MULTIPLY TRANSITIVE PERMUTATION SETS

by

William August Fahle, Jr.

APPROVED BY SUPERVISORY COMMITTEE:

I. Hal Sudborough, Chair

R. Chandrasekaran

Ovidiu Daescu

Sergey Bereg

I dedicate this work to my loving wife Heather, and to my children Scott and Erin.

MULTIPLY TRANSITIVE PERMUTATION SETS

by

WILLIAM AUGUST FAHLE, BS., MS.

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May, 2012

ACKNOWLEDGEMENTS

At the University of Texas at Dallas, many professors and colleagues have assisted my research, none more than Dr. Hal Sudborough. Others include Dr. Linda Morales, Dr. Charles Shields, and fellow students such as Dr. Walter Voit, Dr. Bhadrachalam Chitturi, Zhaobing Meng, and Quan Nguyen. I was greatly inspired by the classes conducted by such professors as Dr. Gopal Gupta, Dr. Kang Zhang, Dr. S. Venkatesan, and Dr. Neeraj Mittal, and committee members Dr. R. Chandrasekaran, Dr. Ovidiu Daescu, and Dr. Sergey Bereg.

My academic endeavors have required many years to complete, and have been supported by many people. My parents, William A. Fahle, Sr., and Patricia Ann Fahle first encouraged me to go off to college. Before I finished my degree, I took a job in computer science, but I was strongly encouraged by Michael Kallet and others at Computer Support Corporation to continue my studies, which I did, at the University of Texas at Dallas. Years later, I went back to school while working at an education company which also highly values education for its employees. They have fully supported my studies from Masters in Computer Science to Ph.D. Without the support of Richard Collins, Sandra Thomas, and Bill Lowrey, none of this would have been possible. I was also supported endlessly by a wife and family who believe in me. Thanks to all of you. And a special thanks to Peter J. Cameron for confirming some ideas.

March, 2012

MULTIPLY TRANSITIVE PERMUTATION SETS

Publication No. _____

William August Fahle, Ph.D.
The University of Texas at Dallas, 2012

Supervising Professor: I. Hal Sudborough

Multiply transitive permutation groups have been studied since the earliest years of group theory. More recently, k-transitive sets of n-length permutations which are not necessarily groups have been shown to have applications in network fault-tolerance, among other things. An important problem is to determine the cardinality of these sets for various n and k.

A set M of permutations on a set χ with cardinality n is called an (n, k)-transet, if for any two k-tuples τ and ρ in χ^k there is a permutation σ in M which maps τ_i to ρ_i. If, for all k-tuples τ and ρ there is a unique permutation σ that maps τ to ρ, then M is called *sharply k-transitive*. The function $F(n,\ k)$ is defined as the cardinality of the smallest set of k-transitive n-length permutations. New lower bounds, upper bounds, and exact values are given for $F(n,\ k)$. A theorem is given showing that $F(n+1,\ k+1) \geq (n+1) \cdot F(n,k)$. Algorithms are presented for finding sharply multiply transitive permutation sets by exhaustive search or failing that,

theorems are given proving lower bounds for related values of $F(n,k)$. A constructive proof is given to show that a non-sharp $(n+1,k)$-transet, say C, can always be created from an (n,k)-transet A and a $(n,k-1)$-transet B, where $|C| \leq |A| + n \cdot |B|$. The theorem $F(n+1,k) \leq F(n,k) + n \cdot F(n,k\text{-}1)$ follows. A table of upper and lower bounds are given for all $F(n,\ k)$ including new values for all cases where $2 < k < n-2$ and $n > 5$.

TABLE OF CONTENTS

LIST OF TABLES

Number Page

LIST OF FIGURES

Number Page

CHAPTER 1

INTRODUCTION

Many scientific and mathematical problems can be represented as problems concerning permutations. For example, in the design and analysis of computer networks, network nodes (computers) can be labeled with permutations and one can define interconnections between nodes when the label of one can be obtained from the label of the other by a specific operation, such as what is called a *transposition* or *reversal* [1,2]. A wide variety of parallel computing and network configurations can be represented in this way, including pancake networks and star networks [3]. Also, in biology, specifically the study of the origin of species, finding minimum length paths of transformations of permutations, each permutation representing a sequence of genetic elements, has often been studied in an attempt to recreate the paths through which species evolved [4]. The number of transpositions of parts of the genome along with inversions of other parts is considered a good measure of how closely related two genomes are in terms of common ancestors or evolution from one to the other.

Since the earliest years of group theory, multiply transitive permutation groups have been studied. Sharply multiply transitive permutation sets imply solutions to wide-ranging problems some of which date back to Euler, others with application to modern parallel computation. The literature is full of applications of such sets to combinatorial design of experiments [5], finite projective geometry, coding theory, mutually orthogonal Latin squares, cryptography, and

analytical software testing [6]. Permutation codes, which are also related to multiple transitivity, can be used in error correction, as in [7].

Existence results about multiply transitive permutation sets are considered in [8], and non-existence results for sharply transitive groups are known for other cases [9], with infinitely many cases for which nothing is known. Far less attention has been paid to this area between where sharply 2,3-transitive groups exist and where they do not exist, though some results are known [10]. We find by relaxing the requirement that sets of permutations meet group axioms, or by relaxing sharpness conditions, unknown cases and even non-existence cases can all be determined, with applications in network fault-tolerance among other areas.

Parallel computer architectures are useful for many large-scale computational applications, for both research and applications of government and private industry. Many different network architectures exist commercially, and many more have been described in the literature. These include processor layouts called hypercubes, meshes, pyramids, trees, star graphs, butterflies, pancake networks and others. Formal definitions of each of these appear, for example, in [11]. Some attributes of interest for parallel architectures are node degree, diameter, and fault tolerance. A star network compares favorably to a hypercube of similar size, when comparing node degree and diameter. Growth in node degree and diameter is sub-logarithmic to network size in the star network but logarithmic in a hypercube [3].

The problem of finding a k-tuple cover, equivalent to finding a k-transitive set of n-permutations, is related to the fault tolerance of a star network [12]. Bounding the size of k-tuple covers (or multiply transitive permutation sets) has been considered in [12,13], and is the main

topic of this thesis. The related problem of covering radius for sets of permutations in coding theory has also been considered in [14].

1.1 Some history

The k-tuple cover problem has had an interesting history, worthy of relating, because that history explains why some of our methods were attempted. This dissertation discusses various approaches and techniques used to find solutions to the k-tuple cover problem, which has far-reaching connections to areas of computer science, mathematics, combinatorics, and experimental design. Knowing the history can shed light on why certain techniques were carried out the way they were, and what is covered in each chapter of this dissertation.

The original k-tuple cover problem was posed by Latifi [12] as a way to determine how many and which node failures in a star graph would cause a smaller star graph to become impossible to recover. The function $F(n,k)$ was defined as the smallest set of destroyed nodes in a star graph of dimension n such that a smaller star graph of dimension $n-k$ can no longer be constructed. As we will see, each case of n and k is almost an independent problem. While the problem was posed correctly and the relationship between star graphs and k-tuple covers is correct, and the construction for primes is correct, one of the proofs in that paper is incorrect (Theorem 2), and furthermore the example given for $F(4,2)$ is not, in fact, a pair cover of degree 4. Since the proof is incorrect, we refer to Latifi's Theorem 2 as

Claim 1.1: Let p be a prime number such that $p \geq n$. The number of nodes to damage all star graphs of degree $n-2$ in a star graph of degree n satisfies the following bound:

$$F(n,2) \leq n(p-1). \text{ [12]}$$

Claim 1.1 leads to a result that 36 is an upper bound on pair covers for permutations of length 6. Also in Latifi's paper is the following conjecture.

Conjecture 1.1: Let p be a prime number such that $p \geq n$. The number of nodes to damage all star graphs of degree $n-k$ in a star graph of degree n for $3 \leq k \leq n-2$, satisfies the following bounds:

$$k!\binom{n}{k} \leq F(n,k) \leq (p-1)k!\binom{n}{k}/(n-k+1). \text{ [12]}$$

In other words we take out the term $(n-k+1)$ from the product and replace it with $p-1$. However, it has proven difficult to discover a pair cover for 6 smaller than 37. In fact, the following conjecture appears to be true.

Conjecture 1.2: $F(6,2) = 37$.

Note that Conjecture 1.2 is incompatible with both Conjecture 1.1 and Claim 1.1. Clearly it directly contradicts Claim 1.1, but also if $F(6,2) = 37$, we will see that $F(7,3)$ is at least 259. However, Conjecture 1.1 would put the *upper* bound of $F(7,3)$ at $7 \cdot 6 \cdot 6 = 252$.

A second paper, from Bein, Latifi, Morales, and Sudborough [13], was published in 2009. When $n = 2$, Bein et al. have a different, correct construction for any non-prime values of n reducing from the next highest prime.

As often happens in theoretical work, two different sets of researchers labor along on similar problems, unaware of each others' work. In this case we were originally unaware of important theorems from group theory researchers [9,10,15]. These results, while known to group theory researchers, had not previously been connected to star graph fault tolerance.

Nonetheless, the results of the research described here is still far beyond what was discovered in [12], and the particular problem of finding the smallest k-transitive set of n-

permutations (not necessarily a group) does not seem to be considered anywhere in the mathematical literature. Many of the group theory results apply directly to our problem, but those mathematical results stand like sharp prime-power islands in an integer sea of possible values for n and k, and are particularly sparse when $k > 3$, whereas our techniques give non-trivial results for *all* valid n and k.

The present research began shortly after Bein et al. was published. Our early approach involved programmatically cutting down a set of permutations known to be a complete cover, such as the set of all permutations of length n, by removing unneeded permutations. The next technique was to combine smaller k-tuple covers in various ways including applying them to pairs of pairs to build k-tuple covers for larger values of n. Later techniques searched for certain semi-orthogonal Latin squares to find exact (sharp) results, or to prove that such results do not exist by enumerating all Latin squares up to isotopy for certain sizes. The structure of these sharp covers, and their existence only for prime powers, led us to the discovery of the connection between k-tuple covers and group theory. All these techniques combined with the group theory results were used to give the currently best-known bounds on $F(n, k)$.

This dissertation is arranged as follows. Preliminaries are in this chapter, along with the existing group theoretical results. Infinitely many cases, where $k = 2$ and $k = 3$, have exact values when n is a prime power, as do $F(11, 4)$ and $F(12, 5)$ [9,16]. Chapter 2 contains theorems proving that sets of permutations making up k-tuple covers have certain symmetries (isotopy classes) that can be exploited, so that exhaustive search for small values of n and k need not check every permutation of the columns and symbols, and one can always assume without loss of generality that the identity permutation is in the set. New theorems strictly limiting the

upper and lower bounds for every n and k are proven in Chapter 3. This theoretical foundation allows other empirical and exact results to be extended to infinitely many cases. Certain earlier techniques involving graph factorization and round-robin tournaments resulted in values for larger n when $k = 3$, 4, or 5, and remain an active area of research particularly because group theory doesn't have much to say when $k = 4$ or $k = 5$. These techniques are described in Chapter 5. Finally, Chapter 6 describes the program which removes redundancies from non-sharp sets to make them smaller when possible, and discusses how all the results were brought together to create **Table 1-3** in the next section.

1.2 Permutation covering sets – preliminaries and new results

Let a *permutation* σ on a set χ be a one-to-one function, mapping the set χ onto itself. The set χ is assumed to have an underlying order. The permutation σ is said to have *degree* n if the cardinality of χ, denoted $|\chi|$, is n. A permutation σ of degree n, then, is a set of n ordered pairs $\langle a, b\rangle$ with $a, b \in \chi$, such that $\sigma(a) = b$ for distinct values of a and b. The *identity* permutation ε is the set of ordered pairs $\langle a, a\rangle$ for all $a \in \chi$. Here, permutations of degree n will operate on the set $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ or $\mathbb{I}_n = \{1, 2, \dots, n\}$, without loss of generality. Turning the ordered pairs on their side, the notation $\sigma = \langle {1\ 2\ 3\ \dots\ n \atop 1\ 3\ 5\ \dots\ \sigma(n)} \rangle$ interpreted as a map from top values to respective bottom values indicates that $\sigma(1) = 1$, $\sigma(2) = 3$, $\sigma(3) = 5$, and n is mapped to $\sigma(n)$. To save space, permutations on sets with a natural ordering such as $\mathbb{Z}_n$ or $\mathbb{I}_n$ are usually written in *passive form*, which is an ordered list of the members from χ such as the following, $\varphi = \langle 1, 3, 5, \dots, \sigma(n)\rangle$. The order of the passive form permutation is considered to be a rearrangement of the items from the identity $\varepsilon = \langle {1\ 2\ 3\ \dots\ n \atop 1\ 2\ 3\ \dots\ n} \rangle$, with the order indicating the

mapping, so that φ is the same as σ above. Clearly, a permutation on $\mathbb{I}_n$ can be converted to a permutation on $\mathbb{Z}_n$ by simply subtracting 1 from each integer of the passive or ordered-pair representation, and in general any permutation on a set χ of cardinality n is equivalent to a permutation on $\mathbb{Z}_n$.

On the set $\chi = \{x_1, x_2, \dots, x_n\}$, the notation $(i\ j)$ denotes a *transposition* which exchanges the i^{th} and j^{th} symbols of χ, and $(1\ j)$ denotes the exchange of the first and j^{th} symbols. More generally, $\sigma = (i\ j\ k)(l\ m)$ represents a *cycle notation* which indicates σ maps x_i to x_j, maps x_j to x_k, and maps x_k to x_i, and transposes x_l with x_m. Note the use of parentheses and the lack of commas for this notation, versus the passive form notation. Any permutation σ in passive form can be written in cycle notation by simply writing the cycles which transform the identity permutation into σ. The identity is written $()$. In cycle notation, changing the underlying set χ does not affect the values of the integers, since they describe positions of the permutation, with numbering starting on the left with 1, then 2, up to the n^{th} position. Thus the cycle notation for a permutation on a set χ of cardinality n remains unchanged for different sets of cardinality n.

Function *composition* on two functions σ and φ, denoted $\sigma \circ \varphi$, involves applying φ to an input, then applying σ to the result. For permutations written in cycle notation, this can be easily obtained by applying φ to the identity permutation ε to create a passive form permutation ρ, then applying σ to ρ instead of ε, giving a final passive-form permutation. For example, on $\mathbb{Z}_5$, the identity $\varepsilon = \langle 0, 1, 2, 3, 4\rangle$. If $\sigma = (1\ 3\ 2)(4\ 5)$ and $\varphi = (2\ 5\ 3\ 4)$, then $\sigma \circ \varphi = \langle 3, 0, 4, 2, 1\rangle$, because $\rho = \langle 0, 4, 3, 1, 2\rangle$. For permutations, function composition is not commutative; that is $\sigma \circ \varphi \neq \varphi \circ \sigma$ in general. Note that the end result is a single permutation, in this case written in cycle notation, $\sigma \circ \varphi = (1\ 4\ 3\ 5\ 2)$.

When a set of permutations on χ in passive form are put into *lexicographic order*, the permutations which start with the lowest value (in χ's natural order) in the first position are listed first. For instance if $\chi = \mathbb{Z}_n$, permutations with 0 in the first position are listed first, then those which start with 1 in the first position are next, and so on, sorting by second and later positions when earlier positions are equal. For example, the set of all permutations on $\mathbb{Z}_3$, $S_3 = \{\langle 0,1,2\rangle,\ \langle 0,2,1\rangle,\ \langle 1,0,2\rangle,\ \langle 1,2,0\rangle,\ \langle 2,0,1\rangle,\ \langle 2,1,0\rangle\}$ is in lexicographic order. Written in cycle notation in the same order, $S_3 = \{(), (2\ 3), (1\ 2), (1\ 2\ 3), (1\ 3\ 2), (1\ 3)\}$.

A k-tuple cover can be described as follows. Let χ be a set of cardinality $n > 2$, and M be a set of permutations on χ. Then M is a *pair cover* on χ if, for every two pairs $\langle x_1, x_2\rangle$ and $\langle y_1, y_2\rangle$ chosen from $\chi \times \chi$ with $x_1 \neq x_2$ and $y_1 \neq y_2$, there is a permutation p in the set M such that $p(x_1) = y_1$ and $p(x_2) = y_2$.

A k-tuple cover is a generalization of a pair cover. Let χ be a set of cardinality n, M be a set of permutations on χ, and k be an integer less than n. The set M is a *k-tuple cover* of *degree* n if, for $1 \leq i \leq k$, for every two k-tuples $\langle x_1, x_2, \dots, x_k\rangle$ and $\langle y_1, y_2, \dots, y_k\rangle$, with $x_i, y_i \in \chi$, there is a permutation $\sigma \in M$ such that $\sigma(x_i) = y_i$, and for $1 \leq j \leq k$, setting $i \neq j$ implies $x_i \neq x_j$ and $y_i \neq y_j$. Since each k-tuple cover is in fact a set of permutations of a particular degree n and the degree is fundamental to the concept, a nomenclature which captures both parameters is desired. Hence we refer to a k-tuple cover M as a *transitive permutation set* with parameters n and k, abbreviated (n, k)-transet. Because M consists of permutations, the underlying set χ can be any set of cardinality n without loss of generality. We are interested in sets of permutations M on n symbols that are k-transitive sets and have minimum cardinality. When $k > 1$, the set M is a *multiply* transitive permutation set. Stated simply, the thesis is that

the techniques and theorems herein improve the known upper and lower bounds on minimum cardinality of multiply transitive permutation sets.

Let $F(n,\ k)$ denote $min\{|M|\big|M\ is\ an\ (n,k)\text{-}transet\}$. For all $n > 1$, and all $1 \leq k < n$, when exact values for $F(n,\ k)$ are not known, upper and lower bounds are given for $F(n,\ k)$. The function *lowerbound(n, k)* $= k!\binom{n}{k}$, in the math literature known as the *falling factorial* and denoted $(n)_k$, has been shown [12] to be a combinatorial lower bound on $F(n,k)$. An $(n,\ k)$-transet will be called *sharp* when its cardinality is equal to $(n)_k$. Clearly every sharp transet is minimal, but it is not true that every minimal transet is sharp.

As a trivial example, the set $S_3 = \{\langle 0,1,2\rangle,\ \langle 0,2,1\rangle,\ \langle 1,0,2\rangle,\ \langle 1,2,0\rangle,\ \langle 2,0,1\rangle,\ \langle 2,1,0\rangle\}$, is a sharp pair cover of degree 3. Let $\mathcal{P}_{n,k}$ be the set of all k-tuples of distinct symbols in $\mathbb{Z}_n$. Then $\mathcal{P}_{3,2} = \{\langle 0,1\rangle,\ \langle 0,2\rangle,\ \langle 1,0\rangle,\ \langle 1,2\rangle,\ \langle 2,0\rangle,\ \langle 2,1\rangle\}$. Each pair in $\mathcal{P}_{3,2}$ can be mapped to each pair in $\mathcal{P}_{3,2}$ by some permutation in S_3. For example, $\langle 0,1\rangle$ is mapped to the elements of $\mathcal{P}_{3,2}$ by the permutations of S_3 in the lexicographic order given.

Finally, note that for an (n,k)-transet, the permutation which maps $\langle x_1, x_2, \dots, x_k\rangle$ to $\langle y_1, y_2, \dots, y_k\rangle$ will also map $\langle x_k, x_{k-1}, \dots, x_1\rangle$ to $\langle y_k, y_{k-1}, \dots, y_1\rangle$ or any other ordering of the coordinates, as long as they are kept in the same parallel order. Thus, we can without loss of generality consider $y_1 < y_2 < \cdots < y_k$ because we can sort the x_i to match.

Some permutation problems can be solved by a search of the entire set of permutations of a particular length. However, the number of permutations of length n is $n!$, and as shown in [17], the value $n! = \theta(n/e)^n$, so this is only feasible for small n. Moreover, in order to find a transet for even small n and k by exhaustive search would require a search for a set of permutations of

size at least $lowerbound(n, k)$, chosen from all permutations of length n. For $F(6,\ 3)$ this means trying at least $\binom{6!}{120} > 600^{120}$ possibilities, so exhaustive search is not feasible.

As mentioned previously, some work has been done on transitive permutation sets and $F(n,\ k)$, including their application to star networks [12]. Many open questions are left, but upper and lower bounds are given for $F(n,\ k)$, and the following theorems are proven in [12,13] for all $n > 1$, and all $1 \leq k < n$.

(1.1) $F(n,\ k) \geq k!\binom{n}{k}$

(1.2) $F(n,\ 1) = n$

(1.3) $F(n,\ n-1) = n!$

(1.4) $F(n,\ n-2) = n!/2$

(1.5) $F(n,\ k-1) \leq F(n,\ k)$

(1.6) $F(n-1,\ k) \leq F(n,\ k)$

(1.7) $F(p,\ 2) = p(p-1)$ for every prime p

(1.8) $F(n,\ 2) \leq 4n(n-1)$

(1.9) $F(n+1, k) \leq F(n,\ k) + 2n \cdot F(n,\ k-1)$

Theorems 1.2, 1.4, 1.5, 1.6, 1.7, and 1.9 are constructive, so that for example a sharp $(p, 2)$-transet can be constructed for any prime p, and a sharp $(n,\ 1)$-transet can be constructed for any $n > 1$. Also, Theorem 1.3 arises from the fact that the symmetric group of degree n is a sharp $(n,\ n-1)$-transet which is easy to generate, and Theorem 1.4 derives from the fact that the alternating group of degree n forms a sharp $(n,\ n-2)$-transet, also easy to generate. Groups

are formally defined in the next section. Theorem 1.9 constructs a non-sharp $(n + 1,\ k)$-transet, but there is no method for always constructing sharp $(n,\ k)$-transet, because for some values of n and k, a sharp set does not exist, as will be seen. The following **Table 1-1** was computed for $F(n,\ k)$ from the above theorems, where bolded values are optimal.

Table 1-1. Published F(n, k) upper bounds

n:	2	3	4	5	6	7	8	9
k=1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
	2	**6**	**12**	**20**	*42*	**42**	*110*	*110*
		3	**24**	**60**	*220*	*692*	*1678*	*4236*
			4	**120**	**360**	*2040*	*8924*	*31272*
				5	**720**	**2520**	*19320*	*110032*
					6	**5040**	**20160**	*181440*
						7	**40320**	**181440**
							k=8	**362880**

More recently, the exact value $F(6,\ 3) = 120$ has been found, and it was shown that $F(6,\ 2)$ has no sharp solution [18]. In fact, $33 \leq F(6,\ 2) \leq 37$ [18]. This dissertation improves the bounds on $F(n,\ k)$ further. Using new techniques and theorems, including sequences of Latin squares, column and row transposition, computer branch-and-bound search, finite group theory, and a new recursive theorem, this table has been improved as seen in **Table 1-3**.

Previously, the best known lower bound was Theorem 1.1. The following theorem giving better lower bounds is proven in Chapter 3:

(3.2) $F\big(n+1,\ k+1\big) \geq (n+1) \cdot F(n,k)$

Additionally for certain values of n and k, the lower bounds for these values of $F(n,\ k)$ have been improved (increased), and coupled with Theorem 3.2 and the Bruck-Ryser Theorem[10], and Lam's result [19], all the following lower bounds are known:

(3.2.1) $F(7,3) \geq 231$. More generally, $F(6+i, 2+i) \geq 33 \cdot (6+i)_i$, for all $i \geq 1$.

(3.2.2) $F(7,4) \geq 841$. More generally, $F(7+i, 4+i) \geq 841 \cdot (7+i)_i$, for all $i \geq 1$.

(3.2.3) $F(9,4) \geq 3025$. More generally, $F(9+i, 4+i) \geq 3025 \cdot (9+i)_i$, for all $i \geq 1$.

(3.2.4) $F(10,4) \geq 5041$. More generally, $F(10+i, 4+i) \geq 5041 \cdot (10+i)_i$, for all $i \geq 1$.

(3.2.5) $F(10,2) \geq 91$. More generally, $F(10+i, 2+i) \geq 91 \cdot (10+i)_i$, for all $i \geq 1$.

(3.2.6) For all $q \equiv 1, 2 (mod\ 4)$, when q is not the sum of two squares, $F(q,2) \geq (q)_2 + 1$. More generally, $F(q+i, 2+i) \geq ((q)_2 + 1) \cdot (q+i)_i$, for all $i \geq 1$.

The latest lower bounds are summarized in the following table:

Table 1-2. F(n, k) latest lower bounds

n:	2	3	4	5	6	7	8	9	10
k=1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	2	**6**	**12**	**20**	*33*	**42**	**56**	**72**	*91*
		3	**24**	**60**	**120**	*231*	**336**	**504**	**720**
			4	**120**	**360**	*841*	*1848*	*3025*	*5041*
				5	**720**	**2520**	*6728*	*16632*	*30250*
					6	**5040**	**20160**	*60522*	*166320*
						7	**40320**	**181440**	*605220*
							k=8	**362880**	**1814400**

The following important new theorem proven in Chapter 3 gives better upper bounds in many cases.

