Multicolor Ramsey Numbers Involving Cycles

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Stanisław P. Radziszowski, Advisor Department of Computer Science Rochester Institute of Technology	Date
Dr. Edith Hemaspaandra, Reader Department of Computer Science Rochester Institute of Technology	Date
Dr. Ivona Bezáková, Observer Department of Computer Science Rochester Institute of Technology	Date

Some Multicolor Ramsey Numbers Involving Cycles

by

David E. Narváez

THESIS

Presented to the Department of Computer Science
of the Golisano College of Computing and Information Sciences
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Computer Science

Rochester Institute of Technology

May 2015

Abstract

Some Multicolor Ramsey Numbers Involving Cycles

David E. Narváez, M.S.
Rochester Institute of Technology, 2015

Supervisor: Dr. Stanisław P. Radziszowski

Establishing the values of Ramsey numbers is, in general, a difficult task from the computational point of view. Over the years, researchers have developed methods to tackle this problem exhaustively in ways that require intensive computations. These methods are often backed by theoretical results that allow us to cut the search space down to a size that is within the limits of current computing capacity.

This thesis focuses on developing algorithms and applying them to generate Ramsey colorings avoiding cycles. It adds to a recent trend of interest in this particular area of finite Ramsey theory. Our main contributions are the enumeration of all $(C_5, C_5, C_5; n)$ Ramsey colorings and the study of the Ramsey numbers $R(C_4, C_4, K_4)$ and $R_4(C_5)$.

Table of Contents

Chapter 1

Introduction

Finite Ramsey theory studies guaranteed properties of partitions of graphs with a finite number of vertices. Unlike infinite Ramsey theory, where most of the important results are presented in the form of existential statements, in finite Ramsey theory we are interested in specific values known as Ramsey numbers, which are the orders of related extremal graphs. Because one of our goals is to compute these numbers, this field is also known as computational Ramsey theory. Partitioning the set of edges of a graph can also be regarded as "coloring" the edges. Ramsey numbers concerning partitions into k sets are also called k-color Ramsey numbers and are called multicolor Ramsey numbers for $k > 2$. As a concrete example, it is known that any complete graph on 25 or more vertices, when partitioned into two subgraphs, will yield a partition where either the first graph contains a complete graph on 4 vertices, or the second graph contains a complete graph on 5 vertices.

During the 1930's and throughout the 1970's, many mathematicians, most notably Paul Erdős, addressed the finite cases, designing techniques to compute either the exact values or upper and lower bounds of Ramsey numbers for many families of graphs. At this stage, the results were still either

analytical or done by hand. With the advent of computers, researchers were able to use these to find exact values for larger cases. This also helped improve on the analytical results and find better bounds. For instance, the concrete example mentioned above is due to Brendan McKay and Stanisław Radziszowski [MR95], who developed novel algorithms for efficiently traversing trees of graphs and use them for related problems. Today, research in computational Ramsey theory lies on the boundary between pure discrete mathematics and computer science, with one often improving on the results of the other.

To illustrate the difficulty of the task of determining the exact value of certain Ramsey numbers, it is useful to consider the following citation from Joel Spencer [Spe94]

> "Erdős asks us to imagine an alien force, vastly more powerful than us, landing on Earth and demanding the value of $R(5,5)$ or they will destroy our planet. In that case, he claims, we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose, instead, that they ask for $R(6,6)$. In that case, he believes, we should attempt to destroy the aliens."

It might be that the computational difficulty of finding Ramsey numbers lies inherently in the large number of cases to check: even for graphs with as few as 12 vertices, the search spaces could be large enough to require distributed computations. It might also lie in our lack of understanding of the characterization of Ramsey critical graphs. In any case, finding Ramsey numbers given our current knowledge and computing power requires us to start

from an insightful theoretical result that will drastically cut down the number of cases to analyze. After this initial step, we are then able to create programs that will perform suitable computations.

The structure of this thesis is as follows: Chapter 2 introduces the theory behind Ramsey numbers and related fields, Chapter 3 deals specifically with monochromatic cycles in graph colorings. Chapters 4 and 5 use the algorithms introduced in Chapter 3 to explain how we generated (C_5, C_5, C_5) colorings and to describe our strategy to approach the problem of finding $R(C_4, C_4, K_4)$, respectively. Finally, Chapter 6 lists directions for future work.

Chapter 2

Background

This chapter introduces the notation that will be used throughout this thesis. While the central topic of this work is Ramsey theory, some other fields of discrete mathematics are used to support the many tools developed in order to solve the problem in hand. A good reference for more in-depth study of many of the concepts described in this chapter is Bollobás book on Extremal Graph Theory [Bol04].

2.1 General Graph Theory

For a graph $G = (V, E)$, V and E are the set of vertices and edges, respectively. Define the functions $n(G) = |V|$ and $e(G) = |E|$. We may also use V and E as functions of a graph, where $V(G)$ (respectively, $E(G)$) denotes the set of vertices (respectively, edges) of a graph G. The number $n(G)$ is also referred to as the *order* of G and we denote by $\mathbb{G}_n$ the set of graphs of order n. The neighborhood $\mathcal{N}(v, G)$ of a vertex v in a graph G is the set of vertices v' such that $\{v, v'\} \in E$, and the degree of that vertex is $d(v, G) = |\mathcal{N}(v, G)|$. We denote by $\Delta(G)$, $\delta(G)$ and $\overline{d}(G)$ the maximum, minimum and average degree of the vertices in G.

K_n is the complete graph on n vertices, $K_n - e$ is a K_n without one edge and $K_{i,j}$ is a complete i by j bipartite graph. C_n is a cycle of length n and P_n is a path on n vertices. As it is customary, (see, e.g. [Rad94]) we define the binary operation $H \cup G$ to be the disjoint union of two graphs H and G and the operation $H + G$ to be $H \cup G$ with the addition of all edges between H and G. nG stands for $\underbrace{G \cup G \cup \ldots \cup G}_{n\text{-times}}$.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if and only if there is a bijection $\phi : V_1 \to V_2$ such that $\{\phi(v_1), \phi(v_2)\} \in E_2 \Leftrightarrow \{v_1, v_2\} \in E_1$ for all $v_1, v_2 \in V_1$. Intuitively, this can be interpreted as two graphs that have the same "shape" but possibly different vertices. An automorphism is an isomorphism from a graph to itself, i.e., a permutation of the vertex labels such that the set of edges remains the same. Because the composition of two automorphisms is also an automorphism, the automorphisms of a graph form a group:

Definition 2.1 (Automorphism Group)**.** $\mathrm{Aut}(G)$ is the group of all automorphisms of a graph G under the composition operation.

For simplicity, we extend the notation for permutations of elements to elements to permutations of sets to sets. For a permutation $\phi : A \to B$ and a set $S \in 2^A$, the notation $\phi(S)$ is defined as the set $\{\phi(s) \,|\, s \in S\}$.

Given a subset V' of the vertices of a graph $G = (V, E)$, the graph induced by V' in G is the graph $G' = (V', E')$ where E' is the set of edges in E with both endpoints in V', formally $E' = \{\{v_i, v_j\} \in E \mid v_i, v_j \in V'\}$. Let

$G[V']$ denote the graph induced by V' in G. We say a graph G contains a subgraph isomorphic to $H = (V_H, E_H)$ (or, more informally, contains a copy of a graph H) when there is an injective function $\phi : V_H \to V$ such that $\{v_i, v_j\} \in E_H \Rightarrow \{\phi(v_i), \phi(v_i)\} \in E$. Notice that the image of ϕ defines a subset V' in V, but the graph induced by V' in G is not necessarily isomorphic to H. The graph $G - v$ is the result of removing a vertex $v \in V(G)$ from a graph G, together with all edges $\{v, w\} \in E(G)$ or, equivalently, $G - v$ denotes the graph $G[V(G) \setminus \{v\}]$. When two graphs $H = (V, E_H)$ and $G = (V, E_G)$ share the same set of vertices, it makes sense to use the notation $H \subseteq G$ to indicate H is a subgraph of G if one takes these graphs to be collections of sets $\{u, v\}$ with $u, v \in V$.

A subgraph of G that is isomorphic to K_k is called a *clique* of order k. The complement of a graph G, denoted $\overline{G}$, is the graph formed by the edges "missing" in G, i.e. $\overline{G} = (V, \{\{v_1, v_2\} | v_1, v_2 \in V, v_1 \neq v_2\} \setminus E)$. A clique of size k in the complement of a graph G is called an *independent set* of G. The independence number $\alpha(G)$ of a graph G is the order of the largest independent set in G. The Turán number of a graph G for graphs of order n, denoted $\tau(G, n)$, is the maximum number of edges among graphs of order n that contain no G as a subgraph.

2.2 Ramsey Theory

While studying graphs, one is often concerned with what are the maximal or minimal graphs that satisfy certain property. These kinds of questions

belong to the field of extremal graph theory. One property of interest is the presence or absence of a specific subgraph and this is the primary focus of Ramsey theory.

Definition 2.2 (Ramsey Numbers)**.** Given k graphs $G_1, G_2, \ldots, G_k$, the multicolor Ramsey number $R(G_1, G_2, \ldots, G_k)$ is the smallest integer N such that, in every assignment of k colors to the edges of a complete graph of order at least N, there is a color $1 \leq i \leq k$ such that the graph formed by the edges of the i-th color contains a copy of G_i.

For brevity, when the graphs G_i are all equal to a graph G, we write this number as $R_k(G)$. Ramsey's theorem [Ram30] states that these numbers exist and are computable. The dynamic survey by Radziszowski [Rad94] is the standard reference for an extensive overview of the advances in this field.

A k-coloring of the set of edges of a graph $G = (V, E)$ is a function $C : E \to \{1, 2, \ldots, k\}$. A $(G_1, G_2, \ldots, G_k; n)$ coloring is a k-coloring of a complete graph of order n such that the i-th color contains no G_i as a subgraph. To simplify our notation, we will refer to $\left(\underbrace{G, G, \ldots, G}_{k}; n\right)$ colorings as G-free k-colorings of order n. It is easy to see that any graph on n vertices can be regarded as a complete graph on n vertices whose edges are painted in two colors, the second one being "transparent," thus a $(G_1, G_2; n)$ coloring can also be called a $(G_1, G_2; n)$ graph. If G is a graph coloring using k colors, the notation $G[i]$ denotes the graph induced by the i-th color of G for $1 \leq i \leq k$.

Notice that the Ramsey number $R(G_1, G_2, \ldots, G_k)$ can then be defined as the smallest integer N such that no $(G_1, G_2, \ldots, G_k; N)$ coloring exists.

A typical example of Ramsey theory is the following problem, which has become part of mathematical folklore [Gar97]:

Problem 2.1 (Problem E 1321, The American Mathematical Monthly, June - July, 1958)**.** *Prove that at a gathering of any six people, some three of them are either mutual acquaintances or complete stranger to each other.*

This can be formulated in terms of graph. This formulation appeared as a problem in the William Lowell Putnam mathematical competition:

Problem 2.2 (William Lowell Putnam Mathematical Competition, 1953, Question A2)**.** *The complete graph with 6 points and 15 edges has each edge colored red or blue. Show that we can find 3 points such that the 3 edges joining them are the same color.*

Solving this problem is equivalent to proving that the Ramsey number $R(K_3, K_3)$ is at most 6. To prove that $R(K_3, K_3) = 6$, the graph in Figure 2.1 shows the well-known 2-coloring of the complete graph on 5 vertices that contains no monochromatic triangle.

The reader can refer to Graham's book on Ramsey theory [Gra90] for more information on this subject. A recent publication by Soifer [Soi11] also provides a fairly complete overview of the field from its beginnings to recent advances and trends.

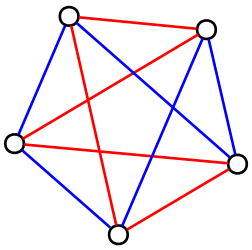


Figure 2.1: Witness graph for $R(K_3, K_3) > 5$

2.3 `nauty`

It is not known whether or not there exists a polynomial-time algorithm to test whether two graphs are isomorphic or not [GJ79]. On the other hand, it is easy to see that this problem is crucial for graph theory research. The development of good algorithms to solve this problem efficiently is then of great importance and has received much attention in the past decades. ***no aut****omorphisms,* ***y****es?* [MP14] is a collection of software tools and a C library to deal with graph isomorphisms and automorphism groups. It was first published by Brendan McKay in 1981 and has been used in several other approaches to establish the exact values of Ramsey numbers [MR95].

