Aspects of Methods for Constructing Random Steiner Triple Systems

Daniel Heinlein* and Patric R. J. Östergård
Department of Communications and Networking
Aalto University School of Electrical Engineering
P.O. Box 15400, 00076 Aalto, Finland
{daniel.heinlein,patric.ostergard}@aalto.fi

Abstract

Several methods for generating random Steiner triple systems (STSs) have been proposed in the literature, such as Cameron’s algorithm and Stinson’s hill-climbing algorithm, but these are not yet completely understood. Those algorithms, as well as some variants, are here assessed for STSs of both small and large orders. For large orders, the number of occurrences of certain configurations in the constructed STSs are compared with the corresponding expected values of random hypergraphs. Modifications of Stinson’s algorithm are proposed.

Keywords: configuration, randomized algorithm, Steiner triple system, subsystem
MSC: 05B07

1 Introduction

Random generation of combinatorial structures is a problem that has been extensively studied both from a theoretical [12, 13] and a practical [1, 8, 20] point of view. When developing practical algorithms for random generation of structures with tight constraints, it can be far from clear how amenable central measures such as computation time and uniformity are to formal study. For the smallest parameters, structures are completely understood, but both the algorithms and the distribution of structures can behave differently for small

*Supported by the Academy of Finland, Grant 331044.
parameters than for large ones. On the other hand, for large parameters where the algorithms are truly needed, it is challenging to evaluate the performance.

Steiner triple systems form one class of structures for which practical random generation algorithms have been developed but where an evaluation for large parameters is lacking, primarily due to the difficulty of judging performance. In this experimental study, we start by considering the case of Steiner triple systems with small orders and carry out evaluations of known algorithms, including some variants. For larger orders, we bring together random generation algorithms and a conjecture on substructures of Steiner triple systems obtained using a random model to arrive at evidence supporting quality of both ingredients.

The paper is organized as follows. In Section 2 Steiner triple systems and related objects and concepts are defined, and the topic of generating random Steiner triple systems is introduced. In Section 3 Cameron’s algorithm and Stinson’s hill-climbing algorithm, as well as old and new variants, are described. Finally, computational experiments are carried out and analyzed in Section 4.

2 Preliminaries

2.1 Steiner Triple Systems and Configurations

A Steiner triple system (STS) is a pair \((V, B)\), where \(V\) is a set of points and \(B\) is a set of 3-subsets of points, called blocks, such that every 2-subset of points occurs in exactly one block. The size of the point set is the order of the Steiner triple system, and a Steiner triple system of order \(v\) is denoted by \(\text{STS}(v)\). An \(\text{STS}(v)\) exists iff

\[ v \equiv 1 \text{ or } 3 \pmod{6}. \]

Direct calculation gives that an \(\text{STS}(v)\) has \(v(v - 1)/6\) blocks and each point is in \((v - 1)/2\) blocks. More information about Steiner triple systems can be found in [1, 5].

A transversal design \(\text{TD}(k, n)\) is a triple \((V, \mathcal{G}, \mathcal{B})\), where \(V\) is a set of \(kn\) elements; \(\mathcal{G}\) is a partition of \(V\) into \(k\) \(n\)-subsets (called groups); \(\mathcal{B}\) is a collection of \(k\)-subsets of \(V\) (called blocks); and every 2-subset of \(V\) is contained either in exactly one group or in exactly one block, but not both.

A configuration is a pair \((V', B')\), where \(V'\) is a set of points and \(B'\) is a collection of subsets of points, typically called lines, such that for any 2-subset of points, there is at most one line passing through them.
A configuration with the property that each point in $V'$ occurs in at least two lines is said to be full. A configuration with $w$ points, $w$ lines, and each line containing $k$ points (which further implies that each point lies on $k$ lines) is called a $w_k$ configuration. Throughout the paper, we assume line size 3. In the context of building up an STS block by block, the configuration consisting of the blocks that have been obtained so far is called a partial STS. (And it is possible that a partial STSs cannot be completed to an STS.)

The configurations considered in this paper are precisely the full $n$-line configurations with $n \leq 6$ and the $w_3$ configurations with $w \leq 8$, all of which are depicted in Figure 1.

![Configurations](image-url)

**Figure 1: Configurations**

An isomorphism between two STSs is a bijection between the point sets that preserves incidences. An isomorphism of an STS onto itself is
an automorphism, and the set of all automorphisms of an STS forms a group under composition. The concepts of isomorphism, automorphism, and automorphism group are similarly defined for configurations.

