Journal of Optimization Theory and Applications
https://doi.org/10.1007/s10957-022-02035-w

The Maximum Clique Problem for Permutation Hamming Graphs

János Barta[1] · Roberto Montemanni[2]

Received: 11 May 2021 / Accepted: 5 April 2022

Abstract

This paper explores a new approach to reduce the maximum clique problem associated with permutation Hamming graphs to smaller clique problems. The vertices of a permutation Hamming graph are permutations of n integers and the edges connect pairs of vertices at a Hamming distance greater than or equal to a threshold d. The maximum clique problem for permutation Hamming graphs is a challenging task due to the size, density and regularity of the graphs. However, symmetry properties, which are still partly unexplored, can help to reduce the problems' size and hardness. A property of edge transitivity with respect to automorphisms is proven and leads to a classification for cycle-equivalent edges. This property enables to reduce the full-size clique problem to a set of significantly smaller (and easier to solve) clique problems. The number of reduced problems can be expressed by means of the partition function of integer numbers. Computational experiments confirm that additional knowledge on the automorphism group leads to a more targeted and efficient solving method for the maximum clique problem associated with permutation Hamming graphs.

Keywords Graph theory · Combinatorics · Maximum clique problem · Permutations · Hamming graphs

Communicated by Martine Labbé.

✉ János Barta
janos.barta@supsi.ch

Roberto Montemanni
roberto.montemanni@unimore.it

[1] Dalle Molle Institute for Artificial Intelligence, IDSIA—USI/SUPSI, 6962 Lugano, Switzerland

[2] Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, 42122 Reggio Emilia, Italy

Published online: 13 May 2022

1 Introduction

How many permutations of the n-tuple $[0, 1, \ldots, n-1]$ can be chosen in such a way that they are at least at a Hamming distance d from each other? This optimization problem, usually referred to as the Maximum Permutation Code Problem (MPCP), has been extensively studied with several approaches, such as mixed-integer linear programming [6, 18], algebraic methods [5, 7, 9, 11], branch and bound algorithms [4], heuristic algorithms [14, 16] and also with graph theoretical methods [2, 3, 10]. The MPCP is a well-known problem in coding theory with relevant applications in engineering (see for instance [7, 8, 12, 15]). As detailed in Sect. 3, the MPCP can always be reformulated in graph theoretical terms as a maximum clique problem on a permutation Hamming graph (see also [2, 17]). We recall that the maximum clique problem is the optimization problem of finding a largest complete subgraph, a so-called maximum clique, in a given graph. The clique number $\omega(G)$ of a graph G is the order of its maximum clique. Despite the high symmetry degree of permutation Hamming graphs, no general formula for their clique number is known so far. On the other hand, from a computational point of view symmetries represent a difficulty for optimizations algorithms, because of the presence of a high number of equivalent suboptimal solutions, which tend to hide the optimum. Therefore, in order to tackle the maximum clique problem, it is crucial to gain more knowledge about the structure and the symmetries of permutation Hamming graphs. In the papers Barta and Montemanni [2, 3], several relevant properties of permutation Hamming graphs were investigated. In Barta and Montemanni [2], exact formulae for the degree of the vertices and for the number of edges were derived. Furthermore, vertex transitivity and r-partiteness of permutation Hamming graphs were proven. In Barta and Montemanni [3], it was shown that permutation Hamming graphs can be generated as the intersection of a set of Turán graphs. This property led to a hitting set formulation of the maximum clique problem.

In this paper, the focus is on automorphism groups of permutation Hamming graphs. The article starts with a definition, in Sect. 2, of position and value permutations of codewords. In Sects. 3 and 4, it is shown that position and value permutations are automorphisms of permutation Hamming graphs. In Sect. 5, we prove that it is always possible to define an automorphism which maps an edge of a permutation Hamming graph into another edge with an equivalent cycle structure (see Theorem 5.1). This theorem turns out to be useful, since it enables to reduce the size of the maximum clique problem by fixing two vertices of the graph. More precisely, the full-size maximum clique problem can be replaced by a set of smaller instances with a potential gain in computational time (see Sect. 6). The computational tests reported in Sect. 7 confirm that the reduction approach proposed can improve running times.

