DIMACS-DIMATIA International REU
Research Experience for Undergraduates 2013

June 2 – August 7 2013,

Piscataway, NJ, USA and Prague, Czech Republic

Martin Balko (ed.)

DIMACS/DIMATIA International REU 2013

Published by IUUK–ITI series 2014-604.
Centre of Excellence – Institute for Theoretical Computer Science,
Faculty of Mathematics and Physics, Charles University
Malostranské náměstí 25, 118 00, Praha 1, Czech Republic

Published by MATFYZPRESS,
Publishing House of the Faculty of Mathematics and Physics,
Charles University in Prague,
Sokolovská 83, 186 75 Prague 8, Czech Republic
as the 451st publication

Prague 2013

ISBN 978-80-7378-257-3

Preface

Charles University in Prague and particularly Department of Applied Mathematics (KAM), Computer Science Institute of Charles University (IUUK) and its international centre DIMATIA, are very proud that they are hosting one of the very few International REU programmes which are funded jointly by NSF and the Ministry of Education of Czech Republic (under the framework of Kontakt programmes ME 521, ME 886 and ME 09074). This programme is a star programme at both ends and it exists for more than a decade since 2001. Repeatedly, it has been awarded for its accomplishments and educational excellence.

The program of Kontakt on the Czech side was not renewed for year 2013 and thus the programme was financed jointly by Section Informatics of MFF and our grants CE-ITI P202/12/G061, ERCCZ LL1201 and SVV - 202-09/267313 (Discrete Models and Algorithms). We thank all the contributors and hope that the next year will bring us a stable support.

This booklet reports just the programme in 2013. I thank to Martin Balko, the Czech mentor of this year, for a very good work both during the programme itself and after.

Prague, October 17, 2013

Jaroslav Nešetřil

DIMACS/DIMATIA Research Experiences for Undergraduates (REU) is a joint program of the DIMATIA center, Charles University in Prague and DIMACS center, Rutgers University, New Jersey. This year's participants from Charles University were students Martin Koutecký, Karel Král, Jitka Novotná, Karel Tesař and Vojtěch Tůma. Their coordinator was Martin Balko, who participated in the scientific work, but mainly took care of organizing the DIMATIA part of the program. Together with more than thirty students from universities from all over the United States, they participated in the first part of the program, at Rutgers University of New Jersey in Piscataway, USA, from June 2nd to July 21st. Four American students were selected to join, together with their coordinator, the Czech students in the second part which took place at Charles University in Prague from July 23th to August 7th. The students were Kaleigh Clary, Elizabeth Field, Kevin Sung and Kevin Wong. The coordinator was Glen Wilson.

The first part of the program mainly consists of students solving open mathematical problems brought by their mentors. Students attended several lectures and they also participated in a trip to AT&T Labs which was organized by DIMACS. Here the students heard about applications of mathematics and computer science.

In Prague, the students attended a series of lectures given by professors mainly from the Department of Applied Mathematics and the Computer Science Institute of Charles University. They also had the opportunity to attend the Midsummer Combinatorial Workshop.

In addition to the scientific program, an important part of the REU is an intercultural experience. During the first part, an afternoon was dedicated to presentations of Czech Republic and cultures from which the American students come from. The students participated together in informal sport activities and sightseeing trips.

The students got important experiences with research and life abroad. For some of them, the program will certainly be an important milestone in their future scientific career.

This booklet presents the results of the Czech students stemming from the REU programme and reports of the American students about their visit to Prague. I would like to thank Josef Cibulka for providing the source files for this booklet.

Martin Balko

The participants of the Prague part of the programme.

Midsummer Combinatorial Workshop excursion to the Klementinum library, meeting at Nebozízek.

Contents

Ramsey Numbers of Ordered Graphs

Martin Balko[1] and Karel Král

1 Introduction

Ramsey theory is one of the most active areas of research within combinatorics. Its underlying philosophy is that every sufficiently large system contains a well-organized subsystem. The main result of this field is the Ramsey theorem which concerns edge-colored graphs. In this paper we derive some Ramsey-type results for graphs with ordered vertex sets which is a concept introduced by Milans et al. [25]. We also discuss the differences between ordered and unordered graphs in the view of Ramsey theory and we pose plenty of questions related to Ramsey numbers of ordered graphs.

Even though ordering of vertices may seem like an insignificant alternation in otherwise well known concept it can be seen that in the terms of Ramsey theory we can get substantially different results compared to the unordered case. The main goal of this paper is to understand the effects of vertex-orderings on the Ramsey numbers. Before stating our results we also mention some motivation examples and applications in which ordered hypergraphs can arise.

The concept of ordered graphs appeared earlier in the literature [25, 27], but we are not aware of any Ramsey-type results for such graphs except for the case of monotone paths and hyperpaths [6, 11, 16, 25, 26].

2 Preliminaries

In this section we state a notation of ordered hypergraphs introduced by Milans et al. [25]. Before doing so, we first mention some basic definitions related to hypergraphs and some fundamental results in Ramsey theory, as we use them later to point out the differences between the unordered and ordered case.

A *hypergraph* (also called a *set system*) is a pair $\mathcal{H} = (X, E)$ where X is a set of *vertices* and E is a set of non-empty subsets of X called *edges*. For a positive integer k, we say that a hypergraph $\mathcal{H}$ is *k-uniform* if each of

[1]The first author was supported by the grant SVV-2013-267313 (Discrete Models and Algorithms), and by the grant Grant Agency of Charles University, GAUK 1262213.

its edges contains exactly k vertices. If $k = 2$ then we speak about *graphs*. We consider only finite hypergraphs without loops or multiple edges in this paper. An edge-hypergraph *coloring* is a function $f\colon E \to C$ where C is a finite set of *colors*. A coloring with c colors is called a *c-coloring*.

We say that two hypergraphs $\mathcal{H}_1 = (X_1, E_1)$ and $\mathcal{H} = (X_2, E_2)$ are *isomorphic*, written $\mathcal{H}_1 \simeq \mathcal{H}_2$, if there is a one-to-one mapping $g\colon X_1 \to X_2$ which maps every edge of $\mathcal{H}_1$ to an edge of $\mathcal{H}_2$. A (hyper)graph $\mathcal{G} = (Y, F)$ is a *sub(hyper)graph* of $\mathcal{H} = (X, E)$ if $Y \subseteq X$ and $F \subseteq E$.

Ramsey's Theorem guarantees the existence of *Ramsey numbers* for edge-colored ordinary (unordered) hypergraphs. A hypergraph which contains all possible hyperedges is called *complete*. Let K_n^k denote the complete (unordered) k-uniform hypergraph on n vertices. For given positive integers c, k and r Ramsey's Theorem says that for sufficiently large n every c-coloring of edges of complete k-uniform hypergraph on n vertices contains a monochromatic copy of K_r^k as a subgraph. The minimum such n is called the *Ramsey number* and we denote it by $\mathbf{R}_k(K_r^k; c)$. For graphs we just use $\mathbf{R}(K_r; c)$ instead of $\mathbf{R}_2(K_r; c)$. Classical results of Erdős [12] and Erdős and Szekeres [13] give $2^{r/2} \leq \mathbf{R}(K_r; 2) \leq 2^{2r}$. Even though there have been many improvements on this bounds during the last sixty years (see [8] for example), the constant factors in these exponents remain the same.

Since every k-uniform hypergraph on r vertices is contained in K_r^k we can consider the following generalization of Ramsey numbers. Let $\mathcal{H}_1, \dots, \mathcal{H}_c$ be finite k-uniform hypergraphs and let c be a positive integer. Then Ramsey's Theorem implies that there exists a number $\mathbf{R}_k(\mathcal{H}_1, \dots, \mathcal{H}_c)$ such that every c-coloring of edges of a complete k-uniform hypergraph with at least $\mathbf{R}_k(\mathcal{H}_1, \dots, \mathcal{H}_c; c)$ vertices contains a monochromatic copy of $\mathcal{H}_i$ in color i for some $i \in \{1, 2, \dots, c\}$. If all the hypergraphs $\mathcal{H}_1, \dots, \mathcal{H}_c$ are isomorphic, we just write $\mathbf{R}_k(\mathcal{H}; c)$.

2.1 Ordered Hypergraphs

An *ordered hypergraph* is a pair $(\mathcal{H}, \prec)$ where $\mathcal{H} = (X, E)$ is a hypergraph and $\prec$ is a total ordering of its vertex set. The ordering $\prec$ is called a *vertex ordering*. Most of the properties of hypergraphs (vertex degrees, coloring, indistinguishable vertices and so on) can be defined the same for ordered hypergraphs as for hypergraphs, but the vertex orderings also bring some new additional properties.

For an ordered hypergraph $(\mathcal{H} = (X, E), \prec)$ and its vertices $x, y \in X$ we say that y is a *left neighbor* of x (*right neighbor*) if x and y are adjacent and

$y \prec x$ ($x \prec y$, respectively). We say that two ordered hypergraphs $(\mathcal{H}_1, \prec_1)$ and $(\mathcal{H}_2, \prec_2)$ are *isomorphic* if we have $\mathcal{H}_1 \simeq \mathcal{H}_2$ via a one-to-one mapping g which preserves the orderings. That is, for every $x, y \in X$, $x \prec_1 y$ implies $g(x) \prec_2 g(y)$. In such a case we write $(\mathcal{H}_1, \prec_1) \simeq (\mathcal{H}_2, \prec_2)$.

An ordered hypergraph $(\mathcal{G}, \prec_1)$ is an *ordered sub(hyper)graph* of $(\mathcal{H}, \prec_2)$ if $\mathcal{G}$ is a sub(hyper)graph of $\mathcal{H}$ and $\prec_1$ is a suborder of $\prec_2$. Having a notation of subgraphs we can introduce *Ramsey numbers of ordered hypergraphs.* For given finite ordered k-uniform hypergraphs $(\mathcal{H}_1, \prec_1), \dots, (\mathcal{H}_c, \prec_c)$ we denote as $\mathbf{RO}_k((\mathcal{H}_1, \prec_1), \dots, (\mathcal{H}_c, \prec_c))$ the least number n such that every c-coloring of edges of a complete ordered k-uniform hypergraph with at least n vertices contains a monochromatic copy of some $(\mathcal{H}_i, \prec_i)$ in color i as an ordered subgraph. If all the given ordered hypergraphs are isomorphic, we may just write $\mathbf{RO}_k((\mathcal{H}, \prec); c)$ and, again, in the case of graphs we omit k.

In complete hypergraphs, all orderings of K_r^k are the same up to isomorphism and thus there is no difference between unordered and ordered cliques in a given hypergraph. We therefore obtain the following.

Observation 2.1. *For arbitrary positive integers $r_1, \dots, r_c$, k, c and total orderings $\prec_1, \dots, \prec_c$ we have*

$$\mathbf{R}_k(K_{r_1}^k, \dots, K_{r_c}^k) = \mathbf{RO}_k((K_{r_1}^k, \prec_1), \dots, (K_{r_c}^k, \prec_c)).$$

$\square$

A simple consequence of this observation is that ordered Ramsey numbers exist for an arbitrary collection of ordered k-uniform hypergraphs, since every ordered k-uniform hypergraph $(\mathcal{H}, \prec)$ on n vertices is a subgraph of $(K_n^k, \prec)$.

Corollary 2.2. *Let c be a positive integer and let $(\mathcal{H}_1, \prec_1), \dots, (\mathcal{H}_c, \prec_c)$ be an arbitrary collection of ordered k-uniform hypergraphs, then the number $\mathbf{RO}_k((\mathcal{H}_1, \prec_1), \dots, (\mathcal{H}_c, \prec_c))$ exists and it is finite.* $\square$

Another simple fact is that

$$\mathbf{R}_k(\mathcal{H}_1, \dots, \mathcal{H}_c) \leq \mathbf{RO}_k((\mathcal{H}_1, \prec_1), \dots, (\mathcal{H}_c, \prec_c))$$

always holds, since if $\mathcal{H}_i$ is not contained in a given graph as (unordered) subgraph, it cannot be there as an ordered subgraph.

For an ordering $\prec$ we use $\prec^{-1}$ to denote the *reversed ordering*, i.e. $x \prec y$ if and only if $y \prec^{-1} x$. Sometimes we want to see how Ramsey numbers behave with increasing number of vertices. To be able to do this for ordered hypergraphs, we need to introduce so called *ordering schemes* which are vertex orderings uniquely determined by the given hypergraph $\mathcal{H}$. That is, there are specific rules which tell us how to order vertices of $\mathcal{H}$ with increasing number of vertices. For example, a k-uniform *monotone hyperpath* $(P_r^k, \lhd_{mon})$ is a k-uniform hypergraph with vertices $v_1 \lhd_{mon} \ldots \lhd_{mon} v_r$ and $r-k+1$ edges, each consisting of k consecutive vertices (see Figure 1 for an example). Throughout the paper we use a symbol $\lhd$ instead of $\prec$ to emphasize the fact that the vertex ordering follows some ordering scheme.

a) b)

Figure 1: Examples of 2-uniform and 3-uniform monotone hyperpaths on seven vertices.

2.2 Motivation

In this subsection we show various examples in which Ramsey-type problems on ordered hypergraphs arise. Some of the results mentioned are nowadays classical statements, but some other ones appeared only recently. In any case these examples should help the reader to get used to the notation and show some interesting applications. Some results and definitions are also used later in this paper.

Erdős-Szekeres Lemma. This well-known statement says that for a given $k \in \mathbb{N}$ one can find a decreasing or increasing subsequence of length k in every sequence of at least $(k-1)^2+1$ distinct integers and that this bound is sharp. This lemma can be proved using many approaches (see [29] for the list of proofs), but one can observe that it is basically a special case of far more general Ramsey-type result on ordered graphs.

Given such a sequence $S = (s_1, \ldots, s_n)$ we can construct an ordered graph $(K_n, \prec)$ with vertex set S and ordering chosen according to the positions in S. That is for $s_i, s_j \in S$ we have $s_i \prec s_j$ if $i < j$. Then we 2-color the edges of this graph in the following manner: an edge $\{s_i, s_j\}$ is red if $s_i < s_j$ and blue otherwise. It is not difficult to see that red monotone paths correspond to increasing and blue monotone pats to decreasing subsequences of the same length in this graph. The rest follows from the following result of Choudum and Ponnusamy (see Milans et al. [25] for the proof in the language of ordered Ramsey theory).

Proposition 2.3 (S. A. Choudum and B. Ponnusamy [6])**.** *For monotone ordered paths* $(P_{r_1}, \lhd_{mon}), \ldots, (P_{r_c}, \lhd_{mon})$*, we have*

$$\mathbf{RO}((P_{r_1}, \lhd_{mon}), \ldots, (P_{r_c}, \lhd_{mon})) = 1 + \prod_{i=1}^{c}(r_i - 1).$$

Note that the decreasing and increasing subsequences actually correspond to monochromatic cliques in our complete colored graph, since the obtained coloring is *transitive*. That is, if the hyperedges $\{x_1, \ldots, x_k\}$ and $\{x_2, , \ldots, x_{k+1}\}$, $x_1 \prec \ldots \prec x_{k+1}$, have the same color, then every hyperedge $\{x_{i1}, \ldots, x_{ik}\}$, $\{i_1, \ldots, i_k\} \subseteq [k+1]$, is of the same color.

Integer Partitions. Another example shows a surprising connection between Ramsey theory of ordered hypergraphs and high-dimensional integer partitions. A *d-dimensional partition* is a d-dimensional (hyper)matrix A of nonnegative integers such that A is decreasing in each line. That is $A_{i_1,\ldots,i_t,\ldots,i_d} \geq A_{i_1,\ldots,i_t+1,\ldots,i_d}$ for every possible $i_1, , \ldots, i_d$ and $1 \leq t \leq d$. For example if $d = 1$, then we have a decreasing sequence of nonnegative integers $a_1 \geq a_2 \geq \ldots$ which is called a *line partition*.

Let $P_d(n)$ denote the number of $n \times n$ d-dimensional partitions with entries from $\{0, \ldots, n\}$. Observe that in the case $d = 1$ we have $P_1(n) = \binom{2n}{n}$, since we can think of such line partition as a lattice path in $\mathbb{Z}^2$ starting at $(0, n)$ and ending at $(n, 0)$. It is also known, although it is much more difficult to prove it, that $P_2(n) = \prod_{1\leq i,j,k\leq n} \frac{i+j+k-1}{i+j+k-2}$.

The following theorem was proved by Moshkovitz and Shapira [26] last year and it establishes a close connection between integer partitions and Ramsey numbers of monotone 3-uniform hyperpaths.

Theorem 2.4 (G. Moshkovitz and A. Shapira [26])**.** *For every* $c \geq 2$ *and* $r \geq 2$ *we have* $\mathbf{RO}_3((P_r^3, \lhd_{mon}); c) = P_{c-1}(r-2) + 1$.

We use this result in the following motivation example which concerns a classical result in Ramsey theory and combinatorial geometry. Studying of Ramsey numbers of monotone hyperpaths brought attention of many researchers in recent years, see [11, 16, 25].

Happy Ending Problem. One of the original results that led to the development of Ramsey theory was the following statement, also called Happy Ending Problem.

Theorem 2.5 (P. Erdős and G. Szekeres [13]). *For $k \in \mathbb{N}$ there exists a number $\mathbf{ES}(k)$ such that every set of at least $\mathbf{ES}(k)$ points in $\mathbb{R}^2$ in general position contains k points in convex position.*

As noted by Erdős and Szekeres, it can be shown that this result is implied by Ramsey theorem applied to 4-uniform hypergraphs, but the upper bound for $\mathbf{ES}(k)$ obtained by this approach is astronomically large. In the original paper of Erdős and Szekeres much more reasonable bound $\mathbf{ES}(k) \leq \binom{2k-4}{k-2} + 1$ is also shown. Even though there is a better upper bound now (Valtr and Tóth [31] hold current record $\mathbf{ES} \leq \binom{2k-5}{k-2} + 1$), Moshkovitz and Shapira [26] showed that Ramsey theory for ordered 3-uniform hypergraphs can be used to derive $\mathbf{ES}(k) \leq \binom{2k-4}{k-2} + 1$.

Suppose that we have a set $S \subset \mathbb{R}^2$ of $n \geq \mathbf{ES}(k)$ points in general position (that is, no three points are collinear). Let $(K_n^3, \prec)$ be an ordered 3-uniform hypergraph with vertex set S where for two vertices $x, y \in S$ $x \prec y$ holds if their x-coordinates satisfy $X(x) < X(y)$. Then we color an edge $\{x, y, z\}$ red if the triangle xyz is oriented counterclockwise and blue otherwise.

Note that then a monochromatic monotone 3-uniform path in this graph corresponds to a (special) convex k-gon in S (which is sometimes called a *k-cup* or a *k-cap*). The previous result of Moshkovitz and Shapira then gives us the desired upper bound $\mathbf{ES}(k) \leq \binom{2k-4}{k-2} + 1$. Note that, similarly as in the first motivation example, the obtained 2-coloring is transitive.

A long standing conjecture of Erdős and Szekeres says that $\mathbf{ES}(k) = 2^{k-2} + 1$. The language of ordered hypergraphs also allows us to state the generalized version of this conjecture introduced by Peters and Szekeres [30]. Let $L \subseteq \{2, \ldots, k-1\}$ for some $k \geq 3$ and let $(PP_L, \prec)$ denote an ordered 3-uniform hypergraph with vertex set $\{v_1, \ldots, v_k\}$ consisting of a red hyperpath P on vertices v_i with $i \in \{1, k\} \cup L$ and a blue hyperpath Q on vertices v_i with $i \in \{1, \ldots, k\} \setminus L$ where $v_i \prec v_j$ if $i < j$. Note that for $L = \emptyset$ the path P does not have any edges and the same holds for Q and $L = \{2, \ldots, k-1\}$.

Conjecture 2.6 (G. Szekeres and L. Peters [30]). *For $k \geq 3$ and for every complete ordered 3-uniform hypergraph $(\mathcal{H}, <)$ on $2^{k-2} + 1$ vertices with edges colored red and blue there exists $L \subseteq \{2, \ldots, k-1\}$ such that $(\mathcal{H}, <)$ contains $(PP_L, \prec)$ as an ordered subgraph.*

Using computer experiments Peters and Szekeres verified this conjecture for $k \leq 5$ and showed a construction which achieves the same lower bound for general k. This construction also follows from the construction of Erdős and Szekeres.

Extremal Problems on Matrices. The last motivation example shows a connection between extremal theory of $\{0,1\}$-matrices (see [4, 17], for example) and ordered Turán numbers of ordered bipartite graphs. This is particularly useful for us, as we use some results from this area later in this paper, see Section 3.2.

A $\{0,1\}$-matrix A *contains* an $r \times s$ submatrix M if A contains a submatrix M which has ones on all the positions where M does. A matrix A *avoids* M if it does not contain M. The *extremal function* of M is the maximum number $\mathbf{ex}_M(m,n)$ of 1-entries in an $m \times n$ $\{0,1\}$-matrix avoiding M.

Consider a complete bipartite graph $K_{r,s}$ with the following vertex ordering which we denote as $\lhd_{sep}$: if $K_{r,s}$ is a complete bipartite graph with vertices divided into classes A and B of size r and s respectively, then for every $x \in A$ and $y \in B$ we have $x \lhd_{sep} y$. The rest of $\lhd_{sep}$ can be completed arbitrarily, as the vertices from the same color class are indistinguishable. See Figure 2 for an example. Note that $(K_{r,s}, \lhd_{sep})$ does not contain $(P_l, \lhd_{mon})$, $l \geq 3$, as a subgraph.

Let $(G = (A \cup B, E), \prec)$, $|A| = r$ and $|B| = s$, be a a subgraph of $(K_{r,s}; \lhd_{sep})$. Then $(G, \prec)$ corresponds to a $r \times s$ $\{0,1\}$-matrix $M(G, \prec)$ where the i-th row represents the i-th vertex of A in $\prec$ (the same holds for columns and vertices in B) and $M(G, \prec)_{i,j} = 1$ if $v_i \in A$ and $v_j \in B$ are adjacent and 0 otherwise. It is easy to see that the Turán number of $(G, \prec)$ in $(K_{m,n}, \lhd_{sep})$ is exactly the value of $\mathbf{ex}_{M(G,\prec)}(m,n)$.

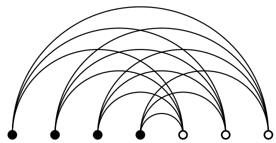


Figure 2: The ordered complete bipartite graph $(K_{4,3}, \lhd_{sep})$ with distinguished vertex classes.

2.3 Our Results

The main field of our interest are graphs and the effects of vertex orderings on Ramsey numbers of various classes of graphs. For many of these classes the unordered Ramsey numbers were resolved a long time ago, but adding the vertex orderings can lead to completely different results. Some examples of differences between ordered and unordered Ramsey numbers can be seen already in the motivation examples.

For example, Proposition 2.3 shows that there is a graph G and its vertex ordering $\prec$ such that the unordered Ramsey number $\mathbf{R}(G;c)$ and the ordered Ramsey number $\mathbf{RO}((G,\prec);c)$ differ in an asymptotically relevant manner, since the unordered Ramsey number $\mathbf{R}(P_r;c)$ for paths is linear with respect to r while $\mathbf{RO}((P_r,\lhd_{mon});c)$ is quadratic.

Even larger gap can be obtained concerning hypergraphs of higher uniformity. It is known that Ramsey numbers $\mathbf{R}_k(\mathcal{H};2)$ of sparse unordered k-uniform hypergraphs $\mathcal{H}$ are linear with respect to the number of vertices $\mathcal{H}$. Formally, for positive integers Δ and k, there exists a constant $C(\Delta,k)$ such that if $\mathcal{H}$ is a k-uniform hypergraph with r vertices and maximum degree Δ, then $\mathbf{R}_k(H;2) \leq C(\Delta,k)r$ (see [9]). In contrast to this result, Theorem 2.4 together with the fact $P_1(n) = \binom{2n}{n}$ gives us $\mathbf{RO}_3((P_r^3,\lhd_{mon});2) = \binom{2r-4}{r-2}$. Thus we see that there are 3-uniform hypergraphs $\mathcal{H}$ and vertex orderings $\prec$ such that $\mathbf{RO}_3((\mathcal{H},\prec);2)$ grows exponentially with the number of vertices of $\mathcal{H}$ while $\mathbf{R}_3(\mathcal{H};2)$ remains linear.