A switch or a trade can be used to modify a Steiner triple system into another Steiner triple system of the same order [18]. A cycle switch of a Steiner triple system is defined as follows. Suppose that an STS contains the blocks

$$T_1 = \{ax_1x_2, bx_2x_3, ax_3x_4, bx_4x_5, \ldots, bx_nx_1\}. \quad (1)$$

Then these blocks can be replaced by the blocks

$$T_2 = \{bx_1x_2, ax_2x_3, bx_3x_4, ax_4x_5, \ldots, ax_nx_1\}.$$

to get another STS.

For fixed and distinct points $a$ and $b$, and $c$ chosen so that $B = \{a, b, c\} \in B$, the points $V \setminus B$ can in a unique way be partitioned into sets of points that occur together with $a$ and $b$ in sets of type $T_1$. The name cycle switch comes from the fact that removal of the points $a$ and $b$ from the blocks in $T_1$ gives edges that form a cycle in a graph with vertex set $\{x_1, x_2, \ldots, x_n\}$.

The smallest possible size of $T_1$ and $T_2$ is 4, and such a switch is called a Pasch switch as $T_1$ and $T_2$ are then Pasch configurations.

A perfect STS($v$) is a Steiner triple system, in which each possible cycle switch involves $|T_1| = |T_2| = v - 3$ blocks. Perfect Steiner triple systems indeed exist; see, for example, [7].

When carrying out cycle switching for partial STSs, one also encounters situations where the two points $a$ and $b$ do not induce a 2-regular graph with vertex set $V \setminus \{a, b, c\}$, but induces a graph with vertex degrees at most 2. Such a graph consists of cycles and paths, and switches for paths are analogous to those for cycles. For example, the switch for the shortest possible path (of length 1) is simply

$$\{ax_1x_2\} \rightarrow \{bx_1x_2\}. \quad (2)$$

2.2 A Hypergraph Model

A central ingredient in our assessment of algorithms is a hypergraph model. Following the work of [16, 9], we define a random 3-uniform hypergraph on $v$ vertices such that the probability of each hyperedge is $p := 1/(v - 2)$, which is the number of blocks in an STS($v$) divided by the number of 3-subsets of the set of $v$ vertices.

In assessing algorithms for constructing random Steiner triple systems, we will for large orders focus on numerical values related to properties of
the systems. For comparison, the related values can also be determined for the hypergraph model. We argue that if the computational results and the results from the hypergraph model approach each other asymptotically, this is evidence for quality of the algorithms as well as the model (with respect to the studied properties, but perhaps even beyond those).

The properties of STSs to be considered are specifically the numbers of occurrences of configurations. For a given configuration with \( w \) points, \( b \) lines, and automorphism group order \( |\Gamma| \), there are \( w!/|\Gamma| \) labelled specimens on \( w \) points and consequently \( w\binom{v}{w}/|\Gamma| \) labelled specimens on \( v \) points. The probability for a specific specimen to occur in the model of random hypergraphs is precisely \( p^b \) so, by linearity of expectation, the expected number of occurrences of the configuration is

\[
\frac{w\binom{v}{w}}{|\Gamma|(v-2)^b} = \frac{v(v-1)\ldots(v-w+1)}{|\Gamma|(v-2)^b} \sim \frac{v^{w-b}}{|\Gamma|}.
\]

(3)

For the configurations in Figure 1, we list in Table 1 the number of lines \( b \), the number of points \( w \), the order of the automorphism group \( |\Gamma| \), and the right-hand side of (3).

| Name       | \( b \) | \( w \) | \( |\Gamma| \) | asymptotic behavior   |
|------------|--------|--------|-------------|----------------------|
| Pasch      | 4      | 6      | 24          | \( v^2/24 \)         |
| Mitre      | 5      | 7      | 12          | \( v^3/12 \)         |
| Fano–line  | 6      | 7      | 24          | \( v/24 \)           |
| Crown      | 6      | 8      | 2           | \( v^2/2 \)          |
| Hexagon    | 6      | 8      | 12          | \( v^3/12 \)         |
| Prism      | 6      | 9      | 12          | \( v^3/72 \)         |
| Grid       | 6      | 9      | 72          | \( v^3/72 \)         |
| Fano       | 7      | 7      | 168         | \( 1/168 \)          |
| Möbius–Kantor | 8   | 8      | 48          | \( 1/48 \)           |

Table 1: Configuration data

### 3 Algorithms for Generating Random STSs

We shall here have a look at Cameron’s algorithm and Stinson’s algorithm. A key difference between these is that an STS is constructed from scratch in Stinson’s algorithm, whereas Cameron’s algorithm repeatedly perturbs a structure that is an STS or nearly an STS (in a way that will become clear later). Only for Stinson’s algorithm do we know that every STS can be
reached with a non-zero probability. Cameron’s algorithm, on the other hand, in practice seems to produce STSs with a distribution closer to the desired uniform distribution.

