Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 65
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 65
Espoo 2000 HUT-TCS-A65

COMPUTATIONAL METHODS FOR RAMSEY NUMBERS

Harri Haanpää

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 65
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 65
Espoo 2000 HUT-TCS-A65

COMPUTATIONAL METHODS FOR RAMSEY NUMBERS

Harri Haanpää

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory for Theoretical Computer Science

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietojenkäsittelyteorian laboratorio

Distribution:
Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O.Box 5400
FIN-02015 HUT
Tel. +358-0-451 1
Fax. +358-0-451 3369
E-mail: lab@tcs.hut.fi

ISBN 951-22-5246-5
ISSN 1457-7615

Picaset Oy
Helsinki 2000

Abstract: The Ramsey number $R(k, l)$ is the least integer n such that all graphs on n or more vertices contain a clique of k vertices or an independent set of l vertices as an induced subgraph. In this work we investigate computational methods for finding lower bounds for Ramsey numbers. Some constructions of lower bounds for multicolor Ramsey numbers, a generalization of Ramsey numbers, are also considered.

Several methods that have been used for finding lower bounds for Ramsey numbers are surveyed. Specifically, constructions which correspond to the structure of finite fields are examined, using local search methods is discussed, and using symmetrical graph colorings are investigated. The main emphasis in this work is on using local search methods for finding lower bounds for Ramsey numbers. By a construction found by using tabu search, we show that $R(5, 9)$ is greater than 120.

Evaluating Ramsey numbers is very difficult due to the combinatorial nature of the problem, and exact values are only known for the smallest values of the parameters. Some recent results concerning the computational complexity of related problems are summarized.

Keywords: Ramsey numbers, graph theory, tabu search

CONTENTS

1 Introduction **1**
1.1 Ramsey Theory . . . 1
1.2 Party Mathematics . . . 2
1.3 Definitions . . . 5
1.4 Fundamental Results . . . 5
1.4.1 Definition and Existence of Ramsey Numbers . . . 5
1.4.2 Recursive Bounds for Ramsey Numbers . . . 7
1.4.3 General Bounds for Ramsey Numbers . . . 8
1.5 Computational Complexity . . . 9
1.5.1 Computational Complexity Classes . . . 9
1.5.2 On The Complexity of Ramsey-Theoretical Problems 13

2 Optimization and Search **15**
2.1 Searching as an Optimization Problem . . . 15
2.2 Local Search Methods . . . 17
2.2.1 Steepest Descent . . . 18
2.2.2 Simulated Annealing . . . 19
2.2.3 Tabu Search . . . 19

3 Groups, Permutations and Fields **22**
3.1 Groups . . . 22
3.2 Permutations . . . 23
3.3 Permutation Groups . . . 24
3.4 Finite Fields . . . 28

4 Schur Numbers **30**
4.1 Introduction to Schur Numbers . . . 30
4.2 The Known Schur Numbers . . . 30
4.3 Lower Bounds for Ramsey Numbers . . . 31
4.4 Lower Bounds for Schur Numbers . . . 31
4.4.1 Backtracking . . . 31
4.4.2 Local Search . . . 32
4.4.3 Incremental Construction . . . 33

5 Lower Bounds for Ramsey Numbers **34**
5.1 Lower Bounds Using Finite Fields . . . 34
5.2 Attempts to Improve $R_k(3)$ Using Finite Fields . . . 35
5.3 Colorings as Unions of Orbits . . . 36
5.4 Attempts to Improve $R_k(3)$ Using Symmetric Colorings . . . 38
5.5 Local Search for Two-Color Ramsey Numbers . . . 40
5.5.1 Exoo on $R(3,10)$, $R(3,12)$ and $R(5,5)$ . . . 40
5.5.2 Piwakowski with Tabu Search . . . 41
5.5.3 Fifteen Lower Bounds by Exoo . . . 41
5.6 A New Lower Bound for $R(5,9)$ . . . 42
5.6.1 The Search Method . . . 42
5.6.2 The Construction . . . 43

6 Conclusions and Suggestions for Further Research **45**

1 INTRODUCTION

The aim of this chapter is to give the reader an overview of Ramsey numbers. Sections 1.1 and 1.2 introduce the subject. Some necessary graph-theoretical definitions are given in Section 1.3. In Section 1.4 some fundamental and illustrative results regarding Ramsey numbers are given. In Section 1.5 some results regarding the computational complexity of determining Ramsey numbers are briefly reviewed.

1.1 RAMSEY THEORY

Ramsey theory shows us that in any combinatorial structure, no matter how random it may seem, there are substructures that are more structured than the original combinatorial structure. The following three examples are the simplest cases of the corresponding theorems.

Ramsey: In every group of 6 people, there are *three* people who know each other, or *three* people who do not know each other.

Van der Waerden: If the integers $1, \dots, 325$ are colored with *two* colors, there will be an arithmetic progression of length *three* (that is, some integers $a, a+d, a+2d$) in one color.

Schur: If the integers $1, \dots, 14$ are colored with *three* colors, then there will be some integers a and b such that a, b, and $a-b$ are all of the same color.

In the Ramsey case, the large group of people who may know or not know each other always contains a subgroup where either everybody knows everybody or nobody knows anybody.

In the Van der Waerden case, the integers $1, \dots, 325$ form an arithmetic progression of length 325. When the long arithmetic progression is colored with two colors, it will necessarily contain a shorter one-colored arithmetic progression.

The Schur case is slightly different; it states that it is impossible to color the integers $1, \dots, 14$ in a manner so unorderly that there would be no two numbers and their difference in the same color.

These simple examples alone are not very interesting. What is more interesting is that the results are general. In each of the three statements above, if one replaces any occurrences of the words *two* and *three* with any arbitrary integers, one can retain the validity of the statement by replacing the 6, 325, or 14 with a sufficiently large finite integer, for example:

Ramsey: In every group of 25 people, there are *four* people who know each other, or *five* people who do not know each other.

Van der Waerden: If the integers $1, \dots, 4.23 \cdot 10^{14616}$ are colored with *three* colors, there will be an arithmetic progression of length *three* (that is, some integers $a, a+d, a+2d$) in one color.

Schur: If the integers $1, \ldots, 45$ are colored with *four* colors, then there will be some integers a and b such that a, b, and $a - b$ are all of the same color.

The numbers mentioned for these special cases of Ramsey's theorem are taken from [36]. The numbers for the examples of van der Waerden's theorem are taken from [25], an extensive survey on Ramsey theory. The numbers for the Schur problem were given in Schur's original paper [43] and in an unpublished work of Baumert, respectively.

Since for any given values of the parameters there is always some integer for which the statement holds, and obviously the statement is then true for all larger integers as well, the natural follow-up question is that of determining the smallest integer for which the statement is true. The smallest such integer is then called the Ramsey number or the van der Waerden number, respectively. For Schur numbers a slightly different definition is usually used: the Schur number is the largest integer for which the statement does not hold.

- The Ramsey number $R(k, l)$ is the smallest number such that in any group of that many people at least k will know each other or at least l will not know each other. (A more formal definition will be given in Section 1.4.)

- The van der Waerden number $W(k, l)$ is the smallest number n such that whenever the integers $1, \ldots, n$ are colored with l colors, there will be a one-colored arithmetic progression of length k.

- The Schur number $s(n)$ is the largest integer k such that the integers $1, \ldots, k$ can be colored with n colors so that there are no integers a and b such that a, b, and $a - b$ would all be of the same color.

In this work, we will consider various methods for obtaining lower bounds for specific Ramsey numbers. Schur numbers are investigated because of their connection to Ramsey numbers. Van der Waerden numbers are not considered in this work. Finally, the new lower bound $R(5, 9) > 120$ will be presented.

1.2 PARTY MATHEMATICS

Recall that the Ramsey number $R(k, l)$ is the smallest integer such that in any group of that many people at least k will know each other or at least l will not know each other. The problem of determining the value of a Ramsey number is also known as the party problem.

Problem 1.2.1 (Party problem) *What is the minimum number of guests that must be invited to a party to ascertain that at least k will know each other, or at least l will not know each other?*

Determining the exact values of Ramsey numbers is rather difficult, and the exact value is only known in a few cases. For most parameter values, only bounds are known.

Example 1.2.2 $R(4,4) = 18$: *it is known that in a group of 18 people there are always four who know each other or four who do not; in a group of 17 it is possible that neither kind of group exists.*

Example 1.2.3 $R(4,5) = 25$: *McKay and Radziszowski [32] expended about a decade of computer time to show that in a group of 25 there are always four who know each other or five who do not. It is also known that in a group of 24 it is possible that neither kind of group exists.*

Example 1.2.4 *The value of* $R(5,5)$ *is unknown. It is only known that* $42 < R(5,5) \leq 49$: *in a group of 49 there are always five who know each other or five who do not, and in a group of 42 neither kind of group necessarily exists.*

Radziszowski has compiled many of the known results in his survey [36]. Some of the results found in it are summarized in Figure 1.1. Known exact values appear as centered entries, lower bounds as top entries and upper bounds as bottom entries. A missing entry indicates that the best currently known bound can be found by applying the basic inequalities given in Section 1.4.2. Only the upper triangle is filled, since $R(k,l) = R(l,k)$.

k \ l	3	4	5	6	7	8	9	10	11
3	6	9	14	18	23	28	36	40 43	46 51
4		18	25	35 41	49 61	55 84	69 115	80 149	96 191
5			43 49	58 87	80 143	95 216	116 316	141 442	153
6				102 165	109 298	122 495	153 780	167 1171	203
7					205 540	1031	1713	2826	
8						282 1870	3583	6090	
9							565 6625	12715	
10								798 23854	

Figure 1.1: Values and bounds for Ramsey numbers $R(k,l)$.

The lower bounds typically arise from explicit constructions, showing how people might know/not know each other so that no large subgroup exists. The upper bounds arise from diverse ways of proving that no matter how people know/do not know each other, a large subgroup must exist. To illustrate, we show that $R(3,3) = 6$. First, we show the lower bound $R(3,3) > 5$ and then the upper bound $R(3,3) \leq 6$. These results are well known; they were presented in [26].

Theorem 1.2.5 $R(3,3) > 5$: *There can be a group of five people, where there are no three people who know each other and no three people, who do not know each other.*

Proof: With the dots representing the people, solid lines representing knowing each other and dashed lines representing not knowing each other, Figure 1.2 represents such a group. Clearly, no three dots are connected to each other by lines of the same type. □

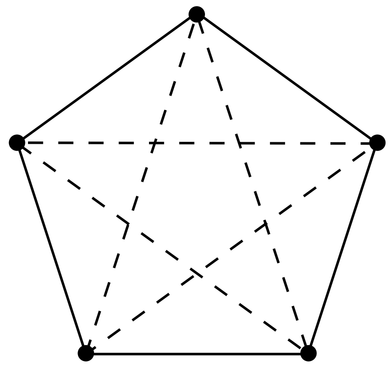

Figure 1.2: $R(3,3) > 5$.

Virtually all known lower bounds have been obtained in essentially the same way, that is, by presenting a construction and verifying that the largest cliques with each type of line are not too large.

Theorem 1.2.6 $R(3,3) \leq 6$: *In every group of six people, there are three people who know each other, or three people who do not know each other.*

Proof: Consider any of the people. There are five other people, each of which the first person either knows or doesn't know. Therefore, there have to be three people the first person knows or three people he doesn't know. Let us assume that there are three people the first person knows. (See Figure 1.3, on the left.) If any two of these three people would know each other, they and the first person would form a group of three who all know each other. Therefore, none of them can know each other. (See Figure 1.3, on the right.) Now, however, they form a group of three who do not know each other. As the case where there are three people the first person doesn't know is completely analogous, the theorem is proven. □

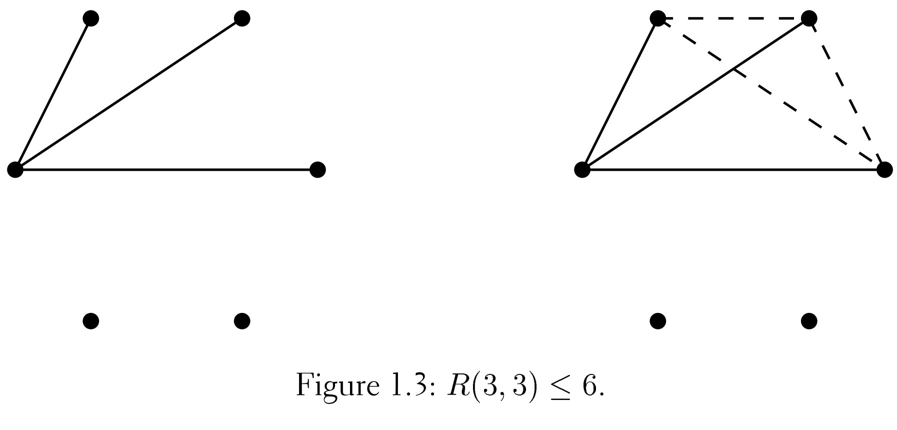

Figure 1.3: $R(3,3) \leq 6$.

1.3 DEFINITIONS

Definition 1.3.1 *A graph is the ordered pair (V, E), where V is a set of vertices, and E is a set of edges, each of which is a 2-element subset of V.*

Definition 1.3.2 *A subgraph $G' = (V', E')$ of a graph (V, E) is a graph for which $V' \subseteq V$ and $E' \subseteq E$.*

Definition 1.3.3 *A complete graph is a graph, where the set of edges E consists of all 2-element subsets of V. A complete graph on n vertices is denoted by $\mathbf{K}_n$. A few small complete graphs are shown in Figure 1.4.*

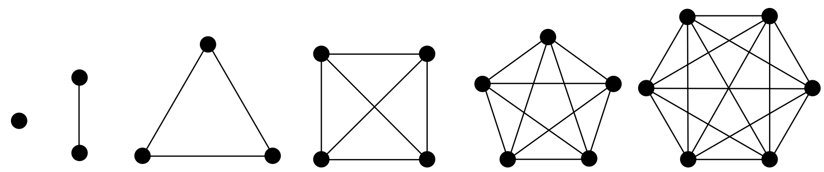

Figure 1.4: The complete graphs $\mathbf{K}_1, \ldots, \mathbf{K}_6$.

Definition 1.3.4 *A clique is a complete graph that is a subgraph of another graph.*

Definition 1.3.5 *An edge-coloring of a graph $G = (V, E)$ is a mapping from the edge set E to the set of colors C. If n colors are used, the coloring is an n-color edge-coloring.*

Definition 1.3.6 *A (k, l)-coloring is a 2-color edge-coloring of a complete graph that contains as a subgraph neither a $\mathbf{K}_k$ in the first color nor a $\mathbf{K}_l$ in the second color.*

1.4 FUNDAMENTAL RESULTS

In this section some fundamental results concerning Ramsey numbers are presented.

1.4.1 Definition and Existence of Ramsey Numbers

In his celebrated paper [38] Ramsey proved the following theorem, quoted verbatim. The subscripts in Γ_m and Δ_n denote the number of elements, and r-combinations simply mean r-element subsets.

Theorem 1.4.1 (Ramsey's theorem) *Given any r, n, and μ we can find an m_0 such that, if $m \geq m_0$ and the r-combinations of any Γ_m are divided in any manner into μ mutually exclusive classes C_i $(i = 1, 2, \ldots, \mu)$, then Γ_m must contain a subclass Δ_n such that all the r-combinations of members of Δ_n belong to the same C_i.*

An interpretation of Ramsey's result is that there is always a structured subsystem in any random system. When one creates a random system by partitioning all the r-element subsets of a set Γ into μ partitions in any manner, there will be a structured subset $\Delta \subseteq \Gamma$, all of whose r-combinations will be in the same partition. Ramsey's theorem tells us that regardless of how the r-element subsets of Γ are partitioned into classes, we can get a structured subset Δ of any cardinality we want by taking a large enough set Γ.

The proof is rather lengthy and is therefore not given here. Ramsey gave a construction for m_0, but his bounds were very large – Ramsey did not attempt to give a good bound, his goal was to prove that there exists a finite m_0 that satisfies the theorem. The question then arose, what the minimal m_0 for given r, μ, and n is. These minimal integers became known as Ramsey numbers and they are denoted by $R_\mu(n; r)$.

