Isomorphism and Invariants of Parallelisms of Projective Spaces

Svetlana Topalova and Stela Zhelezova ^(✉)

Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences, Sofia, Bulgaria
{svetlana,stela}@math.bas.bg

Abstract. We consider the computer-aided constructive classification of parallelisms with predefined automorphism groups in small finite projective spaces. The usage of a backtrack search algorithm makes it very important to filter away equivalent partial solutions as soon as possible and to use a fast method for checking for isomorphism of any two parallelisms. The rejection of most of the equivalent solutions can be done by a test which uses the normalizer of the predefined automorphism group. We consider the applicability and effectiveness of such a test, and present sensitive invariants of resolutions of Steiner 2-designs. They can be used to facilitate any type of test for isomorphism of parallelisms.

Keywords: Resolutions of combinatorial designs · Parallelisms of projective spaces · Classification · Isomorphism · Invariants

1 Introduction

1.1 The Isomorphism Problem and Invariants

The approaches to solving the isomorphism problem are of major importance for the success of computer-aided classifications up to isomorphism of various combinatorial structures, such as graphs, designs, design resolutions, codes, Hadamard matrices, etc. Since the number of isomorphic solutions might be extremely big, classification is usually impossible without an efficient method for their rejection. Many authors consider this problem (recently [1, 8, 9, 17, 19, 21, 24]) and software solving it is available, for instance [4, 23, 25, 32]. Regardless of the difference in methods, they all make use of suitable invariants, namely functions which yield the same value for all members of an isomorphism class. An invariant is *complete* if its value is provably different for members of different isomorphism classes. The invariants that are usually applied are not complete. Structures with different

The research of the first author is partially supported by the National Scientific Program “Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)”, financed by the Ministry of Education and Science, and of the second author by the Bulgarian National Science Fund under Contract No KP-06-N32/2-2019.

invariants, however, can obviously not be isomorphic to each other, and this is very useful. The *sensitivity* of an invariant is measured by the ratio of the number of classes it distinguishes to the number of non-isomorphic objects under consideration [10]. A complete invariant has sensitivity 1. In a less formal manner, an invariant with relatively high sensitivity is called *sensitive*. Very often, however, sensitive invariants need considerable time to be calculated, and a considerable amount of memory to be stored.

The present paper describes invariants of design resolutions and parallelisms of projective spaces. These invariants are quite simple. They can be calculated and compared relatively fast and do not need much memory to be stored. We do not know previous papers presenting them. We show their effectiveness on some of our recent classifications of parallelisms with a predefined automorphism group. The invariants are used after the rejection of most of the isomorphic solutions by a normalizer-based minimality test (NM test). We present the main principles of such a test and point out the cases in which it may not establish that two parallelisms are isomorphic, and therefore the calculation of good invariants might be very helpful.

1.2 Design Resolutions and Parallelisms of Projective Spaces

The basic concepts and notations concerning designs and resolutions, and spreads and parallelisms in projective spaces, can be found, for instance, in [14, 16, 38].

A *t-spread* in the projective space $PG(n, q)$ is a set of distinct t -dimensional subspaces which partition the point set. A *t-parallelism* is a partition of the set of t -dimensional subspaces by t -spreads. Usually 1-spreads and 1-parallelisms are called line spreads (parallelisms) or just spreads (parallelisms). There can be line spreads and parallelisms if n is odd. Two parallelisms are *isomorphic* if there exists an automorphism of the projective space which maps each spread of the first parallelism to a spread of the second one.

Let v , k , and λ be positive integers, $1 < k < v/2$. Let $V = \{P_i\}_{i=1}^v$ be a finite set of *points*, and $\mathcal{B} = \{B_j\}_{j=1}^b$ a finite collection of k -element subsets of V , called *blocks*. $D = (V, \mathcal{B})$ is a *2-design* with parameters $2-(v, k, \lambda)$ if any 2-subset of V is contained in exactly λ blocks of $\mathcal{B}$. If $\lambda = 1$ the design is called a *Steiner 2-design*. A *parallel class* is a partition of the point set by blocks. A *resolution* of the design is a partition of the collection of blocks by parallel classes. Two resolutions are *isomorphic* if there exists an automorphism of the design which maps each parallel class of the first resolution to a parallel class of the second one.

