THE STEINER TRIPLE SYSTEMS OF ORDER 19

PETTERI KASKI AND PATRIC R. J. ÖSTERGÅRD

ABSTRACT. Using an orderly algorithm, the Steiner triple systems of order 19 are classified; there are 11,084,874,829 pairwise nonisomorphic such designs. For each design, the order of its automorphism group and the number of Pasch configurations it contains are recorded; 2,591 of the designs are anti-Pasch. There are three main parts of the classification: constructing an initial set of blocks, the seeds; completing the seeds to triple systems with an algorithm for exact cover; and carrying out isomorph rejection of the final triple systems. Isomorph rejection is based on the graph canonical labeling software *nauty* supplemented with a vertex invariant based on Pasch configurations. The possibility of using the (strongly regular) block graphs of these designs in the isomorphism tests is utilized. The aforementioned value is in fact a lower bound on the number of pairwise nonisomorphic strongly regular graphs with parameters $(57, 24, 11, 9)$.

1. INTRODUCTION

A *Steiner triple system* of order v, briefly $\mathrm{STS}(v)$, is a set of 3-element subsets, called *blocks*, of a v-set of *points*, such that every pair of points occurs in exactly one block. Given an $\mathrm{STS}(v)$, standard counting arguments show that each point must occur in exactly $r = (v-1)/2$ blocks, and that the triple system consists of exactly $b = v(v-1)/6$ blocks. Since r and b are integers, we get necessary conditions for the existence of an $\mathrm{STS}(v)$, which in fact turn out to be sufficient.

Theorem 1.1. *For $v \geq 3$, an $\mathrm{STS}(v)$ exists if and only if either $v \equiv 1 \pmod 6$ or $v \equiv 3 \pmod 6$.*

For a survey of this result and various other aspects of triple systems, the reader is referred to [4].

Two STS 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 a triple system is an isomorphism of the triple system onto itself. The *automorphism group* of the triple system consists of all of its automorphisms.

The number of pairwise nonisomorphic $\mathrm{STS}(v)$ is denoted by $N(v)$. It is known that $N(v)$ grows exponentially; in [23] it is proved (subject to the van der Waerden Permanent Conjecture, which is proved in [6, 7]) that

$$(e^{-5}v)^{v^2/6} \leq N(v) \leq (e^{-1/2}v)^{v^2/6}.$$

2000 *Mathematics Subject Classification.* Primary 05B07; Secondary 05E30, 51E10, 68R10.

Key words and phrases. automorphism group, orderly algorithm, Pasch configuration, Steiner triple system.

The research was supported in part by the Academy of Finland under grants 44517 and 100500. The work of the first author was partially supported by Helsinki Graduate School in Computer Science and Engineering (HeCSE) and a grant from the Foundation of Technology, Helsinki, Finland (Tekniikan Edistämissäätiö).

The first few nonzero values of the function $N(v)$ are $N(3) = 1$, $N(7) = 1$, $N(9) = 1$, $N(13) = 2$, and $N(15) = 80$. All these values were obtained in the days of hand calculations. The (correct) manual calculation of $N(15)$ in the 1910s is a remarkable achievement [5, 22]; it took almost 40 years before this result could be confirmed in one of the first computer-aided results in combinatorics [10]. Perhaps even more remarkable is that it would take almost a century before the next open value, $N(19)$, could be settled, in this paper.

As long as a complete classification of Steiner triple systems of order 19 has been lacking, various results on such designs with special properties have been studied. The results obtained include classifications of STS(19) with nontrivial automorphism group [3] and of STS(19) holding a STS(9) subdesign [20]. The former of these makes a partial correctness check of our results possible; it turns out that the results in [3] are partly in error.

Some basic results on Steiner triple systems needed in the determination of $N(19)$ are presented in Section 2. In particular, a connection between Steiner triple systems and strongly regular graphs is of central importance. The main algorithm used is presented in Section 3, whose central themes are the general backtrack algorithm and isomorph rejection. The statistics of the search, which took approximately two CPU years, are presented in Section 4. For each design found, the order of the automorphism group and the number of Pasch configurations it contains are tabulated. The total number of pairwise nonisomorphic Steiner triple systems of order 19 is $N(19) = 11{,}084{,}874{,}829$. The total number of Steiner triple systems of order 19 is 1,348,410,350,618,155,344,199,680,000. Using data obtained in the computer search, this number can be calculated in two different ways. The paper is concluded by roughly estimating the resources needed to calculate $N(21)$.

2. Preliminaries

All graphs in this paper are undirected. The point set $\{1, \ldots, v\}$ is used for all STS(v), and the vertex set $\{1, \ldots, b\}$ is used for all graphs of order b. The symmetric group on $\{1, \ldots, b\}$ is denoted by S_b. The automorphism group of a graph G is denoted by $\mathrm{Aut}(G)$. Similarly, the automorphism group of a triple system $\mathcal{B}$ is denoted by $\mathrm{Aut}(\mathcal{B})$.

Given a labeling $\mathcal{B} = \{B_1, \ldots, B_b\}$ of the blocks of an STS(v), the *block automorphism group* of the STS(v) consists of all permutations $\tau \in S_b$ for which there exists a $\mu \in \mathrm{Aut}(\mathcal{B})$ such that $\mu(B_j) = B_{\tau(j)}$ for all $j \in \{1, \ldots, b\}$. From the incidence matrix representation of an STS(v) it is easy to see that the block automorphism group is isomorphic to $\mathrm{Aut}(\mathcal{B})$ for $v \neq 3$.

2.1. Block Graphs.

A *block graph* or *line graph* of an STS is a graph whose vertices are in a one-to-one correspondence with the blocks of the triple system, with two vertices joined by an edge if and only if the corresponding blocks have nonempty intersection. More formally, if we label the blocks of an STS(v) as $B_1, \ldots, B_b$, then the block graph (subject to this labeling) over the vertex set $\{1, \ldots, b\}$ contains the edge $\{i, j\}$ if and only if $B_i \cap B_j \neq \emptyset$. Whenever we fix a labeling of the blocks, we assume that the block graph has been constructed as above.

A *strongly regular graph* with parameters (n, d, λ, μ), briefly $\mathrm{srg}(n, d, \lambda, \mu)$, is a graph of order n in which each pair of vertices has exactly d, λ, or μ common neighbours depending on whether the vertices are equal, adjacent, or nonadjacent, respectively.

Straightforward counting arguments establish the following well-known theorem.

Theorem 2.1. *A block graph of an* STS(v) *is a strongly regular graph* $\mathrm{srg}(v(v-1)/6, 3(v-3)/2, (v+3)/2, 9)$.

The converse of this theorem holds on condition that v is large enough. This result was obtained by Bose [1] in the early 1960s, for $v > 67$. The following theorem is one of the central building blocks of our classification approach.