As a generalization, instead of one parameter n, a total of μ parameters $n_1, n_2, \ldots, n_\mu$ may be given. In that case, a separate n_i is given for each color i instead of one common n for all colors. This leads us to the following definition, the subscripts of Γ and Δ again denoting the cardinality of the set:

Definition 1.4.2 (Ramsey number) *Given any r, μ, and $n_1, n_2, \ldots, n_\mu$, the Ramsey number $R(n_1, n_2, \ldots, n_\mu; r)$ is the smallest integer such that if $m \geq R(n_1, n_2, \ldots, n_\mu; r)$ and the r-element subsets of Γ_m are divided in any manner into μ mutually exclusive classes C_i $(i = 1, 2, \ldots, \mu)$, then, for some i, Γ_m must contain a subclass Δ_{n_i} such that all the r-combinations of members of Δ_{n_i} belong to C_i.*

It is an immediate corollary of Theorem 1.4.1 that such a smallest integer exists in all cases.

In this work only the case $r = 2$ is considered. In that case, r may be dropped from the notation: $R(n_1, n_2, \ldots, n_\mu) = R(n_1, n_2, \ldots, n_\mu; 2)$, or $R_\mu(n) = R_\mu(n; 2)$. When $r = 2$, the problem can be conveniently expressed in graph-theoretical form. The 2-combinations of Γ_m correspond to the edges of the complete graph $\mathbf{K}_m$, and μ-coloring the edges corresponds to dividing the 2-combinations of Γ_m into μ mutually exclusive classes. This gives us the following alternative definition:

Definition 1.4.3 (Ramsey number) *The Ramsey number $R(n_1, n_2, \ldots, n_\mu)$ is the smallest integer m such that for every μ-color edge-coloring of $\mathbf{K}_m$ there is an i such that the edge-coloring contains a $\mathbf{K}_{n_i}$ in color i as a subgraph.*

Using this definition it is convenient to restate the Party problem (Problem 1.2.1) from page 2 in graph-theoretical terms.

Problem 1.4.4 (Party problem) *Given k and l, find the smallest integer $m = R(k, l)$ such that every 2-color edge-coloring of $\mathbf{K}_m$ contains as a subgraph a $\mathbf{K}_k$ in the first color or a $\mathbf{K}_l$ in the second color.*

Here the vertices of the complete graph $\mathbf{K}_m$ correspond to the guests at the party. The edges are colored in two colors to signify that each pair of persons either knows or doesn't know each other. The complete graphs $\mathbf{K}_k$

and $\mathbf{K}_l$ respectively represent a group of k people where everybody knows each other and a group of l people where nobody knows any of the others. In this manner, Theorem 1.2.5 can be rephrased as "There is a 2-color edge-coloring of $\mathbf{K}_5$ that contains no monochromatic $\mathbf{K}_3$ as a subgraph" and Theorem 1.2.6 as "Every 2-color edge-coloring of $\mathbf{K}_6$ contains a monochromatic $\mathbf{K}_3$ as a subgraph." The proofs are of course completely analogous to the proofs given earlier.

1.4.2 Recursive Bounds for Ramsey Numbers

The next three theorems are simple, well-known combinatorial results.

Theorem 1.4.5 *If $R(k,p) \geq s$ and $R(k,q) \geq t$, then $R(k,p+q-1) \geq s+t-1$.*

Proof: Take a two-color edge-coloring of $\mathbf{K}_{s-1}$ that contains neither a $\mathbf{K}_k$ in the first color nor a $\mathbf{K}_p$ in the second color, and a two-color edge-coloring of $\mathbf{K}_{t-1}$ that contains neither a $\mathbf{K}_k$ in the first color nor a $\mathbf{K}_q$ in the second color. Create a two-color edge-coloring of $\mathbf{K}_{s+t-2}$ by connecting each vertex of the $\mathbf{K}_{s-1}$ to each vertex of the $\mathbf{K}_{t-1}$ with an edge of the second color. The resulting coloring contains neither a $\mathbf{K}_k$ in the first color nor a $\mathbf{K}_{p+q-1}$ in the second color, which proves the theorem. □

Theorem 1.4.6 $R(k,l) \leq R(k-1,l) + R(k,l-1)$.

Proof: Consider any two-color edge-coloring of $\mathbf{K}_m$ with $m = R(k-1,l) + R(k,l-1)$. To prove the theorem we'll show that such a coloring must contain a $\mathbf{K}_k$ in the first color or a $\mathbf{K}_l$ in the second color. Choose any vertex v. Note first that if v is connected to a $\mathbf{K}_{k-1}$ of the first color by edges of the first color, then v and that $\mathbf{K}_{k-1}$ form a $\mathbf{K}_k$ in the first color. Similarly, if v is connected to a $\mathbf{K}_{l-1}$ of the second color by edges of the second color, then there is a $\mathbf{K}_l$ in the second color.

With $R(k-1,l) + R(k,l-1)$ vertices in the graph, the vertex v has $R(k-1,l) + R(k,l-1) - 1$ neighbors. By the pigeonhole principle, v is connected to $R(k-1,l)$ vertices by edges of the first color or to $R(k,l-1)$ vertices by edges of the second color. In the former case, by definition of $R(k-1,l)$, the vertex v is connected by edges of the first color to a $\mathbf{K}_{k-1}$ in the first color or to a $\mathbf{K}_l$ in the second color. Similarly, in the latter case, the vertex v is connected by edges of the second color to a $\mathbf{K}_k$ in the first color or to a $\mathbf{K}_{l-1}$ in the second color. In each case there is a $\mathbf{K}_k$ in the first color or a $\mathbf{K}_l$ in the second color; thus the theorem is proven. □

Theorem 1.4.7 *If $R(k-1,l)$ and $R(k,l-1)$ are both even, then $R(k,l) < R(k-1,l) + R(k,l-1)$.*

Proof: Consider any two-color edge-coloring of $\mathbf{K}_m$ with $m = R(k-1,l) + R(k,l-1) - 1$. We'll show that any such coloring must contain a $\mathbf{K}_k$ in the first color or a $\mathbf{K}_l$ in the second color. If every vertex were connected to exactly $R(k-1,l)-1$ other vertices with edges of the first color, the number of edges of the first color in the coloring would be $\frac{1}{2}m\,(R(k-1,l)-1)$. Since neither m nor $R(k-1,l)-1$ is even, the number of edges of the first color would not be integer. Now, there has to be a vertex v that is connected to $R(k-1,l)$ vertices with edges of the first color or to $R(k,l-1)$ vertices by edges of the second color, and similarly to the previous proof there must be a $\mathbf{K}_k$ in the first color or a $\mathbf{K}_l$ in the second color. □

1.4.3 General Bounds for Ramsey Numbers

The next two theorems are interesting, since they prove that both the lower and upper bounds of $R(k,k)$ grow exponentially as k grows.

The following proof was given by Erdős [13]. It is purely existential. The proof is slightly tidier, but weaker than the very similar proof given in [25], where it was proven that $R(k,k) > k2^{k/2}\left(\left(1/e\sqrt{2}\right) + o(1)\right)$.

Theorem 1.4.8 *Let* $k \geq 3$*. Then* $R(k,k) > 2^{k/2}$.

Proof: Let $N \leq 2^{k/2}$. The number of different graphs on N distinguishable vertices is $2^{N(N-1)/2}$. The number of different graphs containing a clique of order k is $2^{N(N-1)/2}/2^{k(k-1)/2}$. Thus the number of graphs on $N \leq 2^{k/2}$ vertices that contain a clique of order k is less than

$$\binom{N}{k}\frac{2^{N(N-1)/2}}{2^{k(k-1)/2}} < \frac{N^k}{k!}\frac{2^{N(N-1)/2}}{2^{k(k-1)/2}} < \frac{2^{N(N-1)/2}}{2}, \tag{1.1}$$

since for $N \leq 2^{k/2}$ and $k \geq 3$, by substituting $2^{k/2}$ for N, one obtains by straightforward calculation that

$$2N^k < k!2^{k(k-1)/2}.$$

From (1.1) it is clear that there is a clique of order k in the first color in strictly less than half of all two-color edge-colorings of the complete graph on N vertices. The same holds for the second color, and it follows that there must exist a two-color edge-coloring of the complete graph on N vertices that contains no monochromatic clique of order k. □

The next theorem was proven by Erdős and Szekeres in [14].

Theorem 1.4.9 $R(k,l) \leq \binom{k+l-2}{k-1}$.

Proof: Assuming this theorem proven for $R(k-1,l)$ and $R(k,l-1)$, Theorem 1.4.6 gives us

$$\begin{aligned} R(k,l) &\leq R(k-1,l) + R(k,l-1) \\ &\leq \binom{k+l-3}{k-2} + \binom{k+l-3}{k-1} \\ &= \binom{k+l-2}{k-1}, \end{aligned}$$

proving the theorem for $R(k,l)$. After noting that $R(k,2) = k = \binom{k+2-2}{k-1}$, and $R(2,l) = l = \binom{2+l-2}{2-1}$, the theorem follows from induction on k and l. □

1.5 COMPUTATIONAL COMPLEXITY

The computational complexity of determining Ramsey numbers is investigated in this section in order to gain insight into the level of difficulty of the problem. We shall not attempt to cover any significant part of computational complexity theory in this work; instead only a sketch of the concepts relevant to this work is given. For a good overview of the subject, see [34].

1.5.1 Computational Complexity Classes

Studying the relative computational complexity of various problems has been a subject of much interest during the past couple of decades.

The Turing machine

To analyze the complexity of a problem, a computational model is needed. The prevalent model of computation used for complexity analysis is the Turing machine.

The Turing machine consists of an unbounded tape, a read/write head, and a table describing state transitions that may occur during the computation. There is a finite set of possible states the Turing machine can be in, and the set includes the starting state and the yes and no states. During a computation step, depending on the state it is in and the character under the read/write head, the Turing machine can move to another state, overwrite the character under the read/write head, and may move the read/write head one step to the left or right. To start the computation, the tape is initialized with the input, the read/write head is placed at the beginning of the input and the machine is set to the starting state. The computation terminates immediately if the machine enters the yes state or the no state.

Despite its simplicity the Turing machine is very useful for complexity analysis, and it is essentially equivalent to many computation models that at first glance appear more powerful or more representative of the real world.

Any Turing machine can be seen as the representation of an algorithm. A Turing machine receives a string of symbols as input. After performing computations on the input, the computation may terminate in the yes state or terminate in the no state. If the computation terminates in the yes state, the Turing machine is said to accept the input, whereas if the computation terminates in the no state, the Turing machine is said to reject the input. It can also happen that the computation never terminates.

Definition 1.5.1 *A string is a sequence of zero or more characters, each character taken from a fixed set.*

Definition 1.5.2 *A language is a subset of all possible strings.*

Definition 1.5.3 *A Turing machine M is said to decide the language L, if the Turing machine accepts every input string $x \in L$ and rejects every input string $x \notin L$.*

Deterministic Polynomial Time
Definition 1.5.4 *A Turing machine M is said to decide language L in polynomial time, if M decides L and there exists a polynomial function p such that for every input string x, the computation of M on x will terminate within $p(|x|)$ computation steps.*

Definition 1.5.5 *The complexity class $\mathbf{P}$ is the set of languages L for which there exists a Turing machine that decides L in polynomial time.*

Non-Deterministic Polynomial Time
There is a generalization of the Turing machine called the non-deterministic Turing machine. Loosely speaking, a non-deterministic Turing machine can be seen as a Turing machine that has the additional ability to make lucky guesses during the computation.

Definition 1.5.6 *A non-deterministic Turing machine is said to accept the input string x, if there exists some series of lucky guesses y that results in the computation terminating in the yes state.*

Definition 1.5.7 *A non-deterministic Turing machine N is said to decide language L in polynomial time, if N decides L and there exists a polynomial function p such that for every input string x and every series of guesses y, the computation of M on x will terminate within $p(|x|)$ computation steps.*

Definition 1.5.8 *The complexity class $\mathbf{NP}$ is the set of languages L for which there exists a non-deterministic Turing machine that decides L in polynomial time.*

Example 1.5.9 *Consider the Clique problem: Given a graph G and an integer k, does G contain a clique of at least k vertices as a subgraph? It is not known whether this problem is in $\mathbf{P}$, but it certainly is in $\mathbf{NP}$: If the input pair (G, k) is in the language formed by graphs with a clique of order at least k, then a non-deterministic Turing machine can guess k vertices of the clique and verify the edges between those k vertices are in the graph, all in polynomial time.*

It is worth noting that for any $x \in L$ where $L \in \mathbf{NP}$, the sequence of guesses y made by the nondeterministic machine can serve as a certificate that $x \in L$. In other words, for any $L \in \mathbf{NP}$ there exists a deterministic machine that, given (x, y), can verify in polynomial time whether y is a certificate for $x \in L$, and for any $x \in L$ there exists such a certificate y. Here y can be at most polynomial in length compared to x. Continuing the previous example: Given a graph G, an integer k, and k vertices of the graph G, checking whether the given k vertices induce a clique of order k in the graph can be done in polynomial time in a straightforward manner.

Definition 1.5.10 *The complexity class* **coNP** *is the set of languages L whose complement $\overline{L} \in$* **NP**.

Alternatively, the class **coNP** could be characterized by changing the definition of acceptance for a non-deterministic machine. For a string to be in a language $L \in$ **NP** it was sufficient to have one computation terminating in the yes state. For a string to be in a language $L \in$ **coNP**, each computation must terminate in the yes state.

Example 1.5.11 *Consider the Clique Complement problem: Given a graph G and an integer k, is it true that G contains no clique of k or more vertices? Here the language is the set of graphs whose largest clique contains at most $k - 1$ vertices; the complement of the language is the set of graphs that contain a clique of k or more vertices, which is the Clique problem, which is in* **NP**. *Hence the Clique Complement problem is in* **coNP**.

As we saw, for yes-instances of problems in **NP** there exists a succinct certificate. Similarly, for no-instances of problems in **coNP** there exists a succinct disqualification. In other words, there exists a deterministic machine that, given (x, y), can verify in polynomial time whether y is a certificate for $x \notin L$, and for any $x \notin L$ there exists such a certificate y. Again, y can be at most polynomial in length compared to x.

Reductions and Completeness

Reduction is a fundamental concept in classifying problems into complexity classes. Basically, if a problem is reducible to another problem, then any input to the first problem can be transformed to an instance of the second problem with relatively little computing resources. Reductions of various strengths have been used in the literature, but one of the most commonly used reductions is the polynomial time reduction.

Definition 1.5.12 *A language L is reducible to another language L', if there is a deterministic Turing machine that, for any input string x, outputs the output string x' such that $x' \in L'$ if and only if $x \in L$. Furthermore, the reduction must be done in polynomial time.*

Reductions may be chained: if there is a reduction from L to L' and from L' to L'', then the reduction from L to L'' can be constructed simply by chaining the transformations so that the output of the first transformation is the input for the second transformation, all in polynomial time.

Definition 1.5.13 *A language L is hard for the complexity class C, or C-hard, if all languages $L' \in C$ are reducible to L.*

If a language L is C-hard, it basically means that if one can decide L efficiently, one can efficiently decide all languages in C.

Definition 1.5.14 *A language L is complete for the complexity class C, or C-complete, if $L \in C$ and L is C-hard.*

If a language L is C-complete, it basically means that if one can decide L efficiently, one can decide efficiently all languages in C. Moreover, if it is impossible to decide L efficiently, then it is impossible to decide any C-complete language efficiently. In a sense, complete problems for a class are the most representative problems of the complexity of the class.

To prove that a language L is C-hard, it is sufficient to show that some C-complete language L' can be reduced to L. Then, since reductions may be chained, all languages in C may be reduced to L via L'.

To prove that a language L is C-complete, one also has to prove that $L \in C$. This can be done by a direct construction or by reducing L to a language in C.

Oracle Machines

There are still more powerful computational complexity classes. The next question is: what problems could we solve in polynomial time, if we had a oracle that could instantly solve decision problems in some complexity class C? After constructing a string x, the machine may ask the oracle whether $x \in L$, where L is a language in C. It is assumed that the oracle answers instantaneously and never errs. Oracle availability is denoted with a superscript; for example, the class of languages that are decidable by a deterministic polynomial time machine with an oracle that's capable of deciding problems in $\mathbf{NP}$ is denoted with $\mathbf{P}^{\mathbf{NP}}$.

