Heinlein, Daniel; Östergård, Patric R.J.

Enumerating Steiner triple systems

Published in:
Journal of Combinatorial Designs

DOI:
10.1002/jcd.21906

Published: 01/10/2023

Document Version
Publisher's PDF, also known as Version of record

Received: 17 March 2023 | Revised: 19 June 2023 | Accepted: 20 June 2023

DOI: 10.1002/jcd.21906

RESEARCH ARTICLE

WILEY

Enumerating Steiner triple systems

Daniel Heinlein | **Patric R. J. Östergård**

Department of Information and Communications Engineering, Aalto University School of Electrical Engineering, Aalto, Finland

Correspondence
Daniel Heinlein, Department of Information and Communications Engineering, Aalto University School of Electrical Engineering, P.O. Box 15400, 00076 Aalto, Finland.
Email: daniel.heinlein@aalto.fi

Funding information
Academy of Finland,
Grant/Award Number: 331044

Abstract

Steiner triple systems (STSs) have been classified up to order 19. Earlier estimations of the number of isomorphism classes of STSs of order 21, the smallest open case, are discouraging as for classification, so it is natural to focus on the easier problem of merely counting the isomorphism classes. Computational approaches for counting STSs are here considered and lead to an algorithm that is used to obtain the number of isomorphism classes for order 21: 14,796,207,517,873,771.

KEYWORDS

classification, counting, regular graph, Steiner triple system

MATHEMATICS SUBJECT CLASSIFICATION

05B07

1 | INTRODUCTION

A *Steiner triple system* (STS) is a pair $(V, \mathcal{B})$, where V is a set of *points* and $\mathcal{B}$ 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 STS, and an STS of order v is denoted by STS(v). An STS(v) exists if and only if

$$v \equiv 1 \text{ or } 3 \pmod 6. \tag{1}$$

For more information about STSs, see [4, 6].

An STS(v) is *isomorphic* to another STS(v) if there exists a bijection between the point sets that maps blocks onto blocks; such a bijection is an *isomorphism*. An isomorphism of an STS

onto itself is an *automorphism* of the STS. The automorphisms of an STS form a group under composition, the *automorphism group* of the STS.

STSs have been classified up to order 19. The numbers of isomorphism classes are 1, 1, 1, 2, 80, and 11,084,874,829 for the admissible orders 3, 7, 9, 13, 15, and 19, respectively; see [19, 20] for details about the classification of STSs and [9] for historical remarks and speculations about future results. Indeed, for all these orders, representatives of the isomorphism classes have been determined. For the smallest open case of order 21, however, such an approach does not seem feasible at the moment as the number of isomorphism classes has been estimated [13] to be greater than 10^{16}. There are numerous studies on the classification of subclasses of STSs of order 21 [3, 10, 16–18, 23, 25–29, 36, 37].

If an instance of classifying combinatorial structures with given parameters is infeasible with the available resources, it might still be possible to *count* the isomorphism classes. There are several examples of such studies in the past, for instance, for Latin squares [15, 30], one-factorizations of complete graphs [21], and STSs with subsystems [22]. A classification of STS(21)s with nontrivial automorphisms is available [17], so the number of isomorphism classes can be obtained with the Orbit–Stabilizer theorem and a count of the number of labeled STS(21)s.

A general approach for counting labeled combinatorial structures as fast as possible is to divide the structures into parts and then do the counting for each part. Such divisions come naturally for certain objects, such as Latin squares (via Latin rectangles) and one-factorizations of complete graphs (via sets of one-factors), but for STSs the situation is not so obvious. In the approach developed in the current work, a division of the blocks into two parts is obtained via graphs that are close to being regular. Classification of such graphs is therefore needed as an ingredient. The main result of the current work is as follows.

Theorem 1. *The number of isomorphism classes of STSs of order* 21 *is* 14,796,207,517,873,771.

Note that although this work is about counting and not about classification, representatives of each isomorphism class are *seen piecewise* many times. It might even be possible to modify the counting algorithm to investigate also for STS(21)s some of the properties that were studied for STS(19)s in [5].

The paper is organized as follows. The general approach is presented in Section 2. In Section 3, computational subproblems are considered: the choice of a partition of the blocks (Section 3.1), classification of graphs with given degree sequences (Section 3.2), counting labeled systems (Section 3.4), and counting and validating the number of isomorphism classes (Section 3.5). Finally, in Section 4, the computational results are summarized.

2 | GENERAL APPROACH

The general framework considered here builds on that of [13, 22], where STSs with subsystems are considered. Actually, the approach for classifying STS(19)s used in [19] can also be put into this framework.

In [13, 19, 22], the blocks are partitioned into three sets: a *defining set* $\mathcal{B}'$ that forms a subsystem and sets $\mathcal{F}$ and $\mathcal{D}$ whose blocks intersect the point set of $\mathcal{B}'$ in one point and zero points, respectively. As $\mathcal{B}'$ contains the blocks of a subsystem—which in [19]

is just a sub-STS(3), a single block—no block can intersect the point set of $\mathcal{B}'$ in exactly two points.

Before giving the theorem [13, 22] showing the full picture we need to define configurations. A (v_r, b_k) *configuration* is an incidence structure with v points and b blocks, such that each block contains k points, each point occurs in r blocks, and two different blocks intersect in at most one point.

Theorem 2. *Let $(V, \mathcal{B})$ be an STS(v) that has a sub-STS(w) $(W, \mathcal{B}')$. Then*

(i) $\mathcal{B} = \mathcal{B}' \cup \mathcal{F} \cup \mathcal{D}$, *where $\mathcal{F}$ and $\mathcal{D}$ are the sets of blocks that intersect W in* 1 *and* 0 *points, respectively,*
(ii) $\mathcal{F} = \bigcup_{p \in W} \mathcal{F}_p$, *where $\mathcal{F}_p$ is the set of blocks in $\mathcal{F}$ that contain p,*
(iii) $\mathcal{F}'_p = \{B \backslash \{p\} : B \in \mathcal{F}_p\}$ *with $p \in W$ is a one-factor of a graph G with vertices $V \backslash W$ and edges $\bigcup_{p \in W} \mathcal{F}'_p$,*
(iv) $\{\mathcal{F}'_p : p \in W\}$ *is a one-factorization of G,*
(v) G *is w-regular and its complement $\overline{G}$ is $(v - 2w - 1)$-regular, and*
(vi) $\overline{G}$ *can be decomposed into a set of edge-disjoint* 3*-cycles—$\mathcal{D}$ being one possible set—which forms a*

$$((v - w)_{(v-2w-1)/2}, ((v - w)(v - 2w - 1)/6)_3)$$

configuration.