Theorem 2.2. *For $v \geq 19$, every* STS(v) *can be reconstructed up to isomorphism from its block graph.*

The proof of [17, Theorem 10] contains an explicit algorithm for constructing an STS from a strongly regular graph. We do not, however, need such an algorithm here, since every such graph will be explicitly constructed from an STS, and the transformation between these objects is therefore known.

The following corollary is an immediate consequence of Theorem 2.2.

Corollary 2.3. *For $v \geq 19$, two* STS(v) *are isomorphic if and only if their block graphs are isomorphic.*

The following results are well-known.

Lemma 2.4. *For $v \geq 19$, a block graph of an* STS(v) *contains exactly v r-cliques.*

Proof. The vertices of an r-clique in the block graph clearly correspond to a set of r blocks with pairwise nonempty intersection. A short case-by-case analysis shows that a set of blocks with pairwise nonempty intersection in an STS(v) has size at most 7 unless the blocks share a common point. Since each point occurs in r blocks, and no two blocks contain the same point pair, there are exactly v sets of r blocks with pairwise nonempty intersection when $r > 7$, that is, $v \geq 19$. □

Theorem 2.5. *For $v \geq 19$, the automorphism group of an* STS(v) *and the automorphism group of a corresponding block graph are isomorphic.*

Proof. Label the blocks of an STS(v) as $\mathcal{B} = \{B_1, \ldots, B_b\}$, and consider the associated block graph G. Clearly, the block automorphism group of the STS is a subgroup of $\mathrm{Aut}(G)$. The converse also holds for $v \geq 19$ by Lemma 2.4. Namely, a $\tau \in \mathrm{Aut}(G)$ must then permute the vertex sets of the v r-cliques in G. Since the r-cliques correspond to the points of the underlying STS, we obtain a $\mu \in \mathrm{Aut}(\mathcal{B})$ such that $\mu(B_j) = B_{\tau(j)}$ for all $j \in \{1, \ldots, b\}$. Consequently, τ is a block automorphism of $\mathcal{B}$. □

Corollary 2.6. *Let $\mathcal{B} = \{B_1, \ldots, B_b\}$ and $\mathcal{B}' = \{B'_1, \ldots, B'_b\}$ be isomorphic* STS(v)*, where $v \geq 19$, and suppose G and G' are the associated block graphs. Then, for every isomorphism $\tau \in S_b$ of G onto G', there exists a $\nu \in S_v$ such that $\nu(B_j) = B'_{\tau(j)}$ holds for all $j \in \{1, \ldots, b\}$.*

2.2. **Canonical Labeling of Graphs Using *nauty*.** Our isomorph rejection strategy for STS(v) uses the graph canonical labeling software *nauty* [12, 13]. We need the following definitions and facts about *nauty* to enable subsequent discussion.

An *ordered partition* of a nonempty finite set V is a sequence $(V_1, \ldots, V_m)$ of nonempty pairwise disjoint subsets of V whose union is V; the sets $V_1, \ldots, V_m$ are called *cells* of the partition. Associated with each ordered partition $\Gamma =$

$(V_1,\ldots,V_m)$ of V is a *canonical* ordered partition $c(\Gamma) := (W_1,\ldots,W_m)$, whose cells are defined by

$$W_k := \{t_k+1, t_k+2,\ldots,t_k+|V_k|\},\quad t_1 := 0,\quad t_k := |V_1|+\cdots+|V_{k-1}|,$$

for all $k \in \{1,\ldots,m\}$. A permutation τ of V acts on an ordered partition $\Gamma = (V_1,\ldots,V_m)$ by $\tau(\Gamma) := (\tau(V_1),\ldots,\tau(V_m))$.

Fact 2.7. For a graph G and an ordered partition Γ of the vertex set $\{1,\ldots,b\}$ of G, *nauty* computes a *canonical labeling* $\sigma_{G,\Gamma} \in S_b$ and generators for the group $\mathrm{Aut}_\Gamma(G) := \{\alpha \in \mathrm{Aut}(G) \ : \ \alpha(\Gamma) = \Gamma\}$. The canonical labeling satisfies

$$\sigma_{G,\Gamma}(G) = \sigma_{\tau(G),\tau(\Gamma)}(\tau(G)) \tag{1}$$

$$\sigma_{G,\Gamma}(\Gamma) = c(\Gamma) \tag{2}$$

for all $\tau \in S_b$, graphs G, and ordered partitions Γ of the vertex set $\{1,\ldots,b\}$.

Let $\mathscr{G}$ be a set of graphs over the vertex set $\{1,\ldots,b\}$, and suppose that $\mathscr{G}$ is closed under permutation of the vertices. A *vertex invariant* for $\mathscr{G}$ associates to each graph $G \in \mathscr{G}$ an ordered partition $\Lambda(G)$ of $\{1,\ldots,b\}$ such that $\Lambda(\tau(G)) = \tau(\Lambda(G))$ holds for all $G \in \mathscr{G}$ and $\tau \in S_b$.

Theorem 2.8. *Let Λ be a vertex invariant for $\mathscr{G}$. Then, $\sigma_{G,\Lambda(G)}$ satisfies*

$$\sigma_{G,\Lambda(G)}(G) = \sigma_{\tau(G),\Lambda(\tau(G))}(\tau(G)) \tag{3}$$

for all graphs $G \in \mathscr{G}$ and $\tau \in S_b$. Moreover, $\mathrm{Aut}(G) = \mathrm{Aut}_{\Lambda(G)}(G)$ for all $G \in \mathscr{G}$.

Proof. Equality in (3) is an immediate consequence of (1) and the definition of a vertex invariant. For the second part, observe that $\alpha\Lambda(G) = \Lambda(G)$ for all $\alpha \in \mathrm{Aut}(G)$. □

To lighten the notation, we omit the vertex invariant whenever it is not explicitly required. For example, (3) then becomes

$$\sigma_G(G) = \sigma_{\tau(G)}(\tau(G)). \tag{4}$$

2.3. **Pasch Configurations.** A *Pasch configuration* (or *fragment* or *quadrilateral*) in an STS is a set of four blocks and six points of the form $\{u,v,w\}$, $\{w,x,y\}$, $\{u,x,z\}$, $\{v,y,z\}$. Algorithms for finding the Pasch configurations in a Steiner triple system are considered in [18]. An STS is said to be *anti-Pasch* if it does not contain a Pasch configuration. Note that the blocks in a Pasch configuration have pairwise nonempty intersection. However, all sets of four blocks that have pairwise nonempty intersection do not form a Pasch configuration. A detailed study of four-line configurations in Steiner triple systems appears in [8]; the following theorem suffices for our purposes.

Theorem 2.9. *Each block of an* $\mathrm{STS}(v)$ *occurs in exactly* $(v-3)(v^2-12v+99)/16$ *sets of four blocks that have pairwise nonempty intersection but do not form a Pasch configuration.*

Proof. Consider any set of four blocks with pairwise nonempty intersection. A short case-by-case analysis shows that, unless the blocks form a Pasch configuration, there exists a unique point x that occurs in at least three of the four blocks. Fix any block B of an $\mathrm{STS}(v)$. We count the sets of four blocks of the above type in which B occurs by splitting the count into subcases as follows.