(3.3) $F(n+1,k) \leq F(n,k) + n \cdot F(n,k-1)$

Also, the following new theorems giving exact values are proven in Chapter 3:

(3.4) $F(11,4) = 7920$

(3.5) $F(12,5) = 95040$

(3.6) For $q = p^k$, prime p and $k \geq 1$, $F(q+1,3) = (q+1) \cdot q \cdot (q-1)$

(3.7) For $q = p^k$, prime p and $k \geq 1$, $F(q,2) = q \cdot (q-1)$

The latest known upper bounds are summarized in the following table, where bolded values are exact:

Table 1-3. F(n, k) latest upper bounds

n:	2	3	4	5	6	7	8	9	10	11	12
k=1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	2	**6**	**12**	**20**	*37*	**42**	**56**	**72**	*110*	**110**	*156*
		3	**24**	**60**	**120**	*336*	**336**	**504**	**720**	*1320*	**1320**
			4	**120**	**360**	*1020*	*3265*	*5911*	*7920*	**7920**	*22440*
				5	**720**	**2520**	*9240*	*34502*	*86883*	*95040*	**95040**
					6	**5040**	**20160**	*91714*	*389091*	*1257921*	*2303361*
						7	**40320**	**181440**	*985306*	*4876216*	*18713347*
							k=8	**362880**	**1814400**	*11667460*	*65305836*
								k=9	**3628800**	**19958400**	*148300460*
									k=10	**39916800**	**239500800**
										k=11	**479001600**

Additionally, the following theorems improving the approximation ratio are proven in Chapter 3.

(3.8) For $n > 2$, $(n)_2 \leq F(n,2) \leq (37/25)(n)_2 = 1.48(n)_2$

(3.9) For $n > 3$, $(n)_3 \leq F(n,3) \leq (217/125)(n)_3 = 1.736(n)_3$

These improve the results obtained previously in [13], where the result for 3.8 was $(n)_2 \leq F(n,2) \leq 4(n)_2$, and the result for 3.9 was $(n)_3 \leq F(n,3) \leq 2.667(n)_3$.

1.3 Latin squares

A *matrix* is a rectangular arrangement of integers. For example,

$$A = \begin{bmatrix} 3 & 1 \\ 0 & 2 \\ 3 & 5 \end{bmatrix}.$$

Each horizontal sequence of numbers in a matrix is called a *row* and each vertical sequence of numbers is called a *column*. A matrix with m rows and n columns is an $m \times n$ matrix. To transpose column i and j of a matrix means to exchange the entire column i with the entire column j. The transposition of two rows of a matrix is defined similarly.

A *Latin square* of size n (or n-Latin square) is an $n \times n$ matrix of symbols chosen from the set χ with $|\chi| = n$, such that no symbol appears in any row or column more than once. Thus, each row or column contains all of the symbols from χ. For example, the matrix below represents a Latin square of size 4 taken from $\mathbb{I}_4$.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$$

Figure 1-1. Latin square Λ_0 of size 4.

Equivalence relations on Latin squares can be defined in several ways. Note that one can interchange any pair of rows, any pair of columns, or any pair of symbols and still have a Latin square. Any n-Latin square Λ can be written also as a set, denoted by $S(\Lambda)$, consisting of n^2 ordered triples (a, b, c) with $a, b \in \mathbb{I}_n$ and $c \in \chi$, where a is the row number, b the column number, and c the symbol at row a and column b. For example, **Figure 1-1** can be represented

by the set {(1, 1, 1), (1, 2, 2), (1, 3, 3), (1, 4, 4), (2, 1, 2), (2, 2, 3), (2, 3, 4), (2, 4, 1), (3, 1, 3), (3, 2, 4), (3, 3, 1), (3, 4, 2), (4, 1, 4), (4, 2, 1), (4, 3, 2), (4, 4, 3)}.

Let Λ and Λ' be n-Latin squares of symbols drawn from $\chi = \{x_1, x_2, \dots, x_n\}$. Let θ and π be *column* permutations on the set $\mathbb{I}_n$, and σ be a permutation on χ. We wish to define a binary equivalence relation $\sim$ on n-Latin squares such that $\Lambda \sim \Lambda'$ implies there is a function $g(a, b, c) = (\theta(a), \pi(b), \sigma(c))$ which maps ordered triples to ordered triples such that each ordered triple $(a, b, c) \in \Lambda$ has a matching ordered triple $g(a, b, c)$ in Λ'. Applying θ to the first element in each triple of Λ will interchange the rows in Λ according to the permutation represented by θ, to make the rows of Λ', which if unique to begin with will still be unique afterward by the definition of a permutation. Similarly, applying π to the second element in each triple of Λ will interchange the columns in Λ according to the permutation π to create the columns of Λ'. Applying the function σ to the third element in each triple of Λ will create the symbols of Λ'. Then Λ' is also a Latin square, and is said to be an isotopy of Λ. Many places in the literature [20] define the isotopy relation $\sim$ just as we have and claim it is an equivalence relation; that is, the set of all Latin squares of a given size n is divided into disjoint classes such that every Latin square is in exactly one isotopy class. In the case cited this is done without proof, so proof follows.

To be an equivalence relation, $\sim$ must be reflexive, symmetric, and transitive, in the sense that $\Lambda \sim \Lambda'$ and $\Lambda' \sim \Lambda''$ implies $\Lambda \sim \Lambda''$. Clearly if each of θ, π and σ are the identity permutation, then $\Lambda \sim \Lambda$, so $\sim$ is reflexive. If $\Lambda \sim \Lambda'$, then there are three permutations θ, π and σ which map the triples of Λ to the triples of Λ'. However, each permutation π has an inverse π^{-1} which undoes whatever the permutation does, so we just use θ^{-1}, π^{-1} and σ^{-1} for a function g'

to map all the triples back, so that $\Lambda' \sim \Lambda$, showing that $\sim$ is symmetric. Finally, if $\Lambda \sim \Lambda'$, there are three permutations θ, π and σ to do the mapping from one to the other, and if $\Lambda' \sim \Lambda''$ there are three different permutations θ_2, π_2 and σ_2 to do the mapping. But since permutations are functions, we can use the function composition operator $\circ$ to compose two together to make one permutation with the same effect, so we use the three permutations $\theta_2 \circ \theta$, $\pi_2 \circ \pi$ and $\sigma_2 \circ \sigma$ to map $\Lambda \sim \Lambda''$ directly, proving that $\sim$ is transitive. Hence $\sim$ is an equivalence relation.■

A Latin square Λ is in *reduced* form when the rows are ordered so that the symbols in the first column are in increasing order, and the columns are ordered so that the symbols in the first row are in increasing order. In a standard n-sized Latin square over $\mathbb{Z}_n$ in reduced form, the first row will be the passive form identity permutation $\langle 0, 1, \ldots, n-1 \rangle$. Given any Latin square Λ with first row σ and leftmost column φ, there is a reduced Latin square Λ' such that $\Lambda \sim \Lambda'$ which can be obtained by taking σ^{-1} as the row permutation, φ^{-1} as column permutation, and ε as the symbol permutation. Thus each Latin square is isotopic to a reduced Latin square, so the reduced Latin square makes a good representative for each isotopy class.

Latin squares have been studied extensively for centuries. The numbers of unique isotopy classes for squares from size 2 to size 8 have been published [21]. For example, the Latin squares of size 3 are all in the same isotopy class:

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

Figure 1-2. 3-Latin square isotopy class representative

All Latin squares of size 3×3 are isotopic to the one in **Figure 1-2**. Since each column of any n-Latin square contains every element from $\mathbb{Z}_n$ exactly once, all n-Latin squares are sharply 1-transitive. For 4-Latin squares, there are two disjoint isotopy classes.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

Figure 1-3. Latin 4×4 isotopy class 1

Note that both class representatives are symmetric about the main diagonal. Isotopy class representatives will always be displayed in reduced form.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 3 & 0 & 2 \\ 2 & 0 & 3 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

Figure 1-4. Latin 4×4 isotopy class 2

As the size of Latin squares increase, the number of distinct isotopy classes increases. The following table gives the number of distinct isotopy classes for small size Latin squares.

Table 1-4. Latin square isotopy classes

Latin square	Number of isotopy classes
2	1
3	1
4	2
5	2
6	22
7	564
8	1676267

Let a *shifted* s-Latin square be a Latin square of size $s \times s$ consisting of symbols drawn from a set other than $\mathbb{Z}_s$. For example, an n-Latin square over $\mathbb{I}_n$ is a shifted Latin square. Let $\tau = \langle t_1, t_2, \ldots, t_{k-1} \rangle$ be any $(k-1)$-tuple of symbols from $\mathbb{Z}_n$, and let $s = n - k + 1$. Consider s rows, each starting with τ and preceding one of the rows of a shifted s-Latin square Λ with symbols $\mathbb{Z}_n - \{t_1, t_2, \ldots, t_{k-1}\}$. Each $(k-1)$-tuple coupled with a row of the s-Latin square Λ makes a permutation of length n. Denote the entire collection of s rows by $\tau \cdot \Lambda$.

Now, consider taking each of the $(n)_{k-1}$ different $(k-1)$-tuples from $\mathbb{Z}_n$, say $\tau_1, \ldots, \tau_r$, and forming a sequence of shifted Latin squares $\tau_1 \cdot \Lambda_1, \ldots, \tau_r \cdot \Lambda_r$. The sum of the number of all permutations of length n created is $(n)_{k-1}$ times $(n - k + 1)$, where $(n - k + 1)$ is the size of each Latin square. Of particular interest is when there is a sequence of such Latin squares $\Lambda_1, \ldots, \Lambda_r$ such that the set of all permutations created has the $P(n, k)$ property. As we shall see, there is such a sequence of Latin squares if and only if there is a sharply k-transitive set of permutations of length n. So, to determine if a sharply k-transitive set of permutations exists, we can try all possible sequences of Latin squares for $\Lambda_1, \ldots, \Lambda_r$, taking advantage of the isotopy equivalences to reduce the size of the search.

Now we look at some ways in which Latin squares and k-tuple covers are related to other important objects. A pair of n-Latin squares Λ_1, Λ_2 are called *mutually orthogonal* if and only if whenever the ordered pair $\langle \Lambda_1(i,j), \Lambda_2(i,j) \rangle = \langle \Lambda_1(k,l), \Lambda_2(k,l) \rangle$, then $i = k$ and $j = l$. A set of n-Latin squares $\Lambda_1 \ldots \Lambda_n$ are called *pairwise* mutually orthogonal, or a *set* of mutually orthogonal Latin squares (MOLS) if for all i, j, the square Λ_i is mutually orthogonal with Λ_j. It is known that any time there is a sharply 2-transitive set of permutations of length n, a maximal set of $n - 1$ MOLS of size $n \times n$ can be constructed, and vice versa [22].

The set $S_3 = \{\langle 0,1,2\rangle,\ \langle 0,2,1\rangle,\ \langle 1,0,2\rangle,\ \langle 1,2,0\rangle,\ \langle 2,0,1\rangle,\ \langle 2,1,0\rangle\}$ from above can be used to construct the two MOLS of size 3 as follows. Clearly all six permutations will be required for two Latin squares, so we can start with any one. Starting with $\langle 0,1,2\rangle$, the $\langle 0,2,1\rangle$ overlaps the 0, so instead choose $\langle 1,2,0\rangle$, then $\langle 2,0,1\rangle$. The two squares when overlaid are arranged in such a way that each pair of items will be unique when pair order is considered.

$$\Lambda_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}, \Lambda_2 = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix}.$$

1.4 Groups and designs

Groups are fundamental to the star graph concept, but are also the source of theorems giving exact values for certain $F(n,k)$. Designs are another application of sharply transitive permutation sets, but are also a source of theorems for lower bounds on certain $F(n,k)$. A *group* is mathematical object consisting of a set G and a binary operator $\times$ with the following four axioms.

I. Closure – for any two members a and b of G, $a \times b$ is also in G

II. Associativity – $(a \times b) \times c = a \times (b \times c)$, for all a, b, and c in G

III. Identity – there is an element ε in G such that $a \times \varepsilon = a = \varepsilon \times a$ for all a in G

IV. Invertibility – for every element a in G, there is an element b in G such that $a \times b = b \times a = \varepsilon$

The *order* of a group is the cardinality of the set G. The binary operation $\times$ is sometimes called the *group operation* and the set G by itself is sometimes called the group when the operation is understood. The notation a^{-1} is used to specify the inverse of an element $a \in G$

such that $a \times a^{-1} = \varepsilon$. A set of *generators* for a group G is a subset G' of G so that each member of G is the result of combinations of various members of G' using the group operation. A *subgroup* $H \leq G$ is a subset H of G such that H is still a group under the group operation of G. A *finite group* is a group of finite order; every group discussed herein is assumed to be finite.

By a well-known theorem of Cayley, every finite group is isomorphic to a group known as a *permutation group* with the function composition operation as the group operation and a set of permutations as the elements of the group [23]. Consequently, every group discussed is assumed to be a permutation group. In group theory it is common to write permutations in passive form as rearrangements of $\mathbb{I}_n$, or, more frequently, in cycle notation. The group algorithm program known as GAP [24] was used to test and verify some of our results, and works readily with groups, permutation groups, cycle notation, and passive form permutations of $\mathbb{I}_n$.

The *symmetric group* of *degree* n, known as S_n, is the group $(S_n, \circ)$ where $\circ$ is function composition. Each element of S_n is a permutation of length n, and the group is the set of all permutations of length n, and thus has order $n!$.

Some generators for S_n include:

$S'_n = \{(i\ j) \mid 1 \leq i < j \leq n\}$,

$S''_n = \{(1\ j) \mid j \geq 2\}$,

$S'''_n = \{(1\ 2)(1\ 2\ 3\ \ldots n)\}$.

The *alternating group* of degree n, known as A_n, is the set of all *even permutations*, where an even permutation is one that can be written as an even number of transpositions of two elements. When a permutation is given as cycles of disjoint positions, each cycle can be

transformed into cycles of size two. There are many cycle representations of any given permutation, since one cycle can always undo another, but the number of two-cycles for an even permutation will always be even [23].

Some generators for A_n include:

$A'_n = \{(i\ j\ k) \mid 1 \leq i < j < k \leq n\}$,

$A''_n = \{(1\ 2\ 3)(2\ 3\ \ldots n)\}$.

A permutation group $G \leq S_n$ is called *k-transitive* for an integer $k \leq n$ if for any two k-tuples of symbols $\langle a_1, a_2, \ldots, a_k\rangle$ and $\langle b_1, b_2, \ldots, b_k\rangle$, there exists an element $g \in G$ such that $g(a_i) = b_i$ for $1 \leq i \leq k$. G is called *sharply k-transitive* if g is unique in G. According to [25], it has been long known that the only sharply k-transitive finite groups for $k \geq 4$ are the Mathieu groups $\mathcal{M}_{11}$ and $\mathcal{M}_{12}$ and certain alternating and symmetric groups. By the Classification of Finite Simple Groups, all sharply 2-transitive and 3-transitive groups are also known [16]. As proven in [9], any sharply k-transitive permutation group acting on n points has order $(n)_k$. Clearly the definition of k-transitivity for groups parallels the definition of a k-tuple cover for permutation sets, and sharpness is equivalent for both given the order.

A related concept called a *design* is a set χ together with a collection of subsets of χ called *blocks*, chosen in a way to accomplish a particular purpose, such as experimental design or Hamming codes. Most studied are balanced incomplete block designs, or BIBD's. A *balanced incomplete block design* or 2-(v, b, r, k, λ) design is a set χ of cardinality v and a collection of b blocks of size k such that each symbol of χ appears in r blocks, but no pair of symbols from χ appears in more than λ blocks. The v elements of χ are called *points*. The variable b gives the number of blocks, and k gives the number of points in a block. The repetition factor r gives how

many blocks contain each point, and λ tells how many blocks contain each pair. Since the parameters are not all independent, designs are sometimes designated only by $2\text{-}(v, k, \lambda)$, which determines the remaining parameters. When $b = v$ and $k = r$ it is called a *symmetric* $2\text{-}(v, k, \lambda)$ *design*.

A related concept is a *linear space*, which is a pair $S = (p, \mathcal{L})$ where p is a set of points and $\mathcal{L}$ is a collection of subsets of p called *lines*, satisfying the following axioms:

i. Any two distinct points of S belong to exactly one line of S.

ii. Any line of S has at least two points of S.

iii. There are three points of S not on a common line.

When the set p is finite it is called a *finite linear space*. An *affine plane* is a linear space satisfying one additional axiom:

A. If the point t is not on the line L, then there is a unique line on t missing L.

Real Euclidean 2-dimensional space is an example of an infinite affine plane, but here forward only finite linear spaces are considered. From [15], if S is a linear space on a finite number v of points, S is an affine plane if and only if there is a positive integer n such that $v = n^2$, the number of lines $b = n^2 + n$, each point is on $n + 1$ lines, and each line contains n points. Then n is called the *order* of S. Equivalently, considering lines to be blocks, an affine plane S of order n is a $2\text{-}(n^2, n^2 + n, n + 1, n, 1)$ design, or a $2\text{-}(n^2, n, 1)$ design.

A *finite projective plane* is a linear space P satisfying the following additional axioms:

1. Any two distinct lines have a point in common.

2. There are four points, no three of which are on a common line.

Again from [15], we have that a linear space P is a finite projective plane if and only if there exists an integer $n > 1$ such that there are $v = b = n^2 + n + 1$ points and lines, and each point is on $n + 1$ lines and each line contains $n + 1$ points. The integer n is called the *order* of P. Equivalently, P is a symmetric 2-$(n^2 + n + 1, n + 1, 1)$ design. Furthermore, a projective plane of order n exists if and only if an affine plane of order n exists [15].

The Bruck-Ryser Theorem [10] states that for any finite projective plane of order $n \equiv 1, 2\ (mod\ 4)$, then n must be the sum of exactly two squares of integers. For example, $(6\ mod\ 4) = 2$ but no combination of 0, 1, and 4 can add pairwise to 6, so there is no projective plane of order 6. Also, $(9\ mod\ 4) = 1$, but $3^2 + 0^2 = 9$, so a finite projective plane of order 9 is not excluded, and several exist. However, $(10\ mod\ 4) = 2$, and $3^2 + 1^2 = 10$, but it is known that no finite projective plane of order 10 exists [19]. Order 12 is the smallest for which no result is known. Hence, there is no sharp $(10, 2)$-transet, nor any sharp $(n, 2)$-transet when there is no projective plane of order n by the Bruck-Ryser Theorem.

On the other hand, it is known that for any n where a set of $n - 1$ MOLS exists (or equivalently a sharply 2-transitive n-permutation set exists, what we call a sharp $(n, 2)$-transet) an affine plane of order n and a finite projective plane of order n can be constructed [22]. Hence Lam's result and the Bruck-Ryser Theorem can be restated as follows.

Theorem 1.10 (Bruck-Ryser): For any $n \equiv 1, 2\ (mod\ 4)$, no sharp $(n, 2)$-transet exists unless there exist integers $a, b \geq 0$ such that $a^2 + b^2 = n$. [10]

Theorem 1.11: There is no sharp $(10, 2)$-transet. [19]

Theorem 1.12: A 2-transitive permutation group of degree q and order $q(q - 1)$ exists if and only if there is a prime p and an integer $k > 0$ such that $q = p^k$. [26]

The construction of a projective plane is straightforward. Starting with the two MOLS from the previous section, add a matrix M_1 of points in order.

$$\Lambda_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix},$$

$$\Lambda_2 = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{bmatrix},$$

$$M_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

For the affine plane of order 3, each line has three points. Start with the rows and columns of M_1, {(1, 2, 3), (4, 5, 6), (7, 8, 9), (1, 4, 7), (2, 5, 8), (3, 6, 9)}. Then, let the two MOLS inform the remaining choices from M_1. The top row of Λ_1 is $\langle 0, 1, 2 \rangle$, indicating the next choice starts with the 1, then the 5 of the second row, then the 9 from the third row. The second row of Λ_1 is $\langle 1, 2, 0 \rangle$, indicating (2, 6, 7) from M_1, and finally $\langle 2, 0, 1 \rangle$ chooses (3, 4, 8).

Proceed similarly with Λ_2, and the final affine plane is {(1, 2, 3), (4, 5, 6), (7, 8, 9), (1, 4, 7), (2, 5, 8), (3, 6, 9), (1, 5, 9), (2, 6, 7), (3, 4, 8), (1, 6, 8), (2, 4, 9), (3, 5, 7)}. It can be seen that this fits the definition of an affine plane and is a $2\text{-}(9, 12, 4, 3, 1)$ design.

To construct a finite projective plane of order 3 from the affine plane, we add four new points, A, B, C, and D. We add A to the blocks formed from the original rows of M_1, and B to the blocks formed from the columns, C to the blocks formed by Λ_1 and D to the blocks formed by Λ_2. Finally a block is added containing the new points (A, B, C, D). The final projective plane is {(1, 2, 3, A), (4, 5, 6, A), (7, 8, 9, A), (1, 4, 7, B), (2, 5, 8, B), (3, 6, 9, B), (1, 5, 9, C), (2, 6, 7, C), (3, 4, 8, C), (1, 6, 8, D), (2, 4, 9, D), (3, 5, 7, D), (A, B, C, D)}. It can be seen by viewing the

blocks as lines that this set fits the definition of a finite projective plane of order 3, and is a symmetric 2-$(13, 4, 1)$ design.

The rest of this section describes without proof how to build a particular sharply 3-transitive group, owing largely to [27], and details may require prior experience with abstract algebra.

Two group elements a and x are said to *commute* if $ax = xa$. A group G is *abelian* if for every $a, x \in G$, under the group operation, $ax = xa$. The *center* of a group G, denoted $Z(G)$, is the set of all elements which commute with every element of G. That is, $Z(G)$ is all $z \in G$ such that for every $g \in G$, the element $zg = gz$. The center is always a subgroup of G. For elements $a, x \in G$, the elements are called *conjugate* if there exists $g \in G$ such that as $g^{-1}xg = a$. Conjugacy is an equivalence relation dividing G into equivalence classes so that every element of G is in one class. The conjugacy class of x, $Cl(x) = \{g^{-1}xg \mid g \in G\}$. A subgroup $N \leq G$ is a *normal subgroup* if it is invariant under conjugation by the elements of G. That is, for all $h \in N$ and all $g \in G$, the element ghg^{-1} is still in N. For a normal subgroup $N \leq G$, the notation G/N indicates the *quotient group* built from $\{aN \mid a \in G\}$.

A *ring* is a set G along with two binary operations called *addition* and *multiplication*, such that G is an abelian group under addition, and multiplication distributes over addition. A *field* is a ring which is an abelian group under multiplication, and a finite field has a finite number of elements, so that addition and multiplication are cyclic. A finite field exists, and is unique up to isomorphism, when the number of elements is p^i, for a nonnegative integer i and a prime p. The ring of integers $\mathbb{Z}$ induces a natural field on $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, the integers modulo n. See [27] for a construction of the Galois field $GF(q)$ for $q = p^i$.

Let $V = V(n, q)$ be a vector space of dimension n over $F = GF(q)$. Then V has q^n elements. A *linear automorphism* of $V(n, q)$ is a permutation l of V satisfying

$l(x + y) = l(x) + l(y)$ for all $x, y \in V$,

$l(\lambda x) = \lambda l(x)$ for all $\lambda \in F$ and $x \in V$.

The *general linear group* denoted $GL(V)$ is the group of all linear automorphisms of V. The *projective general linear group* or $PGL(V)$, is defined as $PGL(V) = GL(V)/Z(GL(V))$. It is denoted $PGL(n, q)$ when $V = V(n, q)$.

Theorem 1.13: $PGL(2, q)$ is sharply 3-transitive on $q + 1$ points for all $q = p^k$, with p prime and $k > 0$. [27]

1.5 Graphs

A major application for transets which motivates the study of minimal multiply transitive permutation sets, is fault tolerance in the star graph. In this section we give the formal definition for a star graph. A *directed graph* $G(V, E)$ contains a set $V = \{v_1, v_2, \dots, v_n\}$ of *nodes* and a set $E \in V \times V$ of ordered pairs called *edges*. An edge $e = (v_i, v_j)$ describes a connection considered to be directed from node v_i to node v_j. The *indegree* of a node v_i is the number of edges in E which contain v_i as the second element of the pair, and the *outdegree* of a node v_i is the number of edges in E which contain v_i as the first element of the pair.

An *undirected graph* is a graph which always has the edge $e' = (v_j, v_i)$ whenever it has edge $e = (v_i, v_j)$. The addition of e' to the set E is usually implicit, so that the pairs in E can be considered unordered for an undirected graph. The *degree* of a node v_i in an undirected graph is

the number of edges which contain v_i as either member of the pair. If all nodes of a graph G have the same degree r, G is called an *r-regular graph*.

A *subgraph* $G' = (V', E')$ of a graph $G = (V, E)$ is a graph for which V' consists of a subset of the vertices in V, and E' consists of a subset of the edges in E connecting only those nodes in V', but not necessarily connecting every pair of nodes in V' that are connected in V. For example, if the graph G consists of ($V = \{1, 2, 3\}$, $E = \{(1, 2), (2, 3), (1, 3)\}$), and G′ consists of ($V' = \{1, 2\}, E = \{(1, 2)\}$), G′ is a subgraph of G. G' is said to be a *spanning subgraph* of G if $V' = V$. A subgraph can be either directed or undirected.

The *complete graph* $K_n = (V, E)$ has n vertices in V so that $|V| = n$, and E consists of all pairs which are subsets of V as edges. That is, every node of K_n is connected to every other node exactly once, except no node is connected to itself. K_n is $(n - 1)$-regular.

A *k-factor* is a k-regular spanning subgraph. If the edge set of a graph can be divided into k-factors, such a decomposition is called a *k-factorization* of the graph. A 1-factorization is also called a *factorization*. Factorizations of K_{2m} with integer m can be viewed as schedules for a tournament of $2m$ teams, where each team plays every other team [28]. There are several well-known factorizations of the complete graph [29]. In Chapter 5, tournament scheduling is used to produce small cardinality k-tuple covers.