In the first part of this paper we try to derive Ramsey numbers for various classes of ordered graphs: stars, paths and cycles. First, we show that Ramsey numbers of all ordered stars are linear with respect to the number of vertices.

Theorem 2.7. *For positive integers c and $r_1,\dots,r_c$ and for a collection of ordered stars $(K_{1,r_1-1},\prec_1),\dots,(K_{1,r_c-1},\prec_c)$ there is a constant $C=C(c)$ such that*

$$\mathbf{RO}((K_{1,r_1-1},\prec_1),\dots,(K_{1,r_c-1},\prec_c)) \leq C\max\{r_1,\dots,r_c\}.$$

Considering the multi-colored case we find a graph G and its vertex orderings $\prec$ and $\prec'$ such that Ramsey numbers for $(G,\prec)$ and $(G,\prec')$ differ exponentially in the number of colors (Proposition 3.4). This result is an example of the fact that long monotone paths as ordered subgraphs affect ordered Ramsey numbers significantly. We also derive some exact formulas for Ramsey numbers of specific ordered stars.

In the following section we discuss ordered cycles. First, we show Ramsey numbers for all possible orderings of C_4 (Proposition 3.10). Then we derive the exact formula for ordered Ramsey numbers of so called *monotone cycles* $(C_n,\lhd_{mon})$ which consist of a monotone path on vertices $v_1 \lhd_{mon} \dots \lhd_{mon}$

v_n with the edge $\{v_1, v_n\}$ added. See an example of a *monotone cycle* in Figure 3.

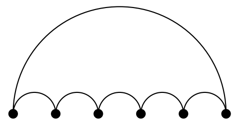


Figure 3: The monotone cycle $(C_6, \lhd_{mon})$.

Theorem 2.8. *For integers $r \geq 2$ and $s \geq 2$ we have*

$$\mathbf{RO}((C_r, \lhd_{mon}), (C_s, \lhd_{mon}); 2) = 2rs - 3r - 3s + 6.$$

As a consequence of this theorem we obtain tight bounds for so called geometric and convex geometric Ramsey numbers of cycles which which were introduced by Károlyi et al. [21, 22]. The definitions as well as the result, Corollary 3.12, are mentioned in Section 3.3.

In Section 3.2 we show, using specific ordered paths, that there are graphs for which different ordering schemes can affect ordered Ramsey numbers in an asymptotically relevant term.

Then we derive some general lower bounds for ordered Ramsey numbers. We apply a probabilistic approach showing a general lower bound for ordered Ramsey numbers which depends on the density of the given graph. See Proposition 4.1 which implies the following assertion.

Proposition 2.9. *Let $c \geq 2$ be a positive integer and let $(G, \prec)$ be an ordered graph with n vertices and $n^{1+\varepsilon}$ edges, $\varepsilon > 0$. Then $\mathbf{RO}((G, \prec); c) = \Omega(nc^{n^{\varepsilon}})$ holds.*

Then we construct a graph with maximum degree three whose ordered Ramsey number grows faster than quadratically with respect to its size (Theorem 4.3). This result is in contrast with unordered Ramsey theory, where, as we previously discussed, it is known that graphs with bounded degrees have linear Ramsey numbers.

At the end of the paper, we discuss several new open problems and possible ways for further research.

3 Ordered Ramsey Numbers for Specific Classes of Graphs

In this section we compute Ramsey numbers for various classes of ordered graphs such as stars, cycles and paths. We use some of the results later to derive more general bounds. We also compare the formulas obtained and bounds with known Ramsey numbers of unordered graphs and discuss relations between those cases.

3.1 Stars

A *star* is a complete bipartite graph $K_{1,r-1}$. Since there are only two groups of indistinguishable vertices, the position of the central vertex determines the ordering uniquely up to isomorphism. Ramsey numbers of unordered stars are known exactly [2] for a long time now and they are given by

$$\mathbf{R}(K_{1,r-1}; c) = \begin{cases} c(r-2)+1 & \text{if } c \equiv r-1 \equiv 0 \pmod{2}, \\ c(r-2)+2 & \text{otherwise.} \end{cases}$$

Using a simple observation a similar formula can be derived for ordered stars $(K_{1,r-1}, \lhd_{min})$ where $\lhd_{min}$ is a vertex ordering in which the central vertex is the minimum element.

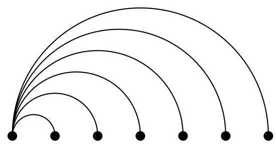


Figure 4: The ordered star $(K_{1,6}, \lhd_{min})$.

Observation 3.1. *For positive integers $c, r_1, \ldots, r_c$ we have*

$$\mathbf{RO}((K_{1,r_1-1}, \lhd_{min}), \ldots, (K_{1,r_c-1}, \lhd_{min})) = 2(1-c) + \sum_{i=1}^{c} r_i.$$

Proof. Assume that we have a complete ordered graph $(K_n, \prec)$ with $n \geq 2(1-c) + \sum_{i=1}^{c} r_i$ vertices and c-colored edges. Then, according to the pigeon-hole principle, the first vertex in $\prec$ has at least $r_i - 1$ right neighbors in color i. This forms a monochromatic copy of $(K_{1,r_i-1}, \lhd_{min})$.

On the other hand, we can construct a graph on $1-2c+\sum_{i=1}^{c} r_i$ vertices which does not contain any forbidden star. It suffices to divide the right neighbors of each vertex v into c parts where the i-th part has size at most $r_i - 2$ and each of its vertices is adjacent to v with an edge colored with i. $\square$

Thus in the case $r_1 = \cdots = r_c = r$ the ordered Ramsey numbers are almost the same as the unordered ones. They differ by one only if $c \equiv r - 1 \equiv 0 \pmod 2$. One can observe that a complete characterization of ordered Ramsey numbers for two arbitrary ordered stars is implied by results of Choudum and Ponnusamy [6]. For integers x, y, let $\prec_{r,s}$ denote an ordering of star on $r + s - 1$ vertices where $r - 1$ vertices are to the left of the central vertex and $s - 1$ vertices are to the right.

Theorem 3.2 ([6]). *For any two integers $r_1, r_2 \geq 2$ we have*

$$\mathbf{RO}((K_{1,r_1-1}, \lhd_{min}), (K_{1,r_2-1}, \lhd_{min}^{-1})) = \\ \lfloor -1 + \sqrt{1 + 8(r_1 - 2)(r_2 - 2)}/2 \rfloor + r_1 + r_2 - 2.$$

Moreover, for arbitrary stars K_{1,r_1-1}, K_{1,r_1+s_1-2} and K_{1,r_2+s_2-2} we have

$$\mathbf{RO}((K_{1,r_1-1}, \lhd_{min}), (K_{1,r_2+s_2-2}), \prec_{r_2,s_2}) = \\ \mathbf{RO}((K_{1,r_1-1}, \lhd_{min}), (K_{2,r_2-1}, \lhd_{min}^{-1})) + r_1 + s_2 - 3$$

and

$$\mathbf{RO}((K_{1,r_1+s_1-2}, \prec_{r_1,s_1}), (K_{1,r_2+s_2-2}, \prec_{r_2,s_2})) = \\ \mathbf{RO}((K_{1,r_1-1}, \lhd_{min}^{-1}), (K_{1,r_2+s_2-2}, \prec_{r_2,s_2})) + \\ \mathbf{RO}((K_{1,s_1-1}, \lhd_{min}), (K_{1,r_2+s_2-2}, \prec_{r_2,s_2})) - 1.$$

We further show that for any vertex ordering Ramsey numbers of ordered stars also remain linear with respect to the size of the stars for an arbitrary number of colors. That is, Theorem 2.7 which we restate for convenience:

Theorem 3.3. *Let $c, r_1, \ldots, r_c$ be positive integers and let $(K_{1,r_1-1}, \prec_1)$, $\ldots, (K_{1,r_c-1}, \prec_c)$ be ordered stars. Then there is a constant $C = C(c)$ such that*

$$\mathbf{RO}((K_{1,r_1-1}, \prec_1), \ldots, (K_{1,r_c-1}, \prec_c)) \leq C \max\{r_1, \ldots, r_c\}.$$

Proof. Let $r = \max\{r_1, \ldots, r_c\}$ and let $(K_n, \prec)$ be an ordered complete graph on $n = Cr$ vertices with edges colored by $\{1, 2, \ldots, c\}$ where C is a sufficiently large constant. Let A_0 be the vertex set of $(K_n, \prec)$. We want to find a vertex with $r-1$ left and $r-1$ right neighbors of the same color i. Then $(K_{1,r_i-1}, \prec_i)$ is clearly contained in $(K_n, \prec)$. So suppose for a contradiction that there is no star $(K_{1,r_i-1}, \lhd_{min})$ of color i in $(K_n, \prec)$.

Note that each vertex which is at least $(c(r-1)+2)$-th in the ordering $\prec$ (taken from left) has, according to the pigeon-hole principle, at least $r-1$ left neighbors of the same color. Thus we have at least $Cr - c(r-1) - 1$ vertices with at least $r-1$ monochromatic left neighbors. We consider a set A_1 of vertices which have at least $r-1$ left neighbors of color 1. Without loss of generality we may assume that $|A_1| \geq (Cr - c(r-1) - 1)/c$.

From the assumption there is no vertex in A_1 with at least $r-1$ right neighbors of color 1, as otherwise we would have $(K_{1,r_1-1}, \prec_1)$ of color 1. Thus between vertices in A_1 there is less than $(r-1)|A_1|$ edges of color 1, since every one of them is counted for its left endpoint. Also we see that A_1 contains at least $(|A_1| - (c(r-1)-1))/c$ vertices which have at least $r-1$ right neighbors in A_1 all of the same color i (without loss of generality, let $i = 2$). We denote this set as A_2. From the assumption the vertices in A_2 have less than $r-1$ left neighbors of color 2 in $(K_n, \prec)$ and thus there is less than $(r-1)|A_2|$ edges of color 2 (and 1, since $A_2 \subseteq A_1$) between vertices in A_2.

We repeat this process analogously, bounding the number of edges of colors $1, \ldots, i$ in A_i by $(r-1)|A_i|$ and keeping $|A_i| \geq (|A_{i-1}| - c(r-1)-1)/c$ for $i \geq 1$. After all colors are processed we get, summing over all colors, that the number of all edges is strictly less than $c(r-1)|A_c|$. The total number of edges connecting vertices from A_i is exactly $\binom{|A_i|}{2}$. Altogether we have obtained $|A_c|(|A_c|-1)/2 < c(r-1)|A_c|$ which can be rewritten as $|A_c| < 2c(r-1)+1$. However $|A_c| = \Omega(Cr/c^c)$ and thus we can choose C large enough so that the upper bound on A_c does not hold and obtain a contradiction. □

Since we know that Ramsey numbers for unordered stars and for stars ordered according to $\lhd_{min}$ are linear even with respect to the number of colors, one might ask if this is a case also for other vertex orderings and if the upper bound from the previous theorem is not too weak. The following proposition shows that this is not the case, since the situation there turns out to be substantially different from the one for $\lhd_{min}$. Even for orderings of stars in which the central point has only a single left neighbor Ramsey numbers grow exponentially with respect to the number of colors. A similar construction as been known for the paths P_3.

Proposition 3.4. *Let c, d, $r_1, \ldots, r_c$ be positive integers, $(G_1, \prec_1), \ldots, (G_c, \prec_c)$ be ordered graphs such that $(P_d, \triangleleft_{mon}) \subseteq (G_i, \prec_i)$ and $|V(G_i)| = r_i$ for every $i = 1, \ldots, c$. Then we have*

$$\mathbf{RO}((G_1, \prec_1), \ldots, (G_c, \prec_c)) > (d-1)^{c-1}(\max\{r_1, \ldots, r_c\} - 1).$$

Proof. Let $(K_n, \prec)$ be an ordered complete graph on vertices $v_1 \prec \ldots \prec v_n$ where $n = (d-1)^{c-1}(r-1)$ and $r = \max\{r_1, \ldots, r_c\}$. Without loss of generality, let $r = r_1$. We color edges of $(K_n, \prec)$ with c colors from $\{1, \ldots, c\}$ such that it does not contain a monochromatic copy of any $(G_i, \prec_i)$, $i = 1, \ldots, c$.

The construction of the coloring is done by induction on c. For $c = 1$ we have a monochromatic clique of color 1 with $r-1$ vertices. Such a clique cannot contain even any unordered monochromatic G_i. For $c > 1$ we color the cliques on vertices which are divided into $(d-1)$ consecutive blocks each of size $(d-1)^{(c-2)}(r-1)$ using $d-1$ colorings from the previous step. Then we use color c to color all edges between those $d-1$ cliques to obtain a c-coloring of all edges. See Figure 5.

Using the inductive hypothesis it suffices to show that this coloring does not contain $(G_r, \prec_c)$ in color c. Since $(P_d, \triangleleft_{mon})$ is an ordered subgraph of $(G_r, \prec_c)$, while $(P_d, \triangleleft_{mon})$ is not contained in the ordered complete $(d-1)$-partite graph which is induced by the edges colored with c, we get the rest. $\square$

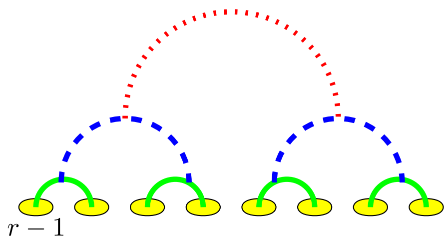


Figure 5: The construction in the proof of Proposition 3.4 for $d = 3$ and $c = 4$.

Corollary 3.5. *Let c and $r_1, \dots, r_c$ be positive integers and let $\prec_i \neq \lhd_{min}$, $\lhd_{min}^{-1}$ be a vertex ordering of K_{1,r_i-1} where $i = 1, \dots, c$. Then we have*

$$\mathbf{RO}((K_{1,r_1-1}, \prec_1), \dots, (K_{1,r_c-1}, \prec_c)) > 2^{c-1}(\max\{r_1, \dots, r_c\} - 1).$$

□

Corollary 3.6. *Let c and $r_1, \dots, r_c$ be positive integers and let $(G_1, \prec_1), \dots, (G_c, \prec_c)$ be ordered graphs on $r_1, \dots, r_c$ vertices respectively. If no G_i is bipartite, then we have*

$$\mathbf{RO}((G_1, \prec_1), \dots, (G_c, \prec_c)) > 2^{c-1}(\max\{r_1, \dots, r_c\} - 1).$$

Proof. According to Proposition 3.4 it suffices to show that $(P_3, \lhd_{min})$ is contained in every given ordered graph. Since each G_i is not bipartite, it contains an odd cycle. Now it is easy to observe that in any ordering of an odd cycle there is always a monotone path with three vertices. □

3.2 Paths and Matchings

Before discussing ordered Ramsey theory for paths, we, again, recall results for unordered Ramsey numbers of paths. For two colors the problem of finding an exact formula for $\mathbf{R}(P_r, P_s)$ has been settled by Gerensér and Gyárfás [18] who showed that for $2 \leq r \leq s$

$$\mathbf{R}(P_r, P_s) = s - 1 + \left\lfloor \frac{r}{2} \right\rfloor$$

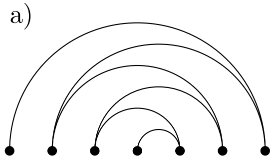
holds. The multi-color case turned out to be more difficult, but some partial results are known (see [15, 20], for an example).

The following natural question arises in ordered Ramsey theory: is there a graph G with two vertex ordering schemes $\lhd$ and $\lhd'$ such that $\mathbf{RO}((G, \lhd); 2)$ and $\mathbf{RO}((G, \lhd'); 2)$ differ in an asymptotically relevant term with respect to the number of vertices of G? Using a specific ordering of the path P_r we show that this is indeed the case.

We know (see Proposition 2.3) that Ramsey numbers for ordered paths $(P_r, \lhd_{mon})$ can grow quadratically with respect to r. Let us define another ordering scheme of a path. If P_r is a path with vertices $v_1, \dots, v_r$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{r-1}, v_r\}$, then an *alternating path* $(P_r, \lhd_{alt})$ is an ordered path where $v_1 \lhd_{alt} v_3 \lhd_{alt} v_5 \lhd_{alt} \dots \lhd_{alt} v_r \lhd_{alt} v_{r-1} \lhd_{alt} v_{r-3} \lhd_{alt} \dots \lhd_{alt} v_2$ for r odd and $v_1 \lhd_{alt} v_3 \lhd_{alt} v_5 \lhd_{alt} \dots \lhd_{alt} v_{r-1} \lhd_{alt} v_r \lhd_{alt} v_{r-2} \lhd_{alt}$

$\ldots \lhd_{alt} v_2$ for r even. That is $(P_r, \lhd_{alt}) \subseteq (K_{\lceil r/2 \rceil, \lfloor r/2 \rfloor}, \lhd_{sep})$. Note that an alternating path is a subgraph of every complete ordered bipartite graph with sufficiently many vertices and that similar statement does not hold for monotone paths. An example of an alternating path is found in Figure 6, part a).

a)



b)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 6: An ordered path $(P_7, \lhd_{alt})$ and its corresponding matrix $M(P_r, \lhd_{alt})$.

Proposition 3.7. *For every positive integer $r > 2$ we have*

$$2r - 2 \leq \mathbf{RO}((P_r, \lhd_{alt}); 2) \leq (4r - 3 + \sqrt{8r^2 - 8r - 7})/2.$$

Thus $\mathbf{RO}(P_r, \lhd_{alt}; 2)$ remains linear with respect to r although they are not precise. To derive the upper bound we use a result from extremal theory of $\{0,1\}$-matrices which was mentioned in the motivation (Section 2.2).

The following definitions are taken from [4]. We say that a $r \times s$ matrix M is *minimalist* if $\mathbf{ex}_M(m,n) = (s-1)m + (r-1)n - (r-1)(s-1)$. If the matrix M' was created from a matrix M by adding a new row (or a column) as the new first or last row (column) and this new row (column) contains a single 1-entry next to a 1-entry of M, then we say the M' was created by an *elementary operation* from M.

Lemma 3.8 (Z. Füredi and P. Hajnal [17])**.** *Let M be an $r \times s$ minimalist matrix and let M' be an $r' \times s'$ nonempty matrix obtained from M by applying several elementary operations. Then M' is minimalist.*

Proof of proposition 3.7. For the lower bound we color the edges $\{v_i, v_j\}$ in $(K_{2r-3}, \prec)$ red if $|i-j|$ is even and blue otherwise. Suppose that there is a red copy of $(P_r, \lhd_{alt})$ in our coloring. Then the number of vertices between the first and last one in the alternating path is at least $2r-4$ which is, together with the first and last one, more than the total number of vertices. An analogous argument works for a blue copy of $(P_r, \lhd_{alt})$.

For the upper bound we find a monochromatic copy of $(P_r, \lhd_{alt})$ in a given edge 2-colored graph $(K_{\lceil n/2\rceil,\lfloor n/2\rfloor}, \lhd_{sep})$. Suppose that at least one half of edges is colored red and consider only such edges. Note that $(P_r, \lhd_{alt})$ is an ordered subgraph of $(K_{\lceil r/2\rceil,\lfloor r/2\rfloor})$ and thus we can consider the $\lceil r/2\rceil \times \lfloor r/2\rfloor$ $\{0,1\}$-matrix $M(P_r, \lhd_{alt}) = M$ introduced in the motivation. An example of such matrix can be found in Figure 6, part b). By Lemma 3.8 all such matrices are minimalist.

Therefore $\mathbf{ex}_M(\lceil n/2\rceil, \lfloor n/2\rfloor) = (\lfloor r/2\rfloor - 1)\lceil n/2\rceil + (\lceil r/2\rceil - 1)\lfloor n/2\rfloor - (\lceil r/2\rceil - 1)(\lfloor r/2\rfloor - 1)$ and this is at most $1/4(2rn + 4r - 3n - 4 - r^2)$. Thus every $K_{\lceil n/2\rceil,\lfloor n/2\rfloor}$ which does not contain $(P_r, \lhd_{alt})$ as a subgraph must have at most this many edges. On the other hand our graph formed by red edges has at least $1/2\lceil n/2\rceil \cdot \lfloor n/2\rfloor \geq n(n-1)/8$ edges. Thus to avoid $(P_r, \lhd_{alt})$ the inequality

$$1/4(2rn + 4r - 3n - 4 - r^2) \geq n(n-1)/8$$

must hold and consequently we obtain $n \leq (4r - 5 + \sqrt{8r^2 - 8r - 7})/2$ and the result follows. $\square$

There is still a place for improvement, as the multiplicative factor is between 2 and $2+\sqrt{2}$. Computer experiments indicate that the right values of $\mathbf{RO}(P_r, \lhd_{alt}; 2)$ could be of the from $\lfloor (r-2)\frac{1+\sqrt{5}}{2}\rfloor + r$. See Table 1.

r	2	3	4	5	6	7	8	9	10	11
$\mathbf{RO}((P_r, \lhd_{alt}); 2)$	2	4	7	9	12	15	17	≥ 20	≥ 22	≥ 25

Table 1: Estimates for Ramsey numbers $\mathbf{RO}((P_r, \lhd_{alt}); 2)$ for $r \leq 13$.

If we consider orderings $\lhd_{mon}$ and $\lhd_{sep}$ and compare the previous result with Proposition 2.3, we see that there is a graph G on n vertices and ordering schemes $\lhd$ and $\lhd'$ of G such that $\mathbf{RO}((G, \lhd); 2)/\,\mathbf{RO}((G, \lhd'); 2) = \Omega(n)$.

Similar result can be derived if we consider *matching* M_n, which is a graph on n vertices consisting of $\lfloor n/2\rfloor$ disjoint pairs of edges, and the two orderings from Figure 7. Again, using a coloring similar to the one from Proposition 2.3 we see that ordered Ramsey number of the first ordered matching grows quadratically with respect to n while one can observe that for the second one the ordered Ramsey number remains linear in n.

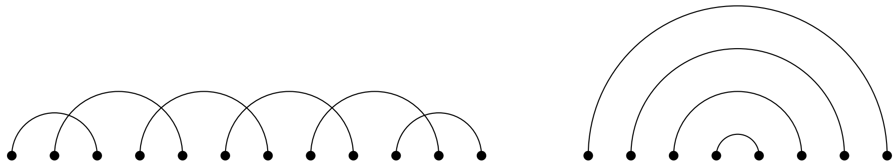


Figure 7: Two orderings of M_n with asymptotically different ordered Ramsey numbers.

For general ordered paths not much is known currently. In [5] J. Cibulka et al. showed that for every ordered path $(P_r, \prec)$ and clique K_s we have

$$\mathbf{RO}((P_r, \prec), (K_s, \prec')) \le 2^{\lceil \log_2(s) \rceil (\lceil \log_2(r) \rceil + 1)}.$$

That is, for general ordered paths $(P_r, \prec)$ we have $\mathbf{RO}((P_r, \prec); 2) \le r^{O(\log(r))}$. We are not aware of any other general bounds.

3.3 Cycles

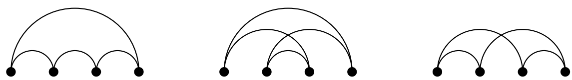
Ramsey numbers for (unordered) cycles are known for some time now. It is a folklore in Ramsey theory that $\mathbf{R}(C_3; 2) = \mathbf{R}(C_4; 2) = 6$ holds. The first partial results on Ramsey numbers of cycles were obtained by Chartrand and Chuster [3], and Bondy and Erdős [1]. These were later extended by Rosta [28], and Faudree and Schelp [14]. Nowadays we know all values of Ramsey numbers for cycles in 2-colored complete graphs:

$$\mathbf{R}(C_r, C_s) = \begin{cases} 2r - 1 & \text{if } (r,s) \neq (3,3) \text{ and } 3 \le s \le r, \\ & s \text{ is odd}, \\ r + s/2 - 1 & \text{if } (r,s) \neq (4,4) \text{ and } 4 \le s \le r, \\ & r \text{ and s are even}, \\ \max\{r + s/2 - 1, 2s - 1\} & \text{if } 4 \le s < r, s \text{ is even and} \\ & r \text{ is odd}. \end{cases}$$

The multicolor case turned out to be more demanding (see [10, 23, 24]), but the following is known.