Case 1. *The point x has multiplicity* 3*, and $x \in B$.* First, there are 3 possibilities for $x \in B$. Second, there are $r-1$ choices for a block B_1 that contains x. Third, there are 4 choices for a block B_2 that has nonempty intersection with both B and B_1 but $x \notin B_2$. Block B_3 that contains x and has nonempty intersection with B_2 is unique. Since the choices for B_1 and B_3 can be interchanged, the total number of sets of four blocks that contain B in this subcase is $6(r-1)$.

Case 2. *The point x has multiplicity* 3*, and $x \notin B$.* There are $v-3$ choices for x. Since x is unique, the three blocks in addition to B are uniquely determined as the blocks that contain $\{x, y\}$, where $y \in B$.

Case 3. *The point x has multiplicity* 4. Clearly, $x \in B$. There are $3\binom{r-1}{3}$ sets of four blocks that contain B in which $x \in B$ occurs with multiplicity 4. Namely, there are 3 ways to choose x and $\binom{r-1}{3}$ ways to choose the other 3 blocks after x has been fixed.

Since the subcases are nonoverlapping, we have that B occurs in exactly

$$6(r-1) + (v-3) + 3\binom{r-1}{3} = \frac{(v-3)(v^2-12v+99)}{16}$$

sets of four blocks that have pairwise nonempty intersection but do not form a Pasch configuration. □

The 4-cliques of a block graph are divided into those that correspond to Pasch configurations and those considered in Theorem 2.9. Fix a labeling $B_1, \ldots, B_b$ of the blocks of an STS(v), and let G be the associated block graph. Denote by $C_4(j)$ the number of 4-cliques in G that contain vertex j, and denote by $P(B_j)$ the number of Pasch configurations in the STS that contain B_j. Then,

$$C_4(j) = \frac{(v-3)(v^2-12v+99)}{16} + P(B_j). \tag{5}$$

Motivated by this observation, we can build a vertex invariant for block graphs by partitioning the vertices according to the number of Pasch configurations in which a block occurs. The Pasch configuration invariant is faster to compute than an invariant that partitions the vertices based on the number of 4-cliques because we need not consider the 4-cliques arising from other four-block configurations. Theorem 2.9 shows that when $v = 19$ there are already 232 such "redundant" 4-cliques for each vertex.

Let G be any block graph of an STS(v), and let $\mathcal{B} = \{B_1, \ldots, B_b\}$ be any STS(v) for which G is the associated block graph. Let $\{p_1, \ldots, p_m\} := \{P(B_j) : j \in \{1, \ldots, b\}\}$, where $p_1 > p_2 > \cdots > p_m$. Define the ordered partition $\Lambda_P(G) := (V_1, \ldots, V_m)$ by $V_i := \{j \in \{1, \ldots, b\} : P(B_j) = p_i\}$.

Theorem 2.10. *For $v \geq 19$, the function Λ_P is well-defined and constitutes a vertex invariant for the set of all block graphs derived from* STS(v).

Proof. Let G be a block graph, and let $\mathcal{B} = \{B_1, \ldots, B_b\}$ be any STS(v) for which G is the associated block graph. Select any $\tau \in S_b$. (To establish well-definedness, take the identity permutation.) Put $G' := \tau(G)$, and let $\mathcal{B}' = \{B'_1, \ldots, B'_b\}$ be any STS(v) for which G' is the associated block graph. Suppose that $\Lambda_P(G) = (V_1, \ldots, V_m)$ and $\Lambda_P(G') = (V'_1, \ldots, V'_{m'})$. By Corollary 2.3, $\mathcal{B}$ and $\mathcal{B}'$ are isomorphic. So, Corollary 2.6 implies that there exists a $\nu \in S_v$ such that $\nu(B_j) = B'_{\tau(j)}$ holds for all $j \in \{1, \ldots, b\}$. Thus, $P(B_j) = P(B'_{\tau(j)})$ holds for all $j \in \{1, \ldots, b\}$, which implies $m = m'$, $p_i = p'_i$, and $\tau(V_i) = \{\tau(j) : P(B_j) = p_i\} = \{j :$

1	11111111	00000000	00000000
1	00000000	11111111	00000000
1	00000000	00000000	11111111
0	10000000		
0	10000000		
0	01000000		
0	01000000		
0	00100000		
0	00100000		
0	00010000	A	B
0	00010000		
0	00001000		
0	00001000		
0	00000100		
0	00000100		
0	00000010		
0	00000010		
0	00000001		
0	00000001		

FIGURE 1. Structure of seeds.

$P(B'_j) = p'_i\} = V'_i$ for all $i \in \{1, \ldots, m\}$. Consequently, Λ_P is well-defined, and $\tau(\Lambda_P(G)) = \Lambda_P(\tau(G))$. □

3. Classification of Steiner Triple Systems

The correspondence between Steiner triple systems and strongly regular graphs given by Theorem 2.2 is very useful from an algorithmic point of view since we can alternate between representations and use the best representation for the task at hand. The core of our algorithm is an efficient exact cover algorithm [11] for constructing STS(19). Isomorph-free generation [15] is achieved using the block graph representation and *nauty* supplemented with the Pasch configuration vertex invariant. The details of our approach are as follows.

The construction process has two stages. The first stage is a preprocessing stage in which the seeds for the main search are determined. The second stage consists of determining all extensions of each seed to an STS(19) and rejecting isomorphs.

3.1. **Constructing and Extending Seeds.** In the preprocessing stage, we fix the first block, $\{1, 2, 3\}$, and construct all pairwise nonisomorphic designs consisting of 3-element blocks that intersect the first block so that the total number of blocks is 25 ($r = 9$ for an STS(19)) and no pair in $\{1, 2, \ldots, 19\}$ occurs in more than one block. Up to isomorphism, the incidence matrix of such a design is as shown in Figure 1.

To fill out the parts A and B in Figure 1, a backtrack search with isomorph rejection is carried out. Actually, the A part can be completed up to isomorphism using combinatorial arguments; there are only seven such completions, which are shown in Figure 2. From left to right these correspond to the seven partitions

$$4+4+4+4, \quad 4+4+8, \quad 4+6+6, \quad 4+12, \quad 6+10, \quad 8+8, \quad 16$$

10000000	10000000	10000000	10000000	10000000	10000000	10000000
01000000	01000000	01000000	01000000	01000000	01000000	01000000
10000000	10000000	10000000	10000000	10000000	10000000	10000000
01000000	01000000	01000000	01000000	00100000	00100000	00100000
00100000	00100000	00100000	00100000	01000000	01000000	01000000
00010000	00010000	00010000	00010000	00100000	00010000	00010000
00100000	00100000	00100000	00100000	00010000	00100000	00100000
00010000	00010000	00001000	00001000	00001000	00010000	00001000
00001000	00001000	00010000	00010000	00010000	00001000	00010000
00000100	00000100	00001000	00000100	00000100	00000100	00000100
00001000	00001000	00000100	00001000	00001000	00001000	00001000
00000100	00000010	00000010	00000010	00000010	00000010	00000010
00000010	00000100	00000100	00000100	00000100	00000100	00000100
00000001	00000001	00000001	00000001	00000001	00000001	00000001
00000010	00000010	00000010	00000010	00000010	00000010	00000010
00000001	00000001	00000001	00000001	00000001	00000001	00000001

FIGURE 2. Possible choices for the A matrix.

of 16 into even integers greater than or equal to 4. (Each completion corresponds to a 1-factor of the complete graph K_{16} that is disjoint from the 1-factor in columns 2 to 9 of Figure 1. The union of two such 1-factors is a 2-regular graph consisting of even-length cycles only; up to isomorphism these correspond to the partitions above.)

