

There are  $1,132,835,421,602,062,347$   
nonisomorphic one-factorizations of  $K_{14}$

Petteri Kaski

Helsinki Institute for Information Technology HIIT  
University of Helsinki, Department of Computer Science  
P.O. Box 68, 00014 University of Helsinki, Finland  
E-mail: [petteri.kaski@cs.helsinki.fi](mailto:petteri.kaski@cs.helsinki.fi)

Patric R. J. Östergård

Department of Communications and Networking  
Helsinki University of Technology  
P.O. Box 3000, 02015 TKK, Finland  
E-mail: [patric.ostergard@tkk.fi](mailto:patric.ostergard@tkk.fi)

### Abstract

We establish by means of a computer search that a complete graph on 14 vertices has  $98,758,655,816,833,727,741,338,583,040$  distinct and  $1,132,835,421,602,062,347$  nonisomorphic one-factorizations. The enumeration is constructive for the  $10,305,262,573$  isomorphism classes that admit a nontrivial automorphism.

*Keywords:* automorphism group; classification; one-factorization

## 1 Introduction

A *one-factor* of a graph is a 1-regular spanning subgraph. A *one-factorization* of a graph is a set of one-factors such that every edge of the graph occurs in a unique one-factor. Two one-factorizations are *isomorphic* if there is a bijection between the vertex sets of the underlying graphs that maps the

one-factors in one one-factorization onto the one-factors of the other. One-factorizations of complete graphs are basic combinatorial objects with a wide variety of applications [1, 18, 19].

A fundamental problem for any family of combinatorial objects is that of classifying these for small parameters, that is, of listing exactly one representative from each isomorphism class. The complete graph  $K_n$  on  $n$  vertices has, up to isomorphism, a unique one-factorization for  $n = 2, 4, 6$ . The graph  $K_8$  has 6 nonisomorphic one-factorizations; this result was obtained by Dickson and Safford [2] almost exactly one century ago. In 1974, Gelling and Odeh [4] published the result that  $K_{10}$  has 396 nonisomorphic one-factorizations. A result by Dinitz, Garnick, and McKay [3] showing that there are 526,915,620 nonisomorphic one-factorizations of  $K_{12}$  appeared exactly two decades after the result by Gelling and Odeh. The result for  $K_{12}$  was one of the most extensive computer classifications at that time, and—with an estimated  $1.132 \times 10^{18}$  nonisomorphic one-factorizations [3] of  $K_{14}$ —the current limit for methods that rely on constructively listing one specimen from each isomorphism class was thereby reached.

For many types of combinatorial objects, no essentially faster method than explicitly constructing one representative from each isomorphism class is known for counting the number of isomorphism classes. However, in cases where one is able to count the “labeled” objects substantially faster, the following alternative approach for determining the number of isomorphism classes becomes possible.

Let  $\Gamma$  be a finite group that acts on a finite set  $\Omega$ . Denote by  $N_i$  the number of orbits on  $\Omega$  whose elements have stabilizer subgroups of order  $i$  in  $\Gamma$ . Then, by the Orbit-Stabilizer Theorem,

$$|\Omega| = |\Gamma| \sum_i \frac{N_i}{i}. \quad (1)$$

If we know  $|\Gamma|$ ,  $|\Omega|$ , and  $N_i$  for each  $i \geq 2$ , then we can solve (1) for  $N_1$  and thereby obtain the number of orbits  $\sum_i N_i$ . This observation can be applied in a combinatorial context by considering a group action for which (i) the set  $\Omega$  consists of the “labeled” objects; (ii) the orbits of  $\Gamma$  on  $\Omega$  correspond to the isomorphism classes of objects (the “unlabeled” objects); and (iii) the stabilizer subgroup of an object under the action of  $\Gamma$  corresponds to the automorphism group of the object. In particular, the values  $N_i$  for  $i \geq 2$  are obtained by classifying up to isomorphism the objects that have a nontrivial automorphism group.

Latin squares and one-factorizations of complete graphs are two types of objects for which the outlined idea is effective. Recently, McKay, Meynert, and Myrvold [14] successfully applied this general idea to the problem of enumerating Latin squares. In the present paper, it is applied to the problem of enumerating one-factorizations of the complete graph  $K_{14}$ .

The approach for counting the number of distinct one-factorizations of  $K_{14}$  is discussed in Section 2; there are 98,758,655,816,833,727,741,338,583,040 such objects. Classification of one-factorizations of  $K_{14}$  with a nontrivial automorphism group is considered in Section 3. Application of (1) then reveals that there are 1,132,835,421,602,062,347 isomorphism classes of one-factorizations of  $K_{14}$ . Particular emphasis is put on verifying the correctness of these computational results.

*Conventions.* For standard graph-theoretic terminology we refer to [20]. All graphs considered in this paper are undirected and without multiple edges or loops. For a graph  $G$ , denote by  $\bar{G}$  the complement of  $G$ , by  $\mathcal{F}(G)$  the set of all one-factors of  $G$ , by  $[G]$  the isomorphism class of  $G$ , by  $\text{Aut}(G)$  the automorphism group of  $G$ , by  $\text{LF}(G)$  the number of distinct one-factorizations of  $G$ , and by  $\text{NF}(G)$  the number of nonisomorphic one-factorizations of  $G$ .