A *Cayley graph* is a graph that encodes the abstract structure of a group. If G is a group and T is a set of generators for G, the Cayley graph $\Gamma = \Gamma(G, T)$ is a directed graph constructed as follows. Each element of G is assigned a vertex, so that $V(\Gamma) = G$. A directed edge $\langle g_i, g_j \rangle$ exists from $g_i \in G$ to another $g_j \in G$ when a generator $t \in T$ generates g_j from g_i; that is, when

$t \circ g_i = g_j$. The usual definition of a Cayley graph is complicated by edge colors when more than one generator exists in T, but color is unimportant for our purposes.

A *star graph* of degree n is a Cayley graph $\Gamma = \Gamma(S_n, T)$ with $n!$ nodes each labeled with a unique element of the symmetric group of degree n. There is a connection between nodes of a star graph whenever the permutations differ by a transposition of the first element with another element; that is, the set of generators T is $\{(1\ i) \mid 2 \leq i \leq n\}$. Since $(1\ i)$ is its own inverse, the connections are bi-directional and the star graph is usually drawn as an undirected graph.

A star graph of degree n represents the layout of $n!$ physical processors and their connections. If any node in a star graph fails, the star graph of that degree cannot be supported, but there may be smaller degree star graphs still represented by the remaining nodes and connections. As demonstrated in [12], whenever the damaged nodes D of a star network form a k-tuple cover for $n - m = k$, no m-degree embedded star network remains. If D is not a k-tuple cover, an m-degree embedded star network does remain, but it may be necessary to repartition the working nodes in the graph. As such, it is not straightforward whether a given set of nodes can be repartitioned, and it is useful to have metrics to know when it is necessary to try. The remainder of this dissertation is dedicated to finding those metrics.

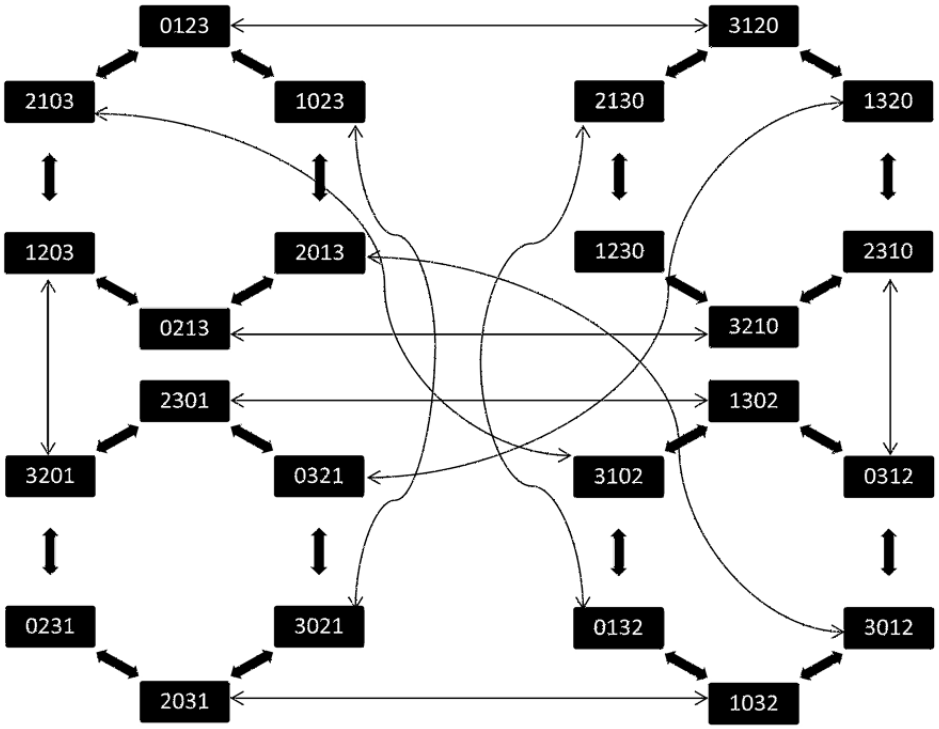

Figure 1-5. Star Graph of degree 4

CHAPTER 2

TRANSITIVE PERMUTATION SET ISOTOPY CLASSES

This chapter will give theorems which allow us to treat certain transitive permutation sets as being the same in important ways. By defining an equivalence relation called *isotopy* on transets for column and symbol permutations, it is possible to assume, for example, that the identity permutation is always in a given set M. Also, when doing exhaustive searches through all possible sets, through symmetry it is only necessary to search one set in each isotopy class. We also give theorems about the structure of the columns of every transet, which are used to prove that the methods involving Latin squares in Chapter 4 are valid.

Let χ be a set of cardinality n, and let M be an (n, k)-transet. Let M' be a set of permutations on χ with $|M'| = |M|$. Define $\sim$ to be a relation between M and M'. We say that $M \sim M'$ if there exists a permutations σ on $\mathbb{I}_n$ and θ on χ, such that permuting the columns of M by θ and composing σ with the permutations of M results in M'. When $M \sim M'$ we say M' is *isotopic* to M.

Claim 2.1: $\sim$ is an equivalence relation.

Proof: Let Ω be the $|M| \times n$ matrix whose rows consist of the permutations of M in passive form in lexicographic order. Represent Ω by triples of row number, column number, and symbol. However since Ω was built from a set, the row number has no specific meaning other than housekeeping, and is only used to identify which sets of n triples belong to a given row.

When θ is applied to the second element of each triple of Ω, and σ is applied to the third element of each triple of Ω, the result is a $|M| \times n$ matrix Ω' whose columns are arranged according to θ and whose symbols are permuted according to σ. If θ and σ are both identity permutations, then $M \sim M$, so $\sim$ is reflexive. If $M \sim M'$, there exist θ and σ which map the columns and symbols of M to M'. However, both θ and σ are functions with inverses, so there exist θ^{-1} and σ^{-1} which map M' to M, so $M' \sim M$, so $\sim$ is symmetric. If $M \sim M'$, there exist θ and σ which map the columns and symbols of M to M'. If $M' \sim M''$, there exist θ' and σ' which map the columns and symbols of M' to M''. By function composition, there exist $\theta' \circ \theta$ and $\sigma' \circ \sigma$ each of which are single permutations mapping M directly to M'', so $M \sim M''$ and $\sim$ is transitive. Thus $\sim$ is an equivalence relation. ■

For example, consider $C = \{\langle 1,3,2\rangle, \langle 2,1,3\rangle, \langle 3,2,1\rangle\}$, a $(3,1)$-transet on $\mathbb{I}_3$, and $C' = \{\langle 1,2,3\rangle, \langle 2,3,1\rangle, \langle 3,1,2\rangle\}$. So, $\Omega = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix}$ and $\Omega' = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$. Then let σ be the identity permutation and $\theta = (2\ 3)$. Represent Ω by the triples {(1, 1; 1), (1, 2; 3), (1, 3; 2), (2, 1; 2), (2, 2; 1), (2, 3; 3), (3, 1; 3), (3, 2; 2), (3, 3; 1)}. Then apply θ to the second element of each triple (swap 3's with 2's), and leave the other two of each triple alone since σ is the identity, and obtain {(1, 1; 1), (1, 3; 3), (1, 2; 2), (2, 1; 2), (2, 3; 1), (2, 2; 3), (3, 1; 3), (3, 3; 2), (3, 2; 1)}. These triples represent Ω', so C' is in the same isotopy class as C, because θ swaps the last two columns of Ω built from C. Additionally, let $C'' = \{\langle 2,3,1\rangle, \langle 3,1,2\rangle, \langle 1,2,3\rangle\}$. Then let σ' be the permutation $(1\ 2\ 3)$ and θ' be the identity permutation, C'' is in the same isotopy class as C and C', since $C' \sim C''$ using σ' and θ' from C', and C'' is in the same isotopy class as C directly with parameters $\sigma' = (1\ 2\ 3)$ and $\theta = (2\ 3)$.

We can also use the isotopy relation to generate new sets which are isotopic to known sets. For example, for M, a $(4,2)$-transet on $\mathbb{Z}_4$, let

$$\Omega = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 1 & 0 & 3 & 2 \\ 1 & 2 & 0 & 3 \\ 1 & 3 & 2 & 0 \\ 2 & 0 & 1 & 3 \\ 2 & 1 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 2 & 1 \\ 3 & 1 & 0 & 2 \\ 3 & 2 & 1 & 0 \end{bmatrix}.$$

Then let σ be a permutation on $\mathbb{Z}_4$ and θ be a permutation on $\mathbb{I}_4$ such that

$$\theta = (1\ 2\ 3\ 4), \text{ and } \sigma = \varepsilon.$$

Each triple of Ω has its second value permuted by θ to determine the rows of Ω', so that

$$\Omega' = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 3 & 1 & 0 \\ 3 & 1 & 2 & 0 \\ 0 & 3 & 2 & 1 \\ 2 & 0 & 3 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 1 & 3 & 2 \\ 1 & 3 & 0 & 2 \\ 3 & 0 & 1 & 2 \\ 0 & 2 & 1 & 3 \\ 1 & 0 & 2 & 3 \\ 2 & 1 & 0 & 3 \end{bmatrix}.$$

The permutation θ moves each column to the left one, and the first column to the last. Once a column permutation has been applied, the rows can be re-sorted into lexicographic order, because Ω' represents a set of permutations, and the order of the permutations is inconsequential.

2.1 Isotopy Class Theorem

Theorem 2.1: For a set χ of cardinality n, if M is an (n, k)-transet on χ and M' is a set of permutations on χ in the same isotopy class as M, then M' is also an (n, k)-transet.

Proof:

Let χ be a set of cardinality n, and let M be an (n, k)-transet with $M = \{\sigma_1, \sigma_2, \dots, \sigma_{|M|}\}$, with $n > k > 1$. Construct a set of permutations M' by taking each permutation σ_i of M and swapping the same two elements α and β of χ in each σ_i to create σ_i', and adding σ_i' to M'. That is, construct a function π mapping χ to χ (so π is a permutation) such that $\pi(\alpha) = \beta$, $\pi(\beta) = \alpha$, and $\pi(x) = x$ for other $x \in \chi$. Using the permutation π, each element σ_i' of M' is obtained from $\pi \circ \sigma_i$. Then M' is also an (n, k)-transet.

Consider two k-tuples of symbols $\tau = \langle t_1, t_2, \dots, t_k \rangle$ and $\rho = \langle p_1, p_2, \dots, p_k \rangle$. If tuple ρ has both α and β in it, let $p_i = \alpha$ and $p_j = \beta$. Consider the k-tuple $\rho' = \langle p_1, p_2, \dots, p_i - 1, \beta, p_i + 1, \dots, p_j - 1, \alpha, p_j + 1, \dots, p_k \rangle$, i.e. with the same elements as ρ but with α and β switched relative to their locations in ρ. That is, $\rho' = \{q_i \in \chi \mid 1 \leq i \leq k \; and \; q_i = \pi(p_i)\}$. Since M is k-transitive, there is a permutation $\sigma \in M$ mapping τ to ρ', by the definition of k-transitivity. Since $\pi(x) = x$ whenever $x \neq \beta$ and $x \neq \alpha$, $\sigma(t_1) = p_1$, $\sigma(t_2) = p_2$, etc., and $\sigma(t_i) = \beta$ and $\sigma(t_j) = \alpha$. Since $\pi(\alpha) = \beta$ and $\pi(\beta) = \alpha$, $\pi\left(\sigma(t_j)\right) = \beta$ and $\pi(\sigma(t_i)) = \alpha$. Hence, if $\sigma' \in M'$ where $\sigma' = \pi \circ \sigma$, then σ' maps τ to ρ. If ρ has neither α nor β in it, let $\rho' = \rho$, and clearly σ' is unaffected and still maps τ to ρ. If ρ only has one of α or β, construct ρ' by exchanging one for the other, and σ' maps τ to ρ. Since this covers all cases for τ as an arbitrary k-tuple, M' is an (n, k)-transet.

For column transpositions, as above let $M = \{\sigma_1, \sigma_2, \ldots, \sigma_{|M|}\}$ be an (n, k)-transet on χ, now viewed as an $|M| \times n$ matrix Ω. Let θ be the permutation on $\mathbb{I}_n$ such that $\theta(i) = j$, and $\theta(j) = i$, and $\theta(a) = a$ otherwise, for $1 \leq i < j \leq n$. Let Ω' be the matrix created by applying θ to the second element of each triple representing Ω. Then the rows of Ω' viewed as the set of permutations M' are also an (n, k)-transet.

The permutation θ will swap elements at column i with elements at column j for all permutations in M. Let $\tau = \langle t_1, t_2, \ldots, t_k \rangle$ and $\rho = \langle p_1, p_2, \ldots, p_k \rangle$ with $k > 1$. Since M is a k-transitive permutation set, there is a permutation $\sigma \in M$ such that $\sigma(t_a) = p_a$ for $1 \leq a \leq k$. Note that if for some b, the symbol $t_b = x_i$, since $\sigma(t_b) = p_b$, the i^{th} position of σ in passive form contains the symbol p_b.

Assume there is a w such that $t_w = x_i$ and a y such that $t_y = x_j$. Let $p_v = \sigma(t_w)$ and $p_z = \sigma(t_y)$. Now consider $\rho' = \langle p_1, p_2, \ldots, p_{v-1}, p_z, p_{v+1}, \ldots, p_{z-1}, p_v, p_{z+1}, \ldots, p_k \rangle$. Since M is a k-transitive permutation set and ρ' is a proper k-tuple of symbols, there is a permutation φ in M with each symbol t_a mapped to each corresponding element of ρ'. In particular, $\varphi(t_y) = p_v$ and $\varphi(t_w) = p_z$. Since $\varphi' \in M'$ was created by swapping the symbols of φ in positions i and j, the permutation φ' will have p_v at position i and p_z at position j. That is, φ' will map τ to ρ. As before, the other cases without such a t_w or t_y are trivial. Since τ and ρ were arbitrary, M' is a k-transitive permutation set on χ.

Thus, if positions i and j are transposed in all permutations in M, the new set of permutations M' will still be an (n, k)-transet, because whatever permutation used to cover the i positions will now cover the j positions, and vice versa.

Any swap of two symbols and any swap of two columns from M to M' retains the $P(n,k)$ property. Clearly these swaps can be composed together one after another as much as needed and will still result in a transet. Since any permutation π can be decomposed into a sequence of transpositions of two objects, M' is an (n,k)-transet. ■

Thus any permutation of columns applied uniformly to all permutations in an $(n,\ k)$-transet, or any permutation of symbols composed with all permutations of an $(n,\ k)$-transet, will result in a set in the same isotopy class which still has the $P(n,\ k)$ property. Obviously, since a transitive permutation set is a set, the order in which those permutations are listed does not matter. So, for any given $(n,\ k)$-transet M considered as an n by $|M|$ matrix Ω, let Ω' be the matrix formed by transpositions of rows or columns of Ω accompanied with a reassignment of all the symbols of Ω to a permutation of those symbols. Then the rows of Ω' are also an (n,k)-transet. The example in the previous section demonstrates a $(3,\ 1)$-transet called C along with sets C' and C'' from the same isotopy class, which each have the $P(3,\ 1)$ property, so that C' is a $(3,1)$-transet, as is C''.

2.2 Properties of pair covers

Let M be a pair cover of permutations of $\mathbb{Z}_n$ of length n, for $n > 2$; i.e., M is an $(n,2)$-transet. Since M maps all pairs of integers from $\mathbb{Z}_n$ to all pairs of integers from $\mathbb{Z}_n$, in particular it must map $\langle 0,1\rangle$ to $\langle 0,1\rangle$. That is to say, there is a permutation σ_1 in M which in passive form starts with $\langle 0,1,\dots\rangle$ in the first two positions. Likewise a permutation σ_2 in M starting with $\langle 0,2,\dots\rangle$ must exist to map $\langle 0,1\rangle$ to $\langle 0,2\rangle$, and $\langle 0,3,\dots\rangle$, and so on up to $\langle 0,n-1,\dots\rangle$ in the first two positions. There must also be a permutation which maps $\langle 0,1\rangle$ to $\langle 1,0\rangle$, and $\langle 0,1\rangle$ to $\langle 1,\ 2\rangle$

up to $\langle 1, n-1, \ldots \rangle$ in the first two positions. This continues up to $\langle n-1, n-2, \ldots \rangle$ in the first two positions. Each of these permutations must be a different σ_i in M, since clearly the permutations differ at least in the first two positions. It is possible in general in a non-sharp transet to have more than one permutation transforming $\langle 0, 1 \rangle$ to a given pair, but at least one must cover each pair. This example discussed the first two positions of each permutation, but clearly this argument applies to any particular two positions chosen from $\mathbb{I}_n$. For example, the permutation which transforms $\langle 0, 2 \rangle$ to $\langle 2, 0 \rangle$ must have $\langle 2, 0 \rangle$ in the first and third position, and must be distinct from the one that converts $\langle 0, 2 \rangle$ to $\langle 2, 1 \rangle$.

Note that the property that one permutation be distinct from another only applies when discussing the same pair of positions for a given set of pairs. For example, the same permutation that maps $\langle 0, 1 \rangle$ to $\langle 2, 3 \rangle$ with $\langle 2, 3 \rangle$ in the first two positions can also map $\langle 0, 2 \rangle$ to $\langle 2, 0 \rangle$ with a 0 in the third position: $\langle 2, 3, 0, \ldots \rangle$.

It is known (Theorem 1.4 in Chapter 1) that $F(n, n-2) = n!/2$. This theorem is due to the fact that all the even permutations of length n are sufficient to map all the tuples of length $n-2$ to each other. By contrast if a tuple of length $n-2$ mapped to another tuple of length $n-2$ required an odd permutation σ, there are two elements α and β in σ which are not part of the mapping, since the length of each tuple is $n-2$. If α and β are interchanged, a single transposition is added so the odd permutation will be even. Since the tuples were arbitrary, the set of all even permutations will map all tuples to all other tuples in this case. For example, the even permutations of length 4 are as follows.

$$\Omega = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 2 & 3 & 1 \\ 0 & 3 & 1 & 2 \\ 1 & 0 & 3 & 2 \\ 1 & 2 & 0 & 3 \\ 1 & 3 & 2 & 0 \\ 2 & 0 & 1 & 3 \\ 2 & 1 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 2 & 1 \\ 3 & 1 & 0 & 2 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

Figure 2-1. Even permutations of length 4

As can be seen in the matrix Ω, each pair is represented in the first two columns; thus $\langle 0, 1 \rangle$ is mapped to all other pairs, for example. Consider what happens if the first two columns are interchanged. Since the operation is a column transposition, the new matrix Ω' represents a set which is in the same isotopy class as **Figure 2-1**, and therefore Ω' is also a $(4, 2)$-transet. However since in each permutation a single pairwise transposition was performed, the resulting permutations are odd. This same reasoning applies in general to any $(n, n-2)$-transet which consists of all the even permutations of length n. That is, the set of odd permutations of length n is also an $(n, n-2)$-transet. Note too that performing the pairwise swap of the first two columns produces the following matrix.

$$\Omega' = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 2 & 0 & 3 & 1 \\ 3 & 0 & 1 & 2 \\ 0 & 1 & 3 & 2 \\ 2 & 1 & 0 & 3 \\ 3 & 1 & 2 & 0 \\ 0 & 2 & 1 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 2 & 0 & 1 \\ 0 & 3 & 2 & 1 \\ 1 & 3 & 0 & 2 \\ 2 & 3 & 1 & 0 \end{bmatrix}$$

Figure 2-2. Odd (4, 2)-transet

Like any $(n, 2)$-transet, **Figure 2-2** has all the pairs in the first two positions represented by distinct permutations. This still applies when interchanging the first two columns of the permutations. Obviously the permutations in **Figure 2-2** can be re-sorted to lexicographic order and each pair is represented. This works no matter which columns are interchanged.

All of the observations that have been made for $(n, 2)$-transets can easily be generalized to (n, k)-transets. For example, if M is an $(n, 3)$-transet on $\mathbb{Z}_n$, each triple of the symbols such as $\langle 0, 1, 2\rangle$, $\langle 0, 1, 3\rangle$, $\langle 0, 1, 4\rangle$, … $\langle 0, 1, n-1\rangle$ must be represented by distinct permutations. Likewise $\langle 0, 2, 1\rangle$, $\langle 0, 2, 3\rangle$, $\langle 0, 2, 4\rangle$ and so on. This continues through tuple $\langle n-1, n-2, n-3\rangle$ in lexicographic order. Again, any ordering of the columns of M will still result in a 3-transitive set, so the same properties will apply, though the permutation which maps $\langle 0, 1, 2\rangle$ to $\langle 0, 1, 2\rangle$, say σ, will not necessarily still map the same tuples as before. For example if $n = 6$ and $\sigma = \langle 0, 1, 2, 5, 4, 3\rangle$ and $\varphi = \langle 1, 0, 2, 4, 5, 3\rangle$ and the first two columns are swapped, the permutation σ will become $\langle 1, 0, 2, 5, 4, 3\rangle$ and will map $\langle 0, 1, 2\rangle$ to $\langle 1, 0, 2\rangle$, while φ will become $\langle 0, 1, 2, 4, 5, 4\rangle$ and will map $\langle 0, 1, 2\rangle$ to $\langle 0, 1, 2\rangle$.

The above examples demonstrate that the set of even permutations are in the same isotopy class with the set of odd permutations with $k = n - 2$ transitivity, and both sets of permutations are sharply $(n-2)$-transitive. However, it is also trivially true that the set of even permutations plus one additional odd permutation is of a separate isotopy class, and clearly still $(n-2)$-transitive, though not sharp. By conjecture this new set is in the same isotopy class as one with any other odd permutation added. There are obviously at least $n!/2$ such separate isotopy classes, but unlike the case with Latin squares, the total number of isotopy classes of transets for any given n and k is a completely open question, even when restricted to sharply k-transitive sets.

By a counting argument, some properties can be determined about a sharply 2-transitive set of permutations. For example the following theorem about the structure of an $(n, 2)$-transet holds.

Theorem 2.2: Given M, a sharply 2-transitive set of permutations on $\mathbb{Z}_n$ containing the identity ε, for each integer $i \in \mathbb{Z}_n$, there are $n-2$ permutations besides ε with exactly one fixed point i in M. The remaining $n-1$ permutations of M have no fixed point.

Proof:

If M contains the identity ε, then no other permutation in M has more than one fixed point. This is clear by sharpness, because only one permutation in M can move a pair $\langle a, b\rangle$ to $\langle a, b\rangle$ and ε will do so by definition of the identity permutation for all a and b.

For any given pair $\langle a, b\rangle$ and $\langle a, c\rangle$ with $b \neq c$, the set M must contain a unique permutation σ such that $\sigma(a) = a$ and $\sigma(b) = c$. Thus σ has a fixed point a, but not b, and hence is not ε. Suppose $a = 0$ and $b = 1$. Then for each element c in $\chi' = \{x \in \mathbb{Z}_n \mid x \neq$

$0\ and\ x \neq 1\}$, a separate permutation σ_i exists such that $\sigma_i(0) = 0$ and $\sigma_i(1) = c$. For example $\sigma_2 = \langle 0, 2, \ldots \rangle$ and $\sigma_3 = \langle 0, 3, \ldots \rangle$. Clearly $|\chi'| = n - 2$, so besides ε, there are $n - 2$ permutations with a fixed point of 0. A similar argument holds for any $a \in \mathbb{Z}_n$, so there are $n - 2$ permutations with a fixed point of a for each $a \in \mathbb{Z}_n$, for a total of $n(n - 2)$.

Counting moved points, for any pairs $\langle a, b \rangle$ and $\langle a, c \rangle$, with $b \neq c$, the permutation σ such that $\sigma(a) = a$ and $\sigma(b) = c$ fixes one point and moves one point. For any pairs $\langle a, b \rangle$ and $\langle c, d \rangle$, with $a \neq b \neq c \neq d$, the permutation σ such that $\sigma(a) = c$ and $\sigma(b) = d$ moves two points with respect to the two pairs involved. For a certain pair such as $\langle 0, 1 \rangle$, there must exist permutations which move $\langle 0, 1 \rangle$ to $\langle 0, b \rangle$ for all b in $\chi' = \{x \in \mathbb{Z}_n \mid x \neq 0\ and\ x \neq 1\}$. Each of these $n - 2$ moves requires one point to move. There must also exist permutations which move $\langle 0, 1 \rangle$ to $\langle a, 1 \rangle$ for all distinct a in χ', again requiring $n - 2$ moves. Finally, there must exist permutations which move $\langle 0, 1 \rangle$ to $\langle a, b \rangle$ for all distinct a, b in χ', each requiring two moved points, giving $2(n - 2)(n - 3)$ moves. The total for $\langle 0, 1 \rangle$ is $2(n - 2) + 2(n - 2)(n - 3) = 2n^2 - 4n + 2$ moves. There are $n(n - 1)/2$ pairs like $\langle 0, 1 \rangle$ which need to be mapped, so the total number of points moved by all permutations in M is $(n(n - 1)/2)(2n^2 - 4n + 2) = n^4 - 3n^3 + 3n^2 - n$ moves.