The complexity classes $\mathbf{P}$, $\mathbf{NP}$, and $\mathbf{coNP}$ together constitute the first level of the polynomial hierarchy. Each successive level of the polynomial hierarchy may be built by taking the capabilities of machines in $\mathbf{P}$, $\mathbf{NP}$, and $\mathbf{coNP}$ and giving each an oracle of one lower level. With superscripts denoting the oracle,

$$\begin{aligned}
\Delta_1^{\mathbf{P}} &= \mathbf{P} \\
\Sigma_1^{\mathbf{P}} &= \mathbf{NP} \\
\Pi_1^{\mathbf{P}} &= \mathbf{coNP} \\
\Delta_{i+1}^{\mathbf{P}} &= \mathbf{P}^{\Sigma_i} \\
\Sigma_{i+1}^{\mathbf{P}} &= \mathbf{NP}^{\Sigma_i} \\
\Pi_{i+1}^{\mathbf{P}} &= \mathbf{coNP}^{\Sigma_i}.
\end{aligned}$$

Thus, for example, the class $\mathbf{coNP}^{\mathbf{NP}} = \Pi_2^{\mathbf{P}}$ is the set of languages whose complement ($\mathbf{NP}^{\mathbf{NP}} = \Sigma_2^{\mathbf{P}}$) can be decided in polynomial time by a nondeterministic Turing machine that is allowed to make queries to an $\mathbf{NP}$-oracle.

Example 1.5.15 *The Minimum Circuit problem is in* $\Pi_2^{\mathbf{P}} = \mathbf{coNP}^{\mathbf{NP}}$*: Given a Boolean circuit, is it true that there is no circuit with fewer gates that computes the same Boolean function?*

Here the non-deterministic machine guesses a smaller circuit, and asks the $\mathbf{NP}$*-oracle if there is an input that for which the output of the two circuits is different, accepting the input if there is and rejecting it if there isn't. If an input is accepted, the Boolean circuit must be minimal: since for a co-nondeterministic class acceptance is defined by requiring that all series of guesses lead to acceptance, if an input is accepted it must be true for every* *smaller circuit that there is some input for which the smaller circuit and the original circuit give a different value.*

The Complexity of Counting
There are still other kinds of complexity classes. One fairly difficult class of problems is $\#\mathbf{P}$, the class of counting problems. Unlike the previous complexity classes, the complexity class $\#\mathbf{P}$ is a function class. Instead of simply accepting or rejecting the input, the machine is expected to write the correct answer on the tape and then finish. For these problems, the correct answer is the number of different series of guesses that lead to the accepting state.

Example 1.5.16 *The #Clique Problem is in $\#\mathbf{P}$: Given a graph G and an integer k, how many cliques of order k does G contain? With a non-deterministic Turing machine that guesses k vertices and accepts if they form a clique. Clearly, the number of accepting computations is exactly the number of cliques in G. In fact, #Clique is $\#\mathbf{P}$-complete.*

To bridge the gap between the polynomial hierarchy ($\mathbf{PH}$) and $\#\mathbf{P}$ we consider the complexity class $\mathbf{PP}$. The class $\mathbf{PP}$ is again a decision class. Recall that in $\mathbf{NP}$ an input is accepted if at least one series of lucky guesses leads to the accepting state and that in **coNP** an input is accepted if all series of guesses lead to the accepting state. In $\mathbf{PP}$ an input is accepted if at least half of the possible series of guesses lead to the accepting state. Certainly, if one can count the accepting computations one can also check if the number at least half the number of all possible computations; clearly $\mathbf{PP}$ cannot be stronger than $\#\mathbf{P}$. Also, Toda proved in [45] that any problem in the polynomial hierarchy ($\mathbf{PH}$) can be reduced to $\mathbf{P}^{\mathbf{PP}}$; thus $\mathbf{PH} = \mathbf{P}^{\mathbf{PH}} \subseteq \mathbf{P}^{\mathbf{PP}} \subseteq \mathbf{P}^{\#\mathbf{P}}$. Thus, it would seem tempting to assume that the problems in $\#\mathbf{P}$ are at least as hard and probably harder than the problems in the polynomial hierarchy; unfortunately that seems to be an open question. It seems unknown whether $\mathbf{PH} \subseteq \mathbf{PP}$.

1.5.2 On The Complexity of Ramsey-Theoretical Problems

It is customary to assess the difficulty of a problem by investigating its theoretical computational complexity. However, the computational complexity of determining Ramsey numbers is unknown. The results for related problems, a few of which are summarized below, seem to suggest that the problem is fairly difficult.

Definition 1.5.17 *$F \rightarrow (G, H)$, if and only if every two-color edge-coloring of the graph F contains the graph G in the first color or the graph H in the second color as a subgraph.*

Problem 1.5.18 (Arrowing) *Given F, G, and H, does $F \rightarrow (G, H)$?*

The Arrowing problem can be decided in $\mathbf{coNP}^{\mathbf{NP}}$. First, guess a two-color edge-coloring of F, and then ask the oracle to verify whether the chosen coloring contains a monochromatic G or H. If it does not, reject the input, since clearly $F \not\rightarrow (G, H)$.

Schaefer [42] proved recently that arrowing is $\mathbf{coNP^{NP}}$-complete. The proof relied heavily on constructing a suitable graph F. Thus it is unlikely that a similar construction could be used for the case when F is a complete graph.

Problem 1.5.19 (Clique-arrowing) *Given the graphs $\mathbf{K}_n$, G, and H, does $\mathbf{K}_n \to (G, H)$?*

Clique-arrowing is trivially in $\mathbf{coNP^{NP}}$, since it is a special case of Arrowing. However, the best known lower bound for the complexity of Clique-arrowing is due to Burr, who proved that the problem is **NP**-hard [8]. In that proof Burr showed that, if H is a path, then the minimum n for which $\mathbf{K}_n \to (G, H)$ often depends on the chromatic number of G, determining which is **NP**-hard. A summary of results on the arrowing problem for various graph classes can be found in [9].

Problem 1.5.20 (Ramsey) *Given k, l, and n, does $\mathbf{K}_n \to (\mathbf{K}_k, \mathbf{K}_l)$?*

It is hard to say anything useful about the complexity of the Ramsey problem. On one hand, if k, l, and n are encoded in unary, the problem is a special case of Arrowing and Clique-arrowing and therefore in $\mathbf{coNP^{NP}}$. However, when k, l, and n are encoded in binary, the input length is but a logarithm of the corresponding input length for the Clique-arrowing problem. With such a compact input, polynomial time is insufficient to even guess a single coloring of the $\mathbf{K}_n$, since the coloring would be exponential in size compared to the input. On the other hand, since we're now dealing with complete graphs only, theoretically it might be possible to find a way of deciding the Ramsey problem in polynomial time by looking at the binary representations of k, l, and n.

2 OPTIMIZATION AND SEARCH

In this chapter, we describe how the problem of finding a combinatorial object that satisfies certain criteria can be expressed as an optimization problem. We then describe local search methods suitable for solving such optimization problems.

2.1 SEARCHING AS AN OPTIMIZATION PROBLEM

The discussion in this section is mostly based on [3], a good reference that covers both the mathematical foundations of optimization and computational methods for nonlinear programming.

A Generic Optimization Problem
Consider the following optimization problem:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \text{ for } i = 1, \dots, m \\ & h_i(x) = 0, \text{ for } i = 1, \dots, l \\ & x \in X, \end{array} \tag{2.1}$$

where f, g_i, h_i are functions defined on X. Any vector x of n components $x_1, \dots, x_n$ is called a solution, regardless of whether all the constraints are satisfied. The function f is called the cost function, the constraints $g_i(x) \leq 0$ are called inequality constraints, and the constraints $h_i(x) = 0$ are called equality constraints. A vector $x \in X$ that satisfies all the constraints is called a feasible solution. A feasible solution x^* is called an optimal solution, or global optimum, if $f(x^*) \leq f(x)$ for all feasible x.

A problem is called feasible, if it has a feasible solution, or infeasible otherwise.

Multitudes of algorithms for solving (2.1) exist, depending on the form of f, g_i, h_i and X.

The Penalty Function Method
Generally it is easier to develop algorithms for optimization problems with no constraints than for problems that have constraints. Such problems are called unconstrained optimization problems. For some types of problems, it may be advantageous to transform the constrained optimization problem to a series of unconstrained optimization problems using the penalty function technique described below.

Consider the optimization problem (2.1). Let us replace it with the unconstrained problem (2.2), where $\mu > 0$ is large.

$$\begin{array}{ll} \text{Minimize} & f(x) + \mu \sum_{i=1}^{m} \Phi\left(g_i(x)\right) + \mu \sum_{i=1}^{l} \Psi\left(h_i(x)\right) \\ \text{subject to} & x \in X, \end{array} \tag{2.2}$$

where Φ and Ψ are continuous functions with the following properties:

$$\begin{aligned} \Phi(y) &= 0, \quad && \text{if } y \leq 0, \\ \Phi(y) &> 0, && \text{if } y > 0, \\ \Psi(y) &= 0, && \text{if } y = 0, \\ \Psi(y) &> 0, && \text{if } y \neq 0. \end{aligned}$$

It is intuitively clear that the optimal solution to (2.2) must have all $h_i(x)$ close to zero, and all $g_i(x)$ non-positive or close to zero, for otherwise a large penalty will be incurred.

In general, a penalty function must incur a positive penalty for infeasible points and no penalty for feasible points.

Given an algorithm for unconstrained optimization, the penalty function technique can be used to solve a constrained optimization problem as follows:

1. Initialize the iteration counter i to 1. Choose a $\mu_i > 0$.
2. Use the unconstrained optimization algorithm to find the optimum x_i^* for (2.2) with $\mu = \mu_i$.
3. If x_i^* satisfies all the constraints, then x_i^* is the optimum for (2.1) as well. Otherwise, choose $\mu_{i+1} > \mu_i$, increment i and go to step 2. The solutions x_i^* will generally converge to an optimal solution of the original problem.

The interested reader may find necessary assumptions and proofs for convergence in [3].

Searching For Any Feasible Solution

When searching for a combinatorial structure with given properties, one is usually willing to accept any solution that satisfies the constraints. In such a case, one can simply choose a cost function that's constant over all $x \in X$, e.g. $f(x) = 0$, resulting in (2.3).

$$\begin{aligned} \text{Minimize} \quad & 0 \\ \text{subject to} \quad & g_i(x) \leq 0, \text{ for } i = 1, \ldots, m \\ & h_i(x) = 0, \text{ for } i = 1, \ldots, l \\ & x \in X. \end{aligned} \tag{2.3}$$

Since in this work we search for combinatorial objects with certain structure, the optimization problems are of this type. Naturally, the penalty function method can be applied here as well, resulting in (2.4). In this work, when optimization techniques are used to search for a combinatorial object, the problem is transformed to this form for optimization.

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{m} \Phi\left(g_i(x)\right) + \sum_{i=1}^{l} \Psi\left(h_i(x)\right) \\ \text{subject to} \quad & x \in X. \end{aligned} \tag{2.4}$$

Note that the factor μ, found in (2.2), is unnecessary in (2.4). Also, note that here for feasible solutions and feasible solutions only, the value of the objective function is zero. This gives a good termination condition to any algorithm for solving (2.4).

2.2 LOCAL SEARCH METHODS

Local search methods are heuristic optimization methods. In a local search method, every iteration only small changes to the current solution are considered. No attempt is made to guarantee that the algorithm would eventually search the whole solution space. Usually the name is reserved for heuristics for solving combinatorial problems, such as tabu search and simulated annealing. Even though finding the optimal solution is usually not guaranteed, local search methods have been used with good success in combatting hard combinatorial problems. Descriptions of the most common local search heuristics can be found in [1] and [39]. The former also contains detailed case studies and describes the state of the art in applying local search methods to a handful of practical problems. In the latter, less emphasis is on specific applications, and a great variety of possible enhancements and modifications to the basic algorithms is given.

A general combinatorial optimization problem is given in (2.5).

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \text{ for } i = 1, \ldots, m \\ & h_i(x) = 0, \text{ for } i = 1, \ldots, l \\ & x \in X, \ X \text{ finite.} \end{array} \tag{2.5}$$

At the start of a local search optimization run, generally a random feasible solution is chosen as the current solution. After that, the local search method will iteratively replace the current solution by new current solutions until some termination condition is satisfied. The process of replacing the current solution with a new solution is called a move.

To adapt a local optimization heuristic to a problem, a suitable neighborhood function has to be chosen. The neighborhood function maps each point x to a set of points $N(x)$, which is a subset of X. Some of the points in $N(x)$ may be infeasible, and x may or may not be included in $N(x)$. At every iteration, with the current solution x, only solutions in $N(x)$ — neighbors of x — may be considered for the next current solution. An outline of a local search algorithm is described in Figure 2.1.

The choice of a neighborhood function is of critical importance for the performance of any local search method. The neighborhood function determines which solutions are local optima: the solution $x \in X$ is a local optimum, if no point in $N(x)$ has a lower cost.

In general, the choice of the neighborhood function is crucial for the performance of any local search method. Usually it will help the search, if the cost of a solution correlates with the distance to the global optimum, so that the search can proceed to the region around the global optimum. Local

1. Select a feasible solution $x \in X$
2. Evaluate $f(x')$ for some or all $x' \in N(x)$
3. Let $x = x'$ for one of the feasible neighbors evaluated in step 2, or keep the previous x.
4. Go to step 2, unless the termination condition is satisfied.

Figure 2.1: An outline of a generic local optimization algorithm.

optima cause problems for local search methods. Similarly, large plateaus of equal cost or ridges of large cost separating two areas of lower cost often cause difficulties.

2.2.1 Steepest Descent

The steepest descent method is the simplest of the local optimization methods. In the steepest descent method, all neighbors of the current solution are evaluated. The next current solution is taken to be the neighbor with the lowest cost, if that move would improve the solution. If there are no improving moves, the search is terminated.

From the description it is obvious that the steepest descend method always finds a local optimum. In many cases, however, the global optimum is not found. Instead, the search ends up at a local optimum that is not a global optimum. Dealing with local optima is one of the key problems of designing local optimization algorithms, and there are a few ways to attempt to remedy the situation. Firstly, using a larger neighborhood may reduce the number of local optima, as some solutions that are local optima with the smaller neighborhood may have neighbors with a lower cost with the larger neighborhood. Secondly, running the search repeatedly with different random may cause the search to finish at different local optima, one of which would be the global optimum.

Neither of these has been universally successful. The problem with choosing a larger neighborhood is that it may not remove local optima efficiently enough. Also, more neighbors have to be evaluated each iteration, which makes the algorithm run slower. The problem with iterating the search with different starting points is that there may be a huge number of local optima, or the global optimum may be surrounded by local optima so that the global optimum can be reached only if the starting point is very close to the optimal solution, the chances of which happening are very slim. Thus, the steepest descent method is not very interesting of itself. It has been used to good effect as a part of hybrid local search methods that alternatingly use steepest descent to find local optima and some mechanism such as simulated annealing for avoiding getting stuck at them.

2.2.2 Simulated Annealing

In [33], a simple algorithm to simulate a collection of atoms in equilibrium at a given temperature was presented. Such a system fluctuates randomly in different energy states, the probability of each state depending on the temperature. Statistically, the cooler the system, the closer to the lowest possible energy state the system is. Simulated annealing is an optimization heuristic that is based on an analogy with such a system. An early presentation can be found in [29].

Only one neighbor $x' \in N(x)$ of the current solution x is considered each iteration. If x' is infeasible, the current solution x is retained for the next iteration. If x' is feasible, and $f(x') \leq f(x)$, x' is taken as the new current solution. If x' is feasible, but $f(x') > f(x)$, x' is taken as the new current solution with a non-zero probability. Such uphill moves move the search to a solution with a worse cost, but making such moves will allow the search to escape local minima.

For quantifying the probability with which an uphill move is accepted, a parameter $T > 0$, the temperature, is used. An uphill move is then accepted with probability $e^{\frac{f(x)-f(x')}{T}}$. Since the exponent is negative for $T > 0$, this probability will be between 0 and 1. Moreover, for large T the probability will be close to 1, whereas for small T it will approach zero.