At the heart of the automorphism detection in `nauty` is the concept of a canonical labeling. As defined in [Pip08], the canonical labeling of a graph G is a graph G', isomorphic to G, representing the whole isomorphism class of G. To test whether two graphs G_1 and G_2 are isomorphic, one can calculate

canonical labelings G'_1 and G'_2 and then compare these graphs for equality. `nauty`'s strategy to find the canonical labeling of a graph is to search for a permutation of the labels of the input graph by traversing a search tree of all possible labelings. The efficient traversal of such a large search space is based on several heuristics that prune parts of the search space. As is often the case with heuristics, these work particularly well for certain types of graphs and may perform poorly for other types of graphs. Other graph isomorphism algorithms that are based on exploring a search tree are found in *saucy* [DLSM04] and *Bliss* [JK07].

Traces [Pip08] is another algorithm that is distributed with `nauty` which improves some of the original heuristics. In particular, it removes some of `nauty`'s original sensitivity to the *node invariant* used to prune the search tree. Node invariants are important components in the heuristics in `nauty` that can make a great difference when used correctly, but the mastering of these requires advanced knowledge of the internals of the algorithms. *Traces* then tries to remove the burden of the know-how from the user by avoiding such a tight dependency between the pruning heuristics and the node invariants.

The concept of canonical labelings is also useful when dealing with a large number of graphs because it allows us to "deduplicate" graph databases by keeping only one representative of the isomorphism classes in it. This feature was critical to keep our dataset of $(C_5, C_5, C_5; n)$ colorings within manageable size, as explained in Chapter 4. This deduplication was achieved through

the `shortg` and `shortmc`[1] tools that are part of `nauty`.

While our use of `nauty` for the enumeration of $(C_5, C_5, C_5; n)$ colorings was limited to external tooling, we used `nauty` extensively as a library in our approach to establishing the value of $R(C_4, C_4, K_4)$. For this problem, the automorphism groups are essential to many of the definitions, as explained in Chapter 5.

[1] `shortmc` is part of a yet unpublished extension to `nauty` developed by Brendan MacKay that deals with edge colorings in graphs.

Chapter 3

Colorings Avoiding Cycles

In this chapter we focus on the task of coloring the set of edges of a graph with k colors so that the graphs induced by the each color contain no cycles of a given length. We then say a graph can be *split* into k colors avoiding C_m if this task can be achieved. Cycle-free Ramsey colorings have received much attention in recent years, as shown by the many results compiled in Radziszowski's [Rad11] survey on this particular area of Ramsey theory.

We employ two strategies to generate k-colorings that avoid cycles in each of the k colors: direct coloring of edges (Section 3.2) and extending good k-colorings by adding a vertex (Section 3.3). In both strategies, a fundamental piece is the cycle detection algorithm described in the next section.

3.1 Cycle Detection

The standard algorithm to detect cycles (possibly of a fixed length m) in a graph G simply searches through all possible paths in G, maintaining a queue of paths to check. Whenever the last vertex of the path matches the first vertex (and optionally, the length of the path is $m+1$ in the cases where we are interested in cycles of a specific length), a cycle is detected. In the "decision"

version of this algorithm, this will be enough to return with a positive answer. In the "search" version of the algorithm, the cycle detected is removed from the queue and added to the list of cycles in the solution.

The naïve implementation of this algorithm will naturally find all $2n$ directed paths in a cycle of length n, which increases the processing time. To improve over this implementation, an idea from Floyd's *tortoise and hare* algorithm [Flo67] can be adopted. The key is to assume the first vertex in each path is the vertex with the largest label in the path. This reduces duplicates to exactly two per cycle, which is a significant speedup over the naïve implementation when the number of cycles is large.

3.2 Coloring the Edges

One way to generate colorings that avoid cycles of length m is to list all edges that form cycles of length m in a graph and finding combinations of these edges such that every set of edges in an m-cycle is intersected at least once and at most $m-1$ times by each color. Notice that, if $\mathcal{E}$ is the set of edges that participate in any cycle of length m in a graph G, then we would need to perform the intersection check described above for every partition $\{\mathcal{E}_1, \mathcal{E}_2\}$ of $\mathcal{E}$. Yet, a significant number of checks can be avoided by using the concept of the Turán number: since we know that a valid partition of $\mathcal{E}$ must satisfy $|\mathcal{E}_1| \leq \tau(C_k, n)$ and $|\mathcal{E}_2| \leq \tau(C_k, n)$ and because $|\mathcal{E}_1| + |\mathcal{E}_2| = |\mathcal{E}|$, we must have

$$|\mathcal{E}| - \tau(C_k, n) \leq |\mathcal{E}_1| \leq \tau(C_k, n) \tag{3.1}$$

In the case of odd cycles, the Turán number $\tau\left(C_{2l+1}, n\right)$ is known to be $\left\lfloor \frac{n^2}{4} \right\rfloor$, so 3.1 can be applied as

$$|\mathcal{E}| - \left\lfloor \frac{n^2}{4} \right\rfloor \leq |\mathcal{E}_1| \leq \left\lfloor \frac{n^2}{4} \right\rfloor .$$

3.2.1 SAT Formulation

An alternative way to formulate the 2-coloring problem is to assign a Boolean variable to every edge in the graph and calculate the set $\mathcal{C}$ of all cycles of length m in it. Then, for every cycle C in $\mathcal{C}$, a Boolean expression

$$\bigvee_{i=1}^{m} f\left(C, i\right) \wedge \bigvee_{i=1}^{m} \overline{f\left(C, i\right)}$$

is formulated, where $f\left(C, i\right)$ is the variable assigned to the i-th edge of C. A truth assignment of the variables can then be trivially mapped to a proper coloring of the complement of the given graph.

We implemented this alternative solution using the PicoSAT [Bie08] library, version 959. In practice, this implementation is faster than the direct enumeration of all possible edge colorings for certain graphs, but not all of them.

3.2.2 Constraint Programming Formulation

Similar to the SAT formulation, one can express the forbidden substructures in colorings through constraint programming. The advantage of this approach over the SAT formulation is that it is easy to express constraints for k-coloring problems for any k, not only $k = 2$. A k-coloring for a graph on

n vertices can be represented by n^2 variables $e_{i,j}$ that can take integer values between 0 and k. Here, 0 means the edge corresponding to that variable is not in the graph. Then a constraint satisfaction solver can assign integers to these variables subject to constraints like

$$\left(\bigvee_{i=0}^{m-3}\left(e_{v_{\sigma(i)},v_{\sigma(i+1)}} \neq e_{v_{\sigma(i+1)},v_{\sigma(i+2)}}\right)\right) \vee \left(e_{v_{\sigma(m-2)},v_{\sigma(m-1)}} \neq e_{v_{\sigma(k-1)},v_{\sigma(0)}}\right)$$

for every m-subset $\{v_0, v_1, \ldots, v_{m-1}\}$ of vertices and every permutation σ of the set $\{0, 1, \ldots, k-1\}$. These constraints guarantee that, in any permutation of m-subsets of vertices, at least two consecutive edges do not have the same color. We implemented this formulation using the Ben-Gurion University Equi-propagation Encoder (BEE) [MC12].

3.3 Extending Good Colorings

If we have the list of all C_m-free k-colorings of order n, we can generate all C_m-free k-colorings of order $n+1$ by adding a vertex v and joining it with all the n previous vertices, then coloring these n new edges with k colors so that no cycle of length k is formed in any color. Specifically, for every candidate coloring C of the n new edges, we require that for every pair of vertices u, w that are endpoints of a monochromatic P_{m-1} in G with all edges colored with the i-th color, we have $C(\{v, u\}) \neq i$ or $C(\{v, w\}) \neq i$.

Despite the fact that there are k^n candidate colorings for this process, it is, in practice, significantly faster than generating colorings by direct assignment of colors to edges when k is small.

Chapter 4

$(C_5, C_5, C_5; n)$ Colorings

This chapter details our search for all $(C_5, C_5, C_5; n)$ colorings. It is known that $R(C_5, C_5, C_5) = 17$, so this search would only be carried out up to graphs of order 16, which is within the reach of current computing power.

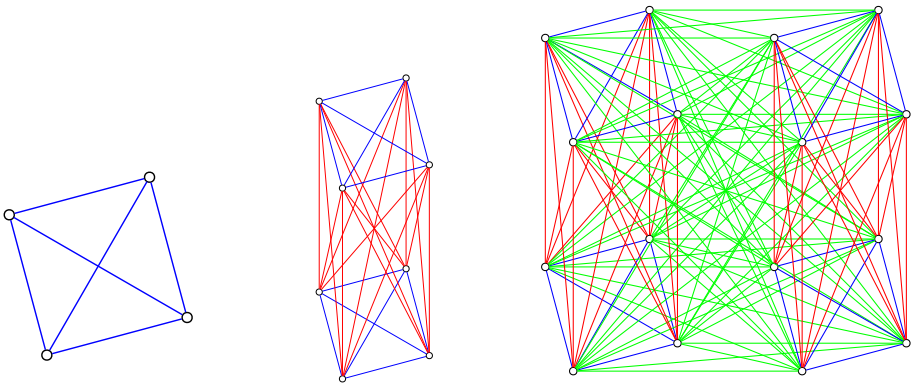
The lower bound $17 \leq R_3(C_5)$ is derived from the fact that the following *blow-up* construction [JMW13] yields a C_n-free k-coloring of order $2^{k-1}(n-1)$ for any odd n[1]:

1. Given a fixed n and k colors, construct a complete graph on $n-1$ vertices using the first color.

2. For $c := 2, 3, \ldots, k-1$, duplicate the current graph and color all edges that join vertices of the two copies using the c-th color.

In [YR92b], Yuansheng and Rowlinson showed that $R_3(C_5) = 17$ by executing an exhaustive, computer-aided search for colorings of the complement of C_5-*critical* graphs of order 17. In this context, a *critical* graph on n vertices stands for a graph of order n that is C_5-free and adding any edge to it

[1]The use of this construction in the context of Ramsey numbers of cycles of odd length is generally attributed to Bondy and Erdős [BE73].



(a) K_4 (b) $2K_4 + K_{4,4}$ (c) $4K_4 + 2K_{4,4} + K_{8,8}$

Figure 4.1: *Blow-up* construction for C_5 and $k = 3$

completes a cycle of length 5. To further reduce the search space, the authors show that the only graphs that need to be checked are those that:

1. Have at least 46 edges (the total number of edges, divided by 3), since it is possible to assume, without loss of generality, that the first color of the $(C_5, C_5, C_5; 17)$ coloring that will be generated is the one with the largest amount of edges.

2. Have no independent set of order 9, since $R_2(C_5) = 9$ [CS71].[2]

There are 52 graphs that meet these criteria, and there is no way to color the complement of these graphs using two colors to produce a $(C_5, C_5, C_5; 17)$ graph, which establishes that $R_3(C_5) = 17$.

[2]This is easy to verify by combining a cycle detection algorithm with `nauty`'s `complg` tool to complement graphs. Witness colorings for $R_2(C_5, C_5) = 9$ are shown in Figure 4.2

At this point it is important to distinguish between the exhaustive search done in [YR92b] and the exhaustive listing of $(C_5, C_5, C_5; n)$ colorings we present in this report: It is possible to generate all critical graphs without listing all $(C_5, K_9; n)$ graphs explicitly, as shown in [YR92a].