From what we know of the permutations so far, ε moves no points, and $n(n - 2)$ permutations σ_i each fix one point, with $n - 1$ unknown permutations π_i. Any permutation can affect $n(n - 1)$ pairs, and in a permutation with a fixed point, $n - 1$ of those pairs will have only one point moved. So overall a permutation with one fixed point moves $(n - 1)(n - 1)$ pair points. Totaling all moved points for all $n(n - 2)$ permutations σ_i, we have $n(n - 2)(n - 1)(n - 1) = n^4 - 4n^3 + 5n^2 - 2n$ moved pair points. This leaves a deficit of $n^3 - 2n^2 + n$

pair points still to be moved by $n-1$ remaining π_i permutations. Dividing gives an average of n^2-n pair points to be moved by each π_i, but even with no fixed points a permutation can only move $n(n-1)=n^2-n$ pair points, so each π_i has no fixed points. ■

2.3 Column coverage theorem

Theorem 2.3: Given M, an (n,k)-transet consisting of $|M|$ permutations on $\mathbb{Z}_n$. Let $\mathcal{T}$ be the set of $n!/k!$ distinct k-tuples drawn from the set $\mathbb{Z}_n$. Each k-tuple τ from $\mathcal{T}$ is covered by at least one distinct permutation $\sigma_\tau \in M$ with each of the k symbols from τ in the first k columns of σ_τ.

Proof:

Consider two k-tuples $\tau_1=\langle t_1,t_2,\dots,t_k\rangle$, $t_i\in\mathbb{Z}_n$, and $\rho=\langle 0,1,\dots,k-1\rangle$. Since M is k-transitive, there is a permutation $\sigma\in M$ mapping ρ to τ_1 such that $\sigma(a)=t_{a+1}$ for all $0\le a<k$. That is, for each t_i, the i^{th} position of $\sigma=t_i$ for all $1\le i\le k$. Now consider a different k-tuple $\tau_2=\langle y_1,y_2,\dots,y_k\rangle$, such that there exists at least one j in $1\le j\le k$ where $t_j\neq y_j$. Tuple τ_2 is a valid k-tuple and ρ is too, so M will have a permutation φ which maps ρ to τ_2. This means that for each y_i in τ_2, the i^{th} position of $\varphi=y_i$. Since $t_j\neq y_j$, σ differs from φ at least at position j. Thus σ and φ are distinct permutations. Note there are n ways to choose the first element of a k-tuple of $\mathbb{Z}_n$, and having chosen the first element, there are $n-1$ ways to choose the second, on to $n-(k-1)$ ways to choose the k^{th} and final element of a k-tuple. Thus the number of such k-tuples is at least $n!/k!$. This is true for all transets, sharp or not. Since τ_1 was chosen arbitrarily, the theorem holds for all k-tuples of symbols. ■

Additionally, if M is sharp, there is no room for more than one permutation to map the same k-tuple to another k-tuple, so φ is the only permutation in M which starts with all the y_i, and σ is the only permutation which starts with all the t_i.

The isotopy class Theorem 2.1 tells us that we can jumble around the columns and symbols of a transitive permutation set and it will remain a transitive permutation set. Theorem 2.2 gives us insight into the structure of sharply transitive permutation sets. The column coverage Theorem 2.3 tells us that when a sharply transitive permutation set in passive form is arranged lexicographically, we can see certain structures in the columns as they line up. The column structure will be used in Chapter 4 regarding Latin squares.

CHAPTER 3

BOUNDING MINIMAL TRANSITIVE PERMUTATION SETS

This chapter explores sharply transitive permutation sets, and gives an application of group-theoretical results for finding infinitely many sharply transitive sets, thus giving infinitely many exact values for $F(n,k)$. A new recursive construction improves the upper bound for all $F(n,k)$ for which a sharply transitive set has not been found. The diagonal sharpness theorem allows us to get lower bounds on infinitely many values of $F(n+i,k+i)$ whenever it can be proven that no sharp (n,k)-transet exists. We are also able to obtain a measure for how good our bounds are as a function of the lower bound.

Let M be a pair cover for permutations of length n on $\mathbb{Z}_n$. There are $\binom{n}{2}$ ways to choose two symbols from $\{0,\ 1,\ \ldots,\ n-1\}$ when the symbols are different from one another. But each pair of symbols can be ordered in two different ways, so there are a total of $2\binom{n}{2}$ pairs covered by M. However, mapping $\langle a,b\rangle$ to $\langle c,d\rangle$ is the same as mapping $\langle b,a\rangle$ to $\langle d,c\rangle$, so there are only $\binom{n}{2}$ ways to choose the second pair. Looking at each permutation in M, note that the symbols at each position are fixed. That is, for $\sigma \in M$, $\sigma(0)$ paired with each of $\{\sigma(1),\sigma(2),\ldots,\sigma(n-1)\}$ maps $n-1$ pairs to $n-1$ other pairs. Then $\sigma(1)$ paired with the symbols other than $\sigma(0)$ (since the pair $\langle\sigma(0),\sigma(1)\rangle$ was already accounted for) covers $n-2$ pairs, and so on. For example, the permutation $\sigma=\langle 0,3,1,2\rangle$ maps $\langle 0,1\rangle$ to $\langle 0,3\rangle$, and $\langle 0,2\rangle$ to $\langle 0,1\rangle$ and $\langle 0,3\rangle$ to $\langle 0,2\rangle$. Starting from the second symbol, σ maps $\langle 1,2\rangle$ to $\langle 3,1\rangle$ and $\langle 1,3\rangle$ to $\langle 3,2\rangle$. Finally, σ maps $\langle 2,3\rangle$ to

$\langle 1, 2\rangle$. Each permutation can cover $\binom{n}{2}$ pairs, and each pair needs to be covered in $\binom{n}{2}$ different pairs of positions, but the order of pairs matter so there are $2\binom{n}{2}$ pairs. Each pair must be covered in each of pair of positions. If each pair in each particular pair of positions is covered by exactly one permutation in M, then M is a sharp pair cover of size $(n)_2 = n(n-1)$. This same reasoning can be extended to the (n, k)-transet case where the sharp cover size will be $(n)_k$. For example, it is known that there exists a $(6, 3)$-transet of size 6·5·4=120, which is sharp. All of the preceding are results from[12,13,18].

3.1 Diagonal sharpness theorem

Theorem 3.1: If M is a sharp (n, k)-transet on $\mathbb{Z}_n$, $n > k > 1$, let the set M' be the set of permutations in M which end with $n-1$. Remove the symbol $n-1$ from every permutation in M'. Then M' is a sharp $(n-1, k-1)$-transet on on $\mathbb{Z}_{n-1}$.

Proof:

Let $\tau' = \langle t_1, t_2, \dots, t_{k-1}\rangle$, and let $\rho' = \langle p_1, p_2, \dots, p_{k-1}\rangle$ be $(k-1)$-tuples from $\mathbb{Z}_{n-1}$. We need to show that there is a unique permutation σ' in M' which maps τ' to ρ'. Consider the k-tuples $\tau = \langle t_1, t_2, \dots, t_{k-1}, n-1\rangle$, and $\rho = \langle p_1, p_2, \dots, p_{k-1}, n-1\rangle$ in $\mathbb{Z}_n$, where τ is just τ' with an extra co-ordinate with value $n-1$, and ρ is just ρ' with an extra co-ordinate of $n-1$ at the end. Since M is an (n, k)-transet, there is a permutation σ in M which maps τ to ρ. Clearly $\sigma(n-1) = n-1$, but also σ maps τ' to ρ'. So, the permutation σ' created from σ by removing $n-1$ from its domain maps τ' to ρ'.

Now suppose some other permutation φ' of M' also maps τ' to ρ'. Add the symbol $n-1$ to the domain and let φ be a permutation which maps $n-1$ to $n-1$, and $\varphi(i) = \varphi'(i)$ for all i

in $\mathbb{Z}_{n-1}$. Then φ maps τ to ρ, and clearly φ is in M or φ' would not be in M'. However σ is also in M, and σ also maps τ to ρ, contradicting the fact that M is an (n,k)-transet. Thus, there is no other permutation in M' which maps τ' to ρ'. Finally, since τ' to ρ' were arbitrary $(k-1)$-tuples, it follows that M' is a sharp $(n-1,k-1)$-transet. ■

Corollary 3.1: If no sharp (n,k)-transet exists, neither does a sharp $(n+1,k+1)$-transet.

Proof by contradiction: If a sharp $(n+1,k+1)$-transet M exists, then by Theorem 3.1 a sharp (n,k)-transet can be constructed from M. But no sharp (n,k)-transet exists. ■

Theorem 3.2: $F(n+1,k+1) \geq (n+1) \cdot F(n,k)$.

Proof: It is easy to see the generalization of Theorem 3.1 to any set of permutations with $n-1$ in a particular position, rather than just the last one. Since an $(n+1,k+1)$-transet on $\mathbb{Z}_n$ can be partitioned into $n+1$ sets which are defined by the symbol n in each of the $n+1$ positions, and even the smallest of these sets with the n removed is an (n,k)-transet, the result follows. ■

Repeated applications of Theorem 3.2 along with the known lower bound for $F(6,2)$ gives result 3.2.1 in Chapter 1.

3.2 Linking finite group theory to transitive permutation sets

From finite group theory[16], the concept of multiply transitive permutation groups has been studied extensively. It is sometimes helpful to think of permutations in cycle notation rather than in their passive representation as a list of integers. Cycle notation captures which positions are mapped to which other positions without regard for the underlying symbols. This can also

help when visualizing the composition of permutation functions, because the first permutation may operate on the identity list of integers, but the second one will operate on the result of the first arrangement. The *image* of a permutation on a set of objects is just the list of objects in the order imposed upon them by the permutation function. The image of an n-length permutation on the set $\mathbb{Z}_n$ is just the normal passive form of the permutation as encountered so far. In cycle notation, the integer positions are always 1-based. For example, if $n = 4$ and in cycle notation $\sigma = (2\ 3\ 4)$, the image of σ on $\mathbb{Z}_4$ is $\langle 0, 2, 3, 1 \rangle$. The image of σ on $\mathbb{I}_4$ is $\langle 1, 3, 4, 2 \rangle$. In the second case, it is easy to think of the cycle of σ mapping 2 to 3, 3 to 4, and 4 to 2.

It is customary in finite group theory to write permutations in cycle notation and to consider the image of those permutations on $\mathbb{I}_n$, because it is easier to convert back and forth between cycle notation and passive form. The definition of a transitive permutation set is easily adapted to have the k-tuples of symbols taken from the set $\mathbb{I}_n$, and to represent the transet as a set of permutations on $\mathbb{I}_n$. For convenience, tuples and permutations on $\mathbb{I}_n$ are represented with parentheses, so that a k-tuple of symbols from $\mathbb{I}_n$ $\tau = (t_1, t_2, \ldots, t_k)$. When listed without commas, such a construct still represents cycle notation.

As a reminder, a permutation group $G \subseteq S_n$ for an integer $k \leq n$ is called *k-transitive* if for any two k-tuples of symbols (a_1, a_2, …, a_k) and (b_1, b_2, …, b_k), there exists an element $g \in G$ such that $g(a_i) = b_i$ for $1 \leq i \leq k$. The group G is called *sharply k-transitive* if g is unique in G. By counting arguments presented earlier, this definition of sharp k-transitivity is the same as the one for transitive permutation sets.

From finite group theory, all sharply k-transitive permutation groups are known, and each sharply k-transitive permutation group of degree n necessarily has order $(n)_k$, so each maps to a

sharply transitive permutation set[9]. In all cases the sharply 2-transitive groups are on permutations of length q for q a prime power[26], and all sharply 3-transitive groups are on permutations of length $q+1$ [9]. The alternating group A_n is sharply $(n-2)$-transitive, and the symmetric group S_n is sharply $(n-1)$-transitive. There is only one other sharply 4-transitive group, with $n=11$, and one other sharply 5-transitive group, with $n=12$ [16]. The question of whether there are other sharply transitive permutation *sets* is still open, which simply removes the requirement that the permutations are a subgroup of a permutation group. Any set of same-length permutations is already a subset of the symmetric group, so relaxing the subgroup requirement means that subsets which are not closed under composition are allowed. Since the odd permutations are a sharp $(n, n-2)$-transet, clearly not all sharp permutation sets are groups (identity permutation is not odd). However, there have been no counterexamples to the following conjecture, and for $k \geq 4$ it is true if it can be shown that every sharp transet is invertible[30].

Conjecture 3.1: All sharp (n,k)-transets with $k=3$ have $n=p^i+1$ for some prime p and integer $i \geq 1$.

Proving this conjecture would solve several open problems in the theory regarding sharply k-transitive permutation sets. This is an area of active research.

3.3 Recursive transitive permutation set theorem

A new theorem previously unpublished allows us to generate an $(n+1,\ k)$-transet from a smaller $(n,\ k)$-transet and an $(n,\ k-1)$-transet, using only n copies of the $(n,k-1)$-transet and one copy of the (n,k)-transet.

Theorem 3.3: $F(n+1,k)\ \leq\ F(n,k)\ +\ n \cdot F(n,k-1)$

Proof: Let A be a set of permutations on $\mathbb{Z}_n$ which are k-transitive, and let B be a set of permutations on $\mathbb{Z}_n$ which are $(k-1)$-transitive. Create a new set C by taking each permutation σ from A and adding the symbol n to the end of it to create a new permutation σ' on $\mathbb{Z}_{n+1}$ and putting σ' into C. Then, take each permutation φ from B and add the symbol n to the end of it to create a new permutation φ' on $\mathbb{Z}_{n+1}$, then create n new permutations φ_i' in C by exchanging each i^{th} element of φ' with the element n and adding those new permutations φ_i' to C (but not adding each φ' itself to C). The set C is a k-transitive set of permutations of $\mathbb{Z}_{n+1}$, as shown below. Consequently such a set C can be created from any (n,k)-transet and $(n,k-1)$-transet, so $F(n+1,k) \leq F(n,k) + n \cdot F(n,k-1)$.

To see that C is k-transitive for $\mathbb{Z}_{n+1}$, consider any two k-tuples $\tau = \langle t_1, t_2, \ldots, t_k \rangle$ and $\rho = \langle p_1, p_2, \ldots, p_k \rangle$ chosen from $\mathbb{Z}_{n+1}$. Without loss of generality, $p_1 < p_2 < \ldots < p_k$ because each t_i can be sorted to match this order, and for any $\sigma \in C$, the relation $\sigma(t_i) = p_i$ is preserved. There are several cases to be considered.

Case 1: The element n is not in τ, and n is not in ρ.

Let $\tau = \langle t_1, t_2, \ldots, t_k \rangle$, and $\rho = \langle p_1, p_2, \ldots, p_k \rangle$. Since n is not in τ and n is not in ρ, τ and ρ represent k-tuples of symbols chosen from $\mathbb{Z}_n$. Since A was originally k-transitive on $\mathbb{Z}_n$, by definition there exists a permutation π in A such that for each $1 \leq i \leq k$ element $\pi(t_i) = p_i$. When C was created, every such permutation from A had n added to the end to create a permutation π' in C. Since adding an n to the end leaves the other elements of π in their original positions, for each $1 \leq i \leq k$, we have $\pi'(t_i) = p_i$. Thus C maps any such τ and ρ for case 1.

Case 2: The element n is not in τ, but n is in ρ.

Let $\tau = \langle t_1, t_2, \ldots, t_k \rangle$, and $\rho = \langle p_1, p_2, \ldots, p_k \rangle$. Since n is in ρ and ρ is in order, n is at the end as p_k. Thus consider the $(k-1)$-tuple $\tau' = \langle t_1, t_2, \ldots, t_{k-1} \rangle$, and $\rho' = \langle p_1, p_2, \ldots, p_{k-1} \rangle$. Since B is $(k-1)$-transitive on $\mathbb{Z}_n$ and τ' and ρ' are chosen from $\mathbb{Z}_n$, there is a permutation φ in B which maps each element of τ' to ρ'. When C was formed, φ' was created by first adding n to the end of φ, and then each element of φ' was swapped with that n to create n new permutations φ_i' in C. Since t_k is not n, t_k is in $\mathbb{Z}_n$ and thus $\varphi(t_k)$ is defined. Let $\beta = \varphi(t_k)$. Clearly β is not in ρ, because β is not n and $n = p_k$, and all the other elements of p_i are already $\varphi(t_i)$ for some element t_i with $i < k$. Let $j = t_k + 1$, so that the j^{th} element of φ is β. Then, when φ_j' is created by swapping the unused element β in position j with the n at the end of φ', β is moved to the end of φ_j' and $\varphi_j'(t_k) = n$. But this move does not affect any other symbols from φ, so $\varphi_j'(t_i) = p_i$ for all $1 \leq i \leq k$. Thus $\varphi_j' \in C$ maps τ to the tuple ρ for case 2.

Case 3: The element n is in τ, but n is not in ρ.

Let $\tau = \langle t_1, t_2, \ldots, t_k \rangle$, and $\rho = \langle p_1, p_2, \ldots, p_k \rangle$. Since n is in τ, there is some j such that $t_j = n$. However, p_i is not n for any i. So, let τ' be the $(k-1)$-tuple formed by removing t_j from τ, $\tau' = \langle t_1, t_2, \ldots, t_{j-1}, t_{j+1}, \ldots, t_k \rangle$, and let ρ' be the $(k-1)$-tuple of positions formed by removing p_j from ρ, $\rho' = \langle p_1, p_2, \ldots, p_{j-1}, p_{j+1}, \ldots, p_k \rangle$. Then τ' and ρ' are $(k-1)$-tuples of symbols from $\mathbb{Z}_n$. Since B is $(k-1)$-transitive on $\mathbb{Z}_n$, there is by definition a permutation φ in B which maps τ' to ρ'. Since the element p_j is not in ρ', there is no element t_i of τ' such that $\varphi(t_i) = p_j$. But $p_j < n$, so there is some β with $0 \leq \beta \leq n-1$, such that $\varphi(\beta) = p_j$. Let $b = \beta + 1$ so that the b^{th} symbol of φ is p_j. When C was formed, φ' was created by adding n to the end of φ, and then each element of φ' was swapped with that n to create a new permutation

φ_i' in C. Consider the new permutation φ_b' in C created from φ' when n was swapped with p_j. Since only the position b is affected by that swap, all the other positions from $\varphi_b'(t_i) = p_i$ when $i \neq j$. Also, the last position of φ_b' now has p_j, meaning $\varphi_b'(n) = p_j$, and $n = t_j$, so φ_b' maps τ to ρ for case 3.

Case 4: The element n is in τ, and n is in ρ, but n is not mapped to n.

Let $\tau = \langle t_1, t_2, \dots, t_k \rangle$, and $\rho = \langle p_1, p_2, \dots, p_k \rangle$. Since n is in ρ and ρ is in order, n is at the end as p_k, but $t_k \neq n$. Let $j < k$ such that $t_j = n$. Let $\tau' = \langle t_1, t_2, \dots, t_{j-1}, t_k, t_{j+1}, \dots, t_{k-1} \rangle$, and $\rho' = \langle p_1, p_2, \dots, p_{k-1} \rangle$. Since B is $(k-1)$-transitive on $\mathbb{Z}_n$ and τ' and ρ' are $(k-1)$-tuples chosen from $\mathbb{Z}_n$, there is a permutation φ in B which maps τ' to ρ'. When C was formed, φ' was created by adding n to the end of φ, and then each element of φ' was swapped with that n to create n new permutations φ_i' in C. Since t_k is in τ', $\varphi(t_k) = p_j$ and for $i \neq j$ and $i \neq k$, $\varphi(t_i) = p_i$. Thus, when φ_β' is created by swapping the element p_j with the n at the end of φ', $\varphi(t_k)$ becomes n, which is p_k, and p_j is moved to the end of φ_β', so that $\varphi_\beta'(n)$ is p_j. But this move does not affect any other symbols from φ, so for $i \neq j$ and $i \neq k$, $\varphi_\beta'(t_i) = p_i$. Thus φ_β' maps τ to ρ for case 4.

Case 5: The element n is in τ, mapped to n in ρ.

Let $\tau = \langle t_1, t_2, \dots, t_k \rangle$, and $\rho = \langle p_1, p_2, \dots, p_k \rangle$. Since n is mapped to n, and ρ is sorted to have all positions in order, n must be t_k and n must be p_k. Let $\tau' = \langle t_1, t_2, \dots, t_{k-1} \rangle$, and $\rho' = \langle p_1, p_2, \dots, p_{k-1} \rangle$. Then τ' and ρ' are $(k-1)$-tuples of symbols from $\mathbb{Z}_n$. Since A is k-transitive on $\mathbb{Z}_n$, it is also a $(k-1)$-transitive on $\mathbb{Z}_n$. Thus there is some permutation π in A which maps τ' to ρ'. To create C, each permutation π had n added to the end to create π' in C.

This does not affect the other positions, so $\pi'(t_i) = p_i$ for all $1 \leq i < k$. But n is at the end of π', so $\pi'(n) = n$, as required. Thus $\pi' \in C$ maps τ to ρ for case 5.

Thus, C is an (n, k)-transet for $\mathbb{Z}_{n+1}$. ■

The result of this theorem is constructive clearly. Also, there is no requirement about the order of the rows and columns of A or B when creating C, except that all isotopic changes be made to the columns and symbols of A or B before starting the process of creating C. There is also no requirement that A or B be sharp. All that is required is that A and B are k-tuple and $(k-1)$-tuple transets on the same size permutations, respectively. Thus from a known set of transets, new transets can be built. It is already known for all n that S_n is an $(n, n-1)$-transet and A_n is an $(n, n-2)$-transet. Also an n-Latin square is easily generated and is an $(n, 1)$-transet. The Mathieu groups $\mathcal{M}_{11}$ and $\mathcal{M}_{12}$ can generate an $(11, 4)$-transet and a $(12, 5)$-transet respectively, and can be generated in GAP from the following generators.

$\mathcal{M}_{11}$: (1 10)(2 8)(3 11)(5 7), (1 4 7 6)(2 11 10 9).

$\mathcal{M}_{12}$: (1 2 3)(4 5 6)(7 8 9), (2 4 3 7)(5 6 9 8), (2 9 3 5)(4 6 7 8), (1 10)(4 7)(5 6)(8 9), (4 8)(5 9)(6 7)(10 11), (4 7)(5 8)(6 9)(11 12).

Also in GAP the projective linear group created by $PGL(2, q)$ with q a prime power, will create a sharply 3-transitive group of order $(q+1)q(q-1)$ which is a sharp $(q+1, 3)$-transet. Iterating the images of all the permutations from that group which fix $q+1$, and removing the $q+1$ from each, will create a sharp $(q, 2)$-transet.

Theorem 3.4: $F(11,4) = 7920$.

Proof: Follows from the Mathieu group $\mathcal{M}_{11}$ being sharply 4-transitive. ■

Theorem 3.5: $F(12,5) = 95040$.

Proof: Follows from the Mathieu group $\mathcal{M}_{12}$ being sharply 5-transitive. ■

Theorem 3.6: For prime p and $k \geq 1$, $F(p^k + 1, 3) = (p^k + 1)p^k(p^k - 1)$.

Proof: Follows from Theorem 1.13, that the projective linear group $PGL(2, q)$ is sharply 3-transitive for prime powers. ■

Theorem 3.7: For prime p and $k \geq 1$, $F(p^k, 2) = p^k(p^k - 1)$.

Proof: Follows from Theorem 3.6 and Theorem 3.1, also from Theorem 1.12. ■

Note that Theorem 1.6 from Chapter 1 states that $F(n-1, k) \leq F(n, k)$. Observe that for $k = 2$ for any prime power q, Theorem 3.7 says that $F(q, 2) = q(q-1)$. Also $F(q, 1) = q$. Thus by Theorem 3.3 $F(q+1, 2) = q(q-1) + q \cdot q = 2q^2 - q$. But also for the next prime power above q, say $q + r$, Theorem 3.7 states that $F(q + r, 2) = (q + r)(q + r - 1) = q^2 + 2qr - q + r^2 - r$. But by Theorem 1.6, $F(q + 1, 2) \leq F(q + r, 2)$ when $r \geq 1$. So if there is a pair of prime powers far enough apart that $r \cong q$, Theorem 3.3 will give a better result than Theorem 3.7, otherwise Theorem 1.6 will let Theorem 3.7 win out. However, by [31], it is known that for $q > 24$, r is no greater than $q/5$ even when restricted to primes, never mind prime powers which are necessarily at least as close together as primes. Thus to improve the upper bounds further for $F(n, 2)$ or $F(n, 3)$, it will be necessary to search exhaustively for solution sets or find a better recursive theorem. However the picture is better for values of $k > 3$, where Theorem 3.3 gives infinitely many improvements on the upper bounds.

The following two theorems are results related to the approximation ratio for $F(n, 2)$ and $F(n, 3)$. Given the combinatorial lower bound of $n(n-1)$ and $n(n-1)(n-2)$ for $F(n, 2)$ and $F(n, 3)$ respectively, and given the upper bounds implied by the theorems above, it is possible to compute the ratio between the two extremes to determine how far off the answer will be in the

worst case. Since we know $F(n,2)$ is sharp when n is prime, and we know the intervals between primes, we have the following.

