The near resolvable 2-$(13, 4, 3)$ designs and thirteen-player whist tournaments*

Harri Haanpää and Petteri Kaski
Laboratory for Theoretical Computer Science
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, FINLAND.
e-mail: harri.haanpaa@hut.fi, petteri.kaski@hut.fi

Abstract

A v-player whist tournament is a schedule of games, where in each round the v players are partitioned into games of four players each with at most one player left over. In each game two of the players play as partners against the other two. All pairs of players must play in the same game exactly three times during the tournament; of those three times, they are to play as partners exactly once. Whist tournaments for v players are known to exist for all $v \equiv 0, 1 \pmod 4$. The special cases of directed whist tournaments and triplewhist tournaments are known to exist for all sufficiently large v, but for small v several open cases remain. In this paper we introduce a correspondence between near resolvable 2-(v, k, λ) designs and a particular class of codes. The near resolvable 2-$(13, 4, 3)$ designs are classified by classifying the corresponding codes with an orderly algorithm. Finally, the thirteen-player whist tournaments are enumerated starting from the near resolvable 2-$(13, 4, 3)$ designs.

Keywords:
near resolvable design, whist tournament, orderly algorithm
2000 Mathematics Subject Classification:
primary 05B05, secondary 05B30, 05-04, 94B25

*Both authors were supported by Helsinki Graduate School in Computer Science and Engineering (HeCSE). In addition, the first author was supported by a grant from the Nokia Foundation and the second author by a grant from the Foundation of Technology, Helsinki, Finland (Tekniikan Edistämissäätiö).

1 Introduction

The mathematical study of whist tournaments was started in the 1890s by E. H. Moore. In the 1970s Baker, Hanani, and Wilson [5] showed that a $\mathrm{Wh}(v)$ exists for all $v \geq 4$ with $v \equiv 0 \pmod 4$. According to Anderson's survey [3], the case $v \equiv 1 \pmod 4$ was also settled in the 1970s; a proof of these existence results may be found in Anderson's book [2].

Particularly since the existence of $\mathrm{Wh}(v)$ was fully settled, the focus of research has been on the existence of whist tournaments with additional structure, such as directed whist tournaments $\mathrm{DWh}(v)$, triplewhist tournaments $\mathrm{TWh}(v)$ and also $\mathbb{Z}$-cyclic $\mathrm{Wh}(v)$, $\mathrm{DWh}(v)$ or $\mathrm{TWh}(v)$, none of which has been completely settled. We will quote only the existence results essential for our work; for more information we refer the interested reader to [3].

It follows from the results of Bennett and Zhang [6] and Zhang [26] on resolvable perfect Mendelsohn designs that a $\mathrm{DWh}(v)$ exists for all $v \geq 5$ with $v \equiv 1 \pmod 4$, that a $\mathrm{DWh}(v)$ does not exist for $v = 4$ or $v = 8$, and that a $\mathrm{DWh}(v)$ exists for all $v \equiv 0 \pmod 4$, $v \geq 12$ except possibly for 48 values, the smallest of which are $v = 12$ and $v = 16$. Following a classification of resolvable 2-$(12, 4, 3)$ designs by Morales and Velarde in [20] it was determined by Haanpää and Östergård in [14] that no $\mathrm{DWh}(12)$ exists.

A $\mathrm{TWh}(v)$ does not exist for $v = 5$ or $v = 9$. Lu and Zhu [16] show that a $\mathrm{TWh}(v)$ exists for all $v \equiv 0, 1 \pmod 4$ with $v \geq 12$ except possibly for

$$v = \{12, 56\} \cup \{13, 17, 45, 57, 65, 69, 77, 85, 93, 117, 129, 133, 153\}.$$

Ge and Zhu give a $\mathrm{TWh}(133)$ in [11]. Based on [20], it was found in [14] and, independently, by Ge and Lam in [10] that no TWh(12) exists. A TWh(45) and a TWh(56) are also constructed in [10]. Very recently, Abel and Ge [1] have constructed a TWh(v) for the remaining open values of v except for $v = 13$ and $v = 17$. In this paper we show that no TWh(13) exists, so only the case $v = 17$ remains open.

In the literature there seem to be very few classification results concerning whist tournaments. The Wh(v), DWh(v), and TWh(v) with $v \leq 12$ are classified in [14]. Finizio [8] determines by computer the $\mathbb{Z}$-cyclic Wh(v) for $v \leq 21$. There are three $\mathbb{Z}$-cyclic Wh(13), one of which allows the construction of a DWh(13) (or, as will be seen, four nonisomorphic ones) on it.

In this paper we classify up to isomorphism the 13-player whist tournaments, directed whist tournaments, and triplewhist tournaments by first producing a classification up to isomorphism of the underlying near resolvable 2-$(13, 4, 3)$ designs and their near resolutions.

The rest of this paper is organized as follows. Section 2 contains definitions of the combinatorial structures considered and the associated definitions of equivalence and isomorphism. Section 3 examines a correspondence between near resolutions of near resolvable designs and a particular class of codes. Section 4 describes a classification algorithm for the nonisomorphic near resolvable 2-$(13, 4, 3)$ designs and their near resolutions that utilizes this correspondence. Section 5 describes an approach for classifying the nonisomorphic Wh(13) and DWh(13) from the nonisomorphic 2-$(13, 4, 3)$ near resolutions. Section 6 presents the classification results for the structures considered. By examining the Wh(13), we find that no TWh(13) exists.

2 Definitions

In this section we give definitions of whist tournaments, 2-(v, k, λ) designs, and related concepts necessary for this work. We first describe whist tournaments in prose and then proceed to give a more formal definition.

