One-Factorizations of Regular Graphs of Order 12

Petteri Kaski[*]

Department of Computer Science and Engineering,
Helsinki University of Technology,
P.O. Box 5400, FI-02015 TKK, Finland
petteri.kaski@hut.fi

Patric R. J. Östergård[†]

Department of Electrical and Communications Engineering,
Helsinki University of Technology,
P.O. Box 3000, FI-02015 TKK, Finland
patric.ostergard@hut.fi

Submitted: Oct 5, 2004; Accepted: Dec 6, 2004; Published: Jan 7, 2005
Mathematics Subject Classifications: 05C70, 05-04

Abstract

Algorithms for classifying one-factorizations of regular graphs are studied. The smallest open case is currently graphs of order 12; one-factorizations of r-regular graphs of order 12 are here classified for $r \leq 6$ and $r = 10, 11$. Two different approaches are used for regular graphs of small degree; these proceed one-factor by one-factor and vertex by vertex, respectively. For degree $r = 11$, we have one-factorizations of K_{12}. These have earlier been classified, but a new approach is presented which views these as certain triple systems on $4n - 1$ points and utilizes an approach developed for classifying Steiner triple systems. Some properties of the classified one-factorizations are also tabulated.

1 Introduction

An *r-factor* of a graph G is an r-regular spanning subgraph of G. An *r-factorization* of G is a partition of the edges of G into r-factors. We consider here one-factorizations (alternatively, 1-factorizations) of small regular graphs of even order $2n$ and degree $1 \leq k \leq 2n - 1$. The complete graph K_{2n} is the unique regular graph of order $2n$ and degree $2n - 1$.

[*] Research supported by the Helsinki Graduate School in Computer Science and Engineering (HeCSE) and a grant from the Foundation of Technology, Helsinki, Finland (Tekniikan Edistämissäätiö).

[†] Research supported by the Academy of Finland under Grants No. 100500 and 202315.

Two one-factorizations are *isomorphic* if there exists a bijection between the vertices of the graphs that maps one-factors onto one-factors; such a bijection is an *isomorphism.*

The problem of classifying one-factorizations of regular graphs up to isomorphism was solved for $2n \leq 10$ in the mid-1980s [12, 28, 29]. With hundreds of objects for order 10, we have millions of objects for order 12; still Dinitz, Garnick, and McKay managed to classify the one-factorizations of K_{12}—there are 526,915,620 such objects—in the early 1990s in less than eight months by distributing the problem to a network of workstations [10]. The order 12 case has remained open for other degrees (except for the smallest, trivial ones), and in fact does so for some parameters even after this study.

In this paper several algorithms for classification of one-factorizations of regular graphs are considered. In Section 2, we discuss algorithms for classifying one-factorizations that are based on a coding-theoretic viewpoint. Two algorithms are utilized, one that proceeds a one-factor at a time, and one that proceeds a vertex at a time. In Section 3, we present an algorithm for classifying one-factorizations that is based on viewing one-factorizations as certain triple systems. We also show how a classification of one-factorizations of K_{2n} can be used to deduce the one-factorizations of graphs of degree $2n-2$; up to isomorphism there is exactly one such graph, the graph obtained by deleting a one-factor from K_{2n}. In this manner, classification results for regular graphs of order 12 are obtained for degrees $k \leq 6$ and $k = 10, 11$. Hence the cases $k = 7, 8, 9$ remain open. In none of the algorithms presented is a classification of regular graphs utilized. The classification results are summarized in Section 4.

2 One-Factorizations of Graphs with Small Degree

The algorithms for constructing one-factorizations of regular graphs of small degree k can be divided roughly into two types.

Algorithms of the first type utilize a classification of the underlying regular graphs (see [22] for an efficient classification algorithm for regular graphs) and classify the nonisomorphic one-factorizations one graph G at a time. This approach is employed in [28]. Also the approach to be presented in Section 3 admits generalization from complete graphs K_{2n} to arbitrary regular graphs, but such a generalization is not considered here.

Algorithms of the second type construct the one-factorizations directly without relying on a classification of regular graphs. Possibilities for such an algorithm include constructing the one-factorizations either vertex by vertex or factor by factor. The latter of these strategies is employed in [10]. (Strictly speaking, the algorithm in [10] is optimized for the case $k = 2n-1$; if the algorithm is used for $k < 2n-1$, it must be slightly relaxed.)

In this section we describe two algorithms that are based on viewing one-factorizations as certain error-correcting codes.

2.1 One-Factorizations and Codes

We recall some standard coding-theoretic terminology. Let $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ and write $\mathbb{Z}_q^\ell$ for the set of all ordered ℓ-tuples (words) $x = x(1)x(2)\cdots x(\ell)$ over $\mathbb{Z}_q$. For a word x we say that $x(i)$ is the *symbol* at *coordinate* $i \in \{1, 2, \ldots, \ell\}$. The (*Hamming*) *distance* between two words $x, y \in \mathbb{Z}_q^\ell$ is

$$d(x, y) = |\{i \in \{1, 2, \ldots, \ell\} \ : \ x(i) \neq y(i)\}|.$$

A q-ary *code* of length ℓ is a nonempty set $C \subseteq \mathbb{Z}_q^\ell$. The *minimum distance* of a code is $d(C) = \min_{x,y \in C : x \neq y} d(x, y)$. A code is *equidistant* if $d(x, y) = d(C)$ for all distinct $x, y \in C$. An $(\ell, M, d)_q$ code is a q-ary code of length ℓ, cardinality M, and minimum distance d. A code is *equireplicate* if q divides $|C|$ and every symbol occurs $|C|/q$ times in every coordinate of the code.

Two codes are *equivalent* if the words in one code can be mapped onto the words of the other code by permuting the coordinates and the symbols separately in each coordinate of the code. In other words, denoting by S_d the symmetric group of degree d, two codes are equivalent if and only if they are in the same orbit under the coordinate- and symbol-permuting action of the wreath product $S_q \wr S_\ell$ on subsets of $\mathbb{Z}_q^\ell$. The automorphism group $\mathrm{Aut}(C)$ of a code C is the subgroup of $S_q \wr S_\ell$ that consists of all group elements that map C onto itself.

By a result of Semakov and Zinov'ev [31], the one-factorizations of K_{2n}—which in the context of [31] should be interpreted as resolutions of a 2-$(2n, 2, 1)$ design—correspond to $(2n{-}1, 2n, 2n{-}2)_n$ codes. For convenience we here give a description of the correspondence in graph-theoretic terminology.

A one-factorization of K_{2n} gives rise to a $(2n-1, 2n, 2n-2)_n$ code as follows. Let $\mathcal{F} = \{F(1), F(2), \ldots, F(2n-1)\}$ be a one-factorization of K_{2n} where $F(1), F(2), \ldots, F(2n-1)$ are the one-factors. For each one-factor $F(i)$, label the edges in $F(i)$ with numbers $0, 1, \ldots, n-1$ so that no two edges in $F(i)$ are labeled with the same number. Now associate with each vertex v in K_{2n} a word x_v such that the symbol $x_v(i)$ is the label of the edge incident with v in $F(i)$. It is straightforward to check that the resulting code $\{x_v : v \in V(K_{2n})\}$ has the desired parameters.