Proposition 1. [37, 2.35-2.36] *The incidence of the points and t -dimensional subspaces of $PG(n, q)$ defines a 2-design. There is a one-to-one correspondence between the t -parallelisms of $PG(n, q)$ and the resolutions of this design.*

Example 1. A parallelism of $PG(3, 2)$. The projective space $PG(3, 2)$ has 15 points and 35 lines with 3 points each. A parallelism has 7 spreads consisting

of 5 disjoint lines. The point-line incidence defines a 2-(15, 3, 1) design (Fig. 1) whose points and blocks correspond respectively to the points and lines of the projective space. The parallelisms of $PG(3, 2)$ correspond to resolutions of this design (Fig. 2).

A \ B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35		
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0		
5	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0		
6	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0		
7	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1		
8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0		
9	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	
10	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	
11	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	
12	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	
13	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	
15	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

Fig. 1. $PG(3, 2)$ - the point-line incidence defines a 2-(15, 3, 1) design

C_1						C_2		C_4		C_6	
P \ B	1	20	26	31	33	bl	points	bl	points	bl	points
1	1	0	0	0	0						
2	1	0	0	0	0						
3	1	0	0	0	0						
4	0	1	0	0	0						
5	0	0	1	0	0						
6	0	0	0	1	0						
7	0	0	0	0	1						
8	0	1	0	0	0						
9	0	0	0	0	1						
10	0	0	1	0	0						
11	0	0	0	1	0						
12	0	1	0	0	0						
13	0	0	0	1	0						
14	0	0	0	0	1						
15	0	0	1	0	0						

C_1	
bl	points
B_1	1,2,3
B_{20}	4,8,12
B_{26}	5,10,15
B_{31}	6,11,13
B_{33}	7,9,14

C_2	
bl	points
B_2	1,4,5
B_{10}	2,8,10
B_{19}	3,13,14
B_{29}	6,9,15
B_{35}	7,11,12

C_4	
bl	points
B_4	1,8,9
B_{12}	2,12,14
B_{15}	3,5,6
B_{23}	4,11,15
B_{34}	7,10,13

C_6	
bl	points
B_6	1,12,13
B_8	2,4,6
B_{17}	3,9,10
B_{27}	5,11,14
B_{32}	7,8,15

C_3	
bl	points
B_3	1,6,7
B_{11}	2,9,11
B_{18}	3,12,15
B_{22}	4,10,14
B_{24}	5,8,13

C_5	
bl	points
B_5	1,10,11
B_{13}	2,13,15
B_{14}	3,4,7
B_{25}	5,9,12
B_{28}	6,8,14

C_7	
bl	points
B_7	1,14,15
B_9	2,5,7
B_{16}	3,8,11
B_{21}	4,9,13
B_{30}	6,10,12

Fig. 2. A parallelism of $PG(3, 2)$ - a resolution of the 2-(15, 3, 1) point-line design. The 7 spreads correspond to the 7 parallel classes $C_1, C_2, \dots, C_7$.

There are several theoretical constructions of infinite families of parallelisms that are presently known [2, 7, 11, 15, 28, 42]. A full classification is available in $PG(3, 2)$ and $PG(3, 3)$ [3], but it is currently out of reach in the other projective spaces. There are computer aided classifications of parallelisms with predefined automorphism groups [5, 6, 13, 29–31, 34, 36, 39–41].

1.3 The Present Paper

Section 2 briefly describes the specifics of the classification problem for parallelisms and the approach that we have used in several recent works. One of the major difficulties in all these cases is the final test for isomorphism of the obtained parallelisms. Since parallelisms can be considered as resolutions of the point-line design of $PG(n, q)$, Sect. 3 is devoted to invariants of design resolutions. The invariants that we offer, are fast to calculate, do not need too much memory to store, and partition the parallelisms to numerous invariant classes. Section 4 is a comment on their further usability for the classification of parallelisms with bigger parameters, and on their applicability to other problems.

