Exploration of Algorithms Related to Independent Sets of Steiner Triple Systems

by

Zhaomeng Wang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2021 by the
Graduate Supervisory Committee:

Charles Colbourn, Chair
Andrea Richa
Zilin Jiang

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

In combinatorial mathematics, a Steiner system is a type of block design. A Steiner triple system is a special case of Steiner system where all blocks contain 3 elements and each pair of points occurs in exactly one block. Independent sets in Steiner triple systems is the topic which is discussed in this thesis. Some properties related to independent sets in Steiner triple system are provided. The distribution of sizes of maximum independent sets of Steiner triple systems of specific order is also discussed in this thesis. An algorithm for constructing a Steiner triple system with maximum independent set whose size is restricted with a lower bound is provided. An alternative way to construct a Steiner triple system using an affine plane is also presented. A modified greedy algorithm for finding a maximal independent set in a Steiner triple system and a post-optimization method for improving the results yielded by this algorithm are established.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

A Steiner system is a type of a block design which is a topic in combinatorial design. *Combinatorial design theory* is the part of combinatorial mathematics that deals with the existence, construction and properties of *systems of finite sets* whose arrangements satisfy generalized concepts of balance and/or symmetry. Modern applications are found in a wide gamut of areas including finite geometry, algorithm design and analysis, networking and cryptography [1].

A *Steiner system* with parameters t, k, n written $\mathrm{S}(t, k, n)$, is an n-element set S together with a set of k-element subsets of S (called blocks) with the property that each t-element subset of S is contained in exactly one block. An $\mathrm{S}(2, 3, n)$ is called a *Steiner triple system*, and its blocks are called triples.

A Steiner triple system of order n, or $STS(n)$ for short, is a pair $(X, \mathcal{T})$, where X is a set of elements, $|X| = n$, called *points* and $\mathcal{T}$ is a set of 3-element subsets of X called *triples*, with the property that each (unordered) pair of points belonging to X occurs in exactly one triple in $\mathcal{T}$.

A *partial Steiner triple system* is a set system (V, B) in which every block has size three, and every pair of points from V is contained in at most one block. Such a set system is denoted $PSTS(n)$, where $n = |V|$.

By the definition of Steiner triple system, it is easy to obtain that the total amount of triples in an STS of order n is $\frac{n*(n-1)}{6}$. An $STS(n)$ exists if and only if $n \equiv 1$ or 3 (mod 6) [2].

A Fano plane (Figure 1) is an example of $STS(7)$. Each line (no matter straight

or not) that goes through 3 points can be interpreted as a triple in the STS, because any pair of points is contained in exactly one line.

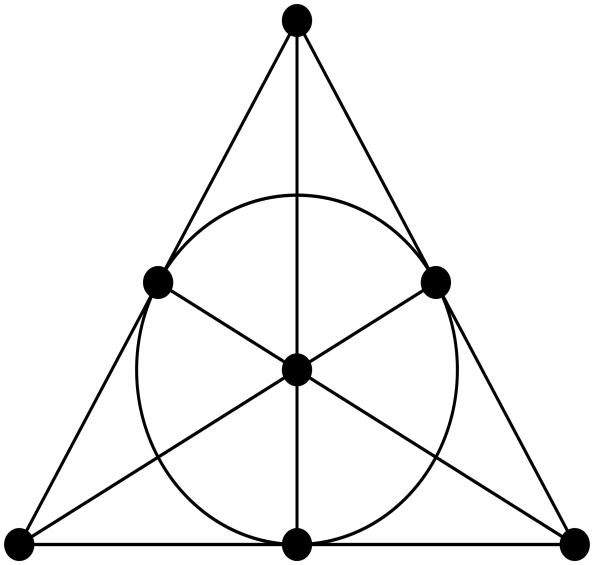

Figure 1: The Fano Plane

An *independent set* in a STS $S = (X, \mathcal{T})$ is a subset $I \subseteq X$ with the property that no triple in $\mathcal{T}$ is contained in I. A *maximal independent set* is an independent set $I \subseteq X$ with the property that for any $x \in X \setminus I$, there exists a triple $T \in \mathcal{T}$ such that $I \cup \{x\}$ contains T. A *maximum independent set* is an independent set $I \subseteq X$ with the property that there does not exist an independent set $M \subseteq X$ where $|M| > |I|$. The *independence number* of S, denoted $\alpha(S)$, is the size of the maximum independent set of S.

A $STS(9)$ is shown in Figure 2. The triangles whose three edges are of same color are the triples of the STS. The set $\{1, 3, 4, 9\}$ is an independent set since none of $\{1, 3, 4\}$, $\{1, 3, 9\}$, $\{1, 4, 9\}$, $\{3, 4, 9\}$ is in a triangle with three edges of same color. Moreover, this set is also a maximal and maximum independent set. One can try add

any point of $\{2, 5, 6, 7, 8\}$ to this set. There is always a triple being contained in the new set.

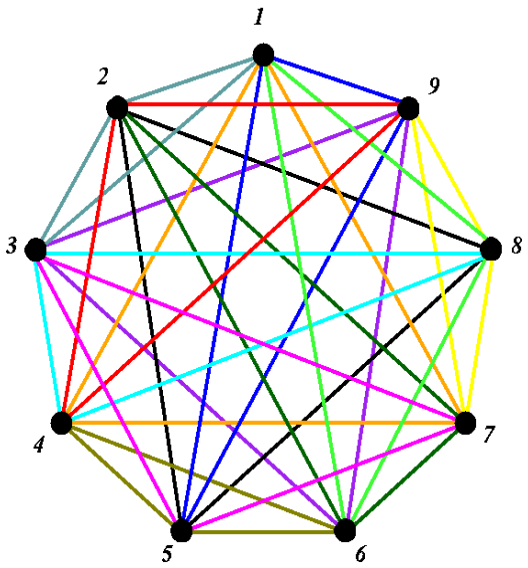

Figure 2: A Steiner triple system of order 9

Let $(\mathcal{V}, \mathcal{B})$ and $(\mathcal{W}, \mathcal{D})$ be two triple systems. Let $\phi : V \mapsto W$ be a one-to-one mapping. Under ϕ, every triple $B = \{x, y, z\} \in \mathcal{B}$ maps to a triple $\phi(B) = \{\phi(x), \phi(y), \phi(z)\}$. If ϕ *preserves the triples*, i.e. $\mathcal{D} = \{\phi(B) : B \in \mathcal{B}\}$, then ϕ is an *isomorphism* from $(\mathcal{V}, \mathcal{B})$ to $(\mathcal{W}, \mathcal{D})$. The two systems are *isomorphic* if there is an isomorphism from one to the other, and are *nonismorphic* otherwise.

Let $NN(n)$ denote the number of nonisomorphic $STS(n)$s, and let $ND(n)$ denote the number of distinct $STS(n)$s. Aleksejev [3] obtained the lower bound that

$$NN(n) \geq n^{\frac{n^2}{12} - O(\frac{n^2}{\log(n)})}$$

Since $NN(n) \leq ND(n)$, we have

$$ND(n) \geq n^{\frac{n^2}{12} - O(\frac{n^2}{\log(n)})}$$

Doyen and Valette [4] established an upper bound for $ND(n)$.

