Des. Codes Cryptogr. (2013) 69:95–106
DOI 10.1007/s10623-012-9623-4

Permutation codes with specified packing radius

Derek H. Smith · Roberto Montemanni

Received: 11 October 2011 / Revised: 19 January 2012 / Accepted: 25 January 2012 /
Published online: 18 February 2012

Abstract Most papers on permutation codes have concentrated on the minimum Hamming distance of the code. An (n, d) permutation code (or *permutation array*) is simply a set of permutations on n elements in which the Hamming distance between any pair of distinct permutations (or *codewords*) is at least d. An $(n, 2e + 1)$ or $(n, 2e + 2)$ permutation code is able to correct up to e errors. These codes have a potential application to powerline communications. It is known that in an $(n, 2e)$ permutation code the balls of radius e surrounding the codewords may all be pairwise disjoint, but usually some overlap. Thus an $(n, 2e)$ permutation code is generally unable to correct e errors using nearest neighbour decoding. On the other hand, if the packing radius of the code is defined as the largest value of e for which the balls of radius e are all pairwise disjoint, a permutation code with packing radius e can be denoted by $[n, e]$. Such a permutation code can always correct e errors. In this paper it is shown that, in almost all cases considered, the number of codewords in the best $[n, e]$ code found is substantially greater than the largest number of codewords in the best known $(n, 2e + 1)$ code. Thus the packing radius more accurately specifies the requirement for an e-error-correcting permutation code than does the minimum Hamming distance. The techniques used include construction by automorphism group and several variations of clique search They are enhanced by two theoretical results which make the computations feasible.

Keywords Permutation codes · Packing radius · Automorphism groups · Clique search

Mathematics Subject Classification (2000) 05A05 · 05E20 · 94B60

Communicated by R. Hill.

D. H. Smith
Division of Mathematics and Statistics, University of Glamorgan, Pontypridd CF37 1DL, Wales, UK
e-mail: dhsmith@glam.ac.uk

R. Montemanni (✉)
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Galleria 2, 6928 Manno, Switzerland
e-mail: roberto@idsia.ch

1 Introduction

Permutation codes have received considerable attention in recent years [1–3,9,10,12–14], [23–25], both for their intrinsic interest and because of potential applications to powerline communications described in [9,15,16,22]. A permutation code is simply a set of permutations in the symmetric group $\mathscr{S}_n$ of all permutations on n elements. The codewords are the permutations and the code length is n. For two permutations $\pi, \pi' \in \mathscr{S}_n$ the Hamming distance is given by:

$$d_H(\pi, \pi') := |\{x \in \{1, 2, \ldots, n\} | \pi(x) \neq \pi'(x)\}|$$

Note that the number of fixed points of $\pi'\pi^{-1}$ is $n - d_H(\pi, \pi')$. The minimum Hamming distance d of the code is then the minimum $d_H(\pi, \pi')$ taken over all pairs π, π' of distinct permutations. A code of length n with minimum distance d is denoted an (n, d) permutation code, often referred to as an (n, d) permutation array. As 1 cannot appear as a distance between two permutations of $\mathscr{S}_n$, the set of all distances that can actually appear for $\mathscr{S}_n$ is $D(\mathscr{S}_n) = \{0, 2, 3, 4, \ldots, n\}$.

Following the terminology in [23], a *ball of radius e* surrounding a *centre* $\pi \in \mathscr{S}_n$ is denoted by:

$$B_e(\pi) := \left\{\pi' \in \mathscr{S}_n | d_H(\pi, \pi') \leq e\right\}.$$

The number of permutations in a ball of radius e is independent of the choice of π and is denoted by V_e where

$$V_e = |B_e(\pi)| = \sum_{k=0}^{e} \binom{n}{k} k! \sum_{x=0}^{k} \frac{(-1)^x}{x!}$$

(using the formula $d_k = k! \sum_{x=0}^{k} \frac{(-1)^x}{x!}$ for the number of derangements of k elements).

Given a code C (a subset of the elements of $\mathscr{S}_n$), the packing radius of C is denoted by $p(C)$, where

$$p(C) := \max\{e' \in D(\mathscr{S}_n) | B_{e'}(\pi) \cap B_{e'}(\pi') = \phi, \forall \pi, \pi' \in C \text{ with } \pi \neq \pi'\}.$$

The packing radius is bounded above by the *covering radius* [6], which has received rather more attention in the literature. A permutation code of length n with packing radius e will be denoted an $[n, e]$ code. Here it will be assumed that if r errors affect a permutation π then the result is a permutation at distance r from π. Then the packing radius more accurately describes the error-correcting capability of the code than the minimum Hamming distance. An $(n, 2e+1)$ code or an $(n, 2e+2)$ code will always correct e errors using nearest neighbour decoding (see Theorem 1). However, an $(n, 2e)$ code is not guaranteed to correct e errors, as a received permutation with e errors may be equidistant from two or more distinct codewords. On the other hand, specifying that the packing radius is e does guarantee that e errors can always be corrected by nearest neighbour decoding. A received permutation with at most e errors will be in a unique ball surrounding a codeword, and can be correctly decoded to the centre of the ball. As illustrated by Quistorff [23] (see also Example 1), an $(n, 2e)$ code may have packing radius e or $e - 1$. This contrasts with the situation in classical coding theory where the codewords are vectors and a code of minimum distance $2e$ cannot have packing radius e.

