Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Kokkala, Janne I.; Östergård, Patric R.J.

Sparse Steiner triple systems of order 21

Published in:
Journal of Combinatorial Designs

DOI:
10.1002/jcd.21757

Published: 01/02/2021

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Kokkala, J. I., & Östergård, P. R. J. (2021). Sparse Steiner triple systems of order 21. *Journal of Combinatorial Designs*, *29*(2), 75-83. https://doi.org/10.1002/jcd.21757

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Sparse Steiner Triple Systems of Order 21

Janne I. Kokkala*
Faculty of Engineering, LTH, Lund University
P.O. Box 118, SE-22100 Lund, Sweden
and
Department of Computer Science
University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark

Patric R. J. Östergård†
Department of Communications and Networking
Aalto University School of Electrical Engineering
P.O. Box 15400, 00076 Aalto, Finland

Abstract

A (k, l)-configuration is a set of l blocks on k points. For Steiner triple systems, $(l + 2, l)$-configurations are of particular interest. The smallest nontrivial such configuration is the Pasch configuration, which is a $(6, 4)$-configuration. A Steiner triple system of order v, an STS(v), is r-sparse if it does not contain any $(l + 2, l)$-configuration for $4 \leq l \leq r$. The existence problem for anti-Pasch Steiner triple systems has been solved, but these have been classified only up to order 19. In the current work, a computer-aided classification shows that there are 83003869 isomorphism classes of anti-Pasch STS(21)s. Exploration of the classified systems reveals that there are three 5-sparse STS(21)s but no 6-sparse STS(21)s. The anti-Pasch STS(21)s lead to 14 Kirkman triple systems, none of which is doubly resolvable.

Keywords: automorphism group, Kirkman triple system, Pasch configuration, quadrilateral, Steiner triple system.

*Supported in part by the Swedish Research Council grant 2016-00782.
†Supported in part by the Academy of Finland, Grant No. 289002

1 Introduction

A *Steiner triple system* $(V, \mathcal{B})$ of order v, briefly STS(v), is a collection $\mathcal{B}$ of 3-element subsets, called *blocks*, of a v-set V of *points*, such that every pair of points occurs in exactly one block. It is a classical result that a necessary and sufficient condition for an STS(v) to exist is that

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

Each point of an STS(v) occurs in exactly $(v-1)/2$ blocks, called the *replication number*. For more information about (Steiner) triple systems, see [3, 6].

A (k, l)-*configuration* is a set of l blocks whose union contains exactly k points. If no point occurs in exactly one block, then the configuration is said to be *full*. If all points occur in an even number of blocks, then the configuration is *even*.

Work by Erdős [7] has set a direction for some of the work on Steiner triple systems *lacking* certain configurations. An STS(v) with no $(l+2, l)$-configurations for any $4 \leq l \leq r$ is said to be r-*sparse*. Erdős [7] conjectured that for every integer r, there is a value $v_0(r)$ such that there exists an r-sparse STS(v) for every admissible order $v > v_0(r)$.

The smallest nontrivial $(l+2, l)$-configuration is a $(6, 4)$-configuration, called a *Pasch configuration* or a *quadrilateral*. The Pasch configuration is the smallest possible configuration with the property that the occurrence number in a Steiner triple system is not fixed by the order of the system. Hence, Pasch configurations are useful as invariants for Steiner triple systems (cf. [12]). The 4-sparse systems are exactly those that do not have Pasch configurations, that is, they are *anti-Pasch*. Anti-Pasch STS(v)s exist for all admissible orders (1) except $v \in \{7, 13\}$, see [9].

There are two $(7, 5)$-configurations, one of which, the *mitre configuration*, is full. The other $(7, 5)$-configuration, the *mia configuration*, is obtained as an extension of the Pasch configuration. The Pasch, mitre, and mia configurations are displayed in Figure 1.