The one-factorization of K_6 and the code in (1) illustrate the correspondence (with the edges in each one-factor labeled $0, 1, 2$ from left to right).

$$\begin{array}{ll} F(1) = \{pq, rs, tu\} & x_p = 00000 \\ F(2) = \{pr, qt, su\} & x_q = 01111 \\ F(3) = \{pu, qr, st\} & x_r = 10122 \\ F(4) = \{ps, qu, rt\} & x_s = 12201 \\ F(5) = \{pt, qs, ru\} & x_t = 21220 \\ & x_u = 22012 \end{array} \qquad (1)$$

By the generalized q-ary Plotkin bound [2, Theorem 3], a $(2n-1, 2n, 2n-2)_n$ code is equidistant and equireplicate. Thus, conversely, a $(2n-1, 2n, 2n-2)_n$ code always defines a one-factorization of K_{2n}. It is straightforward to check that this correspondence is one-to-one between equivalence classes of codes and isomorphism classes of one-factorizations. More generally, an equireplicate $(k, 2n, k-1)_n$ code corresponds to a one-factorization of a regular graph of order $2n$ and degree k.

2.2 Two Classification Methods

Constructing one-factorizations of regular graphs of order $2n$ and degree k one vertex at a time is equivalent to constructing the corresponding equireplicate $(k, 2n, k-1)_n$ codes one word at a time. For this purpose we may employ the algorithm described in [14, 16]; we refer the interested reader to these papers for details. Note that we do not here require that the codes be equidistant, and the algorithm should be modified accordingly.

In what follows we describe an alternative algorithm that constructs the equireplicate $(k, 2n, k-1)_n$ codes one coordinate at a time using the canonical construction path method [20]. In [25] this general approach is applied to classify covering codes; the novelty in the present work is that there is no requirement to store any code equivalence class representatives due to the careful design of the step that extends a code by a new coordinate.

The coordinate-by-coordinate code classification algorithm has the top-level structure of a backtrack search. A partial solution in the search is an equireplicate $(j, 2n, j-1)_n$ code C_j, $1 \leq j \leq k$. For $j = k$, the algorithm outputs C_k as the representative of its equivalence class. For $j < k$, the algorithm extends C_j by adding coordinate $j+1$ so that the result C_{j+1} is an equireplicate $(j+1, 2n, j)_n$ code. After C_{j+1} has been constructed, it is subjected to an isomorph rejection test. If the test accepts C_{j+1}, then the search is invoked recursively with C_{j+1} as input; otherwise C_{j+1} is rejected and the next extension of C_j is considered.

The isomorph rejection test is based on a function m that associates to every code $C \subseteq \mathbb{Z}_q^\ell$ an $\mathrm{Aut}(C)$-orbit $m(C) \subseteq \{1, 2, \ldots, \ell\}$ of coordinates such that, for any two codes C, C', any isomorphism $C \to C'$ maps $m(C)$ onto $m(C')$. The test accepts C_{j+1} if and only if $j+1 \in m(C_{j+1})$. We compute $m(C)$ by encoding C as a vertex-colored graph (see [24]) and executing *nauty* [18] on the graph. As a side effect, we obtain generators for $\mathrm{Aut}(C)$.

We proceed to describe the extension step from C_j to C_{j+1}. Label the codewords in C_j as $x_1, x_2, \ldots, x_{2n}$. The automorphism group $\mathrm{Aut}(C_j)$ acts on C_j by permuting the words among themselves. Let H be the corresponding permutation group that acts on the indices $\{1, 2, \ldots, 2n\}$ instead of the words $\{x_1, x_2, \ldots, x_{2n}\}$. We view each extension of C_j into C_{j+1} as an ordered $2n$-tuple $Y = [y_1, y_2, \ldots, y_{2n}]$ of symbols such that $y_i \in \mathbb{Z}_q$ extends the word x_i for all $1 \leq i \leq 2n$. The direct product group $S_q \times H$ acts on the set of ordered $2n$-tuples of symbols by permuting the symbols and the positions. More precisely, for $\alpha \in S_q$ and $\beta \in H$,

$$\alpha\beta Y = [\alpha(y_{\beta^{-1}(1)}), \alpha(y_{\beta^{-1}(2)}), \ldots, \alpha(y_{\beta^{-1}(2n)})].$$

We assume that the tuples are ordered lexicographically so that $Y < Y'$ if and only if there exists an i such that $y_i < y'_i$ and $y_h = y'_h$ for all $1 \leq h < i$.

The extension step constructs exactly one $2n$-tuple Y from each orbit of $S_q \times H$ such that C_j extended with Y is an equireplicate $(j+1, 2n, j)_n$ code. We use the following orderly backtrack algorithm (see [11, 27]) for this task. For $1 \leq m \leq 2n$, a partial solution in the search is an m-tuple $Y_m = [y_1, y_2, \ldots, y_m]$ that is the lexicographic minimum of its orbit under the action of $S_q \times H_m$, where H_m is the subgroup of H that stabilizes $m+1, m+2, \ldots, 2n$ pointwise. A partial solution is discarded if it violates the minimum distance condition or if it is not the minimum of its $S_q \times H_m$-orbit.

To test minimality of Y_m, we determine for every $\beta \in H_m$ whether there exists an $\alpha \in S_q$ such that $\alpha\beta Y_m < Y_m$. Note that $\min_{\alpha \in S_q} \alpha\beta Y_m$ can be obtained from βY_m by permuting the symbols so that, in order of the positions, every occurrence of every symbol $a > 0$ is preceded by an occurrence of $a-1$.

Permutation group algorithms for manipulating automorphism groups can be found in [4, 32].

3 One-Factorizations of Complete Graphs

The most efficient known algorithm for classifying one-factorizations of complete graphs can be found in [10]; this algorithm constructs one-factors one by one and uses the method of canonical representatives [11, 27] for isomorph rejection. We present here an alternative approach that views one-factorizations as certain triple systems and classifies these using a modification of the algorithm in [15].

In this way we are able to redo the classification of one-factorizations of K_{12} in approximately 50 MIPS years, whereas 160 MIPS years was used for the classification in [10]. The next open instance is still out of reach, since there are apparently about 10^{18} nonisomorphic one-factorizations of K_{14} [10].

3.1 One-Factorizations as Triple Systems

One-factorizations of K_{2n} may be viewed as certain triple systems. For such a one-factorization, we define a set $U = \{u_1, u_2, \ldots, u_{2n-1}\}$ with one element for each one-factor, a set $V = \{v_1, v_2, \ldots, v_{2n}\}$ with one element for each vertex of the complete graph, and a set system containing a set $\{u_a, v_b, v_c\}$ exactly when the edge $\{v_b, v_c\}$ occurs in the one-factor u_a. The elements of U and V are called *points*. For example, the following set system describes a one-factorization of K_4:

$$\begin{aligned}\{&\{u_1, v_1, v_2\}, \{u_1, v_3, v_4\}, \{u_2, v_1, v_3\},\\ &\{u_2, v_2, v_4\}, \{u_3, v_1, v_4\}, \{u_3, v_2, v_3\}\}.\end{aligned}$$

In other words, a one-factorization of K_{2n} is a triple system on $4n-1$ points with $|U| = 2n-1$ and $|V| = 2n$, such that each triple, or *block*, contains one point from U and two points from V. Moreover, each pair of points in V as well as each pair of one point in U and one point in V must occur in exactly one block. Thus, such a triple system is a group divisible design (GDD) of constant block size 3, index 1, and group type $(2n-1)^1 1^{2n}$ (see [23]).