Example 1 The codes C_1 and C_2 (written here in cycle notation with the identity permutation denoted id) both have $d = 6$.

$$C_1 = \{id, (123456)\} \subseteq \mathscr{S}_6$$
$$C_2 = \{id, (123)(456)\} \subseteq \mathscr{S}_6$$

For the code C_1 the balls of radius 3 surrounding the two codewords are disjoint. For the code C_2 the permutation (456) is at distance 3 from each codeword, so the balls of radius 3 surrounding the two codewords are not disjoint.

The maximum number of codewords in an (n, d) code is denoted by $M(n, d)$. Tables of best known lower and upper bounds for $M(n, d)$ have been presented in [9,24]. Here the maximum number of codewords in an $[n, e]$ code is denoted by $P[n, e]$. It can be directly checked that most $(n, 2e)$ codes previously constructed are not $[n, e]$ codes. It will be shown that there do generally exist $[n, e]$ codes (usually with minimum Hamming distance $2e$) which have more codewords than the best known $(n, 2e + 1)$ code. A table of lower and upper bounds for $P[n, e]$ is constructed.

Theoretical results which allow the test for overlapping balls to be carried out efficiently are presented in Sect. 2. The table of bounds for $P[n, e]$ is given in Sect. 3. The automorphism group and maximum clique method used is described in detail in Sect. 4, together with a detailed description of individual constructions. Iterative clique building methods used in many of the other cases and based on either codewords, cycles of length n or cycles of length $n-1$ are described in Sect. 5. The conclusion summarizes the success of the various methods.

2 Efficient computational tests for overlapping balls

The work presented in this paper follows closely some of the methods described in [9,24]. In contrast to the work presented in those two papers, where the computation of the Hamming distance is fast, a test for whether any balls of radius e overlap over all pairs of codewords as centres is potentially very time consuming. In this section two theorems are presented which allow this computation to be done efficiently.

Theorem 1 *Let C be a permutation code (a subset of $\mathscr{S}_n$) with packing radius $p(C) > 1$ and minimum distance $d(C)$. Then*

1. *if $d(C) \geq 2p(C) + 1$, all balls of radius $p(C)$ are disjoint.*
2. *if $d(C) \leq 2p(C) - 1$, at least one pair of balls of radius $p(C)$ are not disjoint.*

Proof The first statement is clear. If two balls are not disjoint then the distance between their centres is at most $2p(C)$. Only the second statement need be considered.

In the proof of the second statement, write $e = p(C)$ and let π_1, π_2 denote a pair of permutations such that $d_H(\pi_1, \pi_2) = 2e - r$ $(r \geq 1)$. By applying the permutation π_1^{-1} it can be seen that without loss of generality the two permutations can be assumed to be the identity id and $\pi = \pi_1^{-1}\pi_2$ with $d_H(id, \pi) = 2e - r$ $(r \geq 1)$. Then there is a set W of elements with $|W| = n - 2e + r$ such that $\pi(x) = x\ \forall x \in W$. Consider the cycle representation of π. Suppose that this representation contains a set of cycles disjoint from W denoted $\{(i_j, \pi(i_j), \pi^2(i_j), \ldots, \pi^{t_j-1}(i_j)) \mid \pi^{t_j}(i_j) = i_j,\ j \in U\}$ with cycle j having length t_j. Associated with each cycle will be open fragments such as
$frag = (i_\ell, \pi(i_\ell), \pi^2(i_\ell), \ldots, \pi^{s-1}(i_\ell))$ (with $\pi^s(i_\ell) \neq i_\ell$) of length s.

There are two cases to consider:

1. There is a subset U_1 of U such that $\sum_{j \in U_1} t_j = e - r + 1$. Let V_1 be the union of all elements of the cycles of U_1. Choose a permutation π' such that $\pi'(x) = \pi(x)$ for $x \in V_1$ and $\pi'(x) = x$ for $x \notin V_1$. Then π' is a derangement of the $e - r + 1$ elements of V_1 and fixes the remaining $e - 1$ elements for which id and π differ. Similarly, π' takes the values of π for the $e - r + 1$ elements of V_1 and the same value as id for the remaining $e - 1$ elements for which id and π differ. Thus $d_H(id, \pi') = e - r + 1$ and $d_H(\pi, \pi') = e - 1$. It follows that the balls surrounding id and π are not disjoint.
2. If the first case does not hold then there must be a subset U_2 of U and a fragment $frag$ with s elements such that $s + \sum_{j \in U_2} t_j = e - r + 1$. Let V_2 be the union of all but the last element of $frag$ and all elements in the cycles of U_2. Let y be the first element of $frag$ and z be the last element of $frag$. Choose a permutation π'' such that $\pi''(x) = \pi(x)$ for $x \in V_2$, $\pi''(z) = y$, and $\pi''(x) = x$ for $x \notin V_2 \cup \{z\}$. If $|frag| = 1$ then $y = z$ and π'' is a derangement of the $e - r$ elements of V_2, has the same value as id for y and the same value as id for the remaining $e - 1$ elements for which id and π differ. Thus $d_H(id, \pi'') = e - r$. If $|frag| > 1$ then $y \neq z$ and π'' is a derangement of the $e - r + 1$ elements of $V_2 \cup \{z\}$ and has the same value as id for the remaining $e - 1$ elements for which id and π differ. Thus $d_H(id, \pi'') = e - r + 1$. In a similar way, π'' has the same value as π for the elements of V_2, a different value for the element z and a different value for the remaining $e - 1$ elements for which id and π differ. Thus $d_H(\pi, \pi'') = e$. It follows that the balls surrounding id and π are not disjoint. □

A key step in a construction of $[n, e]$ permutation codes is to find an efficient means of determining, for two permutations π_1 and π_2, whether or not the two balls of radius $e = p(C)$ surrounding the permutations are disjoint. Theorem 1 shows that if $d_H(\pi_1, \pi_2) > 2e$ they are disjoint and if $d_H(\pi_1, \pi_2) < 2e$ they are not disjoint. Thus it only remains to consider the case $d_H(\pi_1, \pi_2) = 2e$.

Theorem 2 *Let C be a permutation code (a subset of $\mathscr{S}_n$) with packing radius $e = p(C) > 1$ and minimum distance $d(C) = 2e$. Then given two codewords π_1, π_2 with $d_H(\pi_1, \pi_2) = 2e$, the balls of radius e surrounding these codewords overlap if and only if the derangement of $2e$ elements given by $\pi_2 \pi_1^{-1}$ decomposes into two disjoint derangements, each of exactly e of the $2e$ elements.*

Proof If the two disjoint derangements exist then composing π_1 with the first gives a permutation v with $d_H(\pi_1, v) = e$. Composing v with the second derangement gives π_2 with $d_H(\pi_2, v) = e$. Thus $B_e(\pi_1)$ and $B_e(\pi_2)$ overlap. If a permutation v with $d_H(\pi_1, v) = e$ and $d_H(\pi_2, v) = e$ exists then v is a derangement of e elements of the image of π_1. Composing with a further derangement of e elements only gives a derangement of $2e$ elements if the two derangements are of disjoint sets of elements. □

To test two balls for overlap Theorem 1 is applied first. Only if the Hamming distance is $2e = 2p(C)$ need Theorem 2 be applied. Then the derangement of $2e$ positions given by $\pi_2 \pi_1^{-1}$ is written in cycle notation. (By concentrating on these $2e$ positions there will be no cycles of length 1.) Suppose that there are ℓ cycles. Record the lengths of the cycles in non-increasing order as a vector $\mathbf{v}$ of length ℓ. For example, if the derangement is (in cycle notation) (4 8 5 6)(1 2 7 3)(10 9)(12 11) then $\mathbf{v} = (4, 4, 2, 2)$. Now compare the vector with the vectors of this length that correspond to two disjoint derangements. (In this case the derangement does correspond to two disjoint derangements, each on six elements, and the balls are not disjoint). In general, the vector $\mathbf{v}$ can be compared with pregenerated lists where there exist or do not exist two disjoint arrangements on e of the $2e$ elements. Table 1

Table 1 Two disjoint derangements on e elements exist, d = 2e and $4 \leq d \leq 12$

$d = 2e$	ℓ = length of vector	**v** = vector of cycle lengths
4	2	(2,2)
6	2	(3,3)
8	2	(4,4)
8	3	(4,2,2)
8	4	(2,2,2,2)
10	2	(5,5)
10	3	(5,3,2)
10	4	(3,3,2,2)
12	2	(6,6)
12	3	(6,4,2); (6,3,3)
12	4	(6,2,2,2); (4,4,2,2); (4,3,3,2); (3,3,3,3)
12	5	(4,2,2,2,2); (3,3,2,2,2)
12	6	(2,2,2,2,2,2)

shows all vectors **v** corresponding to two disjoint derangements with $e > 1$ and $d = 2e \leq 12$ (two disjoint derangements cannot exist if $e = 1$, in fact the balls consist of the codewords themselves). Table 2 shows all vectors **v** that do not correspond to two disjoint derangements with $e > 1$ and $d = 2e \leq 12$.

