DEVELOPING TRIAD DESIGN ALGORITHMS BASED ON COMPATIBLE FACTORIZATION

HASLINDA IBRAHIM

ZURNI OMAR

RAHELA ABDUL RAHIM

COLLEGE OF ARTS AND SCIENCES

UNIVERSITY UTARA MALAYSIA

JANUARY 2011

ABSTRACT

We consider the problem of listing distinct triples that satisfy certain properties. This problem is known as triad design of order $v, TD(v)$. This design exists for $v \equiv 1$ or 5 (mod v). Much of our work deals with the enumeration of several triad design, for example $TD(7), TD(11)$ and $TD(13)$. These processes have helped us develop algorithm for triad design, the objective of this study. A new technique for triad design algorithm, known as Interval Generation Method was employed to construct $TD(6n + 1)$ and $TD(6n + 5)$. This method depends on analyzing the pattern of triples in the design to build starters. We begin by producing starters from Interval Generation Method as the initial block to begin with. Then the algorithm begins by cycling modular v from the initial block and finishes when the process approaches the initial block. The algorithms for $TD(6n+1)$ and $TD(6n+5)$ are presented in Chapter 4 and 5, respectively. As the entire study depends mainly on $TD(v)$ algorithms, new and remarkable theorems and lemmas for $TD(v)$ development are presented and proved.

ACKNOWLEDGEMENTS

First and foremost, we would like to extend our sincere thanks to The Higher Education, and UUM for providing this research grant, thus making this research possible. Department of Quantitative Sciences, College of Arts and Sciences has provided the support and equipment we needed to produce and complete our research.

We also attribute our appreciation to our colleagues and friends who had provided guidance and ideas as well as encouragement that definitely encouraged us to complete this research. Without their encouragement and effort, this report would not have been completed or written.

We also wish to express our gratitude to the officials and other staff members of Research and Innovation Management Center of UUM who rendered their help during the period of our research work. Last but not least, we wish to avail ourselves of this opportunity to express a sense of gratitude and love to our beloved family members for their manual support, strength, and help.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to combinatorial design

Combinatorics domain covers a broad spectrum in discrete structures and their properties. This includes important topics such as coding theory, combinatorial design, and graph theory. Many modern scientific advances have involved the use of combinatorial structures to model the physical world.

Coding theory focuses on the removal of redundancy and the correction (or detection) of errors in the transmitted data. It also includes the study of the properties of codes and their fitness for a specific application. The two fundamental aspects in coding theory are data compression (source coding) and error correction (channel coding). Numerous applications of coding theory can be found in information theory, electrical engineering, mathematics, and computer science.

Whilst, combinatorial design is the study of arranging elements of a finite set into patterns (subsets, array) according to specified rules. These conditions usually include incidence, set intersection, set containment or other similar conditions. A comprehensive discussion for combinatorial design can be found in [12].

In graph theory, a graph can be defined as points (vertices, nodes) and lines (edge, arcs). Graphs can be represented by diagrams in which the elements are shown as points and the relation as lines joining pairs of points. Graphs appear in a wide range of settings and account for a large portion of real world data sets. For example, in computer networks, the points are routers and lines represent the connection between two routers; in biology, the points are proteins and the lines represent the interaction between two proteins. Thus the importance of graphs is that, as basic mathematical structures, they arise in diverse contexts, both theoretical and applied.

1.2 Combinatorial Design

The main idea of combinatorial design theory can be traced back to the work of Euler, who in 1782 introduced the ”36 officer’s problem” and began the search for pairs of orthogonal Latin squares that lasted 175 years [7]. Numerous studies has been conducted in this area for example the current research in the mid-19th century, Kirkman, Steiner and Cayley worked on such topics as triple systems, Room squares, and other combinatorial objects that are still of interest to modern researchers. The modern history of design theory (since the mid-20th century) includes the seminal work by Bose, Ryser, Hanani and Hall. In recent years, there has been an incredible growth in design theory. Many classical problems have been solved. However, there are many interesting and important questions still to be answered. In recent years, combinatorial design theory has also become quite interdisciplinary with researchers found in both mathematics and computer science departments as well as occasionally in engineering or applied mathematics groups and in industrial groups.

The fundamental domain of combinatorial design theory is classified into two designs, namely latin squares and triple systems. The development and theoretical constructions of latin squares can be found in various books[5, 10, 12]. Meanwhile, a short compendium of the literature on triple systems and related topics are discussed in [5, 6, 11, 12]. However in this study, we only concentrate on triple system design.

1.3 Triple system

Triple systems are generalization of graphs. They have many relationships with other areas of mathematics for example, geometry, algebra, group theory, finite fields, and cyclotomy. Various applications of triple systems can be found in coding theory, cryptography, computer science, and statistics. Such a rich set of connections has made the study of triple systems an extensive field of combinatorics .

The study of triple systems originated from Thomas P. Kirkman (1806-1895) in 1847 and later pursued by Jacob Steiner, a Swiss-born German mathematician (1796-1863), in 1850s. A classical example for triple is

"A teacher would like to take 15 schoolgirls out for a walk, the girls being arranged in 5 rows of three. The teacher would like to ensure equal chances of friendship between any two girls. Hence it is desirable to find different row arrangement for the seven days of the week such that any pair of girls walks in the same row exactly one day of the week".

A solution for this problem is given below:

Table 1.1: Kirkman's schoolgirl problem

Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
1 2 3	1 4 5	1 6 7	1 8 9	1 10 11	1 12 13	1 14 15
4 8 12	2 8 10	2 9 11	2 12 15	2 13 14	2 4 6	2 5 7
5 10 14	3 13 15	3 12 14	3 5 6	3 4 7	3 9 10	3 8 11
6 11 13	6 9 14	4 10 15	4 11 14	5 9 12	5 11 15	4 9 13
7 9 15	7 11 12	5 8 13	7 10 13	6 8 15	7 8 14	6 10 12

In Table 1.1, we found in each row a pair of schoolgirl will walk together exactly once for example $\{1,2\}, \{1,3\}, \{2,3\}, \{1,4\}, \{1,5\}, \{4,5\}, \ldots, \{1,14\}, \{1,15\}, \{14,15\}$. From that on, numerous researches have constructed different types of triple system as summarized in Table 1.2.

Table 1.2: Survey of Triple System

Name of triple	Abbreviation	The first work
Kirkman Triple System	KTS (v)	1847
Steiner Triple System	STS (v)	1850
Mendelssohn Triple System	MTS (v, λ)	1971
Direct Triple System	DTS (v, λ)	1973
Hybrid Triple System	HTS (v, λ)	1989
Partial Triple System	PTS (v)	1997
Triple Design (Triad Design)	TD (v)	2003

Based on Table 1.2, we would like to focus our study on triad design. In this design, algorithm for general triad design will be developed.

1.4 Motivation of the study

Our study has its origin in a request to construct tournaments that are suitable for use in the paintball game. For this purpose, Wallis ([13]), constructed the designs in which teams compete three at a time and in which the order of the teams is significant. For example a group of team $\{1, 2, 3\}$ is not the same as a group of team $\{2, 1, 3\}$. However, we make some modifications in which the order of teams is significant. For example a group of team $\{1, 2, 3\}$ and a group of team $\{2, 1, 3\}$ are the same.

Consider the following combinatorial problem: we are given v teams and we wish to construct a tournament in which team j plays exactly $(v - 1)/2$ times, other teams play equally often and each pair of teams plays precisely once in each round. We would like to have each three teams play exactly once in the tournament. Table 1.3 provides the desired tournament that we want to produce for 7 teams.

Table 1.3: Tournament for 7 teams

127	136	145	235	764
231	247	256	346	175
342	351	367	457	216
453	462	471	561	327
564	573	512	672	431
675	614	623	713	542
716	725	734	124	653

One of such triples (three teams) is called a triad and is denoted by $TD(v)$. The question that arises from this construction is how to develop the design. Triad design on v objects exist when $v = 6n + 1$ and $v = 6n + 3$. The method for constructing triad designs for $v = 5$ and $v = 7$ were illustrated in [8]. These constructions were done by using brute-force method. However, this method is not suitable becasue the number of objects continue to increase, forcing us to enumerate all the cases. Thus, we need to develop algorithms for constructing triad designs for $v \equiv 1$ or 5 (mod 6).

1.5 Objectives

This study embarks on the following objectives:

(i) to establish patterns/structures for constructing triad design

(ii) to formulate theorems for constructing triad design

(iii) to develop generalized algorithms for triad design

1.6 Organization of the study

In this study we will develop algorithm for triad design. Since our study focuses on algorithm development for triad design, the steps to construct the design will be discussed in Chapter 3, 4 and 5. Let us give our plan in details.
Basic notation, concepts and terminology concerning especially triple system are discussed in Chapter 2, including specific definitions and concepts.

In Chapter 3, we consider the construction of compatible factorizations ($CF(v)$) as a basis for triad design algorithm. We use the idea of difference set method for the starter in CFS.

Motivated by starter set generation for CFS, we introduce the idea of Interval Generation Method to develop starter $TD(v)$, or known as $STD(v)$. In Chapter 4 and 5, we present algorithms for $TD(6n+1)$ and $TD(6n+5)$ respectively. In addition, these chapters provide and prove some new theoretical works towards triad design algorithm development.

Finally, in Chapter 6, we summarize our major contributions and briefly describe some related problems. These problems provide some possible directions for future research.

Chapter 2

Fundamental Concepts

The exposition in this chapter is based on [5, 6, 11, 12]. We first review some basic concepts and known results about graph theory. Also we present one-factorization and near-one-factorization that are the basic of developing compatible factorization and triad designs.

2.1 Some Basic Concepts from Graph Theory

We start with the following definitions:

Definition 2.1.1. *A graph $G = (V, E)$, consists of a finite set V of objects called vertices together with a set E of unordered pairs of vertices called edges.*

Definition 2.1.2. *The order of a graph G, denoted by $|G|$, is the number of elements of V, while $|V|$ denotes the number of edges.*

Definition 2.1.3. *The complete graph is a graph with exactly one edge connecting each pair of distinct vertices and no loops .*

The complete graph K_n on n vertices is of particular importance in design theory.

Definition 2.1.4. *A complete graph is a graph having n vertices in which every pair of vertices is joined by an edge. The degree of any vertex is* $d(v) = n - 1$*, and the number of its edges is* $\binom{n}{2} = \frac{n(n-1)}{2}$.

Clearly the complete graph K_n is a regular graph of degree $n - 1$. Figure 2.1 shows the graph of K_3, K_4, K_5 and K_6.

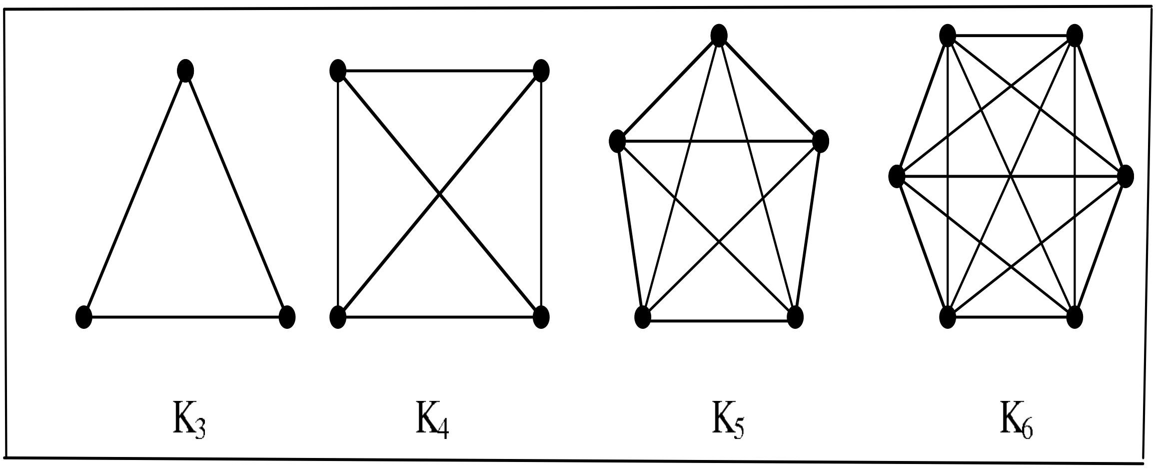

Figure 2.1: Complete graphs

$G_1(V_1, E_1)$ is a subgraph of the graph $G = (V, E)$ if $\emptyset \neq V1 \subseteq V$ and $E_1 \subseteq E$. If $V_1 = V$, then $G_1(V_1, E_1)$ is a spanning subgraph of $G = (V, E)$. Two graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic if there is a bijective function $f : V \longrightarrow W$ such that for all $v, w \in V$: the edge $vw \in E$ the edge $f(v)f(w) \in F$. For standard graph-theoretic terminology the reader is referred to [1, 2, 3, 5].

2.2 Balanced Incomplete Block Designs and Triple systems

We now introduce a relation between triple systems and Balanced Incomplete Block Designs (BIBD). Let V be a set of v elements called points and B a collection of non-empty subsets of V called blocks. (V, B) is a Balanced Incomplete Block Design (BIBD), if the following conditions are satisfied.

i) Each point is contained in exactly r blocks.

ii) Each block contains exactly k points.

iii) Every pair of distinct points is contained in exactly λ blocks.