2 Computer-Aided Classification of Parallelisms

The software for construction of parallelisms is based on the exhaustive backtrack search techniques. A lexicographic order can be defined both on the parallelisms, and on the partial solutions. This allows the rejection of partial solutions which are not minimal with respect to the lexicographic order. Such a technique for classification of various combinatorial structures is known as *orderly generation* [12], [21, chapter 4], [33]. One way to implement the method is by applying a **minimality test** to some of the partial solutions and to all full solutions.

The automorphism group $G \cong P\Gamma L(n+1, q)$ of the projective space $PG(n, q)$, however, is very rich, and this makes the minimality test too slow. That is why only a **normalizer-based minimality test (NM test)** is usually applied when parallelisms invariant under some predefined automorphism group G_c are classified. The NM test checks if the normalizer $N(G_c) = \{g \in G \mid gG_cg^{-1} = G_c\}$ contains an element which maps the constructed (partial) parallelism to a lexicographically smaller (partial) solution. If so, the current (partial) solution is discarded. We briefly explain below how the normalizer can help us remove isomorphic solutions.

Proposition 2. *Let G_c be a Sylow subgroup of the automorphism group of $PG(n, q)$, and let each of the parallelisms P and P' be invariant under G_c . Then to establish an isomorphism of P and P' it is enough to check if there is an element of the normalizer $N_G(G_c)$ of G_c in G which maps P to P' .*

Proof. The statement was first proved in [31] for parallelisms of $PG(3, 5)$ with automorphisms of order 31. We present here the main ideas for the general case. Denote by G_P the full automorphism group of P . We have to check if there is some $\varphi \in G$ such that $P' = \varphi P$. Let $\alpha \in G_c$. Then $\varphi P = \alpha\varphi P$ and thus $P = \varphi^{-1}\alpha\varphi P$. Namely $\varphi^{-1}\alpha\varphi$ is also an automorphism of P . That is why P is invariant both under G_c and under $\varphi^{-1}G_c\varphi$. If $\varphi^{-1}G_c\varphi = G_c$, then φ is in the normalizer $N_G(G_c)$ of G_c in G . If φ is not in $N_G(G_c)$, then $\varphi^{-1}G_c\varphi$ is a conjugate subgroup of G_c in G , and $|G_P| > |G_c|$. If G_c and $\varphi^{-1}G_c\varphi$ are conjugate in G_P too (for instance, this always holds if G_c is a Sylow subgroup of G), then since φ is not an automorphism of P , there must exist an automorphism $\beta \in G_P$, such

that $\varphi^{-1}G_c\varphi = \beta G_c\beta^{-1}$. Then $G_c = \varphi\beta G_c\beta^{-1}\varphi^{-1}$ and therefore $\varphi\beta \in N_G(G_c)$. Since $\varphi\beta P = \varphi P$, the statement follows.

If the predefined group is not a Sylow subgroup of G , the normalizer-based minimality test may not succeed in removing all isomorphic parallelisms. That is why a further test for isomorphism must be applied.

Our experience shows that a very small number of isomorphic parallelisms remain if an NM test has been applied to them. In [40] and [41] we classify parallelisms with predefined groups which are of prime order, but not Sylow, and observe that the NM test removes all isomorphic solutions. Constructing parallelisms of $PG(3, 4)$ invariant under cyclic groups of order 4 [6] we obtain 253344 parallelisms after the NM test, and 252738 after a full test for isomorphism, i.e. the latter removed 0.23% of the solutions obtained after the NM test.

If the number of parallelisms is big, we can first determine the order of their full automorphism groups. Whatever algorithm or software we use for that purpose, invariants of the points, lines and spreads of the parallelism might be very helpful, because only points (lines, spreads) with the same invariants can be mapped to one another. When we know the full automorphism groups, we can look for isomorphisms only among parallelisms with $|G_P| > |G_c|$. In the cases that we have considered, the percentage of these parallelisms is very small. Their number, however, might be quite big, and therefore it might be very slow to test for isomorphisms any two of them. If we can easily calculate sensitive invariants of the parallelisms, we can only check for isomorphisms among parallelisms with the same invariants.