2 Position and Value Permutations

Let Ω_n be the set of all permutations of the n-tuple $x_0 = [0, 1, \ldots, n-1]$. Each element $x \in \Omega_n$ is a codeword and the Hamming distance $d_H(x, y)$ between two codewords x and y is the number of differing components in x and y. Each codeword can be interpreted as an assignment between a set of positions $P = \{0, \dots, n-1\}$ and a set of values $V = \{0, \dots, n-1\}$. The expression $v = x(p)$ indicates that the value at position p in the codeword x is v. On the other hand, we denote by $p = x_{\text{pos}}(v)$ the position of the value v in the codeword x.

There are many possible ways to map a codeword into another. With the following definitions, we focus on two types of permutations which will play an important role in this work:

Definition 2.1 Let $x \in \Omega_n$ and let $\pi \in S_n$ be a permutation of the symmetric group S_n. We refer to x' as the position permutation of x based on π, if x' is obtained by permuting the positions of x according to π, that is, $x'(\pi(p)) = x(p), \forall p \in P$.

As an example, let $x = [2013]$ and $\pi \in S_4$, such that $\pi(0) = 2,\ \pi(1) = 1,\ \pi(2) = 3$ and $\pi(3) = 0$. Then the position permutation of x induced by π is the codeword $x' = [3021]$.

Definition 2.2 Let $x \in \Omega_n$ and $\pi \in S_n$. We refer to x' as the value permutation of x based on π, if x' is obtained by permuting the values of x according to π, that is, $x'_{\text{pos}}(\pi(v)) = x_{\text{pos}}(v)\ ,\ \forall v \in V$.

Consider again the codeword $x = [2013]$ and the permutation π defined above. The value permutation of x induced by π is the codeword $x'' = [3210]$. Note that the same permutation π generates two different images x' and x''. In the first case, π permutes the positions of x, in the second case the values.

Now we can define position and value permutations as bijections from Ω_n to itself.

Definition 2.3 Let $\pi \in S_n$. We call the map $\sigma_\pi : \Omega_n \to \Omega_n$ a position permutation based on π, if for any $x \in \Omega_n$ and $x' = \sigma_\pi x$ holds $x'(\pi(p)) = x(p)\ ,\ \forall p \in P$.

Definition 2.4 Let $\pi \in S_n$. We call the map $\tau_\pi : \Omega_n \to \Omega_n$ a value permutation based on π, if for any $x \in \Omega_n$ and $x' = \tau_\pi x$ holds $x'_{\text{pos}}(\pi(v)) = x_{\text{pos}}(v)\ ,\ \forall v \in V$.

A permutation $\pi \in S_n$ can be represented in different ways. For the purpose of our work, the standard cycle notation turns out to be helpful. A permutation $\pi \in S_n$ contains a cycle $(i_1\ i_2 \cdots i_k)$ if $\pi(i_1) = i_2, \dots, \pi(i_{k-1}) = i_k$ and $\pi(i_k) = i_1$, that is, each value in the cycle is mapped into the next one. A permutation can be formed by more than one cycle. In this case the cycles are listed by a non-decreasing cycle length. The cycles of length 1 correspond to the fixpoints of the permutation, and they may be omitted unless this leads to misunderstanding. As an example $\pi = (02)(145) \in S_6$ is a compact way to write $\pi(3) = 3$, $\pi(0) = 2$, $\pi(2) = 0$, $\pi(1) = 4$, $\pi(4) = 5$ and $\pi(5) = 1$.