### 3.1 Random Steiner Triple Systems

We would like to develop an algorithm that constructs random Steiner triple system of a given order, with uniform distribution over all (labelled) systems of that order. If the Steiner triple systems of the given order have been classified, then there is an obvious solution. Pick a random isomorphism class, where the probability of choosing a class is given by the proportion of STSs in that class, and then apply a random permutation to the point set of the chosen isomorphism class representative. Since Steiner triple systems have been classified for all orders up to 19, this approach does not work for orders greater or equal to 21. Moreover, a rather huge data structure is needed for the more than 11 billion isomorphism classes of STS(19) [14]. (The problem of picking an isomorphism class uniformly at random is obviously straightforward for these parameters.)

What about orders greater than 19? In theory, the problem can be solved for arbitrary orders: the classification problem can be solved in finite time, whereafter the above mentioned approach can be used. Another solution, with finite expected time but infinite worst-case time, is as follows. Take a random $b$-subset of blocks and check whether these form an STS. If not, repeat the procedure. (For the problem of picking isomorphism classes uniformly at random, accept an STS that has been found with the probability given by the size of the automorphism group of the STS divided by a fixed constant that is an upper bound on the size of the automorphism groups of the STSs of the same order). Because of implementation and/or time issues, these methods have no practical importance.

We will next consider some methods that have been proposed for the construction of (random) Steiner triple systems. How to evaluate those? Of course, one could evaluate them for small cases—up to order 19—which are completely understood. Such work is done in [3]. But what if the behavior of an algorithm changes when going from small to larger orders? Moreover, the same question can be asked also for the STSs: What if there are changes in the distribution of STSs when going from small to larger orders?

### 3.2 Cameron’s Algorithm

Jacobson and Matthews [11] published an algorithm for random generation of Latin squares, and Cameron [1][2][3] shows that a similar approach is possible
also for Steiner triple systems. The main difference between the two settings is that stronger theoretical results have been proved for the Latin square algorithm.

The idea in Cameron’s algorithm is to apply perturbations repeatedly to get a sequence of structures, some of which are STSs and some of which are not (and are called proper and improper STSs, respectively). These structures can be defined as pairs \((V, f)\), where \(V\) is a set of points and \(f\) is a function that maps 3-subsets of points into the set \([-1, 0, 1]\) such that at most one 3-subset is mapped to \(-1\) and

\[
\sum_{z \in V \setminus \{x, y\}} f(\{x, y, z\}) = 1
\]

for all 2-subsets \(\{x, y\}\). We say that we have

- a proper STS if \(f(B) \neq -1\) for all 3-subsets \(B\),
- an improper STS if there is a unique 3-subset \(B'\) such that \(f(B') = -1\).

An STS as defined in Section 2 is a proper STS, where \(f\) is the characteristic function of the set of blocks. For an improper STS—which has \(f(B') = -1\)—the 3-subsets \(B\) for which \(f(B) = 1\) correspond to a triple system where each 2-subset of points occur in exactly one block, except for the 2-subsets of \(B'\), which occur in exactly two blocks.

The possible perturbations are described in Table 2 for proper and improper STSs. In each case, the candidates are given by the values of \(x, y, z, x', y',\) and \(z'\) that fulfill the conditions in line \(f\), and the new function values are given in line \(f'\). The total number of candidates is \(v(v - 1)(v - 3)/6\) for proper STSs and 8 for improper STSs \([1]\). One of these is chosen uniformly at random in the algorithm. The value of \(f'(\{x', y', z'\})\) determines whether the new structure is a proper or improper STS. Note that these transformations are essentially about doing Pasch switches.

A directed graph in which the vertices are combinatorial structures and the arcs show possible transformations from one structure into another is called a transition graph. Reversible transformations can be modeled with undirected graphs. Here we may consider an undirected transition graph \(G\) whose vertices are the proper and improper STSs. Repeated perturbations that are carried out uniformly at random correspond to a random walk (Markov chain) in \(G\). It can be shown \([1]\) that the unique limiting distribution of this Markov chain has the property that all proper Steiner triple systems in the connected component where the walk starts have equal probability in the stationary distribution. The central, still open, question is whether \(G\) is connected; this is the case for small orders of Steiner triple systems \([1, 6]\).
Table 2: Perturbations for proper and improper STSs

As we are only interested in proper STSs, it is useful to notice that the subsequence of proper STSs encountered in the walk is also a Markov chain with a unique stationary distribution that is uniform over the set of proper STSs in the same connected component; this follows from the proof of [11, Theorem 4].

The random walk can be started from an arbitrary proper STS and the number of blocks implies that at least $\Theta(v^2)$ time is needed for this initial step. The random walk can further be implemented so that each step takes constant time. The interesting question is now how fast the limiting distribution is reached; we shall get back to this question in Section 4.