With positive integer parameters (v, b, r, k, λ), this is denoted as BIBD (v, b, r, k, λ). BIBD has its origin in statistical design of experiments [4], and recently it has been used in many applications such as coding theory and scheduling sports leagues among others. In literature BIBD (v, b, r, k, λ) is often referred to as (v, k, λ)-BIBD, or alternatively $2 - (v, k, \lambda)$ design.

Triple system is a special case of BIBD with $k = 3$, and Steiner triple system is a BIBD with $k = 3$ and $\lambda = 1$. These are denoted as $TS(v, \lambda)$ and $STS(v)$ respectively. A triple system $TS(v, \lambda)$ is a pair (V, B) such that every 2-subsets of V are contained in exactly triple, where V is a set of n elements and B is a collection of 3-subsets of V called triples. The number $|v| = n$ is called the order of (V, B), and λ is called the index.

Theorem 2.2.1. *In a (v, k, λ)- design with b blocks each element occurs in r blocks where:*

i) $\lambda(v - 1) = r(k - 1)$

ii) $bk = vr$

Proof. Assume we have a design that has r blocks. If the design contains a given element say x, then in each of these blocks x makes a pair with $k-1$ other elements exactly λ times, hence we have

$$\lambda(v-1) = r(k-1)$$

Next, note that each block has k elements, so there are bk appearances of elements in the blocks. But each of the v elements appears in r blocks, thus

$$bk = vr$$

□

2.3 Difference Set

This section will explain the concept of *difference set* that we needed in generating starter in compatible factorization, will be discussed in Chapter 3.

Definition 2.3.1. *Let G be a group of order v, with the operation $+$. A k-subset D of G is called a difference set mod v, if each non-identity element $g \in G$ can be written in precisely λ different ways in the form $x - y$ for $x, y \in D$ (where λ is constant).*

Example 2.3.2. *Let $G = Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$ be the additive group of integers (mod 7). Clearly, $v = 7$, and if $D = \{1, 2, 4\}$, then the differences are $1 - 2 = -1 \equiv 6$ (mod 7), $1 - 4 = -3 \equiv 4$ (mod 7), $2 - 1 = 1$(mod 7), $2 - 4 = -2 \equiv 5$(mod 7), $4 - 1 = 3 \equiv 3$(mod 7). These differences are illustrated in Table 2.1.*

It can be seen that each non-zero integers $1, 2, 3, 4, 5, 6$ in Z_7 occurs exactly once in the off-diagonal position and hence exactly once as a difference. Hence D is a $(7, 3, 1)$ difference set in Z_7.

Example 2.3.3. *Let $G = Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$. If $D = \{1, 2, 5\}$ then from Table 2.2, it is clear that D is not a $(7, 3, 1)$ difference set in Z_7.*

2.4 One-Factors and One-Factorizations

Before introducing *One-factors* and *one-factorizations*, we will present the following definition.

Definition 2.4.1. *A factor of a graph consists of an isolated vertex together with a set of disjoint pairs (edges) that exactly cover the remaining vertices. A factorization of a graph G is a set of factors of G which are pairwise edge-disjoint (that is, no two have a common edge) and whose union is all of G.*

Since G is a factor of itself, $\{G\}$ is a factorization of G so that every graph has a factorization. However, it is more interesting to consider factorizations in which the factors

Table 2.1: Differences

-	1	2	4
1	0	6	4
2	1	0	5
4	3	2	0

Table 2.2: Differences

-	1	2	5
1	0	6	3
2	1	0	4
5	4	3	0

satisfy certain conditions.

Definition 2.4.2. *For a given graph G, a one-factor is a set of edges in which every vertex of G appears exactly once.*

Definition 2.4.3. *A one-factorization of G is a partition of the edge set into edge-disjoint one-factors. In other words, a one-factorization of a graph is a decomposition of its edge set into one-factors.*

One-factors and *one-factorizations* of the complete graph of even order $2n$, written as K_{2n}, have been studied by several authors in [6, 11]. A good discussion that gives a general background on *one-factorizations* of the complete graphs can be found in [5, 6, 9, 11, 12]. In design theory, one-factorization of K_{2n} has been used for constructing a Steiner triple system of order v (written as STS (v)) [5, 6].

Lemma 2.4.4. *If the graph $G = (V, E)$, has one-factor, then $|V|$ is even.*

Proof. A one-factor consists of edges with no common vertices. So it contains an even number of vertices. Clearly $|V|$ is even, because the one-factor spans G. □

2.5 Near-One-Factors and Near-One-Factorizations

A *near-one-factorization* is the nearest thing to one-factor of a set on $n - 1$ edges which cover all but one vertex. A *near-one-factor* consists of one vertex (the focus) and a set of disjoint edges that contain every other vertex, while a *near-one-factorization* is a set of edge-disjoint near-one-factors that together contain all the edges.

Complete graphs of odd order, denoted by K_{2n-1}, have a *near-one-factorization*. A near-one-factor in K_{2n-1} is a set of $n-1$ edges which covers all but one vertex. A *near-one-factorization* is a set of near-one-factor that covers every edge precisely once. If the *near-one-factor* is written as

$$N = x \qquad ab \quad cd \quad \ldots \quad yz$$

then it is convenient to refer to

$$xab \quad xcd \quad \ldots \quad xyz$$

the triples with N.

Example 2.5.1. *Consider the complete graph K_5 of 5 vertices, Figure 2.2. We have 5 near-one-factors listed in Table 2.1 and denoted by F_i where $1 \leq i \leq 5$.*

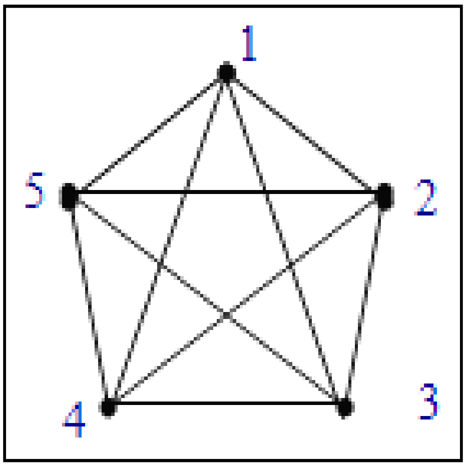

Figure 2.2: K_5

Table 2.3: Near-one-factors $K(5)$

F_1	1	25	34
F_2	2	13	45
F_3	3	15	24
F_4	4	12	35
F_5	5	14	23
	C_1	C_2	C_3

If we append C_1 with C_2 and C_1 with C_3, then we construct the 10 distinct triples $125, 134, 213, 245, 315, 324, 412, 435, 514$, and 523. These are the combination $\binom{5}{3} = 10$ distinct triples. This way of constructing distinct triples is an example of a generalization of *near-one-factor* on v objects. This leads to a special type of factorization called *compatible factorization* that will be introduced in the next section and discussed in details in chapter 3.

2.6 Towards Developing Compatible Factorizations

It is apparent from Table 2.1 and Figure 2.2 that each element in K_5 occurs as a focus exactly once and each associated triples contains no repetition. In order to discuss this type of constructions, it is worth to define such a set of near-one-factors to be a compatible factorization on v elements (objects, treatments), denoted by $CF(v)$.

Example 2.6.1. *Let* $v = 5$ *that is a set of* 5 *points labeled* $1, 2, 3, 4, 5$. *The unique* $CF(5)$ *is constructed in Table 2.4 by iterating, in each row, a pair of vertices in union with an isolated vertex, forming a triple not used in the previous row. Since* $v = 5$, *we have 5 rows and 3 columns with 1 column that has an isolated vertex as illustrated in Table 2.4.*

Table 2.4: A unique $CF(5)$

1	25	34
2	13	45
3	15	24
4	12	35
5	14	23
C_1	C_2	C_3

The associated triples, obtained by appendingC_1 with C_2 and C_1 with C_3, are $\{125\}, \{134\}, \{231\}$, $\{245\}, \{342\}, \{351\}, \{453\}, \{412\}, \{514\}$, and $\{523\}$.

It could be observed from Examples 2.6.1, that the first row is basic to construct the design. Once the triples in the first row have been found, it is easy to list all the distinct triples by addition modular 5. The first row will be called the starter, and will be discussed in details in the following chapters. Moreover, the definition of *compatible factorization* and other related results will be discussed in chapter 3.

Chapter 3

Compatible Factorization

In Chapter 2, a case where $v = 5$ was discussed explaining the concept of compatible factorization of order v. The current chapter presents *compatible factorization* on v objects, $CF(v)$, which is basic for developing triad design that will be presented in Chapter 4 and 5. Then we will explain the definitions, examples concerned with the cases of $v = 7$ and $v = 13$ and some related results of compatible factorization on v objects. In addition, the starter, the first row in the design, will be considered as a basic tool to develop the compatible factorization by adding modular v. We will also give an introduction to the triad design on v objects and its relation to compatible factorization.

3.1 Definitions and related results

Definition 3.1.1. *A compatible-factorization of a graph of order v, denoted by $CF(v)$, is a $v \times \frac{(v-1)}{2}$ array that satisfies the following conditions:*

1) The entries in row i form a near-one-factor with focus i.

2) The triples associated with the rows contain no repetitions.

From this definition, an obvious necessary condition for the existence of $CF(v)$ is that v must be odd greater than three.

Lemma 3.1.2. *There exists a compatible factorization for every odd order $v > 3$.*

Proof. Suppose $v = 2t + 1 > 3$. Therefore, the near-one-factor forms the patterned starter, with i-th factor is given as: $i\ (i + 1)(i - 1)\ (i + 2)(i - 2)\ \ldots (i + n)(i - n) \bmod v$ and is a compatible factorization. □

By Lemma 3.1.2, the condition for the first case for $CF(v)$ to exist is that v must be of order five as shown and discussed in Example 2.6.1. Moreover, $CF(5)$ is unique up to isomorphism. Complete searches show that there are 62,800 $CF(7)$ of which only 231 can be classified up to isomorphism [8]. The number for $CF(v)$ gets arbitrarily large for other orders.

Example 3.1.3. *Let $v = 7$, that is a set of 7 points labeled $1, 2, 3, 4, 5, 6, 7$. Here we have 7 rows and 4 columns with 1 column that has an isolated vertex as illustrated in Table 3.1.*

Table 3.1: $CF(7)$

1	27	36	45
2	31	47	56
3	42	51	67
4	53	62	71
5	64	73	12
6	75	14	23
7	16	25	34
C_1	C_2	C_3	C_4

In order to form triples, append C_1 with C_2, C_1 withC_3 and C_1 with C_4. Thus 3 triples in each row are produced. Hence 21 distinct triples in $CF(7)$ are generated. Note that the triples are unordered (that is, order is not important). For example {453}, {534} are considered the same triple. It is worth mentioning that the first row, called *starter*, is basic in the design. Once the *starter* is constructed, listing all distinct triples can be easily generated by adding modular 7 to the starter's elements. The next section deals with the starter.

3.2 Starter of compatible factorization

The aim of this section is to define the starter of $CF(v)$ which will be illustrated by some examples.

Definition 3.2.1. *The starter of compatible factorization on v objects, denoted by $SCF(v)$, is the set of triples that generates all the triples in $CF(v)$ by adding modular v .*

Example 3.2.2. *The first row in Example 3.1.3 is the starter. That is*

$$SCF(7) = \{127; 136; 145\}.$$

Each triple in $SCF(v)$ consists of three elements (numbers, objects), the first, the second and the third element.

Definition 3.2.3. *The i-th element of $SCF(v)$ is the i-th number in each triple, denoted by $S_rCF(v)$, for $1 \leq r \leq 3$.*

Example 3.2.4. *From Example 3.1.3, $SCF(7) = \{127; 136; 145\}$. Therefore*

$$\begin{array}{ll} S_1CF(7) & 1,1,1 \\ S_2CF(7) & 2,3,4. \\ S_3CF(7) & 7,6,5. \end{array}$$

3.3 Enumeration of CF(13)

Enumerating compatible factorization on v objects, $CF(13)$, depends on the construction of its starter, $SCF(13)$. It can be concluded from the pattern of $S_rCF(7)$, for $1 \leq r \leq 3$ in Example 3.2.2, that the patterns of $SCF(13)$ is as follows.

$$S_1CF(13): \quad 1, 1, 1, 1, 1, 1.$$
$$S_2CF(13): \quad 2, 3, 4, 5, 6, 7.$$
$$S_3CF(13): \quad 13, 12, 11, 10, 9, 8.$$

Hence, $SCF(13)$ can be constructed as explained in the above discussion or by establishing the following Table.

Table 3.2: $SCF(13)$

1	2	13	3	12	4	11	5	10	6	9	7	8
i	$i+1$	$i-1$	$i+2$	$i-2$	$i+3$	$i-3$	$i+4$	$i-4$	$i+5$	$i-5$	$i+6$	$i-6$

Hence $SCF(13) = \{1\ 2\ 13, 1\ 3\ 12, 1\ 4\ 11, 1\ 5\ 10, 1\ 6\ 9, 1\ 7\ 8\}$.

Using $SCF(13)$ and adding modular 13 to enumerate $CF(13)$ as shown in Table 3.3.