Temperature is generally decreased during simulated annealing. An initial temperature is chosen at the start of the algorithm. In the simplest form of simulated annealing, T is multiplied every iteration by a constant $0 < \alpha < 1$, where α is usually close to 1. As the temperature decreases, the probability of accepting an uphill move decreases as well. Much more elaborate cooling schedules have been used; for some problems, good results have been obtained by alternately heating and cooling the system. This and many other suggestions for improving simulated annealing are discussed in [1] and [39].

2.2.3 Tabu Search

Tabu search is a relatively new search heuristic. The first articles presenting tabu search in its current form are from 1986. An early and fairly thorough description can be found in [22] and [23]. Several articles on applications of tabu search can be found in [24].

Tabu search is closely related to steepest descent. The difference is that in tabu search some neighbors of the current solution may be considered tabu, so that they may not be chosen as the next current solution. In effect, tabu neighbors are excluded from the neighborhood for the iteration.

Whether or not a neighbor is considered tabu depends on the history of the search. Throughout the search, some information on the solutions visited is stored. This data is then used to guide the search, the principal aim being preventing the algorithm for getting stuck at local optima.

The tabu search algorithm used to find the construction for the bound $R(5,9) > 120$, described in more detail in Section 5.6, is an application of tabu search at its simplest. The tabu search algorithm is summarized in Figure 2.2.

In the simplest form of tabu search, the search only has short-term mem-

1. Take a random solution as the current solution.

2. Until the termination condition is satisfied, repeat:

   - Calculate the cost of all non-tabu solutions in the neighborhood of the current solution.
   - Make the non-tabu neighbor with the lowest cost the new current solution.

Figure 2.2: The tabu search algorithm.

ory: only the search history from the last few iterations is considered in determining whether or not a neighbor is tabu. In each iteration, a property of the move from the old current solution to the new current solution is stored in a tabu list. Usually, the tabu list has a fixed length t; if the tabu list already contains t entries, the oldest entry in the tabu list may be removed to make room for the new entry. During the search, a neighbor is considered tabu if the move from the current solution to it matches an entry on the tabu list.

To adapt tabu search to a problem, one needs to choose the representation of solutions, the cost function, the neighborhood function, the tabu condition, and the termination condition. Each of these needs to be tailored to the specific application. Usually there are several possibilities.

Example 2.2.1 *In the traveling salesman problem, the problem is to find the cheapest way for a traveling salesman to visit each of n cities exactly once and return to his origin. A possible solution representation would be a list of the cities in the order they are visited. The most obvious choice for the cost function is to use the cost of the tour. The search could be terminated after a fixed number iterations have been performed without finding an improvement. The neighborhood of a solution could be taken as the set of solution obtained by transposing two adjacent cities in the solution. A possible choice for the tabu condition would be to consider a neighbor tabu, if it is obtained from the current solution by transposing a pair of cities that has been transposed during the past t iterations, where t, the length of the tabu list, is a search parameter.*

The above choices are by no means the only possible ones. Another way of representing the problem would be to use a graph that has vertices corresponding to the cities and an edge between a and b if, during his tour, the traveling salesman travels from a to b or from b to a. A possible neighborhood for such a solution representation would be to take as neighbors all graphs obtained from the current solution by deleting k edges and adding k edges such that the result is a cycle. As the tabu condition, one could for example forbid moves that involve adding an edge that has been deleted from the graph during the past t iterations.

An appealing property of such a basic tabu search method is that once the solution representation, the cost function, the neighborhood, the tabu condition, and the termination condition have been chosen, the only search parameter to tune is t, the length of the tabu list. However, there are many enhancements and modifications that may improve the performance of the tabu search algorithm; for more information, see [1] and [39].

3 GROUPS, PERMUTATIONS AND FIELDS

A group may be obtained by defining a binary operation with certain structure on some set of group elements. Group theory is a convenient tool for examining the symmetries of combinatorial objects. This is the main purpose of examining groups in this work.

In a permutation group, the group elements are permutations of some set. In our case, we will define permutation groups whose elements permute the vertices of a graph. In Chapter 5.3 lower bounds for Ramsey numbers are constructed by finding suitable symmetrical edge-colorings of graphs. The symmetries are defined in terms of permutation groups. Groups, permutations and permutation groups are discussed in Sections 3.1, 3.2, and 3.3.

When, instead of one, two binary operations with certain structure are defined on a set, a field may be obtained. In the past, some edge-colorings without large monochromatic cliques have been obtained by a method based on finite fields. In the finite field method, the edges of a graph are colored based on the structure of a finite field. Finite fields are described in in Section 3.4.

Much of the discussion in this chapter is based on [30].

3.1 GROUPS

Definition 3.1.1 *Let G be a non-empty set. Let $\star$ be a binary operation defined on $G \times G$. Then the pair $(G, \star)$ is called a group, if*

1. *The set G is closed under the operation $\star$, i.e.*

$$\forall g, h \in G, g \star h \in G.$$

2. *The operation $\star$ is associative, i.e.*

$$\forall g, h, k \in G, g \star (h \star k) = (g \star h) \star k.$$

3. *G contains a unit-element e, such that*

$$\forall g \in G, e \star g = g \star e = g.$$

4. *Every element of G has an inverse, i.e.,*

$$\forall g \in G \, \exists h \in G, g \star h = h \star g = e.$$

When the group operation is clear from the context, it may be omitted: one may speak of gh instead of $g \star h$, and of the group G instead of $(G, \star)$.

Definition 3.1.2 *A group is called commutative, or abelian, if the operation $\star$ is commutative, i.e.*

$$\forall g, h \in G, g \star h = h \star g.$$

Example 3.1.3 *$(\mathbb{Z}, +)$ is an abelian group: the sum of any two integers is an integer, associativity holds, 0 is a unit element, and every element a has the inverse $-a$. Also commutativity holds.*

Example 3.1.4 *$(\mathbb{R}, \cdot)$ is not a group: the element 0 has no inverse.*

The usual shorthand notation $g^n = \overbrace{g \star g \star \ldots \star g}^{n \text{ times}}$ may be used.

Definition 3.1.5 *A subgroup $(H, \star)$ of a group $(G, \star)$ is a group defined on a subset $H \subseteq G$ with the same operation $\star$ as the group $(G, \star)$.*

Definition 3.1.6 *Given a group $(G, \star)$ and its subgroup $(H, \star)$, then, for any $g \in G$ the sets $gH = \{gh : h \in H\}$ and $Hg = \{hg : h \in H\}$ are the left and right cosets of H with respect to g, respectively.*

The elements of G may be partitioned into $\frac{|G|}{|H|}$ cosets of cardinality $|H|$.

Definition 3.1.7 *It is said that the group G is generated by the elements $\alpha_1, \ldots, \alpha_r \in G$, if every element $g \in G$ can be expressed as a finite product $g = \alpha_{i_1} \alpha_{i_2} \ldots \alpha_{i_m}$, where $1 \leq i_j \leq r$ for all $1 \leq j \leq m$. The elements $\alpha_1, \ldots, \alpha_r \in G$ are called generators for the group G.*

Definition 3.1.8 *Two groups $(G, \star)$ and $(H, \cdot)$ are said to be isomorphic, if there exists a bijection $\phi : G \mapsto H$ such that $\phi(g_1) \cdot \phi(g_2) = \phi(g_1 g_2)$ for all $g_1, g_2 \in G$.*

3.2 PERMUTATIONS

A permutation is a rearrangement of the elements of a finite set, that is, a bijection from a set to itself. A permutation $\pi : X \mapsto X$ can be given by listing for each element $x \in X$ the element π maps it to. For example, if $X = \{1, 2, \ldots, n\}$, the permutation can be given in list notation as

$$[\pi(1), \ldots, \pi(n)].$$

Since π is a bijection, each element of $\{1, 2, \ldots, n\}$ must occur exactly once in this list. For example,

x	1	2	3	4	5	6	7	8	9	10	11
$\pi(x)$	11	2	4	1	6	5	8	9	7	10	3

where the bottom row is a list of the values of π, is a permutation on $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.

Another notation called the cycle notation will be used predominantly in this work. In cycle notation $\pi = (1, 11, 3, 4)(2)(5, 6)(7, 8, 9)(10)$. This notation is best illustrated by drawing the directed graph, illustrated in Figure 3.1, with the vertex set X and the edges $(x, \pi(x))$ for each $x \in X$. The resulting directed graph is always a union of vertex-disjoint directed cycles. In the cycle notation of a permutation, the vertices in each cycle of the corresponding directed graph are listed in order in parentheses separated

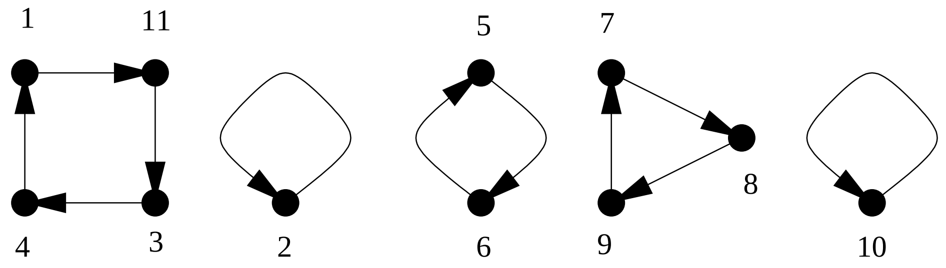

Figure 3.1: The directed graph corresponding to the permutation $\pi = (1, 11, 3, 4)(2)(5, 6)(7, 8, 9)(10)$.

with commas. Cycles of length one may be included for completeness or omitted in the interests of brevity: $\pi = (1, 11, 3, 4)(2)(5, 6)(7, 8, 9)(10) = (1, 11, 3, 4)(5, 6)(7, 8, 9)$. The identity permutation maps each element onto itself and is denoted by $\mathbf{I}$.

The cycle notation of a permutation may be found as follows: While there are elements in X that have not been written yet, find an unwritten element $x \in X$ and write in parentheses x, $\pi(x)$, $\pi(\pi(x))$, etc. until $\pi(\ldots(\pi(x))) = x$, which is not written again.

The multiplication of permutations is most naturally defined as function composition. For two permutations α and β that map each $x \in X$ to $\alpha(x)$ and $\beta(x)$ respectively, the permutation $\alpha\beta$ maps each $x \in X$ to $\alpha(\beta(x))$. For example, if $X = \{0, 1, 2, 3, 4\}$, $\alpha = (1, 2, 3)(0, 4)$, and $\beta = (1, 2, 4)$, then $\alpha\beta = (0, 4, 2)(1, 3)$.

A permutation maps a set to a set by mapping each element separately: $\pi(S) = \{\pi(s) : s \in S\}$.

3.3 PERMUTATION GROUPS

Permutation groups are a useful tool for classifying and analyzing the symmetries of combinatorial objects. Of particular interest is the set of permutations that preserve the structure of a combinatorial object. Such permutations form a permutation group called the automorphism group of the object.

Definition 3.3.1 *The symmetric group* $\mathsf{Sym}(X)$ *is the group of all permutations of the set* X*, with function composition as the group operator.*

$\mathsf{Sym}(X)$ is a group: the composition of two permutations is a permutation, function composition is associative, the identity permutation is an identity element and for each permutation, there is an inverse permutation. Since there are $n!$ of n objects, $\mathsf{Sym}(X)$ has $|\mathsf{Sym}(X)| = |X|!$ elements.

Definition 3.3.2 *A permutation group on* X *is a subgroup of* $\mathsf{Sym}(X)$.

Of particular interest are automorphisms of an object, that is, the mappings from an object to itself that preserve the structure of the object. In the case of graphs, the mappings are permutations and they form a permutation group called the automorphism group of the graph.

Definition 3.3.3 *An automorphism α of a graph $G = (V, E)$ is a permutation of the vertices, for which it holds for all $v_1, v_2 \in V$ that $\{\alpha(v_1), \alpha(v_2)\} \in E$ if and only if $\{v_1, v_2\} \in E$.*

Definition 3.3.4 *The automorphism group* $\mathsf{Aut}(G)$ *of a graph $G = (V, E)$ is the subgroup of* $\mathsf{Sym}(V)$ *consisting of the automorphisms of G.*

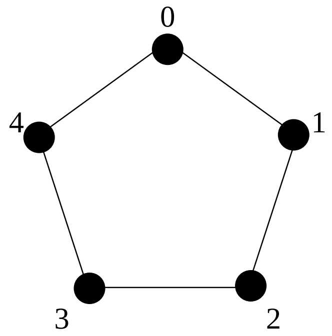

Figure 3.2: The graph of the pentagon.

Example 3.3.5 *Consider the automorphism group of the pentagon P in Figure 3.2. In planar geometry, the automorphisms of the pentagon correspond to the five clockwise rotations of $\frac{2k\pi}{5}$ radians, $k \in \{0, 1, \ldots, 4\}$ about the center of the figure, and the five mirrorings about straight lines, each traveling via the center and one of the corner points of the pentagon. The corresponding permutations form the automorphism group*

$$\mathsf{Aut}(P) = \left\{ \begin{array}{ll} \mathbf{I}, & (0)(1,4)(2,3) \\ (0,1,2,3,4) & (0,2)(1)(3,4) \\ (0,2,4,1,3) & (0,4)(1,3)(2) \\ (0,3,1,4,2) & (0,1)(2,4)(3) \\ (0,4,3,2,1) & (0,3)(1,2)(4) \end{array} \right\}. \tag{3.1}$$

This group consists of 5 rotations and 5 mirrorings (the left and right columns of 3.1 respectively). In general, the automorphism group of an n-vertex cycle graph is isomorphic to the automorphism group D_n of a regular n-gon in planar geometry. The dihedral group D_n consists of n rotations and n mirrorings. For $n > 2$, $|D_n| = 2n$. The cyclic group C_n is the subgroup of D_n with the rotations only: $|C_n| = n$. The cyclic group C_n on $X = \{0, 1, \ldots, n-1\}$ is generated by the permutation $\pi = (0, 1, \ldots, n-1)$; every permutation in the cyclic group can be obtained by applying π repeatedly. The generating permutation of C_5 is illustrated in Figure 3.3.

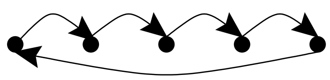

Figure 3.3: The generating permutation of C_5.

Definition 3.3.6 *Let G be a permutation group on X_G and H a permutation group on X_H. The direct product $G \times H$ is the permutation group on $X_G \times X_H$ consisting of all the ordered pairs (g, h) with $g \in G$ and $H \in H$. The permutation (g, h) maps the pair (x_G, x_H) to $(g(x_G), h(x_H))$ for all $x_G \in X_G$, $x_H \in X_H$. The group operation is defined as $(g, h)(g', h') = (gg', hh')$.*

In general, the group $G \times H$ on $X_G \times X_H$ may be interpreted by arranging the elements of $|X_G \times X_H|$ into an array of $|X_G|$ rows and $|X_H|$ columns, and letting then the elements of G permute the rows, while the elements of H permute the columns.

Example 3.3.7 *Recall that the cyclic group C_n on the set $X = \{1, \ldots, n\}$ is generated by the permutation $\pi = (1, \ldots, n)$. The group $C_4 \times C_4$ may be generated by the two generators illustrated in Figure 3.4; the first one rotates the columns and the second one rotates the rows.*

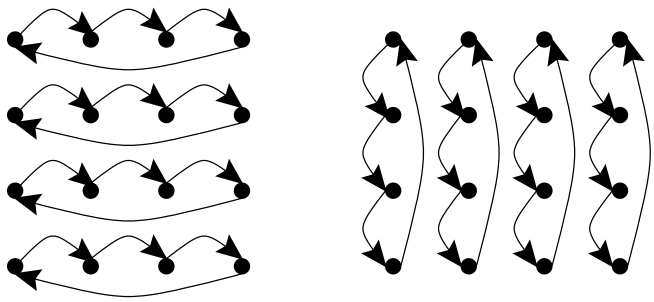

Figure 3.4: The two generators of $C_4 \times C_4$.

