Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 89
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 89
Espoo 2004 HUT-TCS-A89

CONSTRUCTING CERTAIN COMBINATORIAL STRUCTURES BY COMPUTATIONAL METHODS

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 89
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 89
Espoo 2004 HUT-TCS-A89

CONSTRUCTING CERTAIN COMBINATORIAL STRUCTURES BY COMPUTATIONAL METHODS

Harri Haanpää

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering, for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 27 of February, 2004, at 12 o'clock noon.

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 951-22-6941-4
ISSN 1457-7615

Otamedia
Espoo 2004

ABSTRACT: Combinatorics is a branch of mathematics that generally deals with a finite or at most countably infinite set and collections of its subsets. These collections must then satisfy certain criteria depending on the class of objects and the problem being considered.

The most fundamental problem in combinatorics is the problem of existence: Does a combinatorial structure that satisfies the given requirements exist? In general, it is straightforward to verify that a proposed structure satisfies the required criteria, but finding a structure of the required type is difficult. If a structure of the required type exists, any method that constructs one is sufficient to settle the existence question.

Two problems closely related to the existence problem are the enumeration problem—how many different combinatorial structures of the required type exist—and the optimization problem—which combinatorial structure of the required type is the best, judged by some criterion.

A computer may be very useful in solving problems of the three types mentioned above. If it is suspected that a structure of the required kind exists, one may design a computer program to sample the space of possible structures until one that satisfies the criteria is found. To show the nonexistence of a structure, to enumerate the structures of a given kind, or to determine the best structure of a given kind, it is generally necessary to conduct a case-by-case analysis of all possible structures, which is a task for which a computer is especially suited. It is, however, often a nontrivial task to design an efficient algorithm for such an analysis.

In this thesis several ways of applying computational methods to combinatorial problems are described. Tabu search on graphs with cyclic symmetry is used to obtain a lower bound for a Ramsey number, an orderly backtrack search with isomorph rejection is applied to a particular class of codes to classify certain designs and the whist tournaments with up to thirteen players, and another orderly search is used to obtain the optimal sum and difference packings and covers of small Abelian groups.

KEYWORDS: balanced incomplete block design, difference cover, difference packing, isomorph rejection, orderly algorithm, Ramsey number, Sidon set, sum cover, sum packing, whist tournament

CONTENTS

PREFACE

This thesis is a result of studies and research at the Laboratory for Theoretical Computer Science of Helsinki University of Technology from 1998 to 2003. While a dissertation represents the achievements of an individual, there is a great number of people, interaction with whom has been absolutely essential in shaping the thesis.

I am very grateful to the previous head of the laboratory, Professor Emeritus Leo Ojala, and his successor, Professor Ilkka Niemelä, for the opportunity to work in the Laboratory for Theoretical Computer Science. My discussions with my thesis supervisor, Professor Pekka Orponen, have been a source of delight and inspiration. I am greatly indebted to Professor Patric Östergård, my thesis advisor, for guiding me into the world of science. In addition to Professor Östergård, my other co-authors Antti Huima and Petteri Kaski also deserve great credit. Major parts of this thesis are results of cooperation with them. I am also grateful to Prof. Veerle Fack and Dr. Alfred Wassermann for acting as the pre-examiners of this thesis.

In addition to the Laboratory for Theoretical Computer Science, this thesis has been financed or supported by the Helsinki Graduate School of Computer Science and Engineering (HeCSE), the Academy of Finland (project no. 44517), the Nokia Foundation, Foundation of Technology (Tekniikan edistämissäätiö), and Emil Aaltosen Säätiö. Their support is gratefully acknowledged.

I wish to thank my parents for their encouragement. Finally, I would like to thank my wife Elli and our children Helmi and Hannes both for supporting me in my work and for occasionally distracting me from it.

1 INTRODUCTION

> **Combinatorics:** The branch of mathematics dealing with combinations of objects belonging to a finite set in accordance with certain constraints, such as those of graph theory; combinatorial analysis.
>
> (Oxford English Dictionary [40])

Combinatorics is a branch of mathematics that generally deals with a finite or at most countably infinite set and collections of its subsets. These collections must then satisfy certain criteria depending on the class of objects and the problem being considered.

The existence problem is the most fundamental in combinatorics: Does a combinatorial structure that satisfies the given requirements exist? When the existence question has been settled in the affirmative, one may ask how many different combinatorial structures exist, and what they are like. Alternatively, one may wish to choose the combinatorial structure that satisfies the criteria and is best according to some criterion. One may also ask whether there exist combinatorial structures that satisfy the given requirements and certain additional criteria.

In this thesis several ways of using computational methods to solve combinatorial problems are examined. In [P1] tabu search is used to find a graph whose existence proves a lower bound for the Ramsey number $R(5,9)$. The search is restricted to structures with a prescribed automorphism group. In [P2, P3] exhaustive search with isomorph pruning is used to completely classify the whist tournaments with up to 13 players. The whist tournaments are classified by first classifying a class of codes closely related to the block designs underlying the whist tournaments. In [P4, P5, P6] exhaustive searches are carried out to obtain optimal sum and difference covers and packings of cyclic and Abelian groups. Some analytical bounds are also given.

2 STRUCTURE OF THE THESIS

This thesis consists of six articles and this summary. This chapter contains a description of the contents of this thesis and the role of the author in articles with more than one author.

Chapter 3 describes some computational methods central to this thesis. Chapters 4 to 6 contain brief literature surveys and descriptions of the ideas in the articles. Finally, the results in this thesis are summarized in Chapter 7.

Together with this summary, the following six articles constitute this thesis.

[P1] H. Haanpää. A lower bound for a Ramsey number. *Congressus Numerantium*, 144:189–191, 2000.

[P2] H. Haanpää and P. R. J. Östergård. Classification of whist tournaments with up to 12 players. *Discrete Appl. Math.*, 129:399–407, 2003.

[P3] H. Haanpää and P. Kaski. The near resolvable 2-(13,4,3) designs and thirteen-player whist tournaments. *Des. Codes Cryptogr.* To appear.

[P4] H. Haanpää, A. Huima, and P. R. J. Östergård. Sets in $\mathbb{Z}_n$ with distinct sums of pairs. *Discrete Appl. Math.* To appear.

[P5] H. Haanpää and P. R. J. Östergård. Sets in Abelian groups with distinct sums of pairs. Research Report A87, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, February 2004.

[P6] H. Haanpää. Minimum sum and difference covers of Abelian groups. Research Report A88, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, February 2004.

2.1 SUMMARY OF THE ARTICLES IN THE THESIS

In this section, the contents of the articles are summarized briefly.

[P1]: A construction that yields the best currently known lower bound for a Ramsey number, $R(5,9) > 120$, is given. The tabu search method used to obtain the construction is also described.

In Section 4 a brief survey of computational results on Ramsey numbers is given, and the methods used in [P1] for obtaining the lower bound $R(5,9) > 120$ are described.

[P2]: Based on the recent classification of resolvable $(12,4,3)$ designs by Morales and Velarde [38], the whist tournaments with at most twelve players are completely classified. This establishes the nonexistence of a directed whist tournament for twelve players—or, equivalently, the nonexistence of a resolvable perfect $(12,4,1)$ Mendelsohn design—and the nonexistence of a triplewhist tournament for twelve players, also independently noted by Ge and Lam [19].

[P3]: A correspondence between near resolutions of block designs and a particular class of codes is introduced. That correspondence is used to classify the near resolutions of $(13, 4, 3)$ designs by classifying the corresponding codes. Based on the classification of the near resolutions, the thirteen-player whist tournaments are classified. This classification establishes the nonexistence of a triplewhist tournament for thirteen players. The symmetries of the structures with a large automorphism group are examined.

In Section 5 the known results on the existence of whist tournaments, directed whist tournaments and triplewhist tournaments are summarized, and the methods used in [P2, P3] to classify the whist tournaments, directed whist tournaments and triplewhist tournaments of at most thirteen players are described. These classification results build on the classification of the resolvable or near resolvable $(v, 4, 3)$-designs; the method used in [P3] for classifying the near resolutions of $(13, 4, 3)$-designs is also sketched.

[P4]: A backtrack method with isomorph rejection for computing the maximum sum packing and the maximum strict sum packing in a cyclic group is presented. Volume bounds and extensive computations are used to determine for small k the order of the smallest cyclic group that admits a k-element sum packing or strict sum packing.

[P5]: As a natural extension of [P4], a canonicity test for subsets of an Abelian group is described and the maximum sum packing and maximum strict sum packing are computed for small Abelian groups. A bound that links the maximum possible density of a subset and the proportion of involutions in the Abelian group is given.

[P6]: The dual of the packing problem investigated in [P4] and [P5] is a covering problem. A backtrack method with isomorph rejection is presented for computing the minimum sum cover, strict sum cover, and difference cover of a finite Abelian group, and the minimum covers are computed for a number of small groups.

In Section 6, some earlier results on the minimum covers and maximum packings of finite Abelian groups are mentioned and the approach used in [P4, P5, P6] is described.

2.2 CONTRIBUTIONS OF THE AUTHOR

The author of the thesis is the sole author of [P1, P6] and has played a significant role in writing the remaining articles.

In [P4], the design of the algorithm is work by the co-authors. The author programmed one of the two implementations of the algorithm described and obtained the volume bound given. In [P5] the design of the canonicity test, the algorithms and their implementations, and the bounds obtained are work of the author.

In [P2, P3] the algorithm used to generate the whist tournaments given the related resolutions is work of the author, as is the algorithm used to eliminate isomorphic tournaments, and the analysis of the whist tournaments obtained. The design and implementation of the algorithm used for computing the resolutions of $(13, 4, 3)$-designs in [P3] is work of the coauthor.

3 COMPUTATIONAL METHODS

The computational methods most relevant for this thesis are briefly introduced in this section. We briefly describe tabu search, which is a heuristic search method, and backtrack search, the standard method of carrying out an exhaustive search. Two methods of carrying out isomorph rejection in a backtrack search are also described.

3.1 TABU SEARCH

Tabu search is a local search method. Local search methods are iterative heuristic optimization methods, where in every iteration the previous solution x_i is replaced by a new feasible solution x_{i+1} that is, in some sense, close to the previous solution. Solutions with a good value of the objective function f are usually preferred. The general idea is that if $x, x' \in X$ are close to each other, then $f(x)$ and $f(x')$ are likely not to differ very much. With that assumption, one may expect to find an optimal solution or at least a good solution in the vicinity of other good solutions by repeatedly making small improvements to a current solution. Local search methods have been very useful in solving optimization problems, particularly those of combinatorial nature. The book of Aarts and Lenstra [1] is a rich source of examples.