Let $(V, \mathcal{B})$ be an $STS(n)$. Place an arbitrary total order $\prec$ on V, and let $\sqsubset$ be the total order on $\binom{V}{2}$ defined by $\{u, v\} \sqsubset \{x, y\}$ whenever $u \prec v, x \prec y$, and either $v \prec y$, or $v = y$ and $u \prec x$. For each triple $B_i = \{x, y, z\} \in \mathcal{B}$, call the first pair in B_i in the order $\sqsubset$ the *representative pair*, P_i. Now order the blocks $B_1, ..., B_b$ of $\mathcal{B}$ so that $P_i \sqsubset P_{i+1}$ for $1 \leq i < b$. For $1 \leq i \leq b$, let x_i be the element for which $P_i \cup \{x_i\} = B_i$. Form the $b-$tuple $(x_1, ..., x_b)$. Since every distinct $STS(n)$ leads to a distinct $b-$tuple in this way, $ND(n)$ does not exceed the number of ways to form a $b-$tuple of elements from V, not using either of the first two elements under $\prec$.

Hence,

$$ND(n) \leq (n-2)^{\frac{n^2}{6}}$$

Not only the amount of distinct Steiner triple systems for a specific order is huge, the independence numbers of them are also varied. [5]. Define

$$\beta_{max}(n) = \max \{\alpha(S) : S \text{ is an } STS(n)\}$$
$$\beta_{min}(n) = \min \{\alpha(S) : S \text{ is an } STS(n)\}$$

An early result of Sauer and Schönheim [6] determines that

$$\beta_{max}(n) = \begin{cases} (n+1)/2 & \text{if } n \equiv 3,7 (\text{mod } 12) \\ (n-1)/2 & \text{if } n \equiv 1,9 (\text{mod } 12) \end{cases} \tag{1.1}$$

For the minimum independence number, Phelps and Rödl [7] proved that

$$c_1\sqrt{n \log n} \leq \beta_{min}(n) \leq c_2\sqrt{n}\log n, \text{ for all } n \geq n_0(c_2)$$

where c_1 is an absolute constant, c_2 is any constant greater than 4 and $n_0(c_2)$ is a constant dependent on the choice of c_2.

In the following chapter, we discuss the distribution of independence numbers for a given order.

Finding large independent sets of Steiner triple systems has applications on labelling problem. A labelling of S is a bijection $l : X \to \{0, 1, ..., n-1\}$. For each triple $T \in \mathcal{T}$, let sum(T) to be the triple-sum $\sum_{x \in T} l(x)$. Then, the following functions are defined [8].

1. The *min-sum* of the STS with respect to l is given by $\min_{\sum}(\mathcal{T}) \triangleq \min_{T \in \mathcal{T}} \text{sum}(T)$

2. The *max-sum* of the STS with respect of l is given by $\max_{\sum}(\mathcal{T}) \triangleq \max_{T \in \mathcal{T}} \text{sum}(T)$

3. The *difference-sum* of the STS is given by $\Delta_{\sum}(\mathcal{T}) \triangleq \max_{\sum}(\mathcal{T})\text{-}\min_{\sum}(\mathcal{T})$

To find a relatively good labelling l of S which makes the difference-sum of S small, finding two large disjoint independent sets would seem to be useful in this situation. Label one independent set with low values and the other one with high values. For the remaining points which are not chosen to be in an independent set, label with mid-range values. This strategy ensures that no triple gets a small sum and also no triple gets a large sum. Thus, no triple can appear in the low-valued independent set and no triple is contained in the high-valued independent set. Hence, the *min-sum* would be high and the *max-sum* would be low [9].

Given this observation, we then focus on the approaches for finding a maximum independent set in an arbitrary Steiner triple system.

Chapter 2

DISTRIBUTION OF SIZES OF MAXIMUM INDEPENDENT SETS

In this section, we first present the hill-climbing algorithm for constructing a random Steiner triple system and a modified algorithm for constructing a STS whose independence number is restricted with a lower bound.

After discussing the algorithm, we then demonstrate the experimental results of the distribution of sizes of maximum independent sets for some orders.

2.1 A hill-climbing algorithm for constructing a Steiner triple system

Hill-climbing is a mathematical optimization technique which belongs to the family of local search in numerical analysis. Basically, it is an iterative algorithm which starts with an arbitrary solution to a problem and then attempts to find a better solution by iteratively making a non-decreasing change to the solution. By the definition, it is easy to comprehend that hill-climbing is able to find optimal solutions for convex problems. For other problems, it will find only local optima which are not necessarily the best possible solution out of all possible solutions.

The hill-climbing algorithm for Steiner triple systems, which is introduced by Stinson[10]. One can find the description of the algorithm in [11] The problem description can be written as follows.

Instance: A positive integer $v \equiv 1$ or 3 mod 6

a finite set $V, |V| = v$

Find: the maximum value of $|B|$

subject to (V, B) is a $PSTS(v)$

Given v and V, define the universe, X, to consist all sets of blocks B such that (V, B) is a $PSTS(v)$. Hence, any set $B \in X$ is a feasible solution. Thus, an optimal solution is a feasible solution of size equal to $v(v-1)/6$.

The main heuristic of this hill-climbing method is **SWITCH**. The heuristic **SWITCH** transforms any $PSTS(v)$ into a different $PSTS(v)$, such that the size of the system either remains the same or is increased by one.

The description of hill-climbing algorithm is as follows.

Algorithm 1 Hill-climbing algorithm for constructing a Steiner triple system

Input: an integer n congruent to 1 or 3 mod 6

Output: a set of triples $\mathcal{T}$

1: Form a set V containing n points
2: Initialize an empty set $\mathcal{T}$
3: Initialize a counter $C = 0$
4: **While** the amount of triples is less than $\frac{n*(n-1)}{6}$ and C is less than a big value depending on n
5: $C = C + 1$
6: Let $\{a, b\}$ be an available pair which is not contained in a triple.

7: Randomly choose a point c such that $\{a, c\}$ is not contained in a triple. If there is no such a point, choose a pair of points again.

8: **If** $\{b, c\}$ is already contained in a triple, then delete the triple which contains it. Let $\{a, b, c\}$ form a new triple T and add it to $\mathcal{T}$.

9: **Otherwise**, directly form a new triple T with a, b and c. Add it to $\mathcal{T}$

10: If the total amount of triples is $\frac{n*(n-1)}{6}$, return $\mathcal{T}$; Otherwise, return $Fail$

The **Step 4-9** performs exactly what **SWITCH** does.

In each iteration, the algorithm either directly form a new triple or delete an old triple and construct a new one. Hence, in each iteration, the amount of constructed triples does not decrease. However this algorithm sometimes fails to return a Steiner triple system.

For example, for order 15, the algorithm form 25 triples defined by $\{a, 5 + b, 10 + ((a + b) \bmod 5)\}$ for $0 \leq a, b \leq 4$. Starting from this point, it should only succeed in adding 6 more triples. Because the triples that can be added are only from 0,1,2,3,4 or 5,6,7,8,9 or 10,11,12,13,14 respectively. And for each of these three subsets, it can only form two more triples. Since, the total amount of triples of order 15 is $15 * 14/6 = 35$, the algorithm will never reach this value as it should.

In order to compute the independence number of an arbitrary Steiner triple system, a program for applying integer programming (IP) solver to get the optimal solution of independent set in an arbitrary Steiner triple system is implemented.