Example 3.3.8 *Let $G = \{\mathbf{I}, (0,1,2)(0,2,1)\}$ be a permutation group on $X_G = \{0, 1, 2\}$. Let $H = \{\mathbf{I}, (0, 1)\}$ be a permutation group on $X_H = \{0, 1\}$. Let us denote $X_G \times X_H = \{00, 01, 10, 11, 20, 21\}$ in short. Now,*

$$G \times H = \left\{ \begin{array}{ll} \mathbf{I}, & (00, 01)(10, 11)(20, 21), \\ (00, 10, 20)(01, 11, 21), & (00, 11, 20, 01, 10, 21), \\ (00, 20, 10)(01, 21, 11), & (00, 21, 10, 01, 20, 11) \end{array} \right\}.$$

Here G and H are the cyclic groups C_3 and C_2 respectively, so $G \times H$ is $C_3 \times C_2$. It turns out that $C_3 \times C_2$ is isomorphic to C_6: by renaming the elements $[00, 11, 20, 01, 10, 21] \leftrightarrow [0, 1, 2, 3, 4, 5]$ one obtains the group

$$C_6 = \left\{ \begin{array}{ll} \mathbf{I}, & (0, 3)(1, 4)(2, 5), \\ (0, 2, 4)(1, 3, 5), & (0, 1, 2, 3, 4, 5), \\ (0, 4, 2)(1, 5, 3), & (0, 5, 4, 3, 2, 1) \end{array} \right\}$$

on the set $\{0, 1, 2, 3, 4, 5\}$.

Definition 3.3.9 *Given a permutation group G on the set X, the orbit of the subset $S \subseteq X$ under G is defined by $\mathsf{Orb}(S) = \{g(S) : g \in G\}$.*

Example 3.3.10 *Let G be the cyclic group C_6 on $X = \{0, 1, 2, 3, 4, 5\}$. Consider $\mathsf{Orb}(\{0\})$ and $\mathsf{Orb}(\{0, 3\})$: clearly, each element $g \in G$ maps 0 to a different element; $\mathsf{Orb}(\{0\}) = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$. A straightforward calculation shows that $\mathsf{Orb}(\{0, 3\}) = \{\{0, 3\}, \{1, 4\}, \{2, 5\}\}$.*

The set of k-element subsets of X can be partitioned into orbits. The number of orbits of k-element subsets is denoted by N_k. Orbits will play a very important role in the constructions of colorings in Section 5.3. There a suitable group G on the vertex set V of a complete graph is chosen, and the edges are partitioned into orbits.

Definition 3.3.11 *The orbit incidence matrix A_{tk} is the N_t by N_k matrix such that*

1. *The rows of A_{tk} are labeled by the orbits $\Delta_1, \Delta_2, \ldots, \Delta_{N_t}$ of t-element subsets;*
2. *The columns of A_{tk} are labeled by the orbits $\Gamma_1, \Gamma_2, \ldots, \Gamma_{N_k}$ of k-element subsets;*
3. *The entry $[\Delta_i, \Gamma_j]$ is*

$$A_{tk}[\Delta_i, \Gamma_j] = |\{K \in \Gamma_j : K \supseteq T_0\}|,$$

with $T_0 \in \Delta_i$ any fixed representative.

Example 3.3.12 *(from [30]) Let G be the group on $X = \{0, 1, \ldots, 6\}$ generated by the two permutations $(0)\,(1,2,3)\,(4,5,6)$ and $(0)\,(1,2)\,(5,6)$. The orbits of pairs under G are*

$$\begin{aligned}
\Delta_1 &= \{\{4,5\},\{4,6\},\{5,6\}\} \\
\Delta_2 &= \{\{1,5\},\{2,6\},\{3,4\}\} \\
\Delta_3 &= \{\{1,4\},\{1,6\},\{2,4\},\{2,5\},\{3,5\},\{3,6\}\} \\
\Delta_4 &= \{\{1,2\},\{1,3\},\{2,3\}\} \\
\Delta_5 &= \{\{0,4\},\{0,5\},\{0,6\}\} \\
\Delta_6 &= \{\{0,1\},\{0,2\},\{0,3\}\}.
\end{aligned}$$

The orbits of triples under G are

$$\begin{aligned}
\Gamma_1 &= \{\{4,5,6\}\} \\
\Gamma_2 &= \{\{1,4,6\},\{2,4,5\},\{3,5,6\}\} \\
\Gamma_3 &= \{\{1,4,5\},\{1,5,6\},\{2,4,6\},\{2,5,6\},\{3,4,5\},\{3,4,6\}\} \\
\Gamma_4 &= \{\{1,2,5\},\{1,3,5\},\{2,3,6\},\{1,2,6\},\{1,3,4\},\{2,3,4\}\} \\
\Gamma_5 &= \{\{1,2,4\},\{2,3,5\},\{1,3,6\}\} \\
\Gamma_6 &= \{\{1,2,3\}\} \\
\Gamma_7 &= \{\{0,4,5\},\{0,4,6\},\{0,5,6\}\} \\
\Gamma_8 &= \{\{0,1,5\},\{0,2,6\},\{0,3,4\}\} \\
\Gamma_9 &= \{\{0,1,4\},\{0,2,5\},\{0,3,6\},\{0,1,6\},\{0,2,4\},\{0,3,5\}\} \\
\Gamma_{10} &= \{\{0,1,2\},\{0,2,3\},\{0,1,3\}\}.
\end{aligned}$$

The orbit incidence matrix A_{23} for G is

	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Γ_6	Γ_7	Γ_8	Γ_9	Γ_{10}
Δ_1	1	1	2	0	0	0	1	0	0	0
Δ_2	0	0	2	2	0	0	0	1	0	0
Δ_3	0	1	1	1	1	0	0	0	1	0
Δ_4	0	0	0	2	1	1	0	0	0	1
Δ_5	0	0	0	0	0	0	2	1	2	0
Δ_6	0	0	0	0	0	0	0	1	2	2

For example, the 2 at the intersection of row Δ_5 and column Γ_7 means that each pair in Δ_5 is contained in two three-element subsets in Γ_7.

3.4 FINITE FIELDS

When two binary operations are defined on a set, a field may be obtained. Most of the discussion in this section is based on [46]; this introductory book on cryptology contains a 22 page appendix on the theory of finite fields.

Definition 3.4.1 *A triple $(F, +, \cdot)$ is called a field, if*

1. *$(F, +)$ is a commutative group, called the additive group of the field. Its unit-element is denoted by 0.*
2. *$(F \setminus \{0\}, \cdot)$ is a group, called the multiplicative group of the field. The multiplicative unit-element is denoted by e.*
3. *Distributivity holds, i.e.*

$$\forall r, s, t \in F, r \cdot (s + t) = r \cdot s + r \cdot t \text{ and } (r + s) \cdot t = r \cdot t + s \cdot t.$$

With finite fields the notation $g^n =$ refers to the multiplicative group, that is, $g^n = \overbrace{g \cdot g \cdot \ldots \cdot g}^{n \text{ times}}$. The notation $ng = \overbrace{g + g + \ldots + g}^{n \text{ times}}$ is used for the additive group.

Definition 3.4.2 *An element of $g \in G$ is called an m-ic residue, if there exists an element $h \in G$ such that $h^m = g$.*

A field is called a finite field of order n, if the cardinality of the set F is n. Such fields only exist when n is a prime power. Also, all finite fields of order n are isomorphic . Hence, the notation $\mathbb{F}_n$ or $GF(n)$ used for a finite field of order n defines the field up to isomorphism. For details, see [46, p. 155].

For prime p, the integers modulo p form the field $\mathbb{Z}_p$. In this field both addition and multiplication are defined in the usual way, with the results taken mod p. Fields of order p^k with $k > 1$ are slightly more complicated to construct. The construction given below involves polynomials and a few definitions are necessary.

Definition 3.4.3 *Let $p(x) = f_0 + f_1 x + \ldots + f_k x^k$ be a polynomial with $f_i \in \mathbb{Z}_p$. The polynomial $p(x)$ is said to be a polynomial of degree k over the field $\mathbb{Z}_p$.*

Definition 3.4.4 *Let $p(x)$ and $q(x)$ be polynomials, and let $q(x)$ be of degree $k \geq 1$. Now define $p(x) \bmod q(x)$ to be $r(x)$, the polynomial of minimum degree for which there is a polynomial $d(x)$ such that $p(x) = d(x)\, q(x) + r(x)$.*

Definition 3.4.5 *A polynomial of is said to be irreducible, if it can not be expressed as the product of two polynomials of lower degree.*

A finite field of order p^k, p prime, can be constructed by taking as F the set of polynomials of degree $k-1$ over $\mathbb{Z}_p$, that is, polynomials of the form $f_0 + f_1 x + \ldots + f_{k-1}x^{k-1}$ with $f_i \in \mathbb{Z}_p$ for $0 \leq i \leq k-1$. Addition of two polynomials is defined by

$$\sum_{i=0}^{n} f_i x^i + \sum_{i=0}^{n} g_i x^i = \sum_{i=0}^{n} (f_i + g_i)\, x^i,$$

where the sum $(f_i + g_i)$ is taken in $\mathbb{Z}_p$. Multiplication is defined by

$$\left(\sum_{i=0}^{m} f_i x^i\right)\left(\sum_{j=0}^{n} g_j x^j\right) = \sum_{k=0}^{m+n} \left(\sum_{i+j=k} f_i g_j\right) x^k \bmod q(x),$$

where $q(x)$ is an irreducible polynomial of degree k over $\mathbb{Z}_p$, resulting in a polynomial of degree at most $k-1$. For any given p and k there is always at least one irreducible polynomial; there may be more than one, but recalling that finite fields of the same order are isomorphic, it suffices for our purposes to choose any one of them. A table that gives one primitive polynomial for all p and k with $p^k < 30000$ can be found on the World Wide Web [11]; primitive polynomials are also irreducible.

4 SCHUR NUMBERS

4.1 INTRODUCTION TO SCHUR NUMBERS

In 1916, in his article on the modular version of Fermat's Last Theorem [43], Schur proved that when the integers $\{1, \ldots, N\}$ are partitioned into n classes, if $N > n!e$, then one of the classes will necessarily contain two integers and their difference. In other words, one of the classes will contain two not necessarily distinct integers and their sum.

A partition is called sum-free, if none of the classes contain two not necessarily distinct integers and their sum. A partition into n classes is called an n-partition.

Definition 4.1.1 *The Schur number $s(n)$ is the largest integer k for which there exists a sum-free n-partition of the integers $\{1, \ldots, k\}$.*

One should note that the alternative definition "the smallest integer k for which no sum-free n-partition of the integers $\{1, \ldots, k\}$ exists" has also been used. The definitions are equivalent except for the values being larger by one with the latter definition.

Schur himself gave the bounds $\frac{1}{2}(3^n - 1) \leq s(n) \leq n!e$.

4.2 THE KNOWN SCHUR NUMBERS

The exact value of the Schur numbers is only known for $n \leq 4$. The known values along with sample partitions are given in Figure 4.1.

$$
\begin{aligned}
s(1) = 1 \quad &: \quad \left\{ \ \{1\} \right. \\
s(2) = 4 \quad &: \quad \left\{ \begin{array}{l} \{1,4\} \\ \{2,3\} \end{array} \right. \\
s(3) = 13 \quad &: \quad \left\{ \begin{array}{l} \{1,4,10,13\} \\ \{2,3,11,12\} \\ \{5,6,7,8,9\} \end{array} \right. \\
s(4) = 44 \quad &: \quad \left\{ \begin{array}{c} \{1,4,9,12,19,26,33,36,41,44\} \\ \{2,3,10,11,15,16,29,34,35,42,43\} \\ \{5,6,7,8,17,18,27,28,37,38,39,40\} \\ \{13,14,20,21,22,23,24,25,30,31,32\} \end{array} \right.
\end{aligned}
$$

Figure 4.1: The known Schur numbers.

For $n \leq 4$, verifying that the integers $\{1, 2, \ldots, s(n) + 1\}$ cannot be partitioned into n sum-free partitions can be done by trying out all possibilities, e.g. by using a backtracking procedure. For $n \leq 3$ this can be done by hand, but for $n = 4$ a computer is necessary. The result $s(4) = 44$ was allegedly first obtained by Baumert in 1961.

For $n = 5$, Exoo [17] gives the best currently known lower bound $s(5) \geq 160$.

4.3 LOWER BOUNDS FOR RAMSEY NUMBERS

We are not only interested in Schur numbers because of themselves, but also because of their connection to Ramsey numbers. The following theorem was presented in [17].

Theorem 4.3.1 $R_n(3) \geq s(n) + 2$.

Proof: Identify the vertices of a $\mathbf{K}_{s(n)+1}$ with the integers $0, \ldots, s(n)$. Let the sets P_k, $1 \leq k \leq n$, represent the classes in a sum-free n-partition of the integers $\{1, \ldots, s(n)\}$. Now, color each edge ij, where $i < j$, with color k, iff $(j - i) \in P_k$. The resulting coloring contains no monochromatic triangle, as we shall see. Assume, for contradiction, that the edges between the vertices a, b, and c, where $a < b < c$, form a monochromatic triangle. This means that $b - a$, $c - b$ and $c - a$ must be in the same partition. But $(b - a) + (c - b) = (c - a)$, so the partition would not be sum-free, a contradiction. Since the coloring of $\mathbf{K}_{s(n)+1}$ contains no monochromatic triangle, $R_n(3) \geq s(n) + 2$. □

The best currently known lower bound $s(5) \geq 160$ immediately yields the best currently known lower bound $R_5(3) \geq 162$, both given in [17]. It may naturally be possible to obtain better lower bounds by other methods. For example, $s(3) = 13$, but in [26], it was shown that $R_3(3) = 17$. Similarly $s(4) = 44$, but in [12] it was shown that $R_4(3) \geq 51$.

4.4 LOWER BOUNDS FOR SCHUR NUMBERS

For $n > 4$ the exact value of $s(n)$ is unknown. However, by presenting any sum-free n-partition of $\{1, 2, \ldots, m\}$ one immediately obtains the lower bound $s(n) \geq m$. This is a most natural method of establishing lower bounds for Schur numbers.

4.4.1 Backtracking

In [19] and [20], Fredricksen used a backtracking algorithm to obtain the lower bounds $s(5) \geq 138$ and $s(5) \geq 157$ by constructing sum-free 5-partitions of $\{1, \ldots, 138\}$ and $\{1, \ldots, 157\}$ respectively. In [19], Fredricksen reduced the size of the search space by requiring that the partitions be symmetric. An n-partition of $\{1, 2, \ldots, m\}$ is called symmetric, if for all $1 \leq k \leq m$, k and $m + 1 - k$ are in the same class. The reduction reduced the size of the search space from n^m to $n^{\lceil \frac{m}{2} \rceil}$, or roughly to the square root of the size of the original search space.

Theorem 4.4.1 *No symmetric sum-free partition of* $\{1, 2, \ldots, m\}$ *can exist for* $m \equiv 2$ *mod* 3.

Proof: When $m \equiv 2 \bmod 3$, $\frac{m+1}{3}$ is integer. By the symmetry requirement, $\frac{m+1}{3}$ and $(m + 1) - \frac{m+1}{3}$ must be in the same class. However, $\frac{m+1}{3} + \frac{m+1}{3} = (m + 1) - \frac{m+1}{3}$, and there would be a sum in that class. □

Generally, when arbitrary restrictions are imposed on the search space, there is no guarantee that there's an optimal solution to the original problem in the reduced search space. Indeed, no symmetric partition corresponding to $s(4) = 44$ exists, as $44 = 2 \bmod 3$. Nearly symmetric 4-partitions of $\{1, \ldots, 44\}$, with all elements except 15 and 30 symmetrically placed, exist. Whether there always is a nearly symmetric sum-free n-partition of the integers $\{1, 2, \ldots, m\}$ when there's a sum-free n-partition of $\{1, 2, \ldots, m\}$ is an interesting and apparently open question.

Fredricksen also did not place the pairs of integers $\{i, m + 1 - i\}$ in order; rather, at every node of the search tree the pair was placed that had the smallest number of sets where it could be placed while keeping the set sum-free. In this manner, the branching factor (the number of children of a search node) can be kept low near the root of the tree, leading to a decrease in the total effort required.

In [20], no details of the search method were given, other than that a backtrack search technique was used.

4.4.2 Local Search

In [17], Exoo presented a sum-free 5-partition of the integers $\{1, \ldots, 160\}$, thus showing that $s(5) \geq 160$. In total, Exoo found approximately 10,000 different sum-free partitions, of which 4 were symmetric. The partitions were found by using an unspecified local search technique with an interesting objective function.