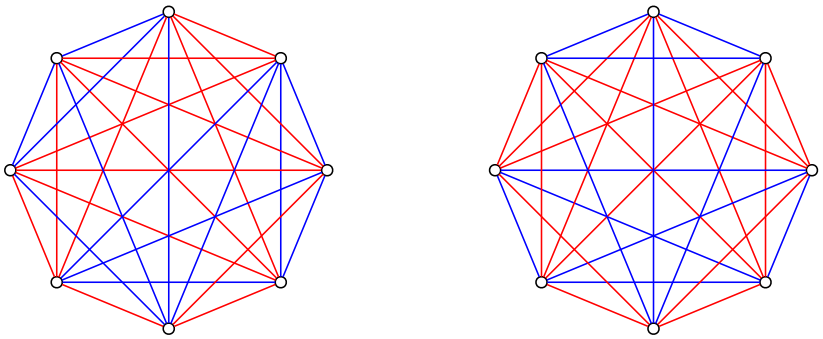


Figure 4.2: Witness colorings for $R(C_5, C_5) = 9$

We can apply the concepts of Sections 3.2 and 3.3 to generate C_5-free 3-colorings for $1 \leq n \leq 16$. For instance, we can take a C_5-free graph and color the complement of this graph with two colors in a way that avoids cycles of length 5. Figure 4.3 shows an example of two $(C_5, C_5, C_5; 8)$ colorings generated using this method. Also, with the complete set of $(C_5, C_5, C_5; n)$ colorings for some n, we can generate all $(C_5, C_5, C_5; n+1)$ colorings by adding a vertex and joining it with n edges colored such that no new cycles are formed.

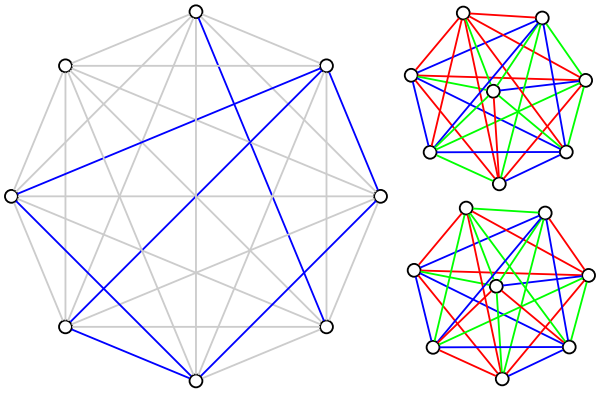


Figure 4.3: Two of the 12,586 ways to split the complement of this graph of order 8 such that there are no cycles of length 5.

4.1 $R_3(C_3)$ vs $R_3(C_5)$

Because $R_3(C_3) = R_3(C_5) = 17$, it is interesting to compare the behavior of the number of colorings relevant to each of these Ramsey numbers. By looking at Table 4.4 we can see several differences between the two families of colorings. An immediate observation one can make is that there are many more $(C_5, C_5, C_5; n)$ colorings than $(C_3, C_3, C_3; n)$ colorings for any given n. Furthermore, despite the fact that the behavior of the number of colorings for both types is that it increases monotonically up to some value of n, then starts decreasing monotonically, the value of n that yields the maximum number of colorings differs between the colorings for C_3 and C_5.

4.2 Open Science Grid

$(C_5, C_5, C_5; n)$ colorings for $13 \leq n \leq 16$, which are shown in Table 4.4, were generated with the help of the Open Science Grid. To adapt to the

Number of Vertices	$(C_3, C_3, C_3; n)$	$(C_5, C_5, C_5; n)$
6	330	2 349
7	3 829	54 927
8	50 391	679 876
9	500 023	3 713 104
10	2 646 593	14 092 138
11	4 821 244	43 945 253
12	1 929 792	140 033 320
13	78 892	448 105 921
14	115	1 142 773 713
15	2	1 844 045 362
16	2	1 701 746 176

Table 4.4: Counts of weakly isomorphic colorings for $(C_3, C_3, C_3; n)$ and $(C_5, C_5, C_5; n)$. The number of $(C_3, C_3, C_3; n)$ colorings were calculated by Radziszowski [Rad14] and independently verified by the author.

workflow of this project, the monolithic file containing all $(C_5, C_5, C_5; 12)$ colorings was split and each part was processed separately by several computing nodes. The output of these processes was then merged together in a hierarchical way using `nauty`'s `shortmc` tool. The orchestration of this workflow was done through the Pegasus project [DSS+05].

4.3 Description of the Colorings

The large number of colorings obtained as a result of this process makes it challenging to provide a succinct description of the dataset. We decided to provide a description in terms of the number of K_4 subgraphs found in each coloring since these graphs are the building blocks of the *blow-up* construction explained earlier in this chapter. Table 4.5 shows the resulting classification

K_4 Count			Number of Colorings
Color 1	Color 2	Color 3	
4	0	0	1 701 423 231
5	0	0	249 540
2	2	0	73 405

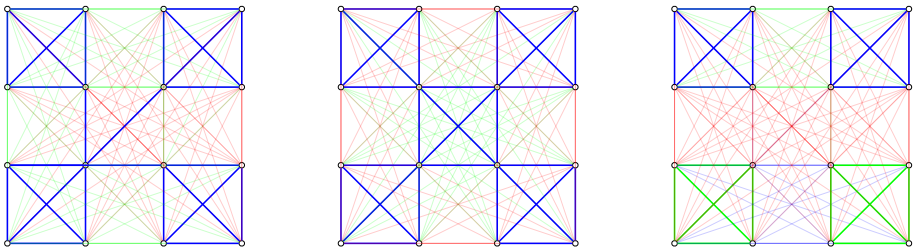
Table 4.5: Tally of $(C_5, C_5, C_5; 16)$ colorings by number of K_4's per color

of the colorings.

Figure 4.6 illustrates each type of coloring described in Table 4.5. All colorings with five K_4's in one color have exactly 30 edges in that color. The subgraph induced by that color corresponds to four K_4's mutually attached through an additional K_4. We will call this graph $4K_4^{\times}$. In the case of colorings with two K_4's in the first color and two K_4's in the second color, the vertex sets forming these K_4's are disjoint and their union is the whole set of vertices. For colorings with four vertex-disjoint K_4's in the first color, there are 29 graphs induced by that color. We will call this set $\mathcal{K}$. An easy way to describe some of the graphs in $\mathcal{K}$ is to notice that the complement of any graph G on 16 vertices with $4K_4 \subseteq G$ is a subgraph of $\overline{4K_4}$, which has a C_5-free 2-coloring. Since both $4K_4$ and $4K_4^{\times}$ are C_5-free, we have the following lemma:

Lemma 4.1. *If G is a graph of order 16 with $4K_4 \subseteq G \subset 4K_4^{\times}$, then there is a $(C_5, C_5, C_5; 16)$ coloring that induces a monochromatic G in some color.* □

From Lemma 4.1 we can deduce that 10 of the 29 graphs in question are four K_4's mutually attached by any of the 10 graphs on 4 vertices that are not K_4. A different strategy to classify these graphs systematically is based



(a) Four K_4's

(b) Five K_4's

(c) Two K_4's in one color and two K_4's in another color

Figure 4.6: Sample $(C_5, C_5, C_5; 16)$ colorings according to our classification.

on the following transformation:

Definition 4.1 (*Shrink* of G)**.** Let $G = (V, E)$ be a graph of order 16 such that $4K_4 \subseteq G$ and let $\mathcal{P} = \{P_0, P_1, P_2, P_3\}$ be a partition of V such that $G[P_i]$ is a K_4. The graph $S(G) = (\mathcal{P}, E')$ is defined such that $\{P_i, P_j\} \in E'$ if and only if there exists a pair of vertices $u', v' \in V(G)$ such that $u' \in P_i$, $v' \in P_j$ and $\{u', v'\} \in E(G)$.

Intuitively, we shrink the four K_4's in G and represent the connections between the original vertex sets as edges in the new graph. Notice that, if $G \in \mathcal{K}$, for every edge $\{P_i, P_j\} \in E(S(G))$ there is exactly one pair of vertices $u', v' \in V(G)$ that satisfies the definition above because, if there was another pair $\{u'', v''\}$ such that $u'' \in P_i$, $v'' \in P_j$ and $\{u'', v''\} \in E(G)$ then G would have a cycle of length 5.

An immediate consequence of Lemma 4.1 is that every graph on 4 vertices is isomorphic to the shrink of at least one graph in $\mathcal{K}$. On the other

hand, counting the number of graphs $G \in \mathcal{K}$ whose shrink is isomorphic to H seems to require a case-by-case analysis of all the graphs on 4 vertices.

4.4 Bounds on $R_4(C_5)$

The lower bound $33 \leq R_4(C_5)$ comes from the fact that we can generate a C_5-free 4-coloring on 32 vertices by using the *blow-up* construction explained before. For the upper bound, the formula in [Rad94] is $R_k(C_5) < \frac{\sqrt{18^k k!}}{10}$, which yields $R_4(C_5) \leq 158$. This inequality, which originally appears in [Li09], is derived from the inequality $R_k(C_5) - 1 < 1 + \sqrt{18k}\,(R_{k-1}(C_5) - 1)$ which is stronger, but can only be used if the value of $R_{k-1}(C_5)$ is known. Since in our case we know $R_3(C_5) = 17$, this last inequality implies $R_4(C_5) \leq 137$.

As in the previous section, we analyze what the number of K_4's in a $(C_5, C_5, C_5, C_5; 32)$ coloring could be. The blowup construction already yields a coloring with eight K_4's in one color. Furthermore, the complement of a graph obtained from eight copies of K_4's mutually attached through a C_5-free graph on 8 vertices is a subgraph of the complement of $8K_4$, so it has a C_5-free 3-coloring. Since any C_5-free graph on 8 vertices has, at most, two K_4's, this shows that there are colorings with ten K_4's in one color.

Chapter 5

$R(C_4, C_4, K_4)$

This chapter focuses on the problem of establishing the value of the Ramsey number $R(C_4, C_4, K_4)$. Before this work, this value was known to be at least 20 [DD11] and at most 22 [XSR09]. As in Section 3.3, our strategy was to generate $(C_4, C_4, K_4; 20)$ colorings by extending $(C_4, C_4, K_4; 19)$ colorings that have a particular structure. This structure guarantees that failing to generate any $(C_4, C_4, K_4; 20)$ colorings by exhaustively trying to extend all $(C_4, C_4, K_4; 19)$ colorings obtained would imply that there are no $(C_4, C_4, K_4; 20)$ colorings at all, proving that $R(C_4, C_4, K_4) = 20$.

5.1 Previous Bounds

The lower bound was established by taking a graph on 18 vertices that contains no cycle of length 4 and coloring the complement of this graph avoiding cycles of length 4 in the second color and the complete graph of order 4 in the third color. The original graph on 18 vertices was described first in 1989 by Clapham, Flockhart and Sheehan in [CFS89]. The proof for the upper bound is also based on [CFS89], since it uses the values for $\tau(C_4, 22)$ to prove that the complement of any K_4-free graph of order 22 cannot be colored using

two colors avoiding C_4 in both of them. The reasoning is very similar to the one used in the forthcoming Theorem 5.1.

Later in 2011, Dybizbański and Dzido showed (Figure 4 of [DD11]) a graph coloring on 19 vertices that contains no cycles of length 4 in the first two colors and no complete graph of order 4 in the third color. The Ramsey number $R(C_4, C_4, K_4)$ must then be at least 20.

One graph mentioned in [CFS89] that is of particular importance for Theorem 5.1 was originally described in the context of cycles of length 4 by Erdős, Rényi and Sós in 1966 [ERS66]. The vertices in this graph correspond to points in the projective plane $PG(2, k)$, for a prime power k. There are $k^2 + k + 1$ of these points. An edge between vertices (points) A and B (in our case, $A \neq B$ to avoid loops) exists if and only if A is in the dual line of B. From this definition and the fact that there is a single line passing through any two points in a projective plane, one can deduce that if edges AC, BC, AC' and BC' exist in this graph, then the dual line of C is the dual line of C' so $C = C'$. In particular, this means that there is no cycle of length 4 in this graph. For $k = 2^2$, this graph of order 21 is shown in Figure 5.1. By removing one vertex of degree 4 from this graph, one obtains a graph of order 20 with no cycle of length 4. These two graphs are unique witnesses for $\tau(C_4, 21)$ and $\tau(C_4, 20)$, respectively [CFS89].