To formalize the notion of closeness of solutions, it is useful to define the neighborhood function: $N : X \mapsto 2^X$ maps each element $x \in X$ to its neighborhood $N(x)$, the set of those elements of X that are deemed to be close to x. In choosing the new solution to replace the current solution x, the objective function is evaluated for one or more solutions in $N(x)$ and one of those solutions may then replace x as the current solution.

Perhaps the simplest nontrivial form of local search is the steepest descent method: to minimize $f(x)$ with $x \in X$, an initial solution $x_0 \in X$ is chosen. Then x_{i+1} is repeatedly chosen from $N(x_i)$ such that $f(x_{i+1}) < f(x_i)$ and $f(x_{i+1}) \leq f(x')$ for all $x' \in N(x_i)$. The algorithm terminates when no such x_{i+1} exists. Clearly, the steepest descent method will terminate at the first local optimum it encounters, which renders it impractical for many problems, particularly those with a large number of local optima.

Tabu search is a local search method reminiscent of steepest descent: in every iteration, the current solution is replaced by the neighbor with the best objective function value—in the case of tabu search also when this does not improve the value of the objective function. To prevent the search from looping, the search is not allowed to undo recent changes to the current solution. In the following description, we roughly follow the notation in Glover's two-part article [22, 23].

To describe tabu search, we define a move: A move is a function $s : X(s) \mapsto X$, where $X(s)$ is the set of those $x \in X$ to which the move s may be applied. Let $S(x)$ denote the set of moves that can be applied to x. Now $N(x) = \{s(x) : s \in S(x)\}$. One may assume that every s is an injection, so that the inverse move s^{-1} exists. We associate the attribute $a(s, x)$ to the move s applied on x.

In tabu search, first an initial solution x_0 is chosen. Then x_i is repeatedly obtained from x_{i-1} by letting $x_i = s_i(x_{i-1})$ for some $s_i \in S(x_{i-1}) \setminus T_i$, where T_i is the set of tabu moves at iteration i. The move s_i is chosen such that $f(x_i)$ is minimized. We compute the attribute of the inverse of the chosen move, $a_i = a(s_i^{-1}, x_i)$. The set of tabu moves is defined by $T_i = \{s : s \in S(x_{i-1}), a(s, x_{i-1}) \in \{a_{i-\ell}, \ldots, a_{i-1}\}\}$, where the search parameter ℓ is the length of the tabu list.

The above is only a very basic form of tabu search. For additional information on the numerous tabu search variants that have been proposed in the context of various applications, we refer to Aarts and Lenstra [1].

In this thesis, tabu search is used in the article [P1]. An optimization algorithm such as tabu search may be used for searching by applying the penalty function method. In the penalty function method, constraints of the original problem are replaced with penalty terms in the objective function. If one only wants to find a feasible solution, one may choose the objective function to consist of penalty terms only, so that the resulting objective function measures, in a sense, how far from feasibility a solution is. If a solution is found for which the penalty terms, and hence the objective function, evaluate to zero, the solution is feasible and the search may stop.

3.2 BACKTRACK SEARCH

Backtrack search is a general method for exhaustive generation of combinatorial objects. On a general level, backtrack search is most conveniently described in terms of formal languages. Let Σ be a finite alphabet, let Σ^* be the set of strings over the alphabet Σ and let $L \subseteq \Sigma^*$. When started by calling `visit` on the empty string, Algorithm 1 outputs all $x \in L$ by recursively appending each possible symbol to the string in turn. In Algorithm 1, $C : \Sigma^* \mapsto 2^\Sigma$ maps a string x to the set of symbols that can be appended to x; to ascertain that all elements of L are obtained, it is necessary that $a \in C(x)$ whenever $xa\Sigma^* \cap L \neq \emptyset$. The algorithm terminates, if for some n_0 the algorithm visits no string longer than n_0; thus we require that $C(x) = \emptyset$ for all $x \in \Sigma^*$ with $|x| \geq n_0$.

def visit(x):
if $x \in L$ **then**
    output x
end if
$C \leftarrow C(x)$
for $a \in C$ **do**
    visit(xa)
end for

Algorithm 1: A backtrack algorithm

To generate some set of combinatorial objects $S \subseteq X$ by backtrack search, one may define a suitable language L over an alphabet Σ and introduce a function $f : L \mapsto X$ such that for all $s \in S$ there is some $x \in L$ for which $f(x) = s$. Then one may use Algorithm 1 to obtain L, apply f on L and test the elements of the resulting set for membership of S.

For example, to generate the k-subsets of $V = \{1, \ldots, v\}$, one may define $\Sigma = V$ and $L = \{\sigma_1\sigma_2\cdots\sigma_k : \sigma_i \in \Sigma, \sigma_i < \sigma_j \text{ for } i < j\}$. To limit the search, let $C(\sigma_1\sigma_2\cdots\sigma_i) = \{\sigma_i{+}1, \ldots, v\}$ for $i < k$ and $C(\sigma_1\sigma_2\cdots\sigma_k) = \emptyset$. Finally, to map the strings to k-subsets, define $f(\sigma_1\sigma_2\cdots\sigma_k) = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. The technique used in this example is essentially the technique used in [P1] for computing the number of cliques and independent sets with a given number of vertices. It also forms the basis of the methods in Section 3.3.

3.3 ISOMORPH-FREE GENERATION

In a set system $X = (V, B)$ the points in V are usually considered indistinguishable and nameless. While some structures are easy to describe, e.g., a tree on n vertices with no vertices of degree higher than 2 is the path on n vertices, in general it is easiest to represent B as lists of elements of V. For this purpose it is a practical necessity to introduce a labeling of the elements of V. Then, however, several different labelings may represent the same unlabeled structure: if $X_1 = (V_1, B_1)$, $X_2 = (V_2, B_2)$, and $B_1^f{=}B_2$ for some bijection $f : V_1 \mapsto V_2$, then X_1 and X_2 are isomorphic and represent the same unlabeled set system. Without loss of generality one may assume that $V = V_1 = V_2 = \{1, \ldots, |V|\}$ so that the functions f are permutations of V and therefore elements of the natural action of the symmetric group $\mathrm{S}_{|V|}$ acting on V. The natural action of $\mathrm{S}_{|V|}$ on V partitions the set systems with point set V into orbits such that the set systems in each orbit are isomorphic. In this manner, the problem of generating the unlabeled set systems of the desired type reduces to generating one labeled set system from each isomorphism class.

3.3.1 Orderly algorithms

Orderly algorithms are a comparatively simple method of isomorph-free generation pioneered by Faradžev [15] and Read [44]. Here we present a particularly simple orderly algorithm.

Suppose that the combinatorial objects being constructed may be represented as subsets S of a finite set X. Some additional structure may be attached to X; the set of permutations on X that preserve the additional structure, if any, is the automorphism group $\mathrm{Aut}(X)$ of X. For example, in constructing an n-vertex graph edge by edge, X could be the set of edges of K_n, the complete graph on n vertices, and $\mathrm{Aut}(X)$ could be the group of permutations that the symmetric group S_n acting on the vertices of the K_n induces on the edges of the K_n. As another example, in Section 6 our set X will be the set of elements in an Abelian group, and $\mathrm{Aut}(X)$ will consist of automorphisms of the Abelian group expressed as permutations of its elements.

Two subsets $S, T \subseteq X$ are considered isomorphic, if $S = T^\pi$ for some $\pi \in \mathrm{Aut}(X)$. Now $\mathrm{Aut}(X)$ partitions the subsets of X into orbits of isomorphic subsets with an equal number of elements. To induce an ordering on the elements in each orbit, we introduce an ordering of X and define the lexicographical order on k-subsets of X as follows: for two distinct k-subsets $S, T \subset X$, we have $S < T$ if the least element that is a member of only one of S, T is a member of S. A bit more formally, $S < T$ if there exists some $x \in X$ such that $x \in S$, $x \notin T$, and $y \in S$ iff $y \in T$ for all $y \in X$ for which $y < x$. The lexicographical minimum of each orbit is said to be the canonical representative of the orbit.

We describe an orderly algorithm in terms of Algorithm 1 in Section 3.2. We let $\Sigma = X$ and $f(\sigma_1 \cdots \sigma_k) = \{\sigma_1, \ldots, \sigma_k\}$. The set L will consist of the strings $x = \sigma_1 \cdots \sigma_k$ where $\sigma_i \in \Sigma$, $\sigma_i < \sigma_j$ for $i < j$, and $f(x)$ is canonical. We let $C(\sigma_1 \cdots \sigma_k) = \{\sigma : \sigma_k < \sigma \text{ and } \sigma_1 \cdots \sigma_k \sigma \text{ is canonical}\}$. With these definitions, Algorithm 1 visits and outputs the canonical subsets of X.

This is essentially the method used in [P3] for generating the gap codes that correspond to near resolutions of $(13, 4, 3)$-designs and in [P4, P5, P6] for searching the relevant subsets of Abelian groups.

A potential weakness of this method is that testing a subset for canonicity may be computationally unnecessarily expensive. If X is the set of edges of a graph, ordered colexicographically so that the edge ij (where $i < j$) precedes the edge kl (where $k < l$) if $j < l$ or if $j = l$ and $i < k$, then it is straightforward to determine the size of the maximum clique from the canonical representative of an orbit. However, computing the maximum clique of a graph is NP-hard, while testing two graphs for isomorphism is generally not believed to be NP-complete. Designing an efficient canonicity test when $\mathrm{Aut}(X)$ is large appears difficult in practice.

3.3.2 The canonical augmentation method

Another method of isomorph rejection in combinatorial generation is described by McKay [34]. In contrast to orderly algorithms, where usually every structure along the construction path is required to be canonical, in the canonical augmentation method it is required that the steps taken to construct each structure are canonical. Here we only sketch the main ideas; for a rather more formal description, examples and literature references we refer the reader to McKay [34].

Let L be a set of combinatorial structures and let G be a permutation group acting on L. The elements of L are called labelled objects, and the orbits of L under G are called unlabelled objects. The set of unlabelled objects is denoted with U. Every element $x \in L$ has an order $o(x) \in \mathbb{N}$, and $o(x) = o(x^g)$ for all $g \in G$, which allows us to define $o(x)$ for $x \in U$ in the obvious way. Let L_i denote the set $\{l : l \in L, o(l) = i\}$ and U_i the set $\{u : u \in U, o(u) = i\}$.