As is usual with local search techniques, Exoo's algorithm maintained P, the current n-partition of $\{1, 2, \ldots, m\}$, which consists of the n classes P_i, $1 \leq i \leq n$. In constructing a sum-free n-partition of $\{1, 2, \ldots, m\}$ Exoo was optimizing

$$\max\ c_1 f_1(P) + c_2 f_2(P), \tag{4.1}$$

where

$$f_1(P) = \arg\max_k \left(P_i \cap \{1, 2, \ldots, k\} \text{ is sum-free for all } i = 1, \ldots, n\right),$$

that is, $f_1(P)$ is one less than the smallest sum occurring in the partition, or m, if the partition is sum-free, and

$$f_2(P) = \sum_{i=1}^{n} \sum_{a,b \in P_i} g(a, b)$$

$$\text{where } g(a, b) = \begin{cases} 2n - a - b, & \text{when } a + b > n \\ 0, & \text{otherwise.} \end{cases}$$

Here f_1 is clearly the major term in the cost function. A sum-free partition of $\{1, 2, \ldots, m\}$ is found when $f_1(P) = m$, irrespective of the value of f_2. The significance of f_2 is more subtle. With c_1/c_2 varying between 2^{12} and 2^{18}, its role is to guide the search towards regions of the solution space, where many pairs of numbers in a partition sum to slightly more than n. According to Exoo, (4.1) clearly outperforms the simple count of sums as an cost function. We independently verified the bad performance of the count of sums cost function.

4.4.3 Incremental Construction

Theorem 4.4.2 *For Schur numbers* $s(n+1) \geq 3s(n)+1$.

The following proof is different from the one presented in Schur's original paper [43].

Proof: With $k = s(n)$, let $P = \{P_1, \ldots, P_n\}$ denote a sum-free n-partition of $\{1, 2, \ldots, k\}$. We will construct $P' = \left\{P'_1, \ldots, P'_{n+1}\right\}$, the sum-free $n+1$-partition of $\{1, 2, \ldots, 3k+1\}$. Let $P'_i = \{x | x \in P_i \vee (3k+1-x) \in P_i\}$ for $1 \leq i \leq n$, and $P'_{n+1} = \{k+1, k+2, \ldots, 2k+1\}$. In effect, the numbers from 1 to k are mirrored and copied to the other end of the range $1 \ldots 3k+1$. Clearly P'_{n+1} is sum-free: the smallest element is $k+1$ and the largest element is less than twice that. Suppose that for some $1 \leq i \leq n$ there are $a, b, c \in P'_i$ such that $a+b=c$. Note that by construction, $P'_i \cap \{1, 2, \ldots, 2k+1\} = P_i$. It cannot be that $a, b \leq k$, since then $a+b = c \leq 2k$, and necessarily $a, b, c \in P_i$, but P_i is sum-free. It cannot be that $a, b \geq 2k+2$, since then $a+b = c \geq 4k+4 \notin P'_i$. Suppose then that $a \leq k$ and $b \geq 2k+2$. Let $b' = 3k+1-b$ and $c' = 3k+1-c$. By construction, since $a \leq k$ and $b, c \geq 2k+2$ and $a, b, c \in P'_i$, the elements $a, b', c' \in P_i$. However, $a + c' = b'$, contradicting the assumption that P_i is sum-free. □

In this construction it is striking that in the last partition, the smallest element is relatively large. Also, the partitions in Figure 4.1 and the partition given by Exoo [17] for $s(5) \geq 160$, where the smallest element in one of the sets in the sample partition presented was 44, exhibit a similar property.

This led to the idea of constructing lower bounds for $s(5) \geq k'$ by first constructing a lower bound $s(4) \geq k$, and then searching for a sum-free partition of $\{1, \ldots, k'\}$ keeping the elements $1, \ldots, k$ and $k'+1-k, \ldots, k'$ fixed according to the partition corresponding to the lower bound $s(4) \geq k$ as per the construction above.

A complete backtracking procedure was carried out, taking each 4-partition of as starting point in turn and restricting the search to symmetric partitions. The partitions corresponding to $s(4) \geq 44$ yielded no lower bounds worth mentioning, but starting from $s(4) \geq 43$ some 5800 symmetric 5-partitions of $\{1, \ldots, 160\}$ were found. Starting from $s(4) \geq 40$ an estimated two million symmetric 5-partitions were found. However, even starting from $s(4) \geq 34$, no lower bound greater than 160 was found for $s(5)$.

5 LOWER BOUNDS FOR RAMSEY NUMBERS

In this chapter, various methods used for constructing lower bounds for Ramsey numbers are reviewed. In section 5.1 lower bounds obtained with a method based on finite fields are summarized. In section 5.2 our attempts at improving some lower bounds with the finite field method are summarized. In section 5.3. Section 5.4 contains a description of our attempts to improve Ramsey numbers of the form $R_k(3)$. Section 5.5 contains a description of various local search methods found in the literature applied to constructing lower bounds for Ramsey numbers for, mostly for Ramsey numbers of the form $R(k,l)$. In section 5.6 we describe how the new lower bound $R(5,9) \geq 121$ was found.

5.1 LOWER BOUNDS USING FINITE FIELDS

Greenwood and Gleason [26] gave several lower bounds for Ramsey numbers using a method based on the structure of finite fields.

Theorem 5.1.1 $R(3,3,3) > 16$.

Proof: Identify the vertices of the graph with the elements of $GF(16)$. Associate with each edge the difference of the field elements corresponding to the endpoints of the edge. If that difference is a cubic residue in the multiplicative group of the field, color the edge red. If the difference belongs to the first coset of cubic residues, color it green. If it belongs to the second coset, color it blue. In $GF(16)$, $-1 \equiv 1$, so the order of differencing makes no difference. Suppose that three vertices were completely interconnected by edges of the same color. Without loss of generality, assume they are 0, 1 and A. Of course, $1 - 0 = 1$ is a cubic residue ($1^3 = 1$). Now there would have to exist some $A \in GF(16) \setminus \{0,1\}$ for which both A and $A-1$ are cubic residues other than 0 or 1. However, by calculating the cubic residues in $GF(16)$ it is found that no such pair exists, and therefore there is no monochromatic triangle in the coloring. □

The finite field $GF(16)$ can be calculated by taking the binary polynomials of degree at most three as the field elements and defining multiplication modulo $x^4 + x + 1$ as described in Section 3.4. The cubic residues of the multiplicative group of the field can be found by brute force by raising each non-zero field element to the third power. The set of cubic residues is $\{1, x^3, x^3 + x, x^3 + x^2, x^3 + x^2 + x + 1\}$. The elements and their cubes are listed below.

In the same paper, the exact values for a few Ramsey numbers of small cliques were also given. The following lower bounds were obtained similarly by using the finite field technique:

- $R(3,5) > 13$, obtained by coloring the edges corresponding to differences that are cubic residues with one color and cubic non-residues with the other,

- $R(4,4) > 17$, obtained by coloring the edges corresponding to differences that are quadratic residues with one color and non-residues with the other,
- $R(3,3,3,3) > 41$, obtained by coloring the edges according to the coset of the quartic residues the corresponding difference belongs to,
- $R(3,3,3) > 16$, as described above.

Of these lower bounds only $R(3,3,3,3)$ is not exact. The best currently known lower bound is $R(3,3,3,3) \geq 51$ by Chung [12]. The best known upper bound $R(3,3,3,3) \leq 64$ is due to Sanchez-Flores [41].

Furthermore, in [40] Robertson proved the lower bounds $R(5,5,5) \geq 242$ and $R(6,6,6) \geq 692$ using the finite field technique.

The finite field technique seems especially suitable for constructing lower bounds for Ramsey numbers of the form $R(k,k)$. In [7] Burling and Reyner colored each edge of $\mathbf{K}_p$, with $p = 4r + 1$ prime, with two colors according to whether the difference of the endpoints of the edge corresponded to a quadratic residue in $\mathbb{Z}_p$ or not. Then they calculated the largest cliques in the colorings obtained; clearly, if the order of the largest clique is k, then $R(k+1, k+1) > p$. They gave lower bounds for $R(k,k)$, $5 \leq k \leq 9$, of which $R(6,6) > 101$ and $R(8,8) > 281$ are still the best known bounds.

In [44] and [31], Shearer and Mathon gave apparently independently the best currently known lower bounds $R(7,7) \geq 205$, $R(9,9) \geq 565$, and $R(10,10) \geq 798$. Again, the elements of $\mathbb{Z}_p$ with $p = 4k + 1$ prime are associated with the vertices of $\mathbf{K}_p$ and each edge is colored with the first color if the difference of its endpoints is a quadratic residue in $\mathbb{Z}_p$ and with the second color otherwise. The bounds are based on calculating the order of the largest clique in the resulting two-color edge-colorings; the bound $R(10,10) \geq 798$ was found by noting that in the coloring of $\mathbf{K}_{797}$ the maximum clique was of order 9. Additionally, both showed that if the maximum clique in such a coloring is of order k, then not only $R(k+1, k+1) > p$, but also $R(k+2, k+2) > 2p + 2$. In the construction, two identical copies H and H' of the coloring of $\mathbf{K}_p$ are taken. Then, the edges between H and H' are colored with the first color if the difference corresponds to a quadratic non-residue. Finally, the vertices v and v' are added; the edges from v to H and from v' to H' are colored with the first color. The rest of the edges are colored with the second color, except that the edges from H to H' whose endpoints are associated with the same field element may be colored freely with either color. This construction resulted in the best currently known lower bounds $R(7,7) \geq 205$ and $R(9,9) \geq 565$.

5.2 ATTEMPTS TO IMPROVE $R_K(3)$ USING FINITE FIELDS

In search for improvements of the lower bounds for the Ramsey numbers $R(3, \ldots, 3)$, we checked for each integer $i = p^n \leq 1000$, p prime, whether i was between the best currently known upper and lower bounds for some Ramsey number of the form $R(3, \ldots, 3)$. The best known bounds are shown in (5.1).

$$
\begin{aligned}
& R(3,3) && = 6 \\
& R(3,3,3) && = 17 \\
51 \leq \; & R(3,3,3,3) && \leq 64 \\
162 \leq \; & R(3,3,3,3,3) && \leq 317 \\
500 \leq \; & R(3,3,3,3,3,3) &&
\end{aligned} \tag{5.1}
$$

For each i between the lower and upper bound for some $R_m(3)$, we attempted to construct an m-color triangle-free edge-coloring of $\mathbf{K}_i$. We first checked whether $m \mid i-1$, which is necessary to have m cosets of m-ic residues. If that was the case, the $\frac{i-1}{m}$ m-ic residues were found by exponentiating the elements of the field. Then a straightforward check was made to see whether for any residue A, the set of residues also contains $A-1$.

This occurred in all cases, so the approach did not yield any new lower bounds. Had a field passed this test, it would also have been necessary to make sure that the order of differencing does not matter, that is, for the field GF_n in question and for every $a, b \in GF_n$, the differences $a-b$ and $b-a$ are in the same coset.

5.3 COLORINGS AS UNIONS OF ORBITS

The search space for Ramsey graphs is enormous. A complete graph with n vertices has $\binom{n}{2} = \frac{n(n-1)}{2}$ edges. The number of m-color edge-colorings of the complete graph on n vertices is $m^{\frac{n(n-1)}{2}}$. This number grows far too quickly for any naïve enumerative technique to be successful with any reasonable-sized problem.

In order to limit the size of the search space, additional requirements may be placed on the solution. In particular, the search may be restricted to colorings with a certain symmetry. By choosing a suitable symmetry, one hopes to reduce the search space enough to make the problem manageable, but not so much as to exclude all feasible solutions.

Let G be a permutation group on the vertices V of a complete graph $\mathbf{K}_n = (V, E)$. With a complete graph, the set of edges is the set of two-element subsets of V. Now partition the edges (that is, two-element subsets of V) into orbits $\mathcal{O} = \{O_1, \ldots, O_N\}$, where N is the number of orbits of edges. In this chapter we shall consider colorings where the subset of edges colored with a certain color is a union of some orbits of edges. This is exactly equivalent to searching for colorings with the automorphism group G.

Example 5.3.1 *To show that $R(3,4) > 8$, we shall construct a two-color edge-coloring of $\mathbf{K}_8$ without a $\mathbf{K}_3$ in the first color and without a $\mathbf{K}_4$ in the second color. Additionally, we shall require the coloring to have the cyclic symmetry.*

Let G be the cyclic group C_8 on the elements $X = \{0,1,2,3,4,5,6,7\}$. Associate with each of the vertices of $\mathbf{K}_8 = (V, E)$ one of the elements of X. In this case, the number of orbits $N = 4$. Partitioning the edges into the four *orbits $\mathcal{O} = \{O_1, O_2, O_3, O_4\}$ gives us*

$$\begin{aligned}
O_1 &= \{\{0,1\},\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{6,7\},\{0,7\}\} \\
O_2 &= \{\{0,2\},\{1,3\},\{2,4\},\{3,5\},\{4,6\},\{5,7\},\{0,6\},\{1,7\}\} \\
O_3 &= \{\{0,3\},\{1,4\},\{2,5\},\{3,6\},\{4,7\},\{0,5\},\{1,6\},\{2,7\}\} \\
O_4 &= \{\{0,4\},\{1,5\},\{2,6\},\{3,7\}\}.
\end{aligned}$$

The orbits are illustrated in Figure 5.1. The size of the search space is reduced dramatically: instead of $2^{28} \approx 2.7 \cdot 10^8$ ways of choosing colors for the edges there are only $2^4 = 16$ ways to choose colors for the orbits.

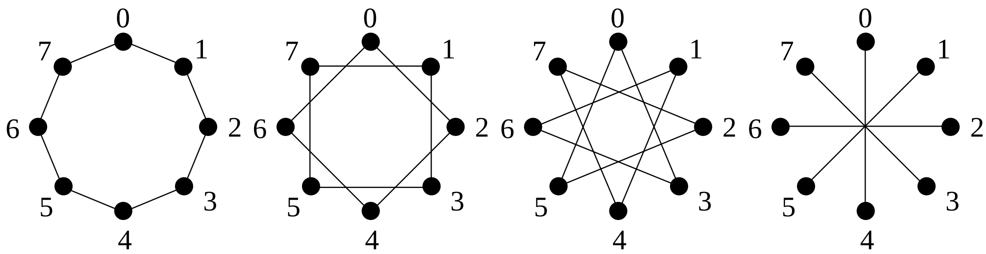

Figure 5.1: The edge-orbits of $\mathbf{K}_8$ under C_8.

By checking each of the 16 symmetric edge-colorings, it is not difficult to find the two that contain no $\mathbf{K}_3$ nor a $\mathbf{K}_4$ in the second color. In one of them, the edges in $O_3 \cup O_4$ is colored with the first color and the edges in $O_1 \cup O_2$ with the second color; in the other, the edges in $O_1 \cup O_4$ are colored with the first color and the edges in $O_2 \cup O_3$ with the second color. The resulting colorings are illustrated in Figure 5.2. Notice that C_8 is only a subgroup of the automorphism group of the coloring, which is D_8. Also notice that the colorings are isomorphic, that is, they are essentially same in the sense that one can be obtained from the other by suitably renaming the vertices.

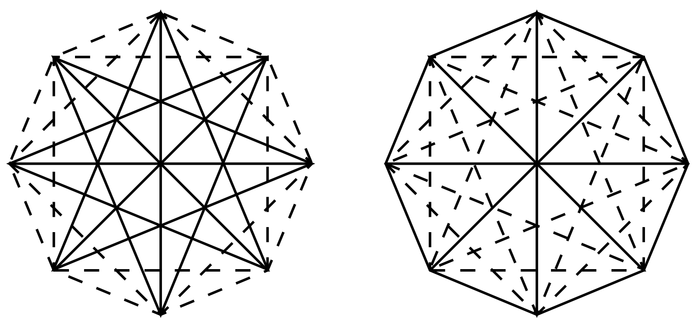

Figure 5.2: Two two-color edge-colorings of $\mathbf{K}_8$.

Typically, the cyclic group C_n has been used as the automorphism group to construct colorings of $\mathbf{K}_n$ without large monochromatic cliques. Constructions with cyclic symmetry account for a significant portion of the lower bounds given in Radziszowski's regularly updated dynamic survey [36].