The fact that G is a regular graph in Theorem 2(v) is essential. STSs with subsystems can therefore be constructed via a classification of certain regular graphs. We shall now modify this approach for situations where we still have a partition of the point set into W and $V \backslash W$, but W does not induce a subsystem. Graphs will play a central role also in the modified approach, but the graphs considered will have also other degree sequences than those of regular graphs. Informally, one could call the graphs nearly regular. The order of the vertices will not matter, so we may use an abbreviated form $d_1^{n_1} d_2^{n_2} \cdots d_k^{n_k}$ for a degree sequence with n_i copies of d_i, $1 \leq i \leq k$.

A PBD(w, K) *pairwise balanced design* is a pair $(W, \mathcal{B}')$, where W is a set of w points, $\mathcal{B}'$ is a set of blocks with sizes from K, and every pair of distinct points occurs in exactly one block. For a $(V, \mathcal{B})$ STS and any point set $W \subseteq V$, the pair $(W, \mathcal{B}')$, where

$$\mathcal{B}' = \{B \cap W : B \in \mathcal{B}, |B \cap W| \geq 2\}$$

is a PBD$(|W|, \{2, 3\})$.

We shall now see what happens when W induces a PBD$(|W|, \{2, 3\})$. The items of Theorem 3 follow those of Theorem 2. As the results follow directly from definitions, the theorem is stated without proof.

Theorem 3. *Let $(V, \mathcal{B})$ be an STS(v) and let $W \subseteq V$. Then*

(i) $\mathcal{B} = \mathcal{B}'' \cup \mathcal{F} \cup \mathcal{D}$, *where $\mathcal{B}''$, $\mathcal{F}$, and $\mathcal{D}$ are the sets of blocks that intersect W in at least* 2*, exactly* 1*, and exactly* 0 *points, respectively,*
(ii) $\mathcal{F} = \bigcup_{p \in W} \mathcal{F}_p$, *where $\mathcal{F}_p$ is the set of blocks in $\mathcal{F}$ that contain p,*

(iii) $\mathcal{F}'_p = \{B \setminus \{p\} : B \in \mathcal{F}_p\}$ *with* $p \in W$ *forms a partition of* $V \setminus (W \cup W_p)$, *where* $W_p = \bigcup_{B:p\in B\in\mathcal{B}''} B \cap (V \setminus W)$; $\mathcal{F}'_p$ *can also be viewed as a color class of a proper edge-coloring of a graph* G *with vertices* $V \setminus W$ *and edges* $\bigcup_{p\in W} \mathcal{F}'_p$,
(iv) $\{\mathcal{F}'_p : p \in W\}$ *is a proper edge-coloring of* G,
(v) *the degree of a vertex* v *in* G *is* $|W| - |\{p : v \in W_p\}|$, *and*
(vi) $\overline{G}$ *can be decomposed into a set of edge-disjoint 3-cycles—*$\mathcal{D}$ *being one possible set.*

For counting the STS(v)s via the graphs G in Theorem 3, one first needs to define the PBD($w, \{2, 3\}$) induced by W. To get an easy formula for the final count, the number of occurrences of this PBD should only depend on the order v. The next step is then to determine possible degrees of the graphs G and classify them. For each classified graph G, there is the computational task of finding the number of proper edge-colorings of G (with some additional requirements) and the number of decompositions of $\overline{G}$ into triangles. From the data of these computations, and a classification of the STS(v)s with nontrivial automorphisms, the number of isomorphism classes can finally be obtained using the Orbit–Stabilizer theorem.

3 | COMPUTATIONAL SUBTASKS

We shall now look at the subtasks of the outlined algorithm, with details for STS(21)s at the end of each subsection and separated from the general discussion.

3.1 | Choice of defining set

The choice of the defining set, that is, the PBD($w, \{2, 3\}$) induced by W, is critical for the length of the computation. The final computation is expected to be very extensive, and a proper and justified choice cannot be made without estimations and experiments. As for time usage, the core subproblem is that of obtaining the number of possible sets $\mathcal{D}$ and $\mathcal{F}$ for a given graph G (Theorem 3(iii)). The latter number further depends on the sets W_p, also defined in Theorem 3(iii), and these numbers are denoted by $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$, where $\mathcal{W}$ is a *multiset* whose elements are the sets W_p, $p \in W$. It is important to notice that $N_{\mathcal{D}}(G)$ is defined to be the number of decompositions of the *complement* of G. Algorithms for obtaining these numbers will be discussed in Section 3.3.

The problem of classifying the graphs G is less time-consuming and even more so in an experimental phase, where randomly generated graphs are considered. Random graphs can be obtained in a rather straightforward manner using a Markov chain Monte Carlo (MCMC) algorithm. For a given degree sequence, one graph can be constructed with the Havel–Hakimi algorithm [11, 12]. Thereafter, an MCMC algorithm [34, 38] can be applied to get a sequence of more graphs. In the core of this algorithm is a switch where two random edges $\{a, b\}$ and $\{c, d\}$ are replaced with the edges $\{a, c\}$ and $\{b, d\}$ if the latter edges do not already exist. As the produced graphs are only used to get rough estimates for the main algorithm, it is not necessary to be meticulous in tuning the details of this approach.

To optimize the computing time of the main counting algorithm, experiments can be carried out to find a good choice of PBD($w, \{2, 3\}$)s $(W, \mathcal{B}')$. We denote $\mathcal{B}' = \mathcal{B}'_3 \cup \mathcal{B}'_2$ where the subindex gives the block size. It is further desired that the STSs do not contain many PBDs

of the given type. Namely, the more copies there are, the more times each STS will be encountered in the final computation and the longer the execution time will be. In this sense, it is generally good to have $|\mathcal{B}'_3|$ large. (The extremal case of $|\mathcal{B}'_3| = 0$ is referred to as an *independent set* in an STS.) We let $W = \{0, 1, 2, ..., w - 1\}$ in the sequel.