We define an extension function $f : U \mapsto 2^U$ such that for $u \in U$ the set $f(u)$ consists of the structures that may be generated from u in one step. For all $w \in f(u)$ we must have $o(w) > o(u)$. We also define a parent function $p : U \setminus U_0 \mapsto U$ such that $o(p(u)) < o(u)$ for all $u \in U \setminus U_0$. We further require that $u \in f(p(u))$ for $u \in U \setminus U_0$. The idea is that from every object encountered in the recursive search, the search generates new objects of higher order, but the new objects are rejected unless they have been generated by canonically augmenting their proper parent defined by the function p. When the `visit` function of Algorithm 2 is called for every $u \in U_0$, every

$u \in U$ with $o(u) \leq n$ is visited: for every element $u \in U$ there is some integer i such that $p^i(u) \in U_0$; the sequence $(p^i(u), p^{i-1}(u), \ldots, p(u), u)$ is the only sequence of structures that may lead to u, and it will.

def visit(u, n):
if $o(u) < n$ **then**
  for $u' \in f(u)$ **do**
    if $p(u') = u$ **then**
      visit(u', n)
    end if
  end for
end if

Algorithm 2: Backtrack with canonical augmentation

Our brief sketch of Algorithm 2 appears deceptively simple, as the sketch operates with unlabelled structures, while in practice it is necessary to handle labelled structures. A number of refinements is necessary. The parent function $p : L \setminus L_0 \mapsto L$ for labelled objects must be invariant such that for $x' = x^g$, where $g \in G$ there is a $h \in G$ such that $p(x') = p(x)^h$. The extension function $f : L \mapsto 2^L$ must contain a labelled structure from each orbit u for which x and $p(u)$ are isomorphic. To avoid duplicate structures and duplicating work, it is necessary to prune $f(x)$ so that it contains at most one representative from each isomorphism class; this pruning can be done before or after eliminating the extensions with the wrong parent.

Choosing a suitable p is crucial for the efficiency of the search. We are no longer confined by the limitations of lexicographic extremality as in using orderly algorithms. This may allow for much more efficient canonicity testing than in the case of orderly algorithms. For example, suppose that the search constructs graphs of some kind. Then it may be convenient to define p by defining a method for choosing a vertex and removing it. Depending on the class of graphs under consideration, it may be possible to select a cheap vertex invariant that uniquely identifies a vertex in the vast majority of graphs encountered during the search. One may then define p as computing that invariant and removing the identified vertex, or, only if the chosen invariant does not uniquely identify a vertex to be removed, computing a canonical labeling of the graph and using the canonical labeling as a basis for selecting the vertex to be removed.

4 RAMSEY NUMBERS

In this Section, we define Ramsey numbers, review computational methods of obtaining their values or bounds — particularly lower bounds — for them, and describe the method used to obtain the lower bound $R(5, 9) > 120$ [P1].

4.1 BACKGROUND AND DEFINITIONS

Loosely speaking, Ramsey theory shows us that complete disorder is impossible. Van der Waerden has shown that if the positive integers are partitioned into a finite number of sets, then an arbitrarily long arithmetic progression can be found in one of the sets; Schur [46] showed that if the positive integers are partitioned into a finite number of sets, then one of the sets contains x, y, z such that $x + y = z$; and Ramsey [43] showed that for a given k, every graph with sufficiently many vertices contains a clique or an independent set of k vertices.

In his paper, Ramsey proved the following theorem, where subscripts of the sets Γ and Δ denote their cardinality:

> THEOREM B. *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.*

Ramsey then gives an equivalent theorem and proves it. We define $m_0(r, n, \mu)$ as the least m_0 that satisfies Ramsey's Theorem B. Next, we give a contemporary definition.

Definition 1 *The Ramsey number $R(k_1, k_2, \ldots, k_\mu; t)$ is the least integer n such that when the t-combinations of an n-element set V are partitioned into μ sets E_i, $1 \leq i \leq \mu$, then for some $1 \leq i \leq \mu$ there is a k_i-element subset $V' \subseteq V$ such that all t-combinations of V' are in E_i. For convenience, define $R(k_1, k_2, \ldots, k_\mu) = R(k_1, k_2, \ldots, k_\mu; 2)$.*

Note that Ramsey's $m_0(r, n, \mu) = R(\overbrace{n, n, \ldots, n}^{\mu \text{ times}}; r)$. Clearly, Definition 1 is more general than the definition arising from Ramsey's Theorem B in that it allows unequal n_i.

Ramsey numbers are notoriously hard to determine and exact values are only known for small parameter values. Radziszowski's survey [42] is the definitive source for the values and bounds of specific Ramsey numbers. For $t > 2$ the only known value is $R(4, 4; 3) = 13$, and for $t = 2$ and $\mu > 2$ the only known value is $R(3, 3, 3) = 17$.

Most of the research on specific Ramsey numbers has been on the case where $t = 2$ and $\mu = 2$. Despite this only a handful of values are known: $R(3, k)$ is known for $k \leq 9$, and $R(4, k)$ is known for $k \leq 5$. Already $R(5, 5)$ is unknown; the best known bounds are $43 \leq R(5, 5) \leq 49$. Some

k \ l	3	4	5	6	7	8	9	10
3	6	9	14	18	23	28	36	40 43
4		18	25	35 41	49 61	56 84	69 115	92 149
5			43 49	58 87	80 143	101 216	121 316	141 442
6				102 165	111 198	127 495	153 780	177 1171
7					205 540	216 1031	 1713	 2826
8						282 1870	 3583	316 6090
9							565 6588	580 12677
10								798 23556

Table 4.1: Bounds and known values for small Ramsey numbers

best currently known bounds are summarized in Table 4.1, based on Radziszowski's survey [42] but taking into account two lower bounds by Harborth and Krause [28]. Centered entries represent exact values known, high entries represent lower bounds and low entries represent upper bounds.

It is natural to view determining $R(k,l)$ as a graph-theoretical problem: What is the least n such that when each edge of an n-vertex complete graph K_n is colored with one of two colors, the resulting coloring will contain a K_k in the first color or a K_l in the second color? An immediate generalization is to replace K_k and K_l with arbitrary graphs G and H: What is the least n such that when each edge of K_n is colored with one of two colors, the resulting coloring will contain a G in the first color or a H in the second color? Radziszowski quotes many known results for various small G and H. A further generalization is the Arrowing problem: Given three graphs F, G, and H, is it true that every two-color edge-coloring of F contains a G in the first color or a H in the second color as a subgraph?

Computing Ramsey numbers appears to be computationally complex. To begin with, one should note that even given k and a graph, it is an NP-complete decision problem to find out whether the graph contains a clique of k vertices. An obvious method of testing whether $R(k,k) > v$ would be to test all v-vertex graphs for the existence of a k-vertex clique and a k-vertex independent set. Indeed, given n encoded in unary and k, a nondeterministic Turing machine with an NP oracle can test in polynomial time whether $R(k,k) > n$: guess nondeterministically an n-vertex graph G, and ask the oracle whether G contains a k-vertex clique or a k-vertex independent set. If not, accept: there is a graph that proves that $R(k,k) > n$. Thus, with this encoding of the input, testing whether $R(k,k) > n$ is in the complexity class $\mathrm{NP}^{\mathrm{NP}}$. If n and k are encoded in binary, the input length is logarithmic in n, and no machine can even construct an n-vertex graph in polynomial time.

It is not easy to obtain a lower bound for the complexity of deciding whether $R(k,k) > n$. A typical technique for demonstrating a lower bound for the complexity of a decision problem is to reduce a problem that is known to be complete for some computational complexity class to it. In this case, it would be necessary to encode instances of the other difficult problem as instances of the Ramsey problem. However, it seems difficult to encode any meaningful problem into just two integers. It appears necessary to consider a more general problem, and, indeed, Schaefer [45] has shown that Arrowing is complete for the complexity class $\mathrm{coNP}^{\mathrm{NP}}$.

There are several results on the asymptotic bounds for $R(k,k)$ and $R(k,l)$— Radziszowski [42] cites several sources —but we restrict our attention to the values of Ramsey numbers $R(k,l)$ with fairly small k and l. To facilitate discussion, we introduce the following definitions:

Definition 2 A (k,l)*-graph is a graph with no* k*-vertex clique and no* l*-vertex independent set.*

Definition 3 A (k,l,n)*-graph is a* (k,l)*-graph with* n *vertices.*

4.2 EASILY COMPUTABLE VALUES

Clearly, $R(k,l) = R(l,k)$, and $R(2,k) = k$. For some k, l, let G be a graph with at least $n(k,l) = R(k-1,l) + R(k,l-1)$ vertices. By the pigeonhole principle, any vertex v of G must have $R(k-1,l)$ neighbors or $R(k,l-1)$ non-neighbors; in either case there must be a k-vertex clique or an l-vertex independent set in G. Thus $R(k,l) \leq R(k-1,l) + R(k,l-1)$. Equality cannot hold here if $R(k-1,l)$ and $R(k,l-1)$ are both even, as that would imply the existence of a graph with $n-1$, an odd number, of vertices, each of which would have $R(k-1,l)-1$, an odd number, of neighbors. However, no graph with an odd number of vertices of odd degree can exist. These elementary observations yield the sharp bounds $R(3,3) \leq 6$, $R(3,4) \leq 9$, $R(3,5) \leq 14$, and $R(4,4) \leq 18$. For the lower bounds, one may consider the $(3,3,5)$-graph C_5 and the $(3,4,8)$-graph obtainable by connecting the antipodal vertices of an octagon with edges. We describe a $(3,5,13)$-graph and a $(4,4,17)$-graph below in the context of lower bounds obtainable from finite field constructions.

4.3 COMPUTED KNOWN VALUES

When k and l increase, computing the value of $R(k,l)$ becomes rapidly more difficult. In this section we briefly describe some methods of obtaining a sharp upper bound for $R(k,l)$ for certain values of k and l; together with a suitable graph G with no k-vertex clique and no l-vertex independent set these bounds yield the value of $R(k,l)$.

The upper bounds for $R(3,6) = 18$, $R(3,7) = 23$, $R(3,8) = 28$, $R(3,9) = 36$ and $R(4,5) = 25$ have been determined by Kéry [30], Graver and Yackel [25], McKay and Zhang [37], Grinstead and Roberts [27], and McKay and Radziszowski [35], respectively.

A key idea in the lengthy paper of Graver and Yackel is preferring a vertex. Given a graph G, they choose a vertex v and consider the subgraph induced by the neighbors of v and the subgraph induced by the non-neighbors of v. They present a large number of lemmas involving, among others, the degree of v and the number of edges in these two induced subgraphs. Having developed suitable machinery, they give a long case by case exhaustive proof that no $(3, 7, 23)$-graph exists. They also present a $(3, 7, 22)$-graph.