## 2 Counting distinct one-factorizations

To count the number of distinct one-factorizations of  $K_{14}$ , that is, to compute  $\text{LF}(K_{14})$ , we essentially rely on a recursion pioneered by Dinitz, Garnick, and McKay [3, Sect. 4], who used the recursion to check their constructive enumeration of the isomorphism classes of one-factorizations of  $K_{12}$ .

In our case, however, we do not have an independently computed value of  $\text{LF}(K_{14})$  available. To gain confidence that the computed value of  $\text{LF}(K_{14})$  is correct, we will use (i) a new “forward accumulation” technique to compute the value of  $\text{LF}(K_{14})$ ; and (ii) Dinitz–Garnick–McKay recursion to *check* both the intermediate results and the final computed value.

It is convenient to start by describing Dinitz–Garnick–McKay recursion and then proceed to describe the new technique.

### 2.1 Dinitz–Garnick–McKay recursion

Consider a graph  $G$  that has a one-factorization. Such a graph is necessarily  $k$ -regular for some nonnegative integer  $k$ . Avoiding trivial cases, we assume

that  $k \geq 2$ . Let  $F$  be a one-factor in  $G$ , and denote by  $G - F$  the  $(k - 1)$ -regular graph obtained by deleting the edges of  $F$  from  $G$ . To arrive at a recursion, suppose that we know the value of  $\text{LF}(G - F)$  for every one-factor  $F$  of  $G$ . By counting in two different ways the number of distinct one-factorizations of  $G$  with one individualized one-factor, we have

$$k \cdot \text{LF}(G) = \sum_{F \in \mathcal{F}(G)} \text{LF}(G - F). \quad (2)$$

In particular, if we know  $\text{LF}(H)$  for every  $(k - 1)$ -regular graph  $H$ , then we can compute  $\text{LF}(G)$  for any given  $k$ -regular graph  $G$  via (2). This is *Dinitz–Garnick–McKay recursion* from  $(k - 1)$ -regular to  $k$ -regular graphs.

In practice it suffices to evaluate the right-hand side of (2) only for one-factors that contain a fixed edge of  $G$ , in which case the multiplication by  $k$  on the left-hand side is not necessary. Similarly,  $\text{LF}(H)$  needs to be computed for only one graph  $H$  in each isomorphism class of regular graphs.

Dinitz–Garnick–McKay recursion has the property that it considers each  $k$ -regular graph in turn, and “looks back” at the  $(k - 1)$ -regular graphs. We describe next a new technique that “looks forward” at the  $k$ -regular graphs from each  $(k - 1)$ -regular graph in turn.

## 2.2 Forward accumulation

The forward accumulation approach is based on the immediate observation that every  $k$ -regular one-factorizable graph  $G$  can be decomposed into a union  $G = H \cup F$  of (i) a  $(k - 1)$ -regular one-factorizable graph  $H$ ; and (ii) a one-factor  $F \in \mathcal{F}(\bar{H})$ . Put otherwise, we can visit every isomorphism class of  $k$ -regular one-factorizable graphs by the following procedure: for each isomorphism class  $[H]$  of  $(k - 1)$ -regular one-factorizable graphs, consider exactly one graph  $H$  from the isomorphism class; for each such graph  $H$ , consider each one-factor  $F \in \mathcal{F}(\bar{H})$ ; for each such pair  $(H, F)$ , visit the isomorphism class  $[H \cup F]$ .

To compute the value  $\text{LF}(G)$  for each visited isomorphism class  $[G]$ , we associate with  $[G]$  an accumulator variable  $x_{[G]}$  that is initially set to zero and incremented whenever  $[G]$  is visited. Our objective is to have the value  $k \cdot \text{LF}(G)$  in the accumulator when the visiting procedure halts. To determine an appropriate increment to  $x_{[G]}$  on each visit, we proceed to analyze the visiting procedure in more detail.

To this end, consider the set of *all* pairs  $(H, F)$  such that  $H$  is a  $(k - 1)$ -regular one-factorizable graph and  $F \in \mathcal{F}(\bar{H})$ . Let us view two such pairs as *isomorphic* if one can be obtained from the other by relabeling the vertices. The following lemmata are immediate consequences of the Orbit-Stabilizer Theorem.

**Lemma 1** *Any graph  $G$  in the class  $[H \cup F]$  admits exactly*

$$\sigma(H, F) = \frac{|\text{Aut}(H \cup F)|}{|\text{Aut}(H) \cap \text{Aut}(F)|}$$

*decompositions  $G = H' \cup F'$  into pairs  $(H', F')$  in the class  $[(H, F)]$ .*

**Lemma 2** *The procedure visits a class  $[H \cup F]$  exactly*

$$\tau(H, F) = \frac{|\text{Aut}(H)|}{|\text{Aut}(H) \cap \text{Aut}(F)|}$$

*times via pairs  $(H', F')$  in the class  $[(H, F)]$ .*

It now follows from Lemma 1 and (2) that, for any  $k$ -regular graph  $G$ ,

$$\sum_{[(H,F)]:H\cup F=G} \sigma(H, F) \cdot \text{LF}(H) = \sum_{(H,F):H\cup F=G} \text{LF}(H) = k \cdot \text{LF}(G). \quad (3)$$