A whist tournament is a schedule of games, where in each round the v players are partitioned into games of four players each with at most one player left over. All pairs of players must play in the same game exactly three times during the tournament. Additionally, the order of the players in a game is relevant. The players are generally thought to sit on the north, east, south, and west side of the playing table. North and south are partners, as are east and west. Every pair of players must play as partners exactly once during the tournament.

A directed whist tournament is a whist tournament with the additional property that for any pair of distinct players, p_i and p_j, the player p_j plays once as p_i's left-hand opponent and once as p_i's right-hand opponent.

In a triplewhist game north and east, as well as south and west, are opponents of the first kind, and north and west, as well as south and east, are opponents of the second kind. A triplewhist tournament is a whist tournament with the additional property that every pair of players plays once as opponents of the first kind and once as opponents of the second kind.

While it is clear that for a whist tournament to run smoothly, it is necessary to assign an order to the rounds as well as to place the players in specific seats, from a mathematical point of view this ordering presents little of interest. Hence, we wish to avoid unnecessary ordering in our definition of a whist tournament. One may interpret our following definition of a whist tournament as the definition of a certain equivalence class of whist tournaments.

For our purposes a *whist game* is a partner relation over four players. The partner relation may be represented as a permutation p over the players in the whist game; when $p(x)$ denotes the partner of x, we must have $p(x) \neq x$ and $p^2(x) = x$ for all players x in the whist game.

A *directed whist game* is two relations over the same four players: the partner relation as for a whist game, and a left-hand opponent relation. The left-hand opponent relation may also be represented as a permutation ℓ over the players in the game; when $\ell(x)$ is the left-hand opponent of x, we must have $\ell^2 = p$.

A *triplewhist game* is three relations over the same four players: the partner relation p as for a whist game, a first-kind opponent relation o_1 and a second-kind opponent relation o_2. As for p, $o_i(x) \neq x$ for each x in the game, and $o_i^2(x) = x$. Furthermore, $p(x) \neq o_1(x) \neq o_2(x) \neq p(x)$.

A *v-player whist tournament (directed whist tournament, triplewhist tournament)* Wh(v) (DWh(v), TWh(v)) is an ordered pair consisting of a v-element set P of players and a set of rounds, each of which is a partition of P into whist games (directed whist games, triplewhist games) with at most one player left over. Each unordered pair of distinct players must play three times at the same whist game, and they must be partners in exactly one of the games. In a directed whist tournament, every player must also meet every other player exactly once as a left-hand opponent. In a triplewhist game, every player must meet every other player exactly once as an opponent of the first kind and exactly once as an opponent of the second kind.

Whist tournaments of various kinds are often presented as sets of sets of ordered four-tuples of players, where the elements of each four-tuple may be interpreted as the player seated on the north, east, south, and west side of the table in the corresponding game. The seating order then induces the partner and left-hand opponent relations.

Certain whist tournaments may be described as an orbit of an initial round under the action of a cyclic group. Such Wh(v) are known as $\mathbb{Z}$-cyclic. For $v = 4q$, the number of rounds in the tournament and the order of the cyclic group is $v - 1$; one player remains fixed. For $v = 4q + 1$, the number of rounds and the order of the cyclic group is v.

If each game in a Wh(v), DWh(v), or TWh(v) is replaced by the four-element set of the players in that game, the result is a (near) resolution of a 2-$(v, 4, 3)$ design.

A 2-(v, k, λ) *design* is a pair $(P, \mathcal{B})$, where P is a set of v *points* and $\mathcal{B}$ is a multiset of k-subsets of P—called *blocks*—such that every pair of distinct points occurs in exactly λ blocks. A *parallel class* in a design is a set of blocks that partitions the point set. A *near parallel class* is a set of pairwise disjoint blocks whose union is the point set minus one point. A *(near) resolution* of a design is a partition of the multiset of blocks into (near) parallel classes. A design is *(near) resolvable* if it has a (near) resolution.

Let S_v denote the symmetric group on v points. Let S_v act on P in the natural way and in the induced fashion on the set systems over the point set P. We consider two designs, two resolutions, two Wh(v), two DWh(v), or two TWh(v) x_1 and x_2 isomorphic, if $\sigma(x_1) = x_2$ for some $\sigma \in S_v$.

3 Near resolutions and gap codes

In this section we describe a correspondence between near resolutions of 2-(v, k, λ) designs and a particular kind of codes. The correspondence is analogous to that between resolutions and certain error-correcting codes discovered by Semakov and Zinov'ev [23]; the main difference is that we have to pay special attention to the points that do not belong to the near parallel classes of a near resolution.

We require some basic results from design theory. A standard double counting argument gives that every point of a 2-(v, k, λ) design occurs in exactly r of the b blocks, where

$$r(k-1) = \lambda(v-1), \qquad vr = bk. \tag{1}$$

For a design to be near resolvable it is clearly necessary that k divides $v-1$. Denoting the number of blocks in a near parallel class by q and the number of near parallel classes by n, we obtain from (1) that

$$v = kq + 1, \qquad n = b/q = r + \lambda/(k-1). \tag{2}$$

Thus, it is a necessary condition for the existence of a near-resolvable 2-(v, k, λ) design that $v \equiv 1 \pmod{k}$, and $\lambda \equiv 0 \pmod{k-1}$.