A central feature of the main algorithm is that it counts $N_{\mathcal{D}}(G)N_{\mathcal{F}}(G, \mathcal{W})$ labeled STSs with a computing time in the order of $N_{\mathcal{D}}(G) + N_{\mathcal{F}}(G, \mathcal{W})$. It is therefore desired to have $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$—alternatively, the computing times for these—in approximately the same order of magnitude.

3.1.1 | STS(21)

For STS(21)s and an STS(3)—a single block—as a defining set (so $w = 3$), there is a huge imbalance and determining $N_{\mathcal{D}}(G)$ is very time-consuming [19]. For large w, there is an imbalance in the other direction: for STS(21)s and an STS(7) as a defining set (so $w = 7$), $N_{\mathcal{D}}(G) = 0$ for most of the graphs [13]. Therefore, possible values of w are here narrowed down to $4 \le w \le 6$.

An STS(21) contains 1260, 4725, and 10,584 PBDs that are independent sets for $w = 3, 4$, and 5, respectively [8]. These numbers are up to more than an order of magnitude bigger than for other choices as we shall soon see, and they support the choice of maximizing $|\mathcal{B}'_3|$.

For STS(21)s with PBD$(w, \{2, 3\})$s $(W, \mathcal{B}')$, $4 \le w \le 6$ restricted in the aforementioned way, experimental results are presented in Table 1. Only the blocks $\mathcal{B}'_3$ of size 3 in a PBD are given, as these uniquely determine the blocks of size 2. The column N' gives the number of w-subsets of points in an STS(21) that induce such a PBD. For $w = 6$, we have a Pasch configuration, also known as a quadrilateral; the number of occurrences is not constant in this case.

For 1000 random graphs G with given degree sequences, the counts $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$ are determined, and the averages are tabulated in the columns $\overline{N_{\mathcal{D}}(G)}$ and $\overline{N_{\mathcal{F}}(G, \mathcal{W})}$, respectively. The average computing times, in milliseconds, are shown in the columns $\overline{t_{\mathcal{D}}(G)}$ and $\overline{t_{\mathcal{F}}(G, \mathcal{W})}$. It turns out that $\overline{N_{\mathcal{D}}(G)} \cdot \overline{N_{\mathcal{F}}(G, \mathcal{W})}$ differs from the average of the product, $\overline{N_{\mathcal{D}}(G) \cdot N_{\mathcal{F}}(G, \mathcal{W})}$, typically by only a few percent. The time estimations are incomparable between Table 1 and the final computation in Section 4, because less optimized versions of the counting algorithms were applied. Nevertheless, the time estimations are consistent among the entries in Table 1 and allow comparisons.

TABLE 1 Impact of different choices of $(W, \mathcal{B}')$.

w	$\mathcal{B}'_3$	N'	G	$\overline{N_{\mathcal{D}}(G)}$	$\overline{t_{\mathcal{D}}(G)}$	$\overline{N_{\mathcal{F}}(G, \mathcal{W})}$	$\overline{t_{\mathcal{F}}(G, \mathcal{W})}$
4	{012}	1260	$2^3 4^{14}$	160,837,000.0	362,091.0	221.7	4.7
5	{012, 034}	945	$1^2 5^{14}$	12,666.6	33.9	3994.0	11.2
			$1^1 3^2 5^{13}$	10,363.2	30.1	7088.9	20.8
			$3^4 5^{12}$	8714.6	26.5	26,439.6	98.5
6	{012, 034,		$0^1 6^{14}$	2.3	0.2	539,449.0	858.1
	135, 245}		$2^1 4^1 6^{13}$	1.7	0.2	538,846.0	817.4
			$4^3 6^{12}$	1.3	0.2	1,306,050.0	2441.3

The value of $\mathcal{W}$ is uniquely determined by G for all instances of Table 1 but the degree sequence $3^4 5^{12}$. Denoting the set of vertices of degree 3 in such a graph by $\{a, b, c, d\}$, there are three possibilities for $\mathcal{W}$:

$$\{\{\}, \{a, b\}, \{c, d\}, \{a, c\}, \{b, d\}\},$$
$$\{\{\}, \{a, b\}, \{c, d\}, \{a, d\}, \{b, c\}\}, \quad \text{and}$$
$$\{\{\}, \{a, c\}, \{b, d\}, \{a, d\}, \{b, c\}\}.$$

The entries in Table 1 for this case are summed over all three subcases.

The choice of w can now be made by comparing the values of

$$\frac{\overline{\mathrm{N}_{\mathcal{D}}(G)} \cdot \overline{\mathrm{N}_{\mathcal{F}}(G, \mathcal{W})}}{\overline{t_{\mathcal{D}}(G)} + \overline{t_{\mathcal{F}}(G, \mathcal{W})}}.$$

The obvious choice is $w = 5$ with $\mathcal{B}'_3 = \{012, 034\}$.

3.2 | Classifying graphs with given degree sequences

Faradžev [7] studied the problem of classifying graphs with given degree sequences already in the 1970s. The main focus in published studies on algorithms for classifying graphs with given degree sequences has been on regular graphs, with work by Meringer [32] showing the true potential of such algorithms. Specific algorithms have been published in particular for graphs with degree 3, that is, cubic graphs [1, 2].

The graph isomorphism program `nauty` [31] contains a suite of programs called `gtools`, which in turn contains the `geng` program for generating graphs up to isomorphism and getting the automorphism groups simultaneously. The `geng` program is able to construct graphs with degrees in a given interval. Moreover, `geng` can be called from another program, thereby eliminating a need of extensive computer memory. This is precisely what is required for the final computation of the counting approach presented here.

3.2.1 | STS(21)

Let us consider the degree sequences for STS(21)s with $w = 5$ and $\mathcal{B}'_3 = \{012, 034\}$ one by one. The final graphs will in all cases have 16 vertices and 36 edges. As the `geng` program will produce also graphs that do not have the desired degree sequences, the produced graphs must be filtered. One feature of `geng` is that it can produce parts of all graphs with little overhead. In the current work where the computation is split in parts and distributed, it is indeed important that one does not have to run the whole graph generation for each part.

Features of the `geng` program like the aforementioned one make it an ideal tool here. As the time consumption for graph generation is negligible with respect to the total computing time of the counting algorithm, it is not necessary to consider the possibility of additional pruning in the search tree of `geng` or using software dedicated to generating graphs with given degree sequences (but lacking certain features of `geng`).