Parallelisms can be considered as resolutions of the point-line design of the projective space. The next section presents the invariants we use, as invariants of resolutions of Steiner 2-designs, because we believe that they can also have various applications outside the parallelisms classification problem.

3 Invariants of Resolutions and Parallelisms

Resolutions of designs with small parameters have been classified in many papers (for instance, [20, 22, 26, 27]). The invariants that are usually used, are the order of the automorphism group and some properties of the underlying design.

The resolution isomorphism problem can be transformed to graph isomorphism problems (for instance, Betten's approach in [5]). This makes it possible to use Nauty [25] or some other graph isomorphism software [4, 18, 23, 32], and to use graph invariants to distinguish resolutions in a way similar to that for designs (for example, [22, 24]). These invariants, however, do not take in consideration the fact that we deal with resolutions and are therefore quite complex.

Invariants of resolutions (not of their underlying design, or related graph) are used in the works of Morales and Velarde [26, 27] and Kaski et al. [20] who construct matrices of the intersections between the parallel classes. The invariants and their usage are different from those that we describe. Invariants of resolutions of $2-(v, 3, 1)$ designs (Kirkman triple systems) are applied by Stinson and

Vanstone in [35]. They are defined by a function which maps 3-subsets of points to 3-subsets of parallel classes and are different from the invariants we present.

Our first aim is to describe the relation of each block to each resolution class. Consider a resolution of a $2-(v, k, \lambda)$ design with b blocks and r parallel classes. Denote by $B_1, B_2, \dots, B_b$ the blocks of the design, and by $C_1, C_2, \dots, C_r$ the parallel classes of the resolution. Denote by N_j^d the number of blocks in the parallel class of block B_j , which are different from B_j and disjoint with all the blocks of class C_d that contain some of the points of B_j . The number N_j^d remains unchanged by a permutation of the points of the design, and a permutation of the blocks which maps parallel classes to parallel classes defines a permutation on the numbers N_j^d , where $d = 1, 2, \dots, r$ and $j = 1, 2, \dots, b$.

For each block B_j define a vector $\Omega_j = (b_0^j, b_1^j, \dots, b_{v/k-1}^j)$, where b_m^j is the number of parallel classes C_d for which $N_j^d = m$. A resolution isomorphism maps parallel classes to parallel classes. That is why the vector Ω_j will be the same for the image of B_j under a resolution isomorphism. So we will call these vectors resolution block invariant vectors, or just block invariants. To compare block invariants we define a lexicographic order such that $\Omega_{j'} < \Omega_{j''}$ if $b_k^{j'} < b_k^{j''}$ and $b_i^{j'} = b_i^{j''}$ for $i < k$. Let n_b be the number of the different block invariants and let us denote them by $\beta_1, \beta_2, \dots, \beta_{n_b}$, and the set they form, by β . The block invariants are relatively easy to calculate, because they are based on the relation of each block to the r parallel classes, while the most used block invariants of designs [22] present the relation of each block to all the $(b-1)(b-2)/2$ pairs of different other blocks. For the design considered in Example 2, for instance, $r = 21$ and $b = 357$.

Each point P_i is in r blocks. Their block invariants become the elements of a resolution point invariant vector $\Pi_i = (\beta_{i_1}, \beta_{i_2}, \dots, \beta_{i_r})$, such that $\beta_{i_j} \leq \beta_{i_{j+1}}$. We next find the number n_p of the different point invariants and denote them by $\pi_1, \pi_2, \dots, \pi_{n_p}$, and the set of these invariants by π . Each parallel class C_l contains v/k blocks. Their block invariants make up a resolution class invariant vector $\Gamma_l = (\beta_{l_1}, \beta_{l_2}, \dots, \beta_{l_{v/k}})$, such that $\beta_{l_j} \leq \beta_{l_{j+1}}$. We denote the different class invariants by $\gamma_1, \gamma_2, \dots, \gamma_{n_c}$ and the set they comprise by γ .

Example 2. Parallelisms of $PG(3, 4)$. They can be considered as resolutions of the $2-(85, 5, 1)$ point-line design. There are 21 parallel classes with 17 blocks each. Table 1 presents three of the parallel classes of a resolution with automorphism group order 960. The blocks are given by their points.