We continue with some coding-theoretic definitions. Let $Z_q = \{0, 1, \ldots, q-1\}$ and $Z_{q,*} = Z_q \cup \{*\}$, where "$*$" is a distinct symbol, which will represent a point missing from a near parallel class, that is, a *gap*. A *word* of length n over $Z_{q,*}$ is an ordered n-tuple of elements of $Z_{q,*}$. We write $Z_{q,*}^n$ for the set of all words of length n over $Z_{q,*}$. The symbol at position $1 \le i \le n$ in a word $x \in Z_{q,*}^n$ is denoted by $x(i)$. The *distance* between two words $x, y \in Z_{q,*}^n$ is defined by

$$d(x, y) := |\{i \ : \ x(i) \neq y(i) \text{ or } x(i) = * \text{ or } y(i) = *\}|.$$

(The term "distance" is somewhat misleading since d is clearly not a metric on $Z_{q,*}^n$. We warn the reader that the symbol "$*$" is used with a different interpretation in e.g. [13, 25].) The *gap weight* of a word $x \in Z_{q,*}^n$ is defined by $w_*(x) := |\{i : x(i) = *\}|$. A *gap code* of length n is a nonempty subset of $Z_{q,*}^n$. The *minimum distance* of a gap code $C \subseteq Z_{q,*}^n$ is

$$d(C) := \min_{x,y \in C,\ x \neq y} d(x,y).$$

We say that C is *equidistant* if $d(x,y) = d(C)$ for all distinct $x, y \in C$ and that C has *constant gap weight* w if $w_*(x) = w$ for all $x \in C$. A $(n, M, d, w)_q$ gap code is a gap code with M codewords over $Z_{q,*}^n$ with minimum distance d and constant gap weight w.

We are now ready to describe the correspondence between near resolutions and codes. Let $\mathcal{N}$ be a near resolution of a 2-(v, k, λ) design $(P, \mathcal{B})$, where for convenience we assume $P = \{1, \ldots, v\}$. Denote the near parallel classes of $\mathcal{N}$ by $N_1, \ldots, N_n$ and the blocks in N_i by $N_i(0), \ldots, N_i(q-1)$ for $1 \leq i \leq n$. Note that upon doing this we implicitly introduce an ordering of the parallel classes and the blocks in each near parallel class. The code C that corresponds to $\mathcal{N}$ under this labeling of the near parallel classes and blocks is now defined as follows. With each point $p \in P$ we associate a word $x_p \in Z_{q,*}^n$ defined by

$$x_p(i) := \begin{cases} j & \text{iff } p \in N_i(j) \text{ and} \\ * & \text{iff no such } j \text{ exists} \end{cases} \qquad \text{for } 1 \leq i \leq n. \tag{3}$$

As every pair of distinct points must occur in the same block in λ near parallel classes, and since every point must occur in a total of r blocks, the resulting code $C = \{x_p : p \in P\}$ is easily seen to be equidistant with distance d and constant gap weight w, where

$$d = n - \lambda, \qquad w = n - r. \tag{4}$$

Example 1 *The* $(9,9,6,1)_2$ *gap code and the near resolution of a* 2-$(9,4,3)$ *design below illustrate the correspondence.*

$$\left\{\begin{array}{l} \texttt{00000000}{*}, \\ \texttt{000111}{*}\texttt{10}, \\ \texttt{01100}{*}\texttt{110}, \\ \texttt{0111100}{*}\texttt{1}, \\ \texttt{10}{*}\texttt{010111}, \\ {*}\texttt{01101101}, \\ \texttt{110}{*}\texttt{01011}, \\ \texttt{1101}{*}\texttt{0100}, \\ \texttt{1}{*}\texttt{1011000} \end{array}\right\} \leftrightarrow \left\{\begin{array}{l} \{\{0,1,2,3\},\{4,6,7,8\}\}, \\ \{\{0,1,4,5\},\{2,3,6,7\}\}, \\ \{\{0,1,6,7\},\{2,3,5,8\}\}, \\ \{\{0,2,4,8\},\{1,3,5,7\}\}, \\ \{\{0,2,5,6\},\{1,3,4,8\}\}, \\ \{\{0,3,4,7\},\{1,5,6,8\}\}, \\ \{\{0,3,6,8\},\{2,4,5,7\}\}, \\ \{\{0,5,7,8\},\{1,2,4,6\}\}, \\ \{\{1,2,7,8\},\{3,4,5,6\}\} \end{array}\right\}$$

Conversely, every $(n,v,d,w)_q$ gap code C whose parameters satisfy (1), (2), and (4) for some positive integers v,k,λ,r,b defines a near resolution of a 2-(v,k,λ) design. Namely, denote the words in C by $x_1,\ldots,x_v$ and for convenience, let the point set of the design be $\{1,\ldots,v\}$. Note that here we implicitly introduce an ordering of the words in the code. The near resolution that corresponds to C under this labeling of the codewords consists of the blocks

$$N_i(j) := \{p : x_p(i) = j\}, \tag{5}$$

where $1 \le i \le n$ and $0 \le j \le q-1$. It is not immediately clear that the blocks defined by (5) constitute a near resolution of a 2-(v,k,λ) design. The following theorem shows that this is the case.