Anti-mitre STS(v)s exist for all admissible orders (1) except $v = 13$, see [22]. A Steiner triple system is 5-sparse if and only if it is anti-Pasch and anti-mitre. Erdős's conjecture is still open for this case and is therefore open for all $r \geq 5$. Families of 5-sparse [18] and 6-sparse [8] Steiner triple systems have been discovered, but not a single nontrivial example is known for $r > 6$ (see, however, [1] for infinite systems).

One approach for studying combinatorial objects is via a classification for parameters up to what is computationally feasible. The Steiner triple systems of order 19 have been classified [12] and studied, for example, with

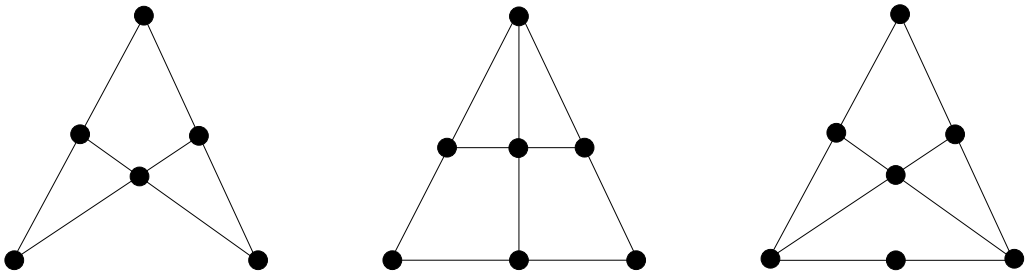

Figure 1: Pasch, mitre, and mia configurations

respect to the properties mentioned earlier [4]. Since a classification of the Steiner triple systems of order 21 is not available, we here aim at classifying sparse STS(21)s in a direct manner. The main results of this work show that there are 83003869 isomorphism classes of anti-Pasch STS(21)s. Exploration of the classified systems reveals that there are three isomorphism classes of 5-sparse STS(21)s but no 6-sparse STS(21)s. An analogous approach could be used to classify the anti-mitre STS(21)s.

The paper is organized as follows. The algorithm for classifying sparse Steiner triple systems is presented in Section 2. The results are tabulated in Section 3, where the classified designs are further investigated. For example, the anti-Pasch STS(21)s give 14 isomorphism classes of Kirkman triple systems of order 21, none of which is doubly resolvable.

2 Classification of Sparse Systems

Before we start the discussion of the classification algorithm, we need a few additional definitions. Two STS(v)s are *isomorphic* if there exists a bijection between the point sets that maps blocks onto blocks; such a bijection is an *isomorphism*. An *automorphism* of an STS(v) is an isomorphism of the triple system onto itself. The set of all automorphisms forms the *automorphism group* of the triple system under composition. These definitions can be generalized to sets of blocks.

We are here facing a computational problem that is more specific than that in [11], where Steiner triple systems with forbidden neighborhood graphs are considered, which in turn is more specific that that in [12], where Steiner triple systems are considered. It therefore makes sense to introduce the algorithm via those in [12] and [11].

In [12] an algorithm for classifying Steiner triple systems is presented and used to classify the STS(19)s. The algorithm has two stages, where the partial structures obtained in the first stage form seeds for the second stage. We will later see how the anti-Pasch property will affect both stages: the number of seeds will be reduced and there will be a speed-up of the final search stage.

Specifically, a *seed* when classifying STS(v)s is formed by a set of blocks over $\{1, 2, \ldots, v\}$ that contains the block $\{1, 2, 3\}$ and $(3v - 9)/2$ blocks of the form $\{i, j, k\}$, $1 \leq i \leq 3$, $4 \leq j < k \leq v$ that intersect pairwise in at most one point. Then the points 1, 2, and 3 occur in $(v-1)/2$ blocks, which equals the replication number.