Table 3.3: $CF(13)$

1	2	13	3	12	4	11	5	10	6	9	7	8
2	3	1	4	13	5	12	6	11	7	10	8	9
3	4	2	5	1	6	13	7	12	8	11	9	10
4	5	3	6	2	7	1	8	13	9	12	10	11
5	6	4	7	3	8	2	9	1	10	13	11	12
6	7	5	8	4	9	3	10	2	11	1	12	13
7	8	6	9	5	10	4	11	3	12	2	13	1
8	9	7	10	6	11	5	12	4	13	3	1	2
9	10	8	11	7	12	6	13	5	1	4	2	3
10	11	9	12	8	13	7	1	6	2	5	3	4
11	12	10	13	9	1	8	2	7	3	6	4	5
12	13	11	1	10	2	9	3	8	4	7	5	6
13	1	12	2	11	3	10	4	9	5	8	6	7

The next section discusses the construction of starter of *compatible factorization* on v objects, $SCF(v)$, for all $v = 2m + 1$ and $m \geq 2$. This construction depends on the patterns of $S_rCF(v)$, for $1 \leq r \leq 3$.

3.4 Construction of SCF(v)-the general case

From the previous section, it can be deduced that the construction of $SCF(v)$, where $v = 2m + 1$ and $m \geq 2$ is as follows: Let k be the number of triples in $SCF(v)$. It is easy to observe from definition of $CF(v)$, that $1 \leq k \leq \frac{v-1}{2} = m$. Furthermore,

(i) $S_1CF(v)$: is always 1.

(ii) $S_2CF(v)$: is $1 + k$.

(iii) $S_3CF(v)$: $(2m + 2) - k$.

The above discussion is summarized in Table 3.4.

Table 3.4: Construction of $SCF(v)$

$v = 2m+1$	$SCF(v)$	$S_1CF(v)$	$S_2CF(v)$	$S_3CF(v)$
5	$\{125, 134\}$	1	2, 3	5, 4
7	$\{127, 136, 145\}$	1	2, 3, 4	7, 6, 5
9	$\{129, 138, 147, 156\}$	1	2, 3, 4, 5	9, 8, 7, 6
11	$\{1\ 2\ 11, 1\ 3\ 10,$ $1\ 4\ 9, 1\ 5\ 8, 1\ 6\ 7\}$	1	2, 3, 4, 5, 6	11, 10, 9, 8, 7
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2m+1$	$\{1\ 2\ (2m+1), \ldots,$ $1\ (m+1)\ (m+2)\}$	1	$1+k$ $1 \leq k \leq m$	$(2m+2)-k$ $1 \leq k \leq m$

3.5 Towards developing triad design

Compatible factorization on v objects, $CF(v,)$ gives $v \times \frac{v-1}{2}$ distinct triples. But our aim is to construct triad design on v objects, denoted by $TD(v)$, which consists of

$$\binom{v}{3} = \frac{v(v-1)(v-2)}{6} = v \times \frac{v-1}{2} \times \frac{v-2}{3}$$

distinct triples. So we need to append additional columns to $CF(v)$ construction to obtain all distinct triples of $\binom{v}{3}$. These additional columns can be categorized as the completion of $CF(v)$ to give us all the distinct triples in $TD(v)$. Denote $\overline{CF(v)}$ to be the completion of $CF(v)$. Therefore, $TD(v) = CF(v) \cup \overline{CF(v)}$.

For example if $v = 5$, then $TD(5) = CF(5) \cup \overline{CF(5)}$. Clearly, $TD(5)$ consists of $\binom{5}{3}$=10 distinct triples and referring to Example 2.6.1, the same number of distinct triples in $CF(5)$ was constructed. Hence, for the case of $v = 5, TD(5) = CF(5)$.

On the other hand, If $v = 7$, then $TD(7)$ consists of $\binom{7}{3} = 35$ distinct triples and referring to Example 3.1.3, the number of distinct triples in $CF(7)$ is equal to 21. Hence, for the case of $v = 7$, $\overline{CF(7)}$ must consist of two columns of 14 distinct triples. Therefore, $TD(7) = CF(7) \cup \overline{CF(v)}$.

The next chapter contains the main results of our work. The definition of triad design on v objects, the existence theorem of triad design and some other related results will be discussed in details. Moreover, the starter of triad design on v objects, denoted by $STD(v)$, will be introduced and analyzed in terms of the patterns of the triples and then employed to develop $TD(13)$ and the general case $TD(v)$ as well.

Chapter 4

Triad Design for 6n+1

The main results of the research are presented and explained in Chapter 4 and 5. These results concern about developing algorithm for triad design on v objects $TD(v)$, where $v = 6n + 1$ and $v = 6n + 5$. In Chapter 3, we introduced the concept of $TD(v)$. In this chapter, the definitions, the theorems of existence of $TD(v)$ and other related results will be presented. Additionally, an algorithm for developing $TD(v)$, for the cases $v = 7$ and 13, will be discussed. Moreover, a new method for developing $TD(13)$ was derived using the starter, whose construction depends on the analysis and the structure of triples in $TD(13)$. This method is also used to develop $TD(v)$ for the general cases $v = 6n + 1$ and $v = 6n + 5$. The core of this research is the starter of $TD(v)$) that is introduced and discussed in details in the following sections.

4.1 Definitions and related results

Definition 4.1.1. *A triad design on v objects, denoted by $TD(v)$, is a way of arranging the distinct triples of $\binom{v}{3}$ into v rows such that:*

(i) Row i contains $\frac{v-1}{2}$ *triples, among which object i meets every other objects precisely once, and contains also other distinct triples;*

(ii) Each triple occurs exactly once in the design;

(iii) No two elements (entries) occur together in two or more triples in any row.

Note that condition (i) in Definition 4.1.1, means that row i contains the triples in row i of $CF(v)$ and the triples in row i of $\overline{CF(v)}$. Moreover, $TD(v) = CF(v) \cup \overline{CF(v)}$.

Lemma 4.1.2. *If* $v = 6n + 1$, *Then the number of triples in* $\overline{CF(v)}$ *denoted by* $|\overline{CF(v)}|$ *is equal to* $2n(6n + 1)(3n - 2)$.

Proof. The number of triples in $TD(v)$, denoted by $|TD(v)|$ is equal to
$|TD(v)| = \binom{v}{3} = \binom{6n+1}{3} = \frac{(6n+1)(6n)(6n-1)}{6} = n(6n + 1)(6n - 1)$.
But $|CF(v)| = v \times \frac{v-1}{2} = (6n + 1) \times \frac{(6n+1)-1}{2} = 3n(6n + 1)$.
Therefore,
$|\overline{CF(v)}| = |TD(v)| - |CF(v)| = n(6n + 1)(6n - 1) - 3n(6n + 1) = 2n(6n + 1)(3n - 2)$. □

Example 4.1.3. *(Construction of TD(7))*
Based on Example 3.13, we have 21 *triples. Since* $TD(7)$ *have* 35 *triple, we need to have* 14 *more triples by Lemma 4.12. Table 4.1 list* $\overline{CF(v)}$. *We observed that that we need two more column.*

Table 4.1: $TD(7)$

		$CF(7)$		$\overline{CF(v)}$	
1	2 7	3 6	4 5	2 3 5	7 6 4
2	3 1	4 7	5 6	3 4 6	1 7 5
3	4 2	5 1	6 7	4 5 7	2 1 6
4	5 3	6 2	7 1	5 6 1	3 2 7
5	6 4	7 3	1 2	6 7 2	4 3 1
6	7 5	1 4	2 3	7 1 3	5 4 2
7	1 6	2 5	3 4	1 2 4	6 5 3

Note that the difference sets for each triple in the first row, the starter, are listed in table 4.2 (using the fact that $\pm 6 = \pm 1, \pm 5 = \pm 2$ and $\pm 4 = 3$).

Moreover, it can be seen from Table 4.2 that the differences are $\pm 1, \pm 2$ and ± 3, where each difference occurs 5 times. We write $\lambda(STD(7)) = 5$, where $STD(7)$ denotes the starter triad design.

Table 4.2: Difference sets for the starter of $TD(7)$

Triples	Difference set
1 2 7	$\pm 1, \pm 2, \pm 1$
1 3 6	$\pm 2, \pm 3, \pm 2$
1 4 5	$\pm 3, \pm 1, \pm 3$
2 3 5	$\pm 1, \pm 2, \pm 3$
7 6 4	$\pm 1, \pm 2, \pm 3$

It is natural to ask about the possible values of v in order for triad design on v objects, $TD(v)$, to exist. From Lemma 3.1.2, we found that $CF(v)$ exists for odd order $v > 3$. However, this is not sufficient for $TD(v)$ to exist. For example, suppose $TD(9)$ exists. Obviously, it must contain $\binom{9}{3} = 84$ distinct triples. However, $CF(9)$ consists of 36 triples. Thus 48 triples left. Since 9 does not divide 48, $\overline{CF(9)}$ cannot be constructed by the definition of *triad design*. Consequently, $TD(9)$ does not exist. The following theorem, due to [8], gives the possible values of v in order for $TD(v)$ to exist.

Theorem 4.1.4. *(Existence of $TD(v)$) Triad designs on v objects, $TD(v)$, exist iff* $v \equiv 1$ *or* 5*(mod* 6*).*

Proof. Suppose there is a triad design on v objects. Then by definition of $TD(v)$, v must divide $\binom{v}{3} = \frac{(v(v-1)(v-2))}{6}$. Therefore 6 divide $(v-1)(v-2)$. This implies that $v \equiv 1$ or 2 (mod 3). Since v must be odd by Lemma 3.1.2 in order for $CF(v)$ to exist, so $v \equiv 1$ or 5 (mod 6). □

4.2 Triad design v = 6n+1

In this section we present an algorithm to develop $TD(6n+1)$. We start by developing the starter for case $v = 13$. Then we analyze the starter to construct for general case $TD(6n+1)$.

4.2.1 Starter of triad design

It could be observed from Example 4.1.3 that generating $\overline{CF(7)}$, the completion of $CF(7)$, was made by brute force method. This method is not suitable when v is large. For example, if $v = 13$, $\overline{CF(13)}$ consist of 16 columns. Therefore, developing new methods to produce $\overline{CF(v)}$, especially the first row, the starter, is the main objective of this research. Once the triples in the first row are constructed, listing all distinct triples can be done by adding modular v.

Definition 4.2.1. *The starter of triad design on v objects, denoted by $STD(v)$, is the set of triples on v that generates all the triples in $TD(v)$ by addition modular v.*

Example 4.2.2. $STD(7) = \{127; 136; 145; 235; 764\}$ *by Example 4.1.3.*

Remark 4.2.3. *It is important to mention that* $STD(7) = SCF(7) \cup S\overline{CF(v)} = \{127; 136; 145\} \cup \{235; 764\}$.

Definition 4.2.4. *The r-th elements of $STD(v)$ are the r-th number in each triple, denoted by $S_rTD(v)$, for $1 \le r \le 3$.*

Example 4.2.5. *From above $STD(7) = \{127; 136; 145; 235; 764\}$. Therefore*

$$S_1TD(7): \quad 1, 1, 1, 2, 7.$$
$$S_2TD(7): \quad 2, 3, 4, 3, 6.$$
$$S_3TD(7): \quad 7, 6, 5, 5, 4.$$

The following two lemmas provide the number of triples in $STD(v)$ as well as in $\overline{SCF(v)}$, denoted by $|STD(v)|$ and $|\overline{SCF(v)}|$ respectively.

Lemma 4.2.6. *If $v = 6n + 1$, then $|STD(v)|$ is equal to $n(6n - 1)$.*

Proof. From the proof of Lemma 4.1.1, the number of triples of $TD(v)$ is equal to $|TD(v)| = n(6n+1)(6n-1)$. By definition of $TD(v)$, the number of rows of $TD(v)$ is equal to $v = 6n+1$. Hence $|STD(v)| = n(6n - 1)$. □

Lemma 4.2.7. *If $v = 6n + 1$, then $|\overline{SCF(v)}|$ is even and equal to $2n(3n - 2)$.*

Proof. By Lemma 4.1.1, $|\overline{CF(v)}|$ is equal to $2n(6n + 1)(3n - 2)$. Since the number of rows equals $v = 6n + 1$, then $|\overline{SCF(v)}| = 2n(3n - 2)$. □

In this section, the triples in $STD(7)$ are analyzed in order to construct formulas to generate them and to develop $TD(7)$. This method will be used for $STD(13)$ and for the general case $STD(v)$ as well.

a. **Interval constructions of STD(7)**

In this section, formulas for $S_rTD(7)$, where $1 \leq r \leq 3$ of $STD(7)$ are constructed. From remark 4.2.1, $STD(7) = SCF(7) \cup \overline{CF(v)} = \{127; 136; 145\} \cup \{235; 764\}$. Let T_k be the k-th triple in $STD(7)$. Obviously,$1 \leq k \leq 5$ by Lemma 4.1.2. Furthermore, $STD(7) = \{T_1, \ldots, T_5\}$. Let $[S_rTD(7)]_k$ be the k-th element of $S_rTD(7)$ for $1 \leq r \leq 3$. Therefore, the formulas for the first, second and third elements of $STD(7)$ are as follows.

$$[S_1TD(7)]_k = \begin{cases} 1 & \text{if } 1 \leq k \leq 3 \\ 2 & \text{if } k = 4 \\ 7 & \text{if } k = 5 \end{cases}$$

$$[S_2TD(7)]_k = \begin{cases} 1 + k & \text{if } 1 \leq k \leq 3 \\ 3 & \text{if } k = 4 \\ 6 & \text{if } k = 5 \end{cases}$$

$$[S_3TD(7)]_k = \begin{cases} 1 - k & \text{if } 1 \leq k \leq 3 \\ 5 & \text{if } k = 4 \\ 4 & \text{if } k = 5 \end{cases}$$

The three elements of $STD(7)$ are summarized in Table 4.2.