This observation enables us to accumulate the value  $k \cdot \text{LF}(G)$  for each  $k$ -regular  $G$ . Namely, each time  $[G]$  is visited via a pair  $(H, F)$ , we increment  $x_{[G]}$  by the rule

$$x_{[G]} \leftarrow x_{[G]} + \sigma(H, F) \cdot \tau(H, F)^{-1} \cdot \text{LF}(H). \quad (4)$$

Equivalently, for each pair  $(H, F)$  considered by the visiting procedure, we apply the rule

$$x_{[H\cup F]} \leftarrow x_{[H\cup F]} + \frac{|\text{Aut}(H \cup F)|}{|\text{Aut}(H)|} \cdot \text{LF}(H). \quad (5)$$

**Lemma 3** *The total accumulation to  $x_{[G]}$  is  $k \cdot \text{LF}(G)$ .*

*Proof.* By Lemma 2 and (4), the total accumulation to  $x_{[G]}$  from a class  $[(H, F)]$  satisfying  $[H \cup F] = [G]$  is  $\sigma(H, F) \cdot \text{LF}(H)$ . Taking the sum over all such classes, the claim follows by (3).  $\square$

### 2.3 Implementation details

Starting with the empty graph of order 14, that is, the unique 0-regular graph on 14 vertices, we use the forward accumulation technique for each  $k = 1, 2, \dots, 13$  in turn to compute both (i) exactly one graph  $G$  from each isomorphism class of  $k$ -regular one-factorizable graphs; and (ii) the value  $\text{LF}(G)$  for the graphs  $G$ .

The representative graph  $G$  in an isomorphism class is the canonical form computed by *nauty* (version 2.2) [11] using the built-in `adjtriang` vertex invariant. The canonical form is stored in 16 bytes of memory as a bit map of the upper triangle of the adjacency matrix; the associated accumulator variable uses 32 bytes. To enable rapid searching of these 48-byte records, we use an open-addressing hash table with  $10^9$  entries indexed by 4-byte hash values of the 16-byte bit maps.

We use the GNU Multiple Precision Arithmetic Library [5] to carry out the accumulator arithmetic. An elementary backtrack algorithm suffices for listing the one-factors  $F \in \mathcal{F}(\bar{H})$ .

The number of isomorphism classes of  $k$ -regular one-factorizable graphs on 14 vertices is, for  $k = 0, 1, \dots, 13$ ,

$$1, 1, 4, 504, 87977, 3459360, 21609293, 21609301, 3459386, 88193, 540, 13, 1, 1.$$

The performance bottleneck is at  $k = 7$ , where 43,218,594 records need to be stored, occupying about 6 GB of memory together with the hash table (cf. [15]). In terms of running time, the entire computation (including the correctness checks described in what follows) took about 13 days on a Linux PC with a 3.66-GHz Intel Xeon CPU and 32 GB of main memory.

### 2.4 Correctness checks

Based on forward accumulation, we have the value  $\text{LF}(G)$  for every regular graph  $G$  on 14 vertices. In particular,

$$\text{LF}(K_{14}) = 98,758,655,816,833,727,741,338,583,040. \quad (6)$$

As a first check, we use Dinitz–Garnick–McKay recursion to verify the  $\text{LF}(G)$  values for each regular  $G$ . The same values are obtained both using forward accumulation and Dinitz–Garnick–McKay recursion.

As a second check, we use Meringer’s classification program **genreg** [16] to generate all the regular graphs on 14 vertices, and then filter out the

graphs that do not have a one-factorization. The obtained graphs agree with those obtained by forward accumulation.

As a third check, we use the following observation due to Dinitz, Garnick, and McKay. Taking the sum over all isomorphism classes  $[G]$  of  $k$ -regular graphs on  $n$  vertices, we have

$$\text{LF}(K_n) = \binom{n-1}{k}^{-1} \sum_{[G]} \frac{n!}{|\text{Aut}(G)|} \cdot \text{LF}(G) \cdot \text{LF}(\bar{G}). \quad (7)$$

Moreover, this holds for every  $k = 0, 1, 2, \dots, n-1$ . Using data obtained from forward accumulation, we evaluate the right-hand side of (7) for  $n = 14$  and  $k = 0, 1, \dots, 13$ . In each case the computed value agrees with (6).

To enable further checks, we display in Table 1 the value

$$\sum_{[G]} \frac{14!}{|\text{Aut}(G)|} \cdot \text{LF}(G)$$

for each  $k = 0, 1, \dots, 13$ , where the sum is taken over all isomorphism classes  $[G]$  of  $k$ -regular graphs on 14 vertices. Put otherwise, the tabulated value is the number of distinct one-factorizations of  $k$ -regular graphs on a fixed set of 14 vertices; cf. [3, Table 7], where a slightly different quantity is tabulated, however.

### 3 Counting nonisomorphic one-factorizations

Our primary objective in this section is to classify the one-factorizations of  $K_{14}$  with nontrivial automorphisms. Once this classification is available, it is a simple matter to determine  $\text{NF}(K_{14})$  based on  $\text{LF}(K_{14})$  and the classification data. Before presenting the classification approach, we discuss a representation for one-factorizations of the complete graph in the framework of group divisible designs, and narrow down the automorphisms that need to be considered to obtain a complete classification.