Theorem 3.8: For $n > 2$, $(n)_2 \leq F(n,2) \leq (37(n)_2)/25 = 1.48(n)_2$

Proof:

For any q a prime power from the set $\{3,4,5,7,9,11,13,16,17,19,23,25\}$, by Theorem 3.7, $F(q,2) = (q)_2$. For any $2 < n < 25$ not in the set, let q be the next higher prime power. Hand calculation reveals $(q(q-1))/(n(n-1))$ does not exceed the bound in any of these cases. For example when $n = 14$, the next prime power $q = 16$, so $(q(q-1))/(n(n-1)) = (16(16-1))/(14(14-1)) \cong 1.318 < (37/25) = 1.48$. Then, from [31], for any $n > 24$, there is a prime p such that $n \leq p \leq 6n/5$. By Theorem 1.1 and 1.7 we have that

$$(n)_2 \leq F(n,2) \leq F(p,2) = p(p-1)$$

$$\leq \left(\frac{6n}{5}\right)\left(\frac{6n}{5} - 1\right) = \frac{36n^2}{25} - \frac{6n}{5}$$

$$\leq \frac{36n^2}{25} - \frac{30n}{25}.$$

Then, since $n > 24$, $n^2 > 24n$, so

$$\frac{36n^2}{25} - \frac{30n}{25} \leq \frac{37n^2}{25} - \frac{24n}{25} - \frac{30n}{25}$$

$$\leq \frac{37n^2}{25} - \frac{54n}{25}$$

$$\leq \frac{37n(n-1)}{25} = 1.48(n)_2,$$

implying the result. ■

Theorem 3.9: For $n > 3$, $(n)_3 \leq F(n,3) \leq (217(n)_3)/125 = 1.736(n)_3$

Proof:

For $n = q + 1$, with q a prime power from the set $\{3, 4, 5, 7, 9, 11, 13, 16, 17, 19, 23, 25\}$, by Theorem 3.6, $F(n,3) = (n)_3$. For any $3 < h < 25$ not in the set, with $n = h + 1$, let q be the next prime power above h. Hand calculation reveals $(q(q-1)(q+1))/(n(n-1)(n-2))$ does not exceed the bound in any of these cases. From [31], for any $n > 24$, there is a prime p such that $n \leq p \leq 6n/5$. By Theorem 1.1 and Theorem 1.13 group theoretical results for $PGL(2,q)$ for any prime power q, and the fact that p^1 is a prime power, we have that

$$(n)_3 \leq F(n+1,3) \leq F(q+1,3) = q(q-1)(q+1)$$

$$\leq \left(\frac{6n}{5}+1\right)\left(\frac{6n}{5}\right)\left(\frac{6n}{5}-1\right) = \frac{216n^3}{125} - \frac{6n}{5}$$

$$\leq \frac{216n^3}{125} - \frac{150n}{125}$$

$$\leq \frac{217n^3}{125} - \frac{n^3}{125} - \frac{150n}{125}.$$

Again, $n > 24$ so $n^3 > 576n$, so

$$\frac{217n^3}{125} - \frac{n^3}{125} - \frac{150n}{125} \leq \frac{217n^3}{125} - \frac{726n}{125}$$

$$\leq \frac{217n^3}{125} - \frac{217n}{125}$$

$$\leq \frac{217(n+1)(n)(n-1)}{125} = 1.736(n)_3,$$

implying the result. ■

From these building blocks and the recursive formula in Theorem 3.3, most of the best known upper bounds for $F(n,k)$ can be established as seen in **Table 1-3**. The remaining upper

bounds were discovered by methods described in Chapter 6. Using Theorem 3.2 and the empirical results from the next chapter, lower bounds have been improved for infinitely many cases, some of which are seen in **Table 1-2**. Theorem 3.8 and 3.9 give us a better approximation ratio than was known before for the cases of $k = 2$ and $k = 3$.

CHAPTER 4

LATIN SQUARES IN TRANSITIVE PERMUTATION SETS

In this chapter, we combine some earlier observations and theorems of Chapter 2 with an exhaustive search on an enumeration of all Latin squares to determine if sharply transitive sets exist for certain n and k. Through isotopy symmetries we are assured of searching the whole space for sharply transitive sets, so if we find none, we know that none exists. By a theorem in Chapter 3 we can then raise the lower bound for infinitely many n and k. We show one such nonexistence result directly without relying on computer search.

By the definition of an n-Latin square on $\mathbb{Z}_n$, no symbol is repeated in any row or column. Because each column has n symbols chosen from the set $\mathbb{Z}_n$ without repetition, each symbol of $\mathbb{Z}_n$ must appear in each column exactly once. Since no symbol is repeated in any row, and every row contains n elements from $\mathbb{Z}_n$, each row can be considered a permutation on the elements of $\mathbb{Z}_n$. Hence every n-Latin square Λ is a sharp $(n, 1)$-transet.

4.1 Observations about a putative sharp (7, 4)-transet

Let the set M represent a sharp $(7, 4)$-transet on $\mathbb{Z}_n$. As noted in Chapter 1, since every quad must be mapped to every quad by a permutation in M, then M must contain a permutation σ_1 which in passive form starts with $\langle 0, 1, 2, 3, \dots \rangle$, to map $\langle 0, 1, 2, 3 \rangle$ to $\langle 0, 1, 2, 3 \rangle$. Likewise other permutations start with $\langle 0, 1, 2, 4, \dots \rangle$, $\langle 0, 1, 2, 5, \dots \rangle$, $\langle 0, 1, 2, 6, \dots \rangle$, $\langle 1, 0, 2, 3, \dots \rangle$, $\langle 1, 0, 2, 4, \dots \rangle$, and so on. Note in particular that the permutation σ_1 starting with $\langle 0, 1, 2, 3, \dots \rangle$

ends with some permutation of $\{4,\ 5,\ 6\}$. The permutation φ_1 which starts with $\langle 0, 1, 3, 2, \dots \rangle$ also ends with a permutation of $\{4,\ 5,\ 6\}$. Since M is sharp, there can be no permutation which maps any quad to the same quad as any other permutation. Any time four symbols are in the same position in two different permutations, those two permutations map the underlying quad to the same quad. Since 0 and 1 are in the same positions in $\langle 0, 1, 2, 3, \dots \rangle$ and $\langle 0, 1, 3, 2, \dots \rangle$, it is important that $\{4, 5, 6\}$ be in distinct positions in σ_1 and φ_1. Equivalently, the following matrix must be filled in with valid values starting at $a_{5,2}$:

$$\begin{matrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\end{matrix}\begin{bmatrix}0 & 1 & 2 & 3 & 4 & 5 & 6\\ 0 & 1 & 2 & 4 & a_{5,2} & a_{6,2} & a_{7,2}\\ 0 & 1 & 2 & 5 & a_{5,3} & a_{6,3} & a_{7,3}\\ 0 & 1 & 2 & 6 & a_{5,4} & a_{6,4} & a_{7,4}\end{bmatrix}$$

One valid value for $a_{5,2}$ is 3, which forces the following choices:

$$\begin{matrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\end{matrix}\begin{bmatrix}0 & 1 & 2 & 3 & 4 & 5 & 6\\ 0 & 1 & 2 & 4 & 3 & 6 & 5\\ 0 & 1 & 2 & 5 & 6 & a_{6,3} & a_{7,3}\\ 0 & 1 & 2 & 6 & 5 & a_{6,4} & a_{7,4}\end{bmatrix}$$

There are only two ways to complete this matrix:

$$\begin{matrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\end{matrix}\begin{bmatrix}0 & 1 & 2 & 3 & 4 & 5 & 6\\ 0 & 1 & 2 & 4 & 3 & 6 & 5\\ 0 & 1 & 2 & 5 & 6 & 3 & 4\\ 0 & 1 & 2 & 6 & 5 & 4 & 3\end{bmatrix} \text{ or } \begin{matrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\end{matrix}\begin{bmatrix}0 & 1 & 2 & 3 & 4 & 5 & 6\\ 0 & 1 & 2 & 4 & 3 & 6 & 5\\ 0 & 1 & 2 & 5 & 6 & 4 & 3\\ 0 & 1 & 2 & 6 & 5 & 3 & 4\end{bmatrix}.$$

Now observe that there is a similar matrix for φ_1 and three other permutations associated with it which must be filled in after making the prior choices:

$$\begin{matrix}\varphi_1\\ \varphi_2\\ \varphi_3\\ \varphi_4\end{matrix}\begin{bmatrix}0 & 1 & 3 & 2 & 5^* & 4 & 6\\ 0 & 1 & 3 & 4 & 6 & 2 & 5\\ 0 & 1 & 3 & 5 & 4 & 6 & 2\\ 0 & 1 & 3 & 6 & 2 & 5 & 4\end{bmatrix}$$

In this case the 5 at * cannot be replaced by a 4, because doing so would force a 5 and 6 for the remainder of φ_1, which would have φ_1 map the quad $\langle 0, 1, 5, 6\rangle$ to $\langle 0, 1, 5, 6\rangle$ just as in σ_1, or a 6 and 5 which would have φ_1 map the quad $\langle 0, 1, 5, 6\rangle$ to $\langle 0, 1, 6, 5\rangle$ in the same way σ_2 does. These concepts motivate the remainder of this chapter.

4.2 Generating reduced Latin squares

Let $\mathcal{L}_n$ be the set of all reduced Latin squares of size n. The following discussion concerns generating the set of all unique $(n, 1)$-transets from $\mathcal{L}_n$ for a given n. The set $\mathcal{L}_n$ consists only of reduced Latin squares, and all reduced Latin squares of size n are in $\mathcal{L}_n$. By the definition of $\mathcal{L}_n$, for any Latin square Λ in $\mathcal{L}_n$, permuting the symbols in Λ and then sorting the rows and columns will create Λ', which is also in $\mathcal{L}_n$. Since a given $(n, 1)$-transet M is a set, the order in which the permutations in M are listed is arbitrary. Thus any reordering of the rows of a square Λ in $\mathcal{L}_n$ will not result in a new $(n, 1)$-transet. However, reordering the columns of a square Λ in $\mathcal{L}_n$ does produce a square Λ'' which is not in $\mathcal{L}_n$, and which is different from Λ. The permutation sets generated from the Latin squares Λ and Λ'' are in the same transitive permutation set isotopy class, but are not the same.

4.3 Row-reduced Latin squares Theorem

Let Σ_n represent the image of all permutations S_n on $\mathbb{I}_n$; that is, the one-based passive form representation. Let $\mathcal{L}_n$ be the set of reduced Latin squares of size n. Let T_n be the set of those permutations in Σ_n which have a 1 in the first position. This set has cardinality $(n-1)!$. Now define the set $\mathcal{R}_n$ to be the *row-reduced Latin squares* of size n, created by taking each square Λ in $\mathcal{L}_n$, and applying every permutation from T_n to the columns of Λ to create $(n-1)!$

new squares, and adding each unique square to $\mathcal{R}_n$. Note that the identity permutation is in T_n, so that all of $\mathcal{L}_n$ is in $\mathcal{R}_n$. To distinguish from symbol permutations under composition, we will use the notation $\Lambda \times \pi$ to represent ordering the columns of Λ by the column permutation π.

Theorem 4.1: Let $\mathcal{L}_n$ be the set of all reduced Latin squares of size *n*. Let $\mathcal{R}_n$ be the row-reduced Latin squares generated from $\mathcal{L}_n$. Let Λ be a square in the set $\mathcal{L}_n$. Choose any column permutation π from Σ_n (even those which do not start with 1), and apply it to the columns of Λ to create a new square $\Lambda' = \Lambda \times \pi$. Sort the rows of Λ' to create Λ''. Then Λ'' is in $\mathcal{R}_n$.

Proof:

Any σ in Σ_n which has 1 at the beginning will not disturb the first column of Λ, so the generated Λ' is already sorted row-wise, and since σ is in T_n, the square Λ is already in $\mathcal{R}_n$. So consider π with i at the beginning, $1 < \mathrm{i} \leq n$. Since Λ is a Latin square of size n, column i of Λ has n elements from the set $\mathbb{Z}_n$, but none of them is repeated. Thus column i of Λ is a permutation of size n. The permutation π moves column i to the beginning of Λ'. Since column i is a permutation, let φ represent column i. For any permutation φ of size n, there exists an inverse of φ in S_n called φ^{-1} such that φ^{-1} maps φ onto the identity permutation, and any permutation φ^{-1} in S_n has an image γ^{-1} on $\mathbb{I}_n$. Sorting the rows of Λ' can be done by applying γ^{-1} to the rows of Λ' to create Λ''. Hence Λ'', with the row which starts with 0 at the top and the row which starts with 1 is moved next, and so on, remains a Latin square in the isotopy class with Λ'. Whatever row does start with 0, call it σ_0, is also a row of a Latin square of size n, and thus a permutation in S_n. Since the first element of σ_0 is 0, then $\sigma_0(0) = 0$. Let δ^{-1} be the image on $\mathbb{I}_n$ of ${\sigma_0}^{-1}$. Since $\sigma_0(0) = 0$, also ${\sigma_0}^{-1}(0) = 0$, and the image on $\mathbb{I}_n$ is 1, so $\delta^{-1}(1) = 1$ and thus δ^{-1} begins with a 1. Applying δ^{-1} to the columns of Λ'' creates a Latin square $\Lambda^{(3)} = \Lambda'' \times$

δ^{-1} whose first row and first column are the identity permutation, which is the definition of reduced form, so by definition $\Lambda^{(3)}$ is in $\mathcal{L}_n$. Let δ be the image on $\mathbb{I}_n$ of σ_0. As before, $\delta(1) = 1$. Since σ_0 is the inverse of ${\sigma_0}^{-1}$, applying δ to the columns of $\Lambda^{(3)}$ will produce Λ''. By the definition of $\mathcal{R}_n$, the square Λ'' is in $\mathcal{R}_n$, since Λ'' was generated from a square in $\mathcal{L}_n$ by applying δ, a column permutation starting with 1, to the Latin square $\Lambda^{(3)}$, a n-Latin square in reduced form. ■

The set of row-reduced n-Latin squares accounts for all possible n-Latin squares if the order of the rows is ignored. This fact is also important to the discussion of Latin squares related to k-transitive permutation sets when $k > 1$.

4.4 Latin squares in (n, k)-transets for k > 1

The following discussion formalizes and generalizes some observations from the earlier section about $(7, 4)$-transets. An n by n square matrix Π of any n unique integers (not necessarily from $\mathbb{Z}_n$) which are not repeated in any row or column is isomorphic to an n-Latin square. To map to a Latin square on $\mathbb{Z}_n$, all occurrences of the lowest integer in Π are replaced by 0, the next lowest by 1, and so on, up to the highest, which are replaced by $n - 1$. Likewise a "shifted" Latin square can be created from a template Latin square by replacing each symbol of the proper Latin square with a distinct symbol.

As shown by Theorem 2.3, a sharp (n, k)-transet M on $\mathbb{Z}_n$with $k > 1$ will have $n!/k!$ distinct permutations each of which starts with one of the k-tuples of symbols chosen from $\mathbb{Z}_n$. Since every k-tuple is represented at the start of some permutation in M, consider μ, a $(k$-1)-tuple of symbols appearing in the first $k - 1$ positions. The k^{th} position can be any of the

remaining symbols from $\mathbb{Z}_n$, and each such k-tuple is represented by a permutation in M, so there will be $n-k+1$ permutations each starting with the elements in μ in order.

Let M' be the set of permutations in M which start with the elements of μ in order. Since M is sharp, there will be no repetition of any k-tuple at the same positions in any permutation, which means that the k^{th} position of each permutation in M' must be unique. In fact, since no k-tuple is duplicated, no permutation of M' can have the same symbol as another permutation in M' in any position past the k^{th} position, either. Since M' consists of permutations, no row will repeat any element in any of the positions, even when considering only the positions after the $(k-1)^{th}$ position.

Since the first $k-1$ positions of M' are all identical, the remaining columns of M' must consist only of the symbols not in μ. All of these restrictions taken together mean that the k^{th} through n^{th} columns of M' determine a shifted Latin square of size $(n-k+1)$. Since the rows of M' can be listed in any order without disrupting the overall transet M, consider the rows to be ordered by least k^{th} position value to highest k^{th} position value. This is the lexicographic order for these permutations.

By Theorem 2.3, there is a permutation σ in M starting with $\langle 0, 1, 2, \dots, k-1\rangle$. If σ is not the identity ε, then by Theorem 2.1, the columns of M can be rearranged by σ^{-1} such that the first permutation of M is ε, and M remains a sharp (n,k)-transet because no permutations are added. So assume without loss of generality that ε is in M. Now consider the first $(n-k+1)$ permutations of M in the lexicographic order. That is, let $\mu = \langle 0, 1, 2, \dots, k-2\rangle$ and let M_1 be the set of $(n-k+1)$ permutations which start with μ. The first permutation of M_1 is in the identity order. As before, order the rows of M_1 lexicographically so that the k-position elements

are in order from $k-1$ to $n-1$. Thus the rightmost $n-k+1$ columns (including the k^{th} column) of M_1 make up a shifted version of a Latin square in reduced form. The following table represents M_1, and the matrix formed by the a_{ij} along with the symbols $k-1$ through $n-1$ just above and just to the left of that matrix together form a shifted Latin square in reduced form.

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-k+1} \end{array} \begin{bmatrix} 0 & 1 & \cdots & k-2 & k-1 & k & k+1 & \cdots & n-1 \\ 0 & 1 & \cdots & k-2 & k & a_{22} & a_{32} & a_{\cdots 2} & a_{n-k+1,2} \\ 0 & 1 & \cdots & k-2 & k+1 & a_{23} & a_{33} & a_{\cdots 3} & a_{n-k+1,3} \\ \vdots & \vdots & \ddots & k-2 & \cdots & a_{2\cdots} & a_{3\cdots} & a_{\cdots} & \vdots \\ 0 & 1 & \cdots & k-2 & n-1 & a_{2,n-k+1} & a_{3,n-k+1} & a_{\cdots n-k+1} & a_{\cdots} \end{bmatrix}$$

Figure 4-1. The matrix M_1

Note that next in the lexicographic order, there is a similar matrix M_2 which has the same constraints, but with different constant values in the first $k-1$ columns. The first $k-1$ columns of M_2 are uniform, but instead of starting with $\langle 0, 1, \ldots, k-3, k-2, \ldots \rangle$, they start with $\langle 0, 1, \ldots, k-3, k-1, \ldots \rangle$. But the remaining $n-k+1$ columns are still a shifted Latin square of slightly different symbols. The following example is M_1 from a sharp $(6, 4)$-transet. The first $k-1=3$ columns are the same, and the remaining 3 columns make a 3×3 shifted Latin square of the symbols 3, 4, 5, in reduced form.

$$\begin{array}{c} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{array} \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 4 & 5 & 3 \\ 0 & 1 & 2 & 5 & 3 & 4 \end{bmatrix}$$

Figure 4-2. M_1 from sharp (6, 4)-transet

M_2 has the additional constraint relative to M_1 that none of the quads from M_1 can be repeated in M_2. Since the two matrices have $k-1$ elements in common in the first $k-1$ columns, the Latin squares can have no pair of elements in common. Thus, the induced Latin

square in M_2 cannot be in reduced form if M_1 is in reduced form, unless $k = n - 1$. The following table represents an M_2 where $k = n - 2$.

$$\begin{matrix}\sigma_4\\ \sigma_5\\ \sigma_6\end{matrix}\begin{bmatrix}0 & 1 & 3 & 2 & 5 & 4\\ 0 & 1 & 3 & 4 & 2 & 5\\ 0 & 1 & 3 & 5 & 4 & 2\end{bmatrix}$$

Figure 4-3. M_2 from sharp (6, 4)-transet

Observe that the first two columns of M_2 are the same as the first two columns of M_1. Since $k = 4$, no quad can be in common between M_1 and M_2. For example, σ_4 cannot end with $\langle \dots, 2, 4, 5\rangle$, because the quad $\langle 0, 1, 5, 6\rangle$ would be mapped to $\langle 0, 1, 4, 5\rangle$ in σ_1 and σ_4. Also, σ_4 cannot end with $\langle \dots, 4, x, y\rangle$ or $\langle \dots, 5, x, y\rangle$ because σ_5 and σ_6 respectively must have $\langle \dots, 4, x, y\rangle$ and $\langle \dots, 5, x, y\rangle$ respectively. Consequently σ_4 must end with $\langle \dots, 2, 5, 4\rangle$. In a sharp transet, every permutation must cover as many k-tuples as it possibly can, so there will be no redundancy. Note that the shifted Latin square in **Figure 4-3** (or the canonical form which is in **Figure 4-4**) is not in reduced form as a result.

$$\begin{matrix}\tau_1\\ \tau_2\\ \tau_3\end{matrix}\begin{bmatrix}0 & 2 & 1\\ 1 & 0 & 2\\ 2 & 1 & 0\end{bmatrix}$$

Figure 4-4. Unshifted non-reduced Latin square

The Latin square represented in **Figure 4-4** is shifted into the square in M_2. The situation is different for a $(6, 5)$-transet, where only 2-Latin squares are needed to create N_1 and N_2.

$$\begin{matrix}\sigma_1\\ \sigma_2\\ \sigma_3\\ \sigma_4\end{matrix}\begin{bmatrix}0 & 1 & 2 & 3 & 4 & 5\\ 0 & 1 & 2 & 3 & 5 & 4\\ 0 & 1 & 2 & 4 & 3 & 5\\ 0 & 1 & 2 & 4 & 5 & 3\end{bmatrix}$$

Figure 4-5. N_1 and N_2 from sharp (6, 5)-transet

In **Figure 4-5**, since $k = n - 1$, it is possible for both Latin squares from N_1 and N_2 to be in reduced form. Permutation σ_1 with σ_2 represent N_1, and σ_3 with σ_4 represent N_2. In both cases, the first four columns are the same, and the Latin square is of size 2. No 5-tuple is repeated in the two squares because no pair can be repeated when the pairs of symbols in the shifted Latin square differ.

4.5 There is no sharp (7, 4)-transet

We now revisit the $(7, 4)$-transet case. If a set M is a $(7, 4)$-transet, then without loss of generality there is a set M' isotopic to M with the identity permutation in it. This can be accomplished simply by reordering the columns using the inverse of the first permutation in the set. So without loss of generality it can be assumed that M starts with the identity permutation $\langle 0, 1, 2, 3, 4, 5, 6\rangle$.

Theorem 4.2: $F(7, 4) \geq 7 \cdot 6 \cdot 5 \cdot 4 + 1$.

Proof:

In a sharp $(7, 4)$-transet M in the lexicographic order, each grouping of four permutations starts with the same three symbols. There are 210 such blocks of four permutations. The first four each start with $\langle 0, 1, 2, \dots\rangle$, and the last four start with $\langle 6, 5, 4, \dots\rangle$. Each of the 210 blocks makes up a shifted Latin square in the last four columns, for a total of $210 \times 4 = (7)_4$ permutations in a sharp transet.

Let $\sigma_0 = \langle 0, 1, 2, 3, 4, 5, 6\rangle$, the first permutation in M. Let φ_0 be a different permutation starting with $\langle 0, 1, 3, 2, \dots\rangle$, which must exist in M to map $\langle 0, 1, 2, 3\rangle$ to $\langle 0, 1, 3, 2\rangle$. By sharpness

of M, there are no two permutations which map the same quad to the same quad. Therefore, while φ_0 must contain the symbols {4, 5, 6} in the final three positions, they cannot appear in the same order as in σ_0, because {0, 1} are already in the same two positions in σ_0 and φ_0, so $\langle 0, 1, 4, 5\rangle$ for example would be mapped to $\langle 0, 1, 4, 5\rangle$ by both permutations. There are five other permutations of {4, 5, 6} from which to choose for the remaining three positions of φ_0, and none of them have a pair in common with $\langle 4, 5, 6\rangle$. Let the permutation π_3 be the column permutation of {1, 2, 3} which maps $\langle 4, 5, 6\rangle$ onto the triple represented by the last three positions of φ_0. Let π_4 be the column permutation of length 4 formed by 1 in the first position, and by adding one to each element of π_3 in the subsequent positions. That is, $\pi_4(1) = 1$, and $\pi_4(i) = \pi_3(i-1) + 1$ otherwise.

Now consider the Latin square Λ_2 formed by the set of permutations M_2 which start with $\langle 0, 1, 3, \ldots\rangle$. The set M_2 contains φ_0. Since the fourth position of φ_0 is 2, φ_0 is the first permutation in M_2 in lexicographic order. Think of Λ_2 as a shifted Latin square of order 4 of the symbols {2, 4, 5, 6}, with positions (4, 5, 6, 7) of φ_0 as the first row of Λ_2. Lexically, φ_0 is first because it has a 2 in position 4. For example, if $\varphi_0 = \langle 0, 1, 3, 2, 6, 4, 5\rangle$, then the first row of Λ_2 is $\langle 2, 6, 4, 5\rangle$ and $\pi_4 = \langle 1, 4, 3, 2\rangle$. Note that if 2 is mapped to 0, 4 to 1, 5 to 2, and 6 to 3, to create $\Lambda_2{}'$ then $\Lambda_2{}'$ is not in reduced form in any case, because φ_0 cannot end with $\langle \ldots, 4, 5, 6\rangle$. However, since $\Lambda_2{}'$ is formed by permutations in M_2 in lexicographic order, and each of those permutations is identical for the first three columns, the left column of $\Lambda_2{}'$ is $\langle 0, 1, 2, 3\rangle$ in that order. Thus, the set $\mathcal{L}_4$ of all reduced Latin squares of size 4 does contain a square Λ_i such that $\Lambda_i \times \pi_4 = \Lambda_2{}'$.