Theorem 2 *Every $(n, M, d, w)_q$ gap code C whose parameters satisfy* (1), (2), (4), *and $M = v$ for some positive integers v, k, λ, r, b is equidistant with distance $n - \lambda$. Moreover, every non-"$*$" symbol occurs exactly k times in every position; the symbol "$*$" occurs exactly once in each position.*

Proof. Denote by k_{ij} the number of occurences of symbol $j \neq *$ at position i in the words of C. By the minimum distance condition we have

$$d\binom{v}{2} \leq \sum_{1 \leq \ell < \ell' \leq v} d(x_\ell, x_{\ell'}) = \sum_{i=1}^{n} \left(\binom{v}{2} - \sum_{j=0}^{q-1} \binom{k_{ij}}{2} \right);$$

equality holds if and only if the code is equidistant. Multiplying by two, using (4), and arranging terms, we obtain

$$\lambda v(v-1) \geq \sum_{i=1}^{n} \sum_{j=0}^{q-1} k_{ij}^2 - \sum_{i=1}^{n} \sum_{j=0}^{q-1} k_{ij}.$$

The rightmost sum above is the number of non-"$*$" symbols in the code, and hence it equals $v(n - w) = vr$. As the sum of the k_{ij} is constant, the sum of their squares is minimized when the k_{ij} are equal. Thus,

$$\lambda v(v-1) \geq nq \left(\frac{vr}{nq} \right)^2 - vr = vr\frac{vr}{nq} - vr,$$

where equality holds when the code is equidistant and every non-"$*$" symbol occurs an equal number of times in every position. Noting that $nq = b$ and using (1), we obtain

$$vr(k-1) \geq vrk - vr.$$

Since equality holds, C must be equidistant and every non-"$*$" symbol must occur exactly k times in every position. $\square$

The correspondence given by (3) and (5) clearly depends on the labeling chosen for the blocks (words). By introducing a suitable equivalence for codes we can ignore this dependence; moreover, the equivalence classes of codes will be in a one-to-one correspondence with the isomorphism classes of near resolutions as we shall see.

We say that two codes are *equivalent* if one can be obtained from the other by permuting the positions and the non-"$*$" values in each position.

It will be convenient to view the equivalence classes of codes as orbits of a group action. For this purpose we require the following definitions. Let S_n denote the symmetric group on $\{1,\ldots,n\}$ and, by a slight abuse of notation, let S_q denote the group of all permutations of $Z_{q,*}$ that fix the symbol "$*$". (Our permutations compose from right to left, that is, $\pi\rho(i) = \pi(\rho(i))$.) Denote by $S_q \wr S_n$ the wreath product of S_q by S_n. We regard the elements of $S_q \wr S_n$ as ordered pairs (μ,π), where $\pi \in S_n$ and $\mu = (\mu_1,\ldots,\mu_n)$ is an ordered n-tuple of permutations $\mu_i \in S_q$. The group operation on $S_q \wr S_n$ is given by

$$(\mu,\pi)(\nu,\rho) = (\xi,\eta), \text{ where} \tag{6}$$

$$\eta := \pi\rho \tag{7}$$

$$\xi_i := \mu_i \nu_{\pi^{-1}(i)} \tag{8}$$

Now, an element $(\mu,\pi) \in S_q \wr S_n$ acts on a word $x \in Z_{q,*}^n$ by permuting the positions so that position i becomes position $\pi(i)$, followed by a permutation μ_i of the non-"$*$" values in each position i. In notation, $(\mu,\pi)x = y$, where

y is a word defined by

$$y(i) := \mu_i(x(\pi^{-1}(i))) \quad \text{for all } 1 \le i \le n. \tag{9}$$

It is straightforward to check that this indeed is a group action. As codes are sets of words, this induces an action on codes. In particular, two codes are equivalent if and only if they are on the same orbit of this action.

Theorem 3 *For n, M, d, q, w, v, k, λ that satisfy* (1), (2), (4), *and $M = v$, the equivalence classes of $(n, M, d, w)_q$ gap codes are in a one-to-one correspondence with the isomorphism classes of near resolutions of 2-(v, k, λ) designs.*

Proof. Let $\mathcal{N}$ and $\mathcal{N}'$ be two isomorphic near resolutions. Hence, there exists a $\sigma \in S_v$ such that $\mathcal{N}' = \sigma\mathcal{N}$. Label the near parallel classes and blocks in $\mathcal{N}$ and $\mathcal{N}'$ arbitrarily as $N_i(j)$ and $N'_i(j)$, respectively, where $1 \le i \le n$ ranges over the near parallel classes and $0 \le j \le q-1$ ranges over the blocks in a near parallel class. Since $\mathcal{N}' = \sigma\mathcal{N}$, there exists a $(\mu, \pi) \in S_q \wr S_n$ such that

$$N'_i(j) = \sigma N_{\pi^{-1}(i)}(\mu_i^{-1}(j)) \quad \text{for all } 1 \le i \le n \text{ and } 0 \le j \le q-1. \tag{10}$$

Apply (3) to $\mathcal{N}$ and $\mathcal{N}'$ (subject to the labelings chosen) to obtain the corresponding codes $C = \{x_1, \ldots, x_v\}$ and $C' = \{x'_1, \ldots, x'_v\}$. Let $p \in \{1, \ldots, v\}$. By (3), we have $x'_{\sigma(p)}(i) = j$ iff $\sigma(p) \in N'_i(j)$, that is, by (10) iff $\sigma(p) \in \sigma N_{\pi^{-1}(i)}(\mu_i^{-1}(j))$. Thus, $x'_{\sigma(p)}(i) = j$ iff $p \in N_{\pi^{-1}(i)}(\mu_i^{-1}(j))$, that is, by (3) iff $x_p(\pi^{-1}(i)) = \mu_i^{-1}(j)$. By (9) we have $x'_{\sigma(p)}(i) = j$ iff $(\mu, \pi)x_p(i) = j$. This shows that $C' = (\mu, \pi)C$, so C and C' are equivalent.