Two triple systems of one-factorizations are *isomorphic* if there exists a permutation of points (an *isomorphism*) that fixes U and V setwise and maps the blocks of one system onto the blocks of the other system.

3.2 Generating Triple Systems

The triple system representation links one-factorizations closely to *Steiner triple systems* (STSs), which consist of 3-element blocks from a given set of points, such that every pair of points occurs in exactly one block. An efficient algorithm for classifying Steiner triple systems is presented in [15]. With small modifications that we present here, this algorithm can be adapted to classify triple systems of one-factorizations.

The main observation behind the algorithm in [15] and the present algorithm is that the construction of triple systems can be seen as an instance of the well known exact cover problem. In the present context, the task is to cover all pairs of points of the form $\{u_a, v_b\}$ and $\{v_b, v_c\}$ with triples of the form $\{u_a, v_b, v_c\}$, where $u_a \in U$ and $v_b, v_c \in V$. Each triple covers the pairs of points that occur in it, and each pair is to be covered exactly once.

The classification algorithm has two stages. The first stage is a preprocessing stage in which the seeds—a select collection of partial triple systems of one-factorizations—for the main search are determined. The second stage is the main search, which consists of determining all extensions of each seed into triple systems of one-factorizations and rejecting isomorphs. The core of the second stage algorithm is an efficient exact cover algorithm [17] for completing the seeds into triple systems. Isomorphic triple systems are filtered from the output of the algorithm using the canonical construction path method [20].

In the preprocessing stage, we fix the first block, $\{u_1, v_1, v_2\}$, and construct all pairwise nonisomorphic triple systems consisting of blocks that intersect the first block. For K_{2n}, the total number of blocks in a seed is $1 + (n - 1) + 2(2n - 2) = 5n - 4$. For the sake of clarity, we now abandon a general discussion for arbitrary n and study the case $n = 6$.

The number of blocks in a seed for $n = 6$ is $5n - 4 = 26$. Up to isomorphism, the incidence matrix of these blocks is as shown in Figure 1.

To complete the 10 final blocks of Figure 1 by filling out the part A, we carry out a backtrack search with isomorph rejection using *nauty* [18] and obtain 393 pairwise nonisomorphic 26-block seeds.

Compared with the approach in [10], where a one-factor at a time is completed, we do indeed start with a one-factor—corresponding to the six first columns in Figure 1—but after that the search proceeds in a different direction. In fact, from the 27th block onwards, we do not even prescribe any order, but let the heuristic of the exact cover algorithm [17] direct the search.

3.3 Isomorph Rejection

To reject isomorphs among the generated triple systems of one-factorizations, we apply the following two tests.

The first test associates with each triple system $\mathcal{X}$ an $\mathrm{Aut}(\mathcal{X})$-orbit $m(\mathcal{X})$ of blocks in $\mathcal{X}$ such that, for any two isomorphic $\mathcal{X}, \mathcal{X}'$, every isomorphism $\mathcal{X} \to \mathcal{X}'$ maps $m(\mathcal{X})$ onto $m(\mathcal{X}')$. A triple system $\mathcal{X}$ generated by extending a seed $\mathcal{S}$ is accepted in the first test if and only if the block that intersects all the blocks in $\mathcal{S}$ occurs in $m(\mathcal{X})$.

111111	0000000000	0000000000
000000	1000000000	1000000000
000000	0100000000	0100000000
000000	0010000000	0010000000
000000	0001000000	0001000000
000000	0000100000	0000100000
000000	0000010000	0000010000
000000	0000001000	0000001000
000000	0000000100	0000000100
000000	0000000010	0000000010
000000	0000000001	0000000001
100000	1111111111	0000000000
100000	0000000000	1111111111
010000	1000000000	A
010000	0100000000	
001000	0010000000	
001000	0001000000	
000100	0000100000	
000100	0000010000	
000010	0000001000	
000010	0000000100	
000001	0000000010	
000001	0000000001	

Figure 1: The structure of seeds

The second test varies depending on the order of Aut($\mathcal{S}$). For $|\mathrm{Aut}(\mathcal{S})| \leq 1000$, the second test is an exhaustive search through elements of Aut($\mathcal{S}$) that accepts $\mathcal{X}$ if and only if $\mathcal{X}$ is the lexicographic minimum of its orbit under Aut($\mathcal{S}$). For $|\mathrm{Aut}(\mathcal{S})| > 1000$, the second test accepts $\mathcal{X}$ if and only if the canonical block graph of $\mathcal{X}$ (which is computed as a by-product of the first test) does not occur in a hash table that contains the canonical block graphs of all the triple systems encountered so far during the search for extensions of the seed $\mathcal{S}$.

A triple system is output as the representative of its isomorphism class if and only if both tests accept it. We remark that these tests are essentially identical to those employed in [15]; however, verifying that the tests function correctly also in the present case requires some work. Also the implementation of the first test differs somewhat from [15]. We proceed to describe these modifications.

A *block graph* or *line graph* of a triple system is obtained by taking one vertex for each block and inserting edges between blocks that have at least (here, exactly) one point in common. For the two isomorph rejection tests to function correctly, the triple systems of one-factorizations must be *strongly reconstructible* (see [1]) from their block graphs. In other words, for any two triple systems of one-factorizations, $\mathcal{X}$ and $\mathcal{X}'$, and their block graphs, $L(\mathcal{X})$ and $L(\mathcal{X}')$, the following implications must hold: if $L(\mathcal{X})$ and $L(\mathcal{X}')$ are isomorphic, then $\mathcal{X}$ and $\mathcal{X}'$ are isomorphic. Furthermore, every isomorphism $L(\mathcal{X}) \to L(\mathcal{X}')$ must be induced by a unique isomorphism $\mathcal{X} \to \mathcal{X}'$ (cf. [15, Theorem 2.2 and Corollary 2.6]).

Theorem 1 *For $n \geq 4$, the triple systems of one-factorizations of K_{2n} are strongly reconstructible from their block graphs.*

Proof. A clique in the block graph corresponds to a set of blocks that have pairwise exactly one point in common. Such a set of blocks is called a *sunflower* if all the blocks have the same point in common.

By a result of Deza [8, 9], a set of m triples that have pairwise exactly one point in common is a sunflower if $m > 7$; if $m = 7$, the triples form either a sunflower or a Fano plane.

Recall that a Fano plane consists of seven triples over a set of seven points, such that each point occurs in exactly three triples, and each pair of points occurs together in exactly one triple. In a triple system of a one-factorization, exactly one of the three points in every triple is in U. Thus, a putative Fano plane in a triple system of a one-factorization must contain at least one point $u_i \in U$. Furthermore, since u_i can occur only in three of the seven triples, the putative Fano plane must contain another point $u_j \in U$. By the structure of a one-factorization of a triple system, the points u_i and u_j do not occur together in a triple. On the other hand, the putative Fano plane requires these points to occur together in a triple. This contradiction shows that a triple system of a one-factorization cannot contain a Fano plane.