In this example $N_2^2 = 12$, because each of the blocks $B_6, B_7, \dots, B_{17}$ of parallel class C_1 is disjoint with each of the blocks $B_{18}, B_{19}, \dots, B_{22}$ (these blocks of C_2 contain points of B_2). In the same manner $N_2^3 = 6$ because blocks $B_1, B_3, B_6, B_7, B_8, B_9$ are pairwise disjoint with blocks $B_{40}, B_{44}, B_{46}, B_{49}, B_{50}$, and $N_2^1 = 16$ because $B_1, B_3, B_4, \dots, B_{17}$ are disjoint with B_2 . In the same way we find N_2^d for $d = 4, 5, \dots, 21$ (for the classes that are not given in Table 1). In total, the value is 6 for 15 classes, 12 for five, and 16 for C_1 , namely $\Omega_2 = (0, 0, 0, 0, 0, 0, 15, 0, 0, 0, 0, 0, 5, 0, 0, 0, 1)$. We proceed with finding Ω_j for each $j = 1, 3, 4, \dots, 357$, and establish that there are $n_b = 5$ different vectors

Table 1. Three parallel classes of a resolution of a 2-(85, 5, 1) design.

C_1			C_2			C_3		
Block	Points	Inv	Block	Points	Inv	Block	Points	Inv
B_1	1,2,3,4,5	1	B_{18}	1,6,7,8,9	3	B_{35}	1,10,11,12,13	3
B_2	6,22,38,54,70	2	B_{19}	2,22,26,30,34	4	B_{36}	2,24,28,32,36	4
B_3	7,26,43,60,77	2	B_{20}	3,38,43,48,53	4	B_{37}	3,40,45,46,51	4
B_4	8,30,48,65,79	2	B_{21}	4,54,60,65,67	4	B_{38}	4,57,59,62,68	4
B_5	9,34,53,67,84	2	B_{22}	5,70,77,79,84	4	B_{39}	5,73,74,80,83	4
B_6	10,28,40,68,80	2	B_{23}	10,29,41,69,81	5	B_{40}	6,23,39,55,71	5
B_7	11,24,45,62,83	2	B_{24}	11,25,44,63,82	5	B_{41}	7,27,42,61,76	5
B_8	12,36,46,59,73	2	B_{25}	12,37,47,58,72	5	B_{42}	8,31,49,64,78	5
B_9	13,32,51,57,74	2	B_{26}	13,33,50,56,75	5	B_{43}	9,35,52,66,85	5
B_{10}	14,33,41,61,85	2	B_{27}	14,31,39,59,83	5	B_{44}	14,30,38,58,82	5
B_{11}	15,37,44,55,78	2	B_{28}	15,35,42,57,80	5	B_{45}	15,34,43,56,81	5
B_{12}	16,25,47,66,76	2	B_{29}	16,23,49,68,74	5	B_{46}	16,22,48,69,75	5
B_{13}	17,29,50,64,71	2	B_{30}	17,27,52,62,73	5	B_{47}	17,26,53,63,72	5
B_{14}	18,35,39,63,75	2	B_{31}	18,36,40,64,76	5	B_{48}	18,37,41,65,77	5
B_{15}	19,31,42,69,72	2	B_{32}	19,32,45,66,71	5	B_{49}	19,33,44,67,70	5
B_{16}	20,27,49,56,82	2	B_{33}	20,28,46,55,85	5	B_{50}	20,29,47,54,84	5
B_{17}	21,23,52,58,81	2	B_{34}	21,24,51,61,78	5	B_{51}	21,25,50,60,79	5

Table 2. The set β of the different block invariants, $n_b = 5$

#inv	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}
β_1	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	1
β_2	0	0	0	0	0	0	15	0	0	0	0	0	5	0	0	0	1
β_3	0	0	0	0	19	0	0	0	0	0	0	0	1	0	0	0	1
β_4	0	0	8	3	4	4	0	0	0	0	0	0	1	0	0	0	1
β_5	4	0	2	4	9	0	1	0	0	0	0	0	0	0	0	0	1

$\beta_1, \beta_2, \dots, \beta_5$. They are presented in Table 2. The number m in column inv of Table 1 means that the block invariant is β_m .