#### 3.1 One-factorizations as group divisible designs

It is well known that a one-factorization of  $K_n$  can be viewed as a particular type of group divisible design (GDD). For our purposes this representation

$k$	Distinct one-factorizations
1	135135
2	5338373040
3	78634135419840
4	461142306338313600
5	1078882420304271623040
6	972197327694773750169600
7	315828427387711768964628480
8	33491835583595013396417085440
9	1006698095378044123991615078400
10	7024525682952576878777802424320
11	8573318527281503086919968358400
12	1283862525618838460637401579520
13	98758655816833727741338583040

Table 1: Distinct one-factorizations of  $k$ -regular graphs on 14 vertices

will be convenient as it enables us to immediately prescribe the action of an automorphism not only on the vertices but also on the one-factors.

To develop the GDD representation, let  $U = \{u_1, u_2, \dots, u_{n-1}\}$  be a set with one element for each one-factor, and let  $V = \{v_1, v_2, \dots, v_n\}$  be a set (disjoint from  $U$ ) with one element for each vertex of  $K_n$ . The elements of  $U \cup V$  are called *points*. We can now let a 3-subset of the form  $\{u_k, v_i, v_j\}$  carry the information that the edge  $\{v_i, v_j\}$  occurs in the one-factor  $u_k$ . In this setting, a *one-factorization* of  $K_n$  is a tuple  $\mathcal{X} = (U, V, \mathcal{B})$ , where  $\mathcal{B}$  is a set of 3-subsets of points, called *blocks*, such that

- (a) for all  $1 \leq k \leq n-1$  and  $1 \leq i \leq n$ , the pair  $\{u_k, v_i\}$  occurs in a unique block;
- (b) for all  $1 \leq i < j \leq n$ , the pair  $\{v_i, v_j\}$  occurs in a unique block; and
- (c) for all  $1 \leq k < \ell \leq n-1$ , no block contains the pair  $\{u_k, u_\ell\}$ .

A reader acquainted with GDDs immediately observes that  $\mathcal{X}$  is in fact a GDD with group type  $(n-1)^1 1^n$ , block size  $k = 3$ , and index  $\lambda = 1$ ; however, here we find it more convenient to speak of one-factorizations.

Two one-factorizations,  $\mathcal{X} = (U, V, \mathcal{B})$  and  $\mathcal{X}' = (U', V', \mathcal{B}')$ , are *isomorphic* if there exists a bijection  $\varphi : U \cup V \rightarrow U' \cup V'$  such that  $\varphi(U) = U'$ ,  $\varphi(V) = V'$ , and  $\varphi(\mathcal{B}) = \mathcal{B}'$ . Such a  $\varphi$  is an *isomorphism* from  $\mathcal{X}$  onto  $\mathcal{X}'$ . An isomorphism of  $\mathcal{X}$  onto itself is an *automorphism* of  $\mathcal{X}$ . We denote by  $\text{Aut}(\mathcal{X})$  the automorphism group of  $\mathcal{X}$ . Note that the restriction of an isomorphism  $\varphi$  to  $V$  uniquely determines  $\varphi$  on  $U$ . It follows that the standard (graphical) and the GDD representations of one-factorizations of  $K_n$  are equivalent for purposes of classification up to isomorphism.

### 3.2 Automorphisms of one-factorizations

Any nontrivial group has a subgroup of prime order, and our classification considers all possible such groups that can occur as a group of automorphisms of a one-factorization. Ihrig and Petrie [6] carried out an extensive study of *all* possible automorphisms for one-factorizations—in particular, for the complete graph  $K_{12}$ —but we indeed only need those with prime order.

It is convenient to assume in what follows that the sets  $U$  and  $V$  are arbitrary but fixed. This enables the following two simplifications. First, isomorphisms between one-factorizations are permutations of  $U \cup V$  that fix  $U$  and  $V$  setwise. Denote by  $\Gamma$  the group of all such permutations of  $U \cup V$ . Second, we can identify a one-factorization  $\mathcal{X}$  with its set of blocks  $\mathcal{B}$ .

A group element  $\alpha \in \Gamma$  of prime order  $p$  is determined up to conjugation in  $\Gamma$  by the number of fixed points it has in the sets  $U$  and  $V$ . Denote by  $f_U$  and  $f_V$  the number of fixed points of  $\alpha$  in  $U$  and  $V$ , respectively. Because the cycle decomposition of  $\alpha$  consists only of fixed points and  $p$ -cycles, it is immediate that  $p$  must divide both  $|U| - f_U = n - 1 - f_U$  and  $|V| - f_V = n - f_V$ . Not all such types  $(p, f_U, f_V)$  define automorphisms of one-factorizations, however.

We proceed to narrow down the possible types  $(p, f_U, f_V)$ . The following lemma is analogous to [7, Lemma 32]; see also [17, Lemma 4.1].