Here is the set of all reduced Latin squares of size four:

$$\begin{bmatrix}0&1&2&3\\1&0&3&2\\2&3&0&1\\3&2&1&0\end{bmatrix}\quad\begin{bmatrix}0&1&2&3\\1&0&3&2\\2&3&1&0\\3&2&0&1\end{bmatrix}$$

$$\begin{bmatrix}0&1&2&3\\1&2&3&0\\2&3&0&1\\3&0&1&2\end{bmatrix}\quad\begin{bmatrix}0&1&2&3\\1&3&0&2\\2&0&3&1\\3&2&1&0\end{bmatrix}$$

Figure 4-6. Reduced 4-Latin squares

When the reduced Latin squares are used, all permutations of the rightmost elements must be tried. That is, all permutations of $\mathbb{I}_4$ which start with 1 must be applied to each square via column transposition to find compatible squares. For permutations starting with $\langle 0, 1, 3, \dots \rangle$, the set of remaining elements is $\{2, 4, 5, 6\}$. Hence to consider all shifted squares, map $\langle {0123 \atop 2456} \rangle$ top to bottom for each element in the reduced 4-Latin square.

$$\begin{bmatrix}2&4&5&6\\4&2&6&5\\5&6&2&4\\6&5&4&2\end{bmatrix}\quad\begin{bmatrix}2&4&5&6\\4&2&6&5\\5&6&4&2\\6&5&2&4\end{bmatrix}$$

$$\begin{bmatrix}2&4&5&6\\4&5&6&2\\5&6&2&4\\6&2&4&5\end{bmatrix}\quad\begin{bmatrix}2&4&5&6\\4&6&2&5\\5&2&6&4\\6&5&4&2\end{bmatrix}$$

Figure 4-7. Shifted reduced squares

One of the squares of **Figure 4-7** represents Λ_i, to which some column permutation π_4 must be applied to create Λ_2. However, it is necessary to avoid leaving any of $\{4, 5, 6\}$ in the same position as the identity permutation, because $\{0, 1\}$ are already matching the identity permutation. This immediately eliminates the idea that π_4 is the identity permutation. Suppose

that Λ_i is the leftmost top square in **Figure 4-7**. All possible π_4 applied to Λ_i result in the following choices for the shifted Latin square Λ_2:

$$\begin{bmatrix} 2 & 4 & 5^* & 6^* \\ 4 & 2 & 6 & 5 \\ 5 & 6 & 2 & 4 \\ 6 & 5 & 4 & 2 \end{bmatrix} \begin{bmatrix} 2 & 4 & 6 & 5 \\ 4 & 2 & 5^* & 6^* \\ 5 & 6 & 4 & 2 \\ 6 & 5 & 2 & 4 \end{bmatrix} \begin{bmatrix} 2 & 5 & 4 & 6 \\ 4 & 6 & 2 & 5 \\ 5 & 2 & 6 & 4 \\ 6 & 4^* & 5^* & 2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 5 & 6 & 4 \\ 4 & 6 & 5 & 2 \\ 5 & 2 & 4 & 6 \\ 6 & 4 & 2 & 5 \end{bmatrix} \begin{bmatrix} 2 & 6 & 4 & 5 \\ 4 & 5 & 2 & 6 \\ 5 & 4 & 6 & 2 \\ 6 & 2 & 5 & 4 \end{bmatrix} \begin{bmatrix} 2 & 6 & 5 & 4 \\ 4 & 5 & 6 & 2 \\ 5 & 4^* & 2 & 6^* \\ 6 & 2 & 4 & 5 \end{bmatrix}$$

Figure 4-8. Permuted shifted Latin square columns

Note that the first three values for all four rows of the M_2 permutations are fixed at $\langle 0, 1, 3, \ldots \rangle$ in every case, so a redundancy of any two of $\{4, 5, 6\}$ at their identity positions in any row of the Latin square eliminates that square from consideration as Λ_2. Squares with two elements starred thus cannot be Λ_2. This leaves only two choices for π_4, which both cause derangements of the elements $\langle \ldots, 4, 5, 6 \rangle$.

Now consider the permutations starting with $\langle 0, 3, 2, \ldots \rangle$ which form M_3. These will be followed by a shifted Latin square on the symbols $\{1, 4, 5, 6\}$. Since the 0 and 2 are in the same positions as in the identity permutation, a square which does not overlap any pair out of $\langle \ldots, 4, 5, 6 \rangle$ in the last three positions is required. Also, with the 0 in the same position as for those permutations starting with $\langle 0, 1, 3, \ldots \rangle$, the symbols $\{4, 5, 6\}$ cannot be in the same order in the first permutation of M_3, or the quad $\langle 0, 4, 5, 6 \rangle$ would be mapped to $\langle 0, 4, 5, 6 \rangle$ by two different permutations. So the other remaining square from **Figure 4-8** is the only possibility. The permutations starting with $\langle 3, 1, 2, \ldots \rangle$ forming M_4 will be followed by a shifted Latin square on the symbols $\{0, 4, 5, 6\}$, with similar restrictions as before since 1 and 2 are in the identity

positions. The permutation starting with $\langle 3, 1, 2, \ldots \rangle$ has an element in common with $\langle 0, 1, 3, \ldots \rangle$ and $\langle 0, 3, 2, \ldots \rangle$ so the same permutation of $\{4, 5, 6\}$ as before cannot be used for the permutation starting with $\langle 3, 1, 2, 0, \ldots \rangle$ from M_4. Creating tables like **Figure 4-8** for the remaining squares in **Figure 4-7** fails to yield any usable square for Λ_i', so there is no sharp $(7,\ 4)$-transet. Thus $F(7,4) \geq 7 \cdot 6 \cdot 5 \cdot 4 + 1$. ■

By Corollary 3.1, since there is no sharp $(7, 4)$-transet, there is also no sharp $(8, 5)$-transet, or any sharp $(7 + i, 4 + i)$-transet for any positive integer i. This, along with Theorem 3.2, implies result 3.2.2 in Chapter 1. For example, consider M, an $(8,\ 5)$-transet on $\mathbb{Z}_8$. By the construction in Theorem 3.1, it is possible to sort the permutations in M by lexicographic order, and group them by the starting symbol. For any column i, the permutations which have a 7 in column i, collectively M_i, can be grouped together, have column i removed, and M_i will comprise a $(7, 4)$-transet on $\mathbb{Z}_7$. However, since there is no sharp $(7, 4)$-transet, $|M_i|$ is at least $(7)_4 + 1$. This implies that any of the 8 sets of permutations from an $(8, 5)$-transet, M_1 through M_8, must have at least $(7)_4 + 1$ permutations in it, which in turn implies that $F(8, 5) \geq 8 \cdot ((7)_4 + 1) = (8)_5 + 8$. Likewise, $F(9, 6) \geq (9)_6 + 72$. In general, $F(7 + i, 4 + i) \geq (7 + i)_{4+i} + (7 + i)_i$. Similar properties will hold along the diagonal any time it can be shown that a sharp (n, k)-transet does not exist, so that

Theorem 4.3: $F(n, k) > (n)_k \rightarrow F(n + i, k + i) \geq (n + i)_{k+i} + (n + i)_i$, for all $i \geq 1$.

CHAPTER 5

TRANSITIVE PERMUTATION SET BUILDING TECHNIQUES

In this chapter we consider the generation of covering sets for even values of $n \geq 14$ and for small k. Since the symmetric group of degree n is quite large when $n > 12$, we demonstrate a useful method for creating covering sets that are much smaller than $n!$, for values of $k = 3$, 4, and 5. The technique involves 1-factorization of the complete graph of size $n/2$, an application of an $(n/2, k)$-transet, and an application of a binary covering suite $BCS(n/2, k)$. Initially, a set of permutations, called a *seed set*, is created so that the adjacent members of each permutation in the seed set are treated as pairs. Each permutation in the initial seed set has its pairs permuted according to the $(n/2, k)$-transet, and each resulting permutation is then flipped according to the binary strings in the $BCS(n/2, k)$.

5.1 Binary covering suites

A binary covering suite $BCS(n, k)$ is a set of binary strings of length n which covers all binary strings of length k in all k-tuples of positions taken from I_n. The function $b(n, k)$ is the minimum cardinality of any $BCS(n, k)$. Binary covering suites are covered as a special case of testing suites in [6].

Theorem 5.1: $b(n, n) = 2^n$.

Proof: Let B be a $BCS(n, n)$ suite. Let α be any binary string of length n. The symbols in α in their given positions must be covered by a string in B. Since the only string of length n that covers α is α itself, α must be in B. So, B must be all binary strings of length n. ■

Theorem 5.2: $b(n + 1, n) = 2^n$

Proof: Let B be the set of all binary strings of length n. Then B is a $BCS(n, n)$ suite. Construct a new set B' by taking each string α_i from B, and adding a final bit to the end of it to make it length $n + 1$, and add that string to B'. The bit will be a 0 if the sum of all the 1-bits in α_i is odd, otherwise it will be 1. We show that B' is a $BCS(n + 1, n)$.

To see this, consider any string β of length n, and any n-tuple of positions ρ chosen from $\mathbb{I}_{n+1}$. Note that exactly one position, denoted p_a, is not included in ρ. If $p_a = n + 1$, then β is covered, because the first n columns of B' is just the set of all binary strings of length n, namely B. If $p_a \neq n + 1$, let us say $p_a = i \leq n$, then there are two strings in B that agree with all the symbols of β in those positions of $\rho \leq n$, namely one, say b_1, that has all of the symbols of β in the required positions and a 0 in position i, and the other, say b_2, has all of the symbols of β in the required positions and a 1 in position i. Clearly, the sum of the 1 bits in b_1 and b_2 have opposite parity. So, by the construction of B', one of the two is extended by adding the bit 0, and the other is extended by adding the bit 1. In either case, there is a row in B' that has all the binary bits of β in the positions given by ρ. ■

Theorem 5.3: $b(2n, 3) = 2 \cdot b(n, 3)$

Proof:

If B is a $BCS(n, 3)$, construct B', a $BCS(2n, 3)$ from four copies of B as follows. Suppose B consists of a rows. Make the first n columns of the first a rows of B' from one copy of B, say

B_1, and the second n columns from an identical copy B_2, so that they are placed side by side. The next a rows of B' are constructed from another identical copy B_3 in the first n columns, and the final n columns of the second a rows of B' are created from a copy B_4 with every bit toggled. This B' is a $BCS(2n, 3)$ if it covers all triples of bits in all triples of columns. Consider $\tau = (t_1, t_2, t_3)$, a triple of bits, in columns $\rho = (p_1, p_2, p_3)$. If all of ρ is in the left n columns, the first copy of B will cover τ because it is a normal $BCS(n, 3)$. Likewise if all of ρ is in the right n columns. Since $k = 3$, there are only two other possibilities.

Case 1: Columns $p_1$and p_2 are in the left columns and p_3 is in the right columns.

Assume the value of p_3 is not exactly n more than one of the other columns. Then consider the triple of bits (t_1, t_2, t_3) in columns $(p_1, p_2, p_3 - n)$. This triple is covered by some row i in B, and thus by row i of B_1. Since p_3 is n more than one of the columns, t_3 will be covered by row i of B_2 at p_3, so the row i of B' covers τ at ρ. If $p_3 = p_1 + n$, consider t_1 and t_2. They are covered in p_1 and p_2 by at least one row i of B. If $t_3 = t_1$, then t_3 will be covered by row i of B_2 at p_3, so row i of B' covers τ at ρ. If $t_3 \neq t_1$, then since B_4 is the opposite of B_2, bit t_3 will be covered by row i of B_4 and of course B_3 is the same as B_1, so row $i + n$ of B' covers τ at ρ. Similarly if $p_3 = p_2 + n$, so case 1 is handled.

Case 2: Column $p_1$is in the left columns and p_2 and p_3 are in the right columns.

Assume the value of p_1 is not exactly n less than one of the other columns. Then consider the triple of bits (t_1, t_2, t_3) in columns $(p_1, p_2 - n, p_3 - n)$. This triple is covered by some row i in B, and thus by row i of B_1. Since p_2 and p_3 are n more than their respective columns, t_2 and t_3 will be covered by row i of B_2 at p_2 and p_3, so the row i of B' covers τ at ρ. If $p_3 = p_1 + n$, consider t_1 and t_2. They are covered in p_1 and $p_2 - n$ by at least one row i of B. If $t_3 = t_1$, then

t_2 and t_3 will be covered by row i of B_2 at p_2 and p_3, so row i of B' covers τ at ρ. If $t_3 \neq t_1$, then consider that toggling every bit of a $BCS(n,k)$ will result in a $BCS(n,k)$. Thus B_4 is a $BCS(n,3)$ which covers t_2 and t_3 at columns $p_2 - n$ and $p_3 - n$ at row j. Since B_4 is the opposite of B_2, bit t_1 will be covered at column p_1 by row j of B_3, so row $j+n$ of B' covers τ at ρ. Similarly if $p_3 = p_2 + n$, so case 2 is handled. ■

Thus, there is an upper bound on the size of a $BCS(n,3)$, and a greedy search can increase the value k by adding just a few rows to a known BCS. In [6], a method is described for creating a $BCS(n,2)$ of size logarithmic in n.

Table 5-1. A size 6 BCS(10, 2)

1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1
1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	0	0

5.2 Round robin scheduling

Many values of $b(n,k)$ have been computed for small k[18]. The usefulness of $b(n,\ k)$ for the $F(n,k)$ problem is discussed below. The results also rely on computing a 1-factorization for the complete graph K_n for even n. That factorization is computed through the following divide-and-conquer strategy.

The problem of round-robin tournament scheduling requires each player to play every other player, and for each player to play one game per day for $n - 1$ days. The problem can be solved by dividing the players in half and solving for a half-size problem. When the problem size is 2, the players play each other. First list the integers from 0 to $n - 1$ in numeric order. This list represents the players for each "day" of the tournament. Each subsequent row under the first represents the opponents of the players in the list. Note that this will be symmetric in that if the first player plays the second player, then the second player plays the first on the same day.

On the first day, divide the teams in two and repeat until teams are of size two. Those two play each other. For example, for a problem size of eight, divide into two teams of size four, (0, 1, 2, 3) and (4, 5, 6, 7). Then recursively apply the same algorithm to come up with a schedule for the smaller problem. So divide again into problems of size two, who play each other. Thus 0 plays 1, 2 plays 3, 4 plays 5, and 6 plays 7. This row is written under the first one like so:

Table 5-2. 8-way round-robin day 1

0	***1***	***2***	***3***	***4***	***5***	***6***	***7***
1	0	3	2	5	4	7	6

The problem now is to have the teams play each other. The lower-numbered players need to play the upper-numbered players. The solution is to have each team finish its internal schedule, and then have player one from team one play player one from team two and so on, pairing off the entire team. On the day after that, do a cyclic shift so that player one from team one plays player 2 from team two, two from one plays three from two, and player x from team

one plays player one from team two. This shift is repeated until the internal schedule is complete at that level. For example, for a starting size of 4, the schedule will be complete in 3 days. The first day, teams are broken down to teams of size two, and those players play each other, so 0 plays 1 and 2 plays 3. On day two, play player one of team one plays against player one of team 2 and so on, so that 0 plays 2 and 1 plays 3. The next day, shift, so that player one of team 1 plays player 2 of team 2 and so on, so that 0 plays 3 and 1 plays 2. This is the full schedule, as follows:

Table 5-3. 4 player round robin

0	***1***	***2***	***3***
1	0	3	2
2	3	0	1
3	2	1	0

This schedule is applied to the left and right sides of the size-8 tables, treating the lower and upper four players as two distinct 4-player teams. After the third day, the internal team schedules are complete. On the fourth day the process of having the lower team play the upper team is carried out, so that the first player of team one, 0, plays the first player of team two, 4, and so on. On day 5 the cyclic shift happens until the 8-way schedule is complete, as follows:

Table 5-4. 8-way round robin schedule

0	***1***	***2***	***3***	***4***	***5***	***6***	***7***
1	0	3	2	5	4	7	6
2	3	0	1	6	7	4	5
3	2	1	0	7	6	5	4
4	5	6	7	0	1	2	3
5	6	7	4	3	0	1	2
6	7	4	5	2	3	0	1
7	4	5	6	1	2	3	0

Note that this is also a Latin square of size 8 including the first row header. Problems that are not powers of 2 will require special consideration. Any team of size one gets a bye for now, but the bye will be filled in by a bye player from the other side. Since the original problem size was even, if there is a bye on the left, there is a bye on the right. For example, for a problem of size 6, first divide the problem in two to problems of size 3, (0, 1, 2) and (3, 4, 5). Then divide again into problems of size two, but 2 and 5 are stranded. So on the left, 0 plays 1 and 1 plays 0, 3 plays 4 and 4 plays three, but 2 plays 5.

Table 5-5. 6-way round robin day 1

0	***1***	***2***	***3***	***4***	***5***
1	0	5	4	3	2

On day two, the tournaments of size one and two schedules are complete, so it is time to have the lower number players play the higher number players. This is accomplished as before by letting 0 play 2, leaving 1 with a bye, and letting 3 play 5, leaving 4 with a bye. Again 1 can play 4. The next day, shift again, so that 0 has the bye while 1 plays 2, and 3 has a bye while 4 plays 5. Meanwhile instead of byes, 0 plays 3. This completes the schedules for the teams of size n/2, so now the lower numbers must play the higher numbers. However with the byes, 0 has already played 3, and so on, so move to the next shift up. So 0 plays 5 and so on. This leaves only one shift more to complete the schedule:

Table 5-6. 6-way round robin schedule

0	***1***	***2***	***3***	***4***	***5***
1	0	5	4	3	2
2	4	0	5	1	3
3	2	1	0	5	4
4	5	3	2	0	1
5	3	4	1	2	0

5.3 An (n, 3)-transet from round robin

Each completed schedule can be used to generate a complete pairing representing each day. For example, the first pairing from day one of Table 5-6 is $\{\langle 0, 1\rangle, \langle 2, 5\rangle, \langle 3, 4\rangle\}$. Note that each of these pairings is a permutation on $\mathbb{Z}_n$. The complete set of $n - 1$ pairings makes up a list of permutations of minimal size such that every pair of symbols from $\mathbb{Z}_n$ is in some permutation. These permutations used as "seed" can generate an $(n, 3)$-transet, similar to how prime

generators are used to generate seed permutations for $(p, 2)$-transets such that every cyclic shift of each seed is added to the final $(p, 2)$-transet.

A cyclic shift on pairings from round-robin tables would put each pair into each position, but since $k = 3$, every other symbol needs to be in all other positions relative to each pair. This implies that each seed permutation must have the pairs permuted so that the pairs are next to the other pairs. For example, the pairing ⟨⟨0, 1⟩, ⟨2, 5⟩, ⟨3, 4⟩⟩ would generate ⟨⟨0, 1⟩, ⟨2, 5⟩, ⟨3, 4⟩⟩, ⟨⟨0, 1⟩, ⟨3, 4⟩, ⟨2, 5⟩⟩, ⟨⟨3, 4⟩, ⟨0, 1⟩, ⟨2, 5⟩⟩, ⟨⟨3, 4⟩, ⟨2, 5⟩, ⟨0, 1⟩⟩, ⟨⟨2, 5⟩, ⟨0, 1⟩, ⟨3, 4⟩⟩, and ⟨⟨2, 5⟩, ⟨3, 4⟩, ⟨0, 1⟩⟩. Each of these would be flattened out to a permutation, such as ⟨0, 1, 2, 5, 3, 4⟩, for the transet.

Since each pair is part of a permutation, for any two triples of symbols ρ and τ, whenever two of the symbols of ρ are adjacent, with the lower symbol even and the higher symbol odd, the corresponding symbols from τ for that pair will appear together in a permutation and necessarily the third symbol will appear as part of another pair. Permuting the pairs will move the pair and the third symbol around. This is sufficient to cover such a pair of symbols if they are lower first, then higher.

To cover triples none of whose symbols in ρ are adjacent, each symbol must be in a different pair. Thus for an $(n, 3)$-transet only three pairs at a time need consideration, each one having one of the symbols from the target triple τ. This implies that instead of every permutation of each set of pairs being required, only every triple of pairs in every triple of relative positions are needed at any given time. So, this just requires an $(n/2, 3)$-transet to arrange the pairs in this way.

Of course, each of the pairs is arranged lower number first, then higher number. So to get a fully 3-transitive set, every permutation generated so far must have the pairs reversed when appropriate. For example, to map $\langle 0, 2, 4\rangle$ to $\langle 1, 3, 5\rangle$ for a $(6, 3)$-transet created this way, each of the symbols is at the wrong end of the pair. So in that case all three pairs must be reversed. However, to cover every set of triples requires mapping the triple $\langle 0, 3, 4\rangle$ to $\langle 1, 3, 5\rangle$. Even though positions 3 and 4 are adjacent, the odd number is lower so they are not in the same pair. Consequently the first and last pair need to be reversed, but the middle pair is fine as is. In general there are two cases: flipping a pair or not flipping a pair. Again only three pairs at a time must be flipped appropriately, for $n/2$ total pairs. The two cases can be envisioned as a binary triple with flipping=1, not flipping=0. Thus a binary covering suite of size $b(n/2, 3)$ is needed to generate the flips needed to cover all triples in all triples of positions. Each string of the binary covering suite is applied to each permutation of pairs, generating $b(n/2, 3)$ new permutations.

The size of a round-robin tournament for n players is $n - 1$ days, and each day represents a pairing permutation. Each pairing permutation has $F(n/2, 3)$ permutations generated for it, and each of those permutations must be flipped in $b(n/2, 3)$ ways. Thus the total size for the final $(n, 3)$-transet M is $|M| = (n - 1) \cdot F(n/2, 3) \cdot b(n/2, 3)$. For example, a $(20, 3)$-transet will use $F(10, 3) = 720$ permutations of the pairs, and $b(10, 3) \leq 13$ binary strings to cover all triples to all triples for items taken from $\mathbb{Z}_n$. Thus $F(20, 3) \leq 19 \cdot 720 \cdot 13 = 177840$. Note that this number was determined without having to find any other $F(n, 3)$ values in between. There are many redundancies in this number, but techniques in the next chapter can be used to deal with redundancies. Note that a $(20, 3)$-transet was generated without having to start from S_{20}, which has cardinality such that $|S_{20}| > 2^{18}$. Further, the $(20, 3)$-transet can be created from a $(10, 3)$-

transet without all the intervening sets for $(19, 3)$, $(19, 2)$, etc., as required by the recursive algorithm. Group theory can do better for generating any $(n, 3)$-transet again by building intermediate sets, but this technique can be generalized to $k = 4$ and $k = 5$ as follows.

5.4 An (n, 4)-transet from pairs of pairs

Now consider what is needed to expand this technique to an $(n, 4)$-transet. In this case it is necessary to map such tuples as $\langle 0, 1, 4, 5\rangle$ to $\langle 1, 3, 7, 5\rangle$. Since 0 and 1 are adjacent and start with an even number, the symbols 1 and 3 must be in a pair together. Likewise 4 and 5 are adjacent, so the symbols 7 and 5 must be in a pair together. In general, every pair of symbols must appear in a seed permutation with every other pair of symbols, so that they can be permuted around and reversed to create a final cover, at the very least to cover this type of case. In the following discussion, pairs are considered to be in arbitrary order.

Assume that a minimal such seed set exists for a $(10, 4)$-transet. That is, each pair of symbols is paired with each other pair of symbols exactly once, considering every pair to be ordered lowest element first. Then the pair $\langle 0, 1\rangle$ will be paired with all other pairs of symbols. Since this is a seed set, consider every pair to be ordered lowest to highest, and every seed permutation of five pairs to list the pairs in lexicographic order defined by the first element of each pair. Each permutation which starts with $\langle 0, 1\rangle$ will pair $\langle 0, 1\rangle$ with four other pairs. There will be $\binom{8}{2} = 28$ other pairs in all, so at least 7 permutations start with $\langle 0, 1\rangle$. Clearly $\langle 0, 2\rangle$ is not in the same permutations which start with $\langle 0, 1\rangle$, but there are still 28 pairs of other symbols to pair with $\langle 0, 2\rangle$ so 7 different permutations start with $\langle 0, 2\rangle$ in the ideal case. This holds all the way up seven permutations starting with $\langle 0, 9\rangle$, but then $\langle 1, 2\rangle$ could ideally be in with other

permutations, say the ones that start with $\langle 0, 3\rangle$, so the very minimum is $9 \cdot 7 = 63$ permutations in a seed set for a $(10, 4)$-transet, or in general $(n-1)(n-3)$ for an $(n, 4)$-transet.

A branch-and-bound exhaustive search of all possible length-10 sets of seed permutations turned up at least one perfect set of pairs of pairs for a $(10, 4)$-transet, of size 63. It was also discovered that there is no such pairing of length 8 for four pairs of pairs, but there is one of length 6 for three pairs of pairs; it is of size 15. More on this below.

If only two positions are paired together, then the symbols associated with those positions must be paired together and the other two symbols must appear in separate pairs. An example of this case is mapping the quad $\langle 0, 2, 3, 4\rangle$ to $\langle 2, 0, 6, 5\rangle$. Since 2 and 3 are adjacent and start with an even number the symbols 0 and 6 must appear in a pair together in a seed permutation. The other symbols 2 and 5 must be in separate pairs. A seed set where every pair appears with every other pair would include the pair $\langle 0, 6\rangle$ paired with $\langle 2, 4\rangle$, and thus the 5 would have to be in its own pair. So the ideal seed set if it exists will accommodate this case.