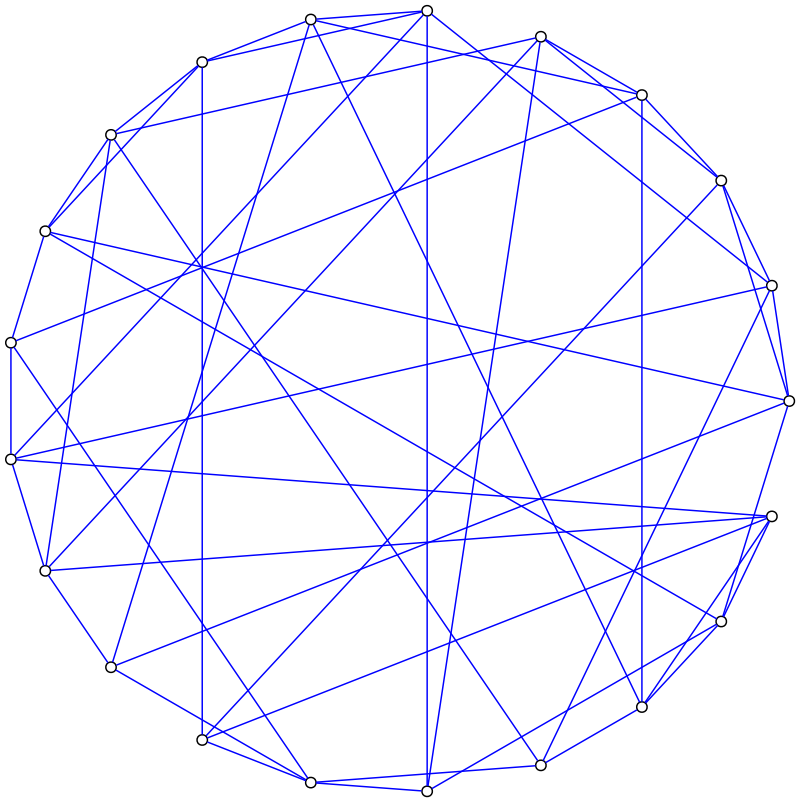


Figure 5.1: A graph of order 21 with no cycle of length 4

5.2 The Structure of the Third Color

Our strategy is based on the following theorem about graphs induced by the third color of any $(C_4, C_4, K_4; 20)$ coloring:

Theorem 5.1. *Let G be a $(C_4, C_4, K_4; 20)$ coloring. Then there are two non-adjacent vertices in $G[3]$ whose degree is 11.* □

Proof. As in the solution of Theorem 5.1 in [XSR09], we note that $\Delta(G[3]) \leq 11$: Indeed, because $R(C_4, C_4) = 12$ [Rad94], the complement of the graph induced by the neighborhood of any vertex in $G[3]$ can be of order at most 11. Let D_{11} be the set of vertices in $G[3]$ whose degree is 11. We claim that $|D_{11}| \geq 4$. Then, take 4 vertices from D_{11}. Because $G[3]$ is K_4-free, these vertices do not induce a complete graph in $G[3]$, so there must be 2 vertices that are non-adjacent.

To prove our claim, assume by contradiction, that $|D_{11}| \leq 3$. It is easy to see that Lemma 4.4 in [CFS89] guarantees the existence of a vertex of degree 4 in $G[3]$, so

$$e(G[3]) \leq \left\lfloor \frac{4 + |D_{11}| \times 11 + (19 - |D_{11}|) \times 10}{2} \right\rfloor = \left\lfloor \frac{194 + |D_{11}|}{2} \right\rfloor \leq 98$$

On the other hand, $\tau(C_4, 20) = 46$ [CFS89] and $G[1]$ and $G[2]$ have no C_4, so

$$e(G[3]) = \binom{20}{2} - e(G[1]) - e(G[2]) \geq 190 - 2 \times 46 = 98$$

so $e(G[3]) = 98$ and $G[1]$ is isomorphic to the unique witness graph for $\tau(C_4, 20) = 46$, but the complement of this graph cannot be colored with

2 colors such that the first color contains no C_4 and the second color contains no K_4 [Rad14], a contradiction. ■

In view of Theorem 5.1, we can choose two non-adjacent vertices u and v of degree 11 in the third color of a $(C_4, C_4, K_4; 20)$ coloring and remove u. We are then left with a $(C_4, C_4, K_4; 19)$ coloring G where the neighborhood of vertex v in $G[3]$ induces a graph X of order 11 with no K_3 and the rest of the vertices (not adjacent to v) induce a graph Y of order 7 with no K_4. Both X and Y have the additional property that their complement can be split into 2 colors avoiding C_4. Furthermore, while the edges between X and Y will be determined by this extension process, the graph induced by the vertex v and the vertices of Y depends only on Y. This means Y also has the property that the complement of the graph obtained by adding a disconnected vertex to it can also be split into 2 colors avoiding C_4. We can now "reconstruct" G from its pieces v, X and Y by choosing graphs X and Y from the sets $\mathcal{G}_{11}$ and $\mathcal{G}_7$ defined in Table 5.2 and defining the set of edges between X and Y so that no K_4 is created and the complement of the resulting graph can be split into 2 colors avoiding C_4.

$\mathcal{G}_7$ can be easily generated by combining `nauty`'s `geng` and `pickg` tools to list all non-isomorphic graphs on 8 vertices and pick those with minimum degree 0, then discarding those that either contain K_4 or whose complement cannot be split into 2 colors avoiding C_4; and finally dropping one vertex of degree 0 in each one of them. One could, in theory, generate $\mathcal{G}_{11}$ using a similar strategy, but in practice there are too many non-isomorphic graphs of order

Set	Description	Size
$\mathcal{G}_7$	K_4-free graphs of order 7 to which we can add one vertex and the complement of the resulting graph will have a 2-coloring avoiding C_4	587
$\mathcal{G}_{11}$	K_3-free graphs of order 11 whose complement can be split into 2 colors avoiding C_4	575
$\mathcal{G}_{13}$	All graphs G induced by the third color of all $(C_4, C_4, K_4; 13)$ colorings with $\Delta(G) = 11$	243997

Table 5.2: Sets of graphs used to reconstruct a $(C_4, C_4, K_4; 19)$ coloring

11. Instead, one can exploit the fact that the set of all $(K_3, C_6; 11)$ colorings[1] is a superset of $\mathcal{G}_{11}$, because $R(C_4, C_4) = 6$.

Figure 5.3 illustrates one graph of 19 vertices that can be obtained via this process. The dashed edges join the vertex v to every vertex in X. The dotted edges join vertices of Y with feasible neighborhoods of X.

5.3 Feasible Neighborhoods

Let y be a vertex in Y. The graph G' induced in $G[3]$ by v, y and the vertices in X is a K_4-free graph of order 13 whose complement can be split into 2 colors avoiding C_4, so it must be in the set of graphs obtained by extracting the graph induced by the third color of every $(C_4, C_4, K_4; 13)$ coloring. Conversely, for every graph H induced by the third color of a $(C_4, C_4, K_4; 13)$

[1]This set is available at `http://cs.anu.edu.au/people/bdm/data/ramsey.html`

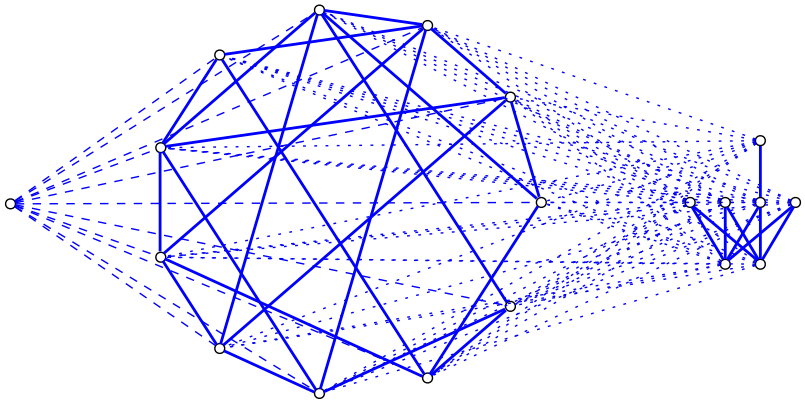


Figure 5.3: A graph produced by the extension process

coloring such that $\Delta(H) = 11$, its vertices can be mapped to v, y and the vertices of X by selecting a vertex of degree 11 to be v, its neighborhood to be the set of vertices of X and the remaining vertex to be y. After this mapping is found, one additional property about X can be noted: the set of vertices $\mathcal{N}(y, H)$ defines, in X, a set of vertices to which a vertex of Y can be joined.

To exploit this mapping, we define an additional set $\mathcal{G}_{13}$ of graphs of order 13 and maximum degree 11 that are induced by the third color of a $(C_4, C_4, K_4; 13)$ coloring. For every graph $H = (V_H, E_H)$ in $\mathcal{G}_{13}$ we define the tuple $\Psi(H) = (v_H, y_H, X_H, \phi_H)$ where:

- $v_H = \min\{h \in V_H \mid d(h, G) = 11\}$
- y_H is the vertex in V_H that is not adjacent to v_H.
- $X_H \in \mathcal{G}_{11}$ is a canonically labeled graph obtained from the algorithm in

`nauty` [MP14].

- ϕ_H is the (unique) isomorphism between the graph induced in H by the vertices in $V_H \setminus \{v_H, y_H\}$ and X_H.

Notice that Ψ is well defined: if $\Psi(H) = (v_{H_1}, y_{H_1}, X_{H_1}, \phi_{H_1})$ and $\Psi(H) = (v_{H_2}, y_{H_2}, X_{H_2}, \phi_{H_2})$, from the definition of v_H, it is clear that $v_{H_1} = v_{H_2}$, and this defines $y_{H_1} = y_{H_2}$; also, from the definition of ϕ_H, the function $\phi_{H_1}\phi_{H_2}^{-1}$ is an isomorphism between X_{H_2} and X_{H_1}, so it must be that $X_{H_2} = X_{H_1}$ and $\phi_{H_1} = \phi_{H_2}$.

Intuitively, by considering all tuples $\Psi(H)$ that associate an $H \in \mathcal{G}_{13}$ with a graph $X \in \mathcal{G}_{11}$, one can find a collection $\mathcal{C}'(X)$ of subsets of vertices of X to which a vertex $y \in Y$ can be joined when reconstructing a $(C_4, C_4, K_4; 19)$ coloring. But $\mathcal{C}'(X)$ may not be the complete collection of subsets that are feasible neighborhoods of a vertex y: one needs to take into account that any subset of vertices of X that is symmetric to a subset in $\mathcal{C}'(X)$ is also a feasible neighborhood for a vertex y. Formally, for a canonically labeled graph $X = (V_X, E_X)$ in $\mathcal{G}_{11}$, a subset $V'_X \subseteq V_X$ is called a *canonically feasible neighborhood* if there is a graph H in $\mathcal{G}_{13}$ such that $\Psi(H) = (v_H, y_H, \phi_H, X)$ and $V'_X = \{\phi_X(\phi_H(v)) \,|\, v \in \mathcal{N}(y_H, H)\}$. We denote by $\mathcal{C}'(X)$ the set of all canonically feasible neighborhoods of in X. The set $\mathcal{C}(X)$ of *feasible* neighborhoods in graph X is then the set of all subsets $\{\phi(v) \,|\, v \in V'_X\}$ of V_X for some $\phi \in \mathrm{Aut}(X)$ and some $V'_X \in \mathcal{C}'(X)$. Compare this definition to the definition of *feasible cones* by McKay and Radziszowski in [MR95] and [MR97].

With this description, it is possible to implement an algorithm that will calculate, for every canonically labeled graph $X \in \mathcal{G}_{11}$, the sets $\mathcal{C}(X)$ and $\mathcal{C}'(X)$ by first iterating over all graphs $H \in \mathcal{G}_{13}$ finding the tuples $\Psi(X)$ that define all subsets in $\mathcal{C}'(X)$ and then applying every automorphism of the graph X to the vertices of every subset in $\mathcal{C}'(X)$ to generate a (possibly larger) set $\mathcal{C}(X)$. We used `nauty` to perform the mappings required to calculate the tuples in $\Psi(H)$ and to calculate the automorphism groups of the graphs X to generate the complete set of feasible neighborhoods. An alternative way of doing this is to iterate over every canonically labeled graph $X \in \mathcal{G}_{11}$, adding a vertex v adjacent to every vertex in X and an additional vertex y, then joining y with every possible subset of vertices in X and looking up the resulting graph of order 13 in $\mathcal{G}_{13}$: if the lookup is successful, the subset to which y was joined belongs to $\mathcal{C}(X)$.