3 The new table

The case $e = 1$ corresponds to an $(n, 2)$ permutation code obtained from the permutations of the symmetric group $\mathscr{S}_n$ (giving $M(n, 2) = n!$) and so need not be shown in the table. The computations described in this paper are feasible for all relevant e up to $n = 12$ and for some e up to $n = 15$. Thus Table 3 covers $4 \leq n \leq 15$ and $2 \leq e \leq 6$. As well as the lower bound LB for $P[n, e]$ obtained, an upper bound is also given. The standard upper bound $M(n, d) \leq n!/(d-1)!$ (see for example [9]) gives $P[n, e] \leq n!/(2e-1)!$ as $d \geq 2e$ for disjoint spheres. Also, as the balls of radius e are disjoint, another upper bound is given by $n!/|V_e|$. The entry UB in Table 3 is the smaller of these two upper bounds. Also shown for comparison is the size of the best known (n, d) permutation code with $d = 2e + 1$ [24], which is also guaranteed to correct e errors. An entry is shown in bold if it is larger than the corresponding $d = 2e + 1$ entry, or if the $(n, 2e + 1)$ code does not exist. The meaning of the superscripts in the table is as follows:

Superscript A: "Iterative clique building"—codewords only (Sect. 5)
Superscript C: "Iterative clique building"—cycles of length n (Sect. 5)
Superscript E: "Iterative clique building"—cycles of length $n - 1$ (Sect. 5)

All other entries were obtained using automorphism groups and maximum clique algorithms, as described in Sect. 4.

4 Automorphism groups and maximum clique algorithms

The construction of $[n, e]$ permutation codes using automorphism groups follows closely the method in [24], which is itself based on the method presented in [9]. A permutation code C

Table 2 Two disjoint derangements on e elements do not exist, d = 2e and $4 \leq d \leq 12$

$d = 2e$	ℓ = length of vector	**v** = vector of cycle lengths
4	1	(4)
6	1	(6)
6	2	(4,2)
6	3	(2,2,2)
8	1	(8)
8	2	(6,2); (5,3)
8	3	(3,3,2)
10	1	(10)
10	2	(8,2); (7,3); (6,4)
10	3	(6,2,2); (4,4,2); (4,3,3)
10	4	(4,2,2,2)
10	5	(2,2,2,2,2)
12	1	(12)
12	2	(10,2); (9,3); (8,4); (7,5)
12	3	(8,2,2); (7,3,2); (5,5,2); (5,4,3); (4,4,4)
12	4	(5,3,2,2)

has a left automorphism group H if $hC = C$ for all $h \in H$. If H is applied to the identity permutation a single orbit is obtained. In general a code C with this automorphism group will consist of a union of $|C|/|H|$ orbits. The automorphism group must be chosen so that the minimum distance between any pair of codewords in a single orbit (referred to as the *internal minimum distance*) is either at least $2e + 1$ (Theorem 1), or is $2e$ and no pair of codewords at distance $2e$ overlap (Theorem 2). A graph $G(n, e)$ is constructed with one vertex for each orbit of H. Given two orbits O_i and O_j corresponding to vertices v_i and v_j of $G(n, e)$, the two vertices are adjacent in $G(n, e)$ if:

1. $d(O_i, O_j) = \min_{g_s \in O_i,\ g_t \in O_j} d_H(g_s, g_t) \geq 2e + 1$ or
2. $d(O_i, O_j) = 2e$ and for any pair $g_s \in O_i,\ g_t \in O_j$ such that $d_H(g_s, g_t) = 2e$, the "no overlap" condition $B_e(g_s) \cap B_e(g_t) = \phi$ holds.

Note that in order to compute $d(O_i, O_j)$ it is only necessary to choose representatives g_s, g_t of the two orbits and then compute $\min_{h \in H} d(g_s, hg_t)$. A maximum clique in $G(n, e)$ then gives a union of orbits that corresponds to the largest $[n, e]$ code with the given automorphism group H. The size of this code is a lower bound for $P[n, e]$.

The maximum clique algorithms used were based on the selection in [24], where further details can be found. In smaller cases the algorithms terminate. In some cases the algorithm described in [20,21] would terminate in at most a few days where other algorithms did not. When no algorithm would terminate in reasonable time it was found useful to take the best result from several algorithms, with no one algorithm dominating. Four different algorithms were used for cases that did not terminate in the final results, as well as for groups that did not ultimately give the best result:

1. The software system FASoft used for radio frequency assignment [18] contains a maximum clique algorithm based on that described in [8]. Seven different vertex orderings