If no even-odd integers are paired together in the first tuple, then each symbol of the second tuple must appear in its own pair. For example, the $(10, 4)$-transet must map $\langle 0, 2, 4, 7\rangle$ to $\langle 3, 8, 2, 5\rangle$. So each of 3, 8, 2, and 5 must appear in separate pairs. However, the seed set which pairs all pairs with all other pairs will have a permutation, say σ, where the pair $\langle 3, 4\rangle$ is paired with the pair $\langle 1, 8\rangle$. If σ happens to also pair 2 with 5, consider a different seed permutation φ where $\langle 3, 4\rangle$ is paired with $\langle 7, 8\rangle$. The new permutation φ will not have 2 paired with 5, because the $\langle 3, 4\rangle$ pair was already paired with $\langle 2, 5\rangle$ in σ. Thus for any quad there will always be a permutation in the seed set with the symbols in different pairs.

As before, each permutation of seed pairs has to be permuted by a $(n/2, 4)$-transet into the an interim set T to account for all the quads of pairs being in all the possible quads of positions, and each pair in T may be flipped or not flipped in all length-4 binary strings in all quads of positions, requiring $b(n/2, 4)$ binary string cover. Using the pairing of size 63 for $n = 10$, and also the fact that $F(5, 4) = 5! = 120$, and by Theorem 5.2, $b(5, 4) = 2^4 = 16$, so $F(10, 4) \leq 120 \cdot 16 \cdot 63 = 120960$.

In the case of a $(10, 5)$-transet, the seed set covering all pairs of pairs is still sufficient as the following discussion shows. Five symbols might be distributed in various ways. Since there are five symbols in the first quintuple $\tau = \langle t_1, t_2, \dots, t_5 \rangle$, which we desire to map to $\rho = \langle p_1, p_2, \dots, p_5 \rangle$, two pairs of even-odd adjacent symbols may appear, but the last symbol is by itself. For this case a seed set with every pair paired with all other pairs would leave the fifth symbol in a third pair automatically. For example to map $\langle 0, 1, 3, 4, 5 \rangle$ to $\langle 2, 5, 0, 3, 1 \rangle$, the pair $\langle 2, 5 \rangle$ is needed in the same permutation with $\langle 3, 1 \rangle$ because starting values $\langle 0, 1 \rangle$ are adjacent and so are $\langle 4, 5 \rangle$. Clearly 0 will be in a third pair. Then by permuting all the pairs in all 120 ways, and flipping all pairs in all 32 ways, all such pairs of pairs will be covered.

If only two symbols t_1 and t_2 are paired together in the first tuple, then for the remaining three symbols each must appear in its own pair. The symbols p_1 and p_2 in the second tuple associated with symbols t_1 and t_2 of the first tuple can always be found paired together in a seed permutation, say σ, and there will be some other pair that contains the next needed symbol t_3. If the other two t_4 and t_5 are paired together in σ, then any other permutation which contains t_1 and t_2 will not have t_4 and t_5 paired together, so choose one which has t_3 paired with any other symbol, and it will work. For example, to map $\langle 0, 2, 4, 6, 7 \rangle$ to $\langle 0, 2, 9, 3, 1 \rangle$, the symbols $\langle 3, 1 \rangle$

must appear in a pair together because 6 is adjacent to 7, and if we choose that permutation σ such that the unused symbol 8 is paired with the 9, the 9 will not pair with 0 or 2. If in σ the pair $\langle 0, 2\rangle$ happen to be paired together, that will only happen once for any permutation with $\langle 3, 1\rangle$ also in it. So choose a permutation φ where $\langle 3, 1\rangle$ is paired with $\langle 7, 9\rangle$ instead (this time 7 is the throwaway symbol), and φ must be a different permutation than σ because $\langle 7, 9\rangle$ and $\langle 8, 9\rangle$ cannot be in the same permutation. In the new permutation φ, $\langle 3, 1\rangle$ will not be paired with $\langle 0, 2\rangle$ again because those pairs met in σ, so 0 and 2 will be in separate pairs as required. This seed will then be permuted and flipped as needed.

If no symbols are paired together in the first tuple, each pair must contain one of the symbols. If in σ, t_1 is paired with another symbol x_i for some x_i distinct from the other t's, $\langle x_i, t_1\rangle$ can be found paired with the pair t_2 and some other uninvolved symbol x_j. If σ also pairs t_3 with t_4, then any other permutation φ with t_1 and x_i together will not pair t_3 with t_4, but it will have t_2 paired with yet another symbol x_k. If x_k happens to be t_5, then choose one more permutation μ with t_1 and x_i paired together, and none of the prior matchups will be there, so each other symbol will be on its own. For example, if we need to map $\langle 0, 2, 4, 6, 8\rangle$ to $\langle 0, 2, 5, 4, 7\rangle$, look at a permutation σ where $\langle 0, 1\rangle$ is paired with $\langle 2, 3\rangle$ because 1 and 3 are not in the second tuple. If σ happens to pair 5 with 4, look at a different permutation φ with $\langle 0, 1\rangle$ paired with $\langle 2, 6\rangle$. For sure 5 is not paired with 4 in φ, but if 5 is paired with 7 in φ we look at yet another permutation μ where $\langle 0, 1\rangle$ is paired with $\langle 2, 8\rangle$. Again μ will not have 5 with 4 or 5 with 7, but it might have 4 with 7. If so, there is one last permutation where $\langle 0, 1\rangle$ is paired with $\langle 2, 9\rangle$ where none of the prior pairs will be repeated.

The same seed set of size 63, then, will work to create a $(10, 5)$-transet. For quintuples all 120 permutations of the five pairs are needed, and a binary covering suite $b(5, 5)$. By Theorem 5.1, $b(5, 5) = 2^5 = 32$, so $F(10, 5) \leq 63 \cdot 120 \cdot 32 = 241920$. In the final analysis, this particular "pairs of pairs" method of finding covers turned out to be unproductive when compared with the methods described in Chapters 3 and 6. After the failure of the 6 and 10 cases to improve the upper bound using this method, however, the following theoretical technique for discovering such sets whenever $n = 2 + i^2$ for some integer i was determined, but has not yet been fully implemented and tested.

5.5 Complete graphs, cliques, hyper-tetrahedron, and pairs of pairs

There is not yet a general algorithm, but the idea behind finding ideal seed sets goes as follows. Each pair of symbols from a set can only be in a permutation with another pair which shares no symbols. For example, $\langle 0, 1\rangle$ can be paired with $\langle 2, 4\rangle$ but not with $\langle 0, 3\rangle$. As the permutation grows, more pairs are excluded until the last one is forced. The goal is to find $(n-1)(n-3) = n^2 - 4n + 3$ such permutations for any n.

The case where $n = 4$ is trivial. The identity pairs $\langle 0, 1\rangle$ with $\langle 2, 3\rangle$. Then since order within the pair is fixed and will be flipped by $b(2, 2)$ later, only two other pairs of pairs can be made. The next is $\langle 0, 2\rangle$ with $\langle 1, 3\rangle$ and finally $\langle 0, 3\rangle$ with $\langle 1, 2\rangle$. Note that if each symbol is a point in three dimensions, and each pair is a line, the collection of all the lines makes a tetrahedron. In each permutation, disjoint lines from the tetrahedron are chosen. This mix and match pattern appears to be fundamental to the general pattern of pairs.

The case where $n = 6$ can be thought of as a hyper-tetrahedron in 5 dimensions. Each time one pair of points is chosen, such as $\langle 0, 1 \rangle$, the remaining four points make up a tetrahedron which must be traversed in the same manner as in the $n = 4$ case. The first few permutations/pairs are in Table 5-7.

Table 5-7. Pairs of pairs n=6

0	1		2	3		4	5
0	1		2	4		3	5
0	1		2	5		3	4
0	2		1	3		4	5
0	2		1	4		3	5
0	2		1	5		3	4

Note that the pair $\langle 0, 1 \rangle$ remains fixed until the tetrahedron of the other four points is exhausted. Then the pattern is then continued for the next pair. This mixing and matching will never repeat two pairs together when $n = 6$, because the four numbers being mixed up are never the same four numbers; one is always being "borrowed" to pair with the 0.

The situation gets more complex as n increases. For $n = 8$, whenever the first pair is being held constant, only cyclic rotations can be made among the three remaining pairs without repeating, so only four permutations starting with $\langle 0, 1 \rangle$ can be created without repeating a pair, but five are required. It appears that the mixing and matching pattern can only occur when after fixing one pair, there are twice as many leftover pairs to match up as for a prior working set.

With $n = 4$, there was one pair left over. With $n = 6$ there are two pairs left over. With $n = 10$ there are four pairs left over.

Conjecture 5.1: A perfect set of pairs of pairs can be created whenever $n = 2^i + 2$ for some integer i.

For the case of $n = 10$, the search works as follows. For the first three permutations, proceed as for the $n = 6$ case, but the pairs $\langle 2, 3\rangle$ and $\langle 4, 5\rangle$ form a mixed tetrahedron at the same time as $\langle 6, 7\rangle$ and $\langle 8, 9\rangle$ do. Then two more permutations are made by mixing $\langle 2, 3\rangle$ with $\langle 6, 7\rangle$ at the same time as mixing $\langle 4, 5\rangle$ and $\langle 8, 9\rangle$ in the ways the have not already appeared. See Table 5-8 below. Finally the pairs $\langle 2, 3\rangle$ and $\langle 8, 9\rangle$ are combined, as are $\langle 4, 5\rangle$ and $\langle 6, 7\rangle$ for two more permutations.

Table 5-8. Pairs of pairs for n=10

0	1		2	3		4	5		6	7		8	9
0	1		2	4		3	5		6	8		7	9
0	1		2	5		3	4		6	9		7	8
0	1		2	6		3	7		4	8		5	9
0	1		2	7		3	6		4	9		5	8
0	1		2	8		3	9		4	6		5	7
0	1		2	9		3	8		4	7		5	6

This can be seen as a kind of higher dimension of the $n = 4$ case, where pairs are being cross-mixed in addition to single integers. This has to be done without breaking the fundamental rule:

Rule 1: No two pairs can appear together more than once.

This completes the set for $\langle 0, 1\rangle$ fixed. After this, $\langle 0, 1\rangle$ can never appear together as a pair again. After this a similar principle is followed, so that with the pair $\langle 0, 2\rangle$, the 1 is paired with the 3 to start with, and with $\langle 0, 3\rangle$ the 1 is paired with the 2. This completes the tetrahedron from the quad $\langle 0, 1, 2, 3\rangle$.

Rule 2: Once a tetrahedron has been started, it must always be completed.

However, the other quads from the identity have been entirely exhausted with one another, so new quads must be formed, similar to what happened automatically in the $n = 6$ case. So to proceed for the $\langle 0, 2\rangle$ case, as mentioned the $\langle 1, 3\rangle$ pair is next. Then 4 must be paired with something, but the rule to follow from here out is that at no time when a pair is completed can the same set of numbers remain.

Rule 3: Other than for rule 2, when a pair is completed the remaining numbers must not all be used up.

This means that 4 cannot be paired with 5, because the set {6, 7, 8, 9} have all filled out the last four positions before and have nothing new to add. So pair the 4 with 6, the 5 with 8 (because using 7 would leave 8 and 9 which breaks rule 2). Then 7 and 9 are paired together and we do the mix and match again between $\{1, 3, 4, 6\}$ and $\{5, 8, 7, 9\}$. However, while we say 7 and 9, it is not clear what the actual order of the $\{5, 8, 7, 9\}$ tetrahedron is. In fact, after $\langle 5, 8\rangle$ and $\langle 7, 9\rangle$, if we naively proceed to put $\langle 5, 7\rangle$ with $\langle 8, 9\rangle$ while pairing $\langle 1, 4\rangle$ with $\langle 3, 6\rangle$, the next permutation will have no choice but to repeat $\langle 3, 4\rangle$ with $\langle 7, 8\rangle$, which were already paired in Table 5-8. So instead we treat $\langle 7, 9\rangle$ as $\langle 9, 7\rangle$ and all is well.

Rule 4: Arrange tetrahedrons so that previously paired pairs do not repeat.

The next conjectured case that works is $n = 18$, for which $\langle 0, 1 \rangle$ can easily be matched by the procedure above, and with some searching compatible permutations starting with $\langle 0, 2 \rangle$ can be found, but the rest has not yet been discovered. In any event, the perfect pairs of pairs can be viewed in several interesting ways as a theoretical object of study.

For example, consider a graph $G(V, E)$ such that nodes in V represent pairs of distinct symbols from $\mathbb{Z}_n$, disregarding order. Let p denote the number of nodes in V. Then $p = n(n-1)/2 = (n^2 - n)/2$. For $s, t, u, v \in \mathbb{Z}_n$, consider a node st labeled by $\langle s, t \rangle$ and another labeled by $\langle u, v \rangle$. Node $\langle s, t \rangle$ has an edge in E to node $\langle u, v \rangle$ whenever s, t, u, v are distinct. Let w be the number of nodes with s or t in them. Since there are $n - 1$ nodes with any given symbol s in it, but exactly one node with both s and t in it, there are $w = 2(n-1) - 1 = 2n - 3$ nodes with either s or t in them. Let d be the degree of node st. The remaining nodes of G which do not have s or t in them are all connected to st, so there are $d = p - w - 1 = (n^2 - 5n + 6)/2$ nodes connected to the node st, but st is arbitrary so each node has degree d; that is, G is a d-regular graph. Thus $|E| = pd/2 = (n^4 - 6n^3 + 11n^2 - 6n)/8$.

A permutation consists of a clique of nodes from G of size $n/2$. No pair can be in more than one permutation with another pair, so once such a clique is found, all edges of that clique can be removed from the graph. Each node is connected once to every other node of the clique, so when the edges of a clique of $n/2$ nodes are removed, each node loses $n/2 - 1$ edges. Since each node has degree d, there are $n - 3$ times this can be done for each node; that is, each node can appear in $n - 3$ permutations. Each traversal of a clique can be considered part of an Eulerian tour of G. It is known that such a tour is possible whenever every node in G has an even degree, or there are only two nodes of odd degree. When $n \equiv 2 (mod\ 4)$, the number of edges

removed from each node is even, otherwise it is odd. So, after the first clique is removed, if $n/2$ is even, the graph is no longer Eulerian, so such a tour no longer exists. If $n/2$ is odd, the graph remains Eulerian until all edges are gone, as long as nodes are chosen to not disconnect the graph until it is necessary. The cliques will of course be maximal and of size $n/2$, but they must also be arranged such that every pair with 1 in it is paired with every other pair in one of the $n-2$ cliques. Thus a perfect pairing as described above represents a clique partition of the graph, which in general is difficult to accomplish.

For example, when $n = 10$, each clique is 5 nodes, and 10 edges are removed for each permutation. Each node s must appear in some permutation with the 28 other pairs that have no point in common. There are 63 permutations in all, and 45 nodes of degree 28, with 4 edges removed from each node for each permutation. Thus each node st can be in 7 permutations, and each permutation connects a pair st with 4 other pairs, so st can be matched with 28 other pairs as required. Since this method is useful for cases of $k = 4$ and $k = 5$, future research may yield improved upper bounds for many such cases.

CHAPTER 6

ALGORITHMS FOR TRANSITIVE PERMUTATION SETS

In this chapter, we examine the idea of redundancies in covering sets, how to reduce them, how to create covering sets recursively using Theorem 3.3, and how to find sharp covering sets if they exist. An implementation of each of these algorithms was created in the research leading to this dissertation. We also show calculations which were used to create **Table 1-2** of lower bounds and **Table 1-3** of upper bounds.

For a given (n, k)-transet M on a set χ, consider the pair (τ, ρ) where τ and ρ are k-tuples of symbols from χ^k. If there exist at least two permutations σ and φ in M which map τ to ρ, then M is said to have a *redundancy* on (τ, ρ). Let t be the number of ways to choose a k-tuple of symbols from χ. Then $t = \binom{n}{k}$. As discussed previously, a given permutation σ can map t k-tuples to t other k-tuples. For a permutation σ, let $\mathcal{R}_\sigma = \{R_1, R_2, \dots, R_t\}$ be the set of pairs of unique k-tuples $R_i = (\tau_i, \rho_i)$ such that σ maps τ_i to ρ_i. If for every pair R_i in $\mathcal{R}_\sigma$, the transet M has a redundancy on R_i, then σ is *completely redundant* in M and can be removed from M, and $M \backslash \{\sigma\}$ will remain an (n, k)-transet. The following describes a system to automate the process of removing completely redundant permutations.

6.1 Main constraint object

Consider a set M of permutations of $\mathbb{I}_n$. As noted previously, when in passive form, a permutation σ which maps a k-tuple $\tau = (t_1, t_2, \dots, t_k)$ to $\rho = (p_1, p_2, \dots, p_k)$ has the symbols of

ρ at the positions in σ, such that the $t_i{}^{th}$ position of σ contains the symbol p_i. For example, if $t_1 = 6$ and $p_1 = 3$, the 6th position of σ is 3. If for every k-tuple of symbols τ and every k-tuple of symbols ρ there exists a permutation σ in M with each symbol p_i of ρ in the $t_i{}^{th}$ position of σ, then M is an $(n,\ k)$-transet.

Define a constraint object Co for inputs M, n, and k whose purpose is to determine coverage for each k-tuple of symbols of $\mathbb{I}_n$ in each k-tuple of positions. We implement Co as a k-dimensional array of k-dimensional arrays of integers. There is one integer for each k-tuple in each k-tuple of positions. In fact, we can initially view Co as a 2k-dimensional array where for any k-tuple of symbols $\tau = \langle t_1, t_2, \dots, t_k \rangle$ and k-tuple of positions $\rho = \langle p_1, p_2, \dots, p_k \rangle$, the integer $Co(p_1, p_2, \dots, p_k, t_1, t_2, \dots, t_k)$ is the number of times τ is covered by a permutation σ in M at positions ρ.

For example, if $n = 3$ and $k = 2$, and $M = \{\langle 1, 3, 2\rangle, \langle 2, 1, 3\rangle\}$, then all possible k-tuples of positions $\mathcal{P}_{3,2} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$. Note that we need not store the fact that $\langle 1, 3, 2\rangle$ maps $\langle 2, 1\rangle$ to $\langle 3, 1\rangle$, because by symmetry we only need to check that $\langle 1, 2\rangle$ is mapped to $\langle 1, 3\rangle$. So, due to the permutation $\langle 1, 3, 2\rangle$, the value of $Co(1, 2, 1, 3)$ is 1, and $Co(1, 3, 1, 2)$ is also 1, as is $Co(2, 3, 3, 2)$. Then due to the permutation $\langle 2, 1, 3\rangle$, the value of $Co(1, 2, 2, 1)$ is 1, as is $Co(1, 3, 2, 3)$, and $Co(2, 3, 1, 3)$. This particular M is not a $(3, 2)$-transet as we will see.

It should be noted in general that several positions within the 2k-dimensional array Co are *null*, for example because one cannot have a k-tuple of symbols with a symbol repeated, or a k-tuple of positions with the same position repeated. Further, as seen in the example, we only need to store the in-order positions. Thus whenever $p_a > p_b$ and $a < b$, the associated entry in Co is *null*. In particular, define an *invalid prefix* as a prefix $i_1, i_2, \dots, i_j$ of a coordinate sequence

with repeated symbols or positions, or positions out of order. Then, for the sake of saving memory cells used by the program, the multi-dimensional array Co is instead implemented by a linked list structure. That is, for a prefix $i_1, i_2, \dots, i_j$ of a full sequence $(i_1, i_2, \dots, i_{2k})$, if the prefix sequence is valid, $Co(i_1, i_2, \dots, i_j)$ is a pointer to a linked list structure representing the portion of the array $Co(i_1, i_2, \dots, i_{2k})$ with $i_1, i_2, \dots, i_j$ as its first j co-ordinates. If instead $i_1, i_2, \dots, i_j$ is invalid, $Co(i_1, i_2, \dots, i_j)$ is *null*.

The constraint object is used in the following ways. For each permutation σ of M, each possible k-tuple of symbols is extracted and used to initialize Co. Since permutations are on $\mathbb{I}_n$, the first k symbols of each σ map $\langle 1, 2, \dots, k\rangle$ to $\langle \sigma(1), \sigma(2), \dots, \sigma(k)\rangle$, so during the initialization phase, first $Co(1, 2, \dots, k-1, k, \sigma(1), \sigma(2), \dots, \sigma(k-1), \sigma(k))$ is set to reflect coverage by one permutation. Then the next set of k positions are considered, by incrementing the last position counter. That is, $Co(1, 2, \dots, k-1, k+1, \sigma(1), \sigma(2), \dots, \sigma(k-1), \sigma(k+1))$ is updated. This continues up to $Co(1, 2, \dots, k-1, n, \sigma(1), \sigma(2), \dots, \sigma(k-1), \sigma(n))$. Then the last counter is reset and the second-last counter is incremented, so $Co(1, 2, \dots, k-2, k, k+1, \sigma(1), \sigma(2), \dots, \sigma(k-2), \sigma(k), \sigma(k+1))$ is updated, and finally $Co(n-k+1, n-k+2, \dots, n, \sigma(n-k+1), \sigma(n-k+2), \dots, \sigma(n))$ is updated. This is repeated for every permutation in M. Once Co is initialized, each possible k-tuple of symbols $\langle t_1, t_2, \dots, t_k\rangle$ at each possible k-tuple of positions $\langle p_1, p_2, \dots, p_k\rangle$ can be traversed to query for a value of at least 1 at each $Co(p_1, p_2, \dots, p_k, t_1, t_2, \dots, t_k)$. If all valid integers are at least 1, M is an (n, k)-transet.

To continue the example above, with $M = \{\langle 1, 3, 2\rangle, \langle 2, 1, 3\rangle\}$, we traverse $\mathcal{P}_{3,2} = \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 3\rangle\}$ for each k-tuple of symbols $T_{3,2} = \{\langle 1, 2\rangle,\ \langle 1, 3\rangle,\ \langle 2, 1\rangle,\ \langle 2, 3\rangle,\ \langle 3, 1\rangle,$

$\langle 3, 2\rangle\}$ and check the corresponding values of Co. Immediately we find that $Co(1, 2, 1, 2) = 0$, which is all that is needed to determine that M is not a transet.

The *trim algorithm* is used to remove extra redundancies in a given (n, k)-transet. Once M is determined to be an (n, k)-transet, each possible k-tuple of positions $\langle p_1, p_2, \dots, p_k\rangle$ can be traversed for each permutation σ of M to query for a value of at least two as the integer value corresponding to $Co(p_1, p_2, \dots, p_k, \sigma(p_1), \sigma(p_2), \dots, \sigma(p_k))$. If so, σ is completely redundant and is removed from M. The values of Co are updated to reflect one less permutation covering each tuple. This is repeated for every σ in M.

For example, if $n = 3$ and $k = 1$, and $M = \{\langle 1, 3, 2\rangle, \langle 1, 2, 3\rangle, \langle 2, 1, 3\rangle, \langle 3, 2, 1\rangle\}$, we traverse $\mathcal{P}_{3,1} = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle\}$ for each 1-tuple of symbols $T_{3,1} = \{\langle 1\rangle, \langle 2\rangle, \langle 3\rangle\}$ and check the corresponding values of Co. We find that M is a $(3, 1)$-transet, so we trim. The following table represents $Co(p, t)$.

Table 6-1. Constraint object

p=	1	2	3
t=1	*2*	*1*	*1*
t=2	*1*	*2*	*1*
t=3	*1*	*1*	*2*

The table indicates that the 1-tuple 2 is covered in position 2 twice, for example. To trim, iterate through the permutations of M and see if any are completely redundant. For the permutation $\langle 1, 3, 2\rangle$, the 1 in position 1 is covered twice, but the 3 in position 2 is only covered once, so this permutation is essential. For the permutation $\langle 1, 2, 3\rangle$, every symbol is covered

twice in every position, so $\langle 1, 2, 3 \rangle$ is completely redundant. Once it is removed all values in the table will be 1.

Experimental results determined that the order in which permutations are examined and removed makes a difference in how many redundant permutations can ultimately be removed. For example if all the even permutations are required to make a cover, and one even permutation is removed right away instead of some errant odd permutations, several odd permutations will be required to cover what one even permutation might have covered. In practice it is difficult to tell what will be the very best permutations to remove first, and trying all combinations is prohibitive; being essentially equivalent to the problem of finding a transet from scratch. Thus a greedy layered approach was used, where permutations were sorted by a tuned heuristic, and permutations were removed sequentially so that if there were some permutations for which all of its tuples had quadruple-coverage and some for which only some had triple-coverage and some had quadruple-coverage, the pure quadruple-coverage permutations would be removed first, then any remaining triple-coverage permutations, and so on.

Because isotopy classes preclude the idea that one symbol is in some way distinct from all others, the symmetric group S_n, always covers each tuple of symbols evenly. There are $\binom{n}{k}$ ways to choose k objects from a set of n objects when order is unimportant. Since the order of positions is unimportant and each permutation is n symbols, there are $\binom{n}{k}$ positions covered by each permutation. For k-tuples of symbols, order does matter, and k objects can be ordered $k!$ ways, so there are $k!\binom{n}{k}$ ways to choose an ordered k-tuple of distinct symbols from a set of n distinct symbols. So to cover all k-tuples of symbols in all k-tuples of positions, we need at most

$k!\binom{n}{k}\binom{n}{k}$ permutations. Each permutation covers a distinct k-tuple in $\binom{n}{k}$ distinct k-tuples of positions, so we need at least $k!\binom{n}{k}$ permutations to form an (n, k)-transet. And S_n has $n!$ permutations, so S_n has $n!/k!\binom{n}{k}$ times as many permutations as it needs. Therefore, when using S_n as a starting transet, each k-tuple of symbols in each k-tuple of positions is covered $n!/k!\binom{n}{k}$ times, which simplifies to $(n-k)!$ times. The highest amount of redundancy for a given (n, k)-transet determines how many iterations will have to be run in the trim algorithm.