Classifying the seeds for the orders considered here does not require extensive computational resources with proper algorithms (cf. [12] and [13, Example 6.20]) and tools [20]. Extending the seeds to complete Steiner systems, on the other hand, is very time-consuming already for these parameters. We shall now consider this subproblem for the different variants before discussing isomorph rejection.

The extension phase can be considered via instances of the exact cover problem. In the exact cover problem, we are given a finite set S and a family $\mathcal{C}$ of subsets of S, and the task is to find one or all partitions of S consisting of sets in $\mathcal{C}$. For $(\{1, 2, \ldots, v\}, \mathcal{B})$ Steiner triple systems of order v, we let

$$\begin{aligned} S &= \{\{i, j\} : 1 \leq i < j \leq v\}, \\ \mathcal{C} &= \{\{\{i, j\}, \{i, k\}, \{j, k\}\} : 1 \leq i < j < k \leq v\}, \end{aligned}$$

that is, there is one set of size 3 in $\mathcal{C}$ for each possible block $\{i, j, k\}$. Prescribing a seed just means that we require the corresponding sets in $\mathcal{C}$ to be a part of the solutions. To find one or all solutions of instances of the exact cover problem, one may use the libexact [14] software, which utilizes ideas from [15].

For two points $\{i, j\}$ of a Steiner triple system $(V, \mathcal{B})$, consider the 2-subsets in

$$\{\{x, y\} : \{x, y, z\} \in \mathcal{B},\ x, y \in V \setminus \{i, j\},\ z \in \{i, j\}\}.$$

We may consider these 2-subsets as the edges of a graph G with vertex set $V \setminus B$, where B is the block that contains i and j. The graph G is 2-regular and consists of cycles of even length. The shortest possible length of a cycle in G is then obviously 4. In fact, a cycle of length 4 corresponds to a Pasch configuration (Fig. 1), which has the form

$$\{\{i, l, m\}, \{i, n, o\}\{j, l, n\}, \{j, m, o\}\}$$

in set notation.

The 2-regular graph obtained in the above mentioned way is called the *neighborhood graph* with respect to the two specified vertices i and j. In [11], a general approach for classifying Steiner triple systems with forbidden neighborhood graphs is described. The algorithm is based on the general algorithm for classifying Steiner triple systems, described above, and maintains a dedicated data structure to expedite pruning. The libexact software supports such pruning via a feature that makes testing of partial solutions possible on every level of the search tree.

In this work we are not focusing on general neighborhood graphs but only on Pasch configurations. A tailored algorithm for this purpose will now be presented.

Our employed data structure for STS(v)s considers vectors of length v over the field GF(2). In this setting, blocks are vectors with exactly three 1s and correspond to columns of an incidence matrix. Addition of vectors is throughout this paper carried out componentwise over GF(2). In C and many other programming languages, vectors over GF(2) of the lengths we are considering here can be handled as integers, using a single bitwise exclusive OR operation for addition.

Now, since the Pasch configuration is the only even configuration in a Steiner triple system with four blocks, we have a Pasch configuration if and only if there are four block vectors that add to the all-zero vector. This is further related to the fact that a Pasch configuration corresponds to a codeword of weight 4 in the dual of the point code of the Steiner triple system.

We maintain a data structure containing all sums of one, two, and three vectors from the set of blocks included so far and update this on every level of the search tree. Naming the three sets S_i, $1 \leq i \leq 3$, we avoid Pasch configurations by requiring that the vector for a new block must not be in S_3.

Actually, it is not necessary to consider all such sums but only those that could lead to a Pasch configuration. It suffices to consider vectors in S_2 that have exactly four 1s and vectors in S_3 that have exactly three 1s. Moreover, we can utilize the following observation regarding blocks in a seed.

Lemma 1. *A Pasch configuration in an* STS(v) *constructed from a seed has an even number of blocks in the seed.*

Proof. For a seed and an STS(v) constructed from that seed, label the points so that $\{1, 2, 3\}$ is the block that intersects all blocks of the seed.