5.4 Compatible Neighborhoods

The previous section was concerned with what can happen when we add one vertex as we try to extend a graph in $\mathcal{G}_{11}$ to a graph on 19 vertices that is a valid third color for some $(C_4, C_4, K_4; 19)$ coloring. We now take one more step in that direction: Consider two vertices y_1 and y_2 in Y that are joined to subsets V_{X_1} and V_{X_2} of vertices in X. Obviously, $V_{X_1} \in \mathcal{C}(X)$ and $V_{X_2} \in \mathcal{C}(X)$, but this is not enough to guarantee the graphs of order 14 formed by this setup when y_1 and y_2 are adjacent and non-adjacent, respectively, is a valid third color for some $(C_4, C_4, K_4; 14)$ coloring. Because, in practice,

testing for this condition requires expensive computations, our goal is to test for simpler conditions that will discard graphs that are not valid third colors, without false negatives. Our choice of conditions is the following:

(K_4) G must not contain K_4 as a subgraph.

$\left(\overline{K_6}\right)$ G must not contain an independent set of order 6.

$(4K_1 \cup K_3)$ No subset of 7 vertices can induce the graph $4K_1 \cup K_3$ in G.

$(3K_1 \cup 2K_2)$ No subset of 7 vertices can induce the graph $3K_1 \cup 2K_2$ in G.

The reason for condition (K_4) is obvious from the purpose of this construction. The reason for condition $\left(\overline{K_6}\right)$ is that $R(C_4, C_4) = 6$ so if G contained an independent set of order 6, at least the complete graph on 6 vertices in the complement of G cannot be split into 2 colors avoiding C_4. The reason for conditions $(4K_1 \cup K_3)$ and $(3K_1 \cup 2K_2)$ is that the complement of these graphs, shown in Figure 5.4, cannot be split into 2 colors avoiding C_4. This fact, while easy to verify computationally, can be stated as a lemma and proven analytically. For this, we rely heavily on another lemma about the coloring of $K_6 - e$ avoiding monochromatic C_4's, which is discussed in Appendix A.

Lemma 5.2. *There is no blue/red coloring of the complements of the graphs* $4K_1 \cup K_3$ *and* $3K_1 \cup 2K_2$ *that avoids monochromatic* C_4*'s.* □

Proof. The exact same reasoning applies to both graphs so, in the following, G will stand for $4K_1 \cup K_3$ or $3K_1 \cup 2K_2$. Assume, by contradiction, that

there is a blue/red coloring of $\overline{G}$ that avoids monochromatic C_4's. Let v be a vertex of minimum degree in $\overline{G}$. Then the graph $\overline{G} - v$ is isomorphic to $K_6 - e$ and, by Lemma A.1, there is only one way to color this graph avoiding monochromatic C_4's. Label the remaining 6 vertices w_0, w_1, w_2, w_3, x, y so that xy is the missing edge and $w_0w_1w_2w_3$ is a monochromatic P_4. Because the two colors in the C_4-free coloring of $K_6 - e$ are symmetric, we can assume without loss of generality that:

- There are at least as many vw_i blue edges as there are red vw_i edges, otherwise we can exchange the colors.
- The path $w_0w_1w_2w_3$ is blue (and the path $w_2w_0w_3w_1$ is red), otherwise we can relabel these vertices.
- The edges w_0x and w_1y are also blue, otherwise we can exchange the labels x and y.

From the blue paths $w_0w_1w_2$, $w_1w_2w_3$, w_0xw_4 and w_1yw_2 one can see that the collection $W = \{\{w_0, w_2\}, \{w_1, w_3\}, \{w_0, w_3\}, \{w_1, w_2\}\}$ is such that, if vw_i and vw_j are blue and $\{w_i, w_j\} \in W$, then we would have a blue C_4. Clearly, if three or more of the vw_i edges were blue, some set $\{w_i, w_j\} \in W$ would be such that vw_i and vw_j are both blue and a blue C_4 would be formed. Then there are exactly two blue vw_i edges and two red vw_i edges. We then have $\binom{4}{2} - |W| = 2$ possible choices of blue edges between v and the w_i vertices: either vw_0 and vw_1 are blue, in which case the edges vw_2 and vw_3 are red and

(a) $4K_1 \cup K_3$ (b) $3K_1 \cup 2K_2$

Figure 5.4: Forbidden subgraphs

we have a red cycle $vw_2w_0w_3$; or vw_2 and vw_3 are blue, in which case the edges vw_0 and vw_1 are red and we would have a red cycle $vw_0w_3w_1$. In any case, we reach a contradiction, so there is no blue/red coloring of $\overline{G}$ that avoids a monochromatic C_4. ■

One can then generate a compatibility matrix of all subsets in $\mathcal{C}(X)$ for the cases when y_1 and y_2 are adjacent and non-adjacent. These matrices are useful even beyond the point where 2 vertices have been added to extend the $(C_4, C_4, K_4; 12)$ coloring because, at the point where we add the k-th vertex to this construction and consider a candidate neighborhood for it, one can quickly check it is compatible with the $k-1$ neighborhoods chosen before for the other vertices.

5.5 Extension Process

Having data for the feasible neighborhoods and the pairs of compatible neighborhoods for every graph $X \in \mathcal{G}_{11}$, one can extend these graphs to graphs

of order 19 that have the structure described in Theorem 5.1.

In adding vertices to a graph in $\mathcal{G}_{11}$ to obtain a graph on 19 vertices that is a valid third color for some $(C_4, C_4, K_4; 19)$ coloring, we mention the following two strategies that can be used:

1. Adding a single vertex in every step. In its simplest form, this strategy supposes we have added k vertices to the initial graph in $\mathcal{G}_{11}$, so at the $k+1$-st step we have a graph or order $12+k$ if we take into account the vertex v adjacent to all the vertices of the original graph. For the $k+1$-st vertex, we need to try all possible connections to the previous k vertices that have been added and then pick a feasible neighborhood that is compatible with the rest of the neighborhoods chosen so far. A nice property of this strategy is that one can assume that, for some ordering of the list of feasible neighborhoods, their assignment to the vertices of the extended graph are in non-decreasing order. Intuitively, this means that, at a successful termination of this extension process, the vertices of some graph $Y \in \mathcal{G}_7$ will have been "discovered" in some order defined by the compatible neighborhoods attached to them. A drawback of blindly selecting the adjacency of the $k+1$-st vertex with respect to the k vertices previously added is that one may end up with a graph of order $k+1$ that is not a subgraph of any graph in $\mathcal{G}_7$.

2. Appending each graph $Y \in \mathcal{G}_7$ and joining each of the 7 vertices of the appended graph using feasible neighborhoods. In every step, a feasible

neighborhood is chosen for the next vertex with the condition that the new neighborhood must be compatible with every neighborhood selected for the previous vertices. This approach improves over the first approach in the sense that we know *a priori* that the 7 vertices added to the graph will yield a graph with the structure described in Theorem 5.1. It also suggests an easy scheme for problem partitioning in order to run the extension process in parallel: every pair of graphs (X, Y) can be checked independently using a total of $|\mathcal{G}_{11}| \cdot |\mathcal{G}_7|$ processes. The drawback of this approach is that no order can be assumed for the feasible neighborhoods assigned to the vertices of Y. Also, in a naive implementation of this approach a graph of order $n < 7$ that is a subgraph of several graphs in $\mathcal{G}_7$ will be checked many times for the same combination of feasible neighborhoods.

In both strategies, the graph that is generated in every step is checked for conditions (K_4), $\left(\overline{K_6}\right)$, $(4K_1 \cup K_3)$ and $(3K_1 \cup 2K_2)$ described in Section 5.4. If the graph fails to satisfy any of these conditions, the extension of X with Y using the current combination of neighborhoods is aborted. It might be that the complement of the final graph generated once feasible neighborhoods of X are chosen for each one of the 7 vertices of Y can not be split into two colors avoiding C_4 in both of them. Thus, this additional final check is needed in order to consider an extended graph as a valid third color for some $(C_4, C_4, K_4; 19)$ coloring.

While the description of the strategies above involves feasible neighborhoods, a way to reduce the number of combinations to try is to use canonically feasible neighborhoods. The following theorem allows us to pick the first feasible neighborhood from the (possibly smaller) set $\mathcal{C}'(X)$:

Theorem 5.3. *coloring Let G be a graph obtained by the extension process described above, composed of a vertex v, a graph $X \in \mathcal{G}_{11}$ and a graph $Y \in \mathcal{G}_7$. Let $X_0, X_1, \ldots, X_6$, $X_i \in \mathcal{C}(X)$, be the sequence of feasible neighborhoods attached to the vertices in Y. Then there is a graph G' composed of v, X and Y, isomorphic to G, with a sequence of feasible neighborhoods $X'_0, X'_1, \ldots, X'_6$ attached to the vertices of Y and with $X'_0 \in \mathcal{C}'(X)$.* □

Proof. Since $X_0 \in \mathcal{C}(X)$ there is, by definition, a permutation $\phi \in \mathrm{Aut}(X)$ and a X'_0 such that $X_0 = \phi(X'_0)$. Let $X'_i = \phi^{-1}(X_i)$. It is easy to check that the permutation ϕ' defined as

$$\phi'(x) = \begin{cases} x & \text{if } x \notin X \\ \phi^{-1}(x) & \text{otherwise} \end{cases}$$

is an isomorphism between G' and G. ■

The following section describes the strategy that we used in our extension process. It strikes a balance between the two strategies listed before and allows for a convenient partitioning of the problems into several subproblems that can be executed in parallel.

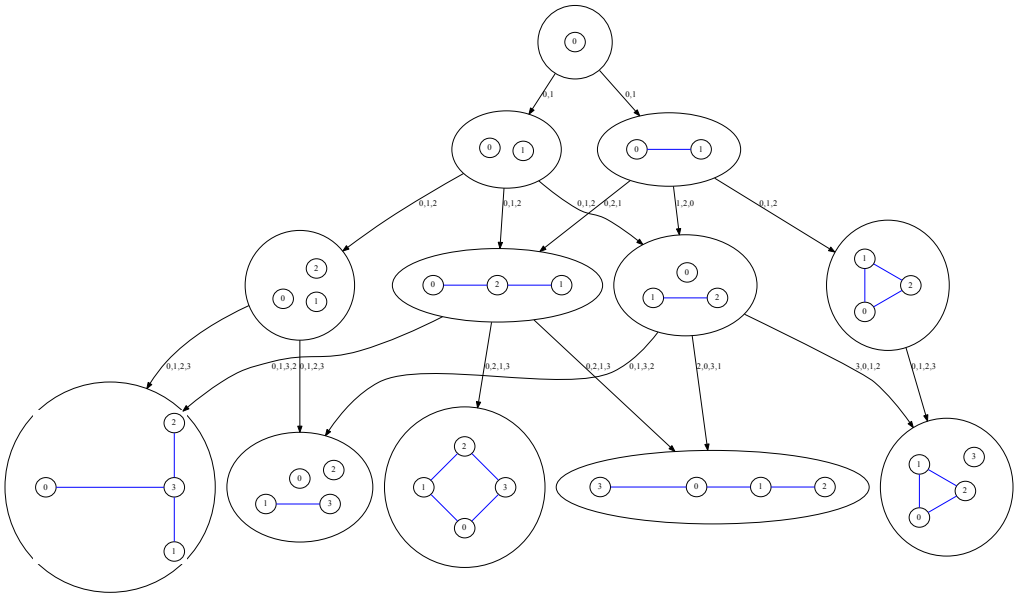


Figure 5.5: The ancestors graph of 5 graphs of order 4

5.6 Ancestors Graph

For a set of (directed or undirected) graphs $\mathcal{G}$, we define the ancestors graph $\aleph(\mathcal{G})$ to be a directed graph whose vertices are canonically-labeled graphs, with $G' \in V(\aleph(\mathcal{G}))$ if and only if G' is a subgraph of at least one graph $G \in \mathcal{G}$. An edge between graphs G_i and G_j exists in $\aleph(\mathcal{G})$ if $n(G_i)+1 = n(G_j)$ and G_i is an induced subgraph of G_j. While this graph is not a tree, it is easy to see that it is proper k-level hierarchy [STT81], where $k = \max\{n(G) | G \in \mathcal{G}\}$. We can identify the root node of this graph as the graph on one vertex. Figure 5.5 illustrates the concept of the ancestor graph of an arbitrarily chosen set of five graphs of order 4.