Table 4.3: Three elements of $STD(7)$

k	1	2	3	4	5
$S_1TD(7)$	1	1	1	2	7
$S_2TD(7)$	2	3	4	3	6
$S_3TD(7)$	7	6	5	5	4

Using $STD(7)$ and addition modular 7 to enumerate $TD(7)$ as shown in Tables 4.3

and 4.4.

Table 4.4: $TD(7)$

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

b . **Algorithms for TD(13)**

In this section, addition modular 13 is used to produce the algorithm for $STD(13)$. Obviously, $STD(13) = SCF(13) \cup \overline{SCF(13)}$. By Lemma 4.2.1, with $n = 2$, $|STD(13)| = 22$, and by Lemma 4.2.2, $|\overline{SCF(13)}| = 16$. The algorithm of $TD(13)$ is as follows:

Step 1. Generate $SCF(13)$. This is done in Chapter 3, Section 3.3. That is

$$SCF(13) = \{1213, 1312, 1411, 1510, 169, 178\}.$$

Note that the difference sets for each triple in $SCF(13)$ are listed in table 4.5 using the fact that

$\pm 12 = \pm 1, \pm 11 = \pm 2, \pm 10 = \pm 3, \pm 9 = \pm 4, \pm 8 = \pm 5$ and $\pm 7 = \pm 6$.

Table 4.5: Difference sets for $SCF(13)$

Triples	1 2 13	1 3 12	1 4 11	1 5 10	1 6 9	1 7 8
Difference set	±1, ±2, ± 1	± 2, ± 4, ± 2	± 3, ± 6, ± 3	± 4, ± 5, ±4	± 5, ± 3, ± 5	± 6, ± 1, ± 6

From Table 4.5, each difference occurs 3 times in $SCF(13)$.

Step 2. Generate $\overline{SCF(13)}$ by using difference set method and addition modular 13. This is done in Table 4.6.

Table 4.6: Difference sets for $SCF(13)$

13	12	4	$(i-1)(i-2)(i+3)$	± 1, ± 5,± 4	2	3	11	$(i+1)(i+2)(i-3)$	± 1, ± 5,± 4
13	11	5	$(i-1)(i-3)(i+4)$	± 2, ± 6,± 5	2	4	10	$(i+1)(i+3)(i-4)$	± 2, ± 6,±5
13	10	6	$(i-1)(i-4)(i+5)$	±3, ± 4,± 6	2	5	9	$(i+1)(i+4)(i-5)$	±3, ± 4,± 6
13	9	7	$(i-1)(i-5)(i+6)$	± 4, ± 2,± 6	2	6	8	$(i+1)(i+5)(i-6)$	± 4 ,± 2,± 6
12	11	6	$(i-2)(i-3)(i+5)$	± 1, ± 5,±6	3	4	9	$(i+2)(i+3)(i-5)$	± 1, ± 5,±6
12	10	7	$(i-2))(i-4)(i+6)$	± 2, ± 3,± 5	3	5	8	$(i+2)(i+4)(i-6)$	± 2, ±3,± 5
11	10	7	$(i-3))(i-4)(i+6)$	± 1, ± 3,± 4	4	5	8	$(i+3)(i+4)(i-6)$	± 1, ± 3,± 4
10	9	7	$(i-4)(i-5)(i+6)$	± 1, ± 2± 3	5	6	8	$(i+4)(i+5)(i-6)$	± 1,± 2,± 3

Hence $\overline{SCF(13)} = \{13124; 2311; 13115; 2410; 13106; 259; 1397; 268;$

$12116; 349; 12107; 358; 11107; 458; 1097; 568\}$.

From Table 4.6, each difference occurs 8 times in Since $STD(13) = SCF(13) \cup \overline{SCF(13)}$, we have $\lambda(STD(13)) = 11$.

Step 3. Starters from Step 1 and Step 2 will be the starter $STD(13)$. That is

$$STD(13) = SCF(13) \cup \overline{SCF(13)}$$

$$= \{1\ 2\ 13; 1\ 3\ 12; 1\ 4\ 11, 1\ 5\ 10; 1\ 6\ 9; 1\ 7\ 8\} \cup \{13\ 12\ 4; 2\ 3\ 11; 13\ 11\ 5; 2\ 4\ 10; 13\ 10\ 6; 2\ 5\ 9; 13\ 9\ 7; 2\ 6\ 8; 12\ 11\ 6; 3\ 4\ 9; 12\ 10\ 7; 3\ 5\ 8; 11\ 10\ 7; 4\ 5\ 8; 10\ 9\ 7; 5\ 6\ 8\}.$$

Step 4. Using starter from step 3 and addition modular 13 to enumerate $TD(13)$ as shown in Table 4.7.

Table 4.7: $TD(13)$

1			2			3			4			5			6		
1	2	13	1	3	12	1	4	11	1	5	10	1	6	9	1	7	8
2	3	1	2	4	13	2	5	12	2	6	11	2	7	10	2	8	9
3	4	2	3	5	1	3	6	13	3	7	12	3	8	11	3	9	10
4	5	3	4	6	2	4	7	1	4	8	13	4	9	12	4	10	11
5	6	4	5	7	3	5	8	2	5	9	1	5	10	13	5	11	12
6	7	5	6	8	4	6	9	3	6	10	2	6	11	1	6	12	13
7	8	6	7	9	5	7	10	4	7	11	3	7	12	2	7	13	1
8	9	7	8	10	6	8	11	5	8	12	4	8	13	3	8	1	2
9	10	8	9	11	7	9	12	6	9	13	5	9	1	4	9	2	3
10	11	9	10	12	8	10	13	7	10	1	6	10	2	5	10	3	4
11	12	10	11	13	9	11	1	8	11	2	7	11	3	6	11	4	5
12	13	11	12	1	10	12	2	9	12	3	8	12	4	7	12	5	6
13	1	12	13	2	11	13	3	10	13	4	9	13	5	8	13	6	7

7			8			9			10			11			12		
13	12	4	2	3	11	13	11	5	2	4	10	13	10	6	2	5	9
1	13	5	3	4	12	1	12	6	3	5	11	1	11	7	3	6	10
2	1	6	4	5	13	2	13	7	4	6	12	2	12	8	4	7	11
3	2	7	5	6	1	3	1	8	5	7	13	3	13	9	5	8	12
4	3	8	6	7	2	4	2	9	6	8	1	4	1	10	6	9	13
5	4	9	7	8	3	5	3	10	7	9	2	5	2	11	7	10	1
6	5	10	8	9	4	6	4	11	8	10	3	6	3	12	8	11	2
7	6	11	9	10	5	7	5	12	9	11	4	7	4	13	9	12	3
8	7	12	10	11	6	8	6	13	10	12	5	8	5	1	10	13	4
9	8	13	11	12	7	9	7	1	11	13	6	9	6	2	11	1	5
10	9	1	12	13	8	10	8	2	12	1	7	10	7	3	12	2	6
11	10	2	13	1	9	11	9	3	13	2	8	11	8	4	13	3	7
12	11	3	1	2	10	12	10	4	1	3	9	12	9	5	1	4	8

13			14			15			16			17			18		
13	9	7	2	6	8	12	11	6	3	4	9	12	10	7	3	5	8
1	10	8	3	7	9	13	12	7	4	5	10	13	11	8	4	6	9
2	11	9	4	8	10	1	13	8	5	6	11	1	12	9	5	7	10
3	12	10	5	9	11	2	1	9	6	7	12	2	13	10	6	8	11
4	13	11	6	10	12	3	2	10	7	8	13	3	1	11	7	9	12
5	1	12	7	11	13	4	3	11	8	9	1	4	2	12	8	10	13
6	2	13	8	12	1	5	4	12	9	10	2	5	3	13	9	11	1
7	3	1	9	13	2	6	5	13	10	11	3	6	4	1	10	12	2
8	4	2	10	1	3	7	6	1	11	12	4	7	5	2	11	13	3
9	5	3	11	2	4	8	7	2	12	13	5	8	6	3	12	1	4
10	6	4	12	3	5	9	8	3	13	1	6	9	7	4	13	2	5
11	7	5	13	4	6	10	9	4	1	2	7	10	8	5	1	3	6
12	8	6	1	5	7	11	10	5	2	3	8	11	9	6	2	4	7

19			20			21			22		
11	10	7	4	5	8	10	9	7	5	6	8
12	11	8	5	6	9	11	10	8	6	7	9
13	12	9	6	7	10	12	11	9	7	8	10
1	13	10	7	8	11	13	12	10	8	9	11
2	1	11	8	9	12	1	13	11	9	10	12
3	2	12	9	10	13	2	1	12	10	11	13
4	3	13	10	11	1	3	2	13	11	12	1
5	4	1	11	12	2	4	3	1	12	13	2
6	5	2	12	13	3	5	4	2	13	1	3
7	6	3	13	1	4	6	5	3	1	2	4
8	7	4	1	2	5	7	6	4	2	3	5
9	8	5	2	3	6	8	7	5	3	4	6
10	9	6	3	4	7	9	8	6	4	5	7

It can be observed that the method of constructing $\overline{SCF(13)}$ is not suitable when v is large. Therefore, developing a new method to produce $\overline{SCF(13)}$ towards developing $\overline{SCF(v)}$, is the main objective of this chapter. Hence, the objective of the following section is to develop a new method to generate $STD(13)$. This new method depends on analyzing the triples in $STD(13)$ using interval techniques of the number of triples.

c. Interval constructions of STD(13)

From Step 3 of the previous section,

$$\begin{aligned} STD(13) &= SCF(13) \cup \overline{SCF(13)} \\ &= \{1\ 2\ 13; 1\ 3\ 12; 1\ 4\ 11, 1\ 5\ 10; 1\ 6\ 9; 1\ 7\ 8\} \cup \{13\ 12\ 4; 2\ 3\ 11; \\ &\quad 13\ 11\ 5; 2\ 4\ 10; 13\ 10\ 6; 2\ 5\ 9; 13\ 9\ 7; 2\ 6\ 8; 12\ 11\ 6; 3\ 4\ 9; \\ &\quad 12\ 10\ 7; 3\ 5\ 8; 11\ 10\ 7; 4\ 5\ 8; 10\ 9\ 7; 5\ 6\ 8\}. \end{aligned}$$

Let T_k be the k-th triple in $STD(13)$. Obviously, $1 \leq k \leq 22$ by Lemma 4.2.1. Furthermore, $STD(13) = T_1, , T_{22}$. Let $[S_rTD(13)]_k$ be the k-th element in $S_rTD(13)$

for $1 \leq r \leq 3$. We are able to summarize $STD(13)$ in terms $S_rTD(13)$ as shown in Table 4.8.

Table 4.8: $S_rTD(13)$

k	$S_1TD(13)$	$S_2TD(13)$	$S_3TD(13)$
1	1	2	13
2	1	3	12
3	1	4	11
4	1	5	10
5	1	6	9
6	1	7	8
7	13	12	4
8	2	3	11
9	13	11	5
10	2	4	10
11	13	10	6
12	2	5	9
13	13	9	7
14	2	6	8
15	12	11	6
16	3	4	9
17	12	10	7
18	3	5	8
19	11	10	7
20	4	5	8
21	10	9	7
22	5	6	8

It can be observed from Table 4.8 that k, the number of triples in $STD(13)$, can be divided into 4 intervals. These intervals and the corresponding elements in $S_1TD(13)$ are illustrated in Table 4.9.

Table 4.9: Intervals of k and the corresponding elements in $S_1TD(13)$

No. of Intervals	Intervals of k	Corresponding elements in $S_1TD(13)$
1	$1 \leq k \leq 6$	1, 1, 1, 1, 1, 1
2	$7 \leq k \leq 14$	13, 2, 13, 2, 13, 2, 13, 2
3	$15 \leq k \leq 18$	12, 3, 12, 3
4	$19 \leq k \leq 22$	11, 4, 10, 5

It can be seen from Table 4.9, that the corresponding elements in $S_1TD(13)$ in the last interval of k need a special formula to produce them. Let f denote the first number of the interval. The formula $y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1]$ can produce the corresponding elements in $S_1TD(13)$ in the indicated interval. For example, if $k = 20$, then $f = 19$ because 20 in the last interval $19 \leq k \leq 22$. Hence,

$$y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1] = \frac{1}{2}[20 - 19 + mod(\frac{20+2}{2}) + 1] = 1.$$

Therefore, the corresponding elements in $S_1TD(13)$ for $k = 20$ is $3 + y = 3 + 1 = 4$, which are the same number as shown in Table 4.9 and 4.10. From Table 4.9 and 4.10, the construction of $S_1TD(13)$ is as follows.

$$[S_1TD(13)]_k = \begin{cases} 1 & \text{if } 1 \leq k \leq 6 \\ 2 & \text{if } 7 \leq k \leq 14, k \text{ is even} \\ 13 & \text{if } 7 \leq k \leq 14, k \text{ is odd} \\ 3 & \text{if } 15 \leq k \leq 18, k \text{ is even} \\ 12 & \text{if } 15 \leq k \leq 18, k \text{ is odd} \\ 3 + y & \text{if } 19 \leq k \leq 22, k \text{ is even} \\ 12 - y & \text{if } 19 \leq k \leq 22, k \text{ is odd} \end{cases}$$

Similarly, intervals of k and the corresponding elements in $S_2TD(13)$ are shown in

Table 4.10.