### 3.3 Modifications of Cameron’s Algorithm

For Cameron’s algorithm, it is natural to address the possibility of considering only proper STSs and perturbations of those. Indeed, it is shown in [11] that the corresponding algorithm for Latin squares can be modified so that there are no improper intermediate structures. But a similar result for Steiner triple systems is missing.

For proper Steiner triple systems one may develop an algorithm, where in each step a cycle switch is carried out. Connectivity of the transition graph is also now the main issue, and here we do know that the graph is not connected as perfect Steiner triple systems necessarily stay perfect. But we do arrive at a uniform stationary distribution within a connected component by defining the switches in the following way.

Choose, uniformly at random, a 2-subset of points $\{a, b\}$ and a point $x$, $x \notin \{a, b\}$. If there is a block $\{a, b, x\}$, then we do nothing; this loop in the transition graph ensures that the Markov chain is aperiodic. Otherwise we
carry out the cycle switch with \(a\) and \(b\) as in (1) and \(x \in \{x_1, x_2, \ldots, x_n\}\). Some switches lead to the same transition, which can be handled in the framework of multigraphs (or graphs with weighted edges). It is an interesting question whether components of the transition graph have odd cycles, which is about aperiodicity after removing the loops.

### 3.4 Stinson’s Algorithm

Stinson [20] developed a celebrated hill-climbing algorithm for constructing STS\((v)\)s. This algorithm is presented as Algorithm 1.

```plaintext
1 \(B \leftarrow \emptyset\)
2 \textbf{while} \(|B| < v(v - 1)/6\) \textbf{do}
3 \hspace{1em} choose a point \(x\) that appears in fewer than \((v - 1)/2\) blocks of \(B\)
4 \hspace{1em} choose a point \(y\) such that \(\{x, y\}\) is not a subset of any block of \(B\)
5 \hspace{1em} choose a point \(z\) such that \(\{x, z\}\) is not a subset of any block of \(B\)
6 \hspace{1em} \textbf{if} there is a block \(B \in B\) such that \(\{y, z\} \subseteq B\) \textbf{then}
7 \hspace{2em} \(B \leftarrow B \setminus \{B\}\)
8 \hspace{1em} \(B \leftarrow B \cup \{\{x, y, z\}\}\)

Algorithm 1: Stinson’s hill-climbing algorithm
```

The distribution of STSs generated by Stinson’s algorithm is not known. Experiments showing that the distribution is not uniform for small orders [8] have perhaps discouraged people from further investigation. However, as argued in Section 3.1, the situation may change with growing order, and we shall later elaborate on this.

There exist situations where Stinson’s algorithm does not terminate, as shown by the example in [5, p. 37] for STS(15)s. That example can easily be generalized to an infinite family.

**Theorem 1.** Let \(v = 6n + 3\) with \(n \equiv 2 \pmod{3}\). Then there is a non-zero probability that an execution of Stinson’s algorithm in the search for an STS\((v)\) will not terminate.

**Proof.** For any partial STS, there is a non-zero probability of encountering it during the execution of the algorithm. Consider the situation where the partial STS is a TD(3, 2n + 1) transversal design. After choosing \(x\), the points \(y\) and \(z\) necessarily come from the same group as \(x\) and hence no blocks of the transversal design are ever removed. As the number of 2-subsets of points
in a group is $2n^2 + n$, which is not divisible by 3 when $n \equiv 2 \pmod{3}$, the partial STS cannot be completed.

An obvious way of dealing with a situation such as that in Theorem 1 is to set a limit on the number of executions of the while loop [20, 5, 8]. Obviously, at least $b = v(v-1)/6 = \Theta(v^2)$ iterations must be performed. The data structure for maintaining partial STSs can be implemented so that execution of any line in Algorithm 1 takes constant time, cf. [20]. There is empirical evidence [20] that Stinson’s algorithm with a limit on the number of steps needs $\Theta(v^2 \log v)$ time on the average to construct STSs.

3.5 Modifications of Stinson’s Algorithm

One weakness of Stinson’s algorithm is that the distribution of constructed STSs is not uniform for small orders. In particular, STS($v$)s with a large number of Pasch configurations are underrepresented. Heap, Danziger, and Mendelsohn [8] therefore suggest an extension of Stinson’s algorithm, where Pasch switches are occasionally carried out: If the constructed STS contains at least one Pasch configuration, then with a fixed probability $p$ a Pasch switch is carried out on a (random) Pasch configuration $P$ and the STS thereby obtained is also output. (A related algorithm that produces single STSs is also presented in [8], but that algorithm actually produces STSs with a different distribution; for example, STSs without Pasch configurations are more frequent.)

In the current work, experiments were made to try to find variants of Stinson’s algorithm that would lead to a more uniform distribution of constructed STSs. Whereas the approach in [8] is to modify complete STSs, the main focus here is on modifying the search algorithm itself.