To complete the B part for each of the seven A matrices and to carry out isomorph rejection, the use of computer is inevitable. (Note that it not necessary to optimize these algorithms, since the amount of computer time consumed in this stage is only a fraction of the total time.) For each completion we perform isomorph rejection with *nauty* against a stored collection of orbit representatives. Each design is encoded as a vertex-colored bipartite graph in which vertices of one color correspond to the points, vertices of another color correspond to the blocks, and edges encode the incidence relation between points and blocks [13, p. 23]. In total, 14,648 pairwise nonisomorphic 25-block *seed subdesigns* are obtained in this way.

The problem of finding all extensions of a seed subdesign to an STS(19) is that of finding all solutions to an instance of *exact cover*. In the exact cover problem, we are given a set and a collection of its subsets; the task is to cover the set with given subsets so that each element of the set is covered exactly once. With our 25-block seed subdesigns, we want to cover the remaining uncovered pairs with 32 3-subsets so that each pair is covered exactly once. A fast, state-of-the-art exact cover algorithm can be found in [11], to which we refer the reader for details.

In fact, it turns out that data of a complete search described above can be combined with previous results to calculate the number of pairwise nonisomorphic STS(19) as follows. Let $\mathcal{B}_1, \ldots, \mathcal{B}_{N(19)}$ be representatives from the isomorphism classes of STS(19). Then the orbit-stabilizer theorem gives as the total number of STS(19)

$$\sum_{i=1}^{N(19)} \frac{19!}{|\mathrm{Aut}(\mathcal{B}_i)|}. \tag{6}$$

Let the seed subdesigns be $\mathcal{S}_i$, $1 \le i \le 14{,}648$, and let M_i denote the total number of completions of $\mathcal{S}_i$ to STS(19). Again, the orbit-stabilizer theorem can be used to get the total number of STS(19):

$$\frac{1}{57}\sum_{i=1}^{14{,}648} \frac{19!}{|\mathrm{Aut}(\mathcal{S}_i)|} \cdot M_i. \tag{7}$$

Since we know [3] the number of STS(19) for which $|\mathrm{Aut}(\mathcal{B}_i)| > 1$, and all values in (7) are known after the search, it is straightforward to determine from (6) the number of STS(19) with trivial automorphism group.

However, to be able to verify the final result in two different ways, and also to be able to check all designs for certain properties, we enhance the search to be able to identify all pairwise nonisomorphic STS(19). Actually, in this way we are able to discover fatal errors in [3].

3.2. **Isomorph Rejection.** The most involved part of the algorithm is the elimination of isomorphic STS(19) from consideration. There are three issues that need to be addressed.

First, the main search must be conducted in parallel because of the considerable resource requirements. This presents a difficulty since the parallel runs should preferably be independent of each other, whereby no comparisons between isomorphism class representatives encountered in distinct runs are allowed. Second, the search is to be conducted in part on computers that do not have enough main memory to store the millions of isomorphism class representatives potentially encountered as extensions of a single seed subdesign. Third, isomorphism testing must be fast since there are in the order of, as we now know, $7 \cdot 10^{11}$ STS(19) candidates that are to be tested for isomorphism. Luckily enough, all of the above difficulties are essentially solved by a recent algorithm framework for isomorph-free exhaustive generation [15].

Our basic isomorph rejection strategy is to use *nauty* to compute the canonical labeling and automorphism orbits of the block graph of a generated STS(19). To enable parallelization we must guarantee that algorithm runs performed on different seed subdesigns do not output isomorphic STS(19). This can be accomplished by using the output of *nauty* to test that a generated STS(19) originates from the correct *parent* seed subdesign. This test is motivated by the general theory in [15].

We fix the following labeling convention to connect the seed subdesign to a block graph. (Recall that the block $\{1, 2, 3\}$ occurs in every seed subdesign.)

Requirement 3.1. We require that the block $\{1, 2, 3\}$ corresponds to the vertex labeled 1 in a block graph whenever the test (8) below is performed.

For every STS(19) generated from a seed subdesign, we construct its block graph G and test with *nauty* that the following condition is satisfied. (Recall that σ_G denotes the canonical labeling computed by *nauty* for G.)

$$1 \in \{\alpha(\sigma_G^{-1}(1)) \;:\; \alpha \in \mathrm{Aut}(G)\}. \tag{8}$$

If the test fails, we reject the STS(19) from further consideration.

To establish correctness of the test (8) we must prove that the test is sound, that is, any two isomorphic STS(19) that pass the test are generated from the same seed subdesign. Moreover, we must prove that the test is complete in the sense that a generated STS(19) from every isomorphism class of STS(19) will pass the test.

Theorem 3.2. *Let $\mathcal{B}$ and $\mathcal{B}'$ be two generated* STS(19) *that pass the test* (8)*. If $\mathcal{B}$ and $\mathcal{B}'$ are isomorphic, then they have been constructed by extending the same seed subdesign $\mathcal{S}$. Moreover, there exists an automorphism of $\mathcal{S}$ that is an isomorphism of $\mathcal{B}$ onto $\mathcal{B}'$.*

Proof. Let $\mathcal{S}$ and $\mathcal{S}'$ be the seed subdesigns from which $\mathcal{B}$ and $\mathcal{B}'$ have been constructed. Suppose $\mathcal{B} = \{B_1, \ldots, B_b\}$ and $\mathcal{B}' = \{B'_1, \ldots, B'_b\}$ satisfy Requirement 3.1 and let G and G' be the associated block graphs. Since both graphs pass the test (8), there exist $\alpha \in \mathrm{Aut}(G)$ and $\beta \in \mathrm{Aut}(G')$ such that

$$\alpha(\sigma_G^{-1}(1)) = 1 = \beta(\sigma_{G'}^{-1}(1)).$$

By Corollary 2.3 and (4) we have

$$\sigma_G(\alpha^{-1}(G)) = \sigma_G(G) = \sigma_{G'}(G') = \sigma_{G'}(\beta^{-1}(G')).$$