**Lemma 4** *Let  $\alpha$  be an automorphism of a one-factorization  $\mathcal{X}$  with  $f_U \geq 1$  and  $f_V \geq 1$ . Then,  $\mathcal{X}$  restricted to the fixed points of  $\alpha$  forms a one-factorization. In particular,  $f_U = f_V - 1$  and  $f_V$  is even.*

*Proof.* Let  $x$  and  $y$  be two distinct points fixed by  $\alpha$ , at least one of which is in  $V$ . Such points clearly exist if  $f_U \geq 1$  and  $f_V \geq 1$ . Then there is exactly one block  $\{x, y, z\}$  of  $\mathcal{X}$  that contains both  $x$  and  $y$ . Since  $\alpha$  is an

automorphism of  $\mathcal{X}$ , also  $\{x, y, \alpha(z)\}$  is a block, and thus  $\alpha(z) = z$ . This shows that no block of  $\mathcal{X}$  intersects the set of fixed points of  $\alpha$  in 2 points. Consequently, every block of  $\mathcal{X}$  intersects the set of fixed points in 0, 1, or 3 points. Disregarding all other blocks except those intersecting in 3 points, we obtain a set system  $\mathcal{X}' = (U', V', \mathcal{B}')$ , where  $U'$  and  $V'$  are the sets of points fixed by  $\alpha$  in  $U$  and  $V$ , respectively.

To see that  $\mathcal{X}'$  is a one-factorization of a complete graph of order  $n' = f_V$ , we first count in two different ways the tuples  $(u_k, v_i, B)$  such that  $u_k \in U'$ ,  $v_i \in V'$ ,  $B \in \mathcal{B}'$ , and  $\{u_k, v_i\} \subseteq B$ . We have  $|U'| \cdot |V'| = 2|\mathcal{B}'|$ . As all 2-subsets of  $V'$  occur in a unique block, we have  $|\mathcal{B}'| = |V'|(|V'| - 1)/2$ . Combining the two equalities and noticing that  $|U'| = f_U$  and  $|V'| = f_V$ , we get  $f_U = f_V - 1$ . Furthermore, it follows that  $\mathcal{X}'$  meets the requirements (a), (b), and (c) in the definition of a one-factorization of a complete graph for  $n' = f_V$ . In particular,  $f_V$  must then be even.  $\square$

The following lemma is due to Seah and Stinson [17, Lemmata 4.2 and 4.3].

**Lemma 5** *Any nonidentity automorphism of a one-factorization satisfies  $f_V \leq n/2$ . Equality holds only if the automorphism has order 2.*

The following two lemmata rule out certain automorphism types with  $p = 2$ ; the argument in the proof of the second one is similar to that of [7, Lemma 33].

**Lemma 6** *Let  $n \equiv 2 \pmod{4}$ . If  $p = 2$  and  $f_V = 0$  for an automorphism of a one-factorization, then  $f_U \leq n/2$ .*

*Proof.* With the objective of bounding  $f_U$ , consider any one-factor  $u_k \in U$  fixed by  $\alpha$ . Since the number of edges in the one-factor is  $n/2 \equiv 1 \pmod{2}$ , not all orbits of edges in the one-factor can have size 2. Thus, because  $f_V = 0$ , there is at least one block  $\{u_k, v_i, v_j\}$  such that  $(v_i v_j)$  is a 2-cycle of  $\alpha$ . The number of such 2-cycles in  $\alpha$  is  $n/2$ , and thereby  $f_U \leq n/2$ .  $\square$

**Lemma 7** *Let  $n \equiv 4 \pmod{8}$  or  $n \equiv 6 \pmod{8}$ . If  $p = 2$  and  $f_V = 0$  for an automorphism of a one-factorization, then  $f_U \neq 1$ .*

*Proof.* To reach a contradiction, assume that  $\alpha$  is an automorphism of a one-factorization with  $p = 2$ ,  $f_V = 0$ , and  $f_U = 1$ . Without loss of generality we

$p$	$f_U$	$f_V$	1F	V1	V2	Seeds
2	1	2	-	F	M	2579
2	3	0	F	M	-	695
2	3	4	-	F	M	10256
2	5	0	F	M	-	894
2	5	6	-	F	M	1206
2	7	0	F	M	-	447
3	1	2	F	F	F	65
5	3	4	F	F	F	8
7	6	0	M	M	M	9
13	0	1	M	F	M	14

Table 2: Automorphism types

may assume that  $U \cup V = \{-(n-1), -(n-2), \dots, n-1\}$  and that  $\alpha(x) = -x$  for all  $x \in U \cup V$ . In particular, because  $\alpha$  is an automorphism, both  $U$  and  $V$  are closed under negation. Because no pair of points occurs in more than one block, it follows by  $\alpha(\{x, -y, y\}) = \{\alpha(x), y, -y\}$  that  $\alpha(x) = x$  and  $x = 0$ . In particular, each of the  $n/2$  pairs of the form  $\{-y, y\} \subseteq V$  occurs in a block of the form  $\{0, -y, y\}$ . All the remaining blocks are moved by  $\alpha$ , that is,  $\{x, y, z\}$  is a block if and only if  $\{-x, -y, -z\}$  is a block, and each such pair of blocks contains either 0 or 4 pairs of points with opposite signs. Considering pairs of points with either one point in  $U$  and one in  $V$  or two points in  $V$ , there are  $2 \cdot (n-2)/2 \cdot n/2 + (n/2)^2$  such pairs of points with opposite signs. The fixed blocks account for  $n/2$  occurrences, so the moved blocks must account for  $3 \cdot n/2 \cdot (n/2 - 1)$  occurrences. For  $n \equiv 4 \pmod{8}$  and  $n \equiv 6 \pmod{8}$  this number is not divisible by 4, a contradiction.  $\square$

For  $n = 14$ , prime orders  $p = 2, 3, 5, 7, 11, 13$  need to be considered. By applying Lemmata 4 to 7, all other automorphisms types  $(p, f_U, f_V)$  except those listed in Table 2 are excluded. The last four columns of the table—1F, V1, V2, and Seeds—are related to the main search to be discussed in the next section.