Theorem 3.9 (T. Luczak et al. [24]). *For every $c \ge 4$ and n odd, we have $\mathbf{R}(C_n; c) \le c2^c n + o(n)$, and for every $c \ge 2$ and n even, we have $\mathbf{R}(C_n; c) \le cn + o(n)$ as $n \to \infty$.*

Similarly to the original development in unordered Ramsey theory we also first focus on small examples before we target the general case. The first obvious case is C_4. This ordered graph has three possible orderings up to isomorphism, see Figure 8. In the following proposition we show how Ramsey numbers of ordered C_4 behave.



$(C_4, \prec_A)$ $(C_4, \prec_B)$ $(C_4, \prec_C)$

Figure 8: Possible orderings of C_4.

Proposition 3.10. *We have*

1. $\mathbf{RO}((C_4, \prec_A); 2) = 14$,
2. $\mathbf{RO}((C_4, \prec_B); 2) = 10$,
3. $11 \leq \mathbf{RO}((C_4, \prec_C); 2) \leq 13$.

Proof. The lower bounds follow from the colorings presented in Figure 9, thus it remains to show the upper bounds for each ordering. For this, suppose that $(K_n, \prec)$ is an ordered complete graph with 2-colored edges (red and blue) where n is relevant to each case.

1. This result is implied by far more general statement, see Theorem 2.8 whose proof is shown bellow.

2. Suppose for contradiction that $(K_{10}, \prec)$ does not contain monochromatic $(C_4, \prec_B)$. That is, no two vertices share a monochromatic right common neighborhood of size at least two. Then our claim is that K_{10} does not contain a vertex with monochromatic right degree grater than five. If it does, then without loss of generality there is a vertex v with right red degree at least six. This red neighborhood contains a vertex w which either has a red right degree at least two or a blue right degree at least four. To avoid $(C_4, \prec_B)$ the second case has to occur. However then the same observation implies that this blue neighborhood of w contains a vertex with either red or blue left degree at least two. In any case we obtain $(C_4, \prec_B)$ and thus a contradiction.

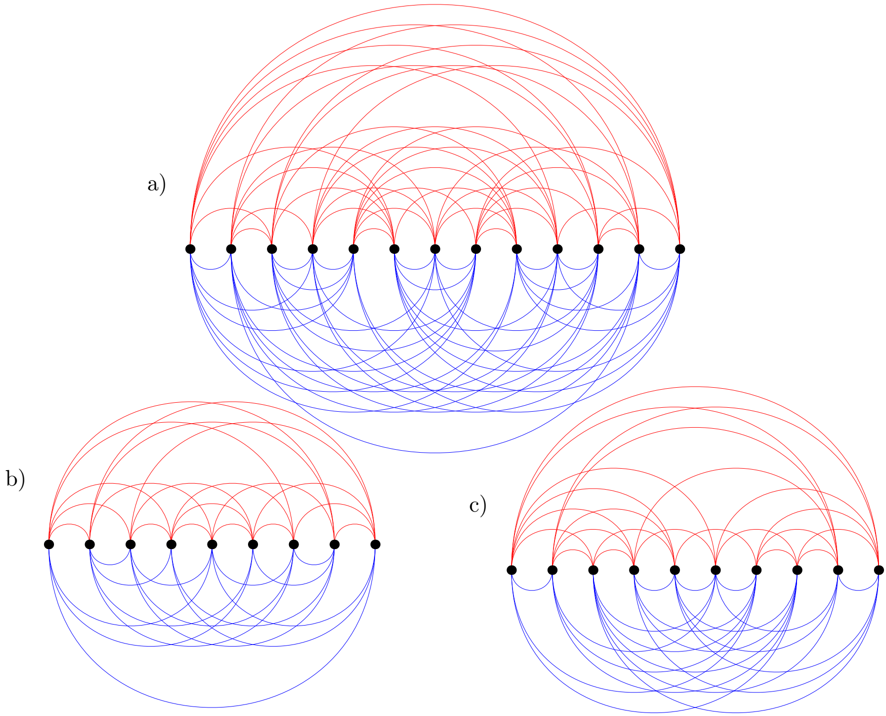


Figure 9: Colorings for the lower bounds in the proof of Proposition 3.10.

Thus there is no vertex with monochromatic right neighborhood of size at least six. From the assumptions each pair of vertices has at most one common neighbor in every color. Without loss of generality we assume that the first vertex v_1 in $\prec$ has a red right degree five and a blue right degree four.

If the second vertex v_2 is a a blue neighbor of v_1, then it has at most one (right) red neighbor between red neighbors of v_1 and the remaining four are its blue neighbors. Then however the third vertex v_3 has at least two common right monochromatic neighbors either with v_1 or v_2 between these four vertices.

If v_2 is a red neighbor of v_1, then it necessarily has three blue right neighbors between the remaining four red neighbors of v_1 and three red neighbors between four blue neighbors of v_1. Independently on the location of v_3 we get that v_3 has at least two common neighbors of the same color either with v_1 or v_2.

3. The first vertex v_1 in $(K_{13}, \prec)$ has either six blue and six red left neighbors, or at least seven monochromatic neighbors. In any case, since every two vertices can have at most a single common neighbor of the same color between them, there are (without loss of generality) at least five vertices which are red right neighbors of v_1 and blue left neighbors of v_n. We can see that those five vertices has always either a vertex with red left degree at least two or a vertex with blue right degree at least two. Both situations imply the existence of $(C_4, \prec_C)$.

$\square$

Computer experiments have shown that the lower bound in the third case is optimal, that is $\mathbf{RO}((C_4, \prec_C); 2) = 11$. Again, one can see that the ordered Ramsey numbers differ from the unordered ones.

Similarly as with paths, the most natural ordering scheme is the monotone one. In the rest of the section we establish the precise value of ordered Ramsey numbers for monotone cycles. That is, Theorem 2.8 which says that

$$\mathbf{RO}((C_r, \lhd_{mon}), (C_s, \lhd_{mon})) = 2rs - 3r - 3s + 6$$

for every $r, s \geq 2$. Note that the obtained formula is much simpler than the one in the unordered case. Before proving this statement, we first prove an auxiliary lemma.

Lemma 3.11. *For positive integers r and s and every total ordering $\prec$ we have*

$$\mathbf{RO}((P_r, \lhd_{mon}), (K_s, \prec)) = (r-1)(s-1) + 1.$$

Proof. The lower bound can be obtained from the same construction as in the proof of Proposition 2.3 (see [29]). For the upper bound we use induction on r. If $r = 2$, then this statement holds since we either have a monochromatic K_s, or a blue edge. Suppose that $r > 2$ and let $(K_{(r-1)(s-1)+1}, \prec')$ be an ordered complete graph with edges colored red and blue. Assume that it does not contain blue $(K_s, \prec)$ nor red $(P_r, \lhd_{mon})$. Using inductive hypothesis we know that there is at least

$$(r-1)(s-1) + 1 - (r-2)(s-1) = s$$

distinct vertices which are the last vertices of a red copy of $(P_{r-1}, \triangleleft_{mon})$. From the assumption every edge between such vertices is blue, otherwise we would extend one of these paths. However then we have a blue copy of K_s, a contradiction. $\square$

As a simple corollary of this lemma one can see that for every ordered graph $(G, \prec)$ on s vertices which contains a monotone Hamiltonian path we have $\mathbf{RO}((P_r, \triangleleft_{mon}), (G, \prec)) = (r-1)(s-1)+1$. This, again, supports the idea that monotone paths play an important role in the ordered Ramsey theory.

Proof of Theorem 2.8. First, we show the upper bound. In an ordered 2-colored complete graph $(K_n, \prec)$ with $n = 2rs - 3r - 3s + 6$ vertices the first vertex v_1 has either at least $(r-2)(s-1)+1$ red neighbors or at least $(r-1)(s-2)+1$ blue neighbors, according to pigeon-hole principle. In the first case there is, according to Lemma 3.11, a red copy of $(P_{r-1}, \triangleleft_{mon})$ which creates red $(C_r, \triangleleft_{mon})$ together with v_1 or a blue copy of $(C_s, \triangleleft_{mon})$. The second case with large blue neighborhood is analogous and we thus always get either red $(C_r, \triangleleft_{mon})$ or blue $(C_s, \triangleleft_{mon})$.

For the lower bound we show a coloring of $(K_n, \prec)$ where $n = 2rs - 3r - 3s + 5$, which avoids a red copy of $(C_r, \triangleleft_{mon})$ and a blue copy of $(C_s, \triangleleft_{mon})$. An example of such coloring for $r = s = 4$ can be found in Figure 9, part a). Consider a partition of the vertex set of $(K_n, \prec)$ into the following consecutive (in the ordering $\prec$) and disjoint subsets which we denote as S_i, $i = 1, 2, \ldots, 2r-3$. If r is odd, then the first and last $(r-1)/2$ subsets S_i in $\prec$ have size $s-1$ and the remaining $r-2$ subsets S_i have size $s-2$. If r is even, then the first and last $(r-2)/2$ subsets S_i contain $s-2$ vertices and the remaining $r-1$ subsets S_i have $s-1$ vertices. Note that in both cases we have n vertices in total. We assume that the vertices of S_i are of the form v^i_j where $j = 1, \ldots, |S_i|$ and $v^i_j \prec v^i_k$ whenever $j < k$. We also refer to the index j as the *index of a vertex* v^i_j.

The coloring of the edges is then defined as follows. First, we color all edges between vertices from the same set S_i blue. Next, we introduce four types of pairs (S_i, S_j), $i < j$, according to which we color edges between vertices from sets S_i and S_j. We say that (S_i, S_j), $i < j$, is of the type:

- $T_<$ if $j - i \leq r - 2$ and $|S_i| \leq |S_j|$. In this case we color the edges $\{v^i_k, v^j_l\}$ blue if $k < l$ and red otherwise.

- $T_{\geq}$ if $j - i > r - 2$ and $|S_i| < |S_j|$. Then the edges $\{v_k^i, v_l^j\}$ are colored blue if $k \geq l$ and red otherwise.
- $T_{>}$ if $j - i > r - 2$ and $|S_i| \geq |S_j|$. Then the edges $\{v_k^i, v_l^j\}$ are colored blue if $k > l$ and red otherwise.
- $T_{\leq}$ if $j - i \leq r - 2$ and $|S_i| > |S_j|$. Then the edges $\{v_k^i, v_l^j\}$ are colored blue if $k \leq l$ and red otherwise.

The main idea is that for the types $T_{<}$ and $T_{\leq}$ we color blue the edges between vertices such that their indices are increasing or non-decreasing (i.e. those vertices are relatively far from each other), while for $T_{>}$ and $T_{\geq}$ the indices are decreasing or non-increasing (such vertices are relatively close to each other). For red edges, the indices behave exactly opposite. The distribution of the types of pairs, as well as the definition of those types, is illustrated on small examples in the following two figures.

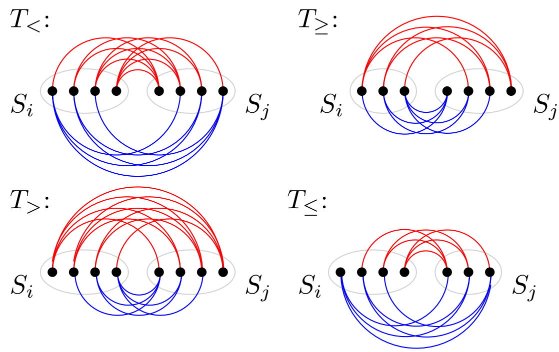


Figure 10: The types of pairs (S_i, S_j) for $s = 5$ and colorings of corresponding edges.

It remains to show that this coloring avoids forbidden cycles. We claim that our coloring does not contain a red copy of $(C_r, \lhd_{mon})$. To prove this claim, suppose for a contradiction that there is such a copy. Note that it contains at most one vertex from each set S_i, because their vertices induce blue cliques. The monotone path of length r induced by a red cycle also cannot have an edge which connects vertices from S_i and S_j where (S_i, S_j) is of type $T_{>}$ or $T_{\geq}$, because in both cases we do not use vertices from at least $r-2$ sets S_k. This leaves at most $2r - 3 - (r - 2) = r - 1$ sets each from we can use a single vertex. This is not possible, as $(C_r, \lhd_{mon})$ contains r

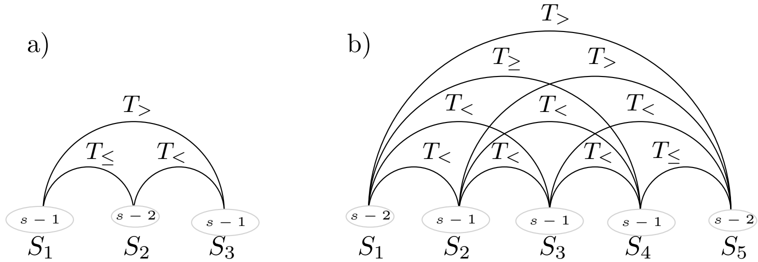


Figure 11: Distribution of types of pairs (S_i, S_j) for $r = 3$ (part a) and $r = 4$ (part b).

vertices. Hence the vertex indices on this monotone path are non-increasing, as the path uses red edges between pairs (S_i, S_j) only of types $T_<$ or $T_\leq$.

Since the number of sets S_i of size $s-1$ and the sets of size $s-2$ is less than r, we have to use vertices from both of those variants. If we have an edge between (S_i, S_j) of type $T_\leq$ in the monotone path, then the vertex indices decrease at least once (as they are connected with a red edge). However the longest edge in our red cycle is between sets of type $T_>$ or $T_\geq$ and thus it connects vertices whose indices are non-decreasing. This is a contradiction, because from our observations their indices should decrease.

The other possibility is that all edges of the red monotone path are between pairs of types $T_<$. Then the longest edge of the red cycle is of type $T_\geq$, because it has to connect S_i with S_j where $|S_i| < |S_j|$. Here we have used the specific distribution of small and large sets S_i. However then the vertex indices have to increase at least once and we already observed that this is not possible. A contradiction.

Now we prove the nonexistence of a blue copy of $(C_s, \lhd_{mon})$. Again, suppose that there is such a cycle. This time, we can use edges whose both endpoints are in the same S_i. However the blue cycle has to use vertices from at least two sets S_i, because neither of them contains s vertices.

Consider the blue monotone path of length s in our blue cycle. If it does not contain an edge between (S_i, S_j) of type $T_>$ or $T_\geq$, then the vertex indices are non-decreasing. According to the distribution of small and large sets S_i, there is at most one edge between vertices with the same vertex index. Such an edge corresponds to a jump from a larger S_i to a smaller S_j, i.e. a pair of type $T_\leq$. Therefore the length of every such blue monotone path is at most vertex index of its last vertex plus one, where the additional one is added only when we use the previously described jump. Since every vertex has index at most $s-1$, we see that the path uses exactly one pair of type $T_{\leq}$. Then the vertices of the path cannot remain in the smaller sets S_i, as their indices would be at most $s-2$. However then the edge between the first and the last vertex of the path must be of type $T_{>}$, because we need to jump from a larger set to a smaller one and then conversely (for r even this is already impossible), thus this edge is between S_i and S_j of the same size and $j-i > r-2$. Then the indices decrease at least once and this is impossible.

The last case to analyze is when the blue monotone path uses (exactly once) an edge e between S_i and S_j with $j-i > r-2$, i.e. a pair $T_{>}$ or $T_{\geq}$. Such an edge is at most one, because it jumps over at least $r-2$ sets and we have $2r-3$ in total. Thus all the other edges of this path are either between pairs (S_i, S_j) of types $T_{<}$ or $T_{\leq}$ or they connect vertices from the same set. That is, except of the edge e the indices of endpoints of all other vertices are non-decreasing. The only case when vertex indices are not increasing is when we jump using an edge from a larger set to a smaller one and this can also happen at most once, according to the distribution of small and large sets. The construction implies that the longest edge of the blue monotone cycle is between a pair of type $T_{>}$ or $T_{\geq}$, therefore the index of the last vertex is at most as large as the index of the first vertex.

Suppose that we do not use a pair of type $T_{\leq}$. Then the indices on the path increase by at least $s-2$, because we use $s-2$ pairs of type $T_{<}$ or edges within the same S_i. The only possibility for the indices to decrease is on the edge of type $T_{>}$, because the decrease must be by at least $s-2$, thus we need to jump from vertex with index $s-1$ to a vertex with index 1. We cannot do this with an edge of type $T_{\geq}$, as it jumps from a smaller set where the indices are at most $s-2$. Now, consider the longest edge in the cycle. It must be of type $T_{>}$ or $T_{\geq}$ as it jumps across at least $r-2$ sets. However neither of the possibilities can occur. The edge of type $T_{>}$ would connect vertices whose indices decrease, but the last vertex has index of size at least as large as the index of the first one, according to the size of the total decrease and increase. The longest edge of type $T_{\geq}$ would connect a vertex from a smaller set S_i with a vertex from larger S_j and this is impossible according to the distribution of the sets, because we have used an edge of type $T_{>}$.

So assume that we have used (exactly once) an edge of type $T_{\leq}$ to jump between vertices whose indices are the same. Such an edge connects a larger set with a smaller one and thus, according to their distribution, the longest edge in the cycle is of type $T_>$. This means that the index of the first vertex is strictly larger than the index of the last one. The total decrease of indices must then be also strictly larger than their increase which is at least $s-3$, as at least $s-3$ edges of the path are of type $T_<$ or are between edges from the same S_i. To finish the proof note that we cannot use an edge of type $T_>$ on the path together with the edge of type $T_\leq$. This is again because of the distribution of the sets and thus the total decrease is at most $s-3$, as edges of type $T_\geq$ jump from a smaller to a larger set. □

It could be interesting to extend this theorem to a multicolored case, even though it might be more demanding, as this question is still open for unordered cycles.

As noted by Cibulka et al. [5], the coloring we have just constructed can be used to show an exact formula for so called geometric and convex geometric Ramsey numbers for cycles which is a concept introduced by Károlyi, Pach and Tóth [21].

For a finite set of points $P \subset \mathbb{R}^2$ in general position (no three points are collinear), we denote as K_P the *complete geometric graph on* P which is a complete graph with vertex set P whose edges are straight-line segments between pairs of points of P. The graph K_P is *convex* if the points from P are in *convex position*, that is, the set of vertices of K_P is the set of vertices of a convex polygon. The *geometric Ramsey number* of G, denoted by $\mathbf{RG}(G)$, is the smallest integer n such that every complete geometric graph K_P on n vertices with edges colored by two colors contains a monochromatic non-crossing copy of G. If we consider only convex complete geometric graphs K_P in this definition, then we get so called *convex geometric Ramsey number* $\mathbf{RC}(G)$. Note that these numbers are finite only if G is outerplanar and that $\mathbf{RC}(G) \leq \mathbf{RG}(G)$ holds for every outerplanar graph G.

For cycles C_n, $n \geq 3$, Károlyi, Pach and Tóth [22] showed an upper bound $\mathbf{RG}(C_n) \leq 2n^2 - 6n + 6 = 2(n-2)(n-1) + 2$ and also observed that $\mathbf{RC}(C_n) \geq (n-1)^2 + 1$ holds. The previous theorem implies that the upper bound is actually tight.

Corollary 3.12. *For every integer* $n \geq 3$, *we have* $\mathbf{RC}(C_n) = \mathbf{RG}(C_n) = 2(n-2)(n-1) + 2$.

Proof. According to the upper bound of Károlyi et al. and the fact $\mathbf{RC}(C_n) \leq \mathbf{RG}(C_n)$, it suffices to show that $\mathbf{RC}(C_n) \geq 2(n-2)(n-1) + 2$. To do so, we use Theorem 2.8. Consider the coloring of complete ordered graph $(K_N, \prec)$, $N = 2(n-2)(n-1) + 1$ obtained in the proof of this theorem.

If $V(K_N) = \{v_1, \dots, v_N\}$ and $v_i \prec v_j$ for $i < j$, then we can map v_i to the points (i, i^2) forming a set $P \subseteq \mathbb{R}^2$ and join these points by straight line segments. Thus we obtain a convex geometric complete graph K_P on N vertices. Now, it suffices to observe that each monochromatic non-crossing copy of C_n in K_P would correspond to a monochromatic copy of $(C_n, \lhd_{mon})$ in $(K_N, \prec)$. This is because edges v_iv_j and v_kv_ℓ cross if and only if $i < k < j < \ell$. $\square$

4 Lower Bounds

The following proposition, whose proof is based on a classical probabilistic argument, gives us a general lower bound on Ramsey numbers of ordered graphs. Using this result we can derive a lower bound for dense graphs which is exponential in the number of vertices. Proposition 2.9 is a special case of this assertion as if a graph G on v vertices has $\Omega(v^{1+\varepsilon})$ edges for some $\varepsilon > 0$, then we get $\mathbf{RO}((G, \prec); 2) = \Omega(v2^{v^\varepsilon})$ for any vertex ordering $\prec$ of G. Applying this result to complete bipartite graphs $K_{k,k}$, we see that their ordered Ramsey numbers are bounded by $\Omega(k2^{k/2})$ no matter what vertex ordering we choose. See Corollary 4.2.

Let us mention that for the unordered case, except of $\mathbf{R}(K_{3,3}; 2) = 18$, basically no exact values for $\mathbf{R}(K_{r,s}; c)$ are known. However Chung and Graham [7] derived the following general bounds

$$(2\pi\sqrt{rs})^{1/(r+s)} \left(\frac{r+s}{e^2}\right) c^{(rs-1)/(r+s)} < \mathbf{R}(K_{r,s}; c) \le (s-1)(c + c^{1/r})^r,$$

where e is the base for natural logarithms.

Proposition 4.1. *Let c, r and s be positive integers and let $\prec_1, \dots, \prec_c$ be vertex orderings of a graph $G = (V, E)$ with v vertices and m edges. Then we have*

$$\mathbf{RO}((G, \prec_1), \dots, (G, \prec_c)) \ge (2\pi v)^{1/v} \left(\frac{v}{e}\right) c^{(m-1)/v}.$$

Proof. Let $(K_n, \prec)$ be a complete ordered graph. We c-color its edges independently at random with probability $1/c$ for each color. Then the probability that a set $S \subset V$ of size v induces $(G, \prec_i)$ in color i is $(1/c)^m$, since the ordering $\prec_i$ determines the set of edges.

Using union bound we derive

$$\Pr[\exists i \in \{1,\dots,c\} \text{ such that } (G,\prec_i) \subseteq (K_n,\prec) \text{ in color } i] \leq$$
$$\binom{n}{v}\cdot c \cdot \left(\frac{1}{c}\right)^m = \binom{n}{v}\left(\frac{1}{c}\right)^{m-1} \leq \frac{n^v}{v!}\left(\frac{1}{c}\right)^{m-1}.$$

Considering the Stirling's approximation formula $k! \sim \sqrt{2\pi k}\left(\frac{k}{e}\right)^k$ we can bound this probability from above by

$$\frac{n^v}{\sqrt{2\pi v}\left(\frac{v}{e}\right)^v c^{m-1}}.$$

This expression is strictly smaller than 1 for $n < \sqrt[v]{2\pi v}(v/e)c^{(m-1)/v}$. Hence we get that for such n there exists a c-coloring of edges of $(K_n,\prec)$ which avoids $(K_{r,s},\prec_i)$ for all $i = 1,\dots,c$. $\square$

Corollary 4.2. *Let c, r and s be positive integers and let $\prec_1,\dots,\prec_c$ be vertex orderings of $K_{r,s}$. Then we have*

$$\mathbf{RO}((K_{r,s},\prec_1),\dots,(K_{r,s},\prec_c)) \geq (2\pi(r+s))^{1/(r+s)}\frac{r+s}{e}c^{(rs-1)/(r+s)}.$$

Proof. It suffices to apply Proposition 4.1 on $(K_{r,s},\prec_1),\dots,(K_{r,s},\prec_c)$ where $m = rs$ and $v = r+s$. $\square$

The same approach used for unordered complete bipartite graphs produces a lower bound which would differ by $\sqrt[r+s]{\binom{s+t}{s}}$. For $n \to \infty$ this can be treated as a constant $C \leq 2$, because the binomial coefficient is largest when $s = t$ and $\binom{2k}{k} \sim 4^k/\sqrt{\pi k}$.