Table 4.10: Intervals of k and the corresponding elements in $S_2TD(13)$

No. of Intervals	Intervals of k	Corresponding elements in $S_2TD(13)$
1	$1 \leq k \leq 6$	2, 3, 4, 5, 6, 7
2	$7 \leq k \leq 14$	12, 3, 11, 4, 10, 5, 9, 6
3	$15 \leq k \leq 18$	11, 4, 10, 5
4	$19 \leq k \leq 22$	10, 5, 9, 6

Using the same formula $y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1]$ to produce the corresponding elements in $S_2TD(13)$ in all intervals of k except the first one. For example, if $k = 17$, then $f = 15$ because 17 in the third interval $15 \leq k \leq 18$. Hence,

$$y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1] = \frac{1}{2}[2 + 1 + 1] = 2.$$

Therefore, the corresponding elements in $S_2TD(13)$ for $k = 17$ are equal to $12 - y = 12 - 2 = 10$, which is the same number as shown in Tables 4.8 and 4.10.

Therefore, the construction of $S_2TD(13)$ is the following

$$[S_2TD(13)]_k = \begin{cases} 1 + k & \text{if } 1 \leq k \leq 6 \\ 2 + y & \text{if } 7 \leq k \leq 14, k \text{ is even} \\ 13 - y & \text{if } 7 \leq k \leq 14, k \text{ is odd} \\ 3 + y & \text{if } 15 \leq k \leq 18, k \text{ is even} \\ 12 - y & \text{if } 15 \leq k \leq 18, k \text{ is odd} \\ 4 + y & \text{if } 19 \leq k \leq 22, k \text{ is even} \\ 11 - y & \text{if } 19 \leq k \leq 22, k \text{ is odd} \end{cases}$$

where $y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1]$.

Finally, similar to the above discussion, intervals of k and the corresponding elements in $S_3TD(13)$ are shown in Tables 4.11.

Table 4.11: Intervals of k and the corresponding elements in $S_3TD(13)$

No. of Intervals	Intervals of k	Corresponding elements in $S_2TD(13)$
1	$1 \leq k \leq 6$	13, 12, 11, 10, 9, 8
2	$7 \leq k \leq 14$	4, 11, 5, 10, 6, 9, 7, 8
3	$15 \leq k \leq 18$	6, 9, 7, 8
4	$19 \leq k \leq 22$	7, 8, 7, 8

Using the same formula $y = \frac{1}{2}[k-f+mod(\frac{k+2}{2})+1]$ to generate the corresponding elements in all intervals of k except the first one. Hence from Tables 4.11, the construction of $S_3TD(13)$ is the following.

$$[S_3TD(13)]_k = \begin{cases} 14 - k & \text{if } 1 \leq k \leq 6 \\ 12 - y & \text{if } 7 \leq k \leq 14, k \text{ is even} \\ 3 + y & \text{if } 7 \leq k \leq 14, k \text{ is odd} \\ 11 - y & \text{if } 15 \leq k \leq 18, k \text{ is even} \\ 4 + y & \text{if } 15 \leq k \leq 18, k \text{ is odd} \\ 8 & \text{if } 19 \leq k \leq 22, k \text{ is even} \\ 7 & \text{if } 19 \leq k \leq 22, k \text{ is odd} \end{cases}$$

where $y = \frac{1}{2}[k - f + mod(\frac{k+2}{2}) + 1]$.
Using $STD(13)$ and addition modular 13 to enumerate $TD(13)$ as shown in Table 4.7.

The next section is about the general case the construction of $STD(v)$, where $v = 6n + 1$, which is the main objective of this research. It depend on analyzing the triples in $STD(v)$ using interval techniques of the number of triples.

4.3 Interval constructions of general cases STD(6n+1)

From the intervals construction of $STD(7), STD(13)$ in Section 4.2. The following table of intervals for the cases $v = 7, 13, 19, \ldots, 6n+1$ is established. Table 4.12 is a result of analyzing the patterns of the triples in the designs of the previous cases. Some theorems and results are deduced from the Intervals Table 4.12.

Table 4.12: Intervals of $STD(6n+1)$

Intervals of k	7 n=1	13 n=2	19 n=3	...	$6n+1$
	$1 \le k \le 3$	$1 \le k \le 6$	$1 \le k \le 9$	...	$1 \le k \le 3n$
		$7 \le k \le 14$	$10 \le k \le 23$	...	$3n+1 \le k \le 9n-4$
		$15 \le k \le 18$	$24 \le k \le 33$	...	$9n-3 \le k \le 15n-12$
		$19 \le k \le 22$	$34 \le k \le 39$	...	$15n-11 \le k \le 21n-24$
			$40 \le k \le 43$	...	$21n-23 \le k \le 27n-40$
			$44 \le k \le 47$	...	⋮
			$48 \le k \le 51$	...	⋮
				⋮	⋮
Total	1	4	7	...	$3n-2$

Theorem 4.3.1. *The number of intervals of $STD(6n+1)$ is equal $3n-2$.*

Proof. It is obvious from Table 4.12 of intervals of $STD(6n+1)$ that the pairs (n, number of intervals) are $(1,1), (2,4), (3,7), \ldots$.

This is a linear function of constant slope equal to 3. Therefore, the numbers of of intervals $= 3n-2$. □

The general rules of the intervals for the general case $v = 6n+1$ are:

1 - The first interval is $1 \leq k \leq 3n$ which is related to $SCF(v)$.

2 - Other intervals except the last one are explained as follows: Let i denotes the interval number starting from the second interval until the second to the last one. It can be observed from Theorem 4.3.1, that the values of i are $1 \leq i \leq 3n-4$ That is, $i = 1$ is the second interval, $i = 2$ is the third interval,..., $i = 3n-4$ is the interval second to last one.

The last numbers of each interval, denoted by L, are constructed as follows:

The intervals		The last number L
$3n+1 \leq k \leq 9n-4$	$\rightarrow$	$9n-4 = (6(1)+3)n - (2(1)^2 + 2(1))$
$9n-3 \leq k \leq 15n-12$	$\rightarrow$	$15n-12 = (6(2)+3)n - (2(2)^2 + 2(2))$
$15n-11 \leq k \leq 21n-24$	$\rightarrow$	$21n-24 = (6(3)+3)n - (2(3)^2 + 2(3))$
	$\vdots$	

$L = (6(i)+3)n - (2i^2 + 2i)$, where $1 \leq i \leq n$.

Now, if $n+1 \leq i \leq 3n-4$ then from the Intervals Table 4.12 the difference between the last numbers of two successive intervals is smaller compared to the case when $1 \leq i \leq n$. Consequently, a value denoted by a_t, defined below, must be added to obtain the desired last numbers of the intervals. Therefore, the last number of the general interval when $n+1 \leq i \leq 3n-4$ is equal to $L + a_t = (6(i)+3)n - (2i^2 + 2i) + a_t$.

If $i = n + t$, clearly $1 \leq t \leq 2n-4$. Define

$$a_t = \begin{cases} 0 & \text{if } t = 0 \\ 2 & \text{if } t = 1 \\ a_{t-2} + 6t - 4 & \text{if } t \geq 2 \end{cases}$$

Similarly, the first numbers of each interval, denoted by F, are constructed as follows:

In general, $F = [6(i-1)+3]n - [2(i-1)^2 + 2(i-1)] + 1$, where $1 \leq i \leq n$.

For the same reason above, if $n+1 \leq i \leq 3n-4$ then a value denoted by a_{t-1}, defined above, must be added to obtain the desired first numbers of the intervals. Therefore, the first number of the general interval when $n+1 \leq i \leq 3n-4$ is equal to

$$F + a_{t-1} + 1 = (6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + a_{t-1} + 1.$$

where $i = n + t, 1 \leq t \leq 2n-4$.

The above discussion and notations are summarized in the following generalization concerning the patterns of the intervals for the general case $v = 6n + 1$.

Interval Generalization Regarding to the above discussion and notations, the patterns of the intervals for the general case $v = 6n + 1$ are

1 - The first interval is $1 \leq k \leq 3n$.

2 - Other intervals except the last one are explained as follows

$F + 1 \leq k \leq L$ when $1 \leq i \leq n$, and

$F + a_{t-1} + 1 \leq k \leq L + a_t$ when $n+1 \leq i \leq 3n-4$, $i = n+t$, $1 \leq t \leq 2n-4$.

Also $a_0 = 0$, $a_1 = 2$, $a_2 = 8$, and if $t \geq 3$, then $a_t = a_{(t-2)} + (6t-4)$

The intervals		The first number F
$3n+1 \leq k \leq 9n-4$	$\rightarrow$	$3n+1 = [6(1-1)+3]n - [2(1-1)^2 + 2(1-1))] + 1$
$9n-3 \leq k \leq 15n-12$	$\rightarrow$	$9n-3 = [6(2-1)+3]n - [2(2-1)^2 + 2(2-1)] + 1$
$15n-11 \leq k \leq 21n-24$	$\rightarrow$	$15n-11 = [6(3-1)+3]n - [2(3-1)^2 + 2(3-1)] + 1$
	$\vdots$	

Example 4.3.2. *Let $n = 3$, that is $v = 19$, then by Theorem 4.3.1, the number of intervals is equal to 7. Clearly,$1 \leq i \leq 5$ and $1 \leq t \leq 2n-4$, that is $1 \leq t \leq 2$. Applying the Interval Generalization to obtain the intervals for the case $v = 19$.*

1 - The first interval is $1 \leq k \leq 3n$, that is $1 \leq k \leq 9$.

2 - If i $= 1$, then the second interval is

$$F + 1 \leq k \leq L$$

$$(6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n - (2i^2 + 2i)$$

$$(6(1-1)+3)3 - (2(1-1)^2 + 2(1-1)) + 1 \leq k \leq (6(1)+3)3 - (2(1)^2 + 2(1))$$

$$10 \leq k \leq 23$$

If i $= 2$, then the third interval is

$$F + 1 \leq k \leq L$$

$$(6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n - (2i^2 + 2i)$$

$$(6(2-1)+3)3 - (2(2-1)^2 + 2(2-1)) + 1 \leq k \leq (6(2)+3)3 - (2(2)^2 + 2(2))$$

$$24 \leq k \leq 33$$

If i $= 3$, then the forth interval is

$$F + 1 \leq k \leq L$$

$$(6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n - (2i^2 + 2i)$$

$$(6(3-1)+3)3 - (2(3-1)^2 + 2(3-1)) + 1 \leq k \leq (6(3)+3)3 - (2(3)^2 + 2(3))$$

$$34 \leq k \leq 39$$

3 - Now $i > n = 3$, so the second formula will be used.

If i $= 4$, *then the fifth interval is*

$$F + a_{t-1} + 1 \leq k \leq L + a_t$$

$$(6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + a_{t-1} + 1 \leq k \leq (6(i)+3)n - (2i^2 + 2i) + a_t$$

$$(6(4-1)+3)3 - (2(4-1)^2 + 2(4-1)) + a_{1-1} + 1 \leq k \leq (6(4)+3)3 - (2(4)^2 + 2(4)) + a_1$$

$$40 \leq k \leq 43$$

If i $= 5$, *then the sixth interval is*

$$F + a_{t-1} + 1 \leq k \leq L + a_t$$

$$(6(i-1)+3)n - (2(i-1)^2 + 2(i-1)) + a_{t-1} + 1 \leq k \leq (6(i)+3)n - (2i^2 + 2i) + a_t$$

$$(6(5-1)+3)3 - (2(5-1)^2 + 2(5-1)) + a_{2-1} + 1 \leq k \leq (6(5)+3)3 - (2(5)^2 + 2(5)) + a_2$$

$$44 \leq k \leq 47$$

We are now in a position to give the constructions of $S_rTD(v)$, for $1 \leq r \leq 3$. Let $v = 6n + 1$. Clearly, $STD(v) = SCF(v) \cup \overline{SCF(v)}$. Let T_k be the k-th triple in $STD(v)$. By Lemma 4.2.1, $1 \leq k \leq 3n$. Let $[S_rTD(v)]_k$ be the k-th element in $S_rTD(v)$ for $1 \leq r \leq 3$. If i denotes the interval number starting from the second interval until the second to last one, then by Theorem 4.3.1, $1 \leq i \leq 3n - 4$ Let t be a variable appears when $i > n$. If $i = n + t$, then clearly $1 \leq t \leq 2n - 4$. Define

$$a_t = \begin{cases} 0 & \text{if } t = 0 \\ 2 & \text{if } t = 1 \\ a_{t-2} + 6t - 4 & \text{if } t \geq 2 \end{cases}$$

We set in addition to the above notations, the formula $y = \frac{1}{2}[k - F + mod(\frac{k+n}{2}) + 1]$.

i) The construction of $S_1TD(6n+1)$ is as follows.