While the definition above suffices to understand the relationship be-

tween the substructures of the graphs in $\mathcal{G}$, we equip the edges of $\aleph(\mathcal{G})$ with additional information that will be needed for our extension process. Denote by $\mathbb{N}_k$ the set $\{0, 1, \ldots, k-1\}$. We associate an edge $(G_i, G_j) \in E(\aleph(\mathcal{G}))$ with a permutation $\phi_{G_j}^{G_i}$ of the set $\mathbb{N}_{n(G_j)}$ with the property that the graph $G_j\left[\phi_{G_j}^{G_i}\left(\mathbb{N}_{n(G_i)}\right)\right]$ is isomorphic to G_i. That is, the permutation $\phi_{G_j}^{G_i}$ maps the labels of the vertices of G_i to labels of vertices of G_j that define a subgraph of G_j that is isomorphic to G_i. For ease of notation, we will write $\phi_{G_i}^{G_j}$ to refer to the inverse of $\phi_{G_j}^{G_i}$. It should be obvious from the context and the fact that $n(G_i) + 1 = n(G_j)$ to which of the permutations we refer to.

We can now use the ancestor graph in our extension process as follows: Every step is associated to a node in the ancestors graph of $\mathcal{G}_7$. The first step is associated to the graph of one vertex. At the k-th step of our extension process, we have added k vertices to our original graph X, the first one being v, which is adjacent to every vertex in X. We then proceed as described in Algorithm 1.

The main advantage of this approach is that, once a graph $G' \in V(\aleph(\mathcal{G}_7))$ is checked under certain combination of feasible neighborhoods, that check applies to all graphs $G \in \mathcal{G}_7$ that contain G' as a subgraph. This eliminates some redundant computations, speeding the overall process. Furthermore, because all graphs that are nodes of the ancestor graph are, by definition, subgraphs of at least one graph in $\mathcal{G}_7$, we never consider combinations on graphs that are not useful for the extension process.

Further speedup can be achieved by recovering the ability to assume

Algorithm 1 Extension process using the ancestors graph $\aleph(\mathcal{G}_7)$

Require: G_i, the graph at the current step

1: **if** G_i satisfies conditions (K_4), $\left(\overline{K_6}\right)$, $(4K_1 \cup K_3)$ and $(3K_1 \cup 2K_2)$ **then**
2: $J \leftarrow \{G_j \,|(G_i, G_j) \in E(\aleph(\mathcal{G}_7))\}$
3: **if** $J = \emptyset$ **then** ▷ We have completed an extension, $n(G_i) = 19$
4: Check if there is a 2-coloring of $\overline{G_i}$ that avoids C_4 in both colors
5: **if** $\overline{G_i}$ can be properly colored **then**
6: Output G_i
7: **end if**
8: **else**
9: **for** $G_j \in J$ **do**
10: $G_i' \leftarrow G_i \cup v$
11: $V_j \leftarrow \left\{13 + k \,\middle|\, k \in \phi_{G_i}^{G_j}\left(\mathcal{N}\left(\phi_{G_j}^{G_i}(n(G_i)), G_j\right)\right)\right\}$
12: Add edges vv' to G_i' for $v' \in V_j$
13: Recurse using G_i'
14: **end for**
15: **end if**
16: **end if**

that the sequence of feasible neighborhoods attached to the vertices of Y is non-decreasingly ordered. Notice that this cannot be assumed given our current definition of the ancestors graph because of the following subtle issue: the permutation $\phi_{G_j}^{G_i}$ is not unique for every edge (G_i, G_j). We would need to consider every possible permutation $\phi_{G_j}^{G_i}$ that defines a subgraph in G_j that is isomorphic to G_i. We can do this by considering a multigraph (a graph that allows more than one edge between two nodes) or by considering permutations $\phi\left(\phi_{G_i}^{G_j}\left(\mathcal{N}\left(\phi_{G_j}^{G_i}(n(G_i)), G_j\right)\right)\right)$ in line 13 of Algorithm 5.5 for every $\phi \in \mathrm{Aut}(G_i)$.

McKay and Radziszowski used an idea similar to the ancestors graph, named *double tree*, in their extension process to find $R(4,5)$ [MR95]. Also,

Calvert, Schuster and Radziszowski used a smaller ancestors graph in [CSR12] to find $R(K_5 - P_3, K_5)$.

5.6.1 Problem Partitioning

For every graph X, an extension process based on the ancestors graph of $\mathcal{G}_7$ eliminates redundant operations, but the order of the ancestors graph is too large to carry out this process on a single computer. This raises the question of what partition scheme to use when working with ancestors graphs. The partitioning scheme we chose for this extension process is based on the paths from the root node to the nodes that are graphs in $\mathcal{G}_7$. The extension process can be defined as a process that advances over all such paths. A path-based partitioning can then be proposed by listing all possible paths from the root to the graphs in $\mathcal{G}_7$ and assigning groups of paths to different computing resources. At this point, many alternatives for path grouping can be chosen: we could, for instance, assign an equal amount of paths to every computing resource. The issue we incur when we do this is that path prefixes that are shared among paths reintroduce redundant operations in the overall process. Unfortunately, this is unavoidable for path-based partition schemes, so our goal was to select a path-based partitioning that would introduce a reasonable amount of redundant operations while keeping the number of paths to be explored under a manageable size.

The partitioning scheme we selected was based on the following observation: because no edge exists between nodes in the same level of the hierarchy,

one can choose a level and the nodes of this level will induce a partition in the set of paths. Choosing a large order to partition the paths will give out larger path prefixes shared among computing resources, while choosing a small one will bundle too many paths per computing resource. We chose to partition the paths at nodes of order 4. There are 10 of these nodes, so the total number of computing resources required to run the extension process was $10\,|\mathcal{G}_{11}| = 5750$.

5.7 Validation

We have access to a dataset of 7 $(C_4, C_4, K_4; 19)$ colorings generated through a similar (but smaller and non-exhaustive) extension process. We can partially validate the correctness of the algorithm described in 5.5 by making sure it generates at least the 7 graphs corresponding to the first color of these 7 colorings.

Moreover, given these 7 graphs, one can already find out what graphs $X \in \mathcal{G}_{11}$, $Y \in \mathcal{G}_7$ and sequence of 7 feasible neighborhoods of X will generate them using the following "reverse engineering" process:

1. Find a vertex v of degree 11.
2. Set X to be the neighborhood of v.
3. Set Y to be the set of vertices that are not adjacent to v.
4. Find canonical labeling of the vertices of X and Y.

5. The i-th feasible neighborhood is then the neighborhood of the i-th vertex in Y restricted to the vertices that define X.

In the above process, there may be different vertices v to choose for step 1, which will yield possibly different graphs X and Y.

5.8 Statistics

The purpose of this section is to provide an overview of the number of objects involved in the calculations for the extension process. These numbers should give the reader an idea of the computational effort needed to carry this process to completion.

Figure 5.6 shows a bar plot of the number of feasible neighborhoods per graph in $\mathcal{G}_{11}$, clustered into 10 groups. One can readily see from this plot that most of the graphs X have between 600 and 800 feasible neighborhoods. Because the number of feasible neighborhoods determines the number of iterations per recursion level (i.e., the number of iterations per node in the graph Y), we can estimate the "difficulty" of examining each graph X by grouping them according to their number of feasible neighborhoods.

A better estimate of the difficulty of examining each graph X involves the number of compatible neighborhoods. A way to relate the number of feasible neighborhoods with the number of compatible neighborhoods is defining the ratio r between the number of compatible neighborhoods and the number of pairs of feasible neighborhoods. Some statistics about r are shown in Table

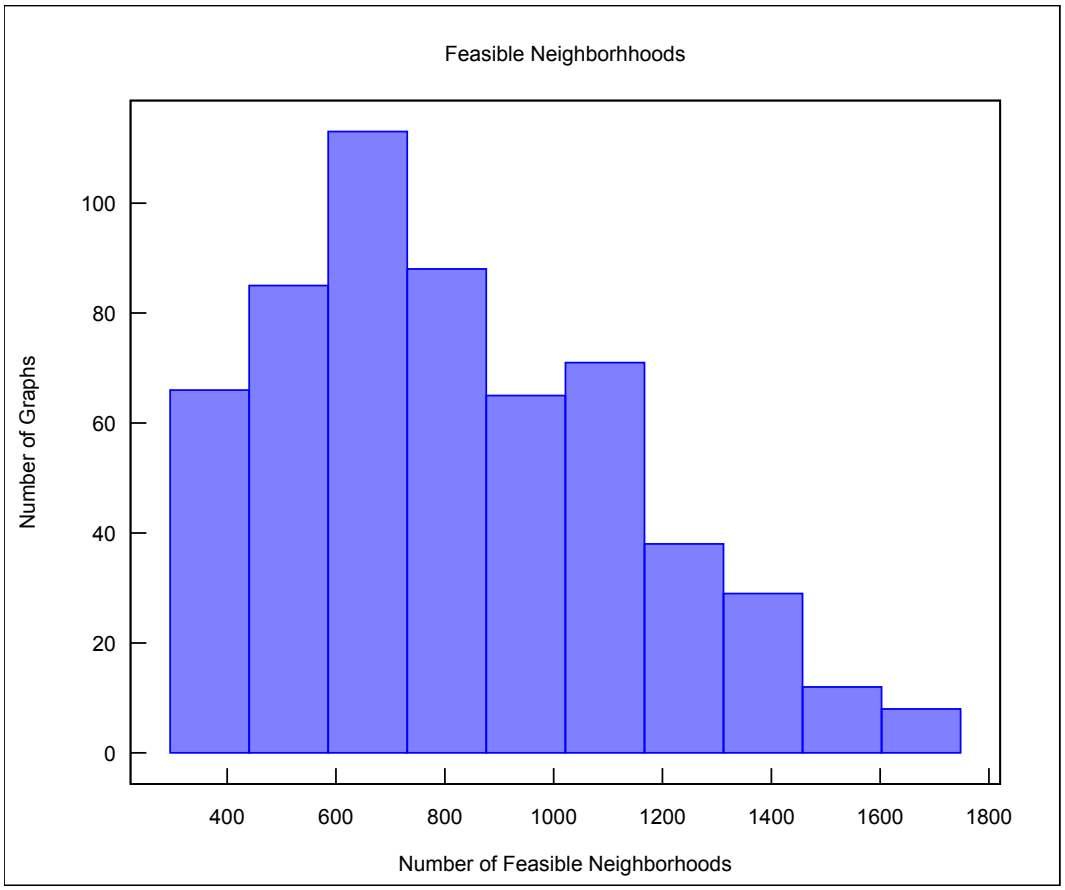


Figure 5.6: Graphs in $\mathcal{G}_{11}$ grouped by number of feasible neighborhoods

5.7. We see that, in general, the number of compatible neighborhoods in the case when y_1 and y_2 are non-adjacent is much larger than when y_1 and y_2 are adjacent.

	r Connected	r Disconnected
Mean	0.05	0.65
Median	0.04	0.64
Standard Deviation	0.03	0.06
Minimum	0.01	0.51
Maximum	0.19	0.83

Table 5.7: Descriptive statistics of the ratio r

Chapter 6

Future Work

We briefly describe directions for future work related to this thesis. Section 6.4, in particular, explains the problems we had in completing our search for $(C_4, C_4, K_4; 19)$ colorings given our current infrastructure. The other sections are proposed lines of work derived from our research.