Consider now the permutation $\pi = (23)$ and the codewords $x_1 = [1032]$ and $x_2 = [3012]$. By applying the position permutation σ_π, we obtain the images $\sigma_\pi x_1 = [1023]$ and $\sigma_\pi x_2 = [3021]$. On the other hand, the value permutation τ_π yields $\tau_\pi x_1 = [1023]$ and $\tau_\pi x_2 = [2013]$. It can be noticed that $\sigma_\pi x_2 \neq \tau_\pi x_2$ although $\sigma_\pi x_1 = \tau_\pi x_1$. Thus, position and value permutations are distinct maps of Ω_n onto itself.

3 Permutation Hamming Graphs

A well-known problem in coding theory is determining the maximum number of codewords in Ω_n, such that the Hamming distance of each pair of codewords is greater than or equal to a given threshold d. This problem is referred to as the Maximum Permutation Code Problem (MPCP). The MPCP can be easily reformulated as a maximum clique problem on a permutation Hamming graph.

Definition 3.1 The permutation Hamming graph $H(n, d)$ is the weighted graph defined on the vertex set $V_H = \Omega_n$, in which a pair of nodes $x, y \in V_H$ is connected by an edge, iff $d_H(x, y) \geq d$. The weight of the edge xy is $w(x, y) = d_H(x, y)$, and the set of edges is denoted by E_H.

Since a clique is a complete subgraph, a maximum clique of $H(n, d)$ corresponds to the largest permutation code C that satisfies the distance constraint $d_H(x, y) \geq d, \forall x, y \in C$. As $|V_H| = n!$, a direct calculation of the clique number $\omega(H(n, d))$ becomes generally a hard task for $n \geq 7$. However, peculiar regularity properties of Hamming graphs investigated in Barta and Montemanni [2, 3] suggest that the calculation of the clique number might be facilitated by an exploitation of the symmetry properties of the graph. Basically, looking for symmetries in a graph means gaining more insight into the structure of its automorphism group.

In Barta and Montemanni [2], it was shown that the vertex degree of permutation Hamming graphs is constant, that is, $H(n, d)$ is a regular graph. Moreover, in Barta and Montemanni [2] it was also proved that $H(n, d)$ is vertex transitive. Vertex transitivity means that for any pair of vertices $x, y \in V_H$ there exists an automorphism which maps x into y. We recall the definition of a graph automorphism:

Definition 3.2 An automorphism of a graph $G(V, E)$ is a bijection $\varphi : V \to V$, which preserves the connectivity between nodes, i.e. for any pair of nodes $x, y \in V$ it holds $x \sim y \Leftrightarrow \varphi(x) \sim \varphi(y)$. The set of all automorphisms of the graph G is a group denoted by $\mathrm{Aut}(G)$. In the case of a weighted graph, an automorphism must be also weight preserving, that is, $w(\varphi(x), \varphi(y)) = w(x, y), \forall x, y \in V$.

4 The Automorphism Group

The following propositions establish a link between permutations and the automorphism group of $H(n, d)$.

Proposition 4.1 *Any position permutation $\sigma_\pi : \Omega_n \to \Omega_n$ is an automorphism of the Hamming graph $H(n, d)$.*

Proof Clearly, a position permutation on Ω_n is a bijection. It remains to prove that σ_π is distance preserving. Let x, y be two codewords and let $x' = \sigma_\pi x$, $y' = \sigma_\pi y$ be their images. Definition 2.3 implies that $x(p) = y(p) \Leftrightarrow x'(\pi(p)) = y'(\pi(p)), \forall p \in P$. In other terms, x and y share component p, if and only if x' and y' share component $\pi(p)$. Since π is bijective, the number of differing components remains unchanged, that is, $d_H(x, y) = d_H(x', y')$. □

Proposition 4.2 *Any value permutation $\tau_\pi : \Omega_n \to \Omega_n$ is an automorphism of the Hamming graph $H(n, d)$.*

Proof Analogously to Proposition 4.1, value permutations are bijections and it holds $x_{\text{pos}}(v) = y_{\text{pos}}(v) \Leftrightarrow x'_{\text{pos}}(\pi(v)) = y'_{\text{pos}}(\pi(v))$. That is, the value v is at the same position in x and y, if and only if $\pi(v)$ is at the same position in x' and y'. It follows that the number of shared components remains unchanged and therefore the mapping τ_π is distance preserving. □

Since $\text{Aut}(H(n, d))$ has a group structure, the next proposition follows immediately.