$1^2 5^{14}$: For such a graph G, the graph G' induced by the vertices of degree 5 has degree sequence 5^{14}, $3^1 5^{13}$, or $4^2 5^{12}$. Moreover, $|\mathrm{Aut}(G)| = 2|\mathrm{Aut}(G')|$ in the first case—the transposition

of the two vertices of degree 1 in G is then always in $\mathrm{Aut}(G)$—and $|\mathrm{Aut}(G)| = |\mathrm{Aut}(G')|$ in the other two cases. We shall later see that the second of the three subcases can be ignored.

$1^1 3^2 5^{13}$: For such a graph G, the graph G' induced by the vertices of degree at least 3 has degree sequence $2^1 3^1 5^{13}$ or $3^2 4^1 5^{12}$. In both cases, $|\mathrm{Aut}(G)| = |\mathrm{Aut}(G')|$.

$3^4 5^{12}$: This case is handled directly.

For the first two degree sequences, there is in all cases a unique way of adding edges to get G from G'.

The parameters of the instances to be computed with `geng` are summarized in Table 2, with the case that can be ignored within parentheses. In the first two columns, the final and intermediate degree sequences are given. The columns contain the order of the graph n, the number of edges e, the minimum degree d, and the maximum degree D.

3.3 | Counting graph decompositions

Given G and $\mathcal{W}$, we want to determine $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$. This is here done with algorithms that explicitly find all such decompositions.

STSs are decompositions of complete graphs into 3-cycles (triangles), and for $N_{\mathcal{D}}(G)$ we have decompositions of other graphs. The problem of finding such decompositions is recurrent in computational studies of STSs [19] and can be considered within the framework of exact cover. In the *exact cover problem*, we have a set S and a collection $\mathcal{S}$ of subsets of S, and the decision problem asks whether there exists a partition of S using sets from $\mathcal{S}$. One can further ask for one or all such partitions. Any graph decomposition problem can be phrased as an instance of the exact cover problem, where S is the set of edges and $\mathcal{S}$ contains the set of candidate subgraphs in a decomposition. Instances of the exact cover problem encountered in the current work were solved using the `libexact` software [24].

For the computation of $N_{\mathcal{F}}(G, \mathcal{W})$, experiments show that a two-stage approach analogous to that used in [13] is efficient. Let $G = (V \setminus W, E)$ and $\mathcal{W}$ be fixed, with $W_0, W_1, \ldots, W_{w-1}$ in $\mathcal{W}$.

In the first stage, for $0 \le i \le w - 1$, all perfect matchings in the graph induced by $V \setminus (W \cup W_i)$ are determined (e.g., using exact cover) and saved in $\mathcal{F}'_i$. In the second stage, we want to decompose G by taking exactly one matching from $\mathcal{F}'_i$ for each i, $0 \le i \le w - 1$. The following theorem shows that it suffices to consider disjointness of matchings.

Theorem 4. *A set of disjoint matchings of a graph G with one matching from each of the collections $\mathcal{F}'_i$, $0 \le i \le w - 1$, decomposes G.*

TABLE 2 Parameters of `geng` instances.

S_1	S_2	n	e	d	D
$1^2 5^{14}$	5^{14}	14	35	5	5
($1^2 5^{14}$	$3^1 5^{13}$	14	34	3	5)
$1^2 5^{14}$	$4^2 5^{12}$	14	34	4	5
$1^1 3^2 5^{13}$	$2^1 3^1 5^{13}$	15	35	2	5
$1^1 3^2 5^{13}$	$3^2 4^1 5^{12}$	15	35	3	5
$3^4 5^{12}$	$3^4 5^{12}$	16	36	3	5

Proof. Each edge of the matchings is an edge of G. As the matchings are disjoint, each edge of G occurs in at most one matching. The theorem now follows as all matchings in $\mathcal{F}'_i$ for a given i have the same number of edges. □

It is straightforward to implement a backtrack search for this problem, cf. [33, p. 36]. With a very small value of w one can use nested loops instead. If matchings are stored as bitmaps, then disjointness of matchings can be determined with an AND operation. Also `libexact` can solve instances of this problem.

We shall now see that the number of solutions can actually be counted without traversing the search tree to its leaves.

Theorem 5. *A set of disjoint matchings, one from each of $w-1$ collections $\mathcal{F}'_i$, can be extended in a unique way to a decomposition of $G=(V\setminus W, E)$ with disjoint matchings from all collections $\mathcal{F}'_i$.*

Proof. For a given i, $0 \le i \le w-1$, every matching of $\mathcal{F}'_i$ saturates the same subset of $V\setminus W$. After deleting the edges of a set of $w-1$ disjoint matchings, one from each of $w-1$ collections $\mathcal{F}'_i$, from E, we get a graph with vertex degrees 0 and 1, that is, a matching, which necessarily occurs in the collection $\mathcal{F}'_i$ from which no matching has yet been taken. □

Theorem 6. *Consider a collection $\mathcal{S}$ of $w-2$ disjoint matchings from different collections $\mathcal{F}'_i$, and assume that no matching is included from the collections $\mathcal{F}'_a$ and $\mathcal{F}'_b$, $0 \le a < b \le w-1$. Further let M_a (M_b) be the number of matchings in $\mathcal{F}'_a$ ($\mathcal{F}'_b$) that are disjoint from all matchings of $\mathcal{S}$. Then $M_a = M_b$ and this is the number of ways $\mathcal{S}$ can be completed to a decomposition of G with elements from all collections $\mathcal{F}'_i$.*

Proof. By Theorem 5, after extending $\mathcal{S}$ with a matching from $\mathcal{F}'_a$, there is exactly one possibility of completing the decomposition in the desired way, so the total number of completions is M_a. In an analogous way, by first extending with a matching from $\mathcal{F}'_b$, we get that the total number is M_b. It then follows that $M_a = M_b$. □

When some of the sets W_i coincide, we can combine the corresponding, identical collections $\mathcal{F}'_i$ to get collections $\mathcal{F}''_j$ for some $0 \le j \le w'$, where $w' < w$. The number of matchings to be taken from a collection $\mathcal{F}''_j$ is precisely the number of identical collections combined, and the matchings can now be taken with respect to some defined total *order*. In this framework, we say that a matching is *compatible* with a collection $\mathcal{S}$ of matchings if it is disjoint from the matchings in $\mathcal{S}$ and greater than the matchings in $\mathcal{S}$ that come from the same collection $\mathcal{F}''_j$.