In [37] Radziszowski and Kreher give an algorithm for searching for graphs on n vertices with no clique of k vertices and no independent set of l vertices. Naturally, this is equivalent to searching for two-colorings of the edges of the complete graph $\mathbf{K}_n$ that contain no clique of k vertices in the first color and no clique of l vertices in the second color.

In their method, a permutation group G on the set of vertices V is chosen, and the orbit incidence matrices A_{2k} and A_{2l} are calculated. For the purposes of the algorithm, the orbit incidence matrices A_{2k} and A_{2l} contain information on which edges occur in which cliques. Each k- or l-subset orbit represents a constraint on the admissible colorings, since, in order not to be monochromatic, each k-clique must contain at least one edge in the second color, and each l-clique must contain at least one edge in the first color:

$$\begin{aligned} \overline{U} \cdot A_{2k} &> 0 \\ U \cdot A_{2l} &> 0 \end{aligned} \tag{5.2}$$

where U is a $(0,1)$-vector, $\overline{U}$ is its complement, and the inequality means that each component of the vector must be greater than zero. Each element of U represents the color of one edge orbit.

Many of the constraints are redundant, and before embarking on the search for a U to satisfy (5.2), the redundant columns of A_{2k} and A_{2l} were removed. In one case ($n = 56$, $k = 5$, $l = 6$, $G = D_{56}$) this led to matrices that were smaller than the original ones by a factor of over 30 000. Then a backtracking procedure was carried out to find admissible colorings.

Radziszowski and Kreher gave lower bounds for $R(4,7)$, $R(4,8)$, $R(4,9)$, $R(5,7)$, and $R(5,8)$ using this method. Of the lower bounds, however, only $R\,(4,9) \geq 69$ has withstood the test of time. In [10] Calkin, Erdős and Tovey searched all cyclic two-color edge-colorings of $\mathbf{K}_p$, p prime; lower bounds were given for $R\,(4,12)$, $R\,(4,15)$, $R\,(5,7)$, and $R\,(5,9)$. The lower bound $R\,(5,7) \geq 80$ is the best currently known lower bound.

5.4 ATTEMPTS TO IMPROVE $R_K(3)$ USING SYMMETRIC COLORINGS

Restricting the search to colorings with cyclic symmetry is equivalent to identifying each node with an integer mod n, and coloring each edge according to the difference (mod n) of its end points. Naturally, since each edge can be of only one color, the colors have to be assigned so that the order of differencing does not matter. For finding triangle-free graphs this reduces to finding Schur partitions of $\{1, \ldots, n-1\}$ that are sum-free mod n. As seen in Chapter 4, the Schur partitions for $s(3) = 13$ and $s(4) = 44$ yield triangle-free colorings of $\mathbf{K}_{14}$ and $\mathbf{K}_{44}$ with 3 and 4 colors respectively.

However, it is possible to do better than that. Triangle-free colorings of $\mathbf{K}_{16}$ and $\mathbf{K}_{50}$ with three and four colors respectively have been constructed. In particular, it is well known that there are exactly two non-isomorphic three-color colorings of $\mathbf{K}_{16}$ that contain no monochromatic triangle in any color [28]. One of these has the automorphism group $C_4 \times C_4$, whereas the other has the automorphism group $C_2 \times C_2 \times C_2 \times C_2$. Also, at least one four-color coloring for $\mathbf{K}_{49}$ with the symmetry $C_7 \times C_7$ exists. This alone should be enough to motivate trying other than cyclic symmetries; groups of the form $C_{n_1} \times C_{n_2} \times \ldots \times C_{n_k}$ were tried next.

First, we searched for new 4-color triangle-free colorings. To improve the lower bounds $R(3,3,3,3) > 50$ by Chung [12], and since $R(3,3,3,3) \leq 64$, shown by Sanchez-Flores [41], it suffices to search the range $51 \leq |V| \leq 63$.

Colorings with automorphism groups of the form C_n were searched exhaustively without success; indeed, a triangle-free cyclic m-coloring on $|V|$ vertices would imply that the Schur number $s(m) \geq |V| - 1$ in a manner very similar to the proof of Theorem 4.3.1, but as discussed in Chapter 4, $s(4) = 44$.

Next, groups of the form $G = C_{n_1} \times C_{n_2} \times \ldots \times C_{n_k}$ were investigated. These groups are finite and abelian. A standard result (found in e.g. [2]) on finite abelian groups shows that every finite abelian group is isomorphic to a direct product of cyclic groups $C_{n_1} \times \ldots \times C_{n_k}$ with $n_1 | n_2 | \ldots | n_m$. It follows that it suffices to search the groups listed in Figure 5.3. Each group G is a permutation group of cardinality $|G| = \prod_{i=1}^{k} n_i$ on $|X| = \prod_{i=1}^{k} n_i$ elements.

$\|V\|$	group
52	$C_2 \times C_{26}$
54	$C_3 \times C_{18}$
54	$C_3 \times C_3 \times C_6$
56	$C_2 \times C_{28}$
56	$C_2 \times C_2 \times C_{14}$
60	$C_2 \times C_{30}$
63	$C_3 \times C_{21}$

Figure 5.3: The automorphism groups tried for improving $R(3,3,3,3) > 50$.

As in the previous section, the orbit incidence matrix A_{23} was calculated in each case, using the GAP software package [21]. An exhaustive backtracking search was then performed, but all colorings contained a monochromatic triangle.

Next, smaller order abelian symmetry groups were tried on graphs with the same number of vertices. Let us define the group Id_n to be the permutation group on the elements $\{1, 2, \ldots, n\}$ consisting of the identity permutation only. Now the group $G = C_{n_1} \times \ldots \times C_{n_{k-1}} \times Id_{n_k}$ on the set X is a permutation group of cardinality $|G| = \prod_{i=1}^{k-1} n_i$ on $|X| = \prod_{i=1}^{k} n_i$ elements. In a sense, n_k parallel copies of $C_{n_1} \times \ldots \times C_{n_{k-1}}$ are created. As an example, the generator of $C_5 \times Id_3$ is illustrated in Figure 5.4.

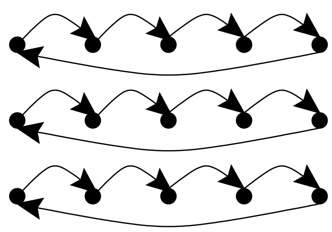

Figure 5.4: The generator of $C_5 \times Id_3$.

Again, orbit incidence matrices were constructed and an exhaustive search was performed under the symmetries $C_n \times Id_2$, where $25 \leq n \leq 31$. Unfortunately, no triangle-free colorings were found.

Even smaller order symmetry groups were then tried. The groups tried included $C_{16} \times Id_3$, $C_{17} \times Id_3$, $C_{19} \times Id_3$, $C_{12} \times Id_4$, $C_{13} \times Id_4$, $C_{11} \times Id_5$, $C_7 \times Id_7$ and $C_5 \times Id_9$. The orbit incidence matrix corresponding to each group was again constructed with GAP. With these groups, however, the search space was too large to search exhaustively. Therefore, the orbit incidence matrix was transformed into a form suitable for input to a tabu search program. The tabu search program was written by Kari Nurmela, and a version of it has been included in the DISCRETA software package [4]. No new lower bounds were found. However, especially for groups of small order this constitutes rather weak evidence for the non-existence of triangle-free colorings with those symmetries, since the smaller order groups do not limit the search space very much, and the search space may simply have been too large.

Some attempts were made to improve the lower bound $R(3,3,3,3,3) \geq 162$ given in [17] using the symmetry groups C_{163}, C_{164}, C_{166}, $C_{13} \times C_{13}$, $C_5 \times C_{35}$, and $C_{34} \times Id_5$. The orbit incidence matrix was calculated and transformed for use with the tabu search program used. No new lower bound was found.

5.5 LOCAL SEARCH FOR TWO-COLOR RAMSEY NUMBERS

In this section, local search methods used to find two-colorings without large monochromatic cliques are described.

Searching all edge-colorings exhaustively for colorings that contain no large monochromatic cliques is computationally feasible only for the smallest problem instances. In the previous section, the requirement that the colorings have a predetermined symmetry was used to limit the search space so that an exhaustive search was again computationally feasible.

For still larger problem instances one would need to use more restrictive symmetries to maintain the computational feasibility of exhaustive search. Reducing in this manner the search space to a smaller and smaller fraction of the original search space increases the risk of all feasible solutions being excluded from the restricted search space. Since local search methods can handle a larger search space than exhaustive search methods, less restrictive symmetries may be used. Due to the stochastic nature of local search methods it may happen that a clique-free solution is never found even if there were one in the search space; however, it is hoped that this is disadvantage is offset by being able to use a less restrictive symmetry, thus increasing the odds that the reduced search space contains a clique-free coloring.

5.5.1 Exoo on $R(3,10)$, $R(3,12)$ and $R(5,5)$

Virtually all lower bounds obtained by local search techniques have made use of cyclic colorings in one form or another. Sometimes slight modifications to a cyclic coloring have been made to construct a coloring of the required type.

In [16] Exoo showed that $R(3,10) \geq 40$ and $R(3,12) \geq 50$, of which the former is still the best known bound for those parameters. First, a cyclic $(3,9)$-coloring on 35 vertices and a cyclic $(3,11)$-coloring on 45 vertices were chosen, respectively. In each case, four vertices were added to the cyclic colorings and the added edges were colored suitably. A simulated annealing procedure was used, where the cost function was based on the number of monochromatic cliques in each color; the number of cliques in each color was multiplied by a suitable weight. For the weights, Exoo suggested the use of the number of the edges in the cliques being counted.

The best currently known lower bound $R(5,5) \geq 43$ was established by Exoo in [15]. First, a cyclic two-coloring of $\mathbf{K}_{43}$ with 43 monochromatic $\mathbf{K}_5$'s in color one was found. Then, one of the vertices was removed from the graph, 16 edges of color one were recolored, and a $(5,5)$-coloring was obtained. A fairly complex genetic algorithm was used, but not described in the article.

5.5.2 Piwakowski with Tabu Search

In [35], Piwakowski gave several lower bounds for two-color Ramsey numbers. The search algorithm was a form of tabu search and the search was restricted to colorings with cyclic symmetry. In the tabu search, the neighborhood of a coloring was the set of colorings obtainable from it by recoloring one edge-orbit. The cost function was simply the number of monochromatic cliques of the given order in the coloring. The tabu condition was that an edge-orbit may not be recolored, if it has been recolored within the past t moves.

Piwakowski gave the lower bounds for $R(3,13) \geq 59$, $R(4,10) \geq 80$, $R(4,11) \geq 96$, and $R(5,8) \geq 95$ using the symmetry groups $C_{29} \times Id_2$, C_{79}, C_{95}, and C_{94}, respectively. Piwakowski also gave lower bounds for $R(4,k)$ with $12 \leq k \leq 14$, but those bounds have been improved on since.

5.5.3 Fifteen Lower Bounds by Exoo

In [18], Exoo gave thirteen new lower bounds for two-color Ramsey numbers and two lower bounds for multi-color Ramsey numbers. Again, the edges were partitioned to edge-orbits and the neighborhood of a coloring was the set of colorings obtainable from it by recoloring one edge-orbit. The search algorithm was a hybrid of simulated annealing and tabu search: Each neighbor is evaluated, one of the T best values obtained is randomly chosen, and the corresponding solution is taken as the new current solution. Additionally, a history list H of previous current solutions was maintained and used as a tabu list; the search was not allowed to move to a previously visited solution. The parameter T, roughly corresponding to the temperature in simulated annealing, was initialized to roughly $\frac{1}{10}$ of the number of edge-orbits and decremented every two or three iterations. If after some time at $T = 1$ no improvement is seen, T was reinitialized to the original value. The history list H was essentially infinite in length.

The lower bounds for $R(6,k)$ with $7 \leq k \leq 10$ and the bounds $R(5,9) \geq 116$, $R(3,4,5) \geq 80$, and $R(3,3,3,4) \geq 87$ were found using the corresponding cyclic groups. Additionally, Exoo gave the lower bounds $R(3,12) \geq 52$ and $R(4,8) \geq 55$ with the symmetry groups $C_{17} \times Id_3$ and $C_{18} \times Id_3$ respectively.

In the same paper [18], Exoo also gave the best currently known lower bounds for $R(5,k)$ with $10 \leq k \leq 15$. The construction was an incremental construction based on smaller colorings with no triangle in the first color or a $\mathbf{K}_{t-1}$ in the second color. Construct B by taking the adjacency matrix of such a coloring and filling the diagonal with ones. Let C be the adjacency matrix of a coloring obtained from B by swapping the colors, but retaining ones on the diagonal. The constructions Exoo gave then had an adjacency matrix of the form

$$A = \begin{pmatrix} B & C & B & B \\ C^t & B & B & B \\ B^t & B^t & B & C \\ B^t & B^t & C^t & B \end{pmatrix}.$$

For each $10 \leq k \leq 15$ Exoo then chose a $(3, k-1)$ coloring and constructed A to establish the bounds. Apparently, the construction alone does not guarantee the non-existence of a $\mathbf{K}_5$ or $\mathbf{K}_k$, but one also has to choose a suitable $(3, k-1)$-coloring: as mentioned, in the same paper Exoo showed that $R(3,12) \geq 52$, but for his construction to show that $R(5,13) \geq 193$ he used a cyclic $(3,12)$-coloring on 48 vertices instead of the 51-vertex $(3,12)$-coloring he had just constructed.

5.6 A NEW LOWER BOUND FOR $R(5,9)$

In this section a new lower bound for a Ramsey number, $R(5,9) > 120$, is presented. The bound is obtained by constructing a two-color edge-coloring of the complete graph with 120 vertices such that the coloring contains neither a clique of order 5 in one color nor a clique of order 9 in the other. The construction was found by using tabu search and restricting the search to cyclic edge-colorings.

For $R(5,9)$, Calkin, Erdős, and Tovey [10] gave the first non-trivial lower bound $R(5,9) > 113$ by constructing a two-color edge-coloring of $\mathbf{K}_{113}$ with no $\mathbf{K}_5$ in the first color nor a $\mathbf{K}_9$ in the second color. The construction was found by searching exhaustively the two-color edge-colorings of $\mathbf{K}_{113}$ with cyclic symmetry.

According to [36] the bound $R(5,9) > 115$, given by Exoo [18], is the best previously known lower bound. The corresponding construction was found by using a local search method and searching for an edge-coloring with cyclic symmetry.

The construction corresponding to the new lower bound $R(5,9) > 120$ was found by applying tabu search and, again, restricting the search to edge-colorings with cyclic symmetry.

5.6.1 The Search Method

We use tabu search in the style of Piwakowski [35] to search for a two-color edge-coloring of $\mathbf{K}_n$ with no $\mathbf{K}_k$ in the first color and no $\mathbf{K}_l$ in the second color as a subgraph. Tabu search is a local search method; in every iteration the current solution is replaced by one of its neighbors. Tabu search is stochastic by nature: it is not guaranteed to always find an optimal solution even if one exists. Nevertheless, tabu search is sometimes very effective in solving combinatorial problems.

The tabu search algorithm used is described on a general level in Figure 5.5.

---

1. Take a random solution as the current solution.
2. Until the termination condition is satisfied, repeat:
   - Calculate the cost of all non-tabu solutions in the neighborhood of the current solution.
   - Make the non-tabu neighbor with the lowest cost the new current solution.

Figure 5.5: The tabu search algorithm

To complete the description of the algorithm, we next specify the solution space, neighborhood, cost function, and tabu condition used.

The solution space used is not the set of all two-color edge-colorings of the $\mathbf{K}_n$. Instead, we restrict the size of the search space by partitioning the edges into edge sets and requiring that all edges in each edge set be colored with the same color. The solution space is then the set of two-colorings of the edge sets. In particular, the vertices are identified with the integers $1, \ldots, n$, and the edges are partitioned according to the distance of their endpoints into $\left\lfloor \frac{n}{2} \right\rfloor$ edge sets $\mathcal{E}_d$:

$$\mathcal{E}_d = \{\{i,j\} : \mathrm{dist}\,(i,j) = d\}\,,\ 1 \le d \le \left\lfloor \frac{n}{2} \right\rfloor ,$$
$$\text{where } \mathrm{dist}(i,j) = \min\left(|i-j|\,, n - |i-j|\right).$$

This is equivalent to restricting the search to edge-colorings whose automorphism group has the cyclic group acting on the vertices as a subgroup.