The IP formulation for the problem of computing the size of maximum independent set in a Steiner triple system $S = (X, \mathcal{T})$ is as follows. (a reader who is unfamiliar with the basic definitions of integer programming is advised to read the relevant section in [12])

$$\max \quad \sum_{i=1}^{|X|} x_i$$

$$\text{s.t.} \quad \sum_{v_i \in T} x_i \leq 2 \quad \text{for each } T \in \mathcal{T}$$
$$x_i \in \{0, 1\} \quad \text{for each } v_i \in X$$

For an arbitrary Steiner triple system with order 61 or less, the program is able to yield the solution in 5 minutes.

Next, we show some bar graphs about the distributions of independence numbers of Steiner triple systems of specific orders constructed using the hill-climbing method introduced previously.

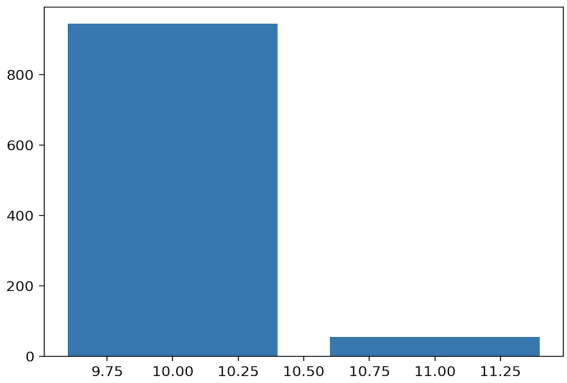

Figure 3: The distribution of order 25 with 1000 samples

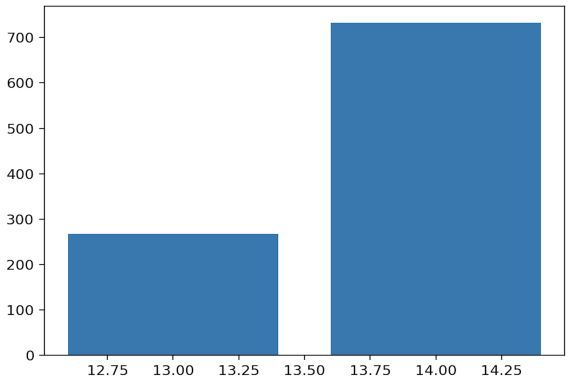

Figure 4: The distribution of order 37 with 1000 samples

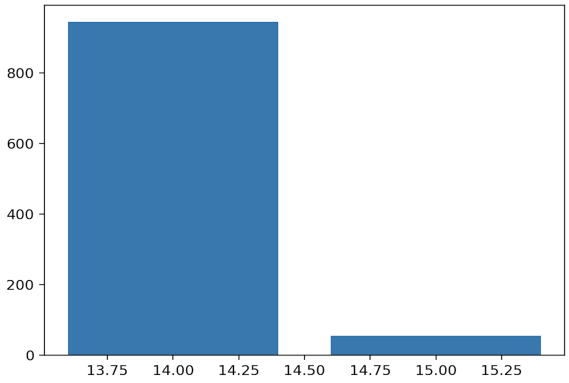

Figure 5: The distribution of order 39 with 1000 samples

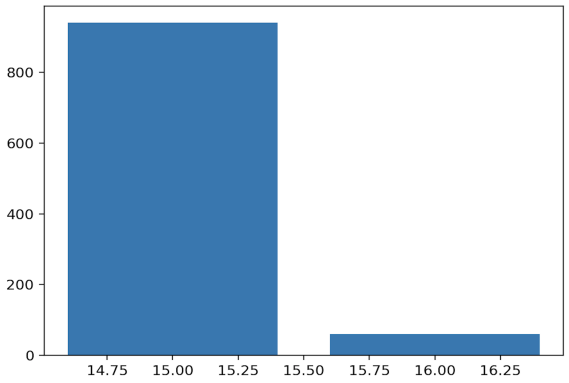

Figure 6: The distribution of order 43 with 1000 samples

v	λ	$B(v,\lambda)$
7	1	$\{4\}$
9	1	$\{4\}$
13	1	$\{6\}$
15	1	$\{6,7,8\}$
19	1	$\{7,8,9,10\}$
21	1	$\supseteq \{8,9,10\}$
25	1	$\supseteq \{9,10,11,12\}$
27	1	$\supseteq \{9,10,11,12,13,14\}$
31	1	$\supseteq \{11,12,13,14,15,16\}$
33	1	$\supseteq \{12,13,14,15,16\}$
37	1	$\supseteq \{13,14,15,16,17,18\}$
39	1	$\supseteq \{13,14,15,16,17,18,19,20\}$
43	1	$\supseteq \{14,15,16,17,18,19,20,21,22\}$
45	1	$\supseteq \{15,16,17,18,19,20,21,22\}$

Figure 7: Table 17.2 from Triple Systems [5]

As you can see in the graphs, the distributions concentrate on one or two values that is far away from the maximum independence numbers. Hence, we implemented following algorithms to construct Steiner triple systems with restricted independence number. One can check Figure 7 which is from [5] showing the independence numbers of Steiner triple systems from order 7 to order 45 for a comparison.

2.2 Algorithm for constructing a Steiner triple system whose independence number is restricted with a lower bound

One can find a brief introduction of this algorithm in [5]. There is an interesting phenomenon about the results yielded by this algorithm. One can check Figure 12 and notice that when the input lower bound is relatively small, the independence number of the system constructed by this algorithm is greater than the lower bound while when the input lower bound is close to the maximum one, the independence number of the system constructed by this algorithm is exactly equal to the lower bound.

Algorithm 2 Algorithm for constructing a Steiner triple system whose independence number is restricted with a lower bound

Input: an integer n congruent to 1 or 3 mod 6 and an integer k

Output: a set of triples $\mathcal{T}$

1: Form a set $V = \{v_0, v_1, v_2, ..., v_{n-1}\}$ containing n points

2: Initialize an empty set $\mathcal{T}$

3: For each pair $\{v_i, v_j\}$ where $0 \leq i < j < k$, construct a triple $\{v_i, v_j, v_{(j+i)\%(n-k)+k}\}$

4: Run the original hill-climbing algorithm on the remaining pairs. Whenever encounter a case that breaking a triple which contains points v_i and v_j where $i, j < k$ is needed, skip this iteration and reselect a pair to deal with.

5: When the total amount of triples is $\frac{n*(n-1)}{6}$, return $\mathcal{T}$

Let k be an integer which satisfies $0 < k \leq \beta_{max}(n)$.

Theorem 1.1: Step 3 is able to construct $\frac{k*(k-1)}{2}$ triples and any two of them have at most one point in common.

Proof. For two pairs $\{v_i, v_j\}$ and $\{v_a, v_b\}$ where $i = a$ and $j \neq b$, $i + j \neq a + b$ and $|b - j| < k$. For the case that $i \neq a$ and $j \neq b$, the theorem is obviously held.

Theorem 1.2: Any pair of $\{v_i, v_j\}$ where $0 \leq i < j < k$ is within a triple with a point whose index is greater than or equal to k.

Proof. $(j + i)\%(n - k) \geq 0$, so $(j + i)\%(n - k) + k \geq k$

Theorem 1.3: Algorithm 2 yields a STS whose independence number is not less than k

Proof. Since any three points whose indexes are less than k are not in a triple, the set $\{v_0, v_1, v_2, ..., v_{k-1}\}$ is an independent set.