In the framework of $\mathcal{F}''_j$ we do not get quite the same situation as in Theorem 6. Namely, when picking the second to last matching it may be that the unique final matching is not a compatible candidate. However, it certainly is compatible if the total number of matchings to be taken from its collection $\mathcal{F}''_j$ is 1. In the situation when we have a collection from which exactly two matchings should be taken, we can avoid considering it during the search, but still maintain its subset of compatible candidates; if it has M compatible candidates when two matchings are missing, then the number of completions is $M/2$.

3.3.1 | STS(21)

When counting STS(21)s with $w = 5$ and $\mathcal{B}'_3 = \{012, 034\}$, there are 36 edges in a graph G, so bitmaps of matchings fit in 64-bit words. For the degree sequence $3^4 5^{12}$ all W_i are distinct. By Theorem 6, we then need only carry out the search to the level with $w - 2 = 3$ matchings. The possibilities of $\mathcal{W}$ listed at the end of Section 3.1 show that in each case we need one matching with 8 edges and four matchings with 7 edges. The number of candidates is larger for 8 edges than for 7 edges, so we have used Theorem 6 and chosen to ignore the matchings with 8 edges in the counting. Obviously, we then need not even determine the matchings with 8 edges, which speeds up the algorithm even further.

In the case of STS(21)s with $w = 5$ and $\mathcal{B}'_3 = \{012, 034\}$, we get situations where not all W_i are distinct for the degree sequences $1^2 5^{14}$ and $1^1 3^2 5^{13}$. For these two degree sequences, the five values of W_i are, respectively,

$$\{\}, \{a, b\}, \{a, b\}, \{a, b\}, \{a, b\} \text{ and}$$
$$\{\}, \{a, b\}, \{a, b\}, \{a, c\}, \{a, c\}.$$

Note that in the former case, with degree sequence $1^2 5^{14}$, the vertices a and b cannot have the same (unique) neighbor, as one of the matchings (corresponding to $\{\}$ in $\mathcal{W}$) contains edges saturating all vertices. A search algorithm will immediately realize this, but the main importance of this observation is that one of the cases in Table 2 can be ignored in the classification of graphs G.

For the degree sequences $1^2 5^{14}$ and $1^1 3^2 5^{13}$, in the first stage of the two-stage approach for computing $N_{\mathcal{F}}(G, \mathcal{W})$ we compute 2 and 3 collections of matchings, respectively. For the degree sequence $1^2 5^{14}$, we search for a decomposition with 1 matching from one collection and 4 matchings from the other, and for the degree sequence $1^1 3^2 5^{13}$, we search for a decomposition with 1 matching from one collection and 2 matchings from each of the two other collections. These cases are not time-critical for the complete algorithm, so `libexact` was used.

3.4 | Counting labeled systems

The main question still remains: how to get the total number of labeled STSs from the data obtained as described in Section 3.3? Let $\mathcal{G}$ denote the set of all pairs $(G, \mathcal{W})$ to consider for given values of v, w, and $\mathcal{B}'_3$.

Theorem 7. *The total number of labeled STS(v)s is*

$$\frac{1}{N'} \cdot \sum_{(G,\mathcal{W})\in\mathcal{G}} \frac{Kv!N_{\mathcal{D}}(G)N_{\mathcal{F}}(G, \mathcal{W})}{w!|\mathrm{Aut}(G)|}, \tag{2}$$

where K is the number of ways to complete the triples and matchings counted in $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$ to STS(v)s.

Proof. We count the number of pairs (A, B), where A is an STS and B is a set of blocks of A isomorphic to $\mathcal{B}'_3$. By definition, N' is the number of sets of blocks of A isomorphic to $\mathcal{B}'_3$, so we have to divide the final count by N' to get the desired result.

The number of ways of choosing the $v - w$ points of G out of the v points is $\binom{v}{v-w}$. The number of labeled graphs on these points is further $(v-w)!/|\mathrm{Aut}(G)|$. For each such graph we have $N_{\mathcal{D}}(G)$ solutions for the $\mathcal{D}$ part and $N_{\mathcal{F}}(G, \mathcal{W})$ solutions for the matchings of the $\mathcal{F}$ part.

As K is the number of ways such a structure can be completed to STS(v)s, the theorem follows. □

We still need to elaborate on K, the number of ways to complete the parts. First, we want to find the number of PBDs of the given type on the point set W with a completion of the blocks of size 2 using points from the (fixed) set $W' = \bigcup_{p \in W} W_p$. This number can be obtained combinatorially or computationally.

Finally, when forming $\mathcal{F}$ from a collection of matchings, there are several possibilities in case of coinciding sets W_i. This is seen also in the forming of the collections $\mathcal{F}''_j$, $0 \le j \le w' - 1$, from the collections $\mathcal{F}'_i$, $0 \le i \le w - 1$. Specifically, letting P_j be the number of collections $\mathcal{F}'_i$ combined into $\mathcal{F}''_j$, the desired count is $\prod_{j=0}^{w'-1} P_j!$. Note, however, that for the smallest instances a collection $\mathcal{F}''_j$ may contain an empty set; then $P_j!$ should be replaced by 1.

3.4.1 | STS(21)

The following examples also apply to STSs of other orders than 21. With $w = 5$ and $\mathcal{B}'_3 = \{012, 034\}$, we have the three possible situations depicted in Figure 1. The desired numbers are obtained by dividing 5! with the order of the automorphism groups of these structures, which are 4, 2, and 1, respectively. We then get the following values of K for the three degree sequences:

$$1^2 5^{14} : K = (5!/4) \cdot 4! = 720,$$
$$1^1 3^2 5^{13} : K = (5!/2) \cdot 2! \cdot 2! = 240,$$
$$3^4 5^{12} : K = (5!/1) \cdot 1 = 120.$$

For orders other than 21, we get an exception due to an empty set in $\mathcal{F}''_j$ in exactly one situation. Namely, for order 7 and the degree sequence $1^2 5^{14}$, we get $K = (5!/4) \cdot 1 = 30$.