Consequently, for $2n - 1 \geq 7$ the maximum cliques of size $2n - 1$ in the block graph are in a one-to-one correspondence with the sunflowers induced by the $2n$ vertices in V. This enables reconstruction of the V part of the triple system: a point $v_i \in V$ appears in exactly those blocks that occur in the maximum clique that corresponds to v_i. To complete the U part of the triple system, just check the blocks that are nonintersecting in the V part to see if the corresponding vertices are joined by an edge.

Any isomorphism between block graphs must map maximum cliques onto maximum cliques, which induces a unique isomorphism between the underlying triple systems. This establishes strong reconstructibility. □

3.4 Implementation Details for Isomorph Rejection

Following the ideas in [3], we implement the first isomorph rejection test as a sequence of subtests of increasing computational difficulty. For this purpose, we require a fast invariant for distinguishing between blocks in a triple system. A *Pasch configuration*, also called a *fragment* or a *quadrilateral*, is a set of four triples of the form

$$\{u, w, y\}, \{u, x, z\}, \{v, w, z\}, \{v, x, y\}. \qquad (2)$$

Pasch configurations have been used in a number of studies as isomorphism invariants for Steiner triple systems—see [6, 7] and the references therein. Pasch configurations are also fundamental to the success of the approach in [15], where the number of times a block occurs in a Pasch configuration is used as a vertex invariant for speeding up isomorphism computations on block graphs. Exactly the same invariant is natural in the context of triple systems of one-factorizations as well. For such triple systems, a Pasch configuration takes the form

$$\{u_a, v_a, v_b\}, \{u_a, v_c, v_d\}, \{u_b, v_a, v_d\}, \{u_b, v_b, v_c\}.$$

This means that the one-factors u_a and u_b form a 4-cycle in the vertices $\{v_a, v_b, v_c, v_d\}$. (The cycle structure of a one-factorization is an important property of one-factorizations [33, 34] and a cornerstone in the approach in [10].)

The implementation of the first isomorph rejection test consists of four subtests. Let $\mathcal{X}$ be a triple system generated as an extension of a seed $\mathcal{S}$. In the first subtest, we form an ordered partition of the blocks in $\mathcal{X}$ according to the number $P(\mathcal{X}, B)$ of Pasch configurations in which a block $B \in \mathcal{X}$ occurs. The cells of the partition consist of blocks with equal $P(\mathcal{X}, B)$ value; the cells are ordered by decreasing value of $P(\mathcal{X}, B)$. The first subtest rejects $\mathcal{X}$ unless the block that induces $\mathcal{S}$ occurs in the first cell (with the maximum $P(\mathcal{X}, B)$ value).

The second subtest refines the first cell of the partition based on the quantity

$$Q(\mathcal{X}, B) = \sum_{x : x \in B} \sum_{B' : x \in B' \in \mathcal{X}} P(\mathcal{X}, B').$$

The subtest rejects $\mathcal{X}$ unless the block that induces $\mathcal{S}$ occurs in the first cell (with the maximum $P(\mathcal{X}, B)$ and $Q(\mathcal{X}, B)$ value).

The third subtest accepts $\mathcal{X}$ if the first cell consists of a single block; otherwise we proceed to the fourth and final subtest. Note that if the third subtest accepts $\mathcal{X}$, the unique block that induces $\mathcal{S}$ is fixed by all automorphisms of $\mathcal{X}$. Thus, $\mathrm{Aut}(\mathcal{X})$ is a subgroup of $\mathrm{Aut}(\mathcal{S})$.

In the fourth subtest we use *nauty* [18] to compute an $\mathrm{Aut}(\mathcal{X})$-orbit $m(\mathcal{X})$ of blocks. We input the triple system $\mathcal{X}$ into *nauty* as the block graph $L(\mathcal{X})$ together with the ordered partition of blocks resulting from the first two subtests. As a by-product of executing *nauty* on $L(\mathcal{X})$ we obtain generators for $\mathrm{Aut}(L(\mathcal{X})) \cong \mathrm{Aut}(\mathcal{X})$ together with a canonically labeled version of $L(\mathcal{X})$ that can be used for isomorph rejection in the case $|\mathrm{Aut}(\mathcal{S})| > 1000$. The orbit $m(\mathcal{X})$ is the $\mathrm{Aut}(\mathcal{X})$-orbit of blocks that maps under isomorphism to the orbit containing the first (that is, lowest numbered) vertex in the canonically labeled version of $L(\mathcal{X})$. The fourth subtest accepts $\mathcal{X}$ if and only if the block that induces $\mathcal{S}$ occurs in $m(\mathcal{X})$.

3.5 One-Factorizations of Degree $2n - 2$ and Order $2n$

Uniqueness of a regular graph of degree $2n - 2$ and order $2n$ follows directly from uniqueness of its complement graph, which is a regular graph of order $2n$ and degree 1, that is, a one-factor. Since a one-factorization of degree $2n - 2$ and order $2n$ can always be extended to a one-factorization of a complete graph, we can use a classification of the latter objects to classify the former objects.

From each one-factorization of the complete graph of order $2n$, there are $2n - 1$ one-factors to remove, and we can get $2n - 1$ one-factorizations of degree $2n - 2$. But some of these may be isomorphic, and such isomorphs must be detected. However, if we know the automorphism group of a one-factorization—in particular, the orbits of one-factors under the automorphism group—this information can be used to directly find the desired objects. Namely, the new one-factorizations we get are nonisomorphic if and only if the removed one-factors are in different automorphism orbits. As a special case, if the full automorphism group is trivial, we obtain $2n-1$ nonisomorphic one-factorizations of degree $2n - 2$.

Since we get information about the automorphism groups in classifying one-factorizations of complete graphs as described earlier, it is a straightforward task to classify the one-factorizations of degree $2n - 2$ simultaneously. Unfortunately, this approach cannot easily be generalized to graphs of order $2n$ with smaller degree than $2n - 2$ because the complement of such a graph does not necessarily admit a one-factorization.

4 The Results

The approaches in Sections 2 and 3 were used to carry out classifications of one-factorizations of regular graphs of order 12 for degrees $k \leq 6$ and $k = 10, 11$, respectively. The cases $k = 7, 8, 9$ still remain open.

4.1 Computing Resources

Before proceeding to the classification results, we briefly outline how the classification was carried out in practice. The classification runs were distributed using the batch system `autoson` [19] to a network of Linux PCs with CPU clocks ranging from 233 MHz to 1.66 GHz.

The duration of the classification of one-factorizations of k-regular graphs of order 12 was as follows. The case $k = 3$ can be solved in a few seconds on a 1.66-GHz PC, for $k = 4$ the time requirement is a few minutes, for $k = 5$ a little over six hours.