### 3.3 The classification

We now present our approach for classifying the one-factorizations that admit at least one automorphism of the types in Table 2. For this task we essentially rely on the framework established in [7].

The classification is based on certain substructures, to be called *seeds*, at least one of which is contained in every one-factorization that we want to classify. Due to the requirement of nontrivial symmetry, the precise definition of a seed will unfortunately be somewhat technical. A simplified intuition to keep in mind is as follows. Consider an arbitrary one-factorization,  $\mathcal{X}$ , and (by some rule) select a set  $T$  of points. Let  $\mathcal{S}$  be the set of blocks in  $\mathcal{X}$  that have nonempty intersection with  $T$ . Now, based on the defining properties of one-factorizations (and the rule for selecting  $T$ ), we can anticipate the structure of  $\mathcal{S}$  without knowing all the possible  $\mathcal{X}$  explicitly, and classify all possible  $\mathcal{S}$  up to isomorphism. These sets  $\mathcal{S}$  will intuitively be the seeds; the added technicality follows because (i) we insist on nontrivial symmetry in the form of a prime-order group of automorphisms  $\Pi \leq \Gamma$ ; and (ii) we must make precise the structure of  $\mathcal{S}$  in relation to  $\Pi$  and  $T$ , without reference to any containing one-factorization.

The technical definition of a seed is as follows. Let  $\Pi \leq \Gamma$  be a prime-order subgroup whose nonidentity elements have one of the types in Table 2. Let  $T \subseteq U \cup V$  be a set of points such that the type of  $\Pi$  and the columns 1F, V1, and V2 in Table 2 determine the size and composition of  $T$  in relation to  $\Pi$  as follows. The size of  $T$  is determined by the number of columns containing either an F (“fixed”) or an M (“moved”). The column 1F (“one-factor”) indicates whether  $T$  contains a point from  $U$  and whether the point is fixed or moved by  $\Pi$ . The columns V1 (“first vertex”) and V2 (“second vertex”) indicate whether  $T$  contains points from  $V$  and whether these points are fixed or moved by  $\Pi$ . (For example, for type  $p = 3$ ,  $f_U = 1$ ,  $f_V = 2$ , the set  $T$  consists of one element of  $U$  fixed by  $\Pi$  and two elements of  $V$  both moved by  $\Pi$ .) Finally, let  $\mathcal{S}$  be a union of  $\Pi$ -orbits of 3-subsets of  $U \cup V$  such that, referring to the elements of  $\mathcal{S}$  as blocks,

- (a') for all  $1 \leq k \leq n - 1$  and  $1 \leq i \leq n$ , the pair  $\{u_k, v_i\}$  occurs in at most one block;
- (b') for all  $1 \leq i < j \leq n$ , the pair  $\{v_i, v_j\}$  occurs in at most one block;
- (c') for all  $1 \leq k < \ell \leq n - 1$ , no block contains the pair  $\{u_k, u_\ell\}$ ;

- (d') for every  $u_k \in T \cap U$ , the point  $u_k$  occurs in exactly  $n/2$  blocks;
- (e') for every  $v_i \in T \cap V$ , the point  $v_i$  occurs in exactly  $n - 1$  blocks;
- (f') the set  $T$  has nonempty intersection with at least one block on every  $\Pi$ -orbit on  $\mathcal{S}$ ; and
- (g') the set  $T$  occurs in at least one block.

Each tuple  $(\Pi, T, \mathcal{S})$  meeting these requirements is called a *seed*. Two seeds,  $(\Pi, T, \mathcal{S})$  and  $(\Pi', T', \mathcal{S}')$ , are *isomorphic* if there exists a  $\gamma \in \Gamma$  such that  $\gamma\Pi\gamma^{-1} = \Pi'$ ,  $\gamma(T) = T'$ , and  $\gamma(\mathcal{S}) = \mathcal{S}'$ . The permutation  $\gamma$  is an *isomorphism* of  $(\Pi, T, \mathcal{S})$  onto  $(\Pi', T', \mathcal{S}')$ . An *automorphism* of a seed is an isomorphism of the seed onto itself. We denote by  $\text{Aut}(\Pi, T, \mathcal{S})$  the automorphism group of a seed. A one-factorization  $\mathcal{X}$  *contains* (or *extends*) a seed  $(\Pi, T, \mathcal{S})$  if  $\Pi \leq \text{Aut}(\mathcal{X})$  and  $\mathcal{S} \subseteq \mathcal{X}$ .

We classify the seeds up to isomorphism by enlarging the set  $T$  one point at a time using the algorithms described in [7]. The number of nonisomorphic seeds associated with each automorphism type is displayed in the column Seeds in Table 2.