Table 3 Table of best permutation codes of length n and packing radius e

n	e				
	2	**3**	**4**	**5**	**6**
4					
UB	3	–	–	–	–
LB	**2**	–	–	–	–
$d = 2e + 1$ best	–	–	–	–	–
5					
UB	10	–	–	–	–
LB	**10**	–	–	–	–
$d = 2e + 1$ best	5	–	–	–	–
6					
UB	45	6	–	–	–
LB	**30**[A]	**6**	–	–	–
$d = 2e + 1$ best	18	–	–	–	–
7					
UB	229	42	–	–	–
LB	**126**	**22**[A]	–	–	–
$d = 2e + 1$ best	77	7	–	–	–
8					
UB	1390	285	8	–	–
LB	**896**	**112**	**8**[A]	–	–
$d = 2e + 1$ best	616	56	–	–	–
9					
UB	9807	1770	72	–	–
LB	**4176**	504	**25**[A]	–	–
$d = 2e + 1$ best	3024	504	9	–	–
10					
UB	78886	12688	720	10	–
LB	**50400**	**2880**	**110**[C]	**10**[A]	–
$d = 2e + 1$ best	18720	720	49	–	–
11					
UB	712800	103411	7920	110	–
LB	**475200**	**15840**	**1210**	**33**[A]	–
$d = 2e + 1$ best	205920	7920	154	11	–
12					
UB	7149277	944776	95040	1320	12
LB	**2471040**	**190080**	**3960**	**144**[C]	**12** [A]
$d = 2e + 1$ best	2376000	95040	1320	60	–
13					
UB	78823048	9565316	878777	17160	156
LB	–	**247104**	**15120**[E]	**612**[E]	**40**[A]
$d = 2e + 1$ best	878778	95040	4810	195	13

Table 3 Continued

n	e				
	2	**3**	**4**	**5**	**6**
14					
UB	947590121	106314989	8869497	240240	2184
LB	–	–	**110682**[E]	**3483**[A]	**169**[E]
$d = 2e + 1$ best	–	–	6552	2184	52
15					
UB	12336550641	1287081070	98313989	3603600	32760
LB	–	–	–	**15120**	**769**[A]
$d = 2e + 1$ best	–	–	–	6076	243

UB denotes an upper bound given by the smaller of a simple sphere packing upper bound and $n!/(2e-1)!$. LB denotes the lower bound given by the $[n, e]$ permutation code (with packing radius e) constructed in this paper. Also shown for comparison is the size of the best known (n, d) permutation code with $d = 2e + 1$, which is also guaranteed to correct e errors. An entry is shown in bold if it is larger than the corresponding $d = 2e + 1$ entry, or if the $(n, 2e + 1)$ code does not exist

can be applied before the algorithm is run, and can give very different results if the algorithm does not terminate. These are:

Initial ordering: The algorithm in [8] is applied with the order of vertices as presented by the problem.
Largest degree first (LF1): The vertices are sorted in decreasing order of their degrees before the algorithm is applied.
LF1 reversed: The reverse of the above ordering.
Largest degree first (LF2): The vertices of largest degree are successively removed from the graph and added to a list. This time the degree calculation excludes vertices that have already been ordered and removed from the graph.
LF2 reversed: The reverse of the above ordering.
Smallest degree last (SL): The vertices of smallest degree are successively removed from the graph and added to a list. Again the degree calculation excludes vertices that have already been ordered and removed from the graph. When all vertices have been removed the list is reversed.
SL reversed: The reverse of the above ordering.

2. Perturbations of the most promising ordering above were used within a multi-start framework. The algorithm described in [8] was repeatedly run for a period of 600 s, each time with perturbations of the chosen ordering. The ordering is perturbed by introducing some noise in the calculation of the vertex degrees. Each vertex degree deg_i is changed into a random value in the interval $[(1-Per)deg_i, (1+Per)deg_i]$, where Per is a user-defined parameter in the interval [0.05, 0.15].
3. The algorithm described in [4] gave the best result in one case.
4. The algorithm described in [5] gave the best result in one case.

Some standard groups such as the cyclic groups $\mathscr{C}_n$, $\mathscr{C}_{n-1}$, the dihedral group $\mathscr{D}_{2n}$, the groups $AGL(1, q)$, $A\Gamma L(1, q)$, $A\Sigma L(1, q)$, $PGL(2, q)$, $PSL(2, q)$, $P\Gamma L(2, q)$, $P\Sigma L(2, q)$, and the Mathieu groups M_{11} and M_{12} were used in [24] and merit consideration here. Again, a database of transitive permutation groups was also used. All computations using

automorphism groups were carried out using Magma.[1] Magma contains a database of all transitive groups with degree at most 30 [7]. This is based on one constructed by Hulpke [17] making use of a classification by Butler and McKay for degree at most 15. In [24] the largest two or three permutation groups in the database satisfying the internal minimum distance condition were used. For the present work, where the overlap condition adds some complexity, it proved useful to consider a somewhat wider selection of groups from the Magma database. However, only the group that gave the best result appearing in Table 3 is detailed here.