$$[S_1TD(v)]_k = \begin{cases} 1 & \text{if } 1 \le k \le 3n \\ i+1 & \text{if } F+1 \le k \le L, k \text{ is even} \\ 6n+2-i & \text{if } F+1 \le k \le L, k \text{ is odd} \\ i+1 & \text{if } F+a_{t-1}+1 \le k \le L+a_t, k \text{ is even} \\ 6n+2-i & \text{if } F+a_{t-1}+1 \le k \le L+a_t, k \text{ is odd} \\ & n+1 \le i \le 3n-4, 1 \le t \le 2n-4 \\ 3n-3+y & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is even} \\ 3n+6-y & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

ii) The construction of $S_2TD(6n+1)$

$$[S_2TD(v)]_k = \begin{cases} k+1 & \text{if } 1 \le k \le 3n \\ i+1+y & \text{if } F+1 \le k \le L, 1 \le i \le n, k \text{ is even} \\ 6n+2-i-y & \text{if } F+1 \le k \le L, 1 \le i \le n, k \text{ is odd} \\ i-1+y & \text{if } F+a_{t-1}+1 \le k \le L+a_t, i=n+t, k \text{ is even} \\ 6n+2-i-y & \text{if } F+a_{t-1}+1 \le k \le L+a_t, i=n+t, k \text{ is odd} \\ & n+1 \le i \le 3n-4, 1 \le t \le 2n-4 \\ 3n-2+y & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is even} \\ 3n+5-y & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

iii) The construction of $S_3TD(6n+1)$

$$[S_3TD(v)]_k = \begin{cases} 6n+2-k & \text{if } 1 \le k \le 3n \\ 6n+2-2i-y & \text{if } F+1 \le k \le L, 1 \le i \le n, k \text{ is even} \\ 2i+1+y & \text{if } F+1 \le k \le L, 1 \le i \le n, k \text{ is odd} \\ 6n+2-2i-y & \text{if } F+a_{t-1}+1 \le k \le L+a_t, i=n+t, k \text{ is even} \\ 2i+1+y & \text{if } F+a_{t-1}+1 \le k \le L+a_t, i=n+t, k \text{ is odd} \\ & n+1 \le i \le 3n-4, 1 \le t \le 2n-4 \\ 3n+2 & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is even} \\ 3n-1 & \text{if } L+a_t+1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

Chapter 5

Triad Design for 6n+5

In this chapter, we employ the same procedure to generate $TD(6n + 5)$. First, we enumerate for case $TD(11)$ to analyze the structure pattern for $TD(11)$. Then we use interval generation method to produce general algorithm.

5.1 Triad design for v=6n+5

Lemma 5.1.1. *If $v = 6n + 5$, then the number of triples in $CF(v)$, denoted by $|\overline{CF(v)}|$ is $2n(6n + 5)(3n + 2)$.*

Proof. The number of triples in $TD(v)$, denoted by $|TD(v)|$ is equal to
$|TD(v)| = \binom{v}{3} = \binom{6n+5}{3} = \frac{(6n+5)(6n+4)(6n+3)}{6} = (6n + 5)(3n + 2)(2n + 1)$.
But $|CF(v)| = v \times \frac{v-1}{2} = (6n + 5) \times \frac{(6n+5)-1}{2} = (6n + 5)(3n + 2)$.
Therefore,
$|\overline{CF(v)}| = |TD(v)| - |CF(v)| = (6n+5)(3n+2)(2n+1) - (6n+5)(3n+2) = 2n(6n+5)(3n+2)$. □

The following two lemmas provide the number of triples in $STD(v)$ as well as in $|\overline{SCF(v)}|$ denoted by $|STD(v)|$ and $|\overline{SCF(v)}|$ respectively.

Lemma 5.1.2. *If* $v = 6n + 5$*, then* $|STD(v)|$ *is* $(3n + 2)(2n + 1)$.

Proof. From the proof of Lemma 5.1.1, the number of triples of $TD(v)$ is equal to $|TD(v)| = (6n+5)(3n+2)(2n+1)$. By the definition of $TD(v)$, the number of rows of $TD(v)$ is equal to $v = 6n + 5$. Hence $|STD(v)| = (3n + 2)(2n + 1)$. □

Lemma 5.1.3. *If* $v = 6n + 5$*, then* $|\overline{SCF(v)}|$ *is even and* $2n(3n + 2)$.

Proof. By Lemma 5.1.1, $|\overline{CF(v)}|$ is equal to $2n(6n + 5)(3n + 2)$. Since the number of rows equal $v = 6n + 5$, then $|\overline{SCF(v)}| = 2n(3n + 2)$. □

In the next section, an algorithm for constructing $TD(11)$ is discussed as a basis for general construction for $TD(6n + 5)$.

5.2 Algorithm for TD(11)

In this section, addition modular 11 is used to produce the algorithm for $STD(11)$. Obviously, $STD(11) = SCF(11) \cup \overline{CF(11)}$. By Lemma 5.1.2, with $n = 1, |STD(11)| = 15$, and by Lemma 5.1.3,$|\overline{SCF(11)}| = 10$. The algorithm of $TD(11)$ is as follows:

Step 1. Generate $SCF(11)$ which consists of 5 triples as shown in Table 5.1.

Table 5.1: $SCF(11)$

1	2	11	3	10	4	9	5	8	6	7
i	$i+1$	$i-1$	$i+1$	$i-1$	$i+1$	$i-1$	$i+1$	$i-1$	$i+1$	$i-1$

Hence $SCF(11)$: {1 2 11; 1 3 10; 1 4 9; 1 5 8; 1 6 7.}

Note that the difference sets for each triple in $SCF(11)$ are listed in Table 5.2 using the facts that ± 10=±1,±9=± 2,± 8=±3,± 7=± 4,and ±6=± 5.

Table 5.2: Difference sets for $SCF(11)$

Triples	1 2 11	1 3 10	1 4 9	1 5 8	1 6 7
Difference set	±1,±2, ± 1	± 2, ± 4, ± 2	± 3, ± 5, ± 3	± 4, ± 3, ±4	± 5, ± 1, ± 5

It can be observed from Table 5.2, that each difference occurs 3 times in $SCF(11)$.

Step 2. Generate $\overline{SCF(11)}$ by using difference set method and addition modular 11. This is done in Table 5.3.

Table 5.3: $SCF(11)$ and their difference sets

2	3	9	$(i+1)(i+2)(i-3)$	± 1, ± 5,± 4	11	10	4	$(i-1)(i-2)(i+3)$	± 1, ± 5,± 4
2	4	8	$(i+1)(i+3)(i-4)$	± 2, ± 4,± 5	11	9	5	$(i-1)(i-3)(i+4)$	± 2, ± 4,±
2	5	7	$(i+1)(i+4)(i-5)$	±3, ± 2,± 5	11	8	6	$(i-1)(i-4)(i+5)$	±3, ± 2,± 5
3	4	7	$(i+2)(i+3)(i-5)$	± 1, ± 3,±4	10	9	6	$(i-2)(i-3)(i+5)$	± 1, ± 3,±4
4	5	7	$(i+3)(i+4)(i-6)$	± 1, ± 2,± 3	9	8	6	$(i-3))(i-4)(i+6)$	± 1, ± 2,± 3

Hence $\overline{SCF(11)} = \{2\,3\,9; 11\,10\,4; 2\,4\,8; 11\,9\,5; 2\,5\,7; 11\,8\,6; 3\,4\,7; 10\,9\,6; 4\,5\,7; 9\,8\,6\}$. From Table 5.3, each difference occurs 6 times in Since $STD(11) = SCF(11) \cup \overline{SCF(11)}$, we have $\lambda(STD(11)) = 9$.

Step 3. Starters from Step 1 and Step 2 will be the starter $STD(11)$. That is

$$
\begin{aligned}
STD(11) &= SCF(11) \cup \overline{SCF(11)} \\
&= \{1\,2\,11; 1\,3\,10; 1\,4\,9, 1\,5\,8; 1\,6\,7\} \cup \{2\,3\,9; 11\,10\,4; \\
&\quad 2\,4\,8; 11\,9\,5; 2\,5\,7; 11\,8\,6; 3\,4\,7; 10\,9\,6; 4\,5\,7; 9\,8\,6\}.
\end{aligned}
$$

Step 4. Using starter from step 3 and addition modular 11 to enumerate $TD(11)$ as shown in Table 5.4.

Table 5.4: $TD(11)$

1			2			3			4			5		
1	2	11	1	3	10	1	4	9	1	5	8	1	6	7
2	3	1	2	4	11	2	5	10	2	6	9	2	7	8
3	4	2	3	5	1	3	6	11	3	7	10	3	8	9
4	5	3	4	6	2	4	7	1	4	8	11	4	9	10
5	6	4	5	7	3	5	8	2	5	9	1	5	10	11
6	7	5	6	8	4	6	9	3	6	10	2	6	11	1
7	8	6	7	9	5	7	10	4	7	11	3	7	1	2
8	9	7	8	10	6	8	11	5	8	1	4	8	2	3
9	10	8	9	11	7	9	1	6	9	2	5	9	3	4
10	11	9	10	1	8	10	2	7	10	3	6	10	4	5
11	1	10	11	2	9	11	3	8	11	4	7	11	5	6

6			7			8			9			10		
2	3	9	11	10	4	2	4	8	11	9	5	2	5	7
3	4	10	1	11	5	3	5	9	1	10	6	3	6	8
4	5	11	2	1	6	4	6	10	2	11	7	4	7	9
5	6	1	3	2	7	5	7	11	3	1	8	5	8	10
6	7	2	4	3	8	6	8	1	4	2	9	6	9	11
7	8	3	5	4	9	7	9	2	5	3	10	7	10	1
8	9	4	6	5	10	8	10	3	6	4	11	8	11	2
9	10	5	7	6	11	9	11	4	7	5	1	9	1	3
10	11	6	8	7	1	10	1	5	8	6	2	10	2	4
11	1	7	9	8	2	11	2	6	9	7	3	11	3	5
1	2	8	10	9	3	1	3	7	10	8	4	1	4	6

11			12			13			14			15		
11	8	6	3	4	7	10	9	6	4	5	7	9	8	6
1	9	7	2	5	8	11	10	7	5	6	8	10	9	7
2	10	8	3	6	9	1	11	8	6	7	9	11	10	8
3	11	9	4	7	10	2	1	9	7	8	10	1	11	9
4	1	10	5	8	11	3	2	10	8	9	11	2	1	10
5	2	11	6	9	1	4	3	11	9	10	1	3	2	11
6	3	1	7	10	2	5	4	1	10	11	2	4	3	1
7	4	2	8	11	3	6	5	2	11	1	3	5	4	2
8	5	3	9	1	4	7	6	3	1	2	4	6	5	3
9	6	4	10	2	5	8	7	4	2	3	5	7	6	4
10	7	5	11	3	6	9	8	5	3	4	6	8	7	5

It clearly showa that an enumeration for $\overline{SCF(11)}$ is not suitable when v is large. Therefore, developing a new method to produce $\overline{SCF(11)}$ towards developing $\overline{SCF(v)}$, is the main objective of this chapter. Thus, the objective of the following section is to develop a new method to generate $STD(6n+5)$. This new method relies on analyzing the triples in using interval techniques of the number of triples.

5.3 Interval construction of STD(11)

From Step 3 of the previous section,

$$\begin{aligned} STD(11) &= SCF(11) \cup \overline{SCF(11)} \\ &= \{1\ 2\ 11; 1\ 3\ 10; 1\ 4\ 9, 1\ 5\ 8; 1\ 6\ 7\} \cup \{2\ 3\ 9; 11\ 10\ 4; \\ &\quad 2\ 4\ 8; 11\ 9\ 5; 2\ 5\ 7; 11\ 8\ 6; 3\ 4\ 7; 10\ 9\ 6; 4\ 5\ 7; 9\ 8\ 6\}. \end{aligned}$$

Let T_k be the k-th triple in $STD(11)$. Obviously, $1 \leq k \leq 15$ by Lemma 5.1.2. Furthermore, $STD(11) = T_1, \ldots, T_{15}$. Let $[S_rTD(11)]_k$ be the k-th element in $S_rTD(11)$ for $1 \leq r \leq 3$. We are able to summarize $STD(11)$ in terms $S_rTD(11)$ as shown in Table 5.5.

Table 5.5: $S_rTD(11)$

k	1	2	3	4	**5**	6	7	8	9	10	**11**	12	13	14	15
$S_1TD(11)$	1	1	1	1	1	2	11	2	11	2	11	3	10	4	9
$S_2TD(11)$	2	3	4	5	6	3	10	4	9	5	8	4	9	5	8
$S_3TD(11)$	11	10	9	8	7	9	4	8	5	7	6	7	6	7	6

It can be observed that from Table 5.5 that k, the number of triples is $STD(11)$, can be divided into three intervals. These intervals and the corresponding elements in $S_1TD(11)$ are summarized in Table 5.6.