Proposition 4.3 *Any sequence of compositions of position permutations and value permutations in Ω_n is an automorphism of $H(n, d)$.*

It is important to remark that $\text{Aut}(H(n, d))$ is not a commutative group. In fact, for $\pi_1, \pi_2 \in S_n$ in general it holds $\pi_1\pi_2 \neq \pi_2\pi_1$ and as a consequence $\sigma_{\pi_1}\sigma_{\pi_2} = \sigma_{\pi_1\pi_2} \neq \sigma_{\pi_2}\sigma_{\pi_1} = \sigma_{\pi_2\pi_1}$ and respectively $\tau_{\pi_1}\tau_{\pi_2} = \tau_{\pi_1\pi_2} \neq \tau_{\pi_2}\tau_{\pi_1} = \tau_{\pi_2\pi_1}$. However, there is a relevant special case of commuting automorphisms in $\text{Aut}(H(n, d))$.

Proposition 4.4 *Let $\pi_1, \pi_2 \in S_n$ and let σ_{π_1} be the position permutation based on π_1 and τ_{π_2} the value permutation based on π_2. Then it holds $\sigma_{\pi_1}\tau_{\pi_2} = \tau_{\pi_2}\sigma_{\pi_1}$.*

Proof Let $x \in \Omega_n$ with $x(i) = u$. Furthermore let $\pi_1(i) = j$ and $\pi_2(u) = v$. The position permutation σ_{π_1} moves the position i into j, whereas the value permutation τ_{π_2} maps the value u into v. Let $y = \sigma_{\pi_1}x$. From the definition of position permutations, it follows that $y(j) = y(\pi_1(i)) = x(i) = u$. Now let $y' = \tau_{\pi_2}y$. This implies that $y'_{\text{pos}}(v) = y'_{\text{pos}}(\pi_2(u)) = y_{\text{pos}}(u) = j$, or equivalently $y'(j) = (\tau_{\pi_2}\sigma_{\pi_1}x)(j) = v$.

On the other hand, let $z = \tau_{\pi_2}x$. From the definition of value permutations, it follows that $z_{\text{pos}}(v) = z_{\text{pos}}(\pi_2(u)) = x_{\text{pos}}(u) = i$, or equivalently $z(i) = v$. Finally, let $z' = \sigma_{\pi_1}z$. This means that $z'(j) = z'(\pi_1(i)) = z(i) = (\sigma_{\pi_1}\tau_{\pi_2}x)(j) = v$. Since this equivalence holds for any position i and any codeword x, it follows that $\tau_{\pi_2}\sigma_{\pi_1} = \sigma_{\pi_1}\tau_{\pi_2}$. □

As an example, consider the permutations $\pi_1 = (0312)$, $\pi_2 = (13) \in S_4$ and the codeword $x = [1032]$. It is not difficult to verify that $\sigma_{\pi_1}x = [3201]$ and $\tau_{\pi_2}\sigma_{\pi_1}x = [1203]$. Applying the automorphisms in the opposite order yields $\tau_{\pi_2}x = [3012]$ and $\sigma_{\pi_1}\tau_{\pi_2}x = [1203]$.

The commutation of position and value permutations has an interesting consequence on the concatenation of position and value permutations.