This algorithm is not always able to terminate and return a STS. But as the experiments we did, the case that the algorithm falls into an endless loop rarely happens.

2.3 Algorithm for constructing a Steiner triple system using an affine plane

In this section, we introduce an algorithm for constructing a STS using an affine plane. The idea was presented by De Brandes and Rödl [13] which is used to establish the bounds for the minimum independence number. So, we suspect that the independence number of the system constructed by this algorithm might have an independence number that is close to the minimum independence number.

An *affine plane* is a system of points and lines that satisfy the following axioms:

1. Any two distinct points lie on a unique line.

2. Each line has at least two points.

3. Given any line and any point not on that line there is a unique line which

contains the point and does not meet the given line.

4. Given a point and a line, there is a unique line which contains the point and is parallel to the line.

A *finite affine plane* of order n satisfies the following conditions:

1. Each line contains n points

2. Each point is contained in $n + 1$ lines

3. There are n^2 points in all

4. There is a total of $n^2 + n$ lines

Next, we will illustrate an algorithm for constructing an affine plane of prime order.

Algorithm 3 Algorithm for constructing an affine plane of prime order

Input: a prime number n

Output: a collection of sets of size n (that is $n^2 + n$ lines)

1: Maintain an empty set $\mathcal{L}$

2: Construct two tables of operations. Both of them are of size $n * n$ One is for addition and the other is for multiplication. For the addition table, the element on (i, j) where $0 \leq i, j < n$ is $(i + j)\%n$. For the multiplication table, the element on (i, j) where $0 \leq i, j < n$ is $(i * j)\%n$.

3: For each pair (a, b) where $0 \leq a, b < n$

4: Initialize an empty set L

5: For each x in range $[0, n-1]$

6: Compute $y = a * x + b$ using the operations predefined and store $x * n + y$ (using normal addition and multiplication) to L

7: add L to $\mathcal{L}$

8: For each x in range $[0, n-1]$

9: Initialize an empty set L

10: For each y in range $[0, n-1]$

11: store $x * n + y$ to L

12: add L to $\mathcal{L}$

13: Return $\mathcal{L}$

Theorem 1.4: Algorithm 3 is able to construct $n^2 + n$ lines.

Proof. There are n^2 different pairs as coefficients. And the Step 8-12 forms n lines. Thus, the algorithm constructs $n^2 + n$ lines in total.

Theorem 1.5: Any two lines constructed by **Algorithm 3** have at most one point in common.

Proof. The addition table has no repeated element in each row or column. Therefore, any two lines of the form $y = a * x + b$ are parallel or intersect on one point. (An idea of a formal proof could be found in [14])

Next, we will introduce how to construct a $STS(n^2)$ using an affine plane of order

n and an $STS(n)$.

Algorithm 4 Algorithm for constructing a $STS(n^2)$ using an affine plane of order n and an $STS(n)$.

Input: an affine plane A of order n and an $STS(n), S_0$ where $n \equiv 1$ or 3 mod 6

Output: a set of triples $\mathcal{T}$ for an $STS(n^2)$, S

1: Initialize an empty set $\mathcal{T}$

2: Sort each triple in S_0

3: For each line L in A

4: Sort L

5: For each triple T in S_0

6: Form a triple $P = \{L[T[0]], L[T[1]], L[T[2]]\}$ and add this triple to $\mathcal{T}$

7: Return $\mathcal{T}$

Theorem 1.6: $\mathcal{T}$ contains $\frac{n^2*(n^2-1)}{6}$ triples.

Proof. With Step 3, 4, 5, 6, each line forms $\frac{n*(n-1)}{6}$ triples. There are n^2+n lines in total. $\frac{n*(n-1)}{6} * (n^2+n) = \frac{n^2*(n^2-1)}{6}$

Theorem 1.7: $\mathcal{T}$ is a valid collection of triples for an $STS(n^2)$.

Proof. By the definition of affine plane, each pair of points appears in at most one line. With Step 3, 4, 5, 6 and **Theorem 1.6**, each pair of points in the affine plane appears in exactly one triple in the resultant collection $\mathcal{T}$.

Hence, running **Algorithm 3** and **Algorithm 4** is able to construct an affine plane of prime order p (except 1 and 3) and then construct a Steiner triple system of

order p^2. However, to construct an affine plane of order which is not a prime number is more complicated.

In mathematics, a *finite field* or *Galois field* is a field that contains a finite number of elements. As with any field, a finite field is a set on which the operations of multiplication, addition, subtraction and division are defined and satisfy certain basic rules. The tables that **Algorithm 3** forms are examples of finite fields.

We will briefly introduce how to construct a non-prime field. Before heading into the method, we will first introduce an essential terminology.

In mathematics, an *irreducible polynomial* is a polynomial that cannot be factored into the product of two non-constant polynomials.

Given a prime power $q = p^n$ with p prime and $n > 1$, the field $GF(q)$ can be constructed in the following way. One first chooses an irreducible polynomial P in $GF(p)[X]$ of degree n. Then the quotient ring $GF(q) = GF(p)[X]/(P)$ of the polynomial ring $GF(p)[X]$ by the ideal generated by P is a field of order p.

Similarly, an affine plane of non-prime order n can also be used to construct a Steiner triple system. If $n \equiv 1$ or 3 mod 6, then it is a valid input for **Algorithm 4**.

One might be curious about if there is a way to construct a Steiner triple system whose order is less than n^2 using an affine plane of order n. The answer is yes. Take affine planes of order 9 as example. The following method is capable to construct a $STS(63)$.

1. Randomly choose two disjoint lines in the affine plane and delete the 18 points such that there are 7 lines with 9 points and 81 lines with 7 points.

2. Construct a $STS(9)$ and a $STS(7)$ using the hill-climbing algorithm

3. Copy the $STS(9)$ to each of the 7 lines with 9 points and copy the $STS(7)$ to the 81 lines with 7 points.

4. Merge the resultant triples

With these steps, one can obtain a $STS(63)$

The general algorithm is as follows.

Algorithm 5 Algorithm for using an affine plane to construct an STS with order less than the square of the order of the affine plane

Input: two prime powers a, b congruent to 1 or 3 mod 6 where $a > b$

Output: a set of triples for a $STS(a * b)$

1: Initialize an empty set $\mathcal{T}$

2: Construct an affine plane A of order a

3: Delete all the points in $a - b$ disjoint lines in A to get b lines of size a and a^2 lines of size b

4: Place an $STS(a)$ on each lines of size a and an $STS(b)$ on each lines of size b to form triples and add to $\mathcal{T}$.

5: Return $\mathcal{T}$

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

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

Figure 8: The finite fields of addition and multiplication for order 7

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

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

Figure 9: The finite fields of addition and multiplication for order 9

The independence number of STS(49) constructed by the hill-climbing algorithm	The independence number of STS(49) constructed with the use of an affine plane of order 7
16	16
17	16
17	16
17	17
16	17
17	17
17	17
16	16
17	16
17	16

Figure 10: A comparison between the independence numbers of STS(49)s constructed by the hill-climbing method and the independence numbers of STS(49)s constructed with the use of affine planes of order 7

The independence number of STS(63) constructed by the hill-climbing algorithm	20	20	20	20	20
The independence number of STS(63) constructed with the use of an affine plane of order 9	22	20	20	20	20

Figure 11: A comparison between the independence numbers of STS(63)s constructed by the hill-climbing method and the independence numbers of STS(63)s constructed with the use of affine planes of order 9

As you can observe from Figure 10 and Figure 11, it seems that the algorithm is not able to yield a system with an independence number that is close to the minimum independence number for order 49 and order 63. In our opinion, the reason is that the bound of the minimum independence number is an asymptotic bound. So it may not work on the small orders.