The neighborhood of a coloring is the set of colorings obtainable from the coloring by changing the color of one edge set.

The cost of a coloring is the number of $\mathbf{K}_k$'s in the first color plus the number of $\mathbf{K}_l$'s in the second color.

A neighboring coloring is tabu, if the edge set where the current and neighboring colorings differ has been recolored within the past t iterations. Here t, the length of the tabu list, is a search parameter.

5.6.2 The Construction

Using the technique described above with $n = 120, k = 5, l = 9$, and $t = 12$, we found a two-color edge-coloring of $\mathbf{K}_{120}$ that contains neither a $\mathbf{K}_5$ in the first color nor a $\mathbf{K}_9$ in the second color. With

$$S = \{2, 3, 6, 7, 13, 15, 17, 18, 19, 20, 22, 23, 28,$$
$$29, 31, 33, 41, 42, 43, 45, 48, 52, 53, 54, 60\},$$

the coloring may be constructed by coloring the edges in edge sets $\mathcal{E}_d$ for which $d \in S$ with the first color and the remaining edges with the second color. Using a computer it is not hard to verify that the resulting coloring contains no $\mathbf{K}_5$ in the first color and no $\mathbf{K}_9$ in the second color. Thus, this construction yields the new lower bound $R(5, 9) > 120$.

6 CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this report, methods used for constructing lower bounds for Ramsey numbers are reviewed and the new lower bound $R(5,9) > 120$ is presented. The new bound is found by using tabu search and restricting the search to cyclic edge-colorings.

Several best currently known lower bounds and the methods used to obtain them are reviewed. The results reviewed are summarized in Figure 6.1.

k \ l	3	4	5	6	7	8	9	10
3	6^S	9^S	14^F	18	23	28^G	36	40^L 43
4		18^F	25	35 41	49 61	55^{Lc} 84	69^C 115	80^{LC} 149
5			43^{Lc} 49	58 87	80^C 143	95^{LC} 216	121^{LC} 316	141^{Lc} 442
6				102^F 165	109^{LC} 298	122^{LC} 495	153^{LC} 780	167^{LC} 1171
7					205^f 540	1031	1713	2826
8						282^F 1870	3583	6090
9							565^f 6625	12715
10								798^f 23854

S:	bound obtained with simple analysis
L:	construction obtained with local search
C:	corresponding construction has cyclic symmetry
c:	corresponding construction is based on cyclic symmetry
F:	construction corresponds to the structure of a finite field
f:	construction is based on the structure of a finite field
G:	construction presented by Grinstead and Roberts [27]

Figure 6.1: Methods used for lower bounds of Ramsey numbers $R(k,l)$

It is striking that almost all best known lower bounds for Ramsey numbers $R(k,k)$ have been found by using constructions based on finite fields, and that virtually all lower bounds for $R(k,l)$ with $k \neq l$ have been found by using cyclic symmetries.

Several ideas for further research have surfaced during the writing of this report. To round off this work, some of them are sketched.

Symmetries:

On one hand, the vast majority of lower bounds listed in Radziszowski's survey [36] has been obtained by constructing a coloring with a cyclic symmetry. On the other hand, according to Bollobás [5, p. 185], "The two-colourings of $\mathbf{K}_n$ without large monochromatic complete subgraphs lack order: they look as if they had been chosen at random." There is a trade-off: with the current search algorithms one must limit the search space, but one hopes that the limitation affect the result as little as possible. There seems to be no particular reason why cyclic groups seem to perform so well, and trying out other groups seems warranted. Unfortunately, choosing the right groups is not easy. If the optimal colorings really lack structure, examining known optimal colorings may not suggest any useful symmetries to try for other cases.

Typically, the symmetries used have restricted certain edges to be colored with the same color. Except for the reduction in the size of the search space, this seems counterproductive — it would seem more sensible to restrict certain edges to be colored with a different color.

In the scheme used, the group $G \subset \mathsf{Sym}(V)$ is a group of permutations of the vertices; the edges have been partitioned into orbits under G and all the edges in an orbit have been colored with the same color. Equivalently, one could partition the set $E \times C$ of edge-color pairs into orbits under G and look for a subset of those orbits such that each edge is covered by exactly one edge-color pair and the resulting edge-coloring contains no large monochromatic clique. Since G does not permute the colors, all edge-color pairs in an orbit have the same color. To generalize this, one could allow the elements of G to permute the colors as well: let $G \subset \mathsf{Sym}(V) \times \mathsf{Sym}(C)$. Then, the element $(g_V, g_C) \in G$ would permute the edge-color pair $(\{v_1, v_2\}, c)$ to $(\{g_V(v_1), g_V(v_2)\}, g_C(c))$. Again, partition $E \times C$ into orbits — now edge-color pairs in the same orbit may have different color — and, again, look for a subset of the orbits such that each edge is covered exactly once and the resulting coloring contains no large monochromatic clique.

As a simple example, number the vertices of a $\mathbf{K}_5$ from 1 to 5 and let G be the group generated by $((1,2,3,4,5), \mathbf{I})$ and $((1,2,4,3), (\mathrm{r}, \mathrm{g}))$ with r and g representing red and green respectively. Taking the edges of the $\mathbf{K}_5$ as E and $\{\mathrm{r}, \mathrm{g}\}$ as C, the group G partitions $E \times C$ into two orbits. One of them is exactly the coloring presented in Figure 1.2 on page 4.

Cost Function:

In most local search techniques used for constructing colorings without large monochromatic cliques, the cost function used has been the number of monochromatic cliques of a certain order. Recall that deciding whether a graph contains a clique of a certain order is an $\mathbf{NP}$-complete problem, and that arrowing is a $\mathbf{coNP}^{\mathbf{NP}}$-complete problem. From Section 1.5 we know that $\mathbf{coNP}^{\mathbf{NP}} \subseteq \mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}}$. That is, it should be possible to decide the arrowing problem—and presumably other, significantly more difficult problems—in polynomial time, with at most a polynomial number of calls to the clique-counting subroutine. However, the local search algorithms offer no guarantee of deciding the problem correctly in any reasonable time. This would seem suggesting trying some slightly different approach, such as trying a computationally easier cost function.

Interestingly, in searching for a Schur partition of the integers $\{1, \ldots, 160\}$ Exoo reported a far better performance with a cost function (Equation 4.1) whose main term was the smallest sum occurring in any of the partitions than with an cost function that calculated the number of sums. Perhaps in the search for colorings without monochromatic cliques one could use as an cost function a function based on finding one clique and considering the vertices in the clique.

Finite Field Method:
Since all finite fields of a given order are isomorphic, it should be possible to try all finite fields up to a certain order systematically. One could then calculate the set of m-ic residues and the corresponding cosets for some reasonable values of m and check whether the corresponding coloring gives a new lower bound for some Ramsey number by calculating the order of the largest clique in each color.

Search for Colorings as Satisfiability Problem:
The satisfiability problem is the problem of deciding whether for a given boolean expression there is a truth assignment that makes the expression true. Searching for a two-color clique-free edge-coloring with a given symmetry can be expressed as a satisfiability problem. Suppose that the edges of the complete graph have been partitioned into orbits $O_1, \ldots, O_N$ and that we're trying to avoid a monochromatic clique on k vertices. Let r_i be a boolean variable which is true if O_i is colored with red, and false if O_i is colored with green. Suppose, for example, that there is a clique of order at least k in the graph formed by taking the edges in O_2, O_7, and O_9. Then, all three orbits may not be colored with the same color. This gives us two constraints: at least one of the orbits must be colored with red, and at least one of the orbits must be colored with green. As a boolean expression, this translates to $(r_2 \vee r_7 \vee r_9) \wedge (\neg r_2 \vee \neg r_7 \vee \neg r_9)$. By calculating all constraints, one can form a logical expression that is satisfiable if and only if there is an edge-coloring with the prescribed symmetry that contains no large monochromatic clique.

This idea is strongly based on the work in [37], described in Section 5.3. In that work, essentially a constraint was created for each color and every orbit of cliques. Then redundant constraints were filtered out. The problem is that the number of k-element subset orbits grows rapidly. There is hope, however, that calculating the non-redundant constraints could be done efficiently without calculating all k-element subset orbits.

Bibliography

[1] E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization.* John Wiley & Sons, New York, 1997.

[2] M. A. Armstrong. *Groups and Symmetry.* Springer, Berlin, 1988.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming – Theory and Algorithms.* John Wiley & Sons, Inc., New York, 2nd edition, 1993.

[4] A. Betten, E. Haberberger, R. Laue, and A. Wassermann. Discreta – a program to construct t-designs with prescribed automorphism group [online]. <URL: http://www.mathe2.uni-bayreuth.de/~discreta/>, 1999. Cited May 31st 2000.

[5] B. Bollobás. *Modern Graph Theory.* Springer, Berlin, 1991.

[6] A. Brauer and H. Rohrbach, editors. *Issai Schur / Gesammelte Abhandlungen*, volume 2. Springer, Berlin, 1973.

[7] J. P. Burling and S. W. Reyner. Some lower bounds of the Ramsey numbers $n(k, k)$. *J. Combin. Theory Ser. B*, 13:168–169, 1972.

[8] S. A. Burr. Determining generalized Ramsey numbers is NP-hard. *Ars Combin.*, 17:21–25, 1984.

[9] S. A. Burr. On the computational complexity of Ramsey-type problems. In J. Nešetřil and V. Rödl, editors, *Mathematics of Ramsey Theory*, pages 46–52. Springer, Berlin, 1990.

[10] N. J. Calkin, P. Erdős, and C. A. Tovey. New Ramsey bounds from cyclic graphs of prime order. *SIAM J. Discrete Math.*, 10:381–387, 1997.

[11] L. Chabaud. Primitive polynomials [online]. <URL: http://www.ens.fr/~chabaud/Poly/primindex.html>, 2000. Cited May 31st 2000.

[12] F. R. K. Chung. On the Ramsey numbers $N(3, 3, \ldots, 3; 2)$. *Discrete Math.*, 5:317–321, 1973.

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

[14] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.

[15] G. Exoo. A lower bound for $R(5, 5)$. *J. Graph Theory*, 13:97–98, 1989.

[16] G. Exoo. On two classical Ramsey numbers of the form $R(3, n)$. *SIAM J. Discrete Math.*, 2:488–490, 1989.

[17] G. Exoo. A lower bound for Schur numbers and multicolor Ramsey numbers of K_3. *Electron. J. Combin.*, 1:#R8, 1994.

[18] G. Exoo. Some new Ramsey colorings. *Electron. J. Combin.*, 5:#R29, 1998.

[19] H. Fredricksen. Five sum-free sets. In *Proceedings of the Sixth Southeastern Conference On Combinatorics, Graph Theory, And Computing*, number XIV in Congressus Numerantium, pages 309–314, 1975.

[20] H. Fredricksen. Schur numbers and the Ramsey numbers $N(3, 3, 3, \ldots, 3; 2)$. *J. Combin. Theory Ser. A*, 27:376–377, 1979.

[21] The GAP Group, Aachen, St Andrews. *GAP – Groups, Algorithms, and Programming, Version 4.1* [online], 1999. <URL: http://www-gap.dcs.st-and.ac.uk/~gap>. Cited May 31st 2000.

[22] F. Glover. Tabu search—Part I. *ORSA J. Comput.*, 1:190–206, 1989.

[23] F. Glover. Tabu search—Part II. *ORSA J. Comput.*, 2:4–32, 1990.

[24] F. Glover, editor. Tabu search methods for optimization. *European Journal of Operational Research*, 106(2/3), 1998.

[25] R. L. Graham, B. L. Rothschild, and J. H. Spencer. *Ramsey Theory*. John Wiley & Sons, 1980.

[26] R. E. Greenwood and A. M. Gleason. Combinatorial relations and chromatic graphs. *Canad. J. Math.*, 7:1–7, 1955.

[27] C. M. Grinstead and S. M. Roberts. On the Ramsey numbers $R(3, 8)$ and $R(3, 9)$. *J. Combin. Theory Ser. B*, 33:27–51, 1982.

[28] J. G. Kalbfleisch and R. G. Stanton. On the maximal triangle-free edge-chromatic graphs in three colors. *J. Combin. Theory Ser. B*, 5:9–20, 1968.

[29] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[30] D. L. Kreher and D. R. Stinson. *Combinatorial Algorithms – Generation, Enumeration, and Search*. CRC Press, Boca Raton, 1999.

[31] R. Mathon. Lower bounds for Ramsey numbers and association schemes. *J. Combin. Theory Ser. B*, 42:122–127, 1987.

[32] B. D. McKay and S. P. Radziszowski. $R(4, 5) = 25$. *J. Graph Theory*, 19:309–322, 1995.

[33] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953.

[34] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[35] K. Piwakowski. Applying tabu search to determine new Ramsey graphs. *Electron. J. Combin.*, 3:#R6, 1996.

[36] S. P. Radziszowski. Small Ramsey numbers. *Electron. J. Combin.*, 1:#DS1, 1994. Revision #6: July 5, 1999.

[37] S. P. Radziszowski and D. R. Kreher. Search algorithm for Ramsey graphs by union of group orbits. *J. Graph Theory*, 12:59–72, 1988.

[38] F. P. Ramsey. On a problem in formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.

[39] C. R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems.* John Wiley & Sons, Inc., New York, 1993.

[40] A. Robertson. New lower bounds for some multicolored Ramsey numbers. *Electron. J. Combin.*, 6:#R3, 1999.

[41] A. Sanchez-Flores. An improved upper bound for Ramsey number $N(3,3,3,3;2)$. *Discrete Math.*, 140:281–286, 1995.

[42] M. Schaefer. Graph Ramsey theory and the polynomial hierarchy. *Proc. 31th STOC*, pages 591–601, 1999.

[43] I. Schur. Über die Kongruenz $x^m + y^m = z^m \bmod p$. *Jahresber. Deutsch. Math.-Verein.*, 25:114–117, 1916. Reprinted in [6].

[44] J. B. Shearer. Lower bounds for small diagonal Ramsey numbers. *J. Combin. Theory Ser. A*, 42:302–304, 1986.

[45] S. Toda. On the computational power of PP and $\oplus$P. *Proc. 30th FOCS*, pages 514–519, 1989.

[46] H. C. van Tilborg. *An Introduction to Cryptology.* Kluwer Academic Publishers, Dordrecht, 1988.

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A52	Ilkka Niemelä, Torsten Schaub (Eds.) Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning. May 1998.
HUT-TCS-A53	Stefan Rönn Semantics of Semaphores. 1998.
HUT-TCS-A54	Antti Huima Analysis of Cryptographic Protocols via Symbolic State Space Enumeration. August 1999.
HUT-TCS-A55	Tommi Syrjänen A Rule-Based Formal Model For Software Configuration. December 1999.
HUT-TCS-A56	Keijo Heljanko Deadlock and Reachability Checking with Finite Complete Prefixes. December 1999.
HUT-TCS-A57	Tommi Junttila Detecting and Exploiting Data Type Symmetries of .... December 1999.
HUT-TCS-A58	Patrik Simons Extending and Implementing the Stable Model Semantics. April 2000.
HUT-TCS-A59	Tommi Junttila Computational Complexity of the Place/Transition-Net Symmetry Reduction Method. April 2000.
HUT-TCS-A60	Javier Esparza, Keijo Heljanko A New Unfolding Approach to LTL Model Checking. April 2000.
HUT-TCS-A61	Tuomas Aura, Carl Ellison Privacy and accountability in certificate systems. April 2000.
HUT-TCS-A62	Kari J. Nurmela, Patric R. J. Östergård Covering a Square with up to 30 Equal Circles. June 2000.
HUT-TCS-A63	Nisse Husberg, Tomi Janhunen, Ilkka Niemelä (Eds.) Leksa Notes in Computer Science. October 2000.
HUT-TCS-A64	Tuomas Aura Authorization and availability - aspects of open network security. November 2000.
HUT-TCS-A65	Harri Haanpää Computational Methods for Ramsey Numbers. November 2000.

ISBN 951-22-5246-5
ISSN 1457-7615