Automorphism groups used in individual cases in the tables will now be listed. In just one case there is only a single orbit of the group, giving a group code [1] and the decoding algorithm given in [1] could be used. Permutation generators for the automorphism groups are given. Here all permutations act on the set $\{0, 1, 2, \ldots, n-1\}$. Files of codewords and files of orbit representatives which also list these permutation generators can be found on the authors' web pages.[2]

1. $[n, e] = [4, 2]$: Use the identity group (no automorphisms). A maximum clique algorithm terminated with 2 orbits, so $P[4, 2] = 2$ is an exact result.
2. $[n, e] = [5, 2]$: Use the cyclic group $\mathscr{C}_5$, with $|\mathscr{C}_5| = 5$ and internal orbit minimum distance 5. A maximum clique algorithm terminated with 2 orbits, so referring to the upper bound $P[5, 2] = 10$.
3. $[n, e] = [6, 3]$: Use the identity group (no automorphisms). A maximum clique algorithm terminated with 6 orbits, so $P[6, 3] = 6$ is an exact result.
4. $[n, e] = [7, 2]$: Use the cyclic group $\mathscr{C}_7$, with $|\mathscr{C}_7| = 7$ and internal orbit minimum distance 7. The FASoft maximum clique algorithm found 18 orbits without terminating. The algorithm described in [20,21] did terminate with the same result, so $P[7, 2] \geq 126$. In fact the dihedral group $\mathscr{D}_7$ with $|\mathscr{D}_7| = 14$ and internal orbit minimum distance 6 gives the same result from 9 orbits with an easier clique search.
5. $[n, e] = [8, 2]$: Use the group denoted $E(8){:}7 = F_{56}(8)$ in the naming system and classification given in [11], with $|E(8){:}7| = 56$ and internal orbit minimum distance 7. A maximum clique algorithm terminated with 16 orbits, so $P[8, 2] \geq 896$. Permutation generators used were:
(0 7)(1 2)(3 4)(5 6); (0 2)(1 7)(3 5)(4 6); (0 4)(1 5)(2 6)(3 7); (0 1 5 2 3 4 6).
6. $[n, e] = [8, 3]$: Use the group $E(8){:}7 = F_{56}(8)$ as for $[n, e] = [8, 2]$. A maximum clique algorithm terminated with 2 orbits, so $P[8, 3] \geq 112$.
7. $[n, e] = [9, 2]$: Use the group denoted $E(9){:}2D_8$ in the naming system and classification given in [11], with $|E(9){:}2D_8| = 144$ and internal orbit minimum distance 6. The FASoft maximum clique algorithm found 29 orbits without terminating, so $P[9, 2] \geq 4176$. The other maximum clique algorithms used did not terminate and did not improve this result. Permutation generators used were:
(0 1 8)(2 3 4)(5 6 7); (0 3 6)(1 4 7)(2 5 8); (0 5 3 4 1 2 7 6); (0 1)(2 4)(5 6).
8. $[n, e] = [9, 3]$: Use $PSL(2, 8)$ with $|PSL(2, 8)| = 504$ and internal orbit minimum distance 7. Only a single orbit is possible, so $P[9, 3] \geq 504$ and the minimum distance of the code is 7. Permutation generators used were:
(0 8)(1 2)(3 4)(5 6); (0 1 3 2 5 6 4); (1 4)(2 5)(3 6)(7 8).
9. $[n, e] = [10, 2]$: Use the group denoted $L(10).2^2 = P\Gamma L(2, 9)$ in the naming system and classification given in [11], with $|L(10).2^2| = 1440$ and internal orbit minimum distance 6. The FASoft maximum clique algorithm found 35 orbits without terminating,

[1] http://magma.maths.usyd.edu.au/magma/.

[2] http://data.research.glam.ac.uk/projects/; http://www.idsia.ch/~roberto/packingradius11.zip.

so $P[10, 2] \geq 50400$. The other maximum clique algorithms used did not terminate and did not improve this result. Permutation generators used were:
(0 1 9)(2 3 4)(5 6 7); (0 6 2 3 1 4 5 7); (0 1)(3 6)(4 7)(8 9); (2 5)(3 6)(4 7).