2.4 Algorithm for constructing a Steiner triple system with maximum independence number

This algorithm is also introduced in [5]

Algorithm 6 Algorithm for constructing an STS with maximum independence number

Input: an integer n congruent to 3 or 7 mod 12

Output: a set of triples for a $STS(n)$ with maximum independence number

1: Maintain a set $V = \{v_0, v_1, ..., v_{n-1}\}$

2: Construct an S_0=$STS(\frac{n-1}{2})$

3: Initialize an empty set $\mathcal{T}$

4: For each v_i where $\frac{n+1}{2} \leq i \leq n-1$

5: Form a triple $\{v_i, v_{\frac{n-1}{2}}, v_{i-\frac{n+1}{2}}\}$ and add to $\mathcal{T}$

6: For each v_i where $\frac{n+1}{2} \leq i \leq n-1$

7: Form a ordered set C of size $n-i-1$ where $C[j] = v_{(j+i-n)\%\frac{n-1}{2}}$

8: For each v_j where $i+1 \leq j \leq n-1$

9: Form a triple $\{C[(j-i)\%\frac{n-1}{2}], v_i, v_j\}$ and add to $\mathcal{T}$

10: Return $\mathcal{T}$

Theorem 1.8: Algorithm 6 is able to construct a STS of order n whose independence number is $\frac{n+1}{2}$

Proof. The proof that the resultant collection of triples is valid for an $STS(n)$ is similar to the proof of **Theorem 1.1**. Any two points in the set $\{v_{\frac{n-1}{2}}, v_{\frac{n+1}{2}}, ..., v_{n-1}\}$ are in a triple with a point in the set $\{v_0, v_1, ..., v_{\frac{n-3}{2}}\}$. Therefore, the set $\{v_{\frac{n-1}{2}}, v_{\frac{n+1}{2}}, ..., v_{n-1}\}$ is an independent set.

Below are the figures which show the independence numbers of Steiner triple systems constructed by **Algorithm 2**.

As you can see in Figure 12, when the lower bound is less than the value that the distribution concentrates on, the algorithm yields systems with the independence number equal to the concentrated value. However, when the lower bound is greater than the concentrated value, the algorithm is only able to yield systems with the independence exactly equal to the bound.

	order=39
Lower bound=11	14
Lower bound=12	14
Lower bound=13	14
Lower bound=14	14
Lower bound=15	15
Lower bound=16	16
Lower bound=17	17
Lower bound=18	18
Lower bound=19	19

	order=43
Lower bound=12	16
Lower bound=13	15
Lower bound=14	15
Lower bound=15	15
Lower bound=16	16
Lower bound=17	17
Lower bound=18	18
Lower bound=19	19
Lower bound=20	20
Lower bound=21	21

	order=55
Lower bound=15	18
Lower bound=16	18
Lower bound=17	18
Lower bound=18	18
Lower bound=19	19
Lower bound=20	20
Lower bound=21	21
Lower bound=22	22
Lower bound=23	23
Lower bound=24	24
Lower bound=25	25
Lower bound=26	26
Lower bound=27	27
Lower bound=28	28

Figure 12: The independence numbers of Steiner triple systems constructed by Algorithm 2

This phenomenon suggests that Steiner triple systems of a specific order tend to have independence numbers around the mid-range value. But we cannot claim that the amount of Steiner triple systems with mid-range independence number are significantly more than the amount of systems with lower or higher independence

number since the hill-climbing method does not guarantee to construct every distinct system with same probability.

From the experiments we did on systems of order 51, the set of independence numbers of this order should include $\{17, 18, 19, 20, 21, 22, 23, 24, 25, 26\}$.

For order 55, the set of independence numbers of this order should include $\{18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28\}$.

Chapter 3

A MODIFIED GREEDY ALGORITHM FOR FINDING MAXIMAL INDEPENDENT SETS AND A POST-OPTIMIZATION ALGORITHM

3.1 Current approaches

In this section, we will introduce two current approaches for finding a maximal independent set in an arbitrary Steiner triple system. The first one is a naive greedy method. The other treats Steiner triple systems as hypergraphs and apply probabilistic method to solve the problem.

3.1.1 A Naive Greedy Method

This approach was established in a paper by Erdös and Hajnal [15]. Consider a Steiner triple system STS(n), denoted $S = (X, \mathcal{T})$. The steps of this algorithm is as follows.

Initialize $I = \emptyset$. Iteratively add points to I from X until there is no point available to be chosen (i.e., for any $x \in X \setminus I$, $\{x\} \cup I$ contains a triple from $\mathcal{T}$).

According to the definition, I is a maximal independent set. Hence, for each $x \in X \setminus I$, there exists at least one triple $T_x \in \mathcal{T}$ with $x \in T_x$ and $T_x \subset I \cup \{x\}$.

Because $T_x \nsubseteq I$, $|T_x \cap I| = 2$. More importantly, for any $x, y \in X \setminus I$, where $x \neq y$, we have $|T_x \cap T_y| \leq 1$. Then these two inferences imply that $|X \setminus I| = n - |I| \leq \binom{|I|}{2}$.

Therefore, we obtain an inequality $n - |I| \leq \frac{|I|*(|I|-1)}{2}$. Hence, we have $|I| \geq \lfloor \sqrt{2n} \rfloor$.

3.1.2 A Probabilistic Method

A *randomized algorithm* is an algorithm that employs a degree of randomness as part of its logic [16].

A *hypergraph* is a generalization of a graph in which an edge can join any number of vertices. A *k-uniform hypergraph* is a hypergraph such that all its hyperedges have size k.

In usual hypergraph terminology, a hypergraph is *uncrowded* iff it has no cycles of length 2,3 or 4.

The *hypergraph representation* of a Steiner triple system $S = (X, \mathcal{T})$ is the completely determined 3-uniform hypergraph $G = (X, \mathcal{T})$, where X is the vertex set and $\mathcal{T}$ is the hyperedge set of G.

The method introduced in [17] treats a Steiner triple system as a 3-uniform hypergraph and uses randomized algorithm to yield an uncrowded hypergraph from the original one in order to apply the efficient derandomized version [18] of Ajtai's algorithm [19] for obtaining independent sets in $k-$uniform uncrowded hypergraphs. The bound of the size of an independent set I obtained in time $O(n^2)$ by this method is

$$|I| \geq c\sqrt{n \ln n}$$

3.2 A modified greedy algorithm

A *greedy algorithm* is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. In many problems, a greedy algorithm does not usually produce an optimal solution, but nonetheless, a greedy heuristic may yield locally optimal solutions that approximate a globally optimal solution in a reasonable amount of time.

The algorithm we present is an iterative algorithm which chooses a point to add to the resultant set at each iteration.

Before presenting the algorithm, we will first give some definitions which are used in the algorithm description.

A point is *chosen* if it is in the resultant set.

An *available* point is a point that no triple contains it and any two chosen points.

An *unavailable* point is a point that there exists a triple which contains it and two chosen points.

An *active triple* is a triple $T \in \mathcal{T}$ such that all elements in T are available to choose.