6.1 Colorings and Constraint Satisfaction Problems

In Sections 3.2.1 and 3.2.2 we briefly covered some methods we used to generate cycle-free colorings using Constraint Satisfaction Problems. One question that remained unanswered with respect to generating colorings using SAT solvers is for which cases is this method faster than direct enumeration. The answer might lie in the clause density of the formulas that are generated by this approach, since the SAT solving community is often concerned with this parameter [DW06]. Conversely, proper benchmarking of several SAT solvers could reveal what implementation—if any—is better suited for this problem.

It should be relatively simple to provide an independent verification of some results of Chapter 4 using SAT solvers. For instance, finding C_5-free 2-colorings of the complement of $4K_4^{\times}$ should agree with the 249,540 colorings

we found. While this is essentially just finding all satisfying assignments for a single formula, in practice we were unable to find these in a short period of time. This problem seems to be suitable for techniques like parallel SAT solving [AAB13].

6.2 Improving the Bounds on $R_4(C_5)$

The large gap between the lower and upper bounds for $R_4(C_5)$ suggests there is great room for improvement in this regard. Notice that the upper bound was obtained by purely analytical results, so an empirical approach is in order. We believe that a closer look at the $(C_5, C_5, C_5; n)$ colorings produced in this thesis should give some insights about the behavior of 4-colorings avoiding C_5.

6.3 Critical Colorings for Cycle Ramsey Numbers

As Yuansheng and Rowlinson [YR92b] showed in their approach to $R_3(C_5)$, understanding critical colorings for certain Ramsey numbers may be an important step towards establishing these. Results obtained via this approach often come in the form of bounds for Ramsey numbers if certain condition holds for all of their critical colorings. For instance, Li [Li09] proved that the inequality $R_k(C_{2m+1}) \leq \left(c^k k!\right)^{\frac{1}{m}}$ holds when the C_{2m+1}-free k-colorings on $R_k(C_{2m+1}) - 1$ vertices are almost regular.

In this respect, and related to the problem of C_5-free colorings, we

propose the following problem, inspired in our classification of $(C_5, C_5, C_5; 16)$ colorings based on the K_4's found in one of the colors:

Problem 6.1. *Let G be a C_5-free k-coloring of order 2^{k+1} with 2^{k-1} disjoint copies of K_4 in one of the colors. We produce a new C_5-free $k+1$-coloring $\widetilde{G}$ of order $2^{k+1}+1$ where the coloring restricted to the first 2^{k+1} vertices matches G, and m of the edges from the additional vertex are colored using the additional color. What is the smallest possible value of m?*

Intuitively, we wish to add a vertex to a C_5-free k-coloring to produce a new C_5-free k-coloring, but since this might be impossible with only k colors (e.g, when $R_k(C_5) = 2^{k+1}$ which is the case for $k = 2, 3$), we optionally admit "missing" edges which are edges of the $k+1$-th color in $\widetilde{G}$. We then wish to find what is the smallest number of missing edges that we could have. This setup is similar to that of star-critical Ramsey numbers [HI11, WSR15].

Notice that there is always at least one coloring G that satisfies Problem 6.1, namely, the one obtained from the *blow-up* construction. For this particular coloring, we conjecture that $m = 3$ and a witness coloring for this value can be constructed inductively by noticing that, for $k = 1$, only one edge from the new vertex can be of color 1 (otherwise a cycle of length 5 would be formed); and for k colors, the graph induced by the k-th color is isomorphic to a complete bipartite graph $K_{2^k,2^k}$ so one can join all the new edges that go to one of the partitions using the k-th color, and use a $k-1$-coloring for the edges going to the remaining vertices.

6.4 $R(C_4, C_4, K_4)$

At the time of this writing, we have completed a fraction of the extension process described in Section 5.5. The main obstacle to overcome in order to complete this search in the Open Science Grid is the fact that grid jobs are not expected to take more than 24 hours, yet the extension process for graphs with thousands of feasible neighborhoods (see Figure 5.6) may take more than 24 hours. This is true even when using finer partitions of the ancestors graph.

Particularly problematic is the fact that all of the graphs obtained from the $(C_4, C_4, K_4; 19)$ colorings we know of (see Section 5.7) have more than a thousand feasible neighborhoods, which can be classified as the "hard" cases to explore. This makes the verification a difficult task. On the other hand, this fact may suggest a stronger characterization of the graphs in the family $\mathcal{G}_{11}$ described in Section 5.3. For instance, a theoretical result bounding the minimum number of feasible neighborhoods required for a graph on 11 vertices to produce a suitable graph on 19 vertices after the extension process would help discard several hundreds of graphs that have relatively few feasible neighborhoods.

Appendices

Appendix A

Colorings of $K_6 - e$ Avoiding C_4

The following lemma can be easily verified with the computational tools described in Chapter 3, yet we include an analytical proof of it for the sake of completeness:

Lemma A.1. *There is exactly one (up to isomorphism) blue/red coloring of $K_6 - e$ with no monochromatic C_4.* □

Proof. We color $K_6 - e$ dividing our options depending on the number of triangles in the blue color. One important observation is that two triangles of the same color cannot share an edge, because the edges that are distinct from the shared edge will form a C_4. We then have the following cases:

3 or more triangles: Because $K_6 - e$ has only 6 vertices, if there were 3 or more triangles in the blue color, two of them will necessarily share an edge, so this case is impossible.

2 triangles: We divide this into two subcases. First, we assume these triangles share a vertex v. Let vr_1r_2 be one triangle and vl_1l_2 be the other triangle. If any of the edges r_il_j was blue, we would have a blue C_4; but also, if they were all red, we would have a red cycle $r_1l_1r_2l_2$, so one edge

must be missing and the other three must be red. Assume, without loss of generality, that r_1l_1 is missing. Now we consider the sixth vertex w: if there were blue edges wr_i and wl_j, we would have a blue cycle wr_ivl_j so there are at least 3 red edges between w and the r_i and l_i vertices. Assume wr_2, wl_1 and wl_2 are red. Then we have a red cycle $wl_1r_2l_2$, so we have reached a contradiction and there cannot be two triangles that share a single vertex. Now we assume these triangles do not share any vertex. Besides the edges that form the triangles, there are 9 edges between the vertices of these triangles. If two of them were blue, we would have a blue cycle of length 4, but if all of them were red we would have $\binom{3}{2}^2 = 9$ red cycles of length 4. We can "fix" some of these cycles by either coloring one edge blue or removing one edge, but each fix affects at most 4 cycles. With two fixes, we can fix at most 8 cycles, a contradiction. We have then shown that there are no blue/red colorings of $K_6 - e$ with 2 triangles that avoid monochromatic C_4's.

1 triangle: Let $t_0t_1t_2$ be the only blue triangle. As in the analysis of the previous case, we note that there are 9 potential cycles between the vertices of this triangle and the remaining vertices. Unlike the previous case, we can fix all of these cycles by deleting one edge, say t_0x, making the edges t_1y and t_2z blue and making the other 6 edges red. This choice of colors forces the edge yz to be red, to avoid a blue cycle t_1yzt_2, and t_1z and t_2y to be red to avoid a new blue triangle. This in turn forces the edges xy and xz to be blue to avoid red cycles $xyzt_1$ and $xzyt_2$. The

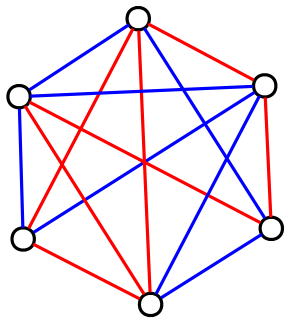


Figure A.1: The only blue/red coloring of $K_6 - e$ with no monochromatic C_4

coloring produced is shown in Figure A.1.

0 triangles: Let xy be the missing edge in $K_6 - e$ and let v_0, v_1, v_2, v_3 be the rest of the vertices. There are 3 options for colorings of the complete subgraph induced by $\{v_0, v_1, v_2, v_3\}$. The first option is that edges v_0v_1 and v_0v_3 are blue and the rest of the edges in the subgraph are red. Because of the red triangle $v_1v_2v_3$, there cannot be two red edges from x or y to the the set $\{v_1, v_2, v_3\}$, so there are two blue edges xv and two blue edges yv for $v \in \{v_1, v_2, v_3\}$. Furthermore, because of the path $v_1v_0v_3$, it cannot be that edges xv_1 and xv_3 are both blue, so it must be that xv_2 is blue. Applying the same reasoning to y, we see that yv_2 is blue. For the additional blue edges from x and y to $\{v_1, v_3\}$ we must choose different vertices for x and y because a blue cycle of length 4

would otherwise be formed, so we can assume xv_1 and yv_3 are blue. But this forces the edge xv_0 to be red, because we would otherwise have a blue triangle v_0v_1x and we are assuming there are no blue triangles in this coloring. With the edge xv_0 set to red, we would then have a red cycle $xv_0v_2v_3$, a contradiction. The second option is that there are 3 blue edges v_0v_1, v_0v_2 and v_0v_3 and a red triangle $v_1v_2v_3$. Again, because of the red triangle, two of the edges from x to $\{v_1, v_2, v_3\}$ must be blue, but because of the blue paths $v_1v_0v_2$, $v_1v_0v_3$ and $v_2v_0v_3$, two of the edges from x to $\{v_1, v_2, v_3\}$ must be red, a contradiction. Finally, the third option is that there is a blue path $v_0v_1v_2v_3$ and a red path $v_2v_0v_3v_1$. Because we are assuming there are no blue triangles in the coloring, at least one of the edges in each pair $\{xv_0, xv_1\}$, $\{xv_1, xv_2\}$ and $\{xv_2, xv_3\}$ must be red, but there cannot be more than 2 red edges from x to $\{v_0, v_1, v_2, v_3\}$ because of the red paths $v_2v_0v_3$ and $v_0v_3v_2$. Thus, there are exactly two blue edges from x to $\{v_0, v_1, v_2, v_3\}$ and, because of the blue paths $v_0v_1v_2$ and $v_1v_2v_3$, these edges must be xv_0 and xv_3. Then xv_1 and xv_2 are red, but applying the same reasoning to y, one would have that yv_1 an yv_2 are red also, and we would have a red cycle yv_1xv_2, a contradiction. We have then shown that there are no blue/red colorings of $K_6 - e$ with 0 triangles that avoid monochromatic C_4's.

Appendix B

Directed Cycle Detection: An Application to Voting Theory

The coloring methods discussed in Chapter 3 show an application of the cycle detection algorithm mentioned in Section 3.1 to undirected cycles. As we mentioned before, these concepts can be extended to detect directed cycles as well. Directed cycle detection plays an important role in the theory of (partial or total) orders, which in turn plays an important role in voting theory. This is because partial orders can be represented as directed acyclic graphs and vice-versa. A canonical example of the importance of cycles in voting theory is Condorcet's paradox, in which 3 votes $a > b > c$, $b > c > a$ and $c > a > b$ induce, by simple majority, an ordering $a > b > c > a$ of the candidates. In terms of graphs, this means that the *majority graph* of three transitive tournaments may not be transitive, i.e., it may contain a cycle. This appendix illustrates how a directed cycle detection algorithm can be used in a larger algorithm to construct a majority graph under certain conditions.

As it is usual in voting theory, our setup consists of a set of candidates C and a set of voters. Voters specify their preference for candidates as orders. The collection of orders specified by voters constitutes a preference profile.

From a preference profile we can build a graph that summarizes the overall preference of the voters as follows:

Definition B.1 (Majority Graph)**.** The *majority graph* of a preference profile is a directed graph $G = (C, E)$ where, for every pair of candidate $a, b \in C$, $(a, b) \in E$ if and only if more voters prefer b over a than a over b. □

We then say a graph G is induced by a preference profile if G is the majority graph of that preference profile. Consider the following theorem about majority graphs and total preorders:

Theorem B.1. *[FH14] If a majority graph G can be induced by two total preorders, it can be induced by two total orders.*

The following proof provides an algorithm that gives two total orders that induce G. The algorithm interprets the partial orders as graphs, and builds a new pair of graphs by first copying edges from the original partial orders and then arbitrarily selecting edges for pairs of candidates that were equivalent in both votes. The resulting graphs are guaranteed to induce G but may contain cycles and thus may not represent total orders. The crucial part of this proof is that we can always "break" these cycles *a posteriori* so that the two resulting graph after this phase do represent total orders.