Proposition 4.5 *For any sequence of compositions of position and value permutations $\psi : \Omega_n \to \Omega_n$, there exist $\pi_a, \pi_b \in S_n$ such that $\psi = \sigma_{\pi_a}\tau_{\pi_b}$.*

Proof Let ψ be an arbitrary concatenation of position and value permutations. By combining the associativity property of automorphisms and the commutation property stated in Proposition 4.4, permutations can be rearranged in such a way that all position permutations precede the value permutations. More formally, there exist $\pi_a, \pi_b \in S_n$ such that $\psi = (\sigma_{\pi_1} \ldots \sigma_{\pi_r})(\tau_{\pi_{r+1}} \ldots \tau_{\pi_s}) = \sigma_{\pi_a}\tau_{\pi_b}$. □

Proposition 4.5 suggests that there exists a subgroup of $\mathrm{Aut}(H(n,d))$ formed by the compositions of position and value permutations. In fact, according to Proposition 4.5, the composition of two position-value permutations is again a position-value permutation.

Definition 4.1 Let $\pi_1, \pi_2 \in S_n$. We refer to the map $\psi = \sigma_{\pi_1}\tau_{\pi_2}$, as a *coupled position-value permutation*. The subgroup of $\mathrm{Aut}(H(n,d))$ formed by coupled position-value permutations is denoted by $\mathcal{PV}_n$.

Summarizing, $\mathcal{PV}_n$ is a well-defined subgroup of the automorphisms group $\mathrm{Aut}(H(n,d))$. But what is the order of $\mathrm{Aut}(H(n,d))$ itself? This interesting question is still an open issue. However, an upper bound can be provided by means of the so-called *Rencontres numbers*.

Definition 4.2 The Rencontres number $D(n,k)$ is the number of permutations of the codeword $[0,\ldots,n-1]$ with exactly k fixpoints, i.e. the number of codewords at Hamming distance $n-k$ from x.

Consider the case of the complete graph $H(n,2)$. A given vertex x has $n!$ possible images $\psi(x)$ under an automorphism ψ. The vertex x has $D(n,0)$ neighbours without fixpoints, which have to be mapped into the $D(n,0)$ neighbours without fixpoints of the image vertex $\psi(x)$. There are at most $D(n,0)!$ ways to do it. Similarly, there are at most $D(n,k)!$ ways to map the neighbours with k fixpoints in a distance-preserving manner. Therefore the following formula provides an upper bound to the number of automorphisms of $H(n,2)$:

$$UB(n,2) = n!\prod_{k=0}^{n} D(n,k)! \tag{1}$$

It is interesting to remark that in the simplest case $n=3$ the upper bound $UB(3,2) = 3!(2!\,3!\,0!\,1!) = 72$ coincides with the exact number of automorphisms of $H(3,2)$. For larger values of n, the upper bound grows rapidly and it is likely that it overestimates the order of $\mathrm{Aut}(H(n,d))$.

5 Automorphisms and Cycle-Equivalence of Edges

A crucial property of coupled position-value permutations $\psi \in \mathcal{PV}_n$ appears when focusing on the edges of the Hamming graph $H(n,d)$, rather than on single vertices. In fact, under a coupled position-value permutation the image of an edge is another edge with the same cyclic structure.

Definition 5.1 We call the permutations $\pi_1, \pi_2 \in S_n$ *cycle-equivalent*, if π_1 and π_2 have the same number of cycles and the same cycle lengths.

For example, $\pi_1 = (5)(14)(023)$ and $\pi_2 = (0)(24)(135)$ are cycle-equivalent. In an analogous way, we can define cycle-equivalence also for the edges of the Hamming graph $H(n,d)$.

Definition 5.2 We refer to τ_{π_1} as the value permutation of the edge $e = xy$, if $\tau_{\pi_1} x = y$. Similarly, σ_{π_2} is the position permutation of the edge e, if $\sigma_{\pi_2} x = y$.