Seeing how lower bounds for dense ordered and unordered graphs start growing exponentially, one might ask: what is the maximum difference between ordered and unordered Ramsey numbers of a given graph? That is, what is the maximum asymptotic ratio between $\mathbf{R}(G;2)$ and $\mathbf{RO}((G,\lhd);2)$ taken over all ordered graphs $(G,\lhd)$? For sparse graphs we can consider the ratio $\frac{\mathbf{RO}((G,\lhd);2)}{\mathbf{R}(G;2)}$, while for dense graphs the ratio $\frac{\log \mathbf{RO}((G,\lhd);2)}{\log \mathbf{R}(G;2)}$ might be more convenient, as we know, according to the previous proposition, that their ordered Ramsey numbers are exponential. So far we have seen examples (monotone paths and cycles) where the order of this first ratio was $O(n)$. Using the fact that graphs with bounded maximum degree have linear Ramsey numbers with respect to their size, the following theorem implies that we can do better.

Theorem 4.3. *There is a graph G with maximum degree three and ordering $\lhd$ of its vertices such that* $\mathbf{RO}((G,\lhd);2) = \Omega(n^{\log_2 5})$.

Proof. First, we show a recursive construction of such ordered graph, which we denote as $(G_k, \lhd_k)$, $k \in \mathbb{N}$. Let $(G_1, \lhd_1)$ consist of a single edge. Then for $k \geq 1$ the graph $(G_{k+1}, \lhd_{k+1})$ is constructed as follows. Let $n_k = |V(G_k)|$ and consider vertices $v_0 \lhd_{k+1} \ldots \lhd_{k+1} v_{2n_k+1}$. On the vertices from $\{v_{in_k+1}, \ldots, v_{(i+1)n_k}\}$, $i = 0, 1$, we build copies of $(G_k, \lhd_k)$ and then we place new edges $\{v_{n_k}, v_{n_k+1}\}$, $\{v_{2n_k}, v_{2n_k+1}\}$, $\{v_0, v_1\}$ and $\{v_0, v_{2n_k+1}\}$. That is, $(G_{k+1}, \lhd_{k+1})$ consists of two consecutive copies of $(G_k, \lhd_k)$ and two new extremal vertices placed on Hamiltonian monotone cycle. The first steps of the construction are depicted in Figure 12, part a). It is easy to check that $n_k = 2^{k+1} - 2$ and that no vertex has degree more than three.

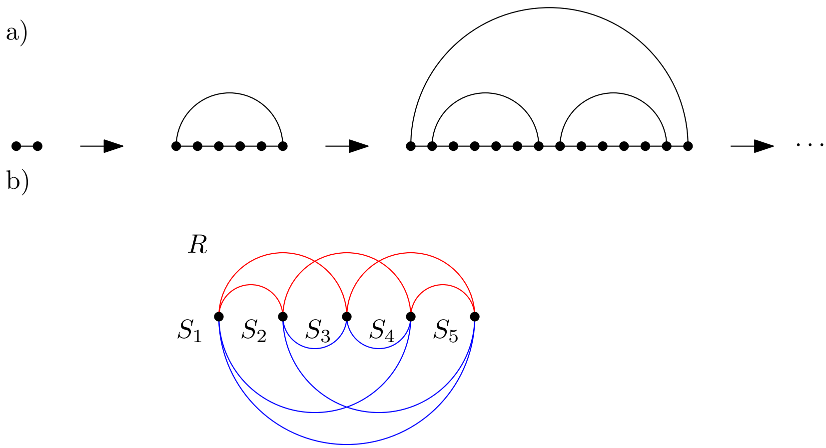


Figure 12: Construction of the graphs G_k (a) and colorings c_k (b).

Now, we show a construction of a coloring c_k of a sufficiently large complete graph which avoids monochromatic copy of $(G_k, \lhd_k)$. We do it, again, recursively with respect to k. Let N_k denote the number of vertices of a complete graph whose edges are being colored according to c_k. The coloring c_1 is trivial, as we set $N_1 = 1$. For $k \geq 1$, let $N_{k+1} = 5N_k = 5^k$ and let the given ordered set of N_{k+1} vertices be separated into five disjoint consecutive subsets S_i of size N_k. We color the edges induced by the vertices from each S_i by c_k. It remains to color the edges between S_i and S_j, $i \neq j$. To do so, we use the coloring from Figure 12, part b), which avoids monochromatic monotone cycles. Each set S_i corresponds to the i-th vertex of the graph R from this figure and all edges between S_i and S_j get the same color as the edge between the i-th and j-th vertex in R.

It remains to show that there is no monochromatic copy of $(G_k, \lhd_k)$ in $(K_{N_k}, \prec)$ colored with c_k. This is done by induction on k. The coloring c_1 avoids $(G_1, \lhd_1)$ as K_{N_1} has only one vertex. Let $k \geq 2$ and suppose for a contradiction that there is a monochromatic copy of $(G_k, \lhd_k)$ in $(K_{N_k}, \prec)$. Then this copy cannot be contained in at most two sets S_i, as otherwise there would be a monochromatic copy of $(G_{k-1}, \lhd_{k-1})$ in one of those sets. However this is impossible, as the edges induced by S_i are colored by c_{k-1} which, using inductive hypothesis, avoids such copy. Thus $(G_k, \lhd_k)$ occupies at least three sets S_i. However this is also impossible, as in this case there are edges between such sets S_i which would form a monochromatic monotone cycle of length at least three in R. In any case we have obtained a contradiction, so we have $\mathbf{RO}((G_k, \lhd_k); 2) \geq N_k$ and expressing k as $k = \log_2 (n_k + 2) - 1$ we get the result. □

The ratio between numbers $\mathbf{R}(G; 2)$ and $\mathbf{RO}((G, \lhd); 2)$ achieved by this theorem is
$\Omega(n^{-1+\log_2 5}) \sim \Omega(n^{1.32})$. However the maximum difference one can obtain is still unknown.

5 Conclusions

We have introduced Ramsey theory for ordered graphs and showed estimates and exact formulas for Ramsey numbers of various classes of graphs, including stars, complete bipartite graphs, cycles and paths. We have also discussed how the vertex orderings can affect Ramsey numbers showing that different orderings really matter.

There is a plenty of new questions which arise in the ordered Ramsey theory. We still do not know exact formulas for wide spectrum of graphs. It could be possible to show such formulas for all ordered stars, since their vertex orderings are not difficult to describe. Showing exact forms of ordered Ramsey numbers for ordered complete bipartite graphs such as $(K_{k,k}, \lhd_{sep})$ or $(K_{k,k}, \lhd_{mix})$ might be more challenging, but some non-trivial upper bounds or explicit constructions for lower bounds could be also interesting.

As we already mentioned, computing the exact formula for ordered Ramsey numbers of monotone cycles with at least three colors involved would be of its own interest too. Although the monotone orderings of cycles seem to be the most natural ones, there are also other possible orderings one can consider. Note that every ordering of C_4 shown in Figure 8 can be naturally extended into the following ordering schemes. We say that a cycles C_r with edges $\{v_1, v_2\}, \ldots, \{v_{r-1}, v_r\}$ and $\{v_1, v_r\}$ is an *alternating cycle* if the vertices are ordered the same as in alternating paths (see Section 3.2). Similarly, this cycle is called *mixed* if $v_1 \lhd_{mix} v_r \lhd_{mix} v_2 \lhd_{mix} v_{r-1} \lhd_{mix} \ldots v_{\lceil r/2+1 \rceil} \lhd_{mix} v_{\lceil r/2 \rceil}$. See Figure 13. Having these schemes we can ask, as in the case for monotone cycles, for the exact forms of Ramsey numbers of these ordered cycles. It could be especially interesting for even alternating cycles, as they do not contain monotone path of length three as an ordered subgraph. Similar situation holds for paths. Except for the well-known monotone case, what are the Ramsey numbers of other ordered paths?

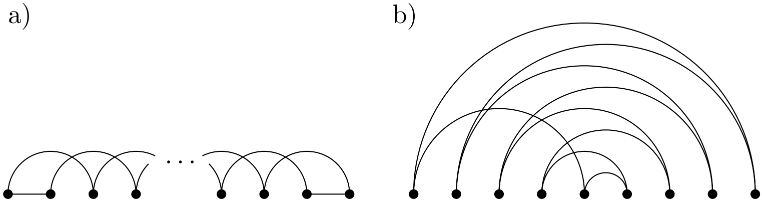


Figure 13: An example of mixed and alternating cycles.

Computing exact formulas for Ramsey numbers are not the only problems ordered Ramsey theory can offer. We can also ask questions which concern the structure of optimal colorings. Let us define the *maximum ordered Ramsey number* $\mathbf{RO}_k^{max}(\mathcal{H}; c)$ for positive integers c and k and for a k-uniform hypergraph $\mathcal{H}$ as the maximum of $\mathbf{RO}_k((\mathcal{H}, \prec); c)$ taken over all possible vertex orderings of $\mathcal{H}$. We can define the *minimum ordered Ramsey number* $\mathbf{RO}_k^{min}(\mathcal{H}; c)$ of a hypergraph $\mathcal{H}$ analogously. Afterwards we can not only ask what are such Ramsey numbers for different hypergraphs, but we can also be interested in what properties of vertex orderings make ordered Ramsey numbers grow faster. We have observed that the length of monotone paths contained in the ordered graphs might be important. For paths this observation indicates that $\lhd_{mon}$ could maximize ordered Ramsey numbers for ordered paths.

Question 5.1. *Is it true that for every positive integer r we have*

$$\mathbf{RO}^{max}(P_r; 2) = \mathbf{RO}((P_r, \lhd_{mon}); 2)?$$

Also, is it true that for every positive integer r we have

$$\mathbf{RO}^{min}(P_r;2) = \mathbf{RO}((P_r, \lhd_{alt});2)?$$

More fundamental question is whether there is a formula for the numbers $\mathbf{RO}^{max}(G_r;2)$ or $\mathbf{RO}^{min}(G_r;2)$ where G_r are graphs on r vertices from some specified graph class (such as paths, cycles, cliques, etc.) and if so, what is the asymptotic difference between those numbers?

Another very natural question is: what are the ordered Ramsey numbers of graphs with bounded degrees? We have seen that they can grow polynomially with respect to the number of vertices. However we do not have any reasonable upper bound. So we do not even know whether there are graphs with bounded degrees such that their ordered Ramsey numbers grow exponentially or if those numbers are always bounded by some polynomial.

Question 5.2. *Let G be a graph with maximum degree Δ and let $\lhd$ be some ordering scheme of its vertices. How fast (with respect to the number of vertices) can $\mathbf{RO}((G, \lhd);2)$ grow?*

We might also ask whether for every sparse graph G there is an ordering of its vertices such that the corresponding ordered Ramsey number behaves similarly as the unordered one.

Question 5.3. *Is it true that for every graph G on n vertices with degrees bounded by a constant Δ there exists an ordering $\prec$ such that*

$$\mathbf{RO}((G, \prec);2) = Cn$$

where $C = C(\Delta)$ is a constant which depends on Δ?

This result, if true, would be a natural strengthening of the fact that such unordered graphs have linear unordered Ramsey numbers. The last question we would like to mention is the following:

Question 5.4. *What is the maximum difference one can get comparing $\mathbf{RO}((G, \lhd);2)$ and $\mathbf{RO}((G, \lhd');2)$ with respect to the size of G?*

So far we have a gap which grows super-quadratically, while the upper bound is exponential.

Acknowledgments

The authors would like to thank to Josef Cibulka, Jan Kynčl and Jiří Matoušek for many helpful comments.

References

[1] J. A. Bondy and P. Erdős, Ramsey numbers for cycles in graphs, *J. Combin. Theory* **11** (1973), 46–54.

[2] S. A. Burr and J. A. Roberts, On Ramsey numbers for stars, *Utilitas Math.* **4** (1973), 217–220.

[3] G. Chartrand and S. Schuster, On the existence of specified cycles in complementary graphs, *Bull. Am. Math. Soc.* **77** (1971), 995–998.

[4] J. Cibulka, Extremal combinatorics of matrices, sequences and sets of permutations, Ph.D. Thesis, Charles University, Prague, 2013.

[5] J. Cibulka, P. Gao, M. Krčál, T. Valla, and P. Valtr, On the geometric Ramsey number of outerplanar graphs, arXiv:1310.7004, 2013.

[6] S. A. Choudum and B. Ponnusamy, Ordered Ramsey numbers, *Discrete Mathematics* **247** (2002), 79–92.

[7] F. R. K. Chung and R. L. Graham, On multicolor Ramsey numbers for complete bipartite graphs, *J. Combin. Theory Ser. B* **18** (1975), 164–169.

[8] D. Conlon, A new upper bound for diagonal Ramsey numbers, *Ann. of Math.* **170** (2009), 941–960.

[9] D. Conlon, J. Fox, and B. Sudakov, Ramsey numbers of sparse hypergraphs, *Random Structures and Algorithms* **35** (2009), 1–14.

[10] T. Dzido, A. Nowik, and P. Szuca, New lower bound for multicolor Ramsey numbers for even cycles, *European Journal of Combinatorics* **12** (2005).

[11] M. Eliáš and J. Matoušek, Higher-order Erdős-Szekeres theorems, *SoCG '12 Proceedings of the twenty-eighth annual symposium on Computational geometry* (2012), 81–90.

[12] P. Erdős, Some remarks on the theory of graphs, *Bull. Amer. Math. Soc.* **53** (1947), 292–294.

[13] P. Erdős and G. Szekeres, A combinatorial problem in geometry, *Compos. Math.* **2** (1935), 463–470.

[14] R. J. Faudree and R. H. Schelp, All Ramsey numbers for cycles in graphs, *Discrete Mathematics* **8**(4) (1974), 313–329.

[15] R. J. Faudree and R. H. Schelp, Path Ramsey numbers in multicolorings, *J. Combin. Theory Ser. B* **19**(2) (1975).

[16] J. Fox, J. Pach, B. Sudakov, and A. Suk, Erdős-Szekeres-type theorems for monotone paths and convex bodies, *Proc. London Math. Soc.* **105** (2012), 953–982.

[17] Z. Füredi and P. Hajnal, Davenport-Schinzel theory of matrices, *Discrete Mathematics* **103**(3) (1992), 233–251.

[18] L. Gerensér and L. Gyárfás, On Ramsey-type problems, *Ann. Univ. Sci. Eötvös* **10** (1967), 167–170.

[19] J. E. Graver and J. Yackel, Some graph theoretic results associated with Ramsey's theorem, *Journal of Combinatorial Theory* **4**(2) (1968), 125–175.

[20] A. Gyárfás, M. Ruszinkó, G. N. Sárközy, and E. Szemerédi, Three-color Ramsey numbers for paths, *Combinatorica* **27**(1) (1968), 35–69.

[21] G. Károlyi, J. Pach, and G. Tóth, Ramsey-type results for geometric graphs. I, *Discrete & Computational Geometry* **18**(3) (1997), 247–255.

[22] G. Károlyi, J. Pach, G. Tóth, and P. Valtr , Ramsey-type results for geometric graphs. II, *Discrete & Computational Geometry* **20**(3) (1998), 375–388.

[23] Yusheng Li and Ko-Wei Lih, Multi-color Ramsey numbers of even cycles, *European Journal of Combinatorics* **30**(1) (2009), 114–118.

[24] T. Luczak, M. Simonovits, and J. Skokan, On the multi-colored Ramsey numbers of cycles, *Journal of Graph Theory* **69** (2012), 169–175.

[25] K. G. Milans, D. Stolee, and D. B. West, Ordered Ramsey theory and track representations of graphs, to appear, 2012.

[26] G. Moshkovitz and A. Shapira, Ramsey theory, integer partitions and a new proof of the Erdős-Szekeres Theorem, arXiv:1206.4001v1, 2012.

[27] J. Pach and G. Tardos, Forbidden paths and cycles in ordered graphs and matrices, *Israel Journal of Mathematics* **155**(1) (2006), 359-380.

[28] V. Rosta, On a Ramsey type problem of J.A. Bondy and P. Erdős, *J, Combin.Theory, Ser B* **15** (1973), 94–120.

[29] J. M. Steele, Variations on the monotone subsequence theme of Erdős and Szekeres, in: *Discrete Probability and Algorithms (Springer-Verlag, 1995)*, *The IMA Volumes in Mathematics and Its Applications* **72** (1995), 111–131.

[30] G. Szekeres and L. Peters, Computer solution to the 17-point Erős-Szekeres problem, *The Anziam Journal* **48** (2006), 151–164.

[31] G. Tóth and P. Valtr, The Erdős-Szekeres theorem: upper bounds and related results, in: *Combinatorial and Computational Geometry*, *MSRI Publications* **52** (2005), 557–568.

Investigating Neighborhood Diversity

Martin Koutecký

1 Introduction

The topic I investigated during my time in the USA was one I already started working on in my master thesis. It lies in the areas of parameterized complexity and structural graph theory. Generally speaking, I tried to further investigate a graph structural parameter called *Neighborhood Diversity.* For a more detailed introduction and formal definitions see chapters 1 and 2 of my thesis.

This parameter was introduced in 2010 by Michael Lampis as an answer to the open question of if there is a parameter for which the MSO_1 model checking problem is fixed-parameter tractable with complexity $O(f(k)n)$ *with f an elementary function.* On neighborhood diversity the function f is just a double exponential and thus Lampis's result answers the question positively.

What turned out to be a somewhat more interesting property of graphs with bounded neighborhood diversity is that many problems which are hard even with respect to treewidth become FPT w.r.t. neighborhood diversity. Moreover *all* the known results use an old ('83) result by Lenstra which says that solving the Integer Linear Programming is FPT w.r.t. the dimension, i.e., if we can come up with an ILP, which solves a given problem, and the ILP uses only $f(nd(G))$ variables for some function f, we've shown the problem to be FPT w.r.t. neighborhood diversity.

This technique works for a fairly large number of problems. This was shown in the original paper of Lampis, in a later paper by Ganian and even later by me and other authors. Let me summarize that in a list:

- **Lampis:** Coloring, Hamiltonian Cycle
- **Ganian:** p-Vertex-disjoint paths, Graph Motif, Precoloring Extension
- **Koutecký:** Capacitated Dominating Set, Equitable Coloring, Complete Coloring
- **Fiala, Gavenčiak, Knop, Kratochvíl:** Generalized Channel Assignment, Distance Constrained Labeling

Let me stress how remarkable this is. In my experience, understanding an algorithm for a specific problem parameterized by, for example, treewidth, takes a fair amount of time. With *all but one* (Graph Motif) of the above mentioned problems the algorithm is a fairly straightforward application of the ILP technique.

At the beginning of the REU the above raised two major questions I investigated:

1. Where does this technique *fail*? Is there anything *hard* about neighborhood diversity?

2. Is there a way to generalize these results to some wider parameter? What is a good way to generalize neighborhood diversity as a parameter?

I've spent most of my time trying to generalize the algorithm for Coloring on graphs with bounded neighborhood diversity to graphs with bounded shrub-depth. At the very end of the REU I learned about a new paper (yet to be published, submitted to IPEC) by Gajarský, Lampis and Ordyniak which actually proves that this is W[1]-hard. Even though this is unfortunate, I think I've gathered some very interesting questions and ideas and I will be investigating them together with professors and other graduate students in the upcoming year.

2 Research directions

2.1 Hardness and Neighborhood Diversity

One of the ways in which neighborhood diversity as a parameter is curious is that graphs with this parameter bounded can be described in a very concise manner. We can look at a graph of neighborhood diversity k as a graph constructed according to some template graph by replacing the vertices of this template with either cliques or independent sets of some given sizes. Thus, it is enough to have the information about the template graph (k^2 bits for the edges, k bits to say which vertices are cliques and which are independent sets) and the information of how big are the nodes ($k \cdot \log n$ bits), in total $O(k^3 \log n)$ bits. This is of course assuming there's no additional information on the graph, such as labels, weights etc.

In an email Michael Lampis send me a few months ago he conjectured that all graph problems where only the graph is the input will be FPT w.r.t. neighborhood diversity. This turns out to be trivially false because of a long-known result of Courcelle, Makowsky and Rotics from 2000 showing that MSO_2 model checking on cliques cannot be even in XP (much less FPT). But this result seems kind of weird, because intuitively, *what could possibly be hard about cliques?*

So I've tried to understand precisely *why* this is happening. Unfortunately the mentioned result relies on old papers about logic spectra by Fagin from 1974 which are *very* hard to read and use a very different vocabulary than today's Finite Model Theory textbooks. Fortunately Lampis reproves this result in his recent ('13) paper using a different, constructive technique.

The reason is, *very* roughly, this. First Lampis reproves the well-known lower-bounds for MSO_1 model checking by Frick and Grohe. These lower-bounds say that checking MSO_1-expressible properties cannot be done in time $f(|\varphi|)n^c$ even on (uncolored) paths, or equivalently on unary strings. This is done by proving that MSO_1 can be used to express the rules for a computation of a Turing machine on a unary string in a very concise manner. For the MSO_2 result we first notice that MSO_2 can be used to select a path within the clique, and second that some formulae in the previous construction are even simpler to express on cliques, giving us the stronger result: MSO_2 model checking on cliques is not possible in time $n^{f(|\varphi|)}$. All of these results use the complexity assumption $EXP \neq NEXP$.

So there *is* a problem whose only input is the graph which is hard w.r.t. neighborhood diversity. But seeing how this is proven makes me wonder: does this really tell us anything about neighborhood diversity specifically? The problem is very artificial, and also it is already hard on cliques And also, who uses MSO_2 to simulate the computation of a Turing machine? This leads to the following questions:

1. **What is a natural problem that is at least W[1]-hard w.r.t. neighborhood diversity?**

2. **What is a good logic other than MSO whose model checking complexity we could investigate?**

As for the first question, my guess is that a natural hard problem could be something like the Edge Odd Circuit Traversal problem. This problem is to find the smallest set of edges in a graph whose removal will make it bipartite. I've tried to prove this problem is FPT w.r.t. neighborhood diversity in my master thesis with no success, and the problems I ran into look very numeric. (We're basically asking what is the best way to divide the vertices within the nodes?, and the number of parts to which we divide doesn't seem to be bounded by the parameter, etc.) What is left to do now is find a W[1]-hard problem similar to the 3-PARTITION problem (or other strongly NP-complete numeric problem) and see if we can reduce it to the EDGE OCT problem.

As for the second question, some work has already been done on that. In a recent paper Pilipczuk[1] shows that all problems expressible in a certain logic (*ECML Existential Counting Modal Logic*) are FPT w.r.t. treewidth *in single exponential time*. Lampis mentions it would be interesting to show what is the complexity of model checking of this (or similar) logic w.r.t neighborhood diversity and related parameters.

Also, there is some interest in generalizing the ILP technique (IPEC paper) how can we describe the set of problems for which this technique works? Can we prove some kind of a meta-theorem for it?

2.2 Generalizing Neighborhood Diversity

Another path to explore are possible generalizations of neighborhood diversity.

2.2.1 Modular-width

In the aforementioned paper by Gajarský, Lampis and Ordyniak a new generalization of neighborhood diversity called *modular-width* is introduced. Roughly speaking, a graph has modular-width bounded by k if it we can find for it a modular decomposition of arbitrary depth and arity bounded by k. Using the idea of templates mentioned above, we can say the graph has to lie in the closure of a following operation: take a graph on at most k vertices and replace every vertex with an arbitrary graph with modular-width at most k; let be neighbors precisely those graphs corresponding to neighboring vertices in the template.

The authors then prove that COLORING and HAMILTONICITY are FPT w.r.t modular-width. Moreover, they are using very similar techniques involving the ILP result of Lenstra that I described were so useful when dealing with neighborhood diversity. That makes it natural to ask the question which other problems we know are FPT w.r.t neighborhood diversity could be proven to be proven to be FPT w.r.t modular-width by generalizing the

[1]Problems parameterized by treewidth tractable in single exponential time: a logical approach

existing solutions? In particular what happens to the family of Distance Constrained Labeling problems? Or the Capacitated Dominating Set and Capacitated Vertex Cover problems?

2.2.2 Information compression parameterization

Above we mentioned that graphs with small neighborhood diversity can be described using significantly less information than general graphs. I was not able to deduce much from this besides that any problem restricted to graphs with bounded neighborhood diversity (with no additional information) is a sparse language. This means (by Mahoney's theorem) that if $P \neq NP$ then it is not NP-complete and that it falls somewhere in the hierarchy that forms between P and NP.

Still, it might be interesting to investigate how other parameters compare to neighborhood diversity in this regard. Are graphs with bounded treewidth significantly simpler in their description than general graphs? What about tree-depth, shrub-depth, modular-width or others?