In a Pasch configuration, each point occurs in two blocks and each block intersects all the other blocks. If the block $\{1, 2, 3\}$ is in a Pasch configuration

$\mathcal{P}$, then the three other blocks of $\mathcal{P}$ intersect this block and are in the seed, so $\mathcal{P}$ has four (that is, all) blocks in the seed.

If the block $\{1, 2, 3\}$ is not in $\mathcal{P}$, then each block of $\mathcal{P}$ contains at most one of the points in $\{1, 2, 3\}$. Since each point occurs in zero or two blocks of $\mathcal{P}$, the total number of blocks of $\mathcal{P}$ in the seed is a sum of even numbers, which is even. □

We can obviously discard all seeds that contain a Pasch configuration. When starting the search from some seed, S_1 and S_3 can be initialized to $\emptyset$ by Lemma 1. When forming S_2 from the blocks of the seed we may ignore the block $\{1, 2, 3\}$ and further only consider pairs of blocks that intersect in one of the points in $\{1, 2, 3\}$.

After extending the seeds in all possible ways to Steiner triple systems, the task of isomorph rejection still remains. To this end, the technique of *canonical augmentation* [19] is convenient and can be used for all discussed variants of the problem. Note, however, that Pasch configurations are used as invariants in the classification of STS(19)s in [12], and this invariant is of course useless in the anti-Pasch case. But having a good invariant is actually not of practical importance in the case of anti-Pasch systems, which are produced at a rather low rate by the exact cover algorithm.

Canonical augmentation consists of two tests, both of which should be passed for a complete object to be accepted. When objects are constructed from seeds in all possible ways, the first test is about considering substructures of the complete object that are isomorphic to seeds and asking whether the seed from which it was constructed, the *parent*, is a *canonical parent*. In our case, the substructures isomorphic to seeds are exactly the configurations obtained by taking any block B and all blocks that intersect B. In the second test, the objects with the same canonical parent should further be tested for isomorphism.

In the case of Steiner triple systems, canonical augmentation is conveniently carried out in the context of block graphs. A *block graph* is a graph with one vertex for each block and edges whenever the corresponding blocks intersect. When $v \geq 19$, an STS(v) can be reconstructed up to isomorphism from its block graph (cf. [12, Theorem 2.2]), so isomorph rejection may indeed be carried out in the context of block graphs. Now the test regarding canonical parents—comparing blocks—becomes a test of vertices: is the vertex of $\{1, 2, 3\}$ a canonical vertex or not? The graph isomorphism software *nauty* [20] can be used to give one canonical order of the (orbits of) vertices. See [19] and [13, Sect. 4.2.3] for details on canonical augmentation in general and [12] and [13, Sect. 6.1.4] for details on an application to Steiner triple systems.

3 Results

3.1 Classification

In the classification of all Steiner triple systems, there are up to isomorphism 14648 and 219104 seeds for order 19 and 21, respectively [12]. The number of anti-Pasch such seeds is 1678 and 25796, respectively, where the former number is from [11] and the latter is from the current work.

The developed program was tested on the anti-Pasch STS(19)s, and the classification of those 2591 designs took about 2 hours with a 3.6-GHz CPU. The more general algorithm in [11] needed about one day with a 2-GHz CPU for this task, so the tailored algorithm indeed speeds up the classification in the anti-Pasch case.

In the main computation, a cluster of 2.3–2.7-GHz Intel Xeon processors with a total of 672 cores was used. The classification, which can conveniently be split into one task for each of the 25796 seeds, was completed in about 4 weeks, that is, using a total of roughly 50 core-years. The classified designs can be obtained from [17], where we also make the anti-Pasch STS(19)s and various other data from the current paper available.

Theorem 1. *There are* 83003869 *isomorphism classes of anti-Pasch* STS(21)*s.*

The number of isomorphism classes of anti-Pasch STS(21)s for each possible order of the automorphism group is shown in Table 1. In the classification of STS(21)s with nontrivial automorphisms in [10], the number of anti-Pasch such systems were counted. Hence only the number of anti-Pasch STS(21)s with no nontrivial automorphisms has been unknown.