In determining $R(3, 8)$, McKay and Zhang [37] use a form of the canonical augmentation method. At each backtracking step, a $(3, t-1, n-\delta-1)$-graph is extended in every possible manner to a $(3, t, n)$-graph with minimum degree δ by adding a new vertex v and its δ neighbors. Characteristically for the canonical augmentation method, it is required that every graph G be constructed from its canonical parent. In this case the canonical parent of a graph is obtained by computing its canonical labelling and removing the first vertex of the minimum degree. Certain lemmas involving the minimum degree of various classes of graphs necessary in the computation allow a significant reduction in the number of graphs that need to be considered to finally find that $R(3, 8) = 28$.

McKay and Radziszowski [35] determined that $R(4, 5) = 25$ by constructing a family of $(4, 5, 24)$-graphs such that any $(4, 5, 25)$-graph would have to be a one-vertex extension of at least one of the graphs in the family. Let v be a vertex of a $(4, 5, 24)$-graph. The graph G induced by the neighbors of v is a $(3, 5, \delta)$-graph, where δ is the degree of v. Similarly, the graph H induced by the non-neighbors of v is a $(4, 4, 24-\delta)$-graph. Thus, in order to obtain the set of $(4, 5, 24)$-graphs that can be extended to $(4, 5, 25)$-graphs it suffices to try all pairs (G, H) where G is a $(3, 5, \delta)$-graph and H is a $(4, 4, 24-\delta)$-graph and to glue them together in every possible way by adding edges between vertices of G and H. McKay and Radziszowski go on to present techniques to reduce the number of gluings that need to be performed and they present an impressive array of computational techniques to compute the gluings sufficiently efficiently.

4.4 LOWER BOUNDS FOR SPECIFIC RAMSEY NUMBERS

In this section we summarize known results for lower bounds of Ramsey numbers $R(k, l)$ for some specific k and l.

4.4.1 Finite field techniques

Some of the best bounds have been found by constructing graphs according to the structure of a finite field. Greenwood and Gleason [26] show that $R(4, 4) > 17$ by labeling the vertices of a 17-vertex graph with the elements of $\mathbb{Z}_{17}$ and introducing the edges uv when $u-v$ is a quadratic residue in $\mathbb{Z}_{17}$. They showed similarly that $R(3, 5) > 13$ by using cubic residues in $\mathbb{Z}_{13}$, and also obtained the lower bounds $R(3, 3, 3) > 16$ and $R(3, 3, 3, 3) > 41$, only the last one of which is not sharp. In a similar fashion, Burling and Reyner [10] label the vertices of a K_p with the elements of $\mathbb{Z}_p$ with $p = 4k+1$ prime, introduce the edges uv when $u-v$ is a quadratic residue in $\mathbb{Z}_p$, and compute the size of the largest clique in the resulting graph. For $p = 101$ and $p = 281$ this results in graphs that yield the best currently known bounds $R(6,6) > 101$ and $R(8,8) > 281$.

Mathon [32] and Shearer [48] independently show that if the maximum clique in the construction by Burling and Reyner is of order k, which implies that $R(k+1,k+1) > p$, then also $R(k+2,k+2) > 2p+2$. This bound is obtained by applying the following doubling construction to a p-vertex graph of the type constructed by Burling and Reyner: Let G be a graph. Construct a new graph G' by introducing the vertices v and v' for every v in G, and add two more vertices w and w'. Let G' have the edges uv and $u'v'$ for every edge uv in G, and uv' for every uv not in G. Finally, add the edges vw and $v'w'$ for all v in G. The best currently known bounds $R(7,7) > 204$, $R(9,9) > 564$, and $R(10,10) > 797$ can be obtained in this manner.

4.4.2 Computational search techniques

Even attempting to obtain a good lower bound for $R(k,l)$ by finding a (k,l,n) graph is a rather discouraging task for even moderately large k and l. Testing all the $2^{n(n-1)/2}$ labelled n-vertex graphs is a daunting prospect for even comparatively small n. Although only the most naïve of brute force searches would explore the whole search space, it seems necessary to limit the search space radically.

An important idea in reducing the search space is that the cliques and independent sets in a graph are phenomena that are, in a sense, local to the vertices involved. One could limit the search so that the graph looks the same regardless of from which of its vertices it is viewed. If one vertex is not involved in too large a clique or too large an independent set, then neither is any of the other vertices.

More formally, experimentally it seems to make good sense to restrict the search to graphs with a vertex-transitive automorphism group. Cyclic groups have been quite popular. The cyclic group C_n partitions the edges of K_n into $\lfloor n/2 \rfloor$ orbits, and the search space is reduced to a more manageable $2^{\lfloor n/2 \rfloor}$. Let $C(k,l)$ denote the least integer n such that there is no cyclic (k,l,n')-graph for any $n' \geq n$. Clearly, $C(k,l) \leq R(k,l)$, but in general equality does not hold. For example, $C(5,5) = 42$, but McKay and Radziszowski [36] have found 656 non-cyclic $(5,5,42)$-graphs. Recently, Harborth and Krause [28] have computed $C(k,l)$ for various small k and l. Some of their results are summarized in Table 4.2, where the values of $C(k,l)$ are laid out above the values of $R(k,l)$, and a dash indicates that the value given is merely the best lower bound known. With few exceptions, for $R(k,l) < 20$ we have $C(k,l) = R(k,l)$, for $20 \leq R(k,l) < 60$ we know that $C(k,l) < R(k,l)$, and for $R(k,l) > 60$ it is not known whether $C(k,l) = R(k,l)$, but many of the best known bounds result from cyclic constructions.

The best known lower bound $R(5,7) \geq 80$ is due to Calkin, Erdős, and Tovey [11], who report having carried out an implicit enumeration of cyclic graphs of prime order. They remark that the existence of a cyclic (k,l,n)-graph does not imply the existence of a cyclic (k,l,n')-graph for $n' < n$. Indeed, there is a cyclic $(4,5,24)$-graph, but no cyclic $(4,5,23)$-graph. Calkin, Erdős, and Tovey also reported the lower bound $R(5,9) > 114$ and bounds

k l	4	5	6	7	8	9	10	11	12
3	9	14	17	22	27	36	39	46	49
	9	14	18	23	28	36	40–	46–	52–
4	18	25	34	47	52	69	92	96	
	18	25	35–	49–	56–	69–	92–	96–	
5		42	57	80	101				
		43–	58–	80–	101–				
6			102						
			102–						

Table 4.2: $C(k,l)$ and lower bounds for $R(k,l)$

on $R(4,12)$ and $R(4,15)$.

The lower bounds $R(3,13) \geq 59$, $R(4,10) \geq 80$, $R(4,11) \geq 96$, and $R(5,8) \geq 95$ are from Piwakowski [41]. These bounds were obtained by applying tabu search and restricting the search to graphs with cyclic symmetry, except for the first one, where the required symmetry may be described as the 29-element subgroup of C_{58} acting on the 58 vertices in the natural way.

Exoo [14] gives a number of best known lower bounds by a curious local search method. The neighborhood of a solution again consists of those graphs that can be obtained from the current graph by adding or removing an edge orbit, and the objective function is again the number of k-vertex cliques plus the number of l-vertex independent sets. In every iteration, the objective function is computed for all neighbors of the current solution, and the values of the objective function for solutions in the neighborhood of the current solution are collected. One of the T smallest values of the objective function is chosen, and a neighbor for which the objective function attains that value is chosen as the next solution. The best known bound $R(6,9) > 152$ results from a cyclic graph. The best known bound $R(3,12) > 51$ results from a graph with the 17-element subgroup of C_{51} as the prescribed automorphism group. Also, Exoo constructed $(5,t,4n)$-graphs from cyclic $(3,t-1,n)$-graphs by a building block construction somewhat similar to that used by Mathon [32]; the best known lower bounds $R(5,10) > 140$, $R(5,11) > 152$, $R(5,12) > 180$, $R(5,13) > 192$, $R(5,14) > 220$, and $R(5,15) > 236$ were obtained in this manner.

4.5 THE LOWER BOUND R(5,9)>120

The best known lower bound $R(5,9) > 120$ results from the construction of a cyclic graph [P1]. It is obtained by a tabu search very similar to the one by Piwakowski [41]. The objective function is the number of k-vertex cliques plus the number of l-vertex independent sets, the neighbors of a solution may be obtained from the solution by adding or removing an edge orbit, and a neighbor is tabu if the difference between the current solution and the neighbor is an edge orbit that has been added or removed within the last t iterations, where t is the length of the tabu list; the graph was found after a lengthy computation with $t = 12$.

It is crucial for the speed of the search to be able to compute the number of k-cliques in the current solution efficiently. In tabu search, it suffices to compute the change in the value of the objective function caused by each possible move; in our tabu search, the number of k-vertex cliques and l-vertex independent sets in the graph is computed incrementally. As cliques and independent sets as well as adding orbits and removing them are handled in essentially the same manner, we only describe the procedure used for counting the cliques introduced by adding an edge orbit.

When an edge orbit E_i is added to the graph, the number of k-cliques introduced to the graph is computed and added to the total number of k-cliques. Consider the cliques with at least one edge in E_i. The action of G partitions the cliques into orbits. From each orbit, choose a clique $K = (V, E)$ and let $r = |E \cap E_i|$. Now, the edges in E_i occur in the cliques in K^G a total of $r \left|K^G\right|$ times. By symmetry, each $e \in E_i$ occurs in $r \left|K^G\right| / |E_i|$ cliques in K^G. Therefore, to compute the number of k-cliques with at least one edge in E_i one may choose an arbitrary edge $e \in E_i$ and compute the sum of $|E_i| / r$ over the cliques that include the edge e. This is equivalent to counting the cliques of order $(k-2)$ in the subgraph of G induced by the vertices adjacent to both endpoints of e.

The cliques are counted by a straightforward backtracking algorithm of the type described in Section 3.2. The vertices are ordered, and as long as the current clique has fewer than k vertices, the choice set is the set of vertices adjacent to all vertices in the current clique that come after the last vertex in the current clique.

As suggestions for possible improvement one may note that in this search, the algorithm repeatedly computes the number of k-vertex cliques in a graph. However, computing the number of k-vertex cliques in a graph is a #P-complete problem, which might suggest that the approach chosen is computationally unnecessarily hard. It should be worthwhile to try maximizing v such that the graph induced by the vertices $\{1, \ldots, v\}$ is a (k, l)-graph. Exoo [13] uses successfully a very similar objective function for the related problem of determining lower bounds for Schur numbers.

A good implementation of clique search is very important for the speed of the algorithm, and it would be interesting to test whether other clique algorithms such as Östergård's clique algorithm [39], which is based on a dynamic programming technique, would perform well in this context.