Because every one-factorization with a nontrivial automorphism group contains at least one seed, we can visit every isomorphism class of one-factorizations by extending each classified seed in all possible ways. The task of finding all one-factorizations that contain a given seed  $(\Pi, T, \mathcal{S})$  is an instance of the *exact cover problem*. Put otherwise, we must cover the remaining uncovered pairs of points of the form  $\{u_k, v_i\}$  and  $\{v_i, v_j\}$  exactly once in all possible ways using  $\Pi$ -orbits of triples of the form  $\{u_k, v_i, v_j\}$ ; each triple covers the pairs that occur in it. For algorithms, we refer to [9, 10].

We reject isomorphs among the visited one-factorizations using the framework in [7], which is an instantiation of the canonical augmentation technique developed by McKay [13]. In essence, we identify a canonical  $\text{Aut}(\mathcal{X})$ -orbit of seeds contained by a visited one-factorization  $\mathcal{X}$ , and then check whether the seed  $(\Pi, T, \mathcal{S})$  from which  $\mathcal{X}$  was extended is in the canonical orbit. If yes, we accept  $\mathcal{X}$  if it is also the (lexicographic) minimum of its  $\text{Aut}(\Pi, T, \mathcal{S})$ -orbit. Otherwise we reject  $\mathcal{X}$ .

In the search we find 581,042,656,543 one-factorizations that extend the classified seeds; among these we find 10,305,262,573 nonisomorphic one-factorizations with a nontrivial automorphism group. Table 3 displays the possible orders  $i$  for the automorphism group and the associated number

$i$	$N_i$	$i$	$N_i$
1	1132835411296799774	21	1
2	10300646080	24	3
3	4497762	32	13
4	104560	39	3
5	2742	42	2
6	9247	48	1
8	1790	64	3
10	168	84	1
12	76	156	1
13	10	192	1
16	109	Total	1132835421602062347

Table 3: Nonisomorphic one-factorizations of  $K_{14}$

$N_i$  of nonisomorphic one-factorizations. ( $N_1$  will be determined in Section 3.4.) The search was distributed to a network of 180 Linux PCs using the batch system `autoson` [12], and required in total a little over 5 years of CPU time. The prevalent PC model in the network was Dell OptiPlex 745 with a 2.13-GHz Intel Core 2 Duo 6400 CPU and 2 GB main memory.

### 3.4 Applying the Orbit-Stabilizer Theorem

As a result of the computer searches, we know the value  $\text{LF}(K_{14})$  and the values  $i$  and  $N_i$  for each  $i \geq 2$ . We now apply (1) in the GDD representation over fixed but arbitrary sets  $U$  and  $V$ . Accordingly,  $\Omega$  is the set of distinct one-factorizations over  $U$  and  $V$ , and  $\Gamma$  is the group of all permutations of  $U \cup V$  that fix  $U$  and  $V$  setwise. Clearly,  $|\Gamma| = 13! \cdot 14!$ . Because there are  $13!$  ways to label the one-factors in each distinct one-factorization of  $K_{14}$  in the standard (graphical) representation, we have  $|\Omega| = 13! \cdot \text{LF}(K_{14})$ . Solving (1) for  $N_1$  and computing  $\sum_{i \geq 1} N_i$ , we have that the complete graph  $K_{14}$  has exactly

$$\text{NF}(K_{14}) = 1,132,835,421,602,062,347$$

nonisomorphic one-factorizations.

### 3.5 Correctness checks

We carry out two checks to gain confidence in the correctness of the values  $N_i$  for  $i \geq 2$  in Table 3.

First, both authors independently classified the seeds up to isomorphism, with identical results. One classification was conducted using the tools in [7] and the other using a basic backtrack search with isomorph rejection based on recorded canonical forms.

Second, for each automorphism type  $(p, f_U, f_V)$ , we count in two different ways the distinct tuples  $(\mathcal{X}, \Pi, T, \mathcal{S})$ , where  $\mathcal{X}$  is a one-factorization and  $(\Pi, T, \mathcal{S})$  is a seed of type  $(p, f_U, f_V)$  contained in  $\mathcal{X}$ .

The double count is implemented as follows. Fix an automorphism type  $(p, f_U, f_V)$ . For a one-factorization  $\mathcal{X}$ , denote by  $\text{seeds}(\mathcal{X})$  the number of distinct seeds of the fixed type that  $\mathcal{X}$  contains. For a seed  $(\Pi, T, \mathcal{S})$  of the fixed type, denote by  $\text{ext}(\Pi, T, \mathcal{S})$  the number of distinct one-factorizations that extend  $(\Pi, T, \mathcal{S})$ . Taking the sum over all isomorphism classes of one-factorizations on the left-hand side, and over all isomorphism classes of seeds of the fixed type on the right-hand side, we have, by the Orbit-Stabilizer Theorem,

$$\sum_{[\mathcal{X}]} \frac{|\Gamma|}{|\text{Aut}(\mathcal{X})|} \cdot \text{seeds}(\mathcal{X}) = \sum_{[(\Pi, T, \mathcal{S})]} \frac{|\Gamma|}{|\text{Aut}(\Pi, T, \mathcal{S})|} \cdot \text{ext}(\Pi, T, \mathcal{S}). \quad (8)$$

To evaluate the right-hand side of (8), we record the number of extensions  $\text{ext}(\Pi, T, \mathcal{S})$  and  $|\text{Aut}(\Pi, T, \mathcal{S})|$  for each classified seed  $(\Pi, T, \mathcal{S})$ . In particular,  $\text{ext}(\Pi, T, \mathcal{S})$  is simply the number of solutions found in the search for exact covers. The right-hand sides obtained for each automorphism type are listed in column Count in Table 4.