Definition 5.3 We call the edges $e_1 = xy$ and $e_2 = uv$ cycle-equivalent, if their value- respectively position permutations are cycle-equivalent.

As an example, consider the edges $e_1 = xy$ with $x = [1032]$, $y = [1023]$ and $e_2 = uv$ with $u = [3012]$, $v = [3210]$. The value permutation of the edge e_1 is $\tau_{\pi_1} = (0)(1)(23)$, whereas the value permutation of e_2 is $\tau_{\pi_2} = (1)(3)(02)$. Since the cycle lengths coincide, the two edges e_1 and e_2 are cycle-equivalent.

Theorem 5.1 *Let e_1 and e_2 be two cycle-equivalent edges of $H(n, d)$. There exists an automorphism ψ of $H(n, d)$, which maps the endpoints of e_1 into the endpoints of e_2.*

Proof Let $e_1 = xy, e_2 = uv$ and let τ_{π_1}, τ_{π_2} be their value permutations, that is, $\tau_{\pi_1} x = y$ and $\tau_{\pi_2} u = v$. Because of the cycle-equivalence of e_1 and e_2, π_1 and π_2 have the same cycle structure. Let $\pi_1 = (a_{11} \dots a_{1l_1})(a_{21} \dots a_{2l_2}) \dots (a_{m1} \dots a_{ml_m})$ and $\pi_2 = (b_{11} \dots b_{1l_1})(b_{21} \dots b_{2l_2}) \dots (b_{m1} \dots b_{ml_m})$, where $l_1 \leq l_2 \leq \dots \leq l_m$ are the cycle lengths of the m cycles. The required automorphism ψ can be constructed in two steps, as a composition of a value permutation τ_γ and a position permutation σ_δ.

Step 1: let the permutation γ be defined as $\gamma(a_{11}) = b_{11}, \dots, \gamma(a_{ml_m}) = b_{ml_m}$. We generate an intermediate edge $e' = x'y'$, such that $x' = \tau_\gamma x$ and $y' = \tau_\gamma y$. From the definition of γ it follows immediately that the value permutation of the new edge $e' = x'y'$ is τ_{π_2}, because if the value a_{ij} in x maps to $a_{i,j+1}$ in y, then similarly the value b_{ij} in x' maps to $b_{i,j+1}$ in y'.

Step 2: since the intermediate edge $e' = x'y'$ has the same value permutation as the target edge $e_2 = uv$, it suffices to apply a suitable position permutation σ_δ in order to rearrange the positions of x' and y'. Let σ_δ be the position permutation, which maps x' into u, i.e. $\sigma_\delta x' = u$. It remains to show that $\sigma_\delta y' = v$. Let's denote the inverse of the value permutation τ_{π_2} by $\tau_{\pi_2}^{-1}$. Relying on the commutation property 4.4, we obtain $\sigma_\delta y' = \sigma_\delta \tau_{\pi_2} \tau_{\pi_2}^{-1} y' = \sigma_\delta \tau_{\pi_2} x' = \tau_{\pi_2} \sigma_\delta x' = \tau_{\pi_2} u = v$. □

As an example, consider the edges $e_1 = xy$, $e_2 = uv$ defined by the vertices $x = [1032]$, $y = [1023]$, $u = [3012]$ and $v = [3210]$. The edges e_1 and e_2 are cycle-equivalent, because their value permutations are respectively $\tau_{\pi_1} = (0)(1)(23)$ and $\tau_{\pi_2} = (1)(3)(02)$. Following the proof of Theorem 5.1, define $\tau_\gamma = (0132)$. The images of x and y are respectively $x' = \tau_\gamma x = [3120]$ and $y' = \tau_\gamma y = [3102]$. It is easy to verify that the intermediate edge $e' = x'y'$ has the same value permutation as the target edge e_2. Therefore, it suffices to rearrange the positions of x' and y' through the position permutation $\sigma_\delta = (0)(123)$ in order to map x' in u and y' in v.