5 WHIST TOURNAMENTS

In this section, we define various types of whist tournaments, review methods of constructing them for different numbers of players, and summarize the methods used to classify the whist tournaments, directed whist tournaments and triplewhist tournaments with up to thirteen players [P2, P3].

5.1 DEFINITIONS AND EXISTENCE

Definition 4 *A whist tournament* $\mathrm{Wh}(v)$ *for* v *players is a schedule of games, each involving two players playing against two others, such that*

1. *in each round, the players are divided into* $\lfloor v/4 \rfloor$ *games with at most one player left over,*
2. *each player partners every other player exactly once, and*
3. *each player opposes every other player exactly twice.*

From the first condition, v must be of the form $v = 4k$ or $v = 4k + 1$ for some integer k. When $v = 4k$, there are $r = v - 1$ rounds, and when $v = 4k + 1$, there are $r = v$ rounds.

It is convenient to represent games by using 4-tuples of the form (n, e, s, w). One may interpret the tuple as listing, in order, the players sitting on the north, east, south, and west side of the playing table. The players sitting opposite each other are partners.

According to Anderson [4], the mathematical study of whist tournaments was started in the 1890s by E. H. Moore. Anderson reports that R. M. Wilson, R. D. Baker, and H. Hanani established the existence of $\mathrm{Wh}(4k)$ and $\mathrm{Wh}(4k+1)$ for all positive integers k in the 1970s. Chapter 13 of Anderson's book on combinatorial designs [3] is a proof of this result. The proof is long, so we only indicate the basic structure of the argument: For a number of small v, a $\mathrm{Wh}(v)$ is constructed either by an algebraic method or by giving a construction. By presenting a number of product theorems and several kinds of combinatorial designs such as self-orthogonal Latin squares, group divisible designs, and spouse-avoiding mixed doubles round robin tournaments, Anderson then shows that the existence of the constructed $\mathrm{Wh}(v)$ with small v implies the existence of $\mathrm{Wh}(v)$ for all v of the form $v = 4k$ or $v = 4k + 1$.

Directed whist tournaments are a special case of whist tournaments. Instead of requiring only that each player meet every other player twice as an opponent, it is required that every other player is met once as the left-hand opponent and once as the right-hand opponent.

Definition 5 *A directed whist tournament* $\mathrm{DWh}(v)$ *for* v *players is a schedule of games, each involving two players against two others, such that*

1. *in each round, the players are divided into* $\lfloor v/4 \rfloor$ *games with at most one player left over,*

16	20	24	32	36	44	48	52	56
64	68	76	84	88	92	96	104	108
116	124	132	148	152	156	172	184	188

Table 5.1: The v for which Zhang [53] leaves existence of $\mathrm{DWh}(v)$ open

2. *each player partners every other player exactly once,*
3. *each player opposes every other player as his left-hand-opponent exactly once, and*
4. *each player opposes every other player as his right-hand-opponent exactly once.*

The existence of a $\mathrm{DWh}(v)$ is equivalent to the existence of a $(v, 4, 1)$-RPMD, a resolvable perfect Mendelsohn design. Bennett and Zhang [7] settle the remaining open cases to show the existence of $\mathrm{DWh}(4k + 1)$ for all $k \geq 1$, and Zhang [54] presents a number of combinatorial constructions to show that a $(v, 4, 1)$-RPMD exists for all $v = 4k$ except for $v = 4, 8$, and possibly except for 49 other values of v between 12 and 336. By classifying the whist tournaments with up to twelve players Haanpää and Östergård [P2] find that no $\mathrm{DWh}(12)$ exists, and in a recent article [53] Zhang narrows the list of potential exceptions down to the 27 values in Table 5.1.

Triplewhist tournaments are another special type of whist tournaments. In a whist game (n, e, s, w) the pairs $\{n, e\}$ and $\{s, w\}$ are said to be opponents of the first kind and the pairs $\{n, w\}$ and $\{s, e\}$ are said to be opponents of the second kind. Now, in addition to requiring that each player meet every other player twice as an opponent, it is required that every other player is met once as an opponent of the first kind and once as an opponent of the second kind.

Definition 6 *A triplewhist tournament* $\mathrm{TWh}(v)$ *for* v *players is a schedule of games, each involving two players against two others, such that*

1. *in each round, the players are divided into* $\lfloor v/4 \rfloor$ *games with at most one player left over,*
2. *each player partners every other player exactly once,*
3. *each player opposes every other player exactly once as an opponent of the first kind, and*
4. *each player opposes every other player exactly once as an opponent of the second kind.*

The existence of a $\mathrm{TWh}(v)$ remains open only for $v = 17$: a $\mathrm{TWh}(v)$ exists for all $v \equiv 0, 1 \bmod 4$ except for $v \in \{5, 9, 12, 13\}$ and possibly $v = 17$. In the literature [2, 31], R. D. Baker is credited with showing in his doctoral thesis [6] the existence of $\mathrm{TWh}(v)$ for $v \in \{4, 8, 16, 24\}$, as well as for all sufficiently large $v \equiv 1 \bmod 4$ and $v \equiv 0, 4, 12 \bmod 16$. Lu and Zhu

[31] show that there exists a $\mathrm{TWh}(v)$ for all $v \equiv 0, 1 \bmod 4$ with the definite exceptions of $v \in \{5, 9\}$ and the fifteen potential exceptions $\{12, 56\} \cup \{13, 17, 45, 57, 65, 69, 77, 85, 93, 117, 129, 133, 153\}$. In their article on $\mathbb{Z}$-cyclic $\mathrm{TWh}(v)$ Ge and Zhu [21] give a construction for a TWh(133), and Haanpää and Östergård [P2] find that no TWh(12) exists. Ge and Lam [19] present a number of $\mathbb{Z}$-cyclic $\mathrm{TWh}(v)$, including a TWh(56) and a TWh(45), and find independently that no TWh(12) exists. Haanpää and Kaski find that no TWh(13) exists, and Abel and Ge [2] present constructions for the remaining open cases except for $v = 17$.

Definition 7 *A* $\mathrm{Wh}(v)$ *is said to have the three person property, if no two games in the tournament have three or more players in common.*

Ge and Lam [18] show that a $\mathrm{Wh}(v)$ with the three person property exists for all $v \equiv 0, 1$ with $v \geq 8$ except for $v = 12$. Given the $\mathrm{Wh}(v)$, $\mathrm{DWh}(v)$, and $\mathrm{TWh}(v)$ for $v \leq 13$ obtained as a result of the enumerations in [P2, P3] it would be straightforward to test them for the three person property. However, such a test has not been carried out.

Definition 8 *A* $\mathrm{Wh}(v)$ *(*$\mathrm{DWh}(v)$*,* $\mathrm{TWh}(v)$*) is said to be* $\mathbb{Z}$*-cyclic, if it can be expressed as an orbit of an initial round under the action of the cyclic group* C_r *acting in the natural way on* r *of the players, where* r *is the number of rounds in the tournament.*

In the case $v = 4k$, the $r = v - 1$ players permuted by the cyclic group are labelled with the elements of $\mathbb{Z}_r$. The remaining player is a fixed point under the action of the cyclic group and is conventionally denoted with ∞. By convention, ∞ partners 0 in the first round. In the case $v = 4k + 1$, the $r = v$ players are labelled with the elements of $\mathbb{Z}_r$. By convention, 0 sits out in the first round. In either case the successive rounds are obtained from the preceding round by adding 1 modulo r to the number of every non-∞ player.

Example 9 *A* $\mathbb{Z}$*-cyclic* $\mathrm{Wh}(8)$*:*

$$\begin{array}{lll}
\textit{Round 1} & (\infty, 4, 0, 5) & (1, 2, 3, 6) \\
\textit{Round 2} & (\infty, 5, 1, 6) & (2, 3, 4, 0) \\
\vdots & & \\
\textit{Round 7} & (\infty, 3, 6, 4) & (0, 1, 2, 5)
\end{array}$$

Example 10 *A* $\mathbb{Z}$*-cyclic* $\mathrm{Wh}(13)$*:*

$$\begin{array}{llll}
\textit{Round 1} & (1, 8, 12, 5) & (2, 3, 11, 10) & (4, 6, 9, 7) \\
\textit{Round 2} & (2, 9, 0, 6) & (3, 4, 12, 11) & (5, 7, 10, 8) \\
\vdots & & & \\
\textit{Round 13} & (0, 7, 11, 4) & (1, 2, 10, 9) & (3, 5, 8, 6)
\end{array}$$

The existence of $\mathbb{Z}$-cyclic whist tournaments of various kinds has been considered by a number of authors, but it is a separate question slightly beyond the scope of this work. For more information, we refer the reader to

Ge and Ling [20], who present a new construction and cite several previous results.

It is of more direct interest to us that whist tournaments of various kinds have in many cases been shown to exist by constructing a $\mathbb{Z}$-cyclic whist tournament. For $1 \le k \le 10$, Anderson [4] quotes examples of $\mathbb{Z}$-cyclic Wh$(4k)$, some of which date back to the late 19th century. As a more recent example, Abel and Ge [2] show the existence of a TWh(v) for $v \in \{57, 65, 69, 77, 85, 93, 117, 129\}$ by presenting in each case a $\mathbb{Z}$-cyclic TWh(v).

5.2 CLASSIFYING THE RESOLUTIONS OF (13,4,3)-NRBIBDS

In this section we sketch the method used to classify the nonisomorphic $(13, 4, 3)$-NRBIBDs and their resolutions. For more specific details we refer the reader to the relevant article [P3].

Definition 11 *A t-(v, k, λ) balanced incomplete block design (a t-(v, k, λ) BIBD) is a collection of k-element subsets, called blocks, of a v-element point set such that every t-element subset of the point set is a subset of exactly λ blocks.*

Definition 12 *A parallel class of a BIBD is a set of disjoint blocks whose union is the point set. A near parallel class of a BIBD is a set of disjoint blocks whose union is the point set minus one point.*

Definition 13 *A (near) resolution of a t-(v, k, λ) BIBD is a partition of the blocks of a t-(v, k, λ) BIBD into (near) parallel classes.*

Definition 14 *A (near) resolvable t-(v, k, λ) balanced incomplete block design (a t-(v, k, λ) (N)RBIBD) is a t-(v, k, λ) BIBD that has a (near) resolution.*

In Definitions 11 to 14, if $t = 2$ it may be omitted from the notation, e.g., a (v, k, λ) BIBD is a 2-(v, k, λ) BIBD.