To validate the obtained result, double counting was here used as in [11, 12]. Let $\mathcal{S}_i$, $1 \le i \le 25796$, denote representatives of isomorphism classes of seeds and let $\mathcal{D}_i$, $1 \le i \le 83003869$, denote representatives of isomorphism classes of anti-Pasch STS(21)s. Further let M_i denote the total number of anti-Pasch STS(21)s—that is, the number before isomorph rejection—that can be obtained by extending the seed $\mathcal{S}_i$. Utilizing the orbit–stabilizer theorem, we must now have

$$\frac{1}{70}\sum_{i=1}^{25796}\frac{M_i \cdot 21!}{|\mathrm{Aut}(\mathcal{S}_i)|} = \sum_{i=1}^{83003869}\frac{21!}{|\mathrm{Aut}(\mathcal{D}_i)|}.$$

From Table 1 we get the values needed for the right-hand side, which evaluates to 424071392094042675049267200 0. We also got the same value by using data from the search in the left-hand side.

Table 1: Anti-Pasch STS(21)s

|Aut|	#STS
1	83002911
2	123
3	792
5	24
6	4
7	8
9	3
18	1
21	3
Total	83003869

3.2 Properties

The anti-Pasch STS(21)s can now be tested for various properties.

Theorem 2. *There are three isomorphism classes of* 5*-sparse* STS(21)*s but no* 6*-spare* STS(21)*s. The automorphism groups of the* 5*-sparse* STS(21)*s have orders* 9*,* 21*, and* 21*.*

We list the three 5-sparse STS(21)s separately in [17]. A 5-sparse STS(21) has earlier been discovered by Wolfe [21, Fig. C.1]; the order of the automorphism group of that system is 9. It is known [5] that there is a cyclic STS(21) that is anti-Pasch but not anti-mitre and a cyclic STS(21) that is anti-mitre but not anti-Pasch. It then follows from Table 1 and Theorem 2 that the former is the only anti-Pasch STS(21) with an automorphism group of order 21 that is not anti-mitre. The automorphism groups of two of the 5-sparse STS(21)s are isomorphic to the non-abelian group of order 21.

Another property worth investigating is resolvability. A partition of all points into blocks of an STS(v) is called a *parallel class*, and a partition of all blocks of an STS(v) into parallel classes is a *resolution*. An STS(v) that has at least one resolution is said to be *resolvable*. An STS(v) together with a resolution of its blocks is a *Kirkman triple system*, briefly KTS(v). An isomorphism between Kirkman triple system is an isomorphism between the underlying Steiner triple system that also maps parallel classes to parallel classes. Hence the automorphism group of a Kirkman triple system is a subgroup of the automorphism group of the underlying Steiner triple system.

The anti-Pasch STS(21)s were tested for resolvability and used to classify the anti-Pasch KTS(21)s. The result is presented in Table 2. There are

only 14 anti-Pasch KTS(21)s, six of which have nontrivial automorphisms (a smaller number is claimed in [2] and corrected in [16]).

Table 2: Anti-Pasch KTS(21)s

|Aut|	#KTS
1	8
3	6
Total	14

Table 3: Underlying STS(21)s

#KTS	#STS
1	12
2	1

Table 4: Group orders of underlying STS(21)s

|Aut|	#STS
1	8
3	5

Table 3 shows the distribution of the numbers of isomorphism classes of anti-Pasch KTS(21)s amongst the underlying anti-Pasch STS(21)s, and Table 4 gives the orders of the automorphism groups of the anti-Pasch STS(21)s underlying the anti-Pasch KTS(21)s.