2.2.3 Intersecting parameterization and approximation

A talk at the REU by Spiros Papadimitriou inspired a different idea. What if we worked with graphs that were *almost* like graphs with bounded neighborhood diversity? Such that we could get them from a graph of bounded neighborhood diversity by *few* edge deletions or insertions?

Of course, like this the idea is silly: if the original graph had neighborhood diversity k and we made l edge operations on it, then it would just have neighborhood diversity $O(k + l)$. In this way it wouldn't make sense to consider this double-parameterization. But what if we could prove some FPT results for graphs with k constant and $l \leq log(n)$ or something similar?

Or, looking at it from the other side, what if we knew that a given graph is within 5% of a graph with bounded neighborhood diversity, i.e., that we would need to flip (insert or delete) at most 5% of all edges to get a graph with neighborhood diversity at most k? Could we then prove that an algorithm exists which runs in polynomial time when k is constant and gives an approximate result whose error depends on the 5% difference? This could be useful for huge graphs if we have a graph with billions of vertices that would have neighborhood diversity in the tens or hundreds if we added or removed say thousands of edges, maybe the optimum values for many discussed problems couldn't change much. In other words, what matters is the rough structured captured by the neighborhood diversity template, not some small irregularities. (Mind that this is just intuition. Nothing is proven yet.)

2.3 Specific questions

1. Hardness results

    1. **Can we show that EOCT is W[1]-hard w.r.t. neighborhood diversity?**

2. Other logics

    1. **Can we show that ECML model checking is FPT w.r.t neighborhood diversity?**
    2. **How to characterize the effectiveness of ILP on neighborhood diversity and related parameters as an algorithmic meta-theorem?**

3. Other parameters

    1. **Can we generalize the algorithms for distance constrained labeling problems to modular-width?**
    2. **What is the information complexity of graphs of bounded tree-width? What about tree-depth, shrub-depth or modular-width?**
    3. **Can we say that if a graph differs from a neighborhood diversity bounded graph by at most 5% of edges, the optimum values of some relevant problems will not differ much?**

3 Conclusion

I am very glad for the time I spend at the REU. I've got to meet interesting people and discuss many problems from various areas. The problems I decided to work on are a long-term interest of mine and I hoped to get easy results soon because I was successful in this area before. Unfortunately that did not happen and I spent most of my time trying to prove a theorem that turns out to be false (which proved a different set of authors).

Still, I believe my time was not completely wasted, because along the way I noticed several more or less interesting research directions, which are discussed and listed in this document. Future will tell where they lead.

Routing

Jitka Novotná

1 Introduction

1.1 Motivation

Suppose that we are given a graph with one vertex marked as goal. We want to decide whether it is possible to prescribe possible moves from a vertex to its neighbors such that starting at arbitrary initial vertex we can find a tour to the goal even if some edges were removed.

1.2 Definitions

Let G be a graph with one vertex marked as *goal*. This is the vertex we want to reach from any starting position in G.

Whenever we use terms like connectivity, cut and so on we always mean edge connectivity, edge cut, and so on.

A *switching table* for a vertex v is a table with $\deg(v)$ rows a k columns. The neighbors of v correspond to the rows of this table. In the first column and the i-th row it is prescribed which neighboring vertex of v we should switch when we arrived to v from the i-th adjacent vertex. In the j-th column, $j \geq 2$, it is prescribed where to switch when the first $j-1$ prescribed edges were removed.

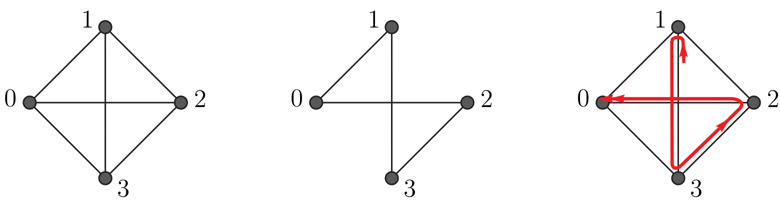


Figure 1: K_4; a ragged K_4'; a prescribed tour from vertex 1.

For edges $e_1, \ldots, e_k$ of a graph G a *ragged graph* $G' = G'(e_1, \ldots, e_k)$ is $G \setminus \{e_1, e_2 \ldots e_k\}$.

vertex 1				vertex 2				vertex 3			
neighbor	*switching table*			neighbor	*switching table*			neighbor	*switching table*		
0	0	2	3	0	0	3	1	0	0	2	1
2	0	2	3	1	0	3	1	1	0	2	1
3	2	3	0	3	1	0	3	2	0	2	1

Table 2: The switching tables for K_4. (0 is the goal)

A *prescribed tour* in G' starting at an edge (u, v) is a tour which goes from every vertex w to an edge which is prescribed in the switching table of w.

We say that a graph G has a *k-routing* if there are switching tables with k columns such that for every ragged graph obtained from G by deleting k edges there exists a prescribed tour which for any vertex says how to continue to reach the goal.

A graph G has a *routing* if it has a $(\mathrm{mincut}(G) - 1)$-routing, i.e. it has a k-routing for the largest possible k.

2 2-connected graphs

We show how to find switching tables with two columns for 2-connected graphs such that those tables work even if some edge was removed. We start by the well known lemma.

Lemma 2.1. *A graph is 2-(edge)-connected if and only if it can by created from cycle by adding paths between two vertices. Those vertices are not necessary distinct.*

Proof. A graph created in such way is 2-connected because we cannot create a bridge.

We can, show how to create a given 2-connected graph G using this construction. Let G_0 by *arbitrary* cycle of G. A graph G_i is created from G_{i-1} by adding a path from G. If we cannot add any path, then $G_i = G$. Now suppose for a contradiction that $G_i \neq G$. Then we focus on an edge $(u, v), u \in V(G_i), v \in V(G \setminus G_i)$. By definition there are two distinct paths between u and v. In a path which does not contain an edge (u, v), we can take the part starting at the vertex v and ending in the first vertex of G_i.

This part together with the edge (u, v) is a path which we can add to G_i. A contradiction. □

Theorem 2.2. *Every 2-connected graph G has a 1-routing.*

Proof. We prove that G has a 2-routing using two $i$ndependent spanning trees a, b. By that we mean two oriented spanning trees which are routed in the goal and do not use any edge of G in same direction.

We create switching tables for a vertex v int the following way:

- If we arrive to v from an edge which is not in a nor b, then we recommend an edge of the spanning tree a which leads from v and as a second option we recommend an edge of b.
- If we arrive to v from an arbitrary edge of a, then we recommend an edge of a and as a second option we recommend an edge of b which leads from v.
- If we arrive to v from an arbitrary edge of b, then we recommend an edge of b and as a second option we recommend an edge of a which leads from v.

Using these tables, we can prescribe a tour which reaches the goal in every ragged graph. We start by using one spanning tree and if we meet a removed edge, then we change to the second spanning tree and subsequently we reach the goal.

We prove that there are such spanning trees by induction on the number of added paths.

First, we choose a cycle which contains the goal. If G is just a cycle, then its spanning trees are two maximal paths, as you can see in Figure 2.

Suppose that G was created from a graph H by adding a path P. Then the goal can not be contained in the path P, because the goal lies in the cycle. The graph H contains, using the induction step, two independent spanning trees. The path P also contains two spanning trees similarly to the case for a cycle. We can combine these spanning trees into spanning trees of G as depicted in Figure 2. □

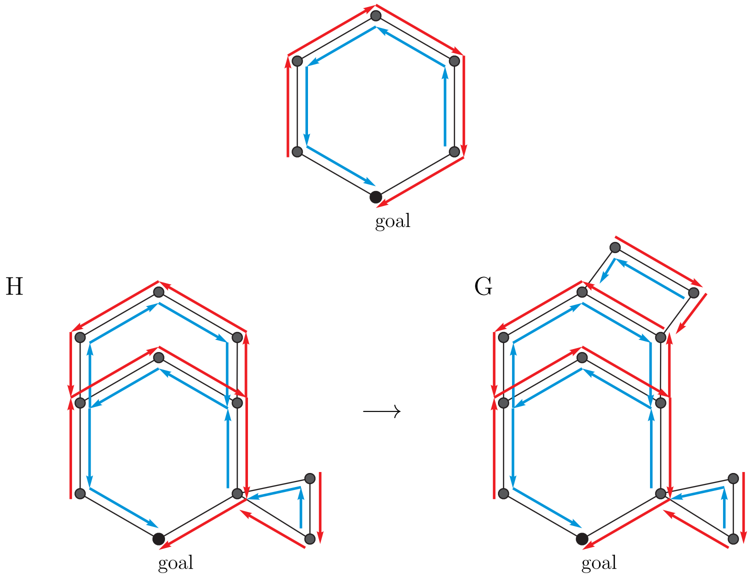


Figure 2: Spanning trees in circle; induction step

3 Three independent spanning trees generate a 3-routing

There is a way how to find switching tables using three *independent spanning trees*. Those are three oriented spanning trees, which are rooted in the goal, and do not use same edge of G in same direction. We assign red, green and blue color to those trees. You can see an example of independent spanning trees in Figure 9.

Theorem 3.1. *If a graph has three independent spanning trees, then it has a 3-routing.*

Proof. Let G be a graph with three independent spanning trees. We show that a prescribed tour reaches the goal even if two arbitrary edge were removed.

We create switching table for a vertex v in the following way:

- If we arrive to v from an edge which is not contained in any of the

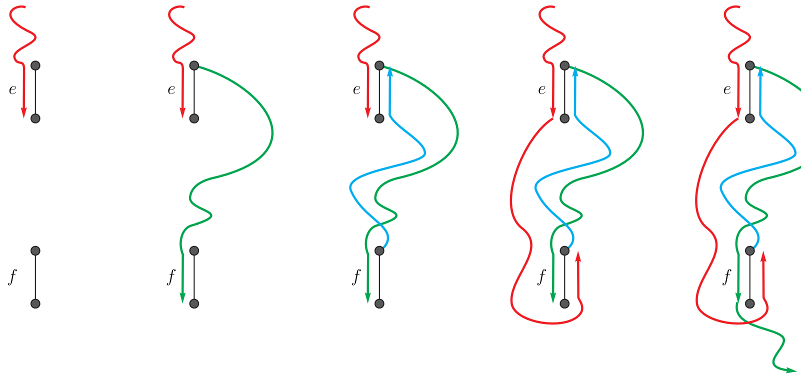


Figure 3: A prescribed tour in a graph with three independent spanning trees.

independent spanning trees, then we recommend an edge of the red spanning tree.

- If we arrive to v from an edge which is contained in some of the independent spanning trees, then we continue to an edge of the same independent spanning tree.
- If a recommended edge of the red tree was removed, then we recommend a green edge.
- If a recommended edge of the green tree was removed, then we recommend a blue edge.
- If a recommended edge of the blue tree was removed, then we recommend a red edge.

We show that every prescribed tour ends in the goal. After the fist step of the tour we are either in the goal or in an edge of some independent spanning tree, w.l.o.g., in a red edge. If some red edge e on the red path was removed, then the tour continues on a green path. The green path cannot use the edge e, because one of its orientation is already occupied by the red tree and the other orientation could create a green cycle. Either the green path reaches the goal or some green edge f was removed. In the latter case we switch to the blue path. The blue path cannot use an edge f for the same reason as the green path could not use the edge e. Blue path either reaches the goal or it uses an edge e in the other orientation then the red path did. In this case we continue on the red path as if the edge e was not removed. Red path the either reaches the goal or an edge f. In the second case we continue on the green path and this time we have to reach the goal. □

4 Reductions

We can prove that every 3-connected graph has three independent spanning trees using so called *reductions*.

A reduction use the following idea: "If a graph G has some attribute, then we can use it to construct a graph H or graphs H_1, H_2 which have three independent spanning trees using induction. There is only a few possibilities how these trees use some specified part of the graph. For each of these possibilities we construct independent spanning trees for G."

In the following chapter we prove that for every graph with more than four vertices we can use at least one reduction.

4.1 Useless edge reduction

Reduction 4.1 (Useless edge reduction)**.** *If a graph G contains an edge e such that $H = G \setminus e$ has three independent spanning trees, then G has three independent spanning trees.*

Proof. Independent spanning trees for G are exactly the same as the ones for H. □

4.2 Nontrivial 3-cut reduction

A 3-cut is *nontrivial* if it separates more than one vertex.

Reduction 4.2 (3-cut reduction)**.** *If a 3-connected graph G has a nontrivial 3-cut and all smaller 3-connected graphs have three independent spanning trees, then G also has three independent spanning trees.*

Proof. A 3-cut divides graph into two parts. Part C_1 is the part which includes the goal and part C_2 is the second one. A 3-cut is composed by edges $(e_1, e_2), (f_1, f_2), (g_1, g_2)$, as you can see in Figure 4.

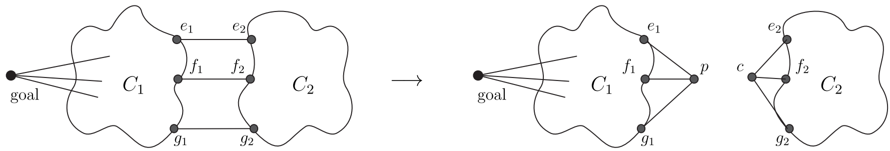


Figure 4: 3-cut reduction

The graph H_1 is created from G by a contraction of part C_2 into the vertex p. The graph H_2 is created from G by a contraction of part C_1 into the vertex v and we mark this vertex as the goal for the graph H_1.

The graphs H_1, H_2 are 3-connected, because they were created from a 3-connected graph by a contraction. Suppose for a contradiction that there is a 2-cut in H_i. Then the same edges create a 2-cut in G. The graphs H_1, H_2 have less edges then G because 3-cut was nontrivial. Thus H_1 and H_2 have three independent spanning trees.

First, we construct oriented colored graphs in G using independent spanning trees of H_1, H_2. Then we prove that every colored path ends in the goal.

To edges in C_1 we assign the same color as they have in H_1. Edges (e_1, e_2), (f_1, f_2), (g_1, g_2) get the same color as edges (e_1, p), (f_1, p), (g_1, p). We rename colors in H_2 such that the edges (c, e_2), (c, f_2), (c, g_2) have the same color as the edges (p, e_1), (p, f_1), (p, g_1) in the corresponding direction. To edges in part C_2 we assign the same color as they have in H_2.

There are no colored cycles in part C_1 or in C_2 and no cycle can use edges from the 3-cut because the path which uses edges from a 3-cut ends in the goal which can not be in the cycle.

We show that from every node v we can, w.l.o.g., use red edges to reach the goal. If $v \in C_2$, then there is a red path from v to c in H_2. This path leads into one of the e_1, f_1, g_1 from these vertex leads in G then leads using red path in H_1 to the goal. If $v \in C_1$, then there was a red path in G_1. The problem is that this path can include the vertex p. In this case this path enters there from the vertex of e_1, f_1 or g_1. W.l.o.g from e_1. In G the path continues to e_2 instead of p. From e_2 it leads in H_2 to c and in G to f_1 or g_1. From one of those nodes it continues using H_1 to the goal. $\square$

4.3 Odd edge reduction

- we call an edge $e = (u, v)$ *odd* if $\deg(u) = 3$ and $\deg(v) \geq 4$.

- we use H to denote the graph $G \setminus e$ which is a graph obtained from G by replacing a vertex u and two of its adjacent edges by an edge f.

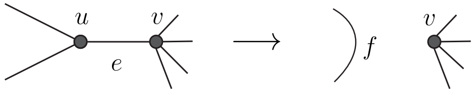


Figure 5: Odd edge reduction.

Reduction 4.3 (Odd edge reduction)**.** *If a graph G contains an odd edge $e = (u, v)$ and H has three independent spanning trees, then G also has three independent spanning trees.*

Proof. We use independent spanning trees of H. We just divide f into two edges and add the edge e. We do not have spanning trees yet since the vertex u is not included in one nor two trees. Fortunately, there is enough free edges to add it to them. See Figure 6.

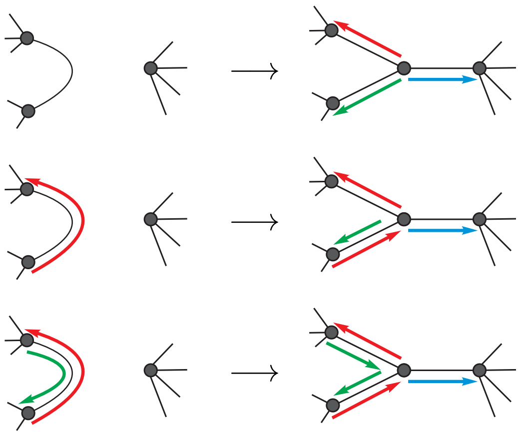


Figure 6: Independent spanning trees near an odd edge.

□

4.4 Edge near the goal reduction

- We call an edge $e = (u, v)$ *near the goal* when u is adjacent with the goal and v is not.

- Let $H = G \setminus \{e\}$ where vertices u, v and their two adjacent edges are replaced by edges f, g. This operation is called *t*opological contraction.

Reduction 4.4 (Reduction for an edge near the goal)**.** *If a cubic 3-connected graph G has an edge e near the goal and the graph H has three independent spanning trees, then G also has three independent spanning trees.*

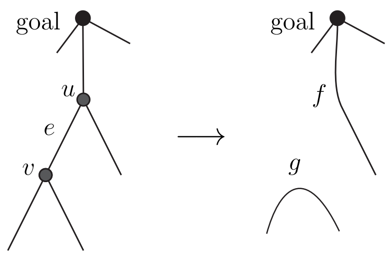


Figure 7: Edge near the goal reduction

Proof. First we prove that only one spanning tree of H uses the edge f which is adjacent to the goal and exactly two spanning trees of H use the edge g.

The graph H is cubic with n vertices, so it has $3n/2$ edges. Three of them have just one one possible orientation and the remaining edges have two possible orientations. So there is $3n - 3$ possible places for edges of the spanning trees. Each spanning tree uses $n - 1$ edges. All three of them thus use $3n - 3$ edges, so every possible place is occupied.

You can see all possible variants of independent spanning trees in H in Figure 8. We can extend spanning trees of H into spanning trees for G as it is in Figure 8. These are spanning trees because they include all vertices and we cannot create a cycle.

$\square$

5 3-connected graphs have three independent spanning trees

Theorem 5.1. *Every 3-connected graph G has three independent spanning trees.*

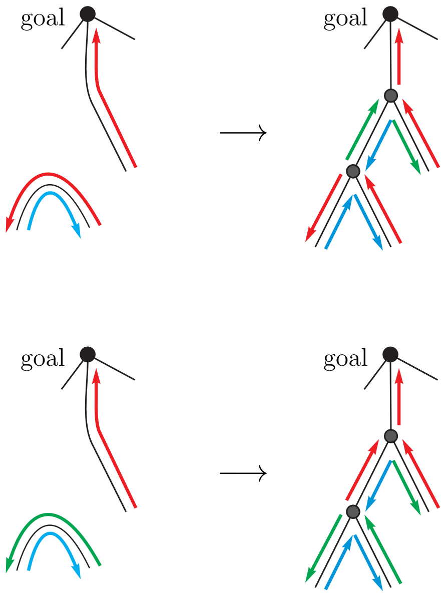


Figure 8: Independent spanning trees near the goal.

Proof. We prove the statement by induction on the number of edges. First of all, if the graph has a nontrivial 3-cut, then we use the nontrivial 3-cut reduction. If G contains an edge $e = (u, v)$ such that $\deg(u) \geq 4$ and $\deg(v) \geq 4$, then we define $H = G \setminus e$. If H has a 2-cut $\{f, g\}$, then $\{e, f, g\}$ is a 3-cut in G. A set $\{e, f, g\}$ is a nontrivial 3-cut because a vertex in a trivial 3-cut has degree three. The graph G has not a nontrivial 3-cut so H is 3-connected and smaller than G so it has three independent spanning trees from induction and we can use the useless edge reduction. The edge $e = (u, v)$ is an *odd* edge if $\deg(u) = 3$ a $\deg(v) \geq 4$. If G contains an odd edge, then we define a smaller graph H as $G \setminus e$ where the vertex u and two adjacent edges are replaced by the edge f. The graph H is 3-connected from a similar reason as in the previous case. (If H has a 2-cut $\{f, g\}$, then $\{e, f, g\}$ is a 3-cut in G. The set $\{e, f, g\}$ is nontrivial 3-cut because a vertex in a trivial 3-cut has degree three and the vertex v has larger degree and the vertex u is not in H) The graph H is 3-connected and smaller than G so it has three independent spanning trees from induction and we can use the odd edge reduction.

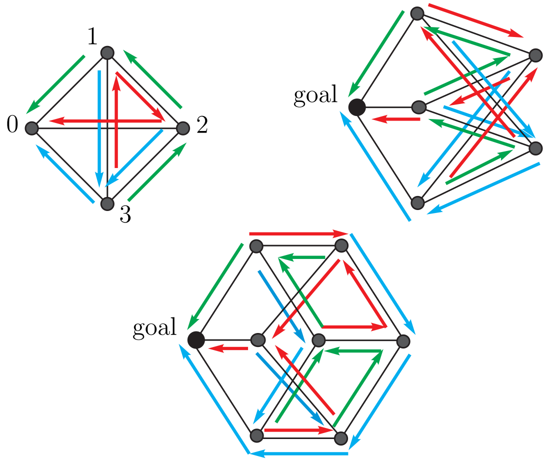


Figure 9: Independent spanning trees in K_4, $K_{3,3}$ and in the cube.

The graph G has to be cubic. If $G = K_4$ or $G = K_{3,3}$ or G is a cube, then it has three independent spanning trees. See Figure 9. We focus on the neighborhood of the goal. The goal is not in a triangle because there is a 3-cut around every triangle and if $G \neq K_4$ this 2-cut is nontrivial. We denote as Y the set of three vertices adjacent to the goal. Six edges are between Y and the set of vertices X. The size of X is at least two because G is cubic. If $|X| = 2$, then G is isomorphic to $K_{3,3}$. If $|X| = 3$, then G is isomorphic to the cube or there is a nontrivial 3-cut separating the goal, Y and X from the rest of G. Finally, if $|X| \geq 4$ then at least one vertex from X is connected to Y by exactly one edge e. We call this edge *edge near the goal*. Let $H = G \setminus \{e\}$ where the vertices u, v and their two adjacent edges are replaced by the edges f, g. If H has a 2-cut, then at least one edge of them is f or g. If not, this 2-cut was in G also. The set $\{f, g\}$ is not a 2-cut because otherwise e and two edges which were removed from G would create a nontrivial 3-cut in G. If $\{f, h \neq g\}$ is a 2-cut, then h and two edges which were removed from G create a nontrivial 3-cut in G. Similarly for the case when $\{f \neq h, g\}$ is a 2-cut. The graph H is 3-connected and smaller than G so it has three independent spanning trees from induction and we can use the edge near the goal reduction. □

6 4-connected graphs

We proved that every 2-connected and 3-connected graph has a routing. We found two, respectively three, independent spanning trees in every 2-connected, respectively 3-connected, graph and proved that they create a routing.

The proofs were very similar so there is question whether we can generalize them. However, it might not be so easy. I believe that there exist four independent spanning trees in all 4-connected graphs, see an example for K_5, but it is not true that "four independent spanning trees generate 3-routing".

When color has order red, green, blue, black, red ... and switching tables are created in a similar way as in the 3-connected case and edges $(1, 2)$, $(3, 4)$ were removed, then the prescribed tour starting on the edge $(4, 1)$ creates a cycle on vertices $1, 3, 2, 4, 1, \ldots$ and never reaches the goal.

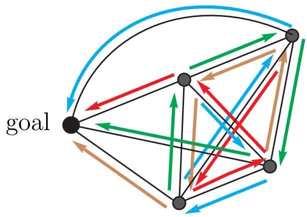


Figure 10: Four independent spanning trees for K_5.

7 Acknowledgment

We would like to thank Mario Szegedy for introducing us the problem and for many useful comments.

Construction of Graphs with a Fixed Peeling Number

Karel Tesař

1 Introduction

We say that a graph G is k-degenerate if every $H \subseteq G$ has a vertex with degree at most k. Specifically we may remove vertices one by one in order such that every removed vertex has degree at most k at the time of removing. This order of vertices we call a k-degenerate-order.