It is straightforward to verify that by replacing the games in a whist tournament by blocks of four players, one obtains a resolution or a near resolution of a $(v, 4, 3)$ BIBD. Thus the v-player whist tournaments may be classified by constructing every possible whist tournament on every nonisomorphic resolution or near resolution of a $(v, 4, 3)$ BIBD and eliminating isomorphic whist tournaments. This is the approach chosen by Haanpää and Östergård [P2] and Haanpää and Kaski [P3] for classifying whist tournaments with up to thirteen players. For $v \in \{4, 5, 8, 9\}$ generation of the resolutions presents no particular problems, and Morales and Velarde [38] classified the five $(12, 4, 3)$ RBIBDs. Classifying the resolutions of $(13, 4, 3)$ BIBDs is considerably more laborious.

In a near resolution of a $(13, 4, 3)$ NRBIBD there are 13 near parallel classes. We number the blocks within each near parallel class with the integers 0, 1, and 2. Now we can form a 13 by 13 array, whose rows correspond to the points and whose columns correspond to the near parallel classes. At the intersection of row i and column j we have a number that indicates the block of near parallel class j that contains the point i, or $\star$ if the point i is absent from the near parallel class j.

Example 15 *A matrix representation of a resolution of a* $(13, 4, 3)$ *BIBD.*

$$\begin{array}{ccccccccccccc}
* & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 \\
0 & * & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 \\
1 & 0 & * & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 \\
1 & 1 & 0 & * & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 \\
2 & 1 & 1 & 0 & * & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 \\
0 & 2 & 1 & 1 & 0 & * & 0 & 1 & 1 & 2 & 0 & 2 & 2 \\
2 & 0 & 2 & 1 & 1 & 0 & * & 0 & 1 & 1 & 2 & 0 & 2 \\
2 & 2 & 0 & 2 & 1 & 1 & 0 & * & 0 & 1 & 1 & 2 & 0 \\
0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 & * & 0 & 1 & 1 & 2 \\
2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 & * & 0 & 1 & 1 \\
1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 & * & 0 & 1 \\
1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 & * & 0 \\
0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 2 & 1 & 1 & 0 & *
\end{array}$$

To enumerate the near resolutions of $(13, 4, 3)$ NRBIBDs, we enumerate the matrices of the type illustrated in Example 15 that satisfy the necessary requirements. An obvious way of constructing near resolutions is to construct them one near parallel class at a time, or, in the matrix representation, to construct them a column at a time. Instead, we construct the matrix row by row. In this manner, the rows of the matrix may be though of as codewords of a particular equidistant q-ary code. In defining the distance between two codewords, the entries with value `*`, which we call gaps, require special attention; we define gap codes as codes whose words may contain gaps. Then we describe the correspondence between near resolvable designs and gap codes by extending the correspondence between equidistant q-ary codes and resolvable designs presented by Semakov and Zinov'ev [47]. The resulting gap codes are equidistant and have constant gap weight; in this case, there is exactly one `*` in every codeword.

We show that the isomorphism classes of near resolutions and the equivalence classes of gap codes are in a one-to-one relation. Then we use an orderly algorithm in the style of Section 3.3.1 to enumerate the gap codes that correspond to near resolutions of $(13, 4, 3)$ BIBDs. We want to choose a subset of the lexicographically ordered codewords, where the group that acts on the set of codewords is generated by permutations of the codeword positions (columns of the matrix representation) and by permutations of the non-`*` values in each position.

5.3 CLASSIFYING WHIST TOURNAMENTS BY BUILDING ON RESOLUTIONS

Let us view a $\mathrm{Wh}(v)$ as a set of rounds, each of which is a set of games. When each game in a $\mathrm{Wh}(v)$ is replaced by the set of the players involved, the result is a resolution (for $v = 4k$) or a near resolution (for $v = 4k+1$) of a $(v, 4, 3)$ BIBD. It is a very natural idea to classify the nonisomorphic $\mathrm{Wh}(v)$ by determining the nonisomorphic resolutions or near resolutions of $(v, 4, 3)$ BIBDs, testing for each of them whether the elements in each block may be arranged into whist games so that the requirements of a $\mathrm{Wh}(v)$ are met, and eliminating isomorphic tournaments.

To be able to test whist tournaments for isomorphism, one must first define which whist tournaments are isomorphic. For running a whist tournament in practice it is necessary to have an ordered list of rounds, each of which contains an ordered list of games, each of which contains an ordered list of players. In enumerating whist tournaments, however, two whist tournaments are not interestingly different if one may be obtained from the other one by reordering the rounds in the tournament, reordering the games in each round, or by any combination of these two. Thus we consider whist tournaments to be sets of rounds, which are sets of games.

Additionally, certain—but not all—permutations of the players in each game preserve the essential properties of the whist game. As an example, in a whist game nothing essential changes, if east and west are exchanged, as the partner pairs remain the same, but exchanging north and east will change the structure. In the approach chosen, the specific seats are abstracted away in favor of representing the whist games as relations on the pairs of players in each game. Since, depending of the type of whist tournament being considered, for every player there is exactly one partner, left-hand opponent, right-hand opponent, opponent of the first kind or an opponent of the second kind, the relations may in fact be thought of as permutations on the set of players in each game. For the purposes of isomorph checking, then, a whist game is considered to consist of the set of the players and the relevant relations in that game, and two whist tournaments are considered isomorphic, if a permutation of the players maps the rounds and games of one tournament to the rounds and games of the other.

For $v \in \{4, 5, 8, 9\}$, there is a unique $(v, 4, 3)$ (N)RBIBD. For $v = 12$, Morales and Velarde [38] compute the five nonisomorphic $(12, 4, 3)$ RBIBDs. All of these designs are uniquely resolvable. To determine the $\mathrm{Wh}(v)$ for $v \leq 12$, it suffices to determine all $\mathrm{Wh}(v)$ that can be constructed on one of these designs. The resulting $\mathrm{Wh}(v)$ must then be tested for isomorphism.

Given a $(v, 4, 3)$ (N)RBIBD, it is a relatively simple matter to formulate a Boolean formula, whose satisfying assignments correspond to whist tournaments of the desired kind. In the simplest case of whist tournaments, we introduce Boolean variables of the type p_{xyb} to represent whether x and y are partners in block B_b. It is then straightforward to form clauses that encode the restrictions of a whist tournament, namely that every player must have exactly one partner at each table, and that every player must partner every other player exactly once. For directed and triplewhist tournaments, it is necessary to introduce additional Boolean variables to represent the different opponent relations. For determining all satisfying assignments to the satisfiability problems, we use Smodels [51].

5.4 ELIMINATING ISOMORPHIC WHIST TOURNAMENTS

Examining the automorphisms of a block design by expressing the design as a block incidence graph is a well-known technique. The graph has one vertex for each point of the design and one vertex for each block of the design. A directed edge is introduced from each point to each block of which the point is a member. The automorphisms of the set system correspond with the automorphisms of the block incidence graph, restricted to act on the vertices that correspond with the points of the block design.

Let us define a generalized set system inductively. Given a point set V, v is a generalized set system for all $v \in V$. If $v_1, \ldots, v_k$ are distinct generalized set systems, then so is $\{v_1, \ldots, v_k\}$. If $v_1, \ldots, v_k$ are generalized set systems, then so is $(v_1, \ldots, v_k)$.

A generalized set system may be converted to a directed graph as follows: the set of vertices is taken to be the set of points, sets and tuples in the generalized set system; each point, set and tuple is included only once irrespective of how many times it occurs in the generalized set system. A directed edge is introduced from every element to every set that contains the element as a member. Encoding ordered tuples is slightly more complicated. In the encoding used in the articles [P2, P3], when an element x is the ith element of an ordered tuple t, an auxiliary vertex u is introduced. Then the directed edges (x, u) and (u, t) are introduced, and u is colored with color i. In examining the automorphisms of a graph, we restrict ourselves to those automorphisms that preserve the color of each vertex. This encoding is rather wasteful, as it requires adding an additional vertex for each membership in an ordered tuple. Another method would be to determine the maximum number of elements in a tuple, say k, to introduce the additional vertices $c_1, \ldots, c_k$, and to color each c_i with color i. Then a tuple $(v_1, \ldots, v_j)$, where $j \leq k$, may be encoded as the set $\{\{v_1, c_1\}, \ldots, \{v_j, c_j\}\}$.

After converting the whist tournaments of various types to directed graphs by first expressing them as generalized set systems, we may use the power of the graph isomorphism software nauty [33] by Brendan McKay for detecting isomorphic tournaments and computing their automorphism groups. For some selected whist tournaments with a relatively large automorphism group, GAP [17] was used to examine their symmetries.

5.5 RESULTS

In this section, the whist tournaments with up to thirteen players are classified. It is found that no DWh(12), TWh(12), or TWh(13) exists. Some whist tournaments with large automorphism groups are presented. Also, in the process of classifying the whist tournaments, the near resolvable $(13, 4, 3)$ balanced incomplete block designs are classified.

Classifying $(16, 4, 3)$ RBIBDs or $(17, 4, 3)$ NRBIBDs seems completely out of reach for the method described. It is not certain whether the number of such designs is too large to handle, but judging by some tentative trial computations, the number of partial codes visited in the orderly search would be prohibitively large.

6 SUM AND DIFFERENCE PACKINGS AND COVERS

In this section we define sum and difference packings and covers, and describe the computational methods used to obtain the optimal packings and covers for small cyclic and Abelian groups in [P4, P5, P6]. Sum and difference packings and covers are related to difference sets. As we restrict the investigation to Abelian groups only, we use additive notation.

Definition 16 *A (v, k, λ) difference set is a k-element subset S of a group G of order v such that every nonzero element of G may be expressed in exactly λ ways as the difference $s - t$ of two elements $s, t \in S$.*

Clearly, one must have $\lambda(v-1) = k(k-1)$. Difference sets with $\lambda = 1$ are known as planar difference sets.

Theorem 17 *Let q be a prime power and let $d \geq 2$ be an integer. By a theorem of Singer, there then exists a cyclic (v, k, λ) difference set with*

$$v = \frac{q^{d+1}-1}{q-1}, k = \frac{q^d - 1}{q-1}, \lambda = \frac{q^{d-1}-1}{q-1}.$$

Corollary 18 *There exists a cyclic $(q^2 - q + 1, q + 1, 1)$ difference set for every prime power q.*

Sum and difference packings and coverings may be viewed as variants and relaxations of planar difference sets. In a planar difference set, every nonzero group element may be expressed in exactly one way as the difference of two elements in the difference set; in a sum or difference packing, every group element may be expressed in at most one way as the sum or difference of two elements of the packing, and in a sum or difference cover, every group element may be expressed in at least one way as the sum or difference of two elements of the packing.

In particular, a $(v, k, 1)$ difference set is always an optimal difference packing and covering.