Table 5.6: Intervals of k and the corresponding elements in $S_1TD(11)$

No. of Intervals	Intervals of k	Corresponding elements in $S_1TD(11)$
1	$1 \leq k \leq 5$	1, 1, 1, 1, 1, 1
2	$6 \leq k \leq 11$	2, 11, 2, 11, 2, 11
3	$12 \leq k \leq 15$	3, 10, 4, 9

It can be seen from Table 5.6, that the corresponding elements in $S_1TD(11)$ in the last interval of k, that is the third interval need a special formula to produce them. Let f denotes the first number of the interval. The formula $y = \frac{1}{2}[k - f + mod(\frac{k+1}{2}) + 1]$ can produce the corresponding elements in $S_1TD(11)$ in the indicated interval. For example, if $k = 13$, then $f = 12$ because 13 in the last interval $12 \leq k \leq 15$. Hence,

$$y = \frac{1}{2}[k - f + mod(\frac{k+1}{2}) + 1] = \frac{1}{2}[13 - 12 + mod(\frac{13+1}{2}) + 1] = 1.$$

Therefore, the corresponding elements in $S_1TD(11)$ for $k = 13$ is $11 - y = 11 - 1 = 10$, which is the same number as shown in Tables 5.5 and 5.6. From Tables 5.5 and 5.6, the construction of $S_1TD(11)$ is as follows.

$$[S_1TD(11)]_k = \begin{cases} 1 & \text{if } 1 \leq k \leq 5 \\ 2 & \text{if } 6 \leq k \leq 11, k \text{ is even} \\ 11 & \text{if } 6 \leq k \leq 11, k \text{ is odd} \\ 2+y & \text{if } 12 \leq k \leq 15, k \text{ is even} \\ 11-y & \text{if } 12 \leq k \leq 15, k \text{ is odd} \end{cases}$$

Similarly, intervals of k and the corresponding elements in $S_2TD(11)$ are shown in Table 5.7.

Table 5.7: Intervals of k and the corresponding elements in $S_2TD(11)$

No. of Intervals	Intervals of k	Corresponding elements in $S_2TD(11)$
1	$1 \leq k \leq 5$	2, 3, 4, 5, 6
2	$6 \leq k \leq 11$	3, 10, 4, 9, 5, 8
3	$12 \leq k \leq 15$	4, 10, 5, 8

Using the same formula $y = \frac{1}{2}[k - f + mod(\frac{k+1}{2}) + 1]$ to produce the corresponding elements in $S_2TD(11)$ in all intervals of k except the first one. For example, if $k = 10$, then $f = 6$ because 10 in the third interval $6 \leq k \leq 11$. Hence,

$$y = \frac{1}{2}[k - f + mod(\frac{k+1}{2}) + 1] = \frac{1}{2}[4 + 1 + 1] = 3.$$

Therefore, the corresponding elements in $S_2TD(11)$ for $k = 6$ is equal to $2 + y = 2 + 3 = 5$, which is the same number as shown in Tables 5.5 and 5.7.

Therefore, the construction of $S_2TD(11)$ is the following

$$[S_2TD(11)]_k = \begin{cases} 1+k & \text{if } 1 \le k \le 5 \\ 2+y & \text{if } 6 \le k \le 11, k \text{ is even} \\ 11-y & \text{if } 6 \le k \le 11, k \text{ is odd} \\ 3+y & \text{if } 12 \le k \le 15, k \text{ is even} \\ 11-y & \text{if } 12 \le k \le 15, k \text{ is odd} \end{cases}$$

Finally, similar to the above discussion, intervals of k and the corresponding elements in $S_3TD(11)$ are shown in Tables 5.8.

Table 5.8: Intervals of k and the corresponding elements in $S_3TD(11)$

No. of Intervals	Intervals of k	Corresponding elements in $S_3TD(11)$
1	$1 \le k \le 5$	11, 10, 9, 8,7
2	$6 \le k \le 11$	9, 4, 8, 5, 7, 6
3	$12 \le k \le 15$	7, 6, 7, 6

Using the same formula $y = \frac{1}{2}[k - f + mod(\frac{k+1}{2}) + 1]$ to produce the corresponding elements in all intervals of k except the first one. Hence from Tables 5.8, the construction of $S_3TD(11)$ is the following.

$$[S_3TD(11)]_k = \begin{cases} 12-k & \text{if } 1 \le k \le 5 \\ 10-y & \text{if } 6 \le k \le 11, k \text{ is even} \\ 3+y & \text{if } 6 \le k \le 11, k \text{ is odd} \\ 7 & \text{if } 12 \le k \le 15, k \text{ is even} \\ 6 & \text{if } 12 \le k \le 15, k \text{ is odd} \end{cases}$$

Now we are ready to consider the general case of the construction of $STD(v)$, where $v = 6n + 5$, which is the main objective of this research and what we are going to do in the next section. The process, as one can conclude from the previous discussions, depends on analyzing the triples in $STD(v)$ using interval techniques of the number of triples.

5.4 Intervals constructions for the general case STD(6n+5)

From the intervals construction of STD(11) in Sections 5.3, and from $STD(17)$ in appendix A, a following table of intervals for the cases $v = 11, 17, 23, \ldots, 6n+5$ is established. Table 5.9 is a result of analyzing the patterns of the triples in the designs of the previous cases. Some theorems and results are deduced from the Intervals Table 5.9.

Table 5.9: Intervals of $STD(6n + 5)$

intervals of k	11 n=1	17 n=2	23 n=3	$\ldots$	$6n + 5$
	$1 \leq k \leq 5$	$1 \leq k \leq 8$	$1 \leq k \leq 11$	$\ldots$	$1 \leq k \leq 3n + 2$
	$6 \leq k \leq 11$	$9 \leq k \leq 20$	$12 \leq k \leq 29$	$\ldots$	$3n + 3 \leq k \leq 9n + 2$
	$12 \leq k \leq 15$	$21 \leq k \leq 28$	$30 \leq k \leq 43$	$\ldots$	$9n + 3 \leq k \leq 15n - 2$
		$29 \leq k \leq 32$	$44 \leq k \leq 53$	$\ldots$	$15n - 1 \leq k \leq 21n + 2 - 12$
		$33 \leq k \leq 36$	$54 \leq k \leq 59$	$\ldots$	$21n - 9 \leq k \leq 27n + 2 - 24$
		$37 \leq k \leq 40$	$60 \leq k \leq 65$	$\ldots$	$\vdots$
			$66 \leq k \leq 69$	$\ldots$	$\vdots$
			$70 \leq k \leq 73$		
			$74 \leq k \leq 77$		$\vdots$
Total	3	6	9		$3n$

Theorem 5.4.1. *The number of intervals of $STD(v)$ is equal $3n$.*

Proof. It is obvious from Table 4.12 of intervals of $STD(6n + 5)$ that the pairs (n, number of intervals) are $(1, 3), (2, 6), (3, 9), \ldots$.
This is a linear function of constant slope equal to 3. Therefore, the numbers of of intervals $= 3n$. □

The general rules of the intervals for the general case $v = 6n + 5$ are:

1 - The first interval is $1 \leq k \leq 3n + 2$ which is related to $SCF(v)$.

2 - Other intervals except the last one are explained as follows: Let i denote the interval number starting from the second interval until the second to the last one. It can be observed from Theorem 5.4.1, that the value of i is $1 \leq i \leq 3n - 2$ That is, $i = 1$ is the second interval, $i = 2$ is the third interval,..., $i = 3n - 2$ is the interval second to last one.

The last numbers of each interval, denoted by l, are constructed as follows: In general

The intervals		The last number l
$3n + 3 \leq k \leq 9n + 2$	$\rightarrow$	$l = 9n + 2 = (6(1) + 3)n + 2 - (2(1)^2 + 2(1))$
$9n + 3 \leq k \leq 15n - 2$	$\rightarrow$	$l = 15n - 2 = (6(2) + 3)n + 2 - (2(2)^2 + 2(2))$
$15n - 1 \leq k \leq 21n - 10$	$\rightarrow$	$l = 21n - 10 = (6(3) + 3)n + 2 - (2(3)^2 + 2(3))$
	$\vdots$	

$l = (6(i) + 3)n + 2 - (2i^2 + 2i)$, where $1 \leq i \leq n + 1$.

Now, if $n + 2 \leq i \leq 3n - 2$ then from the Intervals Table 5.9 the difference between the last numbers of two successive intervals is smaller compared to the case when $1 \leq i \leq n + 1$. Consequently, a value denoted by a_t, defined below, must be added to obtain the desired last numbers of the intervals. Therefore, the last number of the general interval when $n + 1 \leq i \leq 3n - 2$ is equal to $l + a_t = (6(i) + 3)n + 2 - (2i^2 + 2i) + a_t$.

If $i = n + 1 + t$, clearly $1 \leq t \leq 2n - 1$. Define

$$a_t = \begin{cases} 0 & \text{if } t = 0 \\ 4 & \text{if } t = 1 \\ a_{t-2} + 6t - 2 & \text{if } t \geq 2 \end{cases}$$

Similarly, the first numbers of each interval, denoted by f, are constructed as follows:

The intervals		The first number f
$3n + 3 \leq k \leq 9n + 2$	$\rightarrow$	$f = 3n + 3 = [6(1 - 1) + 3]n + 2 - [2(1 - 1)^2 + 2(1 - 1))] + 1$
$9n + 3 \leq k \leq 15n - 2$	$\rightarrow$	$f = 9n + 3 = [6(2 - 1) + 3]n + 2 - [2(2 - 1)^2 + 2(2 - 1)] + 1$
$15n - 1 \leq k \leq 21n - 10$	$\rightarrow$	$f = 15n - 1 = [6(3 - 1) + 3]n + 2 - [2(3 - 1)^2 + 2(3 - 1)] + 1$
	$\vdots$	

In general, $F = [6(i - 1) + 3]n + 2 - [2(i - 1)^2 + 2(i - 1)] + 1$, where $1 \leq i \leq n + 2$. For the same reason above, if $n + 1 \leq i \leq 3n - 2$ then a value denoted by a_{t-1}, defined above, must be added to obtain the desired first numbers of the intervals. Therefore, the first number of the general interval when $n + 2 \leq i \leq 3n - 2$ is equal to

$$f + a_{t-1} + 1 = (6(i - 1) + 3)n + 2 - (2(i - 1)^2 + 2(i - 1)) + a_{t-1} + 1.$$

where $i = n + +1t, 1 \leq t \leq 2n - 3$.

The above discussion and notations are summarized in the following generalization concerning the patterns of the intervals for the general case $v = 6n + 5$.

Interval Generalization

Regarding the above discussion and notations, the patterns of the intervals for the general case $v = 6n + 5$ are

1 - The first interval is $1 \leq k \leq 3n + 2$.

2 - Other intervals except the last one are explained as follows

$f \leq k \leq l$ when $1 \leq i \leq n + 1$, and

$F + a_{t-1} \leq k \leq l + a_t$ when $n + 2 \leq i \leq 3n - 2$, $i = n + t$, $1 \leq t \leq 2n - 3$.

Also $a_0 = 0$, $a_1 = 4$, and if $t \geq 2$, then $a_t = a_{(t-2)} + (6t - 2)$.

Example 5.4.2. *Let $n = 2$, that is $v = 17$, then by Theorem 5.4.1, the number of intervals is equal to 6. Clearly,$1 \leq i \leq 4$ and $t = 1$. By the Interval Generalization the intervals are*

1 - The first interval is $1 \leq k \leq 3n + 2$, that is $1 \leq k \leq 8$.

2 - If $i = 1$, then the second interval is

$$f \leq k \leq l$$
$$(6(i-1)+3)n + 2 - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n + 2 - (2i^2 + 2i)$$
$$(6(1-1)+3)2 + 2 - (2(1-1)^2 + 2(1-1)) + 1 \leq k \leq (6(1)+3)2 + 2 - (2(1)^2 + 2(1))$$
$$9 \leq k \leq 20$$

If $i = 2$, then the third interval is

$$f \leq k \leq l$$
$$(6(i-1)+3)n + 2 - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n + 2 - (2i^2 + 2i)$$
$$(6(2-1)+3)2 + 2 - (2(2-1)^2 + 2(2-1)) + 1 \leq k \leq (6(2)+3)2 + 2 - (2(2)^2 + 2(2))$$
$$21 \leq k \leq 28$$

If $i = 3$, then the forth interval is

$$f \leq k \leq l$$
$$(6(i-1)+3)n + 2 - (2(i-1)^2 + 2(i-1)) + 1 \leq k \leq (6(i)+3)n + 2 - (2i^2 + 2i)$$
$$(6(3-1)+3)2 + 2 - (2(3-1)^2 + 2(3-1)) + 1 \leq k \leq (6(3)+3)2 + 2 - (2(3)^2 + 2(3))$$
$$29 \leq k \leq 32$$

3 - Now $i > n + l = 3$, so the second formula will be used.