So, $\tau := \beta\sigma_{G'}^{-1}\sigma_G\alpha^{-1}$ is an isomorphism of G onto G' that keeps the vertex 1 fixed. Since $B_1 = B'_1 = \{1,2,3\}$ by Requirement 3.1, Corollary 2.6 applied to τ gives an isomorphism $\nu \in S_v$ of $\mathcal{B}$ onto $\mathcal{B}'$ such that $\nu(\{1,2,3\}) = \{1,2,3\}$. Consequently, $\nu(\mathcal{S}) = \mathcal{S}'$, so $\mathcal{S}$ and $\mathcal{S}'$ are isomorphic and hence equal because exactly one representative from each isomorphism class of seed subdesigns is considered for extension in the main search. Thus, $\mathcal{B}$ and $\mathcal{B}'$ have been constructed from the same seed subdesign and ν is an automorphism of $\mathcal{S}$. □

Theorem 3.3. *For every isomorphism class of* STS(19)*, there exists an* STS(19) *that is an extension of a seed subdesign and that passes the test (8).*

Proof. Let $\mathcal{B} = \{B_1, \ldots, B_b\}$ be an arbitrary STS(19) and suppose G is the associated block graph. (Requirement 3.1 need not apply to G.) Put $p := \sigma_G^{-1}(1)$. By the structure of an STS(19), the set of blocks $\mathcal{S} := \{B_i \ : \ B_p \cap B_i \neq \emptyset\}$ is isomorphic to exactly one seed subdesign $\mathcal{S}'$ considered in the main search. Let $\mu \in S_v$ be an isomorphism of $\mathcal{S}$ onto $\mathcal{S}'$. Put $\mathcal{B}' := \mu(\mathcal{B})$. Since $\mathcal{S}' \subset \mathcal{B}'$, $\mathcal{B}'$ will eventually be generated as an extension of the seed subdesign $\mathcal{S}'$. Fix any labeling $\mathcal{B}' = \{B'_1, \ldots, B'_b\}$ that satisfies Requirement 3.1, and suppose G' is the associated block graph. We now show that G' satisfies (8). Define $\tau \in S_b$ by the rule $\mu(B_j) = B'_{\tau(j)}$ for all $j \in \{1, \ldots, b\}$. Clearly, $G' = \tau(G)$. Since B_p is the only block whose points occur with multiplicity r in $\mathcal{S}$, and the same holds for the block $\{1,2,3\}$ in $\mathcal{S}'$, we must have $\mu(B_p) = \{1,2,3\}$. Consequently, $\tau(p) = 1$ since Requirement 3.1 applies to G'. By (4) we have $\sigma_G^{-1}\sigma_{G'}\tau \in \mathrm{Aut}(G)$, or equivalently, $\alpha := \tau\sigma_G^{-1}\sigma_{G'} \in \mathrm{Aut}(G')$. But this implies that G' satisfies (8) since $\alpha\sigma_{G'}^{-1}(1) = \tau\sigma_G^{-1}(1) = \tau(p) = 1$. □

After the test (8) we must still perform isomorph rejection on those STS(19) that have been generated from the same seed subdesign.

If the automorphism group of the seed subdesign is large, then the further test we employ is simply a hash table query to see whether the canonically labeled block graph $\sigma_G(G)$ computed by *nauty* during the test (8) has been encountered earlier. If $\sigma_G(G)$ does not occur in the hash table, then we accept the STS(19) and insert $\sigma_G(G)$ into the hash table; otherwise we reject the STS(19). By Corollary 2.3 this suffices for isomorph rejection together with the test (8).

When the automorphism group of the seed subdesign is small, we employ an alternative test, also motivated by [15], based on automorphisms of the seed subdesign. With this test it is not necessary to store the isomorphism class representatives encountered. Let $\mathcal{S}$ be a seed subdesign and suppose $\mathrm{Aut}(\mathcal{S})$ is the corresponding automorphism group (acting on the points). Suppose $\mathcal{B} \supset \mathcal{S}$ is an extension of $\mathcal{S}$ to an STS(19). We test whether $\mathcal{B}$ satisfies

$$\text{for all } \mu \in \mathrm{Aut}(\mathcal{S}), \quad \mathcal{B} \leq \mu(\mathcal{B}), \tag{9}$$

where "$\leq$" is any (for example, lexicographic) total order on the set of STS(19). For performance reasons it is useful to perform the test (9) first, that is, if a generated STS(19) satisfies (9), then we proceed with the test (8); otherwise we immediately reject the STS(19).

We conclude this section by proving that the tests (8) and (9) work together as intended. First, we prove that the two tests are sound.

Theorem 3.4. *Let $\mathcal{B}$ and $\mathcal{B}'$ be distinct extensions of a seed subdesign $\mathcal{S}$ to an* STS(19)*. If both $\mathcal{B}$ and $\mathcal{B}'$ pass the tests* (8) *and* (9)*, then they are nonisomorphic.*

Proof. To reach a contradiction, suppose $\mathcal{B}$ and $\mathcal{B}'$ are distinct and isomorphic. By Theorem 3.2 there exists a $\mu \in \mathrm{Aut}(\mathcal{S})$ such that $\mu(\mathcal{B}) = \mathcal{B}'$. Hence, either $\mathcal{B}$ or $\mathcal{B}'$ is rejected in the test (9). □

It remains to show completeness, that is, at least one generated STS(19) from every isomorphism class of STS(19) will pass both tests. For an arbitrary isomorphism class of STS(19), let $\mathcal{B}$ be a generated STS(19) from the isomorphism class that passes the test (8); the existence of $\mathcal{B}$ is guaranteed by Theorem 3.3. Let $\mathcal{S}$ be the seed subdesign from which $\mathcal{B}$ is generated and let $\mathcal{B}_{\min}$ be the (lexicographic) minimum of the $\mathrm{Aut}(\mathcal{S})$-orbit of $\mathcal{B}$. The following theorem shows that also $\mathcal{B}_{\min}$ passes the test (8). Since $\mathcal{B}_{\min}$ by definition passes the test (9), we conclude that at least one generated STS(19) from every isomorphism class will pass both tests.