Definition 19 *A k-element subset S of an Abelian group G is a difference packing, if no nonzero $g \in G$ may be expressed as two distinct differences $s - t$ and $u - v$ where $s, t, u, v \in S$.*

Definition 20 *A k-element subset S of an Abelian group G is a sum packing, if no $g \in G$ may be expressed as $g = s + t = u + v$, where $s, t, u, v \in S$ and $\{s, t\} \neq \{u, v\}$.*

Sum packings and difference packings coincide: Choose $s, t, u, v \in S$ such that $s \neq u$ and $s \neq v$. If now $s + t = u + v$, then $s - v = u - t$, and conversely.

Sum packings are closely related to Sidon sets. A subset of the natural numbers $S \subseteq \mathbb{N}$ is a Sidon set, if all sums $s + t$ are distinct where $s, t \in S$ and $s \leq t$; sum packings may be thought of as Sidon sets in a finite group. The problem is to determine the maximum density of a Sidon set.

Graham and Sloane [24] show that $v_{\mathbb{Z}}(n) \sim n^2$ as $n \to \infty$, where $v_{\mathbb{Z}}(n)$ is the order of the smallest cyclic group that admits an n-element sum packing. For the Abelian analogue $v(n)$ they give $\binom{n}{2} \le v(n) < n^2 + O(n^{36/23})$. Babai and Sós [5] show that any n-element subset of any group contains a Sidon subset of size $cn^{1/3}$. In [P5] the maximum density of a sum packing in an Abelian group is linked to the proportion of involutions in the group.

Definition 21 *A k-element subset S of an Abelian group G is a strict sum packing, if no $g \in G$ may be expressed as $g = s+t = u+v$, where $s, t, u, v \in S$ with $s \neq t$, $u \neq v$, and $\{s, t\} \neq \{u, v\}$.*

One motivation for examining strict sum packings of Abelian groups is that they can be used to construct certain constant-weight error-correcting codes, as described by Brouwer et al. in their survey [9]. The idea is to define a mapping $f : \{0, 1\}^k \mapsto G$ by letting $f(a_1 a_2 \cdots a_n) = \sum_{i=1}^{n} g_i a_i$ and partitioning the codewords of length n and weight w according to the value of f. An (n, d, w) constant weight code is a set of binary vectors of length n, pairwise Hamming distance at least d and constant weight w, and $A(n, d, w)$ is the maximum number of codewords in an (n, d, w) code. If g_i are distinct and $S = \{g_1, \ldots, g_k\}$ is a strict sum packing in some Abelian group G, then the Hamming distance between any distinct two codewords x_1 and x_2 of the same weight w for which $f(x_1) = f(x_2)$ must be at least 6: since the words have the same weight, the distance must be even; the distance cannot be 2 as long as g_i are distinct, and it cannot be 4 as the g_i form a strict sum packing. As the $\binom{n}{w}$ codewords are partitioned into $|G|$ sets, each of which is a $(n, 6, w)$ constant weight code, we obtain

$$A(n, 6, w) \ge \frac{1}{|G|}\binom{n}{w} \tag{6.1}$$

by the pigeonhole principle.

Definition 22 *A k-element subset S of an Abelian group G is a sum cover, if every $g \in G$ may be expressed as a sum $s + t$ where $s, t \in S$.*

Definition 23 *A k-element subset S of an Abelian group G is a strict sum cover, if every $g \in G$ may be expressed as a sum $s + t$ where $s, t \in S$ and $s \neq t$.*

A curious observation is that groups of the form $(\mathbb{Z}_2)^k$ have no strict sum cover, as it is impossible to express 0 as the sum of two distinct group elements.

Definition 24 *A k-element subset S of an Abelian group G is a difference cover, if every $g \in G$ may be expressed as a difference $s - t$ where $s, t \in S$.*

Our motivation for examining sum and difference covers of Abelian groups is simply that it is a natural complement of examining the sum and difference packings in Abelian groups. Nevertheless, these problems have independent interest. Chateauneuf, Ling, and Stinson [12] consider the related problems of slope packing and covering in the context of computing discrete logarithms.

There are a handful of computational results on the optimal sum and difference packings and covers of small cyclic groups. Graham and Sloane [24] determine for $k \leq 9$ the largest cyclic group that has a k-element cover, for $k \leq 10$ the largest cyclic group that has a k-element strict sum cover, for $k \leq 12$ except $k = 11$ the smallest cyclic group that admits a k-element sum packing, and for $k \leq 10$ the smallest cyclic group that admits a k-element strict sum packing. Wiedemann [52] determines the minimum difference covers of $\mathbb{Z}_n$ for $n \leq 133$, and Fitch and Jamison [16] determine the minimum sum and strict sum covers of $\mathbb{Z}_n$ for $n \leq 54$. Swanson [50] determines the maximum difference packings of $\mathbb{Z}_n$ for $n \leq 144$.

In [P4, P5, P6] we carry out backtrack searches to determine the optimum packings and covers of cyclic and Abelian groups. In each case, we use a fairly straightforward orderly algorithm of the type described in Section 3.3.1. The equivalence of subsets is discussed in Section 6.1 and our canonicity test is discussed in Section 6.2.

Apart from canonicity testing, the backtrack algorithm is very straightforward. The subsets are constructed by adding elements to the subset one by one in order. In computing a maximum packing an element may not be added to the set if adding it would make it possible to express a group element in more than one way as the sum, strict sum, or difference of two elements of the subset, depending on the type of packing being constructed. The maximum packing is determined by trying out all alternatives in a recursive fashion.

Computing a minimum cover is done in a slightly different manner. The search algorithm determines whether, for given k and G, there exists a cover of G with at most k elements. In this case sums or differences may cover the same elements more than once, but a search branch may be eliminated from consideration once it is clear that no way of completing the current set to a k-subset can yield a cover; this can occur when relatively many sums or differences cover the same elements of G. A simple volume bound for each of the three cases is presented in [P6].

6.1 THE EQUIVALENCE OF SUBSETS

Two subsets S, T of an Abelian group G are equivalent if $T = S^{\psi}$ where $\psi : G \mapsto G$ is a bijection that preserves the equality of two-element sums. Such ψ must be of the form $x^{\psi} = x^{\phi} + c$, where ϕ is an automorphism of G and $c \in G$. The automorphisms of an Abelian group are given by Shoda [49].

Any finite Abelian group may be expressed as a direct product of cyclic groups of prime power order. When the cyclic direct factors whose orders are powers of the same prime are grouped together, a finite Abelian group may be expressed as a direct product of primary Abelian groups of the form $\mathbb{Z}_{p^{e_1}} \times \ldots \times \mathbb{Z}_{p^{e_k}}$, where p is prime and $e_1 \geq \ldots \geq e_k$. Shoda shows that the automorphism group of a finite Abelian group is the direct product of the automorphism groups of the primary Abelian direct factors, and that the automorphism groups of the primary Abelian direct factors consist of automorphisms of the form $\phi(x) = Ax$, where x is a column vector that represents the element of the group in the obvious way, and A is a $k \times k$ matrix of the form

$$A = \begin{pmatrix} h_{11} & p^{e_1-e_2}h_{12} & \cdots & p^{e_1-e_n}h_{1n} \\ h_{21} & h_{22} & & p^{e_2-e_n}h_{2n} \\ \vdots & & \ddots & \vdots \\ h_{n1} & \cdots & \cdots & h_{nn} \end{pmatrix} \tag{6.2}$$

with $\det A \not\equiv 0 \bmod p$.

The automorphisms of a cyclic group $\mathbb{Z}_n$ may be expressed in the much simpler form $\phi(x) = ax \bmod n$, where $a \in \mathbb{Z}_n$ such that $\gcd(a, n) = 1$; the equivalence mappings then take the form $\psi(x) = ax + b \bmod n$ where $a, b \in \mathbb{Z}_n$ and $\gcd(a, n) = 1$.

6.2 CANONICITY TEST FOR SUBSETS OF ABELIAN GROUPS

In examining cyclic groups in [P4], we examine equivalence mappings of the form $f(x) = ax + b \bmod n$, where $a, b \in \mathbb{Z}_n$ and $\gcd(a, n) = 1$. There are a total of $n\varphi(n)$ such mappings, where φ is the Euler totient function. Wiedemann [52] tests the canonicity of a subset S by computing $f(S)$ for each of these $n\varphi(n)$ equivalence mappings. As we have chosen the obvious order for the elements of $\mathbb{Z}_n$, we may take advantage of the fact that all subsets that contain both 0 and 1 precede all subsets that do not. Therefore, to test the canonicity of a k-element subset S, we test for all pairs (s, t) where $s, t \in S$ whether $t - s$ is invertible in $\mathbb{Z}_n$. If it is, we observe that letting $f(x) = (t - s)^{-1}x - s \bmod n$ maps (s, t) to $(0, 1)$. Instead of $n\varphi(n)$ mappings, we only need to consider $k(k-1)$ ordered pairs. Our canonicity test may miss an opportunity for rejecting a subset as noncanonical in the case where S contains no pair whose difference is invertible, but compared to a complete canonicity test, this does not seem to affect the running time of the search significantly. This is the canonicity test used in [P4]. It should be remarked that according to Jamison [29] every sum cover of a cyclic group $\mathbb{Z}_n$ of order $n < 2310$ contains a pair of elements whose difference is invertible. However, this property has not been used in the computations.

For Abelian groups our canonicity test is slightly more complex. A slightly unusual order is chosen for the elements of the Abelian group. The identity element 0 is chosen to be the smallest element, and an element g of maximum order is chosen as the next smallest element. After this we can again consider those equivalence mappings that map an ordered pair (s, t) to $(0, g)$. However, when the automorphism group of the Abelian group is large, there can be a vast number of such equivalence mappings. This occurs when the Abelian group has several cyclic direct factors whose orders are powers of the same prime. The Abelian group $(\mathbb{Z}_p)^k$, for example, has $\prod_{i=0}^{k-1}(p^k - p^i)$ automorphisms; for $\mathbb{Z}_3^4$ there are 24,261,120 automorphisms and 1,965,150,720 equivalence mappings in total. Testing even a significant portion of these at each search step is unthinkable. In the canonicity test in [P5, P6] we only try a subset of the equivalence mappings in trying to prove the current subset noncanonical.