Choices are made in three places in Algorithm 1, in lines 3 to 5. When implementing the algorithm, one needs to specify how these choices are made. Such details are often omitted in studies where the algorithm has been used. One reason behind this is that statements like “choose” and “choose randomly” can be understood as choosing uniformly among candidates. In any case, preciseness is inevitable when analysing and comparing variants of the algorithm.

Possible parameters for deriving probability distributions for the choice of the point $x$ in line 3 of Algorithm 1 are $n_q$, defined as the number of blocks in which the point $q \in V$ already appears, and $m_q = (v-1)/2 - n_q$. For any function $f_x(i)$ with nonnegative values, a probability distribution can now be obtained as
\[ p_x(q) = \frac{f_x(m_q)}{\sum_{j \in V} f_x(m_j)}, \quad (4) \]

where \( f_x(m_q) \) must be positive for at least one point \( q \). For example, uniform distribution over candidate points is obtained with the signum function, \( f_x(i) = \text{sgn} \, i \).

Obviously, nonuniform distributions may also be used for choosing \( y \) and \( z \) in Algorithm 1. The algorithms obtained with various distributions for \( x \), \( y \), and \( z \) are here called \textit{weighted Stinson’s algorithms}.

An alternative possibility of changing Stinson’s algorithm is to change lines 6 to 8 of Algorithm 1. In that part of the algorithm, a block \( B' = \{ x, y, z \} \) is added and a block \( B = \{ y, z, w \} \) is possibly removed. The situation where \( B \) is removed is precisely the switch (2), as noticed in [18, p. 629].

A switch in Stinson’s algorithm—removing one block and adding another—can be done in constant time. By changing this part of the algorithm, one may end up trading speed for a better distribution of constructed STSs. We call algorithms where switching is done in different ways \textit{extended Stinson’s algorithms}.

### 4 Experimental Results

As there is a unique STS(7), a unique STS(9), and two isomorphism classes of STS(13)s, the smallest order that can be used for evaluation of algorithms is 13. (In [5], the experimental work was done for orders 15 and 19.) The two STS(13)s—here called \( S_1 \) and \( S_2 \)—have automorphism groups of orders 6 and 39, that is, there are \( 13! / 6 = 1\,037\,836\,800 \) and \( 13! / 39 = 159\,667\,200 \) labelled such systems, that is, \( 13/15 = 0.8666 \ldots \) of the labelled STS(13)s belong to the former set and \( 2/15 = 0.1333 \ldots \) to the latter. The numbers of Pasch configurations in \( S_1 \) and \( S_2 \) are 8 and 13, respectively.

Our focus is here on (the two) isomorphism classes, overlooking the distribution of labelled systems, but, if necessary, any algorithm can be augmented with a final, random permutation of the points to guarantee uniform distribution amongst the labelled systems within an isomorphism class.

#### 4.1 Cameron’s Algorithm

Considering the case of STS(13)s, the subsequence of isomorphism classes of proper STSs encountered by Cameron’s algorithm can be modeled as a Markov chain with two states, which we also call \( S_1 \) and \( S_2 \). We denote the
transition probabilities $S_1 \rightarrow S_2$ and $S_2 \rightarrow S_1$ by $p$ and $q$, respectively, and the limiting, stationary distribution by $(\pi(1), \pi(2))$. Determining $p$ and $q$ analytically seems challenging, but we can get estimates with computational experiments. In a Markov chain with $10^9$ proper STSs, we get estimates $p \approx 0.108954$ and $q \approx 0.708279$. A two-state Markov chain with such transition probabilities converges to a probability distribution $(q/(p + q) \approx 0.86668, p/(p+q) \approx 0.13332)$, which is reasonably close to $(\pi(1) = 13/15, \pi(2) = 2/15)$.

The values of $p$ and $q$ can also be used to determine the convergence rate. Let $\mu_t(i)$ denote the probability that we are in state $i$ at time $t$. Regardless of the initial state,

$$|\mu_t(i) - \pi(i)| < \left|1 - p - q\right|^t < 0.2^t \quad \text{for} \quad i \in \{1, 2\};$$

see, for example, [19]. That is, we have exponential convergence. In fact, all Markov chains that are indecomposable and aperiodic converge exponentially quickly [19]. Unfortunately, a more formal treatment gets difficult for our problem with STSs of order greater than 13.

### 4.2 Stinson’s Algorithm

There are two main open questions regarding Stinson’s algorithm. Why does Stinson’s algorithm seemingly not generate Steiner triple systems uniformly at random for small parameters? And how could Stinson’s algorithm be modified to get better distributions for small parameters? A partial answer to the second question is provided in [8].