Proof. Let $G_1 = (C, E_1)$ and $G_2 = (C, E_2)$ be the two digraphs that induce G. Construct graphs $G'_1 = (C, E'_1)$ and $G'_2 = (C, E'_2)$ according to the following rules:

Rule 0 If $(a_i, a_j) \in E_1$ and $(a_j, a_i) \notin E_1$, then $(a_i, a_j) \in E_1'$. Similarly, if $(a_i, a_j) \in E_2$ and $(a_j, a_i) \notin E_2$, then $(a_i, a_j) \in E_2'$.

Rule 1 If $(a_i, a_j) \in E_1$, $(a_j, a_i) \in E_1$ but $(a_i, a_j) \in E_2$ and $(a_j, a_i) \notin E_2$, then $(a_i, a_j) \in E_1'$. Similarly, if $(a_i, a_j) \in E_2$, $(a_j, a_i) \in E_2$ but $(a_i, a_j) \in E_1$ and $(a_j, a_i) \notin E_1$, then $(a_i, a_j) \in E_2'$

Rule 2 If $(a_i, a_j) \in E_1$, $(a_j, a_i) \in E_1$, $(a_i, a_j) \in E_2$ and $(a_j, a_i) \in E_2$ then randomly choose to include (a_i, a_j) in E_1' and (a_j, a_i) in E_2' or vice-versa.

It is clear that graphs G_1' and G_2' so constructed are tournaments. We will now prove some additional properties about these graphs:

Lemma B.2. *A cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ in G_1' contains no edges added by Rule 0.*

Proof of Lemma B.2. Let r be the number of edges that were added to the cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ by Rule 0. Assume, by contradiction, that $r > 0$. It is clear that not all edges of the cycle could be added to G_1' by Rule 0, since it would mean G_1 had the cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ so it must also have the cycle $a_{i_0}, a_{i_k}, \ldots, a_{i_1}$ and all of the edges in the cycle $a_{i_0}, a_{i_2}, \ldots, a_{i_k}$ would have been added by Rule 2. Then, without loss of generality, we can assume the edge $\left(a_{i_{k-1}}, a_{i_k}\right)$ was added to G_1' by Rule 0 (so $\left(a_{i_{k-1}}, a_{i_k}\right) \in E_1$) and the edge (a_{i_k}, a_{i_0}) was added to G_1' by Rules 1 or 2 (so $(a_{i_k}, a_{i_0}) \in E_1$ and $(a_{i_0}, a_{i_k}) \in E_1$). But G_1 represents a total preorder, so one must have $\left(a_{i_{k-1}}, a_0\right) \in E_1$ and the sequence of edges $a_{i_0}, a_{i_2}, \ldots, a_{i_{k-1}}$ forms a shorter cycle in G_1', since the edge $\left(a_{i_{k-1}}, a_0\right) \in E_1$ must have been added to G_1' by Rule 0. It is easy to see that

the number of edges in the cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_{k-1}}$ that were added by Rule 0 is also r. By following the same reasoning on this new cycle, one can find a shorter cycle with r edges added by Rule 0. This means that repeating the process $(k+1) - r$ times, one will find:

- A cycle of length r with r edges added by Rule 0 if $r \geq 3$, a contradiction.
- Indices $p < q < r$ such that edges $\left(a_{i_p}, a_{i_q}\right)$ and $\left(a_{i_q}, a_{i_r}\right)$ were added by Rule 0 and edge $\left(a_{i_r}, a_{i_p}\right)$ was added by Rules 1 or 2 if $r = 2$. This means, in particular, that $\left(a_{i_r}, a_{i_q}\right) \notin E_1$, $\left(a_{i_r}, a_{i_p}\right) \in E_1$ and $\left(a_{i_p}, a_{i_r}\right) \in E_1$; but G_1 is transitive, so $\left(a_{i_r}, a_{i_p}\right) \in E_1$ and $\left(a_{i_p}, a_{i_q}\right) \in E_1$ imply $\left(a_{i_r}, a_{i_q}\right) \in E_1$, a contradiction.
- Indices $p < q < r$ such that the edge $\left(a_{i_p}, a_{i_q}\right)$ was added by Rule 0, and edges $\left(a_{i_q}, a_{i_r}\right)$ and $\left(a_{i_r}, a_{i_p}\right)$ were added by Rules 1 or 2 if $r = 1$. This means, in particular, that $\left(a_{i_q}, a_{i_p}\right) \notin E_1$, $\left(a_{i_q}, a_{i_r}\right) \in E_1$, $\left(a_{i_r}, a_{i_q}\right) \in E_1$, $\left(a_{i_r}, a_{i_p}\right) \in E_1$ and $\left(a_{i_p}, a_{i_r}\right) \in E_1$; but G_1 is transitive, so $\left(a_{i_p}, a_{i_r}\right) \in E_1$ and $\left(a_{i_r}, a_{i_q}\right) \in E_1$ imply $\left(a_{i_q}, a_{i_p}\right) \in E_1$, a contradiction.

so it must be the case that $r = 0$. ■

Lemma B.2 can be strengthened to state all edges in a cycle in G_1' were added by Rule 2: If there were $r > 0$ edges added by Rule 1, these same edges would be added to G_2' by Rule 0, but the choice of the direction of the edges added by Rule 2 was arbitrary, so we could have chosen them in the opposite direction, forming a cycle in G_2' with $r > 0$ edges added by Rule 0. Applying lemma B.2 to G_2' would yield a contradiction. So we have

Lemma B.3. *A cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ in G'_1 only contains edges added by Rule 2.*

Moreover, by the definition of Rule 2, this obviously implies the following:

Corollary B.4. *All cycles found in G'_1 are found in G'_2.*

Now let $a_{i_0}, a_{i_1}, \ldots, a_{i_k}$ be the largest cycle in G'_1 (or, equivalently, G'_2). We reverse the orientation of the edge (a_{i_k}, a_{i_0}) in both graphs. Notice that this does not introduce new cycles because, if there was a cycle $a_{i_0}, a_{i_k}, a_{j_0}, a_{j_1}, \ldots, a_{j_l}$, the cycle $a_{i_0}, a_{i_1}, \ldots, a_{i_k}, a_{j_0}, a_{j_1}, \ldots, a_{j_l}$ would be of length $k + j + 2 > k + 1$, a contradiction. This process then reduces the number of cycles by at least 1 in every step. Because the number of cycles is finite, this process terminates and the resulting graphs are acyclic, i.e., they represent total orders. ■

An implementation of the algorithm used to prove Theorem B.1 can use a modified version of the algorithm described in Section 3.1. The modifications are needed to take into account directed edges and to return cycles of all sizes, not only a fixed size.

Bibliography

[AAB13] S. Asghar, E. Aubanel, and D. Bremner. A dynamic moldable job scheduling based parallel SAT solver. In *42nd International Conference on Parallel Processing*, pages 110–119. IEEE, 2013.

[BE73] J. A. Bondy and P. Erdős. Ramsey numbers for cycles in graphs. *Journal of Combinatorial Theory, Series B*, 14(1):46–54, 1973.

[Bie08] A. Biere. PicoSAT essentials. *Journal on Satisfiability, Boolean Modeling and Computation*, 4(2-4):75–97, 2008.

[Bol04] B. Bollobás. *Extremal Graph Theory*. Courier Dover Publications, 2004.

[CFS89] C. Clapham, A. Flockhart, and J. Sheehan. Graphs without four-cycles. *Journal of Graph theory*, 13(1):29–47, 1989.

[CS71] G. Chartrand and S. Schuster. On the existence of specified cycles in complementary graphs. *Bulletin of the American Mathematical Society*, 77(6):995–998, 1971.

[CSR12] J. Calvert, M. Schuster, and S. Radziszowski. Computing the Ramsey number $R(K_5 - P_3, K_5)$. *Journal of Combinatorial Mathematics and Combinatorial Computing*, (82):131–140, 2012.

[DD11] J. Dybizbański and T. Dzido. On some Ramsey numbers for quadrilaterals. *The Electronic Journal of Combinatorics*, 18(1), 2011.

[DLSM04] P. T. Darga, M. H. Liffiton, K. A. Sakallah, and I. L. Markov. Exploiting structure in symmetry detection for CNF. In *Proceedings of the 41st Annual Design Automation Conference*, DAC '04, pages 530–534, New York, NY, USA, 2004. ACM.

[DSS+05] E. Deelman, G. Singh, M. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. C. Laity, J. C. Jacob, and D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

[DW06] E. Dantsin and A. Wolpert. MAX-SAT for formulas with constant clause density can be solved faster than in $\mathcal{O}(2^n)$ time. In Armin Biere and CarlaP. Gomes, editors, *Theory and Applications of Satisfiability Testing*, volume 4121 of *Lecture Notes in Computer Science*, pages 266–276. Springer Berlin Heidelberg, 2006.

[ERS66] P. Erdős, A. Rényi, and V. T. Sós. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 1:215–235, 1966.

[FH14] Z. Fitzsimmons and E. Hemaspaandra. Personal communication, 2014.

[Flo67] R. W. Floyd. Nondeterministic algorithms. *Journal of the ACM*, 14(4):636–644, 1967.

[Gar97] M. Gardner. *Penrose Tiles to Trapdoor Ciphers*. Cambridge University Press, 1997.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability*. Macmillan Higher Education, 1979.

[Gra90] R. L. Graham. *Ramsey Theory*. John Wiley & Sons, 1990.

[HI11] J. Hook and G. Isaak. Star-critical Ramsey numbers. *Discrete Applied Mathematics*, 159(5):328–334, 2011.

[JK07] T. Junttila and P. Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics*, pages 135–149. SIAM, 2007.

[JMW13] T. Jiang, K. G. Milans, and D. B. West. Degree Ramsey numbers for cycles and blowups of trees. *European Journal of Combinatorics*, 34(2):414–423, 2013.

[Li09] Y. Li. The multi-color Ramsey number of an odd cycle. *Journal of Graph Theory*, 62(4):324–328, 2009.

[MC12] A. Metodi and M. Codish. Compiling finite domain constraints to SAT with BEE. *Theory and Practice of Logic Programming*, 12(4-5):465–483, 2012.

[MP14] B. D. McKay and A. Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014.

[MR95] B. McKay and S. Radziszowski. $R(4,5) = 25$. *Journal of Graph Theory*, 19(3):309–322, 1995.

[MR97] B. D. McKay and S. Radziszowski. Subgraph counting identities and Ramsey numbers. *Journal of Combinatorial Theory, Series B*, 69(2):193–209, 1997.

[Pip08] A. Piperno. Search space contraction in canonical labeling of graphs (preliminary version). *CoRR*, abs/0804.4881, 2008.

[Rad94] S. Radziszowski. Small Ramsey numbers. *Electronic Journal of Combinatorics, Dynamic Surveys*, DS1, 1994. Revision #14, 2014.

[Rad11] S. Radziszowski. Ramsey numbers involving cycles. In Soifer [Soi11], pages 41– 62.

[Rad14] S. Radziszowski. Personal communication, 2014.

[Ram30] F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 1930.

[Soi11] A. Soifer. *Ramsey Theory: Yesterday, Today, and Tomorrow*, volume 285. Springer Science & Business Media, 2011.

[Spe94] J. Spencer. *Ten Lectures on the Probabilistic Method*, volume 64. SIAM, 1994.

[STT81] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125, Feb 1981.

[WSR15] Y. Wu, Y. Sun, and S. Radziszowski. Wheel and star-critical Ramsey numbers for quadrilateral. *Discrete Applied Mathematics*, 186:260–271, 2015.

[XSR09] X. Xu, Z. Shao, and S. Radziszowski. Bounds on some Ramsey numbers involving quadrilateral. *Ars Combinatoria*, 90:337–344, 2009.

[YR92a] Y. Yuansheng and P. Rowlinson. On extremal graphs without 4-cycles. *Utilitas Mathematica*, 41:204–210, 1992.

[YR92b] Y. Yuansheng and P. Rowlinson. On the third Ramsey numbers of graphs with five edges. *Journal of Combinatorial Mathematics and Combinatorial Computing*, (11):213–222, 1992.