3.5 | Counting isomorphism classes

Denote by $N_{v,i}$ the number of isomorphism classes of STS(v)s with an automorphism group of order i. By the Orbit–Stabilizer theorem, the number of labeled STSs is

$$v! \sum_i \frac{N_{v,i}}{i}. \tag{3}$$

If, for some fixed v, we know the number of labeled systems and $N_{v,i}$ for all $i \ge 2$, we arrive at an equation where only $N_{v,1}$ is unknown and can be determined. An error has occurred if the result is not an integer. The total number of isomorphism classes is obviously $\sum_i N_{v,i}$.

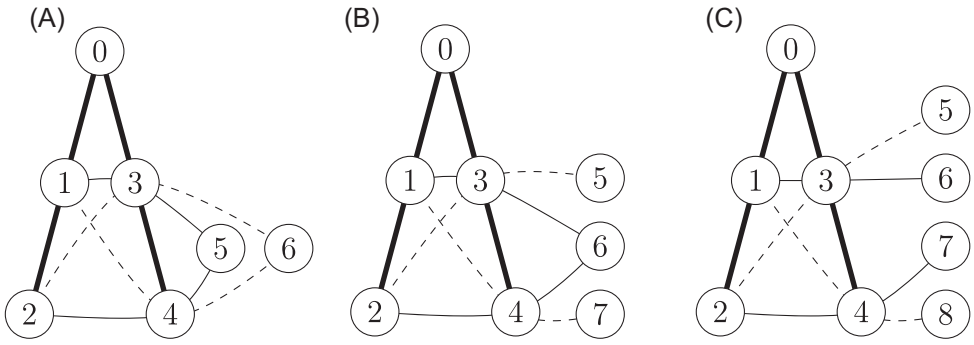

FIGURE 1 Connecting pairwise balanced designs to additional points: (A) $1^2 5^{14}$, (B) $1^1 3^2 5^{13}$, and (C) $3^4 5^{12}$.

The sum (3) contains the term $v! N_{v,1}$. Assuming the correctness of the classification of the STS(v)s with nontrivial automorphisms, this means that an incorrect number of labeled STSs will be accepted only if it differs from the correct number by a multiple of $v!$.

Assume that the value of $N_{\mathcal{D}}(G) N_{\mathcal{F}}(G, \mathcal{W})$ in (2) is incorrect, and denote the difference between the correct and the incorrect value by E. For such an error to go undetected $Kv!E/(w!|\mathrm{Aut}(G)|N')$ should be divisible by $v!$, that is, $KE/(w!|\mathrm{Aut}(G)|N')$ should be an integer. In the common situation of $K = w!$ and $|\mathrm{Aut}(G)| = 1$, we get that E should be divisible by N'. Also in other cases—including missing graphs, run-time errors, and multiple counting errors—divisibility of some integer by N' is a core question and gives an impression of the probability of errors going undetected.

3.5.1 | STS(21)

A classification of STS(21)s with nontrivial automorphisms was carried out by Kaski [17] and validated by double counting. The distribution of such systems based on the order of the automorphism groups is displayed in Table 3.

As argued above, the results produced by the software used in the computations, including `geng` and our own programs, are subject to the overall validation. For the divisibility argument discussed, we can be somewhat more specific in the case of the current parameters: STS(21)s, $w = 5$, and $\mathcal{B}'_3 = \{012, 034\}$.

For the degree sequences $1^2 5^{14}$, $1^1 3^2 5^{13}$, and $3^4 5^{12}$ and respective errors E_1, E_2, and E_3, the impact of these errors on the total sum is

$$\frac{6 \cdot 21! E_1}{945|\mathrm{Aut}(G)|}, \quad \frac{2 \cdot 21! E_2}{945|\mathrm{Aut}(G)|}, \quad \text{and} \quad \frac{21! E_3}{945|\mathrm{Aut}(G)|}.$$

As $|\mathrm{Aut}(G)| = 1$ for the vast majority of graphs, we are essentially asking whether $6E_1 + 2E_2 + E_3$ is divisible by 945. With a uniform distribution of that sum over all possible values mod 945, this probability would obviously be 1/945.

The graphs with degree sequence 5^{14} are regular graphs and their number can be found in the literature [32, 35].

TABLE 3 Steiner triple systems of order 21 with nontrivial automorphisms.

$\lvert\mathrm{Aut}(X)\rvert$	N	$\lvert\mathrm{Aut}(X)\rvert$	N	$\lvert\mathrm{Aut}(X)\rvert$	N
2	60,588,267	14	14	54	1
3	1,732,131	16	12	72	5
4	11,467	18	33	108	1
5	1772	21	10	126	2
6	2379	24	19	144	1
7	66	27	3	294	1
8	222	36	5	504	1
9	109	42	7	882	1
12	85	48	2	1008	1

4 | THE MAIN RESULT

The main computations were carried out in a cluster, and the computing times announced here are for computations on the equivalent of one core of a 2.4-GHz Intel Xeon E5-2665. The total computing time was approximately 82 core-years. The classification of the STS(21)s with nontrivial automorphisms in [17] took 25 core-hours with a 2-GHz CPU.

Due to the very large number of graphs with the given degree sequences, these were not saved, but they can at any point be reproduced with `geng`. The number of isomorphism classes of graphs with degree sequences in column S_2 in Table 2 is shown in Table 4 and grouped based on the order of the automorphism group and the degree sequence. Recall that, as described in Section 3.2, any graph with degree sequence S_2 can be extended in a unique way to a graph with the corresponding degree sequence S_1 in Table 2 (by adding vertices of degree one). Moreover, the order of the automorphism group remains unchanged in all but one of the extensions: it doubles when going from 5^{14} to $1^2 5^{14}$.

The last row in Table 4 shows the amount of core-hours for all computations in the respective cases, including the generation of graphs and the computation of $N_{\mathcal{D}}(G)$ and $N_{\mathcal{F}}(G, \mathcal{W})$.

In Table 5, we split the value of (2) into partial sums, one for each degree sequence. (Notice that we in general cannot assume that such partial sums are integers.)

By Table 5, the total number of labeled STS(21) is

$$755{,}952{,}181{,}048{,}907{,}354{,}964{,}715{,}609{,}522{,}176{,}000.$$

By (3) and Table 3, we then get that the number of isomorphism classes of STS(21)s with trivial automorphism group is

$$N_{21,1} = 14{,}796{,}207{,}455{,}537{,}154.$$

Finally, once again utilizing Table 3, this yields the total number of isomorphism classes of STS(21)s,

$$14{,}796{,}207{,}517{,}873{,}771,$$

TABLE 4 Graphs with certain degree sequences.