For $k = 6$, we divided the codeword-by-codeword search into 413 batch jobs, where each batch job consisted of carrying out the search starting from four of the 1652 six-codeword partial codes. In total the codeword by codeword search required approximately 120 MIPS years (years of time on a computer that executes one million instructions per second; in deriving the MIPS values we used the rough estimate that a PC running backtrack search executes one instruction in one clock cycle, 1 MIPS year corresponds to approximately 5.3 hours of CPU time on a 1.66-GHz PC). The clique search in the algorithm, see [14], took place after nine codewords had been fixed.

The coordinate by coordinate search for $k = 6$ was likewise divided into 157 batch jobs, where each job consisted of carrying out the search starting from one of the 157 three-coordinate partial codes. In total the coordinate by coordinate search required approximately 160 MIPS years.

For $k = 7, 8, 9$, we classified only the uniform one-factorizations (see the following section) by discarding all partial solutions that were not uniform. This required a little less than a day on a 1-GHz PC.

For $k = 10, 11$, one batch job consisted of carrying out the main search from one of the 393 seeds. In total, the classification for $k = 10, 11$ required 50 MIPS years, whereas 160 MIPS years were required in [10]. This suggests a performance improvement; however, it should be noted that the number of executed instructions per second is a somewhat poor performance measure across different CPU architectures and instruction sets. A classification of the one-factorizations of K_{12} can now be carried out in just under eleven days on a single 1.66-GHz PC, compared with just over 8 years of CPU time required by [10] one decade ago.

4.2 The Classification

The number of nonisomorphic one-factorizations of regular graphs of order 12 appears in Table 1 for each possible degree k, with the exception of $k = 7, 8, 9$, which remain open after this study. Also displayed in the table is the number of nonisomorphic regular graphs for each degree and order 12, from [11]. For $k = 11$, our results corroborate those obtained by Dinitz, Garnick, and McKay [10].

Table 1: One-factorizations of regular graphs of order 12

Degree	Regular graphs [11]	One-factorizations
2	9	4
3	94	157
4	1547	32,741
5	7849	5,122,910
6	7849	298,222,859
7	1547	?
8	94	?
9	9	?
10	1	5,794,885,778
11	1	526,915,620

We focus on two key properties of the classified one-factorizations: the structure of the automorphism group and the *cycle structure*, that is, the collections of cycles that result in combining distinct one-factors in all possible ways. Of interest are those one-factorizations for which the cycle structure is *uniform* in the sense that all pairs of distinct one-factors result in isomorphic collections of cycles. In particular, a one-factorization is *perfect* if the union of every pair of distinct one-factors is a Hamiltonian cycle.

Table 2 contains the number of uniform one-factorizations for each of the four possible cycle structures

$$4+4+4, \qquad 4+8, \qquad 6+6, \qquad 12.$$

The six uniform one-factorizations of K_{12} appear in [10]. The uniqueness of the type $6+6$ one-factorization of K_{12} was shown in [5] and the five perfect one-factorizations of K_{12} were classified in [26]. The other classification results for $k \geq 3$ in Table 2 are new. Note that the classification is complete in the sense that also the cases $k = 7, 8, 9$ are included.

Table 2: Uniform one-factorizations of regular graphs of order 12

Degree	4+4+4	4+8	6+6	12	Total
2	1	1	1	1	4
3	2	5	6	32	45
4	0	14	6	2,115	2,135
5	0	21	9	57,106	57,136
6	0	14	4	357,239	357,257
7	0	4	3	471,152	471,159
8	0	2	2	110,624	110,628
9	0	1	1	3,775	3,777
10	0	0	1	28	29
11	0	0	1	5	6

A uniform one-factorization of type $4+4+4$ is only possible for three graphs of order 12: the disjoint union of three copies of C_4, the disjoint union of three copies of K_4, and the union of K_4 and the cube $K_2 \times K_2 \times K_2$. For each graph the one-factorization is unique up to isomorphism.

The complement of the disjoint union of three copies of C_4 is the only 9-regular graph of order 12 that admits a uniform one-factorization of type $4+8$. This one-factorization is unique up to isomorphism and can be obtained by letting the automorphism group

$$\langle (0,1)(2,10,3,11)(6,8,7,9),\ (2,9)(3,8)(4,5)(10,11) \rangle$$

act on the two representatives of one-factor orbits

$$\{\{0,1\},\{2,3\},\{4,5\},\{6,7\},\{8,9\},\{10,11\}\},$$
$$\{\{0,3\},\{1,2\},\{4,11\},\{5,6\},\{7,8\},\{9,10\}\}.$$

4.3 Structure of Automorphism Groups

The nontrivial automorphisms of one-factorizations can be divided into those of prime order and nonprime order. A one-factorization with a nontrivial automorphism group must necessarily admit at least one automorphism of prime order. For one-factorizations of K_{12}, the types of different automorphisms were determined by Seah and Stinson [30]. There are six types of automorphisms of prime order (acting on the 12 vertices):

$$1^4 2^4, \quad 1^2 2^5, \quad 1^2 5^2, \quad 1^1 11^1, \quad 2^6, \quad 3^4.$$

Additionally, there are eight types of automorphisms of nonprime order:

$$1^4 4^2, \quad 1^2 10^1, \quad 1^2 2^1 4^2, \quad 1^2 2^1 8^1, \quad 2^2 4^2, \quad 2^1 10^1, \quad 4^3, \quad 6^2, \quad 12^1.$$

(See [13] for an analysis of the possible automorphism groups of one-factorizations of K_{12} conducted without the use of computers.)

For $k = 10$, the automorphism group of a one-factorization must be a group of automorphisms of a one-factorization of K_{12} such that the group fixes at least a single one-factor. Thus, the admissible nontrivial automorphisms form a subset of the case $k = 11$. In fact, the only type of automorphism that does not occur in the $k = 10$ case is $1^1 11^1$, which necessarily acts transitively on the 11 one-factors.

Based on our classification data, we can partition the one-factorizations with a nontrivial automorphism group into classes according to the automorphism group order and the types of prime order automorphisms (acting on the 12 vertices) that occur. This is done in Tables 4 to 9 for each applicable value k. (For reasons of page layout, the large tables appear before the references in Section 4.8.) The first column in each table contains the full automorphism group order, possibly followed by a letter to indicate division into subclasses according to the types of prime order automorphisms that occur in the class. The last column contains the number of nonisomorphic one-factorizations within a class. The last row contains the total number of nonisomorphic one-factorizations that admit an automorphism of a given type.

The automorphisms of nonprime order are summarized in Table 10 for all applicable values k. Each column in the table gives the number of nonisomorphic one-factorizations that admit an automorphism of the given type. Note that a one-factorization may admit many different types of automorphisms of nonprime order.

A reader with further interest is encouraged to contact the authors for electronic listings of the one-factorizations. For example, all the uniform one-factorizations and the one-factorizations with an automorphism group of order three or greater are readily available in electronic form. We shall now present some more details about the classified one-factorizations.