6 Reduction of the Maximum Clique Problem

The main hurdle in solving the maximum clique problem of the permutation Hamming graph $H(n, d)$ is its order $n!$ and its high degree of symmetry. A possible way to reduce the problem's size is to look for the largest clique of $H(n, d)$, which includes a specific fixed clique. Let K_0 be a k-clique of the graph $H(n, d)$, that is, a clique of order $k \leq \omega(H(n, d))$.

Definition 6.1 Let $R(K_0)$ be the induced subgraph of $H(n, d)$ having the vertex set $V_R = \{x \in \Omega_n \setminus V_{K_0} \mid x \sim y,\ \forall y \in V_{K_0}\}$. We denote by $R(K_0)$, or shorter by R, the *residual graph* of the clique K_0.

Each vertex of the residual graph $R(K_0)$ is connected to all vertices of the clique K_0. Let K_R be a maximum clique of the residual graph R, that is, $\omega(R(K_0)) = |V_{K_R}|$.

Proposition 6.1 *Let K_0 be a k-clique of $H(n, d)$. The largest clique K of $H(n, d)$ which includes the fixed clique K_0, has the order $|V_K| = k + \omega(R(K_0))$.*

Proof K contains all vertices of K_0. Further vertices of K must be connected to each vertex of K_0; therefore, they belong to the residual graph $R(K_0)$. But since these vertices chosen in the residual graph have to form a clique again, their maximum number is equal to the clique number $\omega(R(K_0))$ of the residual graph . □

In other words, the largest clique K is constructed by taking the union of the fixed clique K_0 and the maximum clique K_R of the residual graph. The advantage of this approach lies in the reduction in size of the clique problem to be solved. In fact, the vertex set $|V_R|$ might be significantly smaller than $|V_H|$. However, an additional constraint, forcing the k-clique K_0 into the solution, has been added to the problem. Therefore, in order solve the original maximum clique problem, all possible choices of the k-clique K_0 have to be considered.

Proposition 6.2 *Let $\mathcal{C}_k$ be the set of all k-cliques in $H(n, d)$, for a fixed $k \leq \omega(H(n, d))$. Then it holds*

$$\omega(H(n, d)) = \max_{K_0 \in \mathcal{C}_k} \omega(R(K_0)) + k. \tag{2}$$

Proof Let K_H be a maximum clique of $H(n, d)$. Since K_H certainly contains a k-clique $K_0 \in \mathcal{C}_k$, the order of K_H can be calculated by maximizing $\omega(R(K_0))$ over the set $\mathcal{C}_k$ and by adding k. □

Due to the large number of k-cliques, maximizing over the whole set $\mathcal{C}_k$ is impractical. However, because of the large automorphism group of $H(n, d)$, it can be expected that many k-cliques are mutually isomorphic. In the remainder, we denote the isomorphic graphs G_1 and G_2 by $G_1 \simeq G_2$.

Lemma 6.1 *Let $\psi \in Aut(H(n, d))$, $K_1 \in \mathcal{C}_k$ and $K_2 = \psi(K_1)$. Then it holds $K_2 \simeq K_1$ and $R(K_2) \simeq R(K_1)$.*

Proof As ψ is an adjacency and distance-preserving bijection, it maps any subgraph of $H(n, d)$ into an isomorphic subgraph and therefore $K_2 \simeq K_1$. It is to show that ψ maps $R(K_1)$ into $R(K_2)$. First we prove the inclusion $\psi(V_{R(K_1)}) \subseteq V_{R(K_2)}$. Let u be a vertex of $R(K_1)$, i.e $u \notin V_{K_1}$, but it is connected to all vertices of K_1. Since

$\psi \in \mathrm{Aut}(H(n,d))$ and $\psi(K_1) = K_2$, the vertex $v = \psi(u) \notin V_{K_2}$, but it is connected to all vertices of K_2, that is, $\psi(u) \in V_{R(K_2)}$. The opposite inclusion can be proved in an analogous way by means of the inverse automophism ψ^{-1}. This implies that $\psi(V_{R(K_1)}) = V_{R(K_2)}$, i.e. the automorphism ψ maps the vertex set of $R(K_1)$ into the vertex set of $R(K_2)$. Since ψ is adjacency and distance-preserving, it can be concluded that $\psi(R(K_1)) = R(K_2)$.