A naïve approach to finding $F(n, k)$ is to start with S_n and run the constraint object trimming algorithm to remove the redundant permutations. This approach has a couple of problems. The first is that the problem of determining the order in which to remove redundant permutations is the same difficulty as the original problem of finding a minimal (n, k)-transet. The second problem is that for small k and large n, the size of S_n relative to the size of the minimal transet is very large, such that it is possible to represent the minimal transet in a computer, but not possible to represent S_n for large n. Finally, since each k-tuple of symbols in each k-tuple of positions is covered $(n-k)!$ times, $(n-k)!$ is quite large when n is large and k is small. Thus the trim algorithm will need to be run through a large number of iterations to evenly remove redundant permutations. However, theorems in Chapter 3 provide constructive methods for creating permutation covering sets from sets much smaller than $n!$ permutations, and the trim algorithm provides a way to remove some of the redundancies in those sets.

6.2 Branch-and-bound search for sharp transets

Unless otherwise stated, for the purposes of this section, if a shifted n-Latin square consists of n integers in increasing order $\langle a_1, a_2, \dots, a_n \rangle$, we will consider it to be equivalent to the n-Latin square formed by the integers $\langle 0, 1, \dots, n-1 \rangle$. We will use the terms shifted Latin square, Latin square and n-Latin square interchangeably. If we discuss a set of permutations all starting with the same $k-1$ symbols $\langle t_1, t_2, \dots, t_{k-1} \rangle$ chosen from a set T of $k-1$ symbols from $\mathbb{Z}_n$, and ending with a Latin square, we will be referring to the shifted Latin square created from the set $\{x \in \mathbb{Z}_n | \, x \notin T\}$ mapped to the set $\{0, 1, \dots, k-1\}$ in numeric order from least to most. For example, if a set M' of four permutations of length six all start with $\langle 3, 1 \rangle$, the rightmost four columns of M' comprise a shifted Latin square on the symbols $\{0, 2, 4, 5\}$. Consider the Latin square of size 4, to be the same square as the one with canonical mapping with 0 mapped to 0, 2 to 1, 4 to 2, and 5 to 3, for the purposes of this section.

For M, a sharp (n, k)-transet, there is no redundancy in M, and for any subset M' of M with $n-k+1$ permutations where the leftmost $k-1$ columns are identical, the rightmost $n-k+1$ columns form a shifted Latin square. Indeed grouping permutations with equal leftmost $k-1$ columns forms a partition on the set M. Each set of $n-k+1$ permutations is a *block*. There are n ways to choose the first column, $n-1$ ways to choose the second column, down to $n-k+2$ ways to choose the $(k-1)st$ column for a block. Thus there are $n(n-1)(n-2) \dots (n-k+2)$, or $(n)_{k-1}$ blocks of size $n-k+1$ that comprise a sharp (n, k)-transet.

The Latin square algorithm for finding sharp transets tries all known Latin squares of size $n-k+1$ for each of $(n)_{k-1}$ blocks until it finds an (n, k)-transet with no redundancies. For

larger values of $n-k+1$, there are many such Latin squares as pointed out in Chapter 4. Many shortcuts can be taken to bound this algorithm from having to try every Latin square. The following informal discussion explains these shortcuts and why they are valid.

To determine if $F(n,k)$ is $(n)_k$, we only need to find one sharp $(n,\ k)$-transet. By the following informal argument, we can assume without loss of generality that the first Latin square is in reduced form. Assuming we can find M, a sharp (n,k)-transet, we can sort M's permutations into lexicographic order. Then we can create M' by sorting the columns of M so that the identity permutation is the first permutation in M'. This will ensure that the first Latin square of M' is in reduced form. The set M' is in the same isotopy class as M, so it is also an (n,k)-transet, and no permutations were added so M' is also sharp. Thus we need only search the reduced form Latin squares for the first block.

In fact, we can assume that the first Latin square is one of the isotopy class representatives. The set of reduced Latin squares of a given size can be partitioned into equivalent isotopy classes over permutations of the symbols in a Latin square. Consider a sharp (n,k)-transet M starting with the block created with the shifted equivalent Π' of a reduced Latin square Π of size $s=n-k+1$. Given a set of Latin squares $L_{s,iso}$ consisting of a single representative of each isotopy class of Latin squares of size s, there exists a member Λ of $L_{s,iso}$ and a permutation π of $\mathbb{Z}_s$ such that $\Lambda \cdot \pi = \Pi$. If we shift π through the same function that was used to turn Π into Π', we have a permutation which can be inverted and run on the symbols of M to create M', a set in the same isotopy class with M and hence a sharp (n,k)-transet. The first block of M' can be created with the Latin square Λ. Thus, to search for a sharp (n,k)-transet, we need only use the set $L_{s,iso}$ for the first block.

Finally, to find the remaining blocks of M, we need not concern ourselves with the order of the rows of the Latin squares. This means that any squares which differ only in the order of their rows can be eliminated from the search. The easiest way to do this is to start with the reduced s-Latin squares, and try all possible column permutations for each one. Note that by Theorem 4.1, only the column permutations which start with one are needed to try all the row-reduced squares, and only row-reduced squares are required since row order does not matter.

The search algorithm works as follows. Since there is no redundancy in the finished sharp transet, the main constraint object can check the Latin squares for fit, and a backtrack occurs whenever a redundancy would be introduced by a square. Using the main constraint object, we add each permutation and whenever a value exceeds 1 we have a redundancy. Once the first Latin square is chosen, all the row-reduced Latin squares are tried for the next group. However, note that the row-reduced Latin squares are created from the reduced Latin squares by shuffling the rightmost columns of the square. First, one adds each row of the new square and looks for redundancy. Then, for the next row-reduced Latin square, proceed the same for every permutation of the rightmost columns. Any time an entire Latin square is added without creating redundancy, the next starting symbols are considered for the next Latin square. If any part of a square cannot be added without redundancy, the next square in the list is tried for that block. If all squares are tried, the prior block is removed and the next square is tried for the prior block.

Since every reduced Latin square starts with the identity permutation, the number of row-reduced Latin squares to check can be greatly reduced for any given block. Specifically, at the start of a new block, one can cycle through every s-length permutation π_i which starts with 1. For each such π_i, use it to permute the columns of just the first row consisting of the shifted

symbols of the block, and check whether that permuted row along with the fixed symbols can be added to the current transet candidate without creating redundancies, using the main constraint object. Since the shifted squares are all reduced, that first row is always the same. If a particular π_i-permuted row can be added without redundancies, tag π_i, remove the row, and try the next one. At the end of this process all the permutations which need to be tried as column permutations on the reduced Latin squares for that block have been tagged. In practice, instead of trying $(s-1)!$ column permutations of each reduced s-Latin square, it is necessary to try only an order of magnitude fewer permutations. In many cases, no permutation at all works, and the algorithm can immediately backtrack to the next square for the prior block. When all blocks have been satisfied, it is a sharp (n, k)-transet. If all squares are tried and no combination satisfies all blocks, the negative result is definitive, since every workable combination was tried.

6.3 Fnk program

In the course of this research, a program was developed called fnk which uses the main constraint object and other strategies to create transitive permutation sets. It is capable of reading existing transet files as well as seed files, and also of creating transet files from scratch for certain values of n and k. It can also verify whether a particular file is a transet or not. The program takes n and k as input on the command line, along with a parameter which establishes which kind of operation is to be performed. The following example command lines describe the parameters and the internal functionality which occurs based on known theorems and algorithms.

fnk 8 1

This detects the special case of $k = 1$ and outputs a cyclic-shifted Latin square of size 8, which of course is a sharp $(8, 1)$-transet.

fnk 9 3 e

This instructs the program to create all the even permutations of length 9, then use the main constraint object to trim the excess permutations, always maintaining a set which is a 3-transitive permutation set. The end result is often not an optimal set, unless $k = n - 2$.

fnk 6 3 i fnk74.txt

This parameter instructs the program to read a permutation set from the input file and reduce the permutations from the length they are in the file down to permutations of length n, and determine if the result is a $(6, 3)$-transet. If the permutation is already the correct length, this operation determines if the input set is a $(6, 3)$-transet. If it is, the set is then trimmed to remove redundancies. The reduction to a set of smaller permutations is carried out according to Theorem 1.6 in Chapter 1. For example to change a length-7 permutation σ on $\mathbb{I}_7$ in passive form to a length-6 permutation while maintaining 3-transitivity, move whatever is in the 7^{th} position of σ to the position of σ which has a 7 in it. If σ has a 7 at the end, just remove it. For example, $(2, 1, 3, 7, 4, 5, 6)$ becomes $(2, 1, 3, 6, 4, 5)$. As proven in [13], this operation maintains the integrity of the k-transitivity over the length $n - 1$.

fnk 7 4 r fnk64.txt fnk63.txt

This command builds a recursive definition of a $(7, 4)$-transet from the input files according to the construction in Theorem 3.3. The result is then trimmed to remove

redundancies. The input files are expected to include a $(6, 4)$-transet and a $(6, 3)$-transet, respectively.

fnk 12 3 f seed12.txt fnk63.txt b63.txt

This command operates on a seed file of permutations of length n, and an existing $(n/2, k)$-transet and a binary cover $BCS(n/2, k)$. It creates an (n, k)-transet according to the construction in Chapter 5. The seed set can be a round-robin tournament for $k = 3$, or a list of pairs of pairs if $k = 4$ or $k = 5$.

fnk 6 p

This command uses brute force search to find the 15 length-6 permutations which comprise the exact pairs of pairs set as described in Chapter 5. The only values of n for which this has generated results so far are 4, 6, and 10.

fnk 8 3 l isotopy6.txt reduced6.txt

This command reads the Latin square isotopy file which lists the isotopy class members for 6-Latin squares, and the reduced file which contains all reduced 6-Latin squares. It then tries all Latin squares in combination from both files to attempt to create a sharp (n, k)-transet, in this case an $(8, 3)$-transet, using the methods described in Chapter 4.

6.4 Determination of results

Using the technique from section 6.2 and the 7-Latin row-reduced squares, a sharp $(8, 2)$-transet was found. The same technique and source squares were able to find a sharp $(9, 3)$-transet, and also to demonstrate that no sharp $(10, 4)$-transet exists. By the Corollary 3.1, we know that $F(10, 4) \geq (10)_4 + 1$, and in general $F(10 + i, 4 + i) \geq 5041 \cdot (10 + i)_i$ for

positive integer i. This is result 3.2.4 in Chapter 1. Although a sharp $(7, 2)$-transet was already known due to 7 being a prime, a sharp $(8, 3)$-transet was discovered using the method from section 6.2 along with the 6-Latin row-reduced squares. It was also discovered that no sharp $(9, 4)$-transet exists, and the diagonal lower bounds were raised accordingly giving result 3.2.3 in Chapter 1.

The longer searches using the 7-Latin squares for $F(8, 2)$, $F(9, 3)$ and the negative result for a sharp $(10, 4)$-transet required up to a week to complete. The number of reduced 7-Latin squares is 16,942,080[20], but the number of reduced 8-Latin squares is 535,281,401,856 which is four orders of magnitude larger. Clearly the time required for an exhaustive search for a sharp $(9, 2)$-transet using the 8-Latin row-reduced squares would be impractical over this space. However, an empirical result was discovered for the other known sharp transets, and that result enabled a much narrower search for positive results. Unfortunately the empirical results meant leaving out the majority of the 8-Latin squares, so any negative results are not definitive without a theorem proving that all sharp transets must follow this pattern.

It was discovered that every Latin square which resulted in a complete sharp $(n,\ k)$-transet of size $s = n - k + 1$ uses s-Latin squares in the same Latin square isotopy class as the s-Latin square formed by the s-length identity permutation $\langle 0, 1, \ldots, s - 1\rangle$ and its cyclic left shifts. For example, for a $(6, 3)$-transet, $s = 6 - 3 + 1 = 4$, so the isotopy class representative for the first block is:

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$$

Figure 6-1. Cyclic shift Latin square

Of interest is the fact that **Figure 6-1** is the Cayley table for $\mathbb{Z}_4$.The first block for $F(6,3)$ begins with $\langle 0,1 \rangle$, so the corresponding shifted Latin square is:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 3 & 4 & 5 & 2 \\ 0 & 1 & 4 & 5 & 2 & 3 \\ 0 & 1 & 5 & 2 & 3 & 4 \end{bmatrix}$$

Figure 6-2. First block for sharp (6, 3)-transet

The remaining blocks use as a basis a 4-Latin square from the same isotopy class as **Figure 6-2**, with a permutation of the symbols and a permutation of the columns. Permuting the rows makes no difference to a transet, so row permutations were ignored.

This cyclic row-reduced pattern carries through all the known sharp solutions, except the two created by the Mathieu groups. So, on the hypothesis that other sharp transets might use this same pattern, a new more restricted search was set up using 8-Latin squares up to 11-Latin squares. A single isotopy class representative was needed for the first square, so the length 8 identity permutation cyclic-shifted left was used for the first square. Then all 8! symbol permutations were generated, and sorted into reduced form. Once sorted, there were many duplicates created by this process, so duplicates were eliminated and only 1260 remained. This enabled a very quick search of all these for the remaining blocks of a $(9,2)$-transet by the column permutation method described above, and a sharp $(9,2)$-transet was discovered. The

same technique and pattern was applied to discover a sharp $(10, 3)$-transet, but no sharp $(11, 4)$-transet was discovered. This negative result did not eliminate the possibility of a sharp $(11, 4)$-transet however; indeed one exists from the Mathieu group $\mathcal{M}_{11}$. There is also a sharp $(12, 5)$-transet created from $\mathcal{M}_{12}$.

The same process was carried out for 9-Latin squares from the cyclic shift of the length 9 identity permutation, creating 6720 reduced squares. These were searched for a sharp $(10,\ 2)$-transet without success. This failure does not prove no sharp $(10, 2)$-transet exists to reinforce Lam's result in [19], but it does prove that the cyclic row-reduced method will not work for a sharp $(11, 3)$-transet or others on this diagonal, so no more 9-Latin square searches were done. Nonetheless, Theorem 1.11 implies no sharp $(10, 2)$-transet exists, which is result 3.2.5 from Chapter 1. Likewise, Theorem 1.10 implies result 3.2.6. The process was applied to 10-Latin squares from the cyclic shift, which resulted in 90720 reduced squares. A sharp $(11,\ 2)$-transet was already known, but this method did reveal a sharp $(12, 3)$-transet. No $(13, 4)$-transet was discovered, but existence was not eliminated. There are 362,880 reduced squares in the isotopy class for 11-Latin squares from the cyclic shift, but the search on these produced no results for a $(12, 2)$-transet.

In all, no entirely new sharply transitive sets were discovered. However, many heretofore unknown results for non-sharp transitive permutation sets were discovered using a combination of the recursive construction in Theorem 3.3 and the main constraint object to reduce inherent redundancies created by it, along with known results from group theory. For example, it is known that $|A_6| = 360$, a sharp result for $F(6, 4)$, and also $|PGL(2, 5)| = 120$, a sharp result for $F(6, 3)$. By Theorem 3.3, we can construct M, a $(7, 4)$-transet, from these two sets such that

$|M| = 120 \times 6 + 360 = 1080$. Hence $F(7,4) \leq 1080$. By the main constraint object removal of redundancies, we were able to produce M', a $(7,4)$-transet such that $|M'| = 1020$ by removing 60 unneeded permutations. Since $|PGL(2,7)| = 336$, we have a sharp $(8,3)$-transet from which we can construct a $(7,3)$-transet by Theorem 1.6 from Chapter 1. Then that new $(7,3)$-transet can be used along with M' to construct an $(8,4)$-transet of cardinality $336 \times 7 + 1020 = 3372$, which when reduced by the main constraint object has a cardinality of 3265. **Table 1-3** gives the latest upper bounds as computed by the combination of all these methods.

Unfortunately no Latin square search turned up a sharp (n,k)-transet that could not be generated from group theory methods because all those found were $(q+1,3)$-transets or $(q,2)$-transets for q a prime power. So, the Latin square searches were only good for finding negative results, but those negative results did improve the lower bounds, as was noted in **Table 1-2**.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

A new theorem allowing the creation of infinitely many transitive sets of permutations from other known sets was presented. Analysis of the properties of such transets resulted in the discovery of connections of this problem to sequences of Latin squares, which allowed searches through known Latin squares for such sequences. The result was the determination that for certain values of n and k, there is no sharply k-transitive set of n-permutations. This, coupled with observed properties of $(n+1, k+1)$-transets allowed us to vastly improve lower bounds in infinitely many cases. Theorems from related areas of mathematics were connected to the central problem, which allowed us to find infinitely many exact values for $F(n, k)$. New techniques were found which show promise for improving bounds when $k = 4$ and 5, and an application of software testing suites was added to the toolset.

The application of these results will definitely allow for the creation of more efficient symmetric networks, but there are also other applications of k-transitive sets of permutations in the field of experimental design, software testing, and perhaps other as yet unknown fields. The application of known results about symmetry to computer science has always been fruitful, and perhaps computer searches can fill in some of the blanks left by purely mathematical methods.

A program which automatically finds redundant permutations in generated transets was developed, which appears to be able to always find such redundancies without rearranging the generated permutations in any way when sets are not sharp. This leads us to the conclusion that

the recursive creation of transets has inherent redundancies in it, which might be eliminated by continued theoretical work. For example, the sets created from $(n, n-3)$-transets and $(n, n-2)$-transets result in $(n+1, n-2)$-transets which have many duplicated even permutations involved in their construction. Identifying sets of these permutations which are known to be even and known to be redundant may reduce the upper bound for infinitely many sets of permutations along the $(n, n-3)$ diagonal. Other similar results are being investigated.

A new version of this program may allow us to search further and/or faster for transets or redundancies. So far, a limitation of the program is the amount of memory used. As expected, the main constraint objects grows quickly for higher k, because it is $2k$ dimensions. New methods are being investigated to reduce this memory usage, or at least partition it so that physical memory need not be used for the entire array. Additionally, a program is being created which will allow the investigation of lower bounds, such as for $F(6, 2)$, by running simulated annealing on the sets of permutations and determining if some of the inherent redundancies in the known set of 37 permutations can be aggregated into one permutation, making it completely redundant so that it can be removed. In parallel with this, theoretical investigations on Conjecture 1.2 are being undertaken to determine if results from permutation codes can be leveraged to prove a theorem.

Using techniques described here, future results may also be obtained for $F(12, 2)$, equivalent to the projective plane of order 12, which is the smallest set for which an exact or nonexistence result is not known. Some of the classification theorems for transitive permutation sets, such as Theorem 2.3, were discovered later in the process after computer time was already committed to solving the nonexistence of the $F(10, 4)$ sharp case. If this or other new tools are

useful for analyzing the $F(12, 2)$ case, they can also be used to reconfirm the computer-generated result of Lam, et al. for projective planes of order 10, which has never been verified [19]. Other areas of research include finding better recursive formulas for non-sharp cases, better tools for removing redundancies in the most efficient way, and more theorems about the structure of transitive sets of permutations. Theoretical results might also determine that a certain formulaic number of redundancies can always be removed, which will give a better upper bound for infinitely many cases. Research in these areas is currently underway, and this area promises to yield empirical and theoretical results for a long time to come.

BIBLIOGRAPHY

[1] S. Akers and B. Krihsnamurthy, "A group-theoretic model for symmetric interconnection neworks," *IEEE Trans. Comput.*, vol. C-38, no. 4, pp. 555-566, 1989.

[2] M. Heydari and I. H. Sudborough, "On the Diameter of the Pancake Network," *Journal of Algorithms*, vol. 25, no. 1, pp. 67-94, 1997.

[3] S. Akers, D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," in *Proc. Int'l Conf. Parallel Processing*, 1987, pp. 393-400.

[4] V. Bafna and P. Pevzner, "Genome rearrangements and sorting by reversals," in *34th IEEE Symposium on Foundations of Computer Science*, 1993, pp. 148-157.

[5] G. E. Box, J. S. Hunter, and W. G. Hunter, *Statistics for Experimenters: Design, Innovation, and Discovery"*, 2nd ed.: Wiley, 2005.

[6] S. R. Dalal and C. L. Mallows, "Factor-Covering Designs for Testing Software," *Technometrics*, pp. 234-243, August 1998.

[7] H. Tarnanen, "Upper bounds on permutation codes via linear programming," *European Journal of Combinatorics*, vol. 20, pp. 101-114, 1999.

[8] W. J. Martin and B. E. Sagan, "A new notion of transitivity for groups and permutations," *Journal of the London Mathematical Society*, vol. 73, no. 1, pp. 1-13, 2006.

[9] N. L. Biggs and A. T. White, "Transitivity," in *Permutation Groups and Combinatorial Structures*. London: Cambridge University Press, 1979, ch. 1, pp. 5-9.

[10] R. H. Bruck and H. J. Ryser, "The nonexistence of certain finite projective planes," *Canadian Journal of Mathematics*, vol. 1, pp. 88-93, 1949.

[11] B. Monien and I. H. Sudborough, "Embedding one Interconnection Network into Another," in *Computational Graph Theory*.: Springer-Verlag, 1990, pp. 257-282.

[12] S. Latifi, "On the fault-diameter of the star graph," *Information Processing Letters*, vol. 46, no. 3, pp. 143-150, June 1993.

[13] W. Bein, S. Latifi, L. Morales, and I. H. Sudborough, "Bounding the Size of k-Tuple Covers," in *42nd Hawaii International Conf. on System Sciences*, 2009, pp. 1-8.

[14] P. J. Cameron and I. M. Wanless, "Covering radius for sets of permutations," *Discrete Math*, vol. 293, no. 91-109, 2005.

[15] L. Batten and A. Beutelspacher, "Affine spaces and projective spaces," in *The Theory of Finite Linear Spaces*.: Cambrige University Press, 1993, ch. 1, pp. 1-20.

[16] B. Huppert and N. Blackburn, *Finite Groups III*. New York, Germany: Springer-Verlag, 1982.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Standard notations and common functions," in *Introduction to Algorithms*, 2nd ed.: MIT Press, 2001, ch. 3, pp. 51-61.

[18] Quan Nguyen, "A communication on the k-tuple cover problem," University of Texas at Dallas, Richardson TX, e-mail 2010.

[19] C. W. Lam, L. Thiel, and S. Swiercz, "The non-existence of finite projective planes of order 10," *Canadian Journal of Mathematics*, vol. 41, pp. 1117-1123, 1989.

[20] B. D. McKay and I. M. Wanless, "On the Number of Latin Squares," *Ann. Combin.*, vol. 9, pp. 335-344, 2005.

[21] B. McKay. (2011, November) Latin squares. [Online]. http://cs.anu.edu.au/~bdm/data/latin.html

[22] P. J. Cameron, "Finite Geometries," in *Handbook of Combinatorics*, R. L. Graham, Ed.: The MIT Press, 1995, ch. 13, pp. 647-692.

[23] John B. Fraleigh, "Cycles and Cyclic Notation," in *A First Course in Abstract Algebra*.: Addison-Wesley, 1976, ch. 5, pp. 45-50.

[24] The GAP Group. (2008) GAP -- Groups, Algorithms, and Programming, Version 4.4.12. [Online]. http://www.gap-system.org

[25] W. Kerby, *On infinite sharply multiply transitive groups*. Göttingen: Vandenhoen and Ruprect, 1974.

[26] R. D. Carmichael, "Algebras of certain doubly transitive groups," *American Journal of Mathematics*, vol. 53, no. 3, pp. 631-644, July 1931.

[27] N. L. Biggs and A. T. White, "Finite Geometries," in *Permutation Groups and Combinatorial Structures*.: Cambridge University Press, 1979, ch. 2, pp. 24-52.

[28] Dieter Jungnickel, "Graphs, subgraphs and factors," in *Graphs, Networks and Algorithms*.: Springer-Verlag, 2005, ch. 1, pp. 2-5.

[29] E. Mendelsohn and A. Rosa, "One-factorizations of the complete graph - A survey," *Journal of Graph Theory*, vol. 9, no. 1, pp. 43-65, Spring 1985.

[30] A. Bonisoli and P. Quattrocchi, "Each Invertible Sharply d-Transitive Finite Permutation Set with d>=4 is a Group," *Journal of Algebraic Combinatorics*, vol. 12, pp. 241-250, 2000.

[31] J. Nagura, "On the interval containing at least one prime number," *Proc. Japan Acad.*, vol. 28, no. 4, pp. 177-181, 1952.

VITA

William Fahle has worked in computers for over 20 years, specializing in the area of multimedia computer graphics, animation and sound. He has designed and implemented systems of supporting technology including 2D and 3D graphics engines, audio playback, scripting language compilers, interpreters and debuggers, and worked with various file formats for code, sound and graphics. For more than 10 years, he has put these skills in the service of educational products delivered via the internet and web to elementary and secondary schools throughout the United States. He has also taught classes of technical artist/programmers both informally and in a for-credit university setting. He remains dedicated to the cause of education in many ways.

Publications:

"An (18/11)n upper bound for sorting by prefix reversals." Theoretical Computer Science 410(36):3372-3390, (2009)

"Improved bounds on k-transitive permutation sets." To appear.

U.S. Patent Number 6,269,367, "System and method for automated identification, remediation, and verification of computer program code fragments with variable confidence factors"