A *semi-active triple* is a triple $T \in \mathcal{T}$ such that exactly one element in T is chosen and other two elements are available to choose.

An *inactive triple* is a triple $T \in \mathcal{T}$ such that two elements in T are chosen.

The algorithm can be described as follows.

Algorithm 7 Algorithm for finding a maximal independent set in a Steiner triple system by iteratively choosing the point which makes least points to be unavailable

Input: a Steiner Triple System $S = (X, \mathcal{T})$

Output: a maximal independent set I

1: Initialize $I = \emptyset$, $A = X$

2: **While** $A \neq \emptyset$

3: Choose an available point which is contained in minimum number of *semi-active triples*. Call this point a. If there is a tie, randomly pick one of candidates. (Candidates are the points which are contained in minimum number of *semi-active triples*)

4: $A = A \setminus (\{x | x \in A, x \text{ and } a \text{ are in a } \textit{semi-active triple}\} \cup \{a\})$

5: $I = I \cup \{a\}$

6: **return** I

Apparently, in each iteration, it excludes the points which are within triples with the newly chosen point and the points that have already been chosen. Hence, those points will never be chosen in the result. Moreover, the algorithm terminates when there is no available point to choose. Therefore, this algorithm is able to find a maximal independent set.

Let n be the order of the input Steiner triple system. The running time of this algorithm is $O(n^3)$ when it recomputes the minimum number of *semi-active triples* in each iteration.

A key observation is that in the first iteration, there is no point in the chosen set. Hence, adding any point would not make any point other than itself unavailable. In the second iteration, since there is only one point in the chosen set, then adding

any point would just make exactly one point unavailable. Therefore, in the first two iterations, we may want to apply an extra rule to determine which two points should be chosen.

An experiment that we have tried is to select the first two points based on their cycle structures.

According to the definition of Steiner triple system, two triples can have at most one element in common. So, for any two points in a Steiner triple system, the pairs in the triples that contain the two points (except the triple that contains both of the two points) form some cycles.

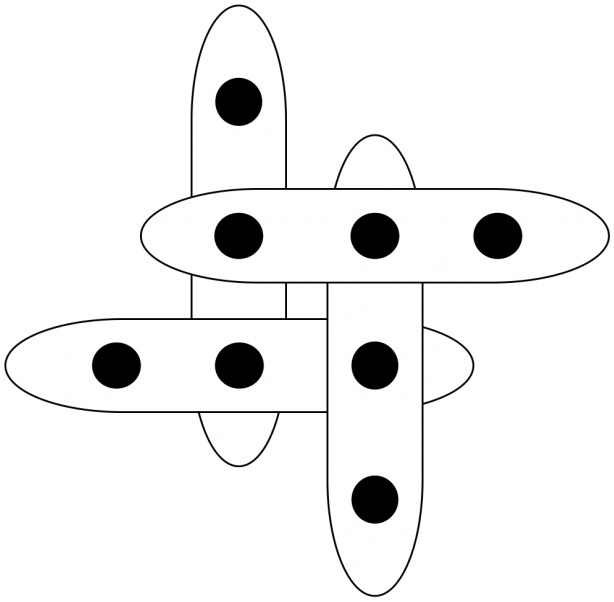

Figure 13: The 4-cycle

Theorem 2.1 The amount of pairs that a cycle contains must be even. [5]

Proof. Let a, b be two points in a Steiner triple system. Then a cycle formed by the pairs are within the triples which must alternatively include a or b but not both. Therefore, the amount of triples must be even.

Theorem 2.2 The minimum amount of pairs that a cycle contains is 4. [5]

Proof. Suppose, there is a cycle with two pairs. Then these two triples that contain the pairs must have two elements in common which contradicts the definition of Steiner triple system.

There is an interesting property of 4-cycles.

Suppose the two points we choose are i and j and there is a 4-cycle $\{\{i, v_1, v_2\}, \{j, v_2, v_3\}, \{i, v_3, v_4\}, \{j, v_4, v_1\}\}$.

At the third iteration of the algorithm, it may pick one point among $\{v_1, v_2, v_3, v_4\}$. Consider it picks v_1. Then v_2 and v_4 are going to be unavailable. So, for the next iteration, the candidate is v_3. The triples that contain it and a chosen point also contain v_2 and v_4. Hence, it will just make one more point be unavailable in this round. Therefore, in the next iteration, there are n-7 available points.

For the cycles with larger sizes, this case can not happen. The two triples, where one of them contains i and a newly added unavailable point and the other contains the j and another newly added unavailable point, do not share a common point so that it will always add two points to be unavailable in this round. Hence, in the next iteration, there are n-8 available points.

Hence, a 4-cycle provides a choice that results more available points in the fourth iteration.

Inspired by this observation, we did some experiments on the performance of this modified greedy approach where the first two points are selected based on the amount of 4-cycles.

Check Figure 14 and 15 for some experimental results. The optimal solutions are computed by IP solver. The average solutions of the modified greedy algorithm and the original greedy algorithm are computed by taking average of 100 runs.

Optimal solution	Average size computed by modified greedy algorithm (considering 4-cycles)	Average size computed by modified greedy algorithm (original)	Average size of original greedy algorithm
17	15.3	15.5	14.1
17	15.7	15.4	13.9
17	15.4	15.3	13.6
17	15.9	15.3	13.7
17	15.4	15.5	13.6
17	15.4	15.3	14.4
17	15.5	15.4	14.4
17	15.4	15.5	14.0
17	15.1	15.4	13.9
17	15.6	15.2	13.7
17	15.4	15.6	14.0
18	15.0	15.3	13.5
19	15.6	15.4	13.9
20	15.3	15.5	13.8
21	16.8	16.1	13.8
22	17.0	15.8	14.2
23	19.7	15.8	14.1
24	15.4	15.8	14.0
25	22.6	17.8	14.2
26	16.7	17.4	13.8

Figure 14: A comparison on the performances of modified greedy algorithm and the original one on STS(51)

The modified greedy algorithm appears to be able to find nearly optimal solution for the systems whose independence number is relatively low. But for the systems with higher independence number, this improved greedy algorithm may not be capable

Optimal solution	Average size computed by modified greedy algorithm (considering 4-cycles)	Average size computed by modified greedy algorithm (original)	Average size of original greedy algorithm
18	16.6	16.4	14.6
18	16.2	16.2	14.9
18	16.6	16.6	14.8
18	16.2	16.5	15.3
19	16.2	16.7	14.4
18	16.2	16.4	14.5
18	16.6	16.0	14.8
18	16.0	16.4	14.7
18	16.5	16.5	14.9
18	15.7	16.1	14.7
19	16.2	16.1	14.9
20	16.1	16.3	14.7
21	16.2	16.2	14.9
22	16.4	16.3	14.3
23	16.0	16.0	14.7
24	18.0	16.6	14.4
25	17.7	16.6	14.2
26	17.2	18.6	14.8
27	21.7	17.7	15.6
28	18.4	17.3	14.7

Figure 15: A comparison on the performances of modified greedy algorithm and the original one on STS(55)

to yield a solution close to the optimal one. For all the test cases that we tried, the modified greedy algorithm yields better solutions than the original greedy algorithm.

3.3 A post-optimization algorithm

As you can observe from the experimental results of the performance of the modified greedy algorithm, the computed values are only 1 or 2 less than the optimal solution in some cases. So we figured out an exchange method to find improvement for the results from the modified greedy algorithm.