Now we generalize this order such that we always remove a vertex v with a minimum degree in a graph G. Let d be the maximum degree which we have already removed. Then after removing a vertex v we define a peeling number of v as d. It is easy to see that peeling numbers are uniquely determined. We define a peeling-order as an order of vertices with nondecreasing peeling numbers.

We may define a peeling numbers in another way. We have a several of phasions of vertex removing. In a phase i we remove vertices with a degree i or less. Vertex removed during a phase i has a peeling number i. Let us see that both process give us the same peeling numbers.

In the next section we will be interested in graphs where all of their vertices have the same peeling number.

Definition 1.1. Let us denote as FP_k a class of graphs such that $G \in FP_k$ if and only if every vertex $v \in G$ has a peeling number k.

Observation 1.2. *Class FP_k contains exactly graphs which are k-degenerate and have a minimum degree equals to k.*

In Section 2, we show a construction of graphs in FP_2, next we define extremal cases of FP_k graphs and find a construction of those graphs. Finally we show that every graph $G \in FP_k$ is a subgraph of some extremal graph from FP_k. Afterwards in Section 3, we show a generalization of trees and show that a class of extremal FP_k graphs equals this class. In Section 4, we show one result for k-degeneracy graphs. Specifically we show that one can find an independent set I in k-degenerate graph G such that the graph $G \setminus I$ is at most $(k-1)$-degenerate.

2 Constructions of FP_k graphs

We would like to find some operations such that we may construct every graph from FP_k using these operations on some set of initial graphs and such that FP_k is closed under these operations. These initial graphs are called primitive graphs with a respect to FP_k and given operations. First, we find operations for the class FP_2 where primitive graphs are isomorphic to disjoint union of triangles and we prove the following theorem.

Theorem 2.1. *If $G = (V, E)$ is a graph from FP_2 with a component C which is not a triangle, then we can apply one of the following operations to obtain a graph $G' \in FP_2$ where $|V(G)| > |V(G')|$.*

(a) Erase a vertex $v \in V$ such that $\deg(v) = 2$ *and every neighbour u of v satisfies* $\deg(u) > 2$

(b) A contraction of an edge $uv \in E$ such that $\deg(u) = \deg(v) = 2$ *and u and v have no common neighbour.*

(c) Erase vertices $u, v \in V$ such that $\deg(u) = deg(v) = 2$ *and u and v have a common neighgbour w such that $deg(w) > 3$.*

(d) Erase a triangle uvw such that $\deg(u) = \deg(v) = 2 \quad \& \quad \deg(w) = 3$.

Proof. We know that there exists a vertex v with a degree two in every component. We proceed by a case analysis according to the neighbors u_1, u_2 of v where we assume that $\deg(u_1) \leq \deg(u_2)$.

- Both neighbors have degree greater than two. Then we may erase a vertex v. According to 2-degeneracy there still exists another vertex with degree two and according to FP_2 there is no vertex with degree less than two.
- $\deg(u_1) = 2$ and u_2 is not neighbour of u_1. Then we may contract an edge vu_1. The new vertex has degree two.
- $\deg(u_1) = 2 \quad \& \quad \deg(u_2) > 3$ and u_2 is a neighbour of u_1. Then we erase vertices v and u_1 and we still stay in FP_2 because of 2-degeneracy.
- $\deg(u_1) = 2 \quad \& \quad \deg(u_2) = 3$ and u_2 is a neighbour of u_1. If another neighbour of u_2 has degree two, we may contract an edge between it and u_2. In the other case we may erase a triangle vu_1u_2.

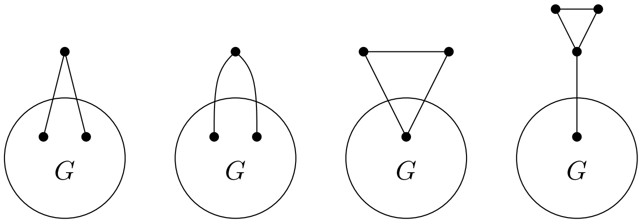


Figure 1: Allowed operations for a construction of graphs from FP_2. From the left to the right there are a vertex adding, an edge subdivision, an edge adding and a triangle adding.

- $\deg(u_1) = 2 \quad \& \quad \deg(u_2) = 2$ and u_2 is a neighbour of u_1. So this component is a triangle.

□

Corollary 2.2. *Every graph $G \in FP_2$ can be constructed from triangles by the following operations*

- *adding vertices of degree two,*
- *subdivision of edges,*
- *adding two vertices of degree two connected by an edge,*
- *adding a triangle and connecting it to the rest of the graph by an edge.*

Note that the class FP_2 is closed under these operations.

Now we define extremal graphs in FP_k. We use them later for construction of the class FP_k for other values of k. We say that $G \in FP_k$ on n vertices is extremal if it has the maximal number of edges. Here is the formal definition.

Definition 2.3. We say that a graph $G = (V, E)$ from FP_k on n vertices is *extremal if for every edge $e \notin E$ the graph $G + e$ is not in FP_k.*

First we notice that every FP_k extremal graph on n vertices has the same number of edges.

Lemma 2.4. *Let G be an extremal graph from FP_k on n vertices. Then G has exactly $\binom{k}{2} + k(n-k)$ edges.*

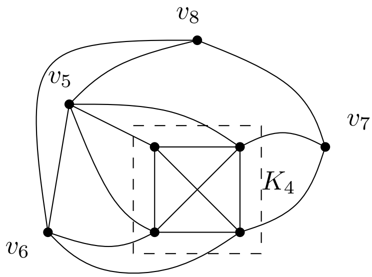


Figure 2: Example of an extremal graph from FP_3. First we have a clique K_3, by adding one vertex we get a clique K_4 and then we add vertices v_5, v_6, v_7, v_8.

Proof. We know that G is k-degenerate and it has a minimum degree k. So every vertex v_i in k-degenerate-order has $\min\{k, i\}$ forward edges where i is a backward position in that order. If this was not true then we could add a new edge and stay into a FP_k class.

Now it follows that the number of edges is exactly $\binom{k}{2} + k(n-k)$. □

From this fact and k-degenerate-order follows the construction of extremal FP_k graphs.

Claim 2.5. *Every extremal graph $G \in FP_k$ on $n > k$ vertices can be constructed from a clique K_k by adding $(n-k)$ vertices of degree k.*

Next we prove that any G from FP_k is a subgraph of some extremal graph from FP_k on the same number of vertices. To prove this, we look at k-degenerate-order of G and expand it into k-degenerate-order of some extremal graph.

Theorem 2.6. *Let G be an graph from FP_k on n vertices. Then there exists an extremal graph $H \in FP_k$ such that H also has n vertices and G is a subgraph of H.*

Proof. We take a k-degenerate-order of G – every vertex $v_i \in G$ has at most $\min\{k, i\}$ forward edges. Additionally we construct an extremal graph H and add to every vertex all forward edges which are also in G. The rest of the edges can be added arbitrarily. □

From the last theorem we get a construction of any graph $G \in FP_k$. First we construct an arbitrary extremal H such that $G \subseteq H$ and then we remove some edges. Let us note that the earlier construction for FP_2 is more convenient for proofs by induction, because we are only adding new vertices and edges and we never delete anything. On the other hand if we want to obtain a result only for extremal graphs from FP_k, then the last construction is also usable.

3 Generalization of trees

In trees every leaf has degree one and the leafs form an independent set and if we remove all of them we obtain another tree. Repeating this process, we end with K_1 or K_2. Now we try to generalize this idea. For a given graph G and the positive integer k we apply the following peeling process.

Algorithm 3.1. *peel*(G, k)*:*

1. $I := \{v : \quad \deg(v) = k\}$
2. *If I is not independent or there exists a vertex v with degree lower than k then return G.*
3. $G := G \setminus I$
4. *Go to step 1.*

This algorithm may output various graphs. We focus on cases where we end up with a clique K_k or K_{k+1}.

Definition 3.2. Let k be a natural number and define a class of generalized trees $\mathcal{T}_k$ such that graph G is in $\mathcal{T}_k$ if and only if *peel*$(G, k) = K_k$ *or* *peel*$(G, k) = K_{k+1}$.

We say that a vertex of $G \in \mathcal{T}_k$ with degree k which is not in the final clique K_k or K_{k+1} is a leaf.

The interesting thing, about this class of graphs, is that the class of generalized trees is equivalent to extremal FP_k graphs.

Claim 3.3. *Let EFP_k be a class of extremal graphs from FP_k. Then for every k : $\mathcal{T}_k = EPF_k$.*

Proof. If we take some $G \in \mathcal{T}_k$ then we may repeatively remove some indepentent sets $I_1, \ldots, I_l$ of vertices of degree k and end with K_k or K_{k+1}. If we remove the vertices one by one we actually get a construction of extremal case of FP_k. So G is also in EFP_k

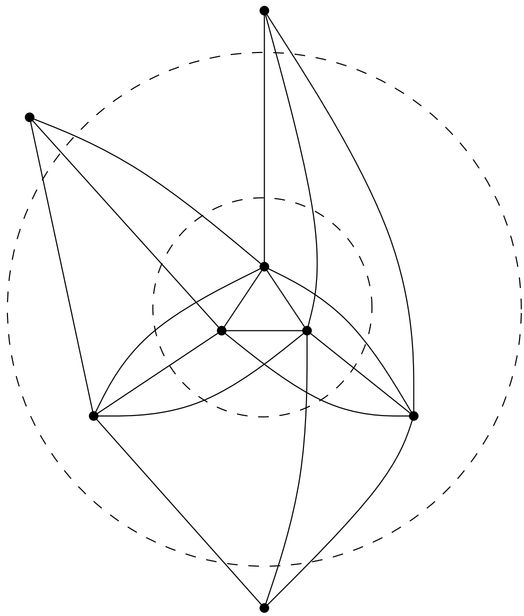


Figure 3: Example of a tree from $\mathcal{T}_3$.

For a given graph $G \in EFP_k$ we prove by induction on $|V(G)|$ that G is also contained in $\mathcal{T}_k$. The cliques K_k and K_{k+1} trivially are in $\mathcal{T}_k$. Let us assume that G is a graph with at least $k+2$ vertices. We have a k-degenerate order of G where every vertex except the last k has k forward edges. Therefore every vertex with degree k has only forward edges. So they form an independent set I if we remove this independent set we get $G \setminus I$ which is also from EFP_k and smaller. So $G \setminus I$ is from $\mathcal{T}_k$ and if we add I into this graph as a leafs we get that G is also from $\mathcal{T}_k$. $\square$

Here follows another interesting thing about $\mathcal{T}_k$.

Theorem 3.4. *For every graph G from $\mathcal{T}_k$ there exist trees $T_1, T_2, \ldots, T_k \in \mathcal{T}_1$ such that G is an edge disjoint union of $T_1, T_2, \ldots, T_k$ and every leaf in G is also a leaf in every tree T_i.*

Proof. We prove it by induction. For a clique K_{k+1} on vertices $\{1, 2, \ldots, k+1\}$, for every $1 \leq i \leq k$ we define T_i as a graph on vertices $\{1, 2, \ldots, k+1\}$

with edges $\{\{i, j\} : \quad j \geq i\}$. Obviously these trees are edge disjoint and every edge is covered.

If we have a larger graph G, then we remove some leaf $l \in V(G)$. By induction on $|V(G)|$ we have trees $T_1, \ldots, T_k$ which are edge disjoint and their union is $G \setminus \{l\}$. Now we add a vertex l in all of them such that every T_i stays to be connected. We may do this because there are at most $i-1$ vertices which are not in T_i. So first we choose a neighbor of l which is in T_k, then we choose a neighbor which is in T_{k-1} and so on until the last one which remains is in T_1. Obviously every T_i is still a tree and T_i and T_j are edge disjoint for every i and j. We also covered every edge incident with l. $\square$

4 k-degeneracy and independent sets

Theorem 4.1. *For every k-degenerate graph $G = (V, E)$ there exists an independent set $I \subseteq V$ such that $G \setminus I$ is at most $(k-1)$-degenerate.*

Proof. We take a k-degenerate order and greedily find a proper coloring of a graph G with colors $\{1, 2, \ldots, k+1\}$. We take vertices one by one from the last one to the first one. Everyone of them has at most k forward edges so there exists a color which we may use. We always use the smallest possible color.

Now we claim that a set of vertices with color 1 is the independent set we are looking for. Every vertex v either have color 1 or some of it's forward neighbour have color one. So if a vertex v remains, then its forward degree decrease at least by one and hence the degeneracy of $G \setminus \{v : v \text{ has color } 1\}$ is at most $(k-1)$-degenerate. $\square$

References

[1] J. Abello and F. Queyroi. "Fixed Points of Graph Peeling," (2013), Advances in Social Networks.

[2] V. Batagelj and M. Zavernik. "An $\mathcal{O}(m)$ algorithm for cores decomposition of networks," (2003), arXiv:cs/0310049.

[3] D. R. Lick, and A. T. White. "k-degenerate graphs," Canad. J. Math, vol. 22, pp. 10821096, 1970.

[4] A. Montresor, F. D. Pellegrini, and D. Miorandi. Distributed k-core decomposition, CoRR, vol. abs/1103.5320, 2011.

[5] S. Seidman, Network structure and minimum degree, Social networks, vol. 5, no. 3, pp. 269287, 1983.

Dynamic Data Structure for Tree-Depth Decomposition

Zdeněk Dvořák, Martin Kupec and Vojtěch Tůma[1]

The concept of tree-depth, introduced in [12], appears prominently in the sparse graph theory and in particular the theory of graph classes with bounded expansion, developed mainly by Nešetřil and Ossona de Mendez [11, 13, 14, 15, 16, 17]. One of its many equivalent definitions is as follows. The tree-depth $\mathrm{td}(G)$ of an undirected simple graph G is the smallest integer t for that there exists a rooted forest T of height t with vertex set $V(G)$ such that for every edge xy of G, either x is ancestor of y in T or vice versa—in other words, G is a subgraph of the closure of F.

Alternatively, tree-depth can be defined using (and is related to) rank function, vertex ranking number, minimum elimination tree or weak-coloring numbers. Futhermore, a class of graphs closed on subgraphs has bounded tree-depth if and only if it does not contain arbitrarily long paths. Tree-depth is also related to other structural graph parameters—it is greater or equal to path-width (and thus also tree-width), and smaller or equal to the smallest vertex cover.

Determining tree-depth of a graph is NP-complete in general. Since tree-depth of a graph G is at most $\log(|G|)$ times its tree-width, tree-depth can be approximated up to $\log^2(|G|)$-factor, using the approximation algorithm for tree-width [1]. Furthermore, for a fixed integer t, the problem of deciding whether G has tree-depth at most t can be solved in time $O(|G|)$. Minimal minor/subgraph/induced subgraph obstructions for the class of all graphs of tree-depth at most t are well characterized, see [4]. Clearly, tree-depth is monotone with respect to all these relations. For more information about tree-depth, see the book [18].

A motivation for investigating structural graph parameters such as tree-depth is that restricted structure often implies efficient algorithms for problems that are generally intractable. Structural parameters have a flourishing relationship with algorithmic meta-theorems, combining graph-theoretical and structural approach with tools from logic and model theory—see for instance [8]. A canonical example of a meta-theorem using a structural parameter is the result of Courcelle [3] which gives linear-time algorithms

[1] The work leading to this invention has received funding from KONTAKT II LH12095 and SVV 267313.

for properties expressible in MSO logic on classes of graphs with bounded tree-width.

Tree-depth is similar to tree-width, in the sense that it measures "tree-likeness" of a graph and also allows decomposition with algorithmically exploitable properties. However, tree-depth is more restrictive, since bounded tree-depth implies bounded tree-width, but forbids the presence of long paths. Long paths turn out to be related to the hardness of model checking for MSO logic [10, 6]. This motivated a search for meta-theorems similar to [3] on more restricted classes of graphs, such as the result of Lampis [9] that provides algorithms with better dependence on the size of the formula for classes such as those with bounded vertex cover or bounded max-leaf number. This result was subsequently generalized by Gajarský and Hliněný to graphs with bounded tree-depth [7].

In the usual static setting, the problem is to decide whether a graph given on input has some fixed property P. Our work is of dynamic kind, that is, the considered graph gradually changes over time and we have to be able to answer any time whether it has the property P. One application comes immediately in mind. Graphs modelling many natural phenomena, such as the web graph, graphs of social networks or graphs of some physical structure all change rapidly. However, there is another area where this dynamic approach is useful. For example, one reduces a graph by removing edges, and each time an edge is removed, some procedure has to be performed. Instead of running the procedure from scratch every time, it makes sense to keep some dynamic information. Classical examples are the usage of a disjoint-find-union data structure in minimal spanning tree algorithms [2] or Link-cut trees for network flow algorithms [19]. A more recent example is a data structure for subgraph counting [5] with applications in graph coloring and social networking.

The main theorem of our paper follows.

Theorem 0.1. *Let ϕ be a MSO$_2$ formula and $D \in \mathbb{N}$. There exists a data structure for representing a graph G with $\mathrm{td}(G) \leq D$ supporting the following operations:*

- *insert edge e, provided that $\mathrm{td}(G + \{e\}) \leq D$,*
- *delete edge e,*
- *query—determine whether G satisfies the formula ϕ.*

The time complexity of deletion depends on D only, in particular, it does not depend on ϕ or $|G|$. The time complexity of insertion depends on ϕ and D, but does not depend on $|G|$. The time complexity of the initialization of the data structure depends on ϕ, D and $|G|$. The query is done in constant time, as is addition or removal of an isolated vertex.

The dependence of the initialization and edge insertion is roughly a tower of height D where the highest element of the tower is the number of nested quantifiers of ϕ squared.

The basic idea of the data structure is to explicitly maintain a forest of smallest depth whose closure contains G, together with its compact constant-size summary obtained by identifying "equivalent" subtrees. This summary is sufficient to decide the property expressed by ϕ, as outlined in the following paragraph.

Two graphs are said to be n-equivalent, if they satisfy the same first order formulas with at most n quantifier alterations—that is, for instance, of the form $\forall x_{1..i_1} \exists x_{i_1+1..i_2} \forall x_{i_2+1..i_3} \dots \exists x_{i_{n-1}+1..i_n} \phi(x_{1..n})$, where ϕ is quantifier-free. This concept of n-equivalency is of practical use for model checking. It serves to reduce the investigated graph to a small one, so that time-expensive approaches as brute force become possible (a technique known as kernelization). An example of such application is the following theorem (from section 6.7 of [18]): for every D, n exists N such that every graph G with $\mathrm{td}(G) \leq D$ is n-equivalent to one of its induced subgraphs of order at most N. This can be extended even to labeled graphs. In our work we use a similar theorem, taken from [7]. Informally, the result says that when one is interested in checking whether a specific formula is true on a class of trees of bounded depth, then one can also assume bounded degree. This allows us to only maintain the summary of the tree-depth decomposition as described above.

In the rest of the paper, the first section reviews necessary definitions and tools we use, and the second section describes in detail the data structure and its operations. We conclude the paper with the application to dynamic model checking.

1 Preliminaries

In this paper, all trees we work with are rooted. For simplicity, we assume in this section that all graphs we work with are connected. If we encounter a disconnected graph, we consider each of its connected components individually.

Let T be a tree, the *depth* of T is the maximum length of a path from the root of T to a leaf of T. Two trees are isomorphic if there exists a graph-isomorphism between them such that the root is preserved under it. Mostly we will work with trees with vertices labelled from some set of l labels – two l-labelled trees are l-isomorphic if they are isomorphic as trees and the isomorphism preserves labels.

The *closure* $\operatorname{clos}(T)$ is the graph obtained from T by adding all edges (x, y) such that x is an ancestor of y, and $x \neq y$. For instance, the closure of a path is a complete graph. The *tree-depth* $\operatorname{td}(G)$ is the minimum number t such that there exists a forest T of depth t such that $G \subseteq \operatorname{clos}(T)$. For instance, the tree-depth of a path on n vertices is $\lceil \log_2(n+1) \rceil$. A *limb* of a vertex $v \in T$ is the subgraph induced by some of the children of v. A second-order logic formula ϕ is in MSO_1 logic, if all second-order quantifiers are over sets of elements (vertices) and the language contains just the relation $edge(u, v)$.

The following result is a simplification of Lemma 3.1 from [7].

Lemma 1.1. *Let ϕ be an MSO_1 sentence, l, $D \in \mathbb{N}$. Then there exists a number S with the following property. Let T be an l-labelled tree of depth at most D with vertices labelled with l labels, and v a vertex of T. If v has more than S pairwise l-isomorphic limbs, then for the tree T' obtained by deleting one of those limbs we have that*

$$T' \text{ satisfies } \phi \iff T \text{ satisfies } \phi.$$

The Lemma implies in particular that with respect to ϕ-checking there are only finitely many l-labelled trees of depth at most D – that is, every l-labelled tree of depth at most D is ϕ-equivalent to some l-labelled tree of depth at most D and maximum degree at most S. We call such trees *ϕ-minimal.*

Let G be a graph of tree-depth D, the *tree decomposition* T of G is a 2^{D-1}-labelled tree such that $G \subseteq \operatorname{clos}(T)$, where a vertex v is labelled by a 0-1 vector of length $D - 1$ that encodes the edges between v and the vertices on the path from v to the root (1 whenever the edge is present, 0 otherwise). Let l_D be a set of labels we describe later, *compressed tree decomposition* of the graph G is an l_D-labelled tree C obtained from a tree decomposition T of G as follows. For every vertex, all its limbs that are pairwise-isomorphic are deleted except for one representative, in which we

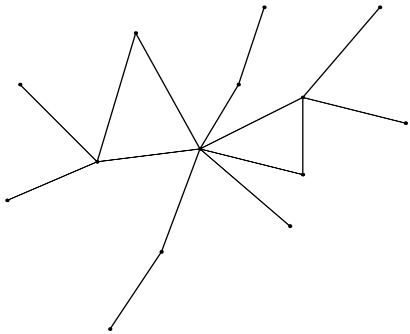


Figure 1: Graph

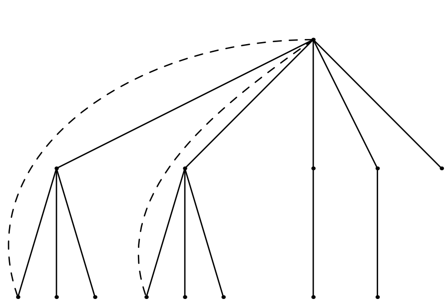


Figure 2: Tree-depth decomposition

additionally store the number of these limbs. Vertices of C are called *cabinets*, and the underlying tree decomposition T is called a *decompression* of C. A set of all vertices corresponding to the same cabinet (that is, inducing l_D-isomorphic limbs) and having the same vertex as a father in the decompression is called a *drawer*. Thus every cabinet is disjointly partitioned into drawers. For an example how a graph, its tree decomposition and compressed tree decomposition look like, see the figures on page 73 (in the compressed tree decomposition, the number next to the drawers denotes how many vertices are there in each drawer).

Now we describe the labelling. We start inductively, with l_0 being just a set of vectors of length $D-1$. Assume that ϕ is some fixed formula we specify later (in Section 2.5) and let S be the number obtained from applying Lemma 1.1 to it. Let B be a cabinet that induces a subtree of depth $t' \leq t$ in C. The label of B consists of the label of a corresponding vertex b of T and of a vector *vec* with entry for every l_{t-1}-labelled ϕ-minimal tree M of depth smaller than t' with value

$$vec_M = \min\{S, \text{number of limbs of } b \text{ which are } \phi\text{-equivalent to } M\}.$$

During the update operations, we will be occasionally forced to have more than one cabinet for a given isomorphism type (that is, a cabinet will have two pairwise-isomorphic limbs). Both the decomposition and the individual cabinets that have isomorphic children will be called *dirty*.

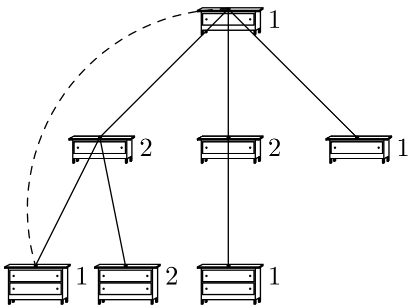


Figure 3: Compressed graph

2 Data structure

Our data structure basically consists of storing some extra information for every vertex v of the represented graph G, and of a compressed tree-depth decomposition T of G with depth at most D. We will store the following for every $v \in G$.