Theorem 3.5. *Let $\mathcal{B}$ be an extension of a seed subdesign $\mathcal{S}$ to an* STS(19)*. Then, $\mathcal{B}$ passes the test* (8) *if and only if $\mu(\mathcal{B})$ passes the test* (8) *for all $\mu \in \mathrm{Aut}(\mathcal{S})$.*

Proof. It suffices to prove the "only if" direction. Let $\mu \in \mathrm{Aut}(\mathcal{S})$. Suppose that $\mathcal{B} = \{B_1, \ldots, B_b\}$ and $\mathcal{B}' := \mu(\mathcal{B}) = \{B'_1, \ldots, B'_b\}$ satisfy Requirement 3.1, and that G and G' are the associated block graphs. Define $\tau \in S_b$ from the rule $\mu(B_j) = B'_{\tau(j)}$ for all $j \in \{1, \ldots, b\}$. Clearly, $G' = \tau(G)$. Since G passes the test (8), there exists an $\alpha \in \mathrm{Aut}(G)$ for which $\alpha\sigma_G^{-1}(1) = 1$. From (4) we obtain that $\beta := \tau\alpha\sigma_G^{-1}\sigma_{G'} \in \mathrm{Aut}(G')$. Thus, $\beta\sigma_{G'}^{-1}(1) = \tau\alpha\sigma_G^{-1}(1) = \tau(1) = 1$, where the last equality follows from $\mu(\mathcal{S}) = \mathcal{S}$ and Requirement 3.1. This shows that G' passes the test (8). □

3.3. Implementation Details.

The use of a vertex invariant is required to guarantee good performance from *nauty* on strongly regular graphs. The Pasch configuration invariant Λ_P derived in Section 2.3 succeeds most of the time in partitioning the vertex set of a block graph so that a single application of the partition refinement procedure of *nauty* produces a discrete partition.

Another significant performance gain is obtained from the following observation.

Theorem 3.6. *Let G be a block graph of an* STS(19)*, and suppose that $\sigma_G = \sigma_{G,\Lambda_P(G)}$. Then, G passes the test* (8) *only if* 1 *occurs in the first cell of $\Lambda_P(G)$.*

Proof. Let $\Lambda_P(G) = (V_1, \ldots, V_m)$ and $c(\Lambda_P(G)) = (W_1, \ldots, W_m)$. Since $\Lambda_P(G)$ is a vertex invariant, any automorphism $\alpha \in \mathrm{Aut}(G)$ satisfies $\alpha(\Lambda_P(G)) = \Lambda_P(G)$. So, since $\sigma_{G,\Lambda_P(G)}^{-1}(W_1) = V_1$ by (2), we have $\alpha\sigma_{G,\Lambda_P(G)}^{-1}(W_1) = V_1$ for all $\alpha \in \mathrm{Aut}(G)$. Thus, because $1 \in W_1$, G will not pass the test (8) unless $1 \in V_1$. □

In other words, if the vertex invariant Λ_P is used, a block graph will not pass the test (8) unless the number of Pasch configurations in which the block B_1 occurs is the maximum taken over all blocks of the STS(19).

This observation translates into the following algorithm for performing the test (8). Suppose a block labeling $\mathcal{B} = \{B_1, \ldots, B_b\}$ that satisfies Requirement 3.1 has been fixed.

(1) Starting from $i = 1$, compute for every block B_i the number of Pasch configurations $P(B_i)$ in which the block occurs.
(2) If $P(B_i) > P(B_1)$ for some i, then reject the STS(19).
(3) Construct the block graph, and partition its vertices into maximal cells of constant Pasch value. Sort the cells into order of decreasing Pasch value so that the cell with the largest Pasch value becomes the first one.
(4) Input the ordered partition to *nauty* together with the block graph.
(5) Perform the test (8) based on the canonical labeling and the automorphism orbits output by *nauty*.

Note that in steps 1 and 2 of the algorithm above we can compute $P(B_i)$ directly from the STS(19) without constructing the block graph. In this way we avoid the overhead of constructing the block graph if the STS(19) is rejected anyway based on Theorem 3.6.

Our implementation of the test (9) precomputes the elements of $\mathrm{Aut}(\mathcal{S})$ for each seed subdesign $\mathcal{S}$, and stores these in an array [2]. The test (9) in the main search is implemented as an exhaustive search that considers each permutation in the array and tests whether the permuted STS(19) is lexicographically smaller than the input STS(19). Since most of the seed subdesigns either have a trivial automorphism group—in which case the test (9) is trivial—or have small automorphism group order, this naïve test suffices for our purposes.

In the search, the threshold automorphism group order was set to 200 elements; the seed subdesigns with automorphism groups larger than this were processed using the isomorph rejection strategy based on storing the canonically labeled block graphs in a hash table. The maximum number of block graphs that had to be stored in the hash table was 100,813 for the 11 seed subdesigns processed in this way.

The main search took approximately 2 years of CPU time. The search was distributed using the batch system `autoson` [14] to a network of 65 IBM Intellistation Pro workstations with 450-MHz Pentium II CPUs and 15 other workstations with CPUs ranging from 1-GHz Athlon Thunderbird to 200-MHz Pentium.

4. The STS(19)

The results of the computer search are summarized in Tables 1 to 6 and in the following theorems. In particular, Theorem 4.3 can be obtained in two ways, using (6) and (7), thereby indicating correctness of the results. If (6) is applied, Theorem 4.3 can be calculated from the entries of Table 1. Table 1 shows, for each possible automorphism group order, the number of pairwise nonisomorphic STS(19) and how many of these are anti-Pasch.

Theorem 4.1. *The number of pairwise nonisomorphic* STS(19) *is* 11,084,874,829.

Corollary 4.2. *There are at least* 11,084,874,829 *pairwise nonisomorphic strongly regular graphs* $\mathrm{srg}(57, 24, 11, 9)$.

Theorem 4.3. *The total number of* STS(19) *is*

$$1{,}348{,}410{,}350{,}618{,}155{,}344{,}199{,}680{,}000.$$

The value of Theorem 4.1 can be compared with the estimates that have been published in [19] and [15]; the correct value actually falls within the interval (between $1.1 \cdot 10^{10}$ and $1.2 \cdot 10^{10}$) estimated in [15]; the older, and not as sophisticated, estimate in [19] (approximately $1.3 \cdot 10^{9}$) is too small.

When the results of Table 1 are compared with those of [3], one observes discrepancies for the group orders 2 and 3. To obtain evidence that the results of this paper are the correct ones, a separate calculation of the number of nonisomorphic STS(19) with a nontrivial automorphism group was carried out, incorporating ideas from [3]. An STS(19) with a nontrivial automorphism group must admit an automorphism that has one of the following *basic* cycle types:

$$19^1, \quad 1^1 2^9, \quad 1^1 3^6, \quad 1^3 2^8, \quad 1^7 2^6, \quad 1^7 3^4.$$

For each basic automorphism type, the number of nonisomorphic STS(19) admitting such an automorphism is:

19^1	4
$1^1 2^9$	184
$1^1 3^6$	12,885
$1^3 2^8$	80,645
$1^7 2^6$	72,150
$1^7 3^4$	124

These numbers are inconsistent with those in [3] on automorphism types $1^1 3^6$, $1^3 2^8$, and $1^7 2^6$, where it is erroneously claimed that the respective numbers are 12,021, 80,591, and 80,558. (To prove that the results in [3] cannot be correct, it suffices to check, for example, that the 12,885 STS(19) admitting an automorphism of type $1^1 3^6$ are indeed nonisomorphic and admit an automorphism of the required type.) An electronic listing of all the 164,758 nonisomorphic STS(19) with a nontrivial automorphism group is available from the authors upon request.