Conversely, let C and C' be two equivalent gap codes. Hence, there exists a $(\mu, \pi) \in S_q \wr S_n$ such that $C' = (\mu, \pi)C$. Label the words in C and C' arbitrarily as $x_1, \ldots, x_v$ and $x'_1, \ldots, x'_v$, respectively. Since $C' = (\mu, \pi)C$, there exists a $\sigma \in S_v$ such that

$$x'_{\sigma(p)} = (\mu, \pi)x_p. \tag{11}$$

Apply (5) on C and C' (subject to the labeling chosen) to obtain the near resolutions $N_i(j)$ and $N'_i(j)$, respectively, where $1 \leq i \leq n$ ranges over the near parallel classes and $0 \leq j \leq q-1$ over the blocks of a near parallel class. By (5), we have $\sigma(p) \in N'_i(j)$ iff $x'_{\sigma(p)}(i) = j$, that is, by (11) iff $(\mu, \pi)x_p(i) = j$. By (9), we have $\sigma(p) \in N'_i(j)$ iff $\mu_i(x_p(\pi^{-1}(i))) = j$. Thus, we have $\sigma(p) \in N'_i(j)$ iff $x_p(\pi^{-1}(i)) = \mu_i^{-1}(j)$, that is, by (5) iff $p \in N_{\pi^{-1}(i)}(\mu_i^{-1}(j))$. This shows that $N'_i(j) = \sigma N_{\pi^{-1}(i)}(\mu_i^{-1}(j))$, so $\mathcal{N}$ and $\mathcal{N}'$ are isomorphic. □

4 Generation of near resolutions

In this section we use an orderly algorithm to generate a complete set of equivalence class representatives of $(13, 13, 10, 1)_3$ gap codes. By Theorem 3 and Equation (5) these correspond to a complete set of isomorphism class representatives of near resolutions of 2-$(13, 4, 3)$ designs. From these we classify up to isomorphism the near resolvable 2-$(13, 4, 3)$ designs by investigating the designs underlying the near resolutions.

Our algorithm for classifying gap codes has the structure of an *orderly algorithm* [7, 22]. (For a survey on computational methods in design theory, see [12].) Let G be a finite group that acts on a finite totally ordered set X. The order on X induces the standard lexicographic order on the set of all subsets of X: for $S, T \subseteq X$ we have $S < T$ iff there exists an $x \in X$ such that $x \in S$, $x \notin T$, and for all $y < x$ we have $y \in S$ iff $y \in T$. The induced action of G on subsets of X partitions the subsets into orbits. We call a subset $S \subseteq X$ *canonical* if it is the lexicographic minimum of its orbit.

Theorem 4 *When started on the empty set, the following method generates every canonical subset of X: Given a canonical subset $S \subseteq X$, construct each of the subsets $S \cup \{x\}$ where $x \in X$ and $s < x$ for all $s \in S$, and apply the procedure recursively to those newly constructed subsets that are canonical.*

Proof. Define $f(S) := S \setminus \{\max S\}$ for $\emptyset \neq S \subseteq X$. We remark that f is weakly monotonic on k-subsets: for two k-subsets $S, T \subseteq X$, $S < T$ implies $f(S) \leq f(T)$. Also note that for any $g \in G$, $f(g(S)) \leq g(f(S))$, as both are obtained from $g(S)$ by removing an element — the maximum element in case of $f(g(S))$. Clearly, every canonical set $C \subseteq X$ may be visited in the search only via $f(C)$. Thus, C will be visited iff $f(C)$ is visited and canonical. By weak monotonicity of f we find that $C \leq g(C)$ implies $f(C) \leq f(g(C)) \leq g(f(C))$. The canonicity of C then implies $f(C) \leq g(f(C))$ for all $g \in G$ and therefore $f(C)$ is canonical. Visiting the empty set provides the induction base necessary for showing that all canonical subsets are visited. □

We construct the canonical codes using a codeword-by-codeword backtrack search algorithm of the type described in Theorem 4. Our X is the set $Z_{q,*}^n$ with standard lexicographic order: for $x, y \in Z_{q,*}^n$ we have $x < y$ iff there exists an $i \in \{1, \ldots, n\}$ such that $x(i) < y(i)$ and for all $j \in \{1, \ldots, i-1\}$ we have $x(j) = y(j)$, where the order on $Z_{q,*}$ is $0 < 1 < \cdots < q-1 < *$. Our G is the group $S_q \wr S_n$, which acts on X by (9).

Since (9) preserves the distances between words, we need not consider augmenting the code under construction with words that would cause the equidistance condition to be violated. The augmenting codewords are constructed using coordinatewise backtrack search with pruning whenever a (partial) codeword cannot satisfy the equidistance condition regardless of how the remaining coordinates are completed.

Our implementation of the canonicity test subject to the action (9) is analogous to that described in [15]; the only difference is that the present test keeps the symbol "$*$" fixed at all times and permutes only the symbols $\{0, 1, \ldots, q-1\}$.

Analogously to the nonexistence proof of a resolvable 2-$(15, 5, 4)$ design in [15], we perform isomorph rejection only up to 6 codewords. For each canonical code with 6 codewords, we complete the search by determining the maximum cliques of a graph. The vertices of the graph correspond to the words of gap weight 1 that are both lexicographically larger than the codewords in the code and at distance 10 from each codeword in the code. Two vertices are connected by an edge iff their distance is 10. For each 7-clique in this graph, the words that correspond to the vertices of the clique form a $(13, 13, 10, 1)_3$ gap code together with the 6 codewords in the canonical code. The codes found in this manner are then tested for canonicity. For determining the maximum cliques, we use the algorithm in [21].

We classify the nonisomorphic near resolvable 2-$(13, 4, 3)$ designs using the classification of the near resolutions as follows. For each near resolution, we find the underlying design and compute its canonical representative. (See e.g. [12, Section 9.5] on how to use the *nauty* [19] graph canonical labeling software for this task.) We then associate the near resolutions to each near resolvable design by sorting the canonical representatives of the near resolvable designs.