Already Stinson in his seminal paper [20] noticed an underrepresentation of Steiner triple systems with large isomorphism groups in experiments. Later studies such as [8] mention an underrepresentation of Steiner triple systems with many Pasch configurations. Notice, however, that by [17, Table 1.29] there is a correlation between the group order and the number of Pasch configurations for order 15, a commonly considered case in experimental studies.

Let us now consider results for variants of Stinson’s algorithm. We study STS(13)s and denote the number of STSs $S$ found in a set of experiments by $n(S)$. Hence, an algorithm that generates uniformly distributed STS(13)s should have $n(S_1)/(n(S_1) + n(S_2)) \approx 13/15$. When tabulating such results, we present the percent error

$$100 \cdot \left|\frac{n(S_1)}{n(S_1)+n(S_2)} - \frac{13}{15}\right|.$$
We modify Stinson’s algorithm with five parameters and each modified algorithm is used to generate $10^8$ STS(13)s in this comparison. Three parameters are for weighted Stinson’s algorithms. For $i \in \{x, y, z\}$, $W_i = 0, 1, \text{and } 2$ mean that the probability distribution $f_i(j)$ is given by (4) and $f_i(j) = \text{sgn } j$, $f_i(j) = j$, and $f_i(j) = \binom{j}{2}$, respectively. The original Stinson’s algorithm corresponds to the case $W_x = W_y = W_z = 0$.

Experimental results for weighted Stinson’s algorithms are shown in Table 3, omitting some cases due to symmetry.

| $W_x$ | 0 | 1 | 2 |
|------|---|---|---|
| 0    | 3.66 | 3.68 | 3.71 |
| 1    | 3.72 | 3.74 | 3.48 |
| 2    | 3.76 | 3.46 | 3.46 |

Table 3: Weighted Stinson’s algorithms

The best result is obtained with the variant $W_x = 2, W_y = W_z = 0$, which cuts roughly 20% of the deviation of the original algorithm.

The remaining two parameters give different extended Stinson’s algorithms. Different ways of additional switching was considered. The best result was obtained by carrying out an additional switch after each execution of the while loop of Algorithm 1 so only the results from that approach will be tabulated here. As discussed in Section 2.1, a switch is carried out with respect to the two points $a$ and $b$. Further consider a point $d$ so that $\{a, b, d\}$ is not a block; the point $d$ gives the cycle or path in which switching is carried out.

The two parameters for extended Stinson’s algorithms are as follows. The values of $a$, $b$, and $d$ are chosen uniformly with these restrictions.

$O \ | \{x, y, z\} \cap \{a, b, d\}| \geq O$

$I$ No impact for $I = 0$, $y = a$ for $I = 1$, and $y = d$ for $I = 2$

Note that if $I \neq 0$, then $|\{x, y, z\} \cap \{a, b, d\}| \geq 1$ and the cases $O = 0$ and $O = 1$ coincide; we therefore omit the latter case.

Table 4 shows computational results for the ten best performing parameter settings.

The overall best variant of Table 4 cuts as much as 99.95% of the deviation of the original algorithm. Note that although the results for weighted Stinson’s algorithms in Table 3 suggest that it is beneficial to set $W_x = 2$, the best
variant with this property is just fourth in the list. All other tested variants were inferior to those presented here.

A major drawback with extended Stinson’s algorithms is that the empirical run time increases by a factor of $v$ to $\Theta(v^3 \log v)$. One remedy for this increase could be to carry out cycle switching only when the STS is almost complete.

Returning to the question about why Stinson’s algorithm behaves as it does, we present data on partial STS(13)s in Tables 5 and 6, the latter also containing some experimental results. In Table 5, we give the number of blocks ($k$), the number of labelled partial STS(13)s that can be completed only to $S_1$ ($N_1$), to both $S_1$ and $S_2$ ($N_{12}$), and only to $S_2$ ($N_2$). Finally, we give the total number of isomorphism classes in the aforementioned three groups ($P'$) as well as the total number of isomorphism classes of partial STS(13)s ($P$). The algorithms used to get these numbers are standard [15]: removing blocks from STS(13)s and building up partial STS(13)s block by block, with isomorph rejection.

Table 6 contains some of the data from Table 5 in processed form as well as some computational results. Specifically, $p_X := N_X/(N_1 + N_{12} + N_2)$.

Finally, in the columns $q_X$, we show the distribution of partial STSs obtained experimentally with Algorithm in $10^6$ runs. Exact values of $q_X$ could also be obtained analytically in the context of absorbing Markov chains, but such an approach seems impracticable here due to the very high number of states (partial STSs).