4.4 Dundas Index and Tightness Index

A quantity of interest in the study of the cycle structure of one-factorizations of complete graphs is the $\mathcal{Q}$-index [21], which is defined as follows. Let $\mathcal{F} = \{F(1), F(2), \ldots, F(2n-1)\}$ be a one-factorization of K_{2n} and let $\mathcal{Q}$ be a set of 2-regular graphs of order $2n$. Partition the one-factors into classes such that if distinct $F(i)$ and $F(j)$ are in the same class, then $F(i) \cup F(j)$ is isomorphic to a graph in $\mathcal{Q}$. The index of a partition is the size of the smallest class in it. The $\mathcal{Q}$*-index* of $\mathcal{F}$ is the maximum index of a valid partition of the one-factors in $\mathcal{F}$. The *spectrum* of a $\mathcal{Q}$-index is the set $B_{\mathcal{Q}}(2n)$ that consists of all possible values for the $\mathcal{Q}$-index over one-factorizations of K_{2n}.

A special case of the $\mathcal{Q}$-index is the *Dundas index* $D(\mathcal{F})$, where $\mathcal{Q}$ consists of a single cycle of length $2n$. Clearly, a one-factorization satisfies $D(\mathcal{F}) = 2n - 1$ if and only if it is perfect. Mendelsohn and Rosa [21] conjecture that when $2n \geq 8$, the spectrum of the Dundas index satisfies

$$B_D(2n) = \{1, 2, \ldots, n-1, 2n-1\}.$$

Previously this has been confirmed for $2n = 8$ and 10 [33]. Based on the classification data, we obtain that the conjecture holds in the case $2n = 12$; Table 3 partitions the nonisomorphic one-factorizations of K_{12} into classes based on the Dundas index.

Table 3: Dundas index in one-factorizations of K_{12}

Index	#
1	888,275
2	270,717,478
3	253,189,600
4	200,202
5	1,920,060
11	5

Another special case of the $\mathcal{Q}$-index is the *tightness index* $T(\mathcal{F})$, where $\mathcal{Q}$ consists of $n/2$ 4-cycles (n even) or one 6-cycle and $(n-3)/2$ 4-cycles (n odd). Based on the nonexistence of type $4+4+4$ uniform one-factorizations of order 12 for $k > 3$ (see Table 2), we obtain that the tightness spectrum $B_T(12)$ satisfies $B_T(12) \subseteq \{1, 2, 3\}$. Moreover, since 3 does not divide 11, we have $B_T(12) \subseteq \{1, 2\}$. For a perfect one-factorization of K_{12} we obviously have $T(\mathcal{F}) = 1$. The case $T(\mathcal{F}) = 2$ occurs for the one-factorization in Figure 2. (We remark that a number of such one-factorizations exist.) Hence, we have $B_T(12) = \{1, 2\}$.

4.5 Number of 4-Cycles

During the main search for $k = 11$ the number of 4-cycles in a one-factorization of K_{12} is easily obtained as a by-product of executing the Pasch configuration invariant. Table 11 partitions the nonisomorphic one-factorizations of K_{12} into classes according to the number of 4-cycles in the cycle structure.

The unique one-factorization of K_{12} with the maximum number of 4-cycles (81 in total) is given in Figure 2.

$F(1):$	0,1	2,3	4,5	6,7	8,9	10,11
$F(2):$	0,2	1,3	4,7	5,6	8,10	9,11
$F(3):$	0,3	1,2	4,9	5,8	6,10	7,11
$F(4):$	0,4	1,5	2,8	3,9	6,11	7,10
$F(5):$	0,5	1,4	2,11	3,10	6,8	7,9
$F(6):$	0,6	1,7	2,9	3,8	4,11	5,10
$F(7):$	0,7	1,6	2,10	3,11	4,8	5,9
$F(8):$	0,8	1,9	2,7	3,6	4,10	5,11
$F(9):$	0,9	1,10	2,6	3,4	5,7	8,11
$F(10):$	0,10	1,11	2,4	3,5	6,9	7,8
$F(11):$	0,11	1,8	2,5	3,7	4,6	9,10

Figure 2: The one-factorization of K_{12} that contains 81 4-cycles

4.6 A Digression: Order 10

In the process of developing the algorithms of this paper, they were tested against published classification results for small instances. This led to the discovery of two erroneous results on the number of nonisomorphic one-factorizations of regular graphs of order 10 in [28]: for degree 4 the number should be 310 instead of 313, and for degree 5 the number should be 3468 instead of 3472.

4.7 Correctness

To gain confidence in correctness of the classification, the classification for degrees $k \leq 6$ was carried out using both algorithms from Section 2. For $k = 10, 11$, data from the computer runs can be used to arrive at a double counting argument. During the main search, we record for each seed $\mathcal{S}_i$ the order of the automorphism group $\mathrm{Aut}(\mathcal{S}_i)$ and the total number M_i of triple systems found by the exact cover algorithm as extensions of $\mathcal{S}_i$. Similarly, for each isomorphism class $\mathcal{X}_j$ of triple systems, we record the order of the automorphism group $\mathrm{Aut}(\mathcal{X}_j)$. This allows us to count in two different ways the total number of triple systems of one-factorizations of K_{12}. By the orbit-stabilizer theorem,

$$\frac{1}{\binom{12}{2}} \sum_{i=1}^{393} \frac{11! \cdot 12! \cdot M_i}{|\mathrm{Aut}(\mathcal{S}_i)|} = \sum_{j=1}^{526{,}915{,}620} \frac{11! \cdot 12!}{|\mathrm{Aut}(\mathcal{X}_j)|}. \tag{3}$$

Note that the scaling on the left-hand side of (3) is necessary because the left-hand side sum counts every triple system once for each of the $\binom{12}{2}$ seeds that occur in a triple system. Using the classification data, both the left-hand side and the right-hand side of (3) evaluate to 10,070,314,878,246,926,155,776,000.

We can extend the double counting argument to work for $k = 10$ as follows. For each isomorphism class $\mathcal{X}'_j$ encountered for $k = 10$, we record the order of the automorphism group $\mathrm{Aut}(\mathcal{X}'_j)$. By the orbit-stabilizer theorem, the total number of triple systems for $k = 10$ is

$$\sum_{j=1}^{5{,}794{,}885{,}778} \frac{10! \cdot 12!}{|\mathrm{Aut}(\mathcal{X}'_j)|}. \tag{4}$$

These systems are in a one-to-one correspondence with the systems counted in (3): delete the last one-factor from a triple system for $k = 11$ to get a triple system with $k = 10$; conversely, append the unique missing one-factor to a triple system with $k = 10$ to get back the original triple system with $k = 11$. Evaluating (4) using the classification data gives the same result as (3), which gives us confidence that the classification for $k = 10$ is correct.