If $i = 4$, then $t = 1$. Hence the fifth interval is

$$f + a_{t-1} \leq k \leq l + a_t$$

$$(6(i-1)+3)n + 2 - (2(i-1)^2 + 2(i-1)) + a_{t-1} + 1 \leq k \leq (6(i)+3)n + 2 - (2i^2 + 2i) + a_t$$

$$(6(4-1)+3)2 + 2 - (2(4-1)^2 + 2(4-1)) + a_{1-1} + 1 \leq k \leq (6(4)+3)2 + 2 - (2(4)^2 + 2(4)) + a_1$$

$$33 \leq k \leq 36$$

We are now in a position to give the constructions of $S_rTD(v)$, for $1 \leq r \leq 3$. Let $v = 6n+5$. Let T_k be the k-th triple in $STD(v)$. By Lemma 5.1.2, $1 \leq k \leq (3n+2)(2n+1)$. Let $[S_rTD(v)]_k$ be the k-th element in $S_rTD(v)$ for $1 \leq r \leq 3$. If i denotes the interval number starting from the second interval until the second to last one, then by Theorem 5.4.1, $1 \leq i \leq 3n-2$ Let t be a variable appears when $i > n$. If $i = n+1+t$, then clearly $1 \leq t \leq 2n-3$. Define

$$a_t = \begin{cases} 0 & \text{if } t = 0 \\ 4 & \text{if } t = 1 \\ a_{t-2} + 6t - 2 & \text{if } t \geq 2 \end{cases}$$

We set in addition to the above notations, the formula $y = \frac{1}{2}[k - f + mod(\frac{k+n}{2}) + 1]$.

i) The construction of $S_1TD(6n+5)$ is as follows.

$$[S_1TD(v)]_k = \begin{cases} 1 & \text{if } 1 \le k \le 3n+2 \\ i+1 & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is even} \\ 6n+6-i & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is odd} \\ i+1 & \text{if } f + a_{t-1} + 1 \le k \le f + a_t, i = n+1+t, k \text{ is even} \\ 6n+6-i & \text{if } f + a_{t-1} + 1 \le k \le f + a_t, i = n+1+t, k \text{ is odd} \\ & n+1 \le i \le 3n-2, 1 \le t \le 2n-3 \\ 3n-1+y & \text{if } l + a_t + 1 \le k \le n(6n-1), k \text{ is even} \\ 3n+8-y & \text{if } l + a_t + 1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

ii) The construction of $S_2TD(6n+5)$ is as follows.

$$[S_2TD(v)]_k = \begin{cases} k+1 & \text{if } 1 \le k \le 3n+2 \\ i+1+y & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is even} \\ 6n+6-i-y & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is odd} \\ i+1+y & \text{if } f + a_{t-1} + 1 \le k \le f + a_t, i = n+1+t, k \text{ is even} \\ 6n+6-i-y & \text{if } f + a_{t-1} + 1 \le k \le f + a_t, i = n+1+t, k \text{ is odd} \\ & n+1 \le i \le 3n-2, 1 \le t \le 2n-3 \\ 3n+y & \text{if } l + a_t + 1 \le k \le n(6n-1), k \text{ is even} \\ 3n+7-y & \text{if } l + a_t + 1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

iii) The construction of $S_3TD(6n+5)$ is as follows.

$$[S_3TD(v)]_k = \begin{cases} 6n+6-k & \text{if } 1 \le k \le 3n+2 \\ 6n+6-2i-y & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is even} \\ 2i+1+y & \text{if } f \le k \le l, 1 \le i \le n+1, k \text{ is odd} \\ 6n+6-2i-y+Z_x & \text{if } f+a_{t-1}+1 \le k \le f+a_t, i = n+1+t, k \text{ is even} \\ 2i+1+y-Z_x-1 & \text{if } f+a_{t-1}+1 \le k \le f+a_t, i = n+1+t, k \text{ is odd} \\ & n+1 \le i \le 3n-2, 1 \le t \le 2n-3 \\ 3n+5-y & \text{if } l+a_t+1 \le k \le n(6n-1), k \text{ is even} \\ 3n+4-y & \text{if } l+a_t+1 \le k \le n(6n-1), k \text{ is odd} \end{cases}$$

Where $Z_x = Z_{x-1} + 1 + mod(\frac{x+1}{2}), Z_1 = 0, x = i-n$, and $Z_x = 0$ when $x < 1$.

Chapter 6

Conclusions

This research focused on with the development of triad design $TD(v)$ algorithm. The general idea to construct $TD(v)$ which was based on compatible factorization, $CF(v)$ a generalization from one-factorization. We employed $CF(v)$ as a basic step to generate starters as presented in Chapter 3. Theoretical work for $SCF(v)$ was highlighted in Section 3.4.

The crux of our study was a development of new algorithms for triad design. This design existed when we have $v = 6n + 1$ and $v = 6n + 5$ (Theorem 4.1.4). From this theorem, we develop $TD(6n + 1)$ and $TD(6n + 5)$. We started by providing several lemmas (refer to Lemma 4.1.2, Lemma 4.2.6, Lemma 4.2.7) as a ground work for $TD(v)$ algorithm. In order to develop general case for $TD(6n + 1)$, we have to enumerate several cases e.g $v = 7$ and $v = 13$. We analyzed the structure for these cases to provide algorithm for starters. We introduced new technique, "Interval Generation Method" to generate starters for $TD(6n + 1)$ as presented in Section 4.3.

A host of extensions for these ideas to case $TD(6n+5)$ follow in Chapter 5, where the Interval Construction Method for the general case $STD(6n + 5)$ were presented in Section 5.4. Theorems and lemmas (refer to Lemma 5.1.1, Lemma 5.1.2, Lemma 5.1.3, Theorem 5.4.1 and Interval Generalization Rules) were provided to support construction of $TD(6n + 5)$. We have developed a new algorithm for $TD(6n + 1)$ and $TD(6n + 5)$.

Now could we go further and develop algorithms for other cases e.g. $v = 6n, v = 6n+2, v = 6n + 3$ or $v = 6n + 4$? We know that it is impossible to construct TDs for $v \equiv 0, 2, 3$ or 4 (mod 6). However, we can ask how close we can come to construct these cases. One can think of using the ideas of *covering and packing* [9].

References

[1] T. M. Abu Saa and H. Ibrahim. On the structure of 1-factorizations of the complete graph. *Proceedings of the 5th UAE Math-Day Conference, Etisalat University College, Sharjah, United Arab Emirates*, pages 1–4, 2007.

[2] T. M. Abu Saa and H. Ibrahim. Triad design: A development of five and seven objects using difference set method. *Proceeding of the 6th UAE Math day Conference. The Petroleum Institute, Abu Dhabi, UAE*, pages 12–20, 2008.

[3] T. M. Abu Saa and H. Ibrahim. A new algorithm for some triad designs. *Proceeding of The First International Conference on Mathematics and Statistics. University of Sharjah, Sharjah, United Arab Emirates*, pages 30–38, 2010.

[4] C. J. Colbourn and J. H.. Ainitz. *Handbook of Combinatorial Designs*. Taylor and Francis Group, LLC, 2nd edition edition, 2007.

[5] C. J. Colbourn and Alexander R. *Triple System*. Oxford University Press Inc., New York, 1999.

[6] J. A. Davis and J. Jedwab. *Some Recent Developments in Difference Sets*, chapter Some Recent Developments in Difference Sets. Pitmen research note, London, 1997.

[7] J. H. Dinitz and Stinson D. R. *One-Factorization of Complete Graph, Contemporary Design Theory*. Wiley, New York, 1997.

[8] H. Ibrahim. *A Family of Designs for Triads and Related Factorizations*. PhD thesis, Southern Illinois University, Carbondale, 2003.

[9] C. C. Lindner and Roger C. A. *Design Theory*. CRC Press, New York, 1997.

[10] C. C. Lindner, E. Mendelson, and A. Rosa. On the number of 1-factorization of the complete graph. *J. combin. Theory*, 1976.

[11] W.D. Wallis. *One-factorization*. Kluwer Acad. Publ., 1977.

[12] W.D. Wallis. *Combinatorial Designs*. Marcel Dekker, New York, 1988.

[13] W.D. Wallis. Tournaments for triads. *Bulletin Inst. Combin. Appl.*, pages 27–32, 2005.

Appendix A: TD(6n+1)

```
clear all
n=4
v=6*n+1;
NN=nchoosek(v,3)/v;
Numbers=zeros(v,NN,3);
DisplayArray=zeros(v,NN);
jj=1;
At(1)=2;At(2)=8;
for tt=3:15; At(tt)=At(tt-2)+(6*tt-4); end
Ck(1)=0;
for kk=2:15; Ck(kk)=Ck(kk-1)+mod(kk+1,2)+1; end
Att=0;
Ckk=-1;
for ii=0:3*n-4
    tt=ii-n;
    if tt>0; Att=At(tt); Ckk=Ck(tt); end
    UpperLimit=(6*ii+3)*n-2*ii*(ii+1)+Att;
    for xx=jj:UpperLimit
```

```
        if ii==0
            yy(xx)=xx;
            Numbers(1,xx,1)=1;
            Numbers(1,xx,2)=1+yy(xx);
            Numbers(1,xx,3)=6*n+2-yy(xx);
        else and(ii>3*n,ii<=3*(n-1)-1)
            if mod(xx,2)==0
                yy(xx)=0.5*(xx-jj+mod(xx+n,2)+1);
                Numbers(1,xx,1)=ii+1;
                Numbers(1,xx,2)=ii+1+yy(xx);
                Numbers(1,xx,3)=2*ii+yy(xx)-Ckk;%mod(xx+n,2);
            else
                yy(xx)=0.5*(xx-jj+mod(xx+n,2)+1);
                Numbers(1,xx,1)=v-ii+1;
                Numbers(1,xx,2)=6*n+2-ii-yy(xx);
                Numbers(1,xx,3)=6*n+2-2*ii-yy(xx)+Ckk+1;%+mod(xx+n,2);
            end
        end
    end
    jj=UpperLimit+1
end
Numbers(1,NN-3,1)=3*n-2;
Numbers(1,NN-2,1)=3*n+5;
Numbers(1,NN-1,1)=3*n-1;
Numbers(1,NN,1)=3*n+4;
```

```
for jj=2:v
    for ii=1:NN
        for kk=1:3
            if Numbers(jj-1,ii,kk)==v
                Numbers(jj,ii,kk)=1;
            else
                Numbers(jj,ii,kk)=Numbers(jj-1,ii,kk)+1;
            end %if
        end %for kk
    end %for ii
end %for jj
for ii=1:v
    for jj=1:NN
    DisplayArray(ii,jj)=Numbers(ii,jj,3)+
    100*Numbers(ii,jj,2)+10000*Numbers(ii,jj,1);
    end
end
DisplayArray
```

Appendix : TD(6n+5)

```
clear all
n=1
v=6*n+5;
NN=nchoosek(v,3)/v;

Numbers=zeros(v,NN,3);
DisplayArray=zeros(v,NN);

jj=1;

At(1)=4;At(2)=10;
for tt=3:15; At(tt)=At(tt-2)+(6*tt-2); end

Ck(1)=0;
for kk=2:15; Ck(kk)=Ck(kk-1)+mod(kk+1,2)+1; end
Att=0;
Ckk=-1;
for ii=0:3*n-2
```

```
    tt=ii-n-1;
    if tt>0;Att=At(tt);end
    UpperLimit=(6*ii+3)*n+2-2*ii*(ii-1)+Att;
    for xx=jj:UpperLimit
        if ii==0
            yy(xx)=xx;
            Numbers(1,xx,1)=1;
            Numbers(1,xx,2)=1+yy(xx);
            Numbers(1,xx,3)=6*n+6-yy(xx);
        else and(ii>3*n+2,ii<=3*(n-1)-1)
            if mod(xx,2)==0
                yy(xx)=0.5*(xx-jj+mod(xx+n,2)+1);
                Numbers(1,xx,1)=ii+1;
                Numbers(1,xx,2)=ii+1+yy(xx);
                Numbers(1,xx,3)=6*n+6-2*ii-yy(xx)-Ckk-1;%+mod(xx+n,2);
            else
                yy(xx)=0.5*(xx-jj+mod(xx+n,2)+1);
                Numbers(1,xx,1)=6*n+6-ii;
                Numbers(1,xx,2)=6*n+6-ii-yy(xx);
                Numbers(1,xx,3)=2*ii+yy(xx)-Ckk;%+mod(xx+n,2);
            end
        end
    end
    jj=UpperLimit+1
end
```

```
if mod(NN,2)==0
    Numbers(1,NN-3,1)=3*n+7;
    Numbers(1,NN-2,1)=3*n;
    Numbers(1,NN-1,1)=3*n+6;
    Numbers(1,NN,1)=3*n+1;

    Numbers(1,NN-3,2)=3*n+6;
    Numbers(1,NN-2,2)=3*n+1;
    Numbers(1,NN-1,2)=3*n+5;
    Numbers(1,NN,2)=3*n+2;

    Numbers(1,NN-3,3)=3*n+3;
    Numbers(1,NN-2,3)=3*n+4;
    Numbers(1,NN-1,3)=3*n+3;
    Numbers(1,NN,3)=3*n+4;
else
    Numbers(1,NN-3,1)=3*n;
    Numbers(1,NN-2,1)=3*n+7;
    Numbers(1,NN-1,1)=3*n+1;
    Numbers(1,NN,1)=3*n+6;

    Numbers(1,NN-3,2)=3*n+1;
    Numbers(1,NN-2,2)=3*n+6;
    Numbers(1,NN-1,2)=3*n+2;
    Numbers(1,NN,2)=3*n+5;
```

```
    Numbers(1,NN-3,3)=3*n+4;
    Numbers(1,NN-2,3)=3*n+3;
    Numbers(1,NN-1,3)=3*n+4;
    Numbers(1,NN,3)=3*n+3;
end


for jj=2:v
    for ii=1:NN
        for kk=1:3
            if Numbers(jj-1,ii,kk)==v
                Numbers(jj,ii,kk)=1;
            else
                Numbers(jj,ii,kk)=Numbers(jj-1,ii,kk)+1;
            end %if
        end %for kk
    end %for ii
end %for jj

for ii=1:v
    for jj=1:NN
        DisplayArray(ii,jj)=Numbers(ii,jj,3)+
        100*Numbers(ii,jj,2)+10000*Numbers(ii,jj,1);
    end
```

```
end

DisplayArray
```