The search for the nonisomorphic 2-$(13, 4, 3)$ near resolutions was performed using the batch system `autoson` [18] on a network consisting of 15 Linux workstations with CPUs ranging from 1-GHz Athlon Thunderbird to 200-MHz Pentium. The classification was completed in approximately two months of CPU time. Following the classification of the near resolutions, the underlying resolvable designs were classified in less than two minutes of CPU time on a 450-MHz Pentium II.

5 Generation of whist tournaments

In this section we generate the nonisomorphic Wh(13) and DWh(13). The whist tournaments may be generated from the near resolutions by converting the near resolutions into logic programs, essentially instances of the satisfiability problem, such that the satisfying truth assignments correspond to whist tournaments with the given underlying resolution. We then use Smodels [24] to solve the resulting logic programs. The encoding is straightforward. For every pair of points $x < y$ in block B_b, the variable p_{xyb} is introduced to represent whether the players x and y play as partners in game b. For DWh(v) the variable ℓ_{xyb} represents whether y is the left-hand opponent of x in game b. It is convenient to encode the restrictions by using cardinality clauses, allowed in Smodels, using which one may constrain the number of true literals in some subset of the literals. For example, we may specify the equivalent of $|\{p_{xyb} | x, y \in B_b \wedge p_{xyb}\}| = 1$ for all player pairs $x < y$ to ascertain that each player partners every other player exactly once.

Isomorph rejection for the whist tournaments is carried out by transforming the problem to graph isomorphism. We defined Wh(v) and DWh(v) as a kind of a set system in Section 2; now we represent the set system as a directed graph whose vertices correspond to the elements of the set system and whose edges represent the membership relation; $x \in y$ is represented by the edge (x, y). For every x, i, and t such that x is the ith element of an ordered tuple t, we add an auxiliary vertex u_{xit}, introduce the edges (x, u_{xit}) and (u_{xit}, t) and color the vertex u_{xit} with color i so that information about the position in the tuple is preserved in the graph encoding.

Thus, to construct the graph we introduce one vertex for the tournament, n vertices for the rounds, b vertices for the games, v vertices for the players, and $v(v-1)$ vertices for the ordered pairs of players. We add an edge from each of the round vertices to the tournament vertex and from each game vertex to the round vertex. From each player vertex add an edge to each ordered pair the player appears in; then, split these last edges in two by adding a midvertex to each edge and color that vertex with color 1 or 2 according to whether the player is the first or second player in that ordered pair. Finally, add edges to each game from the ordered pairs that appear in the relevant relation, which is the partner relation for $\mathrm{Wh}(v)$ and the left-hand opponent relation for $\mathrm{DWh}(v)$.

The automorphism group of the graph, limited to act on the player vertices only, gives us the automorphism group of the whist tournament.

For performance reasons we use undirected graphs instead of directed graphs. For Wh(13), as the partnership relation is symmetric, we represent the relation as two pairs of unordered partner pairs instead of four pairs of ordered partner pairs. One may verify that these modifications do not interfere with the computation of the automorphism group of the whist tournament.

We identify the nonisomorphic graphs, that is, the nonisomorphic whist tournaments, by computing an invariant with *nauty* [19]. By examining the automorphism group orders of the whist tournaments and their underlying near resolutions, we may verify that the invariant is powerful enough to distinguish between the whist tournaments with the same underlying resolution.

Classifying the 13-player whist tournaments took less than an hour on a 500 MHz Pentium III computer after computing the 2-$(13, 4, 3)$ near resolutions.

6 Results

There are 10171 nonisomorphic near resolutions, with 10121 nonisomorphic underlying designs. All of these designs are simple, that is, they contain no repeated blocks. The near resolvable 2-$(13, 4, 3)$ designs are grouped by the order of their automorphism group and the number of nonisomorphic near resolutions they admit in Table 1. No design has two near resolutions with automorphism groups of different order, and in the vast majority of cases the order of automorphism group of the near resolutions is equal to the order of the automorphism group of the underlying near resolvable design. The few exceptions are indicated by prefixing superscripts of the form $a \cdot b$, where a indicates the number of near resolvable designs whose resolutions all have automorphism groups of order b.

It is possible to construct a Wh(13) on 414 of the near resolutions. For 399 of them the Wh(13) is unique and has the same automorphism group as the underlying resolution. These are grouped by order of automorphism group in Table 2. The 15 near resolutions with more than one Wh(13) are summarized in Table 3.

The partner relations in a TWh(v) obviously define a Wh(v). By symmetry, the first kind opponent relation and the second kind opponent relation define another two Wh(v). These three Wh(v) have the property that every block of the underlying near resolution is partitioned into pairs of partners once in each of the three possible ways. Since all near resolvable 2-$(13, 4, 3)$ designs are simple, the existence of a TWh(13) would imply the existence of three Wh(13) with the same underlying resolution with no whist game in common. However, as only four near resolutions admit the construction of more than two different but not necessarily nonisomorphic whist tournaments, it can be verified by hand that any two Wh(13) with the same one of those four as the underlying near resolution always have at least one game where the players are split into partner pairs identically. This implies that no TWh(13) exists.

There were six resolutions upon which a DWh(13) could be constructed. All DWh(13) constructed on a particular resolution had the same underlying Wh(13); thus the Wh(13)-orbit is of length 1 under the automorphism group of the resolution. The DWh(13) are summarized in Table 4.