4.8 Tables

Table 4: Automorphisms of prime order for degree $k = 3$

Class	$1^8 2^2$	$1^6 2^3$	$1^6 3^2$	$1^4 2^4$	$1^3 3^3$	$1^2 2^5$	2^6	3^4	#
2304	*	*		*	*	*	*	*	1
432		*	*	*	*	*	*	*	1
192	*	*		*	*	*	*		1
72		*	*	*	*	*	*	*	1
64a	*	*		*		*	*		1
b	*			*			*		1
48		*		*		*	*	*	1
32a	*	*		*		*	*		1
b	*			*		*	*		1
24a	*			*	*				1
b				*			*	*	3
c						*	*	*	3
18					*	*		*	1
16	*			*			*		1
12a		*				*	*	*	1
b				*			*	*	1
c							*	*	2
8a		*					*		1
b				*			*		5
c						*	*		3
6a				*	*				1
b					*	*			1
4a		*				*	*		2
b				*			*		15
c						*	*		4
d							*		13
3					*				1
2a				*					6
b						*			24
c							*		25
Total	8	11	2	42	9	47	88	15	123

Table 5: Automorphisms of prime order for degree $k = 4$

Class	$1^6 2^3$	$1^6 3^2$	$1^4 2^4$	$1^2 2^5$	2^6	3^4	#
72		*	*		*	*	1
48a	*		*	*	*	*	2
b			*		*	*	2
36		*			*	*	1
24a			*		*	*	7
b				*	*	*	2
16a	*		*	*	*		4
b			*		*		1
12a		*	*		*		2
b			*		*	*	3
c					*	*	12
8a	*		*	*	*		1
b	*				*		6
c			*	*	*		3
d			*	*			1
e			*		*		44
f			*				1
g				*	*		13
6a		*			*		1
b					*	*	13
4a	*		*	*			2
b	*			*	*		2
c	*				*		2
d			*	*			6
e			*				10
f			*		*		139
g				*	*		8
h					*		258
3						*	13
2a	*						22
b			*				206
c				*			190
d					*		1,821
Total	41	5	435	234	2,348	56	2,799

Table 6: Automorphisms of prime order for degree $k = 5$

Class	$1^6 2^3$	$1^6 3^2$	$1^4 2^4$	$1^2 2^5$	$1^2 5^2$	2^6	3^4	#
240			$*$		$*$	$*$	$*$	2
120			$*$		$*$	$*$	$*$	1
72		$*$	$*$			$*$	$*$	2
48a	$*$		$*$	$*$		$*$	$*$	4
b			$*$			$*$	$*$	4
36		$*$				$*$	$*$	2
24a	$*$		$*$	$*$		$*$	$*$	1
b	$*$					$*$	$*$	2
c			$*$			$*$	$*$	12
d				$*$		$*$	$*$	7
20			$*$		$*$			1
16a	$*$		$*$	$*$		$*$		10
b			$*$			$*$		21
12a	$*$			$*$		$*$	$*$	4
b	$*$					$*$	$*$	1
c		$*$	$*$			$*$		2
d			$*$			$*$	$*$	6
e				$*$		$*$	$*$	2
f						$*$	$*$	32
10a			$*$		$*$			1
b				$*$	$*$			4
c					$*$	$*$		2
8a	$*$		$*$	$*$		$*$		18
b	$*$					$*$		8
c			$*$	$*$		$*$		11
d			$*$	$*$				1
e			$*$			$*$		198
f			$*$					1
g				$*$		$*$		48
6a	$*$						$*$	2
b		$*$	$*$					2
c		$*$				$*$		8
d				$*$			$*$	21
e						$*$	$*$	53
5					$*$			7
4a	$*$		$*$	$*$				14
b	$*$			$*$		$*$		38
c	$*$					$*$		4
d			$*$	$*$				70
e			$*$			$*$		508
f			$*$					66
g				$*$		$*$		112
h						$*$		2,094
3a		$*$						4
b							$*$	78
2a	$*$							78
b			$*$					3,476
c				$*$				7,414
d						$*$		38,046
Total	184	20	4,432	7,779	18	41,263	236	52,503

Table 7: Automorphisms of prime order for degree $k = 6$

Class	$1^6 2^3$	$1^6 3^2$	$1^4 2^4$	$1^3 3^3$	$1^2 2^5$	$1^2 5^2$	2^6	3^4	#
432	*	*	*	*	*		*	*	1
240			*			*	*	*	2
216		*		*	*		*	*	1
144	*	*	*		*		*	*	1
120a			*	*		*			1
b			*			*	*	*	1
72a		*	*				*	*	1
b			*	*			*	*	1
60						*	*	*	1
48a	*		*	*	*		*		2
b	*		*		*		*	*	1
c			*				*	*	4
36a		*	*	*					1
b		*	*	*			*	*	1
24a			*	*			*		4
b			*				*	*	27
c				*	*		*		2
d					*		*	*	2
18a		*		*			*	*	1
b			*	*				*	1
16a	*		*		*		*		7
b			*				*		48
12a	*		*	*	*				1
b		*	*				*		5
c			*				*	*	18
d				*			*		2
e							*	*	94
10a			*			*			1
b						*	*		5
9				*				*	4
8a	*		*		*		*		4
b	*						*		24
c			*		*		*		14
d			*				*		715
e			*						1
f					*		*		59
6a		*	*						3
b		*					*		6
c			*	*					22
d				*	*				4
e					*			*	2
f							*	*	183
5						*			28
4a	*		*		*				16
b	*				*		*		16
c	*						*		32
d			*		*				12
e			*				*		2,194
f			*						191
g					*		*		70
h							*		10,424
3a		*							7
b				*					396
c								*	415
2a	*								536
b			*						24,486
c					*				9,162
d							*		390,413
Total	641	28	27,787	445	9,377	39	404,386	762	439,676

Table 8: Automorphisms of prime order for degree $k = 10$

Class	1^42^4	1^22^5	1^25^2	2^6	3^4	#
240	*		*	*	*	2
60			*	*	*	1
48	*			*	*	8
32	*			*		4
24	*			*	*	24
20	*		*			2
16	*			*		182
12a	*			*	*	11
b				*	*	277
10a		*	*			7
b			*	*		4
8a	*					20
8b	*			*		1,821
6a	*				*	2
6b				*	*	713
5			*			93
4a	*			*		7,246
4b	*					947
4c				*		50,662
3					*	1,809
2a	*					102,380
b		*				5,669
c				*		2,512,801
Total	112,649	5,676	109	2,573,756	2,847	2,684,685

Table 9: Automorphisms of prime order for degree $k = 11$

Class	$1^4 2^4$	$1^2 2^5$	$1^2 5^2$	$1^1 11^1$	2^6	3^4	#
660			*	*	*	*	1
240	*		*		*	*	2
110		*	*	*			1
55			*	*			1
48	*				*	*	6
32	*				*		4
24	*				*	*	25
20	*		*				2
16	*				*		76
12a	*				*	*	14
b	*					*	1
c					*	*	123
11				*			2
10a		*	*				6
b			*		*		4
8a	*						20
b	*				*		590
6a	*					*	1
b		*				*	4
c					*	*	240
5			*				92
4a	*				*		2,401
b	*						478
c					*		11,922
3						*	669
2a	*						34,041
b		*					5,665
c					*		397,730
Total	37,661	5,676	109	5	413,138	1,086	454,121