10. $[n, e] = [10, 3]$: Use $PSL(2, 9)$ with $|PSL(2, 9)| = 360$ and internal orbit minimum distance 8. A maximum clique algorithm terminated with 8 orbits, so $P[10, 3] \geq 2880$. Permutation generators used were:
(0 1 9)(2 3 4)(5 6 7); (0 2 1 5)(3 4 7 6); (0 1)(3 6)(4 7)(8 9).
11. $[n, e] = [11, 2]$: Use the Mathieu group M_{11} with $|M_{11}| = 7920$ and internal orbit minimum distance 8. The clique search algorithm in [4] found 60 orbits without terminating, so $P[11, 2] \geq 475200$. Permutation generators used were:
(0 1 2 3 4 5 6 7 8 9 10); (0 2 8 4 3)(1 5 6 9 7); (1 5 9 6)(2 8 3 4).
12. $[n, e] = [11, 3]$: Use the Mathieu group M_{11} as for $[n, e] = [11, 2]$. A maximum clique algorithm terminated with 2 orbits, so $P[11, 3] \geq 15840$.
13. $[n, e] = [11, 4]$: Use the group denoted $F_{110}(11) = 11{:}10$ in the naming system and classification given in [11], with $|F_{110}(11) = 11{:}10| = 110$ and internal orbit minimum distance 10. A maximum clique algorithm terminated with 11 orbits, so $P[11, 4] \geq 1210$. Permutation generators used were:
(0 1 2 3 4 5 6 7 8 9 10); (0 1 3 7 4 9 8 6 2 5).
14. $[n, e] = [12, 2]$: Use the Mathieu group M_{12} with $|M_{12}| = 95040$ and internal orbit minimum distance 8. The FASoft maximum clique algorithm found 26 orbits without terminating, so $P[12, 2] \geq 2471040$. The other maximum clique algorithms used did not terminate and did not improve this result. Permutation generators used were:
(0 10 1 2 3)(4 7 11 5 6); (0 8 4 11 10 7 1 3)(5 9).
15. $[n, e] = [12, 3]$: Use the Mathieu group M_{12} as for $[n, e] = [12, 2]$. A maximum clique algorithm terminated with 2 orbits, so $P[12, 3] \geq 190080$.
16. $[n, e] = [12, 4]$: Use the group denoted $L(2, 11)$ in the naming system and classification given in [11], with $|L(2, 11)| = 660$ and internal orbit minimum distance 10. A maximum clique algorithm terminated with 6 orbits, so $P[12, 4] \geq 3960$. Permutation generators used were:
(0 1 2 3 4 5 6 7 8 9 11); (0 9)(1 4)(2 6)(3 7)(5 8)(10 11).
17. $[n, e] = [13, 3]$: Use $PSL(3, 3)$ with $|PSL(3, 3)| = 5616$ and internal orbit minimum distance 8. In this case it did not prove feasible to generate all constraints for a maximum clique algorithm, so a sample of the first 4000 orbits generated was used. The maximum clique algorithm in [5] found 44 orbits without terminating, so $P[13, 3] \geq 247104$. The other maximum clique algorithms used did not terminate and did not improve this result. Permutation generators used were:
(0 1 2 3 4 5 6 7 8 9 10 11 12); (1 11)(3 10)(4 5)(6 9).
18. $[n, e] = [15, 5]$: Use the group denoted $A_7(15)$ in the naming system and classification given in [11], with $|A_7(15)| = 2520$ and internal orbit minimum distance 12. In this case it did not prove feasible to generate all constraints for a maximum clique algorithm, so a sample of the first 6000 orbits generated was used. A maximum clique algorithm terminated with 6 orbits, so $P[15, 5] \geq 15120$. Permutation generators used were:
(0 8 9 2 13)(1 14 6 11 5)(3 4 10 12 7); (0 1 2)(4 5 6)(7 9 8)(11 13 12).

5 Iterative clique building

As in [24], the graph $G(n, e)$ is sometimes too large to handle directly by a maximum clique algorithm. An alternative method that leads to smaller maximum clique problems is described

in [24], where a more detailed description can be found. This iterative method starts from a given solution (clique in $G(n, e)$), that can either be provided by the user or generated by a lexicographic search method (see for example [19]) on permutations or orbit representatives.

Starting from the given initial clique $G_0(n, e)$, a random subset of the vertices of $G_0(n, e)$ is removed, leaving a first complete subgraph $G_1(n, e)$ of $G(n, e)$. A parameter $CSRem$ defines the percentage of vertices removed. All compatible vertices of $G(n, e)$ (adjacent to all those in $G_1(n, e)$) can be identified via a lexicographic search, and these compatible vertices induce a subgraph $G_2(n, e)$ of $G(n, e)$. Here compatibility can be determined by making use of Theorems 1 and 2 to determine adjacency as in Sect. 4. The subgraph $G_2(n, e)$ is much smaller than $G(n, e)$. A maximum clique algorithm is then applied to $G_2(n, e)$. If the subgraph of $G(n, e)$ induced by the vertices of $G_1(n, e)$ and the vertices of the maximum clique obtained is larger than $G_0(n, e)$, it becomes the new reference solution. Otherwise the initial solution is kept. The procedure is iteratively repeated, always starting from the best solution available. The method stops when a given maximum computation time (a few days in these experiments) has elapsed. To solve the maximum clique problem the algorithm described in [8] is used, or alternatively its multi-start iterative modification described in Sect. 4. These internal maximum clique algorithms are run for a maximum time of 3,600 s at each iteration. Typically, values between 8 and 16 can be used for the parameter $CSRem$.

This iterative clique building approach has been applied as described above to graphs $G(n, e)$ obtained directly from permutations (superscript A in Table 3). It has also been applied to graphs $G(n, e)$ obtained using cyclic automorphisms $\mathscr{C}_n$ (superscript C) and $\mathscr{C}_{n-1}$ (superscript E).