To test the canonicity of a subset in the backtrack search of [P5, P6], we only use a number of precomputed equivalence mappings. For every ordered pair (s, t) such that $s, t \in G$ and $t - s$ is an element of maximum order in the Abelian group, we precompute an equivalence mapping $f_{s,t}$ that maps (s, t) to $(0, g)$. Then, to test the canonicity of a subset $S \subseteq G$, we determine all ordered pairs (s, t) such that $s, t \in S$ and find the associated $f_{s,t}$. A subset S is accepted as canonical if $S \leq f_{s,t}(S)$ for each such $f_{s,t}$. For cyclic groups this essentially reduces to the test described for cyclic groups, and since the number of elements of maximum order in an Abelian group is at least as large as the number of elements of maximum order in the cyclic group of the same order, this performance of this test on an Abelian group should be at least comparable in performance to the performance on a cyclic group of the same order.

The canonicity test described is not entirely satisfactory for Abelian groups with a large automorphism group, as only a small subset of the equivalence mappings are used in canonicity testing. It would be interesting to examine whether the canonical augmentation method could be applied here with success. Assuming that the subsets would be constructed an element at a time, this would require a fast parent function that would identify an element in a subset of an Abelian group. However, designing such a parent function does not seem entirely trivial.

6.3 RESULTS

In [P4], sum and strict sum packings in cyclic groups were considered. The optimum strict sum packings were obtained up to $\mathbb{Z}_{183}$ and the optimum sum packings up to $\mathbb{Z}_{168}$. Also, the analytical lower bound $n \geq k(k-3)$ was obtained for the order of a cyclic group $\mathbb{Z}_n$ that admits a k-element strict sum packing. The 15-element strict sum packing of $\mathbb{Z}_{183}$ comes remarkably close to this bound.

The objective of [P5] was to investigate whether non-cyclic Abelian groups would allow denser strict sum packings than cyclic groups. The maximum strict sum packings were computed for Abelian groups of order up to 183. The results show no clear pattern. In particular, we determined for $2 \leq k \leq 15$ the smallest Abelian group that admits a k-element strict sum packing. It was found that non-cyclic Abelian groups outperform the cyclic groups in this regard only for $k = 6, 7, 9$. The analytical bounds obtained would seem to support the hypothesis that non-cyclic Abelian groups do not allow asymptotically denser packings than cyclic groups.

In [P6], the sum covers, strict sum covers, and difference covers of Abelian groups of order up to 85, 90, and 127, respectively, were determined. Again, with only few exceptions, non-cyclic Abelian groups require at least as many elements to cover as the cyclic group of the same order.

7 CONCLUSIONS

In this thesis, computational methods are applied to combinatorial problems.

Ramsey theory is an intriguing field of combinatorics, and the Ramsey numbers $R(k, l)$ are one of its simplest manifestations. Consequently, Ramsey numbers have been the subject of much research. The lower bound presented in this thesis is the best known lower bound for the Ramsey number $R(5, 9)$.

The results on whist tournaments settle questions that have been open for a long time. In particular it is found that no DWh(12), no TWh(12), and no TWh(13) exist. The result on DWh(12) establishes that no resolvable perfect $(12, 4, 1)$ Mendelsohn design exists.

Determining the minimum covers and maximum packings of Abelian groups is an open ended problem. For cyclic groups, the results in this thesis expanded the range for which the optimum covers and packings are known, and it seems that these problems have not been systematically examined before for Abelian groups.

Computational methods can clearly be very useful in combinatorics. There are still many open problems to which computational methods could be applied, and there is indubitably much room for improvement in the currently known methods.

BIBLIOGRAPHY

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

[2] R. J. R. Abel and G. Ge. An almost completion for the existence of triplewhist tournaments TWh(v). Submitted for publication.

[3] I. Anderson. *Combinatorial Designs*. Elis Horwood, Chichester, 1990.

[4] I. Anderson. A hundred years of whist tournaments. *J. Combin. Math. Combin. Comput*, 19:129–150, 1995.

[5] L. Babai and V. T. Sós. Sidon sets in groups and induced subgraphs of Cayley graphs. *Europ. J. Combinatorics*, 6:101–114, 1985.

[6] R. D. Baker. *Factorization of graphs*. Doctoral thesis, Ohio State University, 1975.

[7] F. E. Bennett and X. Zhang. Resolvable Mendelsohn designs with block size 4. *Aequationes Math.*, 40:248–260, 1990.

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

[9] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith. A new table of constant weight codes. *IEEE Trans. Inform. Theory*, 36:1334–1380, 1990.

[10] 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.

[11] 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.

[12] M. Chateauneuf, A. C. H. Ling, and D. R. Stinson. Slope packings and coverings, and generic algorithms for the discrete logarithm problem. *J. Combin. Des.*, 11:36–50, 2002.

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

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

[15] I. A. Faradžev. Constructive enumeration of combinatorial objects. *Colloq. Internat. CNRS*, 260:131–135, 1978.

[16] M. A. Fitch and R. E. Jamison. Minimum sum covers of small cyclic groups. *Congr. Numer.*, 147:65–81, 2000.

[17] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.3*, 2002. (`http://www.gap-system.org`).

[18] G. Ge and C. W. H. Lam. Whist tournaments with the three person property. *manuscript*, 2002.

[19] G. Ge and C. W. H. Lam. Some new triplewhist tournaments $TWh(v)$. *J. Combin. Theory Ser. A*, 101:153–159, 2003.

[20] G. Ge and A. C. H. Ling. A new construction for z-cyclic whist tournaments. *Discrete Appl. Math.*, 131:643–650, 2003.

[21] G. Ge and L. Zhu. Frame constructions for $\mathbb{Z}$-cyclic triplewhist tournaments. *Bull. Inst. Combin. Appl.*, 32:53–62, 2001.

[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] R. L. Graham and N. J. A. Sloane. On additive bases and harmonious graphs. *Siam J. Alg. Disc. Meth.*, 1:382–404, 1980.

[25] J. E. Graver and J. Yackel. Some graph theoretic results associated with Ramsey's theorem. *J. Combin. Theory*, 4:125–175, 1968.

[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] H. Harborth and S. Krause. Ramsey numbers for circulant colorings. *Congr. Numer.* To appear.

[29] R. E. Jamison. The Helly bound for singular sums. *Discrete Math.*, 249:117–133, 2002.

[30] G. Kéry. Ramsey egy gráfelméleti tételéröl. *Mat. Lapok*, 15:204–224, 1964.

[31] Y. Lu and L. Zhu. On the existence of triplewhist tournaments TWh(v). *J. Combin. Des.*, 5:249–256, 1997.

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

[33] B. D. McKay. nauty user's guide. Technical Report TR-CS-90-02, Computer Science Department, Australian National University, 1990.

[34] B. D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26:306–324, 1998.

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

[36] B. D. McKay and S. P. Radziszowski. Subgraph counting identities and Ramsey numbers. *J. Combin. Theory Ser. B*, 69:193–209, 1997.

[37] B. D. McKay and K. M. Zhang. The value of the Ramsey number $R(3, 8)$. *J. Graph Theory*, 16:99–105, 1992.

[38] L. B. Morales and C. Velarde. A complete classification of $(12, 4, 3)$-RBIBDs. *J. Combin. Des.*, 9:385–400, 2001.

[39] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Appl. Math.*, 120:197–207, 2002.

[40] Oxford English Dictionary. `http://www.oed.com/`, cited October 2003.

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

[42] S. P. Radziszowski. Small Ramsey numbers. *Electron. J. Combin.*, 1:#DS1, 1994. Revision #9: July 15, 2002.

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

[44] R. C. Read. Every one a winner *or* how to avoid isomorphism search when cataloguing combinatorial configurations. *Ann. Discrete Math.*, 2:107–120, 1978.

[45] M. Schaefer. Graph Ramsey theory and the polynomial hierarchy. *J. Comput. System Sci.*, 62:290–322, 2001.

[46] I. Schur. Über die Kongruenz $x^m + y^m = z^m \bmod p$. *Jahresber. Deutsch. Math.-Verein.*, 25:114–117, 1916. Reprinted in [8].

[47] N. V. Semakov and V. A. Zinov'ev. Equidistant q-ary codes with maximal distance and resolvable balanced incomplete block designs. *Problems Inform. Transmission*, 4(2):1–7, 1968.

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

[49] K. Shoda. Über die Automorphismen einer endlichen Abelschen Gruppe. *Math. Ann.*, 100:674–686, 1928.

[50] C. N. Swanson. Planar cyclic difference packings. *J. Combin. Des.*, 8:426–434, 2000.

[51] T. Syrjänen and I. Niemelä. The Smodels systems. In T. Eiter, W. Faber, and M. Truszczyński, editors, *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 2173 of *Lecture Notes in Artificial Intelligence*, pages 434–438, Vienna, Austria, Sept. 2001. Springer-Verlag, Berlin.

[52] D. Wiedemann. Cyclic difference covers through 133. *Congr. Numer.*, 90:181–185, 1992.

[53] X. Zhang. A few more RPMDs with k=4. *Ars Combin.* To appear.

[54] X. Zhang. On the existence of $(v, 4, 1)$-RPMD. *Ars Combin.*, 42:3–31, 1996.

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A76	Timo Latvala On Model Checking Safety Properties. December 2002.
HUT-TCS-A77	Satu Virtanen Properties of Nonuniform Random Graph Models. May 2003.
HUT-TCS-A78	Petteri Kaski A Census of Steiner Triple Systems and Some Related Combinatorial Objects. June 2003.
HUT-TCS-A79	Heikki Tauriainen Nested Emptiness Search for Generalized Büchi Automata. July 2003.
HUT-TCS-A80	Tommi Junttila On the Symmetry Reduction Method for Petri Nets and Similar Formalisms. September 2003.
HUT-TCS-A81	Marko Mäkelä Efficient Computer-Aided Verification of Parallel and Distributed Software Systems. November 2003.
HUT-TCS-A82	Tomi Janhunen Translatability and Intranslatability Results for Certain Classes of Logic Programs. November 2003.
HUT-TCS-A83	Heikki Tauriainen On Translating Linear Temporal Logic into Alternating and Nondeterministic Automata. December 2003.
HUT-TCS-A84	Johan Wallén On the Differential and Linear Properties of Addition. December 2003.
HUT-TCS-A85	Emilia Oikarinen Testing the Equivalence of Disjunctive Logic Programs. December 2003.
HUT-TCS-A86	Tommi Syrjänen Logic Programming with Cardinality Constraints. December 2003.
HUT-TCS-A87	Harri Haanpää, Patric R. J. Östergård Sets in Abelian Groups with Distinct Sums of Pairs. February 2004.
HUT-TCS-A88	Harri Haanpää Minimum Sum and Difference Covers of Abelian Groups. February 2004.
HUT-TCS-A89	Harri Haanpää Constructing Certain Combinatorial Structures by Computational Methods. February 2004.

ISBN 951-22-6941-4
ISSN 1457-7615