Table 10: Automorphisms of nonprime order

Type	$k=3$	$k=4$	$k=5$	$k=6$	$k=10$	$k=11$
$1^6 6^1$	0	0	0	1	0	0
$1^4 2^2 4^1$	4	0	0	0	0	0
$1^4 2^1 6^1$	0	0	0	1	0	0
$1^4 4^2$	3	4	38	103	648	222
$1^3 3^1 6^1$	2	0	0	2	0	0
$1^2 2^3 4^1$	3	0	0	0	0	0
$1^2 2^2 6^1$	2	0	0	2	0	0
$1^2 2^1 4^2$	4	10	36	78	304	304
$1^2 2^1 8^1$	1	0	0	0	12	12
$1^2 10^1$	0	0	0	0	7	7
$1^1 2^1 3^1 6^1$	2	0	0	17	0	0
$2^4 4^1$	5	0	0	0	0	0
$2^3 3^2$	2	0	0	2	0	0
$2^3 6^1$	2	3	4	11	0	0
$2^2 4^2$	3	5	54	150	192	66
$2^1 10^1$	0	0	4	4	3	3
$3^2 6^1$	5	2	14	3	0	0
4^3	13	40	134	362	516	384
$4^1 8^1$	1	0	0	0	0	0
6^2	12	32	85	191	457	235
12^1	5	4	10	6	4	6

Table 11: Number of 4-cycles in one-factorizations of K_{12}

4-C.	#
0	185
1	2,152
2	15,881
3	76,254
4	282,578
5	829,195
6	2,036,931
7	4,288,776
8	7,930,913
9	13,088,463
10	19,551,367
11	26,712,470
12	33,746,800
13	39,724,409
14	43,909,462
15	45,838,895
16	45,474,022
17	43,057,701
18	39,127,228
19	34,233,721
20	28,954,857
21	23,755,756
22	18,955,373
23	14,762,731
24	11,241,485
25	8,400,148
26	6,155,430
27	4,453,996
28	3,168,561
29	2,240,287
30	1,553,558
31	1,082,488
32	736,461
33	505,937
34	336,998
35	231,404
36	150,563
37	104,606
38	65,710
39	46,501
40	28,340
41	20,467
42	11,817
43	9,025
44	4,843
45	4,070
46	2,004
47	1,804
48	782
49	854
50	303
51	375
52	135
53	211
54	43
55	85
56	23
57	88
58	6
59	33
60	3
61	22
62	1
63	11
65	10
67	3
69	6
71	1
73	1
81	1

References

[1] L. Babai, Automorphism groups, isomorphism, reconstruction, Handbook of Combinatorics, Vol. 2, R. L. Graham, M. Grötschel, L. Lovász (Editors), North-Holland, Amsterdam, 1995, pp. 1447–1540.

[2] G. T. Bogdanova, A. E. Brouwer, S. N. Kapralov, and P. R. J. Östergård, Error-correcting codes over an alphabet of four elements, Des. Codes Cryptogr. 23 (2001), 333–342.

[3] G. Brinkmann and B. D. McKay, Posets on up to 16 points, Order 19 (2002), 147–179.

[4] G. Butler, Fundamental Algorithms for Permutation Groups, Lecture Notes in Computer Science 559, Springer-Verlag, Berlin, 1991.

[5] P. J. Cameron, Parallelisms of Complete Designs, Cambridge University Press, Cambridge, 1976.

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

[7] M. J. Colbourn, Algorithmic aspects of combinatorial designs: A survey, Ann. Discrete Math. 26 (1985), 67–136.

[8] M. Deza, Une propriété extrémale des plans projectifs finis dans une classe de codes équidistants, Discrete Math. 6 (1973), 343–352.

[9] M. Deza, Solution d'un probléme de Erdős–Lovász, J. Combin. Theory Ser. B 16 (1974), 166–167.

[10] J. H. Dinitz, D. K. Garnick, and B. D. McKay, There are 526,915,620 nonisomorphic one-factorizations of K_{12}, J. Combin. Des. 2 (1994), 273–285.

[11] I. A. Faradžev, Constructive enumeration of combinatorial objects, Problèmes combinatoires et théorie des graphes (Univ. Orsay, Orsay, 1976), Colloq. Internat. CNRS 260, Paris, 1978, pp. 131–135.

[12] E. N. Gelling and R. E. Odeh, On 1-factorizations of the complete graph and the relationship to round-robin schedules, Congr. Numer. 9 (1974), 213–221.

[13] E. Ihrig and E. Petrie, Automorphism groups of 1-factorizations of K_{12}, Congr. Numer. 109 (1995), 179–192.

[14] P. Kaski and P. R. J. Östergård, There exists no $(15, 5, 4)$ RBIBD, J. Combin. Des. 9 (2001) 357–362.

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

[16] P. Kaski and P. R. J. Östergård, Miscellaneous classification results for 2-designs, Discrete Math. 280 (2004) 65–75.

[17] D. E. Knuth, Dancing links, Millennial Perspectives in Computer Science, J. Davies, B. Roscoe, and J. Woodcock (Editors), Palgrave Macmillan, Basingstoke, 2000, pp. 187–214.

[18] B. D. McKay, *nauty* user's guide (version 1.5), Technical Report TR-CS-90-02, Computer Science Department, Australian National University, 1990.

[19] 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.

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

[21] E. Mendelsohn and A. Rosa, On some properties of 1-factorizations of complete graphs, Congr. Numer. 24 (1979), 739–752.

[22] M. Meringer, Fast generation of regular graphs and construction of cages, J. Graph Theory 30 (1999), 137–146.

[23] R. C. Mullin and H.-D. O. F. Gronau, PBDs and GDDs: The basics, The CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz (Editors), CRC Press, Boca Raton, 1996, pp. 185–193.

[24] P. R. J. Östergård, Classification of binary/ternary one-error-correcting codes, Discrete Math. 223 (2000), 253–262.

[25] P. R. J. Östergård and U. Blass, On the size of optimal binary codes of length 9 and covering radius 1, IEEE Trans. Inform. Theory 47 (2001), 2556–2557.

[26] A. P. Petrenyuk and A. Y. Petrenyuk, Intersection of perfect 1-factorizations of complete graphs, Cybernetics 16 (1980), 6–9.

[27] R. C. Read, Every one a winner; or, How to avoid isomorphism search when cataloguing combinatorial configurations, Ann. Discrete Math. 2 (1978), 107–120.

[28] A. Rosa and D. R. Stinson, One-factorizations of regular graphs and Howell designs of small order, Utilitas Math. 29 (1986), 99–124.

[29] E. Seah and D. R. Stinson, An enumeration of non-isomorphic one-factorizations and Howell designs for the graph K_{10} minus a one-factor, Ars Combin. 21 (1986), 145–161.

[30] E. Seah and D. R. Stinson, On the enumeration of one-factorizations of complete graphs containing prescribed automorphism groups, Math. Comp. 50 (1988), 607–618.

[31] N. V. Semakov and V. A. Zinov'ev, Equidistant q-ary codes with maximal distance and resolvable balanced incomplete block designs, Problemy Peredachi Informatsii 4(2) (1968), 3–10. (Translated from Russian in Problems Inform. Transmission 4(2) (1968), 1–7.)

[32] A. Seress, Permutation Group Algorithms, Cambridge University Press, Cambridge, 2003.

[33] W. D. Wallis, One-factorizations of complete graphs, Contemporary Design Theory: A Collection of Surveys, J. H. Dinitz and D. R. Stinson (Editors), Wiley, New York, 1992, pp. 593–631.

[34] W. D. Wallis, One-Factorizations, Kluwer, Dordrecht, 1997.