Notably, the skew distribution emerges in late stages. For example, the experimentally obtained distribution for 21-block partial STSs is close to the theoretically obtained one.
| k  | N₁  | N₁₂  | N₂  | P'  | P   |
|----|-----|------|-----|-----|-----|
| 1  | 0   | 286  | 0   | 1   | 1   |
| 2  | 3645|      | 0   | 2   | 2   |
| 3  | 2748460 | 0 | 5 | 5 |
| 4  | 136963255 | 0 | 16 | 16 |
| 5  | 4777742970 | 0 | 53 | 54 |
| 6  | 119432267955 | 0 | 232 | 250 |
| 7  | 2112851941200 | 0 | 1259 | 1419 |
| 8  | 2445029874000 | 0 | 7843 | 9768 |
| 9  | 152456669164800 | 0 | 46338 | 71311 |
| 10 | 4830680444990400 | 10805956761600 | 201856 | 482568 |
| 11 | 893688887931840 | 86263438060800 | 562120 | 27226981 |
| 12 | 1109870976614400 | 2953455208704000 | 1041051 | 12142983 |
| 13 | 1012594014432000 | 589917494973600 | 4113388 | 41023224 |
| 14 | 711933731308800 | 8102692870272000 | 1504182 | 103043009 |
| 15 | 3940635194496000 | 8341343442432000 | 1310150 | 189057254 |
| 16 | 1729108757376000 | 6740366016384000 | 950220 | 248583304 |
| 17 | 595271842560000 | 4387496585472000 | 578055 | 228896680 |
| 18 | 161995946112000 | 2332247613696000 | 295188 | 143386618 |
| 19 | 331381290240000 | 1017142334208000 | 126036 | 58839345 |
| 20 | 485707622400000 | 3627447183360000 | 44573 | 15142370 |
| 21 | 456648192000000 | 1045724359680000 | 12875 | 2306220 |
| 22 | 207567360000000 | 238494896640000 | 2994 | 197746 |
| 23 | 415134720000000 | 548 | 9348 |
| 24 | 518918400000000 | 76 | 267 |
| 25 | 415134720000000 | 10 | 10 |
| 26 | 159667200000000 | 2 | 2 |

Table 5: Partial STS(13)s
| $k$ | $p_1$ | $p_{12}$ | $p_2$ | $q_1$ | $q_{12}$ | $q_2$ |
|-----|-------|---------|-------|-------|---------|-------|
| 1   | 0.0000 | 1.0000  | 0.0000| 0.0000| 1.0000  | 0.0000|
| 2   | 0.0000 | 1.0000  | 0.0000| 0.0000| 1.0000  | 0.0000|
| 3   | 0.0000 | 1.0000  | 0.0000| 0.0000| 1.0000  | 0.0000|
| 4   | 0.0000 | 1.0000  | 0.0000| 0.0000| 1.0000  | 0.0000|
| 5   | 0.0005 | 0.9995  | 0.0000| 0.0005| 0.9995  | 0.0000|
| 6   | 0.0045 | 0.9955  | 0.0000| 0.0043| 0.9957  | 0.0000|
| 7   | 0.0234 | 0.9766  | 0.0000| 0.0221| 0.9779  | 0.0000|
| 8   | 0.1050 | 0.8950  | 0.0000| 0.0913| 0.9087  | 0.0000|
| 9   | 0.3315 | 0.6674  | 0.0012| 0.2775| 0.7219  | 0.0006|
| 10  | 0.5711 | 0.4196  | 0.0094| 0.4898| 0.5054  | 0.0048|
| 11  | 0.7110 | 0.2635  | 0.0254| 0.6267| 0.3599  | 0.0134|
| 12  | 0.7807 | 0.1732  | 0.0461| 0.7088| 0.2666  | 0.0246|
| 13  | 0.8168 | 0.1158  | 0.0674| 0.7627| 0.1989  | 0.0384|
| 14  | 0.8370 | 0.0762  | 0.0867| 0.7977| 0.1488  | 0.0535|
| 15  | 0.8491 | 0.0484  | 0.1025| 0.8194| 0.1107  | 0.0699|
| 16  | 0.8565 | 0.0293  | 0.1142| 0.8365| 0.0786  | 0.0849|
| 17  | 0.8610 | 0.0167  | 0.1223| 0.8437| 0.0572  | 0.0991|
| 18  | 0.8637 | 0.0089  | 0.1274| 0.8540| 0.0361  | 0.1099|
| 19  | 0.8652 | 0.0043  | 0.1305| 0.8587| 0.0223  | 0.1190|
| 20  | 0.8661 | 0.0018  | 0.1322| 0.8644| 0.0119  | 0.1237|
| 21  | 0.8665 | 0.0006  | 0.1329| 0.8656| 0.0051  | 0.1293|
| 22  | 0.8666 | 0.0001  | 0.1333| 0.8717| 0.0020  | 0.1263|
| 23  | 0.8667 | 0.0000  | 0.1333| 0.8762| 0.0000  | 0.1238|
| 24  | 0.8667 | 0.0000  | 0.1333| 0.8828| 0.0000  | 0.1172|
| 25  | 0.8667 | 0.0000  | 0.1333| 0.8985| 0.0000  | 0.1015|
| 26  | 0.8667 | 0.0000  | 0.1333| 0.8985| 0.0000  | 0.1015|