We find the same three $\mathbb{Z}$-cyclic Wh(13) as Finizio [8]. There are three Wh(13) that have an automorphism group whose order is divisible by 13, and their automorphism groups must contain the cyclic group of order 13 as a subgroup. The orbit of a round under the cyclic group must be of length 13, as the group permutes each player in turn to the sitout position. Thus, these three tournaments are $\mathbb{Z}$-cyclic. One of them allows the construction of a DWh(13) (actually four nonisomorphic ones) on it.

Some examples of structures with a large automorphism group are given in Table 5. Naturally the structures underlying the structures of more restrictive types serve as additional examples of structures of less restrictive types. We have used GAP4 version 4r2 [9] in manipulating the structures into a convenient form.

Table 1: Number of near resolvable designs by order of automorphism group and number of nonisomorphic near resolutions

		Order of automorphism group								
		1	2	3	4	6	12	39	156	total
near resolutions	1	9806	$^{4\cdot 1}46$	213	10	1	$^{2\cdot 6}6$		1	10083
	2	19		10	$^{2\cdot 1}2$			1		32
	4	2	$^{2\cdot 1}2$				$^{2\cdot 3}2$			6
total		9827	48	223	12	1	8	1	1	10121

Table 2: Near resolutions with a unique Wh(13)

Automorphism group order	1	2	3	4	6	12	39	156
Number of whist tournaments	344	3	44	2	1	2	2	1

Table 3: Near resolutions with more than one Wh(13)

Number of resolutions	7	1	4	1	1	1
Order of automorphism group of resolution	1	3	3	6	6	12
Wh(13) orbit lengths	1,1	3	1,1	1,2,2	1,1,1	1,2

Table 4: Near resolutions of 2-$(13, 4, 3)$ designs with DWh(13) by automorphism group order

Automorphism group order	4	6	6	12	12	156
DWh(13) orbit lengths	1,1	1,1	1,1,3,3	1,1	1,1,3,3	1,1,3,3

Table 5: Examples of structures

Aut. group order	DWh(13)
156	$\mathbb{Z}_{13}(\{(1,5,12,8),(2,10,11,3),(4,7,9,6)\})$
12	$\mathbb{Z}_{12}(\{(0,3,6,9),(1,4,7,10),(2,5,8,11)\})\cup$ $\mathbb{Z}_{12}(\{(\infty,4,0,7),(1,3,9,2),(6,10,11,8)\}$

Aut. group order	Wh(13)
39	$\mathbb{Z}_{13}(\{(1,2,4,8),(3,6,12,11),(5,9,7,10)\})$
39	$\mathbb{Z}_{13}(\{(1,8,5,10),(2,4,3,11),(6,7,9,12)\})$

Aut. group order	2-$(13, 4, 3)$ near resolution
12	$A_4(\{\{\varepsilon, a^2ba, aba^2, b\}, \{a, ab, ba, bab\}, \{a^2, a^2b, aba, ba^2\}\})\cup$ $A_4(\{\{\infty, \varepsilon, a, b\}, \{a^2bab, aba^2, ba\}, \{a^2, a^2ba, aba, bab\})$.

Aut. group order	near resolvable 2-$(13, 4, 3)$ design
12	$A_4(\{v_1, v_2, e_{13}, e_{34}\}) \cup A_4(\{v_1, v_2, m_{12,34}, m_{13,24}\})\cup$ $A_4(\{v_1, e_{12}, e_{13}, m_{14,23}\}) \cup A_4(\{e_{12}, e_{13}, e_{34}, m_{13,24}\})\cup$ $A_4(\{e_{12}, e_{34}, m_{12,34}, m_{13,24}\})$

In Table 5, the DWh(13) with automorphism group of order 156 may be obtained by a well known Galois Field construction given by Baker. The underlying Wh(13) and the underlying near resolvable 2-$(13, 4, 3)$ design also have an automorphism group of order 156. The point set is understood to be $\mathbb{Z}_{13}$, and the automorphism group consists of permutations of the form $x \mapsto ax + b$, where $0 \neq a \in \mathbb{Z}_{13}$ and $b, x \in \mathbb{Z}_{13}$. Another DWh(13) with automorphism group order 156 may be obtained by mirroring (inverting the left-hand opponent relation in) all three tables of the initial round; if 1 or 2 tables are mirrored, the resulting DWh(13) will have an automorphism group of order 52. By mirroring the second and third table one may obtain the DWh(13) given by Finizio. The second tournament is a DWh(13) with $\mathbb{Z}_{12}$ as the full automorphism group; ∞ is a fixed point under the action of the group. The DWh(13) consists of two orbits of rounds. The orbits have length 1 and 12 respectively. Note that while the group permutes the players to different seats in the first orbit, the relations in the games remain unchanged and we consider the rounds identical. The underlying Wh(13), near resolution and design also have the same automorphism group. Another DWh(13) with the same automorphism group may be obtained by mirroring this DWh(13).

The two Wh(13) in Table 5 were given by Finizio, have automorphism groups of order 39 and have nonisomorphic underlying near resolutions. Both underlying near resolutions also have automorphism group order 39. By mapping the players of the second by $x \mapsto 2x \pmod{13}$, one may verify that both have the unique near resolvable 2-$(13, 4, 3)$ design with an automorphism group of order 39 as the underlying design.