We now correct the tables in [3]. The basic automorphism structure of STS(19) with a nontrivial automorphism group is summarized in Table 2. The format of Table 2 is identical to [3, Table 1] for ease of reference. The STS(19) are partitioned into classes according to the order of the full automorphism group. Each such class is partitioned further into subclasses according to the types of basic automorphisms that the STS(19) admit. Whenever more than one subclass exists, these are denoted by the letters a, b, c, d, e. For example, the class 4b contains the 662 STS(19) that have full automorphism group order 4 and admit only automorphisms of type $1^3 2^8$ among the basic automorphism types.

In [3] it is also obtained that, in addition to the basic automorphism types, an STS(19) can admit a *nonbasic* automorphism whose type belongs to the list

$$1^1 9^2, \quad 1^1 6^3, \quad 1^1 3^2 6^2, \quad 1^1 2^1 4^4, \quad 1^1 2^1 8^2, \quad 1^3 8^2, \quad 1^3 4^4, \quad 1^3 2^2 6^2, \quad 1^3 2^2 4^3.$$

Together with the basic automorphism types, these are the only nontrivial types of automorphism an STS(19) can admit.

The classes in Table 2 partition further into subclasses according to the nonbasic automorphisms admitted by an STS(19). This subdivision is given in Table 3. Table 3 is identical to [3, Table 2] with the exception of the class 8a, which partitions into two subclasses instead of the one given in [3]. Namely, among the 84 STS(19) in class 8a there exist two STS(19) whose full automorphism group has order 8 and that admit automorphisms of type $1^3 2^8$, $1^7 2^6$, and $1^3 4^4$. Accordingly, the number of STS(19) that admit an automorphism of type $1^3 4^4$ is 185 instead of the 183 reported in [3].

The maximum number of Pasch configurations in a Steiner triple system of order v is denoted by $P(v)$. See [21] for a discussion of $P(v)$ and [9] for some recent results on this function. Obviously, $P(v)$ is known for $v \leq 15$; for $v = 19$ it has been known that $P(19) \geq 84$ with three known designs attaining this value. In this work, the exact value of $P(19)$ is obtained. We also find the number of nonisomorphic anti-Pasch STS(19); previously it has been known that there are more than one thousand such designs [16]. The large number of nonisomorphic anti-Pasch STS(19) prohibits listing them here, however, the authors are happy to provide them electronically to anyone interested.

Theorem 4.4. *The maximum number of Pasch configurations in an* STS(19) *is* $P(19) = 84$*; there are three such designs (with automorphism groups of order* 108, 144*, and* 432*). The number of nonisomorphic anti-Pasch* STS(19) *is* 2,591.

It appears that the maximum number of Pasch configurations are to be found among the STS with large automorphism group. In Tables 4 to 6, the frequency of STS(19) with a given number of Pasch configurations, indicated by P, are shown for the automorphism groups of order at most 3.

Because not too many CPU years were consumed to obtain a classification of STS(19)—and an enumeration can be completed in much shorter time—one may ask whether an enumeration or a classification of STS(21) is within reach. To answer this question, the seed subdesigns for STS(21) were classified—there are 219,104 seeds—and a few of these were fed to the completion algorithm. It turned out that we were not able to process these seeds within reasonable time. The amount of CPU time needed to process one such seed was estimated to more than one year on a 500-MHz personal computer. A total amount of hundreds of thousands of CPU years needed for a classification of STS(21) does not look promising.

Acknowledgments

The authors would like to thank Peter Cameron for rewarding discussions. Juhani Markula and Jukka Seppänen at the Computing Center of Helsinki University of Technology are gratefully acknowledged for providing computing resources.

References

[1] R. C. Bose, *Strongly regular graphs, partial geometries and partially balanced designs*, Pacific J. Math. **13** (1963), 389–419. MR **28:**1137

[2] G. Butler, *Fundamental Algorithms for Permutation Groups*, Lecture Notes in Computer Science, Vol. 559, Springer-Verlag, Berlin, 1991. MR **94d:**68049

[3] C. J. Colbourn, S. S. Magliveras, and D. R. Stinson, *Steiner triple systems of order* 19 *with nontrivial automorphism group*, Math. Comp. **59** (1992), 283–295. MR **92k:**05022

[4] C. J. Colbourn and A. Rosa, *Triple Systems*, Clarendon Press, Oxford, 1999. MR **2002h:**05024

[5] F. N. Cole, L. D. Cummings, and H. S. White, *The complete enumeration of triad systems in* 15 *elements*, Proc. Nat. Acad. Sci. U.S.A. **3** (1917), 197–199.

[6] G. P. Egorychev, *The solution of van der Waerden's problem for permanents*, Adv. in Math. **42** (1981), 299–305. MR **83b:**15002b

[7] D. I. Falikman, *Proof of the van der Waerden conjecture on the permanent of a doubly stochastic matrix (in Russian)*, Mat. Zametki **29** (1981), 931–938, 957. MR **82k:**15007
[8] M. J. Grannell, T. S. Griggs, and E. Mendelsohn, *A small basis for four-line configurations in Steiner triple systems*, J. Combin. Des. **3** (1995), 51–59. MR **95j:**05041
[9] B. D. Gray and C. Ramsay, *On the number of Pasch configurations in a Steiner triple system*, Bull. Inst. Combin. Appl. **24** (1998), 105–112. MR **99c:**05030
[10] M. Hall, Jr. and J. D. Swift, *Determination of Steiner triple systems of order* 15, Math. Tables Aids Comput. **9** (1955), 146–152. MR **18,192d**
[11] D. E. Knuth, *Dancing links*, Millennial Perspectives in Computer Science, J. Davies, B. Roscoe, and J. Woodcock, editors, Palgrave, Houndmills, 2000, pp. 187–214.
[12] B. D. McKay, *Practical graph isomorphism*, Congr. Numer. **30** (1981), 45–87. MR **83e:**05061
[13] B. D. McKay, *nauty user's guide (version* 1.5*)*, Technical Report TR-CS-90-02, Computer Science Department, Australian National University, 1990.
[14] B. D. McKay, *autoson – a distributed batch system for UNIX workstation networks (version* 1.3*)*, Technical Report TR-CS-96-03, Computer Science Department, Australian National University, 1996.
[15] B. D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms **26** (1998), 306–324. MR **98k:**68132
[16] J. P. Murphy, *Steiner Triple Systems and Cycle Structure*, Ph.D. thesis, University of Central Lancashire, 1999.
[17] D. A. Spielman, *Faster isomorphism testing of strongly regular graphs*, Proc. 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, 22–24 May 1996, pp. 576–584. MR **97g:**68005
[18] D. R. Stinson, *A comparison of two invariants for Steiner triple systems: Fragments and trains*, Ars. Combin. **16** (1983), 69–76. MR **85f:**05023
[19] D. R. Stinson and H. Ferch, 2000000 *Steiner triple systems of order* 19, Math. Comp. **44** (1985), 533–535. MR **86e:**05014
[20] D. R. Stinson and E. Seah, 284 457 *Steiner triple systems of order* 19 *contain a subsystem of order* 9, Math. Comp. **46** (1986), 717–729. MR **87c:**05027
[21] D. R. Stinson and Y. J. Wei, *Some results on quadrilaterals in Steiner triple systems*, Discrete Math. **105** (1992), 207–219. MR **93h:**05021
[22] H. S. White, F. N. Cole, and L. D. Cummings, *Complete classification of triad systems on fifteen elements*, Memoirs Nat. Acad. Sci. U.S.A. **14** (1919), 1–89.
[23] R. M. Wilson, *Nonisomorphic Steiner triple systems*, Math. Z. **135** (1974), 303–313. MR **49:**4803