Table 6: Data for partial STS(13)
4.3 Constructing Large Steiner Triple Systems

Of course, the fact that Stinson’s algorithm apparently does not perform optimally for STS(13)s and other small parameters does not necessarily mean that the same holds for larger parameters. Although we do conjecture that Stinson’s algorithm perform nonoptimally with respect to certain individual STSs for any order, we do not exclude the possibility that the obtained distribution converges in some manner to the desired distribution. (We refrain from formulating this statement explicitly, which would also require picking an appropriate measure for comparing distributions.)

But we do have some experimental results showing that the situation may be different for small and large orders. Specifically, we ran both Stinson’s algorithm and Cameron’s algorithm (returning the vth proper STS) generating each $10^6$ random STS per order for orders up to 200, counted the number of the configurations listed in Figure 1, divided this number by the left hand side of (3), and show the result in Figures 2 to 10. Figures 9 and 10 only show the average value divided by the left hand side of (3), as a boxplot would not provide any information given that most Steiner triple systems contain neither Fano nor Möbius–Kantor configurations. Algorithms from [10] were used for counting the configurations.

The fact that all curves seem to converge to 1 is encouraging both for the relevance of the random model and the performance of the algorithms.

References

[1] P. J. Cameron, A Markov chain for Steiner triple systems, 2002, available at http://www.maths.qmw.ac.uk/~pjc/csgnotes/random.pdf

[2] P. J. Cameron, Research problems from the 18th British Combinatorial Conference, Discrete Math. 266 (2003), 441–451.

[3] P. J. Cameron, A generalisation of t-designs, Discrete Math. 309 (2009), 4835–4842.

[4] C. J. Colbourn, Triple systems, in: C. J. Colbourn and J. H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall/CRC, Boca Raton, 2007, pp. 58–71.

[5] C. J. Colbourn and A. Rosa, Triple Systems, Clarendon Press, Oxford, 1999.

[6] A. L. Drizen, M. J. Grannell, and T. S. Griggs, Pasch trades with a negative block, Discrete Math. 311 (2011), 2411–2416.
Figure 2: Pasch

Figure 3: Mitre
Figure 4: Fano–line

Figure 5: Crown
Figure 10: Möbius–Kantor

[7] M. J. Grannell, T. S. Griggs, and J. P. Murphy, Some new perfect Steiner triple systems, *J. Combin. Des.* 7.5 (1999), 327–330.

[8] D. Heap, P. Danziger, and E. Mendelsohn, Hill-climbing to Pasch valleys, *J. Combin. Des.* 15.5 (2007), 405–419.

[9] D. Heinlein and P. R. J. Östergård, Steiner triple systems of order 21 with subsystems, submitted for publication, preprint at arXiv:2104.06825.

[10] D. Heinlein and P. R. J. Östergård, Algorithms and complexity for counting configurations in Steiner triple systems, *J. Combin. Des.* 30 (2022), 527–546.

[11] M. T. Jacobson and P. Matthews, Generating uniformly distributed random Latin squares, *J. Combin. Des.* 4 (1996), 405–437.

[12] M. Jerrum, Counting, Sampling and Integrating: Algorithms and Complexity, Birkhäuser, Basel, 2003.

[13] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theoret. Comput. Sci.* 43 (1986), 169–188.

[14] P. Kaski and P. R. J. Östergård, The Steiner triple systems of order 19, *Math. Comp.* 73 (2004), 2075–2092.
[15] P. Kaski and P. R. J. Östergård, Classification Algorithms for Codes and Designs, Springer, Berlin, 2006.

[16] M. Kwan, Almost all Steiner triple systems have perfect matchings, *Proc. London Math. Soc. (3)* 121 (2020), 1468–1495.

[17] R. Mathon and A. Rosa, 2-$(v,k,\lambda)$ designs of small order, in: C. J. Colbourn and J. H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall/CRC, Boca Raton, 2007, pp. 25–58.

[18] P. R. J. Östergård, Switching codes and designs, *Discrete Math.* 312 (2012), 621–632.

[19] J. S. Rosenthal, Convergence rates for Markov chains, *SIAM Rev.* 37 (1995), 387–405.

[20] D. R. Stinson, Hill-climbing algorithms for the construction of combinatorial designs, *Ann. Discrete Math.* 26 (1985), 321–334.