$\lvert Aut(G)\rvert$	5^{14}	$4^2 5^{12}$	$2^1 3^1 5^{13}$	$3^2 4^1 5^{12}$	$3^4 5^{12}$
1	3,063,687	143,619,210	714,665,364	9,352,704,447	50,268,563,872
2	344,187	11,440,917	52,316,559	565,093,004	3,265,605,905
3		47	24	50	1507
4	42,047	959,414	3,865,385	33,876,404	185,390,630
6	461	19,402	103,883	952,540	5,479,846
7	1				
8	6713	89,361	278,358	2,045,349	11,122,205
12	456	9664	42,587	290,483	1,373,479
14	5				
16	1103	8785	19,688	130,440	722,715
18					53
24	251	3088	9996	56,321	265,512
28	4				
32	209	993	1411	9181	53,049
36	4	89	359	1785	5509
48	114	767	1788	9455	46,312
60	2				1
64	37	123	113	740	4504
72	9	98	297	1082	3929
80	3				
96	30	161	318	1590	7661
108					1
120				1	3
128	14	24	6	75	526
144	11	60	101	359	1576
160					1
192	10	35	69	281	1420
216		3	3	6	41
240				1	5
256	5	8		5	74
288	7	20	35	116	489
320					1
384	2	8	7	40	248
432	1	4	2	7	54

(Continues)

TABLE 4 (Continued)

|Aut(G)|	5^{14}	$4^2 5^{12}$	$2^1 3^1 5^{13}$	$3^2 4^1 5^{12}$	$3^4 5^{12}$
480				1	7
512	1	1		1	17
576	2	11	10	35	160
720			8	94	434
768	2	3	2	4	37
864		2	1		26
960					1
1024				1	4
1152	1	1	1	8	53
1296					1
1440		3	13	86	283
1536			1		11
1728	1	1		1	17
1792	1				
1920					1
2048					1
2160					2
2304	1		1	3	13
2592					1
2880	1	5	12	40	133
3072					3
3456					3
3840					1
4096					1
4320				1	6
4608					5
5120					1
5760		2	2	9	36
6144		1			
6912					4
8640		2	3	5	6
9216					2
10,368					4
11,520	1		1	3	10

TABLE 4 (Continued)

$\lvert Aut(G)\rvert$	5^{14}	$4^2 5^{12}$	$2^1 3^1 5^{13}$	$3^2 4^1 5^{12}$	$3^4 5^{12}$	
13,824						1
17,280					1	4
18,432						1
23,040			1			4
25,920						3
34,560						4
43,200		1				
46,080						1
51,840			1			1
55,296						1
69,120						1
92,160		1				
207,360						1
460,800						1
24,883,200						1
Total	3,459,386	156,152,315	771,306,408	9,955,174,055	53,738,652,436	
Core-hours	80	1799	12,745	135,073	572,276	

TABLE 5 Partial sums of (2) per degree sequence.

S_1	S_2	Partial sum
$1^2 5^{14}$	5^{14}	133,088,588,244,979,214,201,168,855,040,000
$1^2 5^{14}$	$4^2 5^{12}$	2,538,696,865,871,668,928,235,196,907,520,000
$1^1 3^2 5^{13}$	$2^1 3^1 5^{13}$	10,154,787,463,486,675,712,940,787,630,080,000
$1^1 3^2 5^{13}$	$3^2 4^1 5^{12}$	77,792,298,507,007,219,219,438,529,150,976,000
$3^4 5^{12}$	$3^4 5^{12}$	665,333,309,624,296,811,889,899,926,978,560,000
Total		755,952,181,048,907,354,964,715,609,522,176,000

which completes a computer-aided proof of Theorem 1.

Using a random hypergraph model, it is conjectured in [13] that the proportion of STS(v)s containing at least one sub-STS(7) is $\alpha = 1 - e^{-1/168} \approx 0.00593$ for large v. By [13], the number of isomorphism classes of STS(21)s that contain at least one sub-STS(7) is 116,635,963,205,551. The real proportion for STS(21)s is hence

$$\frac{116,635,963,205,551}{14,796,207,517,873,771} \approx 0.00788 \approx 1.3\alpha,$$

which approximately coincides with the real proportion of 1.3α in the case of STS(19)s and is in line with the experimental results of [14].

The approach was also applied with the same main parameters ($w = 5$ and $\mathcal{B}'_3 = \{012, 034\}$) to the cases of STS(13)s, STS(15)s, and STS(19)s. The number of labeled STS(19)s was obtained in about 60 core-hours. We did not investigate whether $w = 5$ is the optimal choice for STS(19).

ACKNOWLEDGMENTS

The authors thank the referees for their valuable comments. Supported by the Academy of Finland, Grant 331044.

ORCID

Daniel Heinlein http://orcid.org/0000-0002-3429-3572
Patric R. J. Östergård http://orcid.org/0000-0003-0426-9771