Before presenting the algorithm, we will first give some definitions which are used

in the algorithm description.

Denote I is the resultant independent set by the modified greedy algorithm.

A point v_i is called a *candidate* if there is only one triple $T_i = \{v_a, v_b, v_i\}$ where $v_a, v_b \in I$.

The post-optimization algorithm is as follows.

Algorithm 8 Algorithm for finding candidates for post-optimization

Input: a Steiner Triple System $S = (X, \mathcal{T})$ and an independent set I

Output: a collection E of 3-tuples of the form $T = (v_{c_i}, v_{c_j}, v)$ where v_{c_i}, v_{c_j} are the two points to swap in and v is the point in I to swap out.

1: Initialize an empty set E

2: Find *candidate* points and add the candidate points to a set C

3: For each pair of points $\{v_{c_i}, v_{c_j}\}$ where $v_{c_i}, v_{c_j} \in C$

4: If $T_{c_i} \cap T_{c_j} \neq \emptyset$, add (v_{c_i}, v_{c_j}, v) to E where $v \in T_{c_i} \cap T_{c_j}$

5: return E

The time complexity of **Algorithm 8** is $O(|I|^2)$, since there are $|I| * (|I| - 1)/2$ triples in total to check. For **Step 3,4**, the total amount of the corresponding triples is less than or equal to $|I| * (|I| - 1)/2$. Therefore, **Algorithm 8** is able to terminate in $O(|I|^2)$ time.

Algorithm 9 The post-optimization algorithm

Input: a Steiner Triple System $S = (X, \mathcal{T})$ and an independent set I

Output: a modified independent set I

1: Run **Algorithm 8** to obtain a set E of 3-tuples.

2: Randomly choose one tuple $T \in E$ and swap in $T[0], T[1]$ to I and swap out $T[2]$ from I.

3: If there exists a point $v_i \in I$ such that $\{T[0], T[1], v_i\} \in \mathcal{T}$, remove v_i from I.

4: Keep doing **Step 1,2,3** until the set E from **Step 1** is empty.

5: return I

One can check the following example to understand the algorithm.

The set of triples of an $STS(19)$ is as follows.

{{0, 1, 16}, {0, 2, 3}, {0, 4, 11}, {0, 5, 9}, {0, 6, 8}, {0, 7, 18}, {0, 10, 12}, {0, 13, 15}, {0, 14, 17}, {1, 2, 18}, {1, 3, 7}, {1, 4, 5}, {1, 6, 15}, {1, 8, 11}, {1, 9, 12}, {1, 10, 17}, {1, 13, 14}, {2, 4, 16}, {2, 5, 10}, {2, 6, 14}, {2, 7, 15}, {2, 8, 12}, {2, 9, 13}, {2, 11, 17}, {3, 4, 12}, {3, 5, 6}, {3, 8, 17}, {3, 9, 15}, {3, 10, 13}, {3, 11, 18}, {3, 14, 16}, {4, 6, 13}, {4, 7, 10}, {4, 8, 15}, {4, 9, 17}, {4, 14, 18}, {5, 7, 8}, {5, 11, 16}, {5, 12, 14}, {5, 13, 17}, {5, 15, 18}, {6, 7, 17}, {6, 9, 16}, {6, 10, 18}, {6, 11, 12}, {7, 9, 14}, {7, 11, 13}, {7, 12, 16}, {8, 9, 18}, {8, 10, 14}, {8, 13, 16}, {9, 10, 11}, {10, 15, 16}, {11, 14, 15}, {12, 13, 18}, {12, 15, 17}, {16, 17, 18}}

The independent set computed by the modified greedy algorithm is $\{12, 11, 18, 10, 7, 17, 8\}$. A tuple for doing the swap is $(4, 5, 7)$ where 4 and 5 are candidate points, because $\{7\} = \{4, 7, 10\} \cap \{5, 7, 8\}$. Since $\{4, 5\} \subseteq \{1, 4, 5\}$ and 1 is not contained in the independent set, 4 and 5 are eligible to be swapped in when 7 is swapped out. Hence, the improved independent set is $\{12, 11, 18, 10, 17, 8, 4, 5\}$.

Theorem 2.3 The resultant set after making exchange in each iteration is still an independent set

Proof. Each point to swap in is only contained in one triple with two chosen points. Also the two corresponding triples for the two points to swap in share one point which is exactly the point to swap out. Moreover, since the point in the independent set which is contained in a triple containing both two points to swap in will also be removed according to the algorithm, the resultant set is always an independent set.

Theorem 2.4 The post-optimization algorithm is able to yield an independent set whose size is no less than the original resultant independent set.

Proof. In each iteration, the post-optimization algorithm either swaps in 2 points and remove 1 point from the independent set or swaps in 2 points and remove 2 points from the independent set. Hence, the size of the independent set is never decreased.

One can check Figure 16, 17, 18 for the performance of this post-optimization algorithm. The Steiner triple systems are generated by the hill-climbing algorithm. The optimal solutions are computed by IP solver.

As you can see in the figures, this post-optimization algorithm is not always able to find an increment to the original resultant independent set computed by the modified greedy algorithm. Some more alternative approaches are raised in the following chapter.

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=37	14	12	12
order=37	14	14	14
order=37	14	12	13
order=37	14	13	13
order=37	14	12	13
order=37	14	13	13
order=37	14	13	13

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=39	14	13	13
order=39	14	12	12
order=39	14	13	13
order=39	14	14	14
order=39	15	12	13
order=39	14	12	14
order=39	14	13	13

Figure 16: The performances of the post-optimization algorithm (1)

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=43	15	13	14
order=43	16	14	14
order=43	15	14	14
order=43	15	14	14
order=43	15	13	13
order=43	15	15	15
order=43	15	14	14

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=45	16	14	14
order=45	16	14	14
order=45	16	15	15
order=45	16	14	15
order=45	16	14	14
order=45	16	15	15
order=45	15	15	15

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=49	17	15	16
order=49	16	15	16
order=49	17	14	15
order=49	17	16	16
order=49	16	15	15
order=49	17	15	16
order=49	17	15	15

Figure 17: The performances of the post-optimization algorithm (2)

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=51	17	16	16
order=51	17	15	16
order=51	17	16	16
order=51	18	17	17
order=51	17	16	16
order=51	17	16	16
order=51	17	16	16

	Optimal solution	Original result by modified greedy algorithm	Result by the post-optimization algorithm
order=55	18	17	17
order=55	18	17	17
order=55	18	16	16
order=55	18	16	16
order=55	18	16	17
order=55	18	16	16
order=55	19	17	17

Figure 18: The performances of the post-optimization algorithm (3)

Chapter 4

SUMMARY AND FUTURE WORKS

4.1 Summary

In this thesis, we first introduce the traditional hill-climbing algorithm to construct a random Steiner triple system. Observing from the distributions of independence numbers of systems constructed by the hill-climbing algorithm, it appears that the distributions concentrate only on one or two mid-range values, so we present an algorithm for constructing systems with independence numbers restricted by a lower bound. Furthermore, we also provide algorithms for constructing a STS using an affine plane and an efficient algorithm for constructing an affine plane of prime order.