We next calculate the point invariants and establish that $n_p = 2$, namely $\Pi_i = \pi_1 = (1, 3, \dots, 3, 4, 4, 4, 4)$ for $i \leq 5$ and $\Pi_i = \pi_2 = (2, 3, 3, 3, 3, 4, 5, \dots, 5)$ for $i > 5$. This means that the first five points, for instance, are in one block with invariant β_1 , 12 blocks with invariant β_3 , and 4 blocks with invariant β_4 . Finally we calculate the invariants of the classes and obtain that $\Gamma_1 = \gamma_1 = (1, 2, \dots, 2)$ for the first parallel class and $\Gamma_l = \gamma_2 = (3, 3, 3, 3, 4, 5, \dots, 5)$ for $l > 1$.

Interested readers can obtain the whole example using the invariant calculation C++ source available at <http://www.moi.math.bas.bg/moiuser/~stela> and the example files going with it.

The invariant sets $\beta = \{\beta_1, \beta_2, \dots, \beta_{n_b}\}$, $\pi = \{\pi_1, \pi_2, \dots, \pi_{n_p}\}$ and $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_{n_c}\}$ make up an invariant of the resolution. We calculated the resolution invariants for some of the known nonisomorphic parallelisms of $PG(3, 4)$ [5, 6, 39–41]. They partition the parallelisms to invariant classes that contain either one, or two parallelisms. The results are presented in Table 3, where $|G_P|$ is the order of the full automorphism group, I the number of invariant classes, N the number of isomorphism classes, and S the sensitivity of the invariant, $S = I/N$.

Table 3. PG(3,4)

$ G_P $	4	5	6	7	10	12	15	20	24	30	48	60	96	960	All
I	251836	31648	4488	482	72	40	26	52	14	20	12	4	2	3	288699
N	251836	31830	4488	482	76	52	40	52	14	38	12	8	2	4	288934
S	1	0.9943	1	1	0.9474	0.7692	0.65	1	1	0.5263	1	0.5	1	0.75	0.9992

The most complex part of the invariant calculation is the determination of the block invariants $\Omega_1, \Omega_2, \dots, \Omega_b$. It can be done, for instance, as shown in Example 3, where the main operations are repeated $\frac{b^3}{r}$ times, b is the number of blocks, and r the number of parallel classes. The complexity is $O(b^3)$, but the actual performance is faster because $\frac{b}{r} = \frac{v}{k}$ is much smaller than b .

Example 3. Calculation of $\Omega_1, \Omega_2, \dots, \Omega_b$. All arrays in this code segment are of integer type, except `covered` (boolean); `pclass[j]` is the parallel class of block j , `cpoints[j1][j2]` is the number of common points of blocks $j1$ and $j2$, `Omega[j]` is Ω_j , and `covered` and `inv` are auxiliary arrays.

```
for(j=1; j<=b; j++) // b iterations
{
    for(i=0; i<=r; i++) inv[i]=0;
    for(i=0; i<b/r; i++) Omega[j][i]=0;
    for(jj=1; jj<=b; jj++) if(jj!=j && pclass[jj]==pclass[j])//b/r iterations
    {
        for(i=1; i<=r; i++) covered[i] = false;
        for(j1=1; j1<=b; j1++) if (!covered[pclass[j1]]) //at most b iterations
            if(cpoints[j1][j]>0 && cpoints[j1][jj]>0) covered[pclass[j1]]=true;
        for(i=1; i<=r; i++) if(!covered[i]) inv[i]++;
    }
    for(i=1; i<=r; i++) Omega[j][inv[i]]++;
}
```


4 Comments

- Our experience with classification of parallelisms with predefined automorphism groups, shows that the normalizer-based minimality test is a powerful fast way of filtering away most of the isomorphic solutions.
- The invariants presented in Sect. 3 are very useful for the classification of the parallelisms we applied them to, because they partition them to numerous small invariant classes. We believe that they will be helpful to future classifications of parallelisms with bigger parameters too.
- We suppose that these invariants will work well on the resolutions of any $2-(v, k, 1)$ design (Steiner 2-design). For resolutions of designs with $\lambda \neq 1$, however, modifications of the block invariants might be more suitable, such that the exact number of common points of two blocks is encountered (not only if these blocks are disjoint or not).