6 Conclusion

It has been shown that it is feasible to construct permutation codes with packing radius e for $n \leq 15$ by the methods previously introduced. Examination of Table 3 shows that substantially larger e-error-correcting permutation codes can be obtained by specifying packing radius e than by specifying minimum distance $2e + 1$. In fact the minimum distance of the code obtained is $2e$ in all cases except [9, 3]. As a result of the use of Theorems 1 and 2, application of the methods from [24] becomes only mildly more computationally demanding.

It is interesting to note that although automorphism groups with internal orbit minimum distance $2e$ and without overlap certainly exist in some cases (for example the dihedral group for [8, 3]), they do not appear in any of the final constructions. It is the inter-orbit minimum distances that may be $2e$ (without overlap). This perhaps explains why the case [9, 3], is not an improvement. At least one group for this case has internal orbit minimum distance 6 and several orbits without overlap. However, by far the best group has a single orbit with internal orbit minimum distance 7 and no inter-orbit distances. As was the case in [24], all the codes constructed appear to be maximal by inclusion. It has been checked directly that it is impossible to add even a single extra permutation in any of the cases with $n \leq 10$.

Acknowledgment R. Montemanni acknowledges the support of the Hasler Foundation through Grant 11158.

References

1. Bailey R.F.: Error-correcting codes from permutation groups. Discrete Math. **309**, 4253–4265 (2009).
2. Blake I.F.: Permutation codes for discrete channels. IEEE Trans. Inf. Theory **20**(1), 138–140 (1974).

3. Bogaerts M.: New upper bounds for the size of permutation codes via linear programming. Electron. J. Comb. **17**(#R135), 9 (2010).
4. Burer S., Monteiro R.D.C., Zhang Y.: Maximum stable set formulations and heuristics based on continuous optimization. Math. Program. A **94**, 137–166 (2002).
5. Busygin S.: A new trust region technique for the maximum weight clique problem. Discrete Appl. Math. **154**, 2080–2096 (2006).
6. Cameron P.J., Wanless I.M.: Covering radius for sets of permutations. Discrete Math. **293**, 91–109 (2005).
7. Cannon J.J., Bosma W. (eds.): Handbook of Magma Functions, Version 2.13. The University of Sydney, Sydney (2006).
8. Carraghan R., Pardalos P.M.: An exact algorithm for the maximum clique problem. Oper. Res. Lett. **9**, 375–382 (1990).
9. Chu W., Colbourn C.J., Dukes P.: Constructions for permutation codes in powerline communications. Des. Codes Cryptogr. **32**, 51–64 (2004).
10. Colbourn C.J., Kløve T., Ling A.C.H.: Permutation arrays for powerline communication and mutually orthogonal latin squares. IEEE Trans. Inf. Theory **50**, 1289–1291 (2004).
11. Conway J.H., Hulpke A., McKay J.: On transitive permutation groups. LMS J. Comput. Math. **1**, 1–8 (1998).
12. Deza M., Vanstone S.A.: Bounds for permutation arrays. J. Stat. Plan. Inference **2**, 197–209 (1978).
13. Dukes P., Sawchuck N.: Bounds on permutation codes of distance four. J. Algebr. Comb. **31**, 143–158 (2010).
14. Frankl P., Deza M.: On maximal numbers of permutations with given maximal or minimal distance. J. Comb. Theory Ser. A **22**, 352–360 (1977).
15. Han Vinck A.J.: Coded modulation for power line communications. A.E.Ü. Int. J. Electron. Commun. **54**(1), 45–49 (2000).
16. Huczynska S.: Powerline communications and the 36 officers problem. Phil. Trans. R. Soc. A. **364**, 3199–3214 (2006).
17. Hulpke A.: Constructing transitive permutation groups. J. Symb. Comput. **39**(1), 1–30 (2005).
18. Hurley S., Smith D.H., Thiel S.U.: FASoft: a system for discrete channel frequency assignment. Radio Sci. **32**(5), 1921–1939 (1997).
19. Montemanni R., Smith D.H.: Heuristic algorithms for constructing binary constant weight codes. IEEE Trans. Inf. Theory **55**(10), 4651–4656 (2009).
20. Östergård P.R.J.: A new algorithm for the maximum-weight clique problem. Nord. J. Comput. **8**(4), 424–436 (2001).
21. Östergård P.R.J.: A fast algorithm for the maximum clique problem. Discrete Appl. Math. **120**, 197–207 (2002).
22. Pavlidou N., Han Vinck A.J., Yazdani J., Honary B.: Power line communications: state of the art and future trends. IEEE Commun. Mag. **41**(4), 34–40 (2003).
23. Quistorff J.: A survey on packing and covering problems in the Hamming permutation space. Electron. J. Comb. **13**(#A1) (2006).
24. Smith D.H., Montemanni R.: A new table of permutation codes. Des. Codes Cryptogr. (2011). doi:10.1007/s10623-011-9551-8.
25. Tarnanen H.: Upper bounds on permutation codes via linear programming. Eur. J. Combin. **20**, 101–114 (1999).