- Label of the cabinet corresponding to v, that is, the vector of its neighbors on the path to root and the vector with the numbers of limbs of v isomorphic to individual ϕ-minimal trees.
- Pointer to the father of v in T (more precisely, pointer to a vertex u of G that is the father of v in the decompression of T). However, in some operations we need to change the father of all vertices in a drawer at once – thus instead of storing father individually for every vertex, for every drawer we will maintain a pointer to the common father of the vertices in this drawer, and every vertex in the drawer will have a pointer to this pointer.
- Linked list of sons of v in T. This is again implemented by having a linked list of drawers at v, and for every drawer in it, a linked list of vertices in this drawer.

Additionally, we keep the vertex v which is the root of T and we call it r – we again assume connectedness of G in this section, otherwise we keep a list of roots corresponding to individual components.

2.1 Extraction of a path

In this subsection we describe an auxiliary operation of extracting a path. It can be seen as a temporary decompression of a part of T in order to make some vertex accessible. The result of extracting v from T is a dirty compressed tree decomposition T' of G, such that on the cabinets in T' corresponding to $r - v$ path there are no cabinets to which corresponds more than one vertex of G.

First, we find the vertices of the $r - v$ path, and the corresponding cabinets in T. This is done by simply following the father-pointers from v, and then by going backwards from r, always picking the cabinet that corresponds to the label of the vertex on the $r - v$ path. Then, for every cabinet B on this path with more than one vertex, let b be the vertex of the $r - v$ path lying in B, and c its father – which we assume to be the only vertex in its cabinet, C. We remove b from the lists of sons of c of the label of b, and move b into a new list for c, and do the corresponding change in T', that is, creating a new cabinet of the same label as a son of C, thus making C a dirty cabinet.

The complexity of this operation is clearly linear in D.

2.2 Edge deletion

Edge deletion is simple – let vu be the edge to be deleted, with v the lower vertex (in the tree-order imposed by T). We extract the vertex v from T. Now u lies on the $r - v$ path, and as there are no other vertices in the cabinets on the corresponding path in T, we remove the edge vu from the graph and change the labels for the cabinets and vertices accordingly. The only affected labels are on the $r - v$ path, and we will precompute during initialization what the label should change into. It can also happen that removal of such edge disconnects the graph – this also depends only on labels and thus will be precomputed in advance. When such situation occurs, we split T into two components – the new root depends only on the labels, and the vertices for which labels change are only on the $r - v$ path.

Now, we need to clean the dirty cabinets. As the only dirty cabinets are on the $r-v$ path, we traverse this path, starting from v and going upwards, and for every vertex w in a dirty cabinet, we compare the label of w with the labels of other present drawers at the father of w, and move w to the correct drawer/cabinet.

The complexity of this operation is clearly linear in D.

2.3 Rerooting

Rerooting is also an auxiliary operation, which will allow us to easily handle the edge insertion. This operation takes a compressed tree decomposition T and a vertex r_N of G, for which we have a guarantee that there is a tree decomposition with depth at most D such that r_N is its root, outputs one such compressed tree decomposition T' and updates data for vertices in G accordingly. In this subsection we denote by r_O the root of T, that is, the old root.

We proceed as follows:

1. extract r_N from T,
2. remove r_N from T entirely,
3. consider the connected components thereof – those that do not contain r_O have depth $< D$ and thus can be directly attached under r_N. Recurse into the component with r_O.

Only the third point deserves further explanation. The components are determined by the labels only, so we will precompute which labels are in which components and what vertices are the roots of the components. Every connected component of $T - r_N$ that does not contain r_O must have as its highest vertex (under the tree-order) a son of r_N, thus these components are already in their proper place. For the component C with r_O, either it has depth $< D$ and thus can be attached under r_N, with r_O being a son of r_N. We have to deal with two details – firstly, there might be some edges to r_N from vertices that were above r_N in T – but none of these vertices was in a cabinet with more than one vertex, thus we only change the labels accordingly.

Secondly, the limbs of r_N in T that are in C have no father after removal of r_N. But as they are in C, for every such limb there is an edge from it to some vertex on the $r_N - r_O$ path T. Choose lowest such vertex, and make it new father for that limb. This refathering is done by using the pointers for the drawers – note that every cabinet that is a root of such limb consist only of single drawer, thanks to the extraction of r_N. Thus the total number of operations we have to do is linear in D and the maximum number of children of r_N, which is l_d. As in the case of edge deletion, we have to clean dirty cabinets (which are in C) in the end. This can again be done by simply comparing labels on that former $r_N - r_O$ path.

However, it might happen that C has depth exactly D. But we are guaranteed that there exist a tree decomposition with r_N as a root, which implies that there exists a tree decomposition of C with depth $D-1$. If we know which vertex can serve as a root of such decomposition, we can apply the operation recursively. We describe the procedure to find a root in Section 2.5. An additional thing we have to care about is that some vertices of C have an edge to r_N – this information has to be preserved in the recursive call. But the number of such vertices is bounded by a function of D and thus it is not a problem – we only modify their labels accordingly. After this recursive call, we again clean dirty cabinets.

The complexity of this operation for one call is linear in $D + l_d +$ time to find new root, and there are at most D recursive calls.

2.4 Edge insertion

Let u, v be two vertices not connected with an edge, such that $G + uv$ has treedepth at most D, we now describe how to add such edge. If the edge uv respects the tree-order (that is, either u lies on $v - r$ path or vice versa), we just extract the lower of the two vertices, add the edge, and get rid of dirty cabinets.

Otherwise, there exists a vertex r_1 which is a root of some tree decomposition of $G + uv$. We describe the procedure for finding it in Section 2.5. Reroot into this vertex to obtain decomposition T_1. Now, u and v must be in the same connected component C_1 of $T_1 - r_1$. Again, unless the edge uv respects the tree-order now, we can find a vertex r_2 in C_1 which is a root of some tree decomposition of $C_1 + uv$ of depth at most $D - 1$, and reroot into it to obtain decomposition T_2 of C_1, with u and v lying in the same component C_2 of $C_1 - r_2$. Carrying on in the obvious manner, this process stops after at most D iterations.

The complexity of this operation is $O(D\cdot$ complexity of rerooting).

Finally, we just remark that addition and removal of a vertex (without incident edges) is implemented trivially by just adding/removing new component with the corresponding label.

2.5 Finding a root

Let us recall what we have to face in this section. We want to find a vertex v such that there is a tree T of depth at most $t' \leq t$ such that its closure contains the connected component C of the graph $G+(a, b)-\{v_1, v_2, \ldots, v_k\}$

as a subgraph and v is a root of T. The vertices $v_1, v_2, \dots, v_k$ correspond to the roots found in previous applications of this procedure, (a, b) denotes the edge we are trying to add.

At this point we define the formula ϕ according to which we constructed the labelling of our trees. Let $\gamma(C)$ be a formula which is true whenever C is connected — this is easily seen to be expressible in MSO_1 logic — and $\tau_d(G)$ the following formula:

$$\tau_d(G) = (\exists v \in G)(\forall C \subseteq G)(\gamma(C - \{v\}) \Rightarrow \tau_{d-1}(C - \{v\})),$$

with $\tau_1(v)$ being always true. Then $\tau_d(G)$ says that there exists a tree T with depth at most d such that $G \subseteq \mathrm{clos}(T)$. Furthermore, as we need to express the addition of an edge, we work with the logic with two extra constants a, b, and modify the formula for γ accordingly to obtain γ' and τ'. The resulting formula τ'_t is the formula ϕ.

Using Lemma 1.1, we construct all ϕ-minimal trees – note that we have to consider every possible evaluation of the constants a, b, that is, we construct all trees of depth at most t such that no vertex has more than S pairwise-isomorphic limbs, and then for every two of labels, we choose two arbitrary vertices having that label, and choose them to be a and b. For every such minimal tree, we evaluate the formula, that is, we find which vertex is to be the root of the tree decomposition. It might happen that the formula is false, that is, no such vertex exists, which means we evaluated a and b so that the graph has tree-depth greater than d. But such evaluation will not occur during the run of the structure — recall that we restricted the edge additions — and thus we can safely discard these minimal trees. Thus for every minimal tree we store the label of the vertices that can be made root, and when applying the rerooting subroutine, we find an arbitrary vertex of this label. This has complexity at most D, because when looking for the given vertex, we follow first pointer from the corresponding linked list of children for a vertex.

This means that the total complexity of the edge insertion is $O(D(D + l_D))$. Finally, let us remark on the complexity of initialization. From [7] we conclude that l_D, that is, the number of ϕ-minimal trees, is roughly a tower of 2's of height linear in D, to the power $|\phi|^2$. The complexity of operations we do for every ϕ-minimal tree is bounded by a polynomial in D and l_D.

2.6 Dynamic model checking

We now describe how to modify the structure so that it also allows queries of the form "does G satisfy the formula φ", where φ is some fixed MSO formula. The modifications affect only the Section 2.5. Instead of using just the formula ϕ to obtain the ϕ-minimal trees, we apply the Lemma 1.1 to the formula φ also and in the construction of the minimal trees and the labelling, we use the higher of the two numbers obtained from the Lemma. Then for every such obtained minimal tree we evaluate whether it satisfies φ or not, this time without evaluating the constants a, b.

References

[1] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, *Approximating treewidth, pathwidth, frontsize, and shortest elimination tree*, Journal of Algorithms, 18 (1995), pp. 238–255.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, The MIT Press, 2009.

[3] B. Courcelle, *The monadic second-order logic of graphs. i. recognizable sets of finite graphs*, Information and computation, 85 (1990), pp. 12–75.

[4] Z. Dvořák, A. C. Giannopoulou, and D. M. Thilikos, *Forbidden graphs for tree-depth*, European Journal of Combinatorics, 33 (2012), pp. 969–979.

[5] Z. Dvořák and V. Tůma, *A dynamic data structure for counting subgraphs in sparse graphs*, in Algorithms and Data Structures Symposium, WADS 2013, London (Ont.), Proceedings, Springer, 2013. To appear.

[6] M. Frick and M. Grohe, *The complexity of first-order and monadic second-order logic revisited*, in Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on, IEEE, 2002, pp. 215–224.

[7] J. Gajarsky and P. Hlineny, *Faster Deciding MSO Properties of Trees of Fixed Height, and Some Consequences*, in IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012), D. D'Souza, T. Kavitha, and J. Rad-

hakrishnan, eds., vol. 18 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2012, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 112–123.

[8] M. Grohe and S. Kreutzer, *Methods for algorithmic meta theorems*, Model Theoretic Methods in Finite Combinatorics, 558 (2011), pp. 181–206.

[9] M. Lampis, *Algorithmic meta-theorems for restrictions of treewidth*, Algorithmica, 64 (2012), pp. 19–37.

[10] M. Lampis, *Model checking lower bounds for simple graphs*, CoRR, abs/1302.4266 (2013).

[11] J. Nešetřil and P. Ossona de Mendez, *Linear time low tree-width partitions and algorithmic consequences*, in Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, STOC '06, ACM, 2006, pp. 391–400.

[12] ——, *Tree-depth, subgraph coloring and homomorphism bounds*, European Journal of Combinatorics, 27 (2006), pp. 1022–1041.

[13] ——, *Grad and classes with bounded expansion i. decompositions*, European Journal of Combinatorics, 29 (2008), pp. 760–776.

[14] ——, *Grad and classes with bounded expansion ii. algorithmic aspects*, European Journal of Combinatorics, 29 (2008), pp. 777–791.

[15] ——, *Grad and classes with bounded expansion iii. restricted graph homomorphism dualities*, European Journal of Combinatorics, 29 (2008), pp. 1012–1024.

[16] ——, *First order properties on nowhere dense structures*, J. Symbolic Logic, 75 (2010), pp. 868–887.

[17] ——, *On nowhere dense graphs*, European J. Combin., 32 (2011), pp. 600–617.

[18] ——, *Sparsity: Graphs, Structures, and Algorithms*, vol. 28, Springer, 2012.

[19] D. D. Sleator and R. Endre Tarjan, *A data structure for dynamic trees*, Journal of computer and system sciences, 26 (1983), pp. 362–391.

DIMATIA/MCW Program Report

Kaleigh Clary
Hendrix College

1 Introduction

My experiences in Prague have provided me with a unique insight into the Czech culture and exposed me to many interesting problems in discrete mathematics.

2 Workshop Talks

I appreciated the lectures from Martin Balko, Dr. Andrew Goodall and Dr. Jiří Fiala. They provided excellent introductions to various proofs of Cayley's formula, graph chromatic polynomials, and Hamiltonian paths in interval graphs.

2.1 Lectures: Dr. Andrew Goodall

A graph coloring is an assignment of colors to each vertex of the graph. A proper coloring of a graph is one in which no adjacent vertices share the same color. The chromatic polynomial $P(G, k)$ of a graph $G = (V, E)$ is the number of proper colorings of G using k colors. $P(G, k)$ can be defined recursively using deletion and contraction of the edges of G. The Fundamental Reduction Theorem states $P(G, k) = P(G - uv, k) - P(G/uv, k)$ for some edge $uv \in E$ [1].

2.2 Problem Solutions

If G is a tree on n vertices, find $P(G, k)$.

Let $S(n)$ state that for a tree G on n vertices, $P(G, k) = k(k-1)^{n-1}$.

Since there are k ways to color one vertex with k colors, and $k(k-1)^0 = k$, $S(1)$ is true.

Assume $S(i)$ is true. Note that $P(G_1 \cup G_2, k) = P(G_1, k)P(G_2, k)$, where $G_1 \cup G_2$ is the disjoint union of a graph. Because a tree is connected, removing any edge results in a disjoint union of two trees. Let e be an edge connected to a vertex of degree 1 in a tree T on $i + 1$ vertices and T_1 be the vertex of degree 1 with edge e, where $T_1 \cup T_2 = T$. So T_2 has i vertices. Then the chromatic polynomial of T is

$$\begin{aligned}
P(T, k) &= P(T - e, k) - P(T/e, k) \\
&= P(T_1, k)P(T_2, k) - P(T/e, k) \\
&= kP(T_2, k) - P(T_2, k) \\
&= (k - 1)P(T_2, k) \\
&= (k - 1)k(k - 1)^{i-1} \\
&= k(k - 1)^i,
\end{aligned} \tag{1}$$

so $S(i)$ is true. By the Principle of Mathematical Induction, a tree on n vertices has $k(k-1)^{n-1}$ proper k-colorings.

3 Midsummer Combinatorial Workshop Talks

I really enjoyed hearing so many interesting problems presented by the mathematicians at the MCW. I found Dr. Maria Garijo's presention over a solution to the the red-blue intersection problem in $\mathbb{R}^2$ most compelling.

3.1 Red-Blue Intersection Problem

We are given a set of n red and blue points in $\mathbb{R}^2$ in general position i.e., where no three points are colinear. We draw complete graphs on the set of points per color, and we want to report the number of intersections between different colored line segments in the plane. This has applications in map overlays in GIS and polygon clipping in computer graphics, and is related to separability.

Let s_b be the set of blue line segments intersecting red line segments. We can determine s_b by using the convex hull of the red lines. Blue points on the interior form a complete graph from line segments that do not intersect any red line segments. However, lines that start from points on the exterior and end on the interior of the hull will intersect red line segments. Because there are still red line segments inside the hull that may intersect the blue segments on th interior, this algorithm must iterate over convex regions of the plane formed by red line segments. We can do this by defining an equivalence relation $b_i \sim b_j$ if an only if $b_i b_j$ crosses no red line. This partitions the lines into planar subdivisions. s_b is then the set of lines starting and ending at points belonging to different equivalence classes.

3.2 Open Problem: Extending to 3D

Dr. Garijo closed by introducing an open problem for extending the red-blue intersection solution to $\mathbb{R}^3$ using monochromatic triangles. I attempted to identify potential approaches to a solution. Since monochromatic triangles in $\mathbb{R}^3$ would define a plane, I thought we could start by finding a solution for red-blue intersections of planes and find a reduction to line segments. Garijo's algorithm would essentially remain the same by dividing the space into convex polyhedra by defining a similar equivalence relation $b_i \sim b_j$ if and only if $b_i b_j$ does not intersect a red plane. It is perhaps possible this could be restricted to edge-edge intersections only.

Another approach I considered is based on an existing algorithm for separability problems in $\mathbb{R}^3$ using infinite prisms. The algorithm determines whether there exists an infinite prism containing all the blue points and no red points [2]. In this algorithm, the infinite prism is defined by a direction vector, which is restricted based on the placement of red points. Perhaps the infinite prism algorithm could iterate over the set of all points and determine blue polygons in $\mathbb{R}^3$ whose edges do not intersect red line segments. Another approach may be to use the vectors forbidden by red points to determine red-blue segment intersections between monochromatic polygons.

References

[1] F. M. Dong, K. L. Teo. Chromatic polynomials and chromacity of graphs. *World Scientific Publishing Company* 2005.

[2] F. Hurtado, C. Seara, S. Sethia. Red-Blue Separaility Problems in 3D. *In Proc. 3rd Int. Conf. Comput. Sci. and Its Appl* 2003; 766–775.

DIMATIA/MCW Program Report

Elizabeth Field
Southern Connecticut State University

1 Talks

The first week that we were here, there were talks given by some professors from Charles University. Two of those talks were especially interesting to me. The first was one given by Jiří Fiala on interval graphs. An interval graph is the graph which represents the intersection of intervals along a real line and a given graph is an interval graph if and only if it has an interval model. Fiala explained to us that the model of an interval graph can be simplified because the usual model of intersecting invervals contains unneeded information. For instance, an interval model involving n intervals can be represented as a sequence of at most $n-1$ cliques in which each vertex appears in a consecutive subsequence of cliques. Fiala also informed us of the various practical applications of interval graphs, including scheduling problems wherein the maximum independent set represents the maximum number of tasks which can be done without conflict and the colouring of a graph minimizes the number of machines needed to complete a task.

What really intrigued me about Fiala's talk was when he discussed the problem of finding Hamiltonian paths, cycles, and k-staves on interval graphs. While not every interval graph has a Hamiltonian path, $K_{1,3}$ for instance is an interval graph with no Hamilton path, Fiala explained that if an interval graph has a Hamiltonian path, then it must also have a monotone Hamiltonian path beginning at vertex v_1 and ending at v_n. This claim is a consequence of a lemma by Peter Damaschke which gives a means of constructing a monotone path from two vertex disjoint paths. Fiala then went through the algorithm that Damaschke presents in [1] to find a spanning k-stave in an interval graph. After going through the proof of the algorithm and discussing the relationship between scattering number and the existence of spanning k-staves, Fiala left us with two problems to think about. The first was the problem of finding a Hamiltonian path in an interval graph with one fixed endpoint and the second was the problem of finding a Hamiltonian path in an interval graph with two fixed endpoints. These problems were very interesting to me and although I was not able to make much progress toward solving them, I learned a lot while trying.

The other topic which interested me was the one presented to us by Andrew Goodall. He gave two talks on counting the proper k-colourings of graphs and acyclic orientations of graphs. I was familiar with the definition of a proper k-colouring of a graph, G, as a function, $f : V(G) \to [k]$, such that if $uv \in E(G)$, $f(u) \neq f(v)$. However, I had never considered counting the number of such colourings and was quite intrigued when Goodall introduced the following definition:

Definition 1.1. The chromatic polynomial of a graph, G, counts the number of proper k-colourings of the vertices of G and is denoted by $P(G; k)$.

Therefore, $P(G; k) > 0$ if and only if G has a proper k-colouring. Goodall then went on to explain that a proper colouring of G defines a unique acyclic orientation of G whereby $u \to v$ if $f(u) < f(v)$ because f is a total order. Therefore, the longest directed path that can exist in this orientation is a P_k. This motivated us to prove the following theorem about the converse of this statement:

Theorem 1.2. *If G has an acyclic orientation in which all dipaths have length less than k, $P(G; k) > 0$.*

To prove this statement, we needed that every acyclic orientation has at least one source and one sink. This follows from the fact that if P_r with $r \leq k$ is the longest dipath in G, then the first vertex in this path can only have edges leaving from it and the last vertex in this path can only have edges entering. The graph G can now be shown to have a proper $r \leq k$-colouring as follows. Colour all sinks in G with the colour r (since no edges can exist between sinks) and then delete these vertices. The resultant subgraph still has an acyclic orientation (if it did not, this would imply G did not have an acyclic orientation) with maximum dipath P_{r-1} and so we can colour all sinks in this subgraph with $r - 1$. Continuing on in this fashion, it is clear that G can be given a proper r-colouring and, since $r \leq k$, it follows that $P(G; k) > 0$.

Goodall then talked about the chromatic polynomial in more depth. We were able to deduce that $P(\bar{K_n}; k) = k^n$ and that $P(K_n; k) = k^{\underline{n}}$, where $k^{\underline{n}}$ denotes the falling factorial $k(k-1)\cdots(k-n+1)$ by greedily colouring the vertices. After some explanation, we arrived at the following definition of the chromatic polynomial:

Definition 1.3. $P(G; k) = \sum_{r=1}^{k} a_r(G) k^{\underline{r}}$, where $a_r(G)$ is the number of ways to partition the vertices of G into r non-empty stable sets.

This definition makes sense because for each partition of the vertices into r non-empty stable sets, there are $k^{\underline{r}}$ ways to assign colours, since vertices in each set can receive the same colour.

We then established the following recurrence relation for the chromatic polynomial of a graph, G with some non-edge, $xy \notin E(G)$:

$$P(G;k) = P(G + xy;k) + P(G \cdot xy;k).$$

This recurrence is true because the number of colourings of G is the number of colourings of G where $f(x) \neq f(y)$ plus the number of colourings of G where $f(x) = f(y)$. The first is the number of colourings of the graph G' where $xy \in E(G)$ and the second is the number of colourings of the graph G'' where x and y are identified as a single vertex. From here, we were able to arrive at the following recurrence by setting $H = G + xy$, $G = H \backslash xy$, and $G \cdot xy = H/xy$, where $H \backslash xy$ denotes the deletion of the edge xy and H/xy denotes the contraction of the edge xy:

$$P(H;k) = P(H \backslash xy;k) - P(H/xy;k).$$

The first recurrence is useful for dense graphs and the second is useful for sparse graphs.

Goodall then left us with several exercises to attempt, several of which I will present in the following section.

A few days later, Goodall returned to talk to us more about acyclic orientations and the chromatic polynomial. We began by reviewing the chromatic polynomials of cycles and trees on n vertices (proofs in next section). They are as follows :

$$P(C_n;k) = (k-1)^n + (-1)^n(k-1)$$

$$P(T_n;k) = k(k-1)^{n-1}.$$

We then discussed the coefficients of terms in the chromatic polynomial. We will denote the coefficient of the term k^r by $[k^r]P(G;k)$. Goodall explained that the deletion-contraction recurrence could be used to show that for all graphs G on n vertices without loops, $[k^n]P(G;k) = 1$. To find $[k^{n-1}]P(G;k)$, we returned to our definition of the chromatic polynomial as $P(G;k) = \sum_{i=1}^{n} a_i(G)k^{\underline{i}}$. We saw that $a_n(G) = 1$, since there is only one way to partition $V(G)$ into n stable sets (each vertex goes into its own set). We also saw that $a_{n-1}(G) = \binom{n}{2} - |E(G)|$ because the only way partition $V(G)$ into $n-1$ sets is for one set to contain two vertices and the rest to be singletons. There are $\binom{n}{2}$ choices for the set which contains two vertices, however $|E(G)|$ of these will be an edge and therefore will not be stable. From here, we were able to show that

$$\begin{aligned}[k^{n-1}]P(G;k) &= -(\sum_{i=1}^{n-1} i)a_n(G) + a_{n-1}(G) = -\binom{n}{2} + \binom{n}{2} - |E(G)| \\ &= -|E(G)|.\end{aligned}$$

We then began to calculate the coefficient of k^{n-2}, but ran into problems when we could not count the number of partitions of vertices into $n-2$ stable sets. I was very interested in this, however, and so I looked at [2] in which Meredith proves that if a graph G with n vertices and m edges has a girth of $n-s+1$ and p cycles of this length, then for $r > s$, $|[k^r]P(G;k)| = \binom{m}{n-r}$ and $|[k^s]P(G;k)| = \binom{m}{n-s} - p$. This implies that for any graph on at least three vertices, $[k^{n-2}]P(G;k) = \binom{m}{2} - p$, where p is the number of triangles in G. Additionally, for any triangle-free graph on at least four vertices, $[k^{n-3}]P(G;k) = \binom{m}{3} - q$, where q is the number of C_4's in G. Beyond these small values, however, it seems quite difficult to compute the coefficients of terms in the chromatic polynomial.