The last table, Table 5, categorizes the anti-Pasch KTS(21)s based on the number of parallel classes orthogonal to all parallel classes of the system (#OPC). This information is useful for the study of doubly resolvable designs. An STS(v) is said to be *doubly resolvable* if it has two resolutions such that every pair of parallel classes, one from each of the resolutions, intersects in zero or one blocks. A necessary (but not sufficient) condition for a doubly resolvable STS(v) to exist is that the there exist at least $(v-1)/2$ parallel classes that are orthogonal to all parallel classes of a resolution. Since the largest value in the #OPC column of Table 5 is 3, which is smaller than

Table 5: Orthogonal parallel classes

#OPC	#KTS
0	6
1	4
2	2
3	2

$(v-1)/2 = 10$, the question whether there is a doubly resolvable STS(21) remains open after the current study.

References

[1] K. M. Chicot, M. J. Grannell, T. S. Griggs, and B. S. Webb, On sparse countably infinite Steiner triple systems, *J. Combin. Des.* **18** (2010), 115–122.

[2] 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.

[3] 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.

[4] 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.

[5] C. J. Colbourn, E. Mendelsohn, A. Rosa, and J. Širáň, Anti-mitre Steiner triple systems, *Graphs Combin.* **10** (1994), 215–224.

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

[7] P. Erdős, Problems and results in combinatorial analysis, in *Colloquio Internazionale sulle Teorie Combinatorie. Tomo II.* (Rome, 3–15 September 1973), Atti dei Convegni Lincei, No. 17, Accademia Nazionale dei Lincei, Rome, 1976, pp. 3–17.

[8] A. D. Forbes, M. J. Grannell, and T. S. Griggs, On 6-sparse Steiner triple systems, *J. Combin. Theory Ser. A* **114** (2007), 235–252.

[9] M. J. Grannell, T. S. Griggs, and C. A. Whitehead, The resolution of the anti-Pasch conjecture, *J. Combin. Des.* **8** (2000), 300–309.

[10] P. Kaski, Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms, *SIAM J. Discrete Math.* **19** (2005), 664–690.

[11] P. Kaski, Nonexistence of perfect Steiner triple systems of orders 19 and 21, *Bayreuth. Math. Schr.* **74** (2005), 130–135.

[12] P. Kaski and P. R. J. Östergård, The Steiner triple systems of order 19, *Math. Comp.* **73** (2004), 2075–2092.

[13] P. Kaski and P. R. J. Östergård, *Classification Algorithms for Codes and Designs*, Springer, Berlin, 2006.

[14] P. Kaski and O. Pottonen, libexact user's guide, version 1.0, HIIT Technical Reports 2008-1, Helsinki Institute for Information Technology HIIT, 2008.

[15] D. E. Knuth, Dancing links, in J. Davies, B. Roscoe, and J. Woodcock (Eds.), *Millennial Perspectives in Computer Science*, Palgrave, Houndmills, 2000, pp. 187–214.

[16] J. I. Kokkala and P. R. J. Östergård, Kirkman triple systems with subsystems, *Discrete Math.* **343** (2020), 111960.

[17] J. I. Kokkala and P. R. J. Östergård, Dataset for Sparse Steiner triple systems of order 21 [Dataset]. Zenodo. `https://doi.org/10.5281/zenodo.3899950` (June 18, 2020).

[18] H. Lefmann, K. T. Phelps, and V. Rödl, Extremal problems for triple systems, *J. Combin. Des.* **1** (1993), 379–394.

[19] B. D. McKay, Isomorph-free exhaustive generation, *J. Algorithms* **26** (1998), 306–324.

[20] B. D. McKay and A. Piperno, Practical graph isomorphism, II, *J. Symbolic Comput.* **60** (2014), 94–112.

[21] A. J. Wolfe, 5-sparse Steiner triple systems, Ph.D. thesis, Ohio State University, 2005.

[22] A. Wolfe, The resolution of the anti-mitre Steiner triple system conjecture, *J. Combin. Des.* **14** (2006), 229–236.