The near resolution and near resolvable design given in Table 5 both have an automorphism group that is isomorphic to the alternating group

$A_4 = \langle a, b\rangle$, where $a = (1, 2, 3)$ and $b = (1, 2)(3, 4)$. The action of the group on the points is different, however. The thirteen points of the near resolution are the twelve elements of A_4 together with ∞, a point fixed by the action of A_4; the group A_4 acts on the points by multiplication from the left. Clearly, A_4 acts transitively on the points other than ∞. The points of the near resolvable design, on the other hand, may be seen as the vertices, edges, and perfect matchings of the complete graph with vertex set $V = \{v_1, v_2, v_3, v_4\}$. Let A_4 act on V in the obvious way, and let the action on the edges and matchings be the induced action. In Table 5, e_{ab} stands for the edge $\{v_a, v_b\}$, and $m_{ab,cd}$ represents the matching $\{e_{ab}, e_{cd}\}$. Here the vertices, edges and matchings form orbits of lengths 4, 6, and 3, respectively.

7 Conclusions

We present a correspondence between the near resolutions of near resolvable balanced incomplete block designs and a particular class of codes. We then use the correspondence to generate the 10171 near resolutions of 2-$(13, 4, 3)$ designs. The near resolutions have a total of 10121 nonisomorphic underlying near resolvable designs, all of which are simple. We find that it is possible to construct a Wh(13) on 414 of the near resolutions, and that there are a total of 421 nonisomorphic Wh(13). Similarly, six near resolutions admit the construction of a DWh(13), and there are 18 nonisomorphic DWh(13). Finally, by examining Wh(13) with the same underlying near resolution, we find that no TWh(13) exists.

Performing a similar classification for $v = 16$ or $v = 17$ would be interesting since $v = 16$ is the smallest case for which the existence of a DWh(v) is open, and $v = 17$ is the only case for which the existence of a TWh(v) remains open. In our estimation, however, considerable improvements to the method in this paper are necessary to make such a classification feasible.

Acknowledgements

The authors wish to thank Clement Lam and Patric Östergård for discussions and numerous helpful comments.

References

[1] R. J. R. Abel and G. Ge, An almost completion for the existence of triplewhist tournaments $TWh(v)$, submitted for publication.

[2] I. Anderson, *Combinatorial Designs: Construction Methods*, Ellis Horwood, Chichester, 1990.

[3] I. Anderson, A hundred years of whist tournaments, *J. Combin. Math. Combin. Comput.*, Vol. 19 (1995) pp. 129–150.

[4] I. Anderson, Whist tournaments, In (C. J. Colbourn and J. H. Dinitz, eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton (1996) pp. 504–508.

[5] R. Baker, Whist tournaments, *Congr. Numer.*, Vol. 14 (1975) pp. 89–100.

[6] F. E. Bennett and X. Zhang, Resolvable Mendelsohn designs with block size 4, *Aequationes Math.*, Vol. 40 (1990) pp. 248–260.

[7] I. A. Faradžev, Constructive enumeration of combinatorial objects, *Colloq. Internat. CNRS*, 260, CNRS, Paris (1978) pp. 131–135.

[8] N. J. Finizio, Orbits of cyclic Wh[v] of Z_N-type, *Congr. Numer.*, Vol. 82 (1991) pp. 15–28.

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

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

[11] G. Ge and L. Zhu, Frame constructions for Z-cyclic triplewhist tournaments, *Bull. Inst. Combin. Appl.*, Vol. 32 (2001) pp. 53–62.

[12] P. B. Gibbons, Computational methods in design theory, In (C. J. Colbourn and J. H. Dinitz, eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton (1996) pp. 718–740.

[13] R. L. Graham and H. O. Pollak, On the addressing problem for loop switching, *Bell System Tech. J.*, Vol. 50 (1971) pp. 2495–2519.

[14] H. Haanpää and P. R. J. Östergård, Classification of whist tournaments with up to 12 players, *Discrete Appl. Math.*, to appear.

[15] P. Kaski and P. R. J. Östergård, There exists no (15,5,4) RBIBD, *J. Combin. Des.*, Vol. 9 (2001) pp. 357–362.

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

[17] R. Mathon and A. Rosa, 2-(v, k, λ) designs of small order, In (C. J. Colbourn and J. H. Dinitz, eds.), *The CRC Handbook of Combinatorial Designs*, CRC Press, Boca Raton (1996) pp. 3–41.

[18] B. D. McKay, `autoson` – a distributed batch system for UNIX workstation networks (version 1.3), Technical Report TR-CS-96-03, Computer Science Department, Australian National University (1996).

[19] B. D. McKay, *nauty* user's guide (version 1.5). Technical Report TR-CS-90-02, Computer Science Department, Australian National University (1990).

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

[21] P. R. J. Östergård, A fast algorithm for the maximum clique problem, *Discrete Appl. Math.*, Vol. 120 (2002) pp. 197–207.

[22] R. C. Read, Every one a winner; or, how to avoid isomorphism search when cataloguing combinatorial configurations, *Ann. Discrete Math.*, Vol. 2 (1978) pp. 107–120.

[23] N. V. Semakov and V. A. Zinov'ev, Equidistant q-ary codes with maximal distance and resolvable balanced incomplete block designs, *Problems of Information Transmission*, Vol. 4, No. 2 (1968) pp. 1–7; translated from *Problemy Peredachi Informatsii*, Vol. 4, No. 2 (1968) pp. 3–10 (Russian).

[24] T. Syrjänen and I. Niemelä, The Smodels system, In (T. Eiter, W. Faber, M. Truszczyński, eds.), Logic Programming and Nonmonotonic Reasoning: Proc. 6th International Conference, LPNMR 2001, Lecture Notes in Artificial Intelligence, Springer, Berlin, 2173 (2001) pp. 434–438.

[25] P. M. Winkler. Proof of the squashed cube conjecture, *Combinatorica*, Vol. 3 (1983) pp. 135–139.

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