Next, we illustrate an improved greedy algorithm to find approximately maximum independent set and an exchange algorithm to do potential post-optimization. The modified greedy algorithm appears to be able to find nearly optimal solution for the systems whose independence number is around the relatively lower value. But for the systems with higher independence number, this modified greedy algorithm may not be capable to yield a solution close to the optimal one. The post-optimization algorithm can sometime find an increment to the independent returned by the improved greedy algorithm.

4.2 Future works

4.2.1 Some more strategies for the improvement of the post-optimization

Instead of iteratively searching for a 2 for 1 or 2 for 2 swap, it is also worthy to consider about a general method to find a swap so that m points are swapped out and n points are swapped in where $m < n$.

4.2.2 Applying the Algorithm for Finding a Maximum Independent Set in a Bipartite Graph

Consider a situation that some points have been chosen into the potential independent set. Denote the set as I.

Then, $E = \{\{u, v\} | \{u, v, x\} \in \mathcal{T}$ where $x \in I$ and $u, v \notin I\}$ is a set of pairs of remaining available points. By the definition of independent set of Steiner triple system, for each $\{u, v\} \in E$, at most one point can be selected. (There might be a case that some triples where none of the points inside them has been chosen)

Hence, the problem at the ideal circumstance becomes a problem of finding a maximum independent set in a bipartite graph.

Consider a bipartite graph $G = (A \cup B, E)$.

Let $\nu(G)$ denote the minimum cardinality of a vertex cover in G and $\gamma(G)$ denote the maximum cardinality of a matching in G.

Consider a graph $G' = (A \cup B \cup \{s,t\}, E \cup \{\{s,a\}|a \in A\}) \cup \{\{b,t\}|b \in B\}$ with unit edge capacities.

It is easy to notice that $\gamma(G)$ is the maximum number of internally disjoint s-t-path and $\nu(G)$ is the minimum number of vertices whose deletion disconnect s and t. Menger proved that $\gamma(G) = \nu(G)$ in 1927 [20]. $\gamma(G)$ is also equal to the value of maximum flow from s to t. Thus, we can find a vertex cover in a bipartite graph with minimum cardinality in polynomial time.

Denote the vertex cover of minimum cardinality as C. Since, an independent set with maximum cardinality in G is $(A \cup B) \setminus C$, then we can certainly obtain an independent set with maximum cardinality in a bipartite graph in polynomial time.

Therefore, if we could find an effective method to select some points in advance such that the resulting graph is a bipartite graph, we would have a polynomial time algorithm for solving the remaining problem. If there seems no efficient way to reduce the original one to a bipartite graph, we can also try to firstly ignore some triples and rule out some points after obtaining a solution from the bipartite graph with the restriction of those ignored triples.

The accuracy of the result yielded by this idea is also worthy to investigate.

4.2.3 The NP-completeness of Finding Independent Set in Steiner Triple System

The decision variant of the problem which is to find a maximum independent set

in a Steiner triple system is that given a Steiner triple system $S = (X, \mathcal{T})$ and an integer k, is there an independent set $I \subseteq X$ such that $|I| \geq k$. Call the decision variant F.

An NP-complete problem is believed that there is no polynomial time algorithm for solving it. To prove the NP-completeness of F, we need to show that

a) F is in NP

b) An NP-complete problem is polynomial time reducible to F [21]

Showing a) is easy. Given a potential solution, a set of vertices I, we can verify if it is a correct solution by checking

1. $I \subseteq X$

2. $|I| \geq k$

3. there does not exist a triple $T \in \mathcal{T}$ such that $T \subseteq I$

This process can be done in polynomial time. Hence, the problem is in NP.

It would be of interest to figure out a reduction to prove b) and establish the NP-completeness of this problem.

REFERENCES

[1] D. Stinson, *Combinatorial designs: constructions and analysis*. Springer Science & Business Media, 2007.

[2] R. C. Bose, "On the construction of balanced incomplete block designs," *Annals of Eugenics*, vol. 9, no. 4, pp. 353–399, 1939.

[3] V. E. Alekseev, "On the number of steiner triple systems," *Mathematical notes of the Academy of Sciences of the USSR*, vol. 15, no. 5, pp. 461–464, 1974.

[4] J. Doyen and G. Valette, "On the number of non isomorphic steiner triple systems," *Mathematische Zeitschrift*, vol. 120, no. 2, pp. 178–192, 1971.

[5] C. J. Colbourn and A. Rosa, *Triple systems*. Oxford University Press, London, 2001.

[6] N. Sauer and J. Schönheim, "Maximal subsets of a given set having no triple in common with a steiner triple system on the set," *Canadian Mathematical Bulletin*, vol. 12, no. 6, pp. 777–778, 1969.

[7] K. T. Phelps and V. Rödl, "Steiner triple systems with minimum independence number," *Ars Combin*, vol. 21, pp. 167–172, 1986.

[8] H. Dau and O. Milenkovic, "Maxminsum steiner systems for access balancing in distributed storage," *SIAM J. Discret. Math.*, vol. 32, no. 3, pp. 1644–1671, 2018.

[9] Y. M. Chee, C. J. Colbourn, H. Dau, R. Gabrys, A. C. H. Ling, D. Lusi, and O. Milenkovic, "Access balancing in storage systems by labeling partial steiner systems," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 566–570.

[10] D. Stinson, "Hill-climbing algerithms for the construction of combinatorial designs," in *Annals of Discrete Mathematics (26): Algorithms in Combinatorial Design Theory*, ser. North-Holland Mathematics Studies, C. Colbourn and M. Colbourn, Eds., vol. 114, North-Holland, 1985, pp. 321–334.

[11] D. Kreher and D. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*. CRC Press, 2020.

[12] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, ser. Dover Books on Computer Science. Dover Publications, 2013.

[13] M. De Brandes and V. Rödl, "Steiner triple systems with small maximal independent sets," *Ars Combin*, vol. 17, pp. 15–19, 1984.

[14] P. Dembowski, *Finite Geometries: Reprint of the 1968 Edition*, ser. Classics in Mathematics. Springer Berlin Heidelberg, 2012.

[15] P. Erdös, A. Hajnal, and B. Rothschild, "On chromatic number of graphs and set systems," in. Nov. 2006, vol. 17, pp. 531–538.

[16] P. Raghavan, "Probabilistic construction of deterministic algorithms: Approximating packing integer programs," *J. Comput. Syst. Sci.*, vol. 37, no. 2, pp. 130–143, 1988.

[17] C. Bertram-Kretzberg and H. Lefmann, "The algorithmic aspects of uncrowded hypergraphs," in *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '97, New Orleans, Louisiana, USA: Society for Industrial and Applied Mathematics, 1997, pp. 296–304.

[18] A. D. Fundia, "Derandomizing chebyshev's inequality to find independent sets in uncrowded hypergraphs," *Random Struct. Algorithms*, vol. 8, no. 2, pp. 131–147, 1996.

[19] M. Ajtai, J. Komlós, J. Pintz, J. Spencer, and E. Szemerédi, "Extremal uncrowded hypergraphs," *J. Comb. Theory, Ser. A*, vol. 32, no. 3, pp. 321–335, 1982.

[20] K. Menger, "Zur allgemeinen kurventheorie," ger, *Fundamenta Mathematicae*, vol. 10, no. 1, pp. 96–115, 1927.

[21] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, M. A. Harrison, R. B. Banerji, and J. D. Ullman, Eds., ACM, 1971, pp. 151–158.