REFERENCES

1. G. Brinkmann, *Fast generation of cubic graphs*, J. Graph Theory. **23** (1996), 139–149.
2. G. Brinkmann, J. Goedgebeur, and B. D. McKay, *Generation of cubic graphs*, Discrete Math. Theor. Comput. Sci. **13** (2011), no. 2, 69–79.
3. M. B. Cohen, C. J. Colbourn, L. A. Ives, and A. C. H. Ling, *Kirkman triple systems of order 21 with nontrivial automorphism group*, Math. Comp **71** (2002), 873–881.
4. C. J. Colbourn, *Triple systems*, Handbook of Combinatorial Designs (C. J. Colbourn and J. H. Dinitz, eds.), 2nd ed., Chapman & Hall/CRC Press, Boca Raton, FL, 2007, pp. 58–71.
5. C. J. Colbourn, A. D. Forbes, M. J. Grannell, T. S. Griggs, P. Kaski, P. R. J. Östergård, D. A. Pike, and O. Pottonen, *Properties of the Steiner triple systems of order 19*, Electron. J. Combin. **17** (2010), #R98.
6. C. J. Colbourn and A. Rosa, *Triple systems*, Clarendon Press, Oxford, 1999.
7. I. A. Faradžev, *Generation of nonisomorphic graphs with a given distribution of the degrees of vertices (in Russian)*, Algorithmic Studies in Combinatorics (in Russian) (I. A. Faradžev, ed.), Izdat. "Nauka", Moscow, 1978, pp. 11–19.
8. A. D. Forbes, M. J. Grannell, and T. S. Griggs, *Independent sets in Steiner triple systems*, Ars Combin. **72** (2004), 161–169.
9. T. S. Griggs, *Steiner triple systems and their close relatives*, Quasigroups Related Systems. **19** (2011), 23–68.
10. Y. Guan, M. Shi, and D. S. Krotov, *Steiner triple systems of order 21 with a transversal subdesign TD(3,6) (in Russian)*, Problemy Peredachi Informatsii. **56** (2020), no. 1, 23–32; English translation in Probl. Inf. Transm. 56 (2020), no. 1, 23–32.
11. S. L. Hakimi, *On realizability of a set of integers as degrees of the vertices of a linear graph, I*, J. Soc. Indust. Appl. Math. **10** (1962), 496–506.
12. V. Havel, *Eine Bemerkung über die Existenz der endlichen Graphen (in Czech)*, Časopis Pěst. Mat. **80** (1955), 477–480.
13. D. Heinlein and P. R. J. Östergård, *Steiner triple systems of order 21 with subsystems*, Glas. Mat. Ser. III, to appear.
14. D. Heinlein and P. R. J. Östergård, *Constructing random Steiner triple systems: An experimental study*, Stinson66 – New Advances in Designs, Codes and Cryptography (C. J. Colbourn and J. H. Dinitz, eds.), Springer Nature, Cham, Switzerland.
15. A. Hulpke, P. Kaski, and P. R. J. Östergård, *The number of Latin squares of order 11*, Math. Comp. **80** (2011), 1197–1219.
16. S. N. Kapralov and S. Topalova, *On the Steiner triple systems of order 21 with automorphisms of order 3*, Proceedings of the Third International Workshop on Algebraic and Combinatorial Coding Theory (Voneshta Voda, Bulgaria, June 22–28, 1992), pp. 105–108.
17. P. Kaski, *Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms*, SIAM J. Discrete Math. **19** (2005), 664–690.

18. P. Kaski, *Nonexistence of perfect Steiner triple systems of orders 19 and 21*, Bayreuth. Math. Schr. **74** (2005), 130–135.
19. P. Kaski and P. R. J. Östergård, *The Steiner triple systems of order 19*, Math. Comp. **73** (2004), 2075–2092.
20. P. Kaski and P. R. J. Östergård, *Classification algorithms for codes and designs*, Springer, Berlin, 2006.
21. P. Kaski and P. R. J. Östergård, *There are 1,132,835,421,602,062,347 nonisomorphic one-factorizations of* K_{14}, J. Combin. Des. **17** (2009), 147–159.
22. P. Kaski, P. R. J. Östergård, and A. Popa, *Enumeration of Steiner triple systems with subsystems*, Math. Comp. **84** (2015), 3051–3067.
23. P. Kaski, P. R. J. Östergård, S. Topalova, and R. Zlatarski, *Steiner triple systems of order 19 and 21 with subsystems of order 7*, Discrete Math. **308** (2008), 2732–2741.
24. P. Kaski and O. Pottonen, `libexact` *user's guide*, version 1.0, HIIT Technical Reports 2008-1, Helsinki Institute for Information Technology HIIT, 2008.
25. J. I. Kokkala and P. R. J. Östergård, *Kirkman triple systems with subsystems*, Discrete Math. **343** (2020), 111960.
26. J. I. Kokkala and P. R. J. Östergård, *Sparse Steiner triple systems of order 21*, J. Combin. Des. **29** (2021), 75–83.
27. C. W. H. Lam and Y. Miao, *Cyclically resolvable cyclic Steiner triple systems of order 21 and 39*, Discrete Math. **219** (2000), 173–185.
28. R. A. Mathon, K. T. Phelps and A. Rosa, *A class of Steiner triple systems of order 21 and associated Kirkman systems*, Math. Comp. **37** (1981), 209–222 and 64 (1995), 1355–1356.
29. R. Mathon and A. Rosa, *The 4-rotational Steiner and Kirkman triple systems of order 21*, Ars Combin. **17A** (1984), 241–250.
30. B. D. McKay, A. Meynert, and W. Myrvold, *Small Latin squares, quasigroups, and loops*, J. Combin. Des. **15** (2007), 98–119.
31. B. D. McKay and A. Piperno, *Practical graph isomorphism, II*, J. Symbolic Comput. **60** (2014), 94–112.
32. M. Meringer, *Fast generation of regular graphs and construction of cages*, J. Graph Theory. **30** (1999), 137–146.
33. P. R. J. Östergård and L. H. Soicher, *There is no McLaughlin geometry*, J. Combin. Theory Ser. A. **155** (2018), 27–41.
34. A. R. Rao, R. Jana, and S. Bandyopadhyay, *A Markov chain Monte Carlo method for generating random (0,1)-matrices with given marginals*, Sankhyā Ser. A. **58** (1996), 225–242.
35. G. Royle, *Graphs and multigraphs*, Handbook of Combinatorial Designs (C. J. Colbourn and J. H. Dinitz, eds.), 2nd ed., Chapman & Hall/CRC Press, Boca Raton, FL, 2007, pp. 731–740.
36. V. D. Tonchev, *Steiner triple systems of order 21 with automorphisms of order 7*, Ars Combin. **23** (1987), 93–96 and 39 (1995), 3.
37. S. Topalova, *STS(21) with automorphisms of order 3 with 3 fixed points and 7 fixed blocks*, Math. Balkanica (N.S.). **18** (2004), 215–221.
38. F. Viger, and M. Latapy, *Efficient and simple generation of random simple connected graphs with prescribed degree sequence*, Proceedings of the 11th Annual International Conference on Computing and Combinatorics (COCOON 2005; Kunming, August 16–19, 2005), LNCS 3595 (L. Wang, ed.), Springer, Berlin, 2005, pp. 440–449.

How to cite this article: D. Heinlein and P. R. J. Östergård, *Enumerating Steiner triple systems*, J. Combin. Des. (2023), 1–17. https://doi.org/10.1002/jcd.21906