Department of Computer Science and Engineering, Helsinki University of Technology, P.O. Box 5400, 02015 HUT, Finland
E-mail address: petteri.kaski@hut.fi

Department of Electrical and Communications Engineering, Helsinki University of Technology, P.O. Box 3000, 02015 HUT, Finland
E-mail address: patric.ostergard@hut.fi

TABLE 1. The STS(19).

$\lvert\mathrm{Aut}(\mathcal{B})\rvert$	STS(19)	Anti-Pasch
1	11,084,710,071	2,538
2	149,522	1
3	12,728	41
4	2,121	0
6	182	5
8	101	0
9	19	4
12	37	0
16	13	0
18	11	0
19	1	0
24	11	0
32	3	0
54	2	0
57	2	1
96	1	0
108	1	0
144	1	0
171	1	1
432	1	0
Total	11,084,874,829	2,591

Table 2. Basic automorphisms.

Order	Class	19^1	$1^1 2^9$	$1^1 3^6$	$1^3 2^8$	$1^7 2^6$	$1^7 3^4$	STS(19)
432				$*$	$*$	$*$	$*$	1
171		$*$		$*$				1
144				$*$	$*$	$*$		1
108				$*$	$*$	$*$	$*$	1
96				$*$	$*$	$*$		1
57		$*$		$*$				2
54				$*$		$*$	$*$	2
32					$*$	$*$		3
24				$*$	$*$	$*$		11
19		$*$						1
18	a		$*$	$*$				1
	b			$*$	$*$		$*$	2
	c			$*$		$*$		2
	d			$*$		$*$	$*$	6
16					$*$	$*$		13
12	a			$*$	$*$			7
	b			$*$	$*$	$*$		8
	c			$*$		$*$		12
	d				$*$	$*$	$*$	10
9				$*$				19
8	a				$*$	$*$		84
	b				$*$			17
6	a		$*$	$*$				14
	b			$*$	$*$			14
	c			$*$		$*$		116
	d				$*$		$*$	10
	e					$*$	$*$	28
4	a				$*$	$*$		839
	b				$*$			662
	c					$*$		620
3	a			$*$				12,664
	b						$*$	64
2	a		$*$					169
	b				$*$			78,961
	c					$*$		70,392
Total		4	184	12,885	80,645	72,150	124	164,758

TABLE 3. Nonbasic automorphisms.

Class	$1^1 9^2$	$1^1 6^3$	$1^1 3^2 6^2$	$1^1 2^1 4^4$	$1^1 2^1 8^2$	$1^3 8^2$	$1^3 4^4$	$1^3 2^2 6^2$	$1^3 2^2 4^3$	STS(19)
432			*			*	*	*		1
171	*									1
144			*	*	*		*			1
108			*					*		1
96			*				*			1
57										2
54			*							2
32				*			*			3
24			*							11
19										1
18a		*								1
18b								*		2
18c			*							6
18d			*							2
16				*	*		*			5
16				*						6
16						*	*			1
16							*			1
12a			*							8
12b										7
12c										12
12d								*		10
9	*									9
9										10
8a							*			2
										82
8b				*			*			5
					*		*			10
						*	*			2
6a		*								14
6b										14
6c			*							104
										12
6d								*		10
6e										28
4a										839
4b				*						498
							*			153
										11
4c									*	48
										572
Total	10	15	137	518	16	4	185	24	48	

TABLE 4. The STS(19) with $|\mathrm{Aut}(\mathcal{B})| = 1$.

P	STS(19)	P	STS(19)	P	STS(19)
0	2,538	22	347,316,148	44	10,567
1	35,742	23	255,585,528	45	5,943
2	263,580	24	182,930,596	46	3,864
3	1,314,921	25	127,610,069	47	2,125
4	4,958,394	26	86,994,788	48	1,558
5	15,095,241	27	58,048,786	49	715
6	38,479,651	28	38,001,524	50	664
7	84,328,790	29	24,453,668	51	350
8	162,042,722	30	15,483,681	52	316
9	276,885,482	31	9,660,784	53	78
10	426,046,203	32	5,948,963	54	126
11	596,271,490	33	3,621,508	55	68
12	765,950,843	34	2,183,650	56	93
13	910,509,472	35	1,300,661	57	19
14	1,008,606,577	36	770,041	58	56
15	1,047,848,142	37	451,540	59	5
16	1,027,119,044	38	263,545	60	11
17	954,708,823	39	151,688	62	17
18	845,586,319	40	89,084	64	1
19	716,600,889	41	50,804	66	2
20	583,312,837	42	29,632	70	2
21	457,752,251	43	16,852		

TABLE 5. The STS(19) with $|\mathrm{Aut}(\mathcal{B})| = 2$.

P	STS(19)	P	STS(19)	P	STS(19)
0	1	23	3,602	43	120
2	35	24	7,756	44	1,184
4	216	25	3,943	45	50
6	794	26	8,078	46	687
7	3	27	3,892	47	23
8	2,024	28	8,432	48	436
9	18	29	3,261	49	7
10	4,119	30	8,132	50	261
11	63	31	2,657	51	8
12	6,506	32	7,481	52	135
13	242	33	1,751	54	121
14	8,538	34	6,277	55	1
15	571	35	1,247	56	38
16	9,748	36	4,994	58	28
17	1,247	37	742	60	7
18	9,354	38	3,915	62	23
19	2,049	39	386	64	1
20	8,604	40	2,848	66	6
21	2,920	41	203	70	3
22	7,920	42	1,814		

TABLE 6. The STS(19) with $|\mathrm{Aut}(\mathcal{B})| = 3$.

P	STS(19)	P	STS(19)	P	STS(19)
0	41	20	421	40	6
1	16	21	694	41	11
2	31	22	224	42	14
3	240	23	296	43	2
4	70	24	412	44	6
5	131	25	156	45	4
6	602	26	200	46	4
7	190	27	251	47	3
8	266	28	86	48	16
9	1,016	29	135	49	2
10	350	30	148	50	1
11	441	31	54	51	1
12	1,298	32	67	52	5
13	404	33	76	54	2
14	556	34	36	57	1
15	1,306	35	43	58	1
16	416	36	40	59	1
17	528	37	18	60	5
18	987	38	18		
19	352	39	27		