After discussing these coefficients, we went on to define the following polynomial:

$$Q(G;k) = (-1)^n P(G;-k) = \sum_{i=1}^{n} b_i(G)k^{n-i},$$

where $b_i(G)$ counts the number of subgraphs of G with i edges containing no broken cycles. Goodall pointed out that if the girth of $G = g$, then $b_i(G) = \binom{m}{i}$ for $0 < i < g-1$ and $b_{g-1}(G) = \binom{m}{g-1} - q$, where q is the number of g-cycles. This makes sense because if the smallest cycle in G is a C_g, then any subgraph on up to $g-2$ vertices cannot possibly contain a broken cycle, and so the number of subgraphs of G with $i \leq g-2$ edges without a broken cycle is just the number of subgraphs of G with i edges. Additionally, for each subgraph on $g-1$ vertices, there is only one way to have a broken cycle for each cycle of length g. These two formulas correspond exactly with Meredith's theorem in [2]. The point of this discussion was to bring us back to the topic of acyclic orientation of graphs. Goodall explained that the evaluation of $Q(G;1) = (-1)^n P(G;-1) = \sum b_i(G)$ calculates the number of acyclic orientations of G. He then asked us to think about a combinatorial interpretation for why the number of acyclic orientations of a graph should be the same as the total number of subgraphs with no broken cycles. I thought about this problem for quite some time and this question was the topic of my final presentation. I present one thought below.

In [3] when Stanley proves that $(-1)^n P(G; -1)$ is the number of acyclic orientations of G, he also shows that the number of acyclic orientations of G are the number of equivalence classes for a particular equivalence relation defined as follows:

Definition 1.4 (Stanley 1973)**.** Let G be a graph on n vertices and let σ be a bijective mapping $\sigma : V(G) \to [n]$. Then if $\sim$ is an equivalence relation on the set of all σ labelings of G with the condition that $\sigma \sim \sigma'$ if whenever $uv \in E(G)$, then $\sigma(u) < \sigma(v) \Leftrightarrow \sigma'(u) < \sigma'(v)$, $(-1)^n P(G; -1)$ is the number of equivalence classes of this relation.

This made me wonder whether there may be some sort of relation that could be defined on G for which the subgraphs with no broken cycles determine the equivalence classes for the relation. I have not yet been successful in finding one, however, and I have not yet been able to come up with a combinatorial interpretation for why the number of acyclic orientations of a graph is the same as the number of subgraphs with no broken cycles.

Finally, Goodall went through a proof for why the function $Q(G; 1) = A(G)$, where $A(G)$ is the number of acyclic orientations of G. This involved using Q to show that the deletion-contraction recurrence held for A. Goodall left us with many problems to work out, several of which I present solutions to in the next section.

2 Exercises

1. Find the chromatic polynomial for a tree on n vertices, T_n.
   The chromatic polynomial for a tree on n vertices is

$$P(T_n; k) = k(k-1)^{n-1}.$$

Proof. When $n = 2$, it is clear that there are k colours available for the first vertex and $k - 1$ colours available for the second vertex, giving a chromatic polynomial of $k(k-1)$. So, assume that up to some m, the chromatic polynomial for a given tree on m vertices is $k(k-1)^{m-1}$. Now consider a tree on $m+1$ vertices. By the deletion-contraction recurrence, we have that

$$\begin{aligned} P(T_{m+1};k) &= P(T_{m+1} \backslash e \in E(T);k) - P(T_{m+1}/e \in E(T);k) \\ &= P(T_m + K_1;k) - P(T_m;k) \\ &= k(P(T_m;k)) - P(T_m;k) \\ &= P(T_m;k)(k-1) \\ &= k(k-1)^m. \end{aligned}$$

□

2. Find the chromatic polynomial for the cycle on n vertices, C_n.
The chromatic polynomial for a cycle on n vertices is

$$P(C_n;k) = (k-1)^n + (-1)^n(k-1).$$

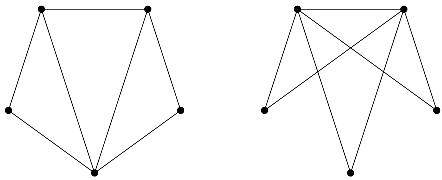
Proof. When $n = 3$, it is clear that there are k colours available for the first vertex, $k-1$ colours available for the second vertex, and $k-2$ colours available for the third vertex. Therefore, $P(C_3;k) = k(k-1)(k-2) = k^3 - 3k^2 + 2k = (k-1)^3 + (-1)^3(k-1)$. So, assume that up to some m, the chromatic polynomial for the cycle on m vertices is $(k-1)^m + (-1)^m(k-1)$. Now, consider the cycle on $m+1$ vertices. By the deletion-contraction recurrence, we have that

$$\begin{aligned} P(C_{m+1};k) &= P(T_{m+1};k) - P(C_m;k) \\ &= k(k-1)^m - [(k-1)^m + (-1)^m(k-1)] \\ &= k(k-1)^m - (k-1)^m + (-1)^{m+1}(k-1) \\ &= (k-1)^m(k-1) + (-1)^{m+1}(k-1) \\ &= (k-1)^{m+1} + (-1)^{m+1}(k-1). \end{aligned}$$

□

3. Find the chromatic polynomial for the wheel graph on $n+1$ vertices, W_n.

To find the chromatic polynomial for the wheel graph, we will first start with the graph H which consists of the wheel graph W_n with an extra vertex attached to some 3-clique in W_n. Note that once the vertices of W_n are coloured, there are $k-3$ colours left for the additional vertex. Therefore, $P(H;k) = (k-3)P(W_n;k)$. We now apply the deletion-contraction recurrence to H where the edge used is the edge, e, adjacent to



the two degree four vertices in H. Note that $P(H\backslash e;k) = P(W_{n+1};k)$ and $P(H/e;k) = (k-2)P(W_{n-1};k)$, since H/e is the graph which consists of W_{n-1} with an additional vertex adjacent to a degree three vertex and the degree $n-1$ vertex. This is sufficient to establish the following recursive definition:

$$P(W_n;k) = (k-3)P(W_{n-1};k) + (k-2)P(W_{n-2}).$$

This has the characteristic polynomial $r^2 - (k-3)r - (k-2) = 0$ with roots $k-2$ and -1. Therefore, the chromatic polynomial for W_n must be of the form $P(W_n;k) = A(k-2)^n + B(-1)^n$. Using as initial conditions $P(W_3;k) = k(k-1)(k-2)(k-3)$ and $P(W_4;k) = k^5-8k^4+24k^3-31k^2+14k$ (calculated using the deletion-contraction relation described above), we get that

$$P(W_n;k) = k(k-2)^n + k(k-2)(-1)^n.$$

4. Find two non-isomorphic graphs G and G' such that $P(G;k) = P(G';k)$.

Each of these graphs has a chromatic polynomial of $k(k-1)(k-2)^3$. This can be seen by using a greedy colouring. In the first graph, there are k choices for how to colour the degree 4 vertex. Then, there are $k-1$ choices for the next vertex, and $k-2$ colours for each subsequent vertex because vertices can be selected so that they are adjacent to two adjacent and already coloured vertices. In the second graph, there are k choices for how to colour one degree 4 vertex and then $k-1$ choices for the second degree 4 vertex. Since none of the remaining vertices are adjacent to one another but are all adjacent to the degree 4 vertices, there are $k-2$ choices for how to color each of them.

This particular example can be extended into two infinite families whereby for each $n/geq5$, the graphs on n vertices are not isomorphic but have the same chromatic polynomial. One family consists of a cycle on n vertices which is triangulated in such a way that one vertex is of degree $n-1$. The other family is a connected graph on n vertices which has $n-2$ triangles such that each triangle shares one common edge, creating two degree $n-1$ vertices. Both families have chromatic polynomials of $k(k-1)(k-2)^{n-2}$ as can be easily seen by extending the greedy colourings described above.

5. By considering the definition of the chromatic polynomial, prove that

$$k^n = \sum_{1 \leq i \leq n} S(n,i) k^{\underline{i}},$$

where $S(n,i)$ is equal to the number of partitions of an n-set into i non-empty sets.

Proof. We know that $P(\bar{K_n}; k) = k^n = \sum_{1 \leq i \leq n} a_i(G) k^{\underline{i}}$, where $a_i(G)$ is the number of partitions of the vertices of G into i non-empty stable sets. Since any set of vertices of $\bar{K_n}$ forms a stable set, $a_i(\bar{K_n}) = S(n,i)$. Therefore, $k^n = \sum_{1 \leq i \leq n} S(n,i) k^{\underline{i}}$. □

6. If G is a connected graph, what does the coefficient of k in the chromatic polynomial $P(G;k)$ count?

Expanding the terms of $P(G;k) = \sum_{i=1}^{n} a_i(G) k^{\underline{i}}$, it can be seen that $[k]P(G;k) = \sum_{i=1}^{n} (-1)^{i+1}(i-1)! a_i(G)$, where $a_i(G)$ denotes the number of partitions of the vertex set of G into i non-empty stable sets. However, this gives no information about the meaning of the coefficient of the chromatic polynomial. So, we seek other interpretations.

Since $(k-1)$ is a factor of $P(G;k)$ whenever G has an edge (this can be shown by the deletion-contraction relation), $P(G;1) = 0$. Therefore, the sum of the coefficients of the chromatic polynomial must equal 0, and so $[k]P(G;k) = -\sum_{i=2}^{n} [k^i]P(G;k)$. Again, this definition is unsatisfying as it sheds no light on how to interpret this coefficient.

However, using the definition of the chromatic polynomial presented by Whitney in [4] of $P(G;k) = \sum_i (-1)^i b_i k^{n-i}$, we can clearly see that when we take $i = n-1$, $[k]P(G;k) = (-1)^{n-1} b_{n-1}$, where b_{n-1} is the number of subgraphs of G with $n-1$ edges which contain no broken cycles.

Interestingly enough, there is another equivalent definition of the chromatic polynomial which gives an alternative definition of the coefficient of k. Goodall and Nešetřil leave as a problem to show that for a connected graph G, $P(G;k) = \sum_{i=0}^{n-1} (-1)^i c_i(G) k^{n-i}$, where $c_i(G)$ is the number of cocliques of order $n-i$ occurring as leaf nodes in the computation tree for G. This interpretation of the chromatic polynomial implies that $[k]P(G;k) = (-1)^{n-1} c_{n-1}(G)$. In absolute value, this is the number of cocliques of order 1 which occur as leaf nodes in the computation tree for G.

These two different interpretations of the coefficient of k in the chromatic polynomial make me wonder whether there is some connection between the number of cocliques of order 1 which occur as leaf nodes in the computation tree for G and the number of subgraphs of G on $n-1$ edges which contain no broken cycles. It seems that there should be some combinatorial interpretation for why these two values are equivalent. Additionally, it must also be true that the number of acyclic orientations of G is the number of cocliques of any order occuring as leaf nodes in the computation tree of G. Perhaps this may give another combinatorial explanation for the number of acyclic orientations of G.

References

[1] PETER DAMASCHKE. Paths in interval graphs and circular arc graphs, *Discrete Math.*, 112(1-3):49-64, 1993.

[2] G. H. J. MEREDITH. Coefficients of chromatic polynomials, *J. Combinatorial Theory Ser. B*, 13:14-17, 1972.

[3] RICHARD P. STANLEY. Acyclic orientations of graphs, *Discrete Math.*, 5:171-178, 1973.

[4] HASSLER WHITNEY. A logical expansion in mathematics, *Bull. Amer. Math. Soc.*, 38(8):572-579, 1932.

DIMATIA/MCW Program Report

Kevin Sung
Rutgers University

1 Experience

I feel very lucky to have been chosen to come to Prague. The experience was a nice mix of work and play. The first week consisted of lectures from professors at Charles University. Martin Balko presented some proofs of Cayley's formula, Professor Goodall talked about the chromatic polynomial, and Professor Fiala talked about interval graphs and the problem of finding hamiltonian paths on them.

The second week I attended the Midsummer Combinatorial Workshop. One of the talks that stood out to me was Iersal's talk on reconstructing phylogenetic networks. I have an interest in evolutionary biology, and it was enlightening to learn about how people actually figure out evolutionary relationships from DNA using graph theory analysis. Most, if not all of the speakers at the workshop presented open problems, which I really had no luck in figuring out.

On the last day of the workshop, Professor Nešetřil showed us around the National Gallery. I was impressed with his knowledge of art history and I really enjoyed seeing and learning about the art in the museum.

I am grateful to my hosts for showing me around Prague and other parts of the Czech Republic. It is really refreshing to be in a new place and experience a different culture. Thank you!

2 Math

The very first lecture we received consisted of three proofs of Cayley's formula, presented by Martin Balko. Cayley's formula states that for $n \geq 2$, the number of spanning trees of K_n is equal to n^{n-2}. I was amazed that the number could be stated so simply, and I was impressed at the cleverness of the proofs. I found the first proof presented, using vertebrates, particularly elegant. However, I did not completely understand all of it, and Mr. Balko did not quite complete the proof. I looked for the proof in the textbook provided to us by Dr. Nešetřil, and it turned out that he, too, left parts of the proof as an exercise to the reader. I will present below a complete proof.

Theorem 2.1. *For each $n \geq 2$, the number of spanning trees of K_n is equal to n^{n-2}.*

3 Proof

Consider a spanning tree of K_n. Mark one vertex with a circle, and one with a square. We allow that the same vertex may be marked twice. We call such an arrangement a vertebrate. Let B be the set of all vertebrates that can arise from K_n. For a given spanning tree, there are n^2 possible vertebrates. Therefore, the number of spanning trees is equal to B/n^2.

Lemma 3.1. *There exists a bijection F between the set B of all vertebrates for a given n and the set of all mappings of the set $\{1, 2, \ldots, n\}$ to itself.*

Since the number of mappings of the set $\{1, 2, , n\}$ to itself is n^n, it follows from the lemma that the number of vertebrates is also n^n, so the number of spanning trees is n^{n-2}.

Proof. Take a tree with n vertices, and order the vertices from 1 to n (the order doesn't matter). Now mark one with a circle and one with a square to create a vertebrate. Since the graph is a tree, there is a unique path from the circled vertex to the squared vertex; let's call this path the spine. The order of the vertices on the spine from circle to square defines a permutation of the numbers assigned to them. Draw the cycles of the permutation. This graph is a disjoint union of directed cycles. Call this graph P. Now, in the original graph, remove the edges of the spine. The remaining graph is a forest. In each tree of the forest, orient each edge to point towards the vertex that was originally in the spine. Now, put this into the graph P: take P, and to each vertex in P attach the tree that was remaining from the vertebrate when the edges of the spine were removed, and orient these edges towards the vertex. Call this graph G. We finally have that G defines this mapping of $\{1, 2, \ldots, n\}$ to itself. It is a mapping because each vertex has exactly one edge coming out of it. This is clearly true for the original permutation graph, and it is true for the remaining vertices because there is a unique path from each vertex to the spine.

Now, what remains to show is that we can reconstruct the vertebrate from the mapping, and that any such mapping can be derived from some vertebrate. Constructing the vertebrate is easy. Given the mapping derived above, we can construct the graph G. G contains some directed cycles. The vertices in the directed cycles are the vertices that are on the spine on the vertebrate, and the directed cycles define a permutation of them. The ordering of the permutation corresponds to the ordering on the spine, and the rest of the graph is just attached to these vertices as they are in the graph G. To prove the other part, we need a definition and a lemma. $\square$

Definition 3.2. Given a mapping f from a vertex set V to itself, the *directed graph of f* is the graph with vertex set V and edges from each v to $f(v)$.

Lemma 3.3. *The directed graph of a mapping from V to V is always the disjoint union of directed cycles, with each cycle possibly having trees hanging off it with edges directed towards the cycle.*

We can use this lemma to finish the proof of the above lemma. Given a mapping from a vertex set V to itself, we can construct the directed graph. Then, applying the same procedure as above, we can construct a vertebrate.

Proof. Draw each isolated vertex V. Now on each vertex put a loop. This graph is a disjoint union of directed cycles, with trees possibly hanging off the cycle. Let us construct the directed graph of the mapping by visiting each vertex, removing the originally drawn loop and replacing it with the actual edge. I will show that at each step, the graph retains the property stated in the lemma, so that when we have visited every vertex, the final directed graph satisfies the property. Suppose that at some point in drawing the edges, the current graph is a disjoint union of directed cycles with trees possibly hanging off of the cycles with edges pointed towards the cycles. We erase the loop on the current vertex v. This vertex v is no longer part of the cycle. Now we draw the edge from v to $f(v)$. Case 1: $f(v) = v$. In this case, the graph does not change and still satisfies the property. Case 2: $f(v)$ lies on the same component as v. In this case, we have completed a cycle, and the graph still satisfies the property. Case 3: $f(v)$ lies on a different component. In this case, the component that v is in has become a tree hanging off of another component, so the graph still satisfies the property. Thus, when we have drawn all the edges, we end up with a graph that satisfies the property stated in the lemma. $\square$

DIMATIA/MCW Program Report

Kevin Wong
Rutgers University

1 Introduction

In the week leading up to the Midsummer Combinatorial Workshop, we were graciously escorted by the Czech students to a number of sights and activities both inside and outside of Prague. We were also fortunate enough to receive visits by a number of professors who gave us small lectures on relevant areas of graph theory. At the end of each talk, they presented to us a number of problems to work on.

2 Math

The talk that interested me the most was the one on interval graphs by Jiří Fiala as a part of the REU seminar series. He started by defining and showing us an example of an interval model: simply put it is a set of intervals in the real line. Based on this interval model, we can determine the intersection graph there is one vertex for each interval, and there is an edge between two vertices if their corresponding intervals intersect. This graph is called the interval graph.

Note that multiple interval models can have the same interval graph, and also that not all graphs are interval graphs. For example, the cycle graph with 4 vertices (C_4) is not a valid interval graph because it is impossible to construct a corresponding interval model. Professor Fiala then stated two criteria that are both necessary and sufficient for a graph to be interval it must be chordal and asteroidal triple-free. A graph is chordal if each cycle of length 4 or more has a chord (an edge joining two non-adjacent vertices), and an asteroidal triple is a set of 3 vertices with the property that each pair can be joined by a path that avoids the neighborhood of the third.

He continued to speak about methods of finding hamiltonian paths and k-staves in interval graphs. Because each interval model corresponds to exactly one graph, we worked with interval models rather than the graphs themselves. We found that it was visually easy to determine the existence of a hamiltonian path using this method. He concluded the talk by presenting us with a murder mystery(The Duke of Densmore) by Claude Berge whose design was rooted in interval graphs.

3 Problems

3.1 Who Killed the Duke of Densmore?

"The Duke of Densmore is found dead in the explosion of his castle. The murder was committed with a bomb placed carefully in the labyrinth, which would require a long preparation in hiding in the mazes of the labyrinth. During his last years, the Duke had received eight visitors to his castle; each of them was brought first to the island and then back to the mainland by a motor boat. None of them recalls the precise dates or duration of her stay on the island, but each remembers with certainty whom else she had met on the isle. Determine who killed the Duke of Densmore."

Solution: In addition to the description above, we are given a list of female suspects, and, for each suspect, we are also given a list of people she had met on the isle. We can then construct a graph where each vertex corresponds to a suspect, and an edge is drawn between two vertices if their corresponding suspects saw each other on the isle (there are no such conflicts where A met B, but B did not meet A). This graph is shown in Fig 1.

Note that the graph is neither chordal nor asteroidal triple-free: vertices A, B, H, C form a cycle on 4 vertices without a chord, as do vertices A, C, H, G. Also, vertices B, D, F form an asteroidal triple. With these points in mind, we can see that vertex A is the cause of this trouble. If we remove vertex A and all its edges from the graph, we are left with a valid interval graph (Fig 2). This implies that A's interval was actually a disjoint union of two intervals, which points to the following story: person A met some people in the beginning, then went into hiding as she prepared the bomb, then came out and met a few others later on. I propose a potential interval model for the timeline in Fig 3., showing the time periods of A's disappearance. Thus we conclude that person A (Ann in the story) is the culprit.

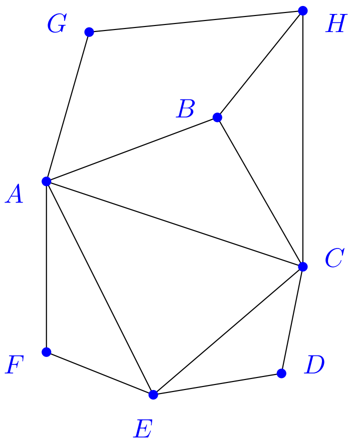


Figure 1: Original intersection graph, as described by the problem

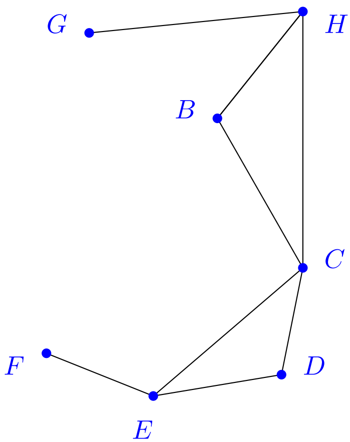


Figure 2: Intersection graph with vertex A removed

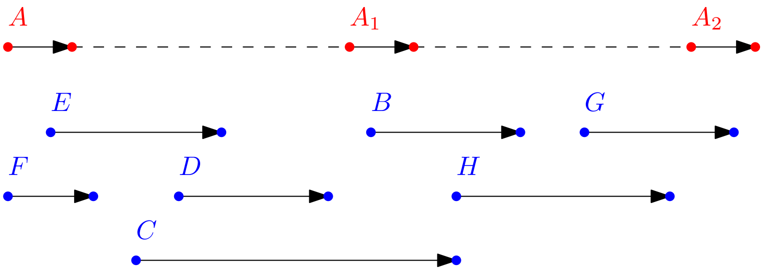


Figure 3: One possible interval model

3.2 Rational Roots of $P(G;k)$

$P(G;k)$ is a polynomial in k. Prove that all rational roots of $P(G;k)$ are non-negative integers when G has at least one edge.

Solution: First, we show that all rational roots of $P(G;k)$ are integral.

Note that $P(G;k)$ is a monic polynomial. This can be proven by starting with a single-edge graph its monic chromatic polynomial is $k_{n-1}(k-1)$, as $n-1$ vertices can have any of the k colors, and the last one only has $k-1$ choices. Then, by using the deletion-contraction recursion and an inductive argument, we can prove that this result holds for all possible G.

We can write $P(G;k)$ as $k^n + a_{n-1}k^{n-1} + \cdots + a_m k^m$, where n is the number of vertices in G, a_i are all integers, and $0 \le m < n$ (we cannot have $m = n$ since G has at least one edge). We can factor k_m from the polynomial: $P(G;k) = k^m(k^{n-m} + a_{n-1}k^{n-m-1} + \cdots + a_m)$. We can then use the rational roots theorem on the inner polynomial of degree $n-m$: it states that all rational roots must be of the form $\frac{p}{q}$ where p is a divisor of a_m, and q is a divisor of $a_n = 1$. Since the only choice for q is 1, we can conclude that all rational roots are integers.

Next, we show that $P(G;k)$ has no negative roots. To do this, we will show that either $P(G;-k) > 0$ or $P(G;-k) < 0$ for $k > 0$.

Consider the formula proven by Stanley in 1972:
$(-1)^n P(G;-k) = x$, where $x =$ the number of ordered pairs (α, f) where α is an acyclic orientation, f is a mapping from $V(G)$ to $1, 2, \ldots, k$ such that if $u \to v$ in the orientation α, then $f(u) \le f(v)$.

If G is a graph with at least one edge, the $x > 0$ (the justification of this point eludes me). Thus $x = (-1)^n P(G; -k) > 0$. Because $n = |V(G)|$,it is a constant with regards to k. Thus, we can divide both sides by $(-1)^n$, and we have shown that either $P(G; -k) > 0$or$P(G; -k) < 0$ for all $k > 0$. Thus it cannot have a negative root.

We have shown that all rational roots of $P(G; k)$ are integral, and that they cannot be negative. Thus we are done.