References

1. Al-ogaidi, A., Betten, A.: Large Arcs in Small Planes, accepted for publication in *Congressus Numerantium* (2020)
2. Baker, R.D.: Partitioning the planes of $AG_{2m}(2)$ into 2-designs. *Discrete Math.* **15**, 205–211 (1976)
3. Betten, A.: The packings of $PG(3, 3)$. *Des. Codes Cryptogr.* **79**(3), 583–595 (2016)
4. Betten, A.: Orbiter - A program to classify discrete objects, 2016–2018. <https://github.com/abetten/orbiter>. Accessed 24 Jan 2019
5. Betten, A., Topalova, S., Zhelezova, S.: Parallelisms of $PG(3, 4)$ invariant under a Baer involution. In: *Proceedings of the 16th International Workshop on Algebraic and Combinatorial Coding Theory*, Svetlogorsk, Russia, pp. 57–61 (2018). <http://acct2018.skoltech.ru/>
6. Betten, A., Topalova, S., Zhelezova, S.: Parallelisms of $PG(3, 4)$ invariant under cyclic groups of order 4. In: Ćirić, M., Droste, M., Pin, J.É. (eds.) *CAI 2019. LNCS*, vol. 11545, pp. 88–99. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21363-3_8
7. Beutelspacher, A.: On parallelisms in finite projective spaces. *Geom. Dedicata.* **3**(1), 35–40 (1974)
8. Bouyukliev, I.: About the code equivalence. In: *Advances in Coding Theory and Cryptography. Series on Coding Theory and Cryptology*, pp. 126–151 (2007)
9. Bouyukliev, I., Dzhumalieva-Stoeva, M.: Representing equivalence problems for combinatorial objects. *Serdica J. Comput.* **8**(4), 327–354 (2014)
10. Colbourn, M.: Algorithmic aspects of combinatorial designs: a survey. *Ann. Discret. Math.* **26**, 67–136 (1985)
11. Denniston, R.H.F.: Some packings of projective spaces. *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur.* **52**(8), 36–40 (1972)
12. Faradžev, I.A.: Constructive enumeration of combinatorial objects. In: *Problèmes Combinatoires et Théorie des Graphes*, pp. 131–135. CNRS, Paris, (Université d’Orsay, 9–13 July 1977) (1978)
13. Fuji-Hara, R.: Mutually 2-orthogonal resolutions of finite projective space. *Ars Combin.* **21**, 163–166 (1986)