The left-hand side of (8) is accumulated for each classified one-factorization,  $\mathcal{X}$ . For each such  $\mathcal{X}$ , we find all distinct prime-order subgroups  $\Pi \leq \text{Aut}(\mathcal{X})$ . For each such  $\Pi$ , we find its type  $(p, f_U, f_V)$ , and accumulate the left-hand side of (8) for this type by  $|\Gamma|/|\text{Aut}(\mathcal{X})| \cdot m(p, f_U, f_V)$ , where  $m(p, f_U, f_V)$  is the number of distinct seeds of the form  $(\Pi, T, \mathcal{S})$  contained in  $\mathcal{X}$ . Because  $\mathcal{S}$  is uniquely determined by  $\mathcal{X}$ ,  $\Pi$ , and  $T$ , we can determine  $m(p, f_U, f_V)$  by combinatorial arguments based on  $(p, f_U, f_V)$  and Table 2. For example, consider  $p = 5$ ,  $f_U = 3$ ,  $f_V = 4$ . From Table 2 we find that  $T$  consists of one point of  $U$  fixed by  $\Pi$  and two points of  $V$  fixed by  $\Pi$ . By (g') and  $\mathcal{S} \subseteq \mathcal{X}$ , we have that  $T$  is a block of  $\mathcal{X}$ . There are  $\binom{4}{2} = 6$  possibilities to select a

$p$	$f_U$	$f_V$	$m(p, f_U, f_V)$	Count
2	1	2	24	598566905953570569439936512000
2	3	0	42	10362562621908790673701601280000
2	3	4	40	11064634216108608459689164800000
2	5	0	70	109764651070947200382428774400000
2	5	6	48	314439007330643189170176000000
2	7	0	98	65958792604977492862823301120000
3	1	2	1	814728009186504568995840000
5	3	4	6	1840950333789938122752000
7	6	0	49	2850020421206999040000
13	0	1	13	6016709778103664640000

Table 4: Double counting check

pair of points in  $V$  fixed by  $\Pi$ , each of which occurs in a unique block of  $\mathcal{X}$ . Thus,  $m(5, 3, 4) = 6$ . The other values  $m(p, f_U, f_V)$  are displayed in Table 4.

For each type  $(p, f_U, f_V)$ , we find that the computed left-hand and right-hand sides of (8) agree. This, together with the observation that the left-hand side of (8) depends on each of the computed values  $i$  and  $N_i$  for  $i \geq 2$ , gives us confidence that Table 3 is correct.

## Acknowledgments

This research was supported in part by the Academy of Finland, Grants 107493, 110196, and 117499.

## References

- [1] L. D. Andersen, “Factorizations of graphs,” Handbook of combinatorial designs, C. J. Colbourn and J. H. Dinitz (Editors), 2nd ed., Chapman & Hall/CRC, Boca Raton, 2007, pp. 740–755.
- [2] L. E. Dickson and F. H. Safford, Solution to problem 8 (group theory), Amer Math Monthly 13 (1906), 150–151.

- [3] 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.
- [4] 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.
- [5] GNU Multiple Precision Arithmetic Library, Version 4.2.1. Available at <http://gmplib.org/>.
- [6] E. Ihrig and E. Petrie, Automorphism groups of 1-factorizations of  $K_{12}$ , *Congr Numer* 109 (1995), 179–192.
- [7] P. Kaski, Isomorph-free exhaustive generation of designs with prescribed groups of automorphisms, *SIAM J Discrete Math* 19 (2005), 664–690.
- [8] P. Kaski and P. R. J. Östergård, Classification algorithms for codes and designs, Springer, Berlin, 2006.
- [9] P. Kaski and O. Pottonen, **libexact** User’s Guide, Version 1.0, in preparation.
- [10] D. E. Knuth, “Dancing links,” in Millennial perspectives in computer science, J. Davies, B. Roscoe, and J. Woodcock (Editors), Palgrave Macmillan, Basingstoke, 2000, pp. 187–214.
- [11] B. D. McKay, *nauty* user’s guide (version 1.5), Technical Report TR-CS-90-02, Computer Science Department, Australian National University, Canberra, 1990.
- [12] 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.
- [13] B. D. McKay, Isomorph-free exhaustive generation, *J Algorithms* 26 (1998), 306–324.
- [14] B. D. McKay, A. Meynert, and W. Myrvold, Small Latin squares, quasi-groups, and loops, *J Combin Des* 15 (2007), 98–119.

- [15] B. D. McKay and I. M. Wanless, On the number of Latin squares, *Ann Comb* 9 (2005) 335–344.
- [16] M. Meringer, Fast generation of regular graphs and construction of cages, *J. Graph Theory* 30 (1999), 137–146.
- [17] 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.
- [18] W. D. Wallis, “One-factorizations of complete graphs,” in *Contemporary design theory: A collection of surveys*, J. H. Dinitz and D. R. Stinson (Editors), Wiley, New York, 1992, pp. 593–631.
- [19] W. D. Wallis, *One-factorizations*, Kluwer, Dordrecht, 1997.
- [20] D. B. West, *Introduction to graph theory*, 2nd ed., Prentice–Hall, Upper Saddle River, 2001.