14. Colbourn, C., Dinitz, J. (eds.) Handbook of Combinatorial Designs. Discrete mathematics and its applications, 2nd edn. CRC Press, Boca Raton (2007)
15. Johnson, N.L.: Some new classes of finite parallelisms. *Note Mat.* **20**(2), 77–88 (2000)
16. Johnson, N.L.: Combinatorics of Spreads and Parallelisms. Chapman & Hall Pure and Applied Mathematics. CRC Press, Boca Raton (2010)
17. Jungnickel, D., Tonchev, V.D.: New invariants for incidence structures. *Des. Codes Cryptogr.* **68**, 163–177 (2013)
18. Junttila, T., Kaski, P.: Bliss: a tool for computing automorphism groups and canonical labelings of graphs. <http://www.tcs.hut.fi/Software/bliss/>. Accessed 24 Jan 2019
19. Junttila, T., Kaski, P.: Engineering an efficient canonical labeling tool for large and sparse graphs. In: Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX07), pp. 135–149. SIAM (2007)
20. Kaski, P., Morales, L.B., Östergård, P., Rosenblueth, D.A., Velarde, C.: Classification of resolvable 2-(14,7,12) and 3-(14,7,5) designs. *J. Comb. Math. Comb. Comput.* **47**, 65–74 (2003)
21. Kaski, P., Östergård, P.: Classification Algorithms for Codes and Designs. Springer, Berlin (2006)
22. Kramer, E.S., Magliveras, S.S., Mathon, R.: The Steiner systems $S(2, 4, 25)$ with nontrivial automorphism group. *Discret. Math.* **77**(1–3), 137–157 (1989)
23. Magma. The Computational Algebra Group within the School of Mathematics and Statistics of the University of Sydney (2004). <https://www.magma-software.com/en/company/about-magma/>
24. McKay, B.D., Piperno, A.: Practical graph isomorphism, II. *J. Symb. Comput.* **60**, 94–112 (2014)
25. McKay, B.: Nauty User’s Guide (Version 2.6). Australian National University (2016)
26. Morales, L.B., Velarde, C.: A complete classification of $(12, 4, 3)$ -RBIBDs. *J. Comb. Des.* **9**(6), 385–400 (2001)
27. Morales, L.B., Velarde, C.: Enumeration of resolvable 2-(10, 5, 16) and 3-(10, 5, 6) designs. *J. Comb. Des.* **13**(2), 108–119 (2005)
28. Penttilä, T., Williams, B.: Regular packings of $PG(3, q)$. *Eur. J. Combin.* **19**(6), 713–720 (1998)
29. Prince, A.R.: Parallelisms of $PG(3, 3)$ invariant under a collineation of order 5. In: Johnson, N.L. (ed.) Mostly Finite Geometries. Lecture Notes in Pure and Applied Mathematics, vol. 190, pp. 383–390. Marcel Dekker, New York (1997)
30. Prince, A.R.: Uniform parallelisms of $PG(3, 3)$. In: Hirschfeld, J., Magliveras, S., Resmini, M. (eds.) Geometry, Combinatorial Designs and Related Structures, London Mathematical Society Lecture Note Series, vol. 245, pp. 193–200. Cambridge University Press, Cambridge (1997)
31. Prince, A.R.: The cyclic parallelisms of $PG(3, 5)$. *Eur. J. Combin.* **19**(5), 613–616 (1998)
32. Q-extension, a software for classification of linear codes over small finite fields. <http://www.moi.math.bas.bg/moiuser/~data/>. Accessed 24 Jan 2020
33. Read, R.C.: Every one a winner; or, How to avoid isomorphism search when cataloguing combinatorial configurations. *Ann. Discret. Math.* **2**, 107–120 (1978)
34. Sarmiento, J.: Resolutions of $PG(5, 2)$ with point-cyclic automorphism group. *J. Combin. Des.* **8**(1), 2–14 (2000)
35. Stinson, D.R., Vanstone, S.A.: Some non-isomorphic Kirkman triple systems of orders 39 and 51. *Utilitas Math.* **27**, 199–205 (1985)

36. Stinson, D.R., Vanstone, S.A.: Orthogonal packings in $PG(5, 2)$. *Aequationes Math.* **31**(1), 159–168 (1986)
37. Storme, L.: Finite geometry. In: Colbourn, C., Dinitz, J. (eds.) *Handbook of Combinatorial Designs*. 2nd edn. Rosen, K. (eds.) *Discrete mathematics and its applications*, pp. 702–729. CRC Press, Boca Raton (2007)
38. Tonchev, V.D.: *Combinatorial Configurations*. Longman Scientific and Technical, New York (1988)
39. Topalova, S., Zhelezova, S.: On transitive parallelisms of $PG(3, 4)$. *Appl. Algebra Engrg. Comm. Comput.* **24**(3–4), 159–164 (2013)
40. Topalova, S., Zhelezova, S.: On point-transitive and transitive deficiency one parallelisms of $PG(3, 4)$. *Des. Codes Cryptogr.* **75**(1), 9–19 (2015)
41. Topalova, S., Zhelezova, S.: New parallelisms of $PG(3, 4)$. *Electron. Notes Discret. Math.* **57**, 193–198 (2017)
42. Zaicev, G., Zinoviev, V., Semakov, N.: Interrelation of Preparata and Hamming codes and extension of Hamming codes to new double-error-correcting codes. In: *Proceedings of the Second International Symposium on Information Theory*, (Armenia, USSR, 1971), Budapest, *Academiai Kiado*, pp. 257–263 (1973)