□

An obvious implication of Lemma 6.1 is that also the clique number of the residual graphs coincide, i.e. $\omega(R(K_1)) = \omega(R(K_2))$. Therefore, instead of maximizing over all k-cliques, according to Eq. (2), it suffices to consider one representative of each class of isomorphic k-cliques.

Definition 6.2 Let $K \in \mathcal{C}_k$. We refer to $O_K = \{K' \in \mathcal{C}_k | K' = \psi(K), \psi \in \mathrm{Aut}(H)\}$ as the orbit of K generated by the automorphism group $\mathrm{Aut}(H)$.

According to Lemma 6.1, each orbit O_K is a set of isomorphic k-cliques. The collection of orbits partitions $\mathcal{C}_k$ in equivalence classes.

Definition 6.3 We denote by $\mathcal{Q}_k$ the collection of orbits of k-cliques generated by $\mathrm{Aut}(H)$. Furthermore, we denote by $\mathcal{T}_k$ a transversal of $\mathcal{Q}_k$, that is, a subset of k-cliques, containing one representative for each orbit of $\mathcal{Q}_k$.

Equation (2) can now be reformulated by taking the maximum over the transversal $\mathcal{T}_k$, instead of considering all k-cliques:

$$\omega(H(n,d)) = \max_{K_0 \in \mathcal{T}_k} \omega(R(K_0)) + k. \tag{3}$$

7 Computational Experiments

The aim of the computational experiments described in this section is to highlight the advantage of handling a clique problem of the Hamming graph $H(n,d)$ by solving a set of reduced clique problems. In particular, in these preliminary tests we focus on Hamming graphs $H(n,d)$ with $n = 6, 7$ and we apply a reduction by means of 1-cliques, respectively 2-cliques. As a first step, the properties of the automorphism group $\mathrm{Aut}(H)$ are used to construct a transversal $\mathcal{T}_k$. Then, according to Eq. (3), the clique number of the residual graph $R(K_0)$ is computed for each $K_0 \in \mathcal{T}_k$.

7.1 The 1-Clique Case

As shown in Barta and Montemanni [2], a basic property of the permutation Hamming graph $H(n,d)$ is vertex transitivity. We recall that a graph $G(V,E)$ is vertex transitive iff for any pair of nodes $x, y \in V$ there exists an automorphism $\psi \in \mathrm{Aut}(G)$ such that $\psi(x) = y$. The vertex transitivity of $H(n,d)$ implies that all 1-cliques belong to the same orbit of $\mathrm{Aut}(H(n,d))$ and therefore the transversal $\mathcal{T}_1$ is formed by a single

1-clique. In other terms, any vertex $x \in \Omega_n$ can be chosen as the fixed clique K_0 and it suffices to calculate the clique number of the residual graph $R(K_0)$:

$$\omega(H(n,d)) = \omega(R(K_0)) + 1. \tag{4}$$

7.2 The 2-Clique Case

In the 2-clique case, the fixed clique K_0 is formed by a pair of nodes $x, y \in V_H$ connected by an edge $e = xy$. According to Theorem 5.1, any cycle-equivalent edge e' can be mapped by an automorphism into the edge e. In other terms, 2-cliques formed by cycle-equivalent edges belong to the same orbit of $\mathrm{Aut}(H(n,d))$. Therefore, in order to obtain the transversal $\mathcal{T}_2$, it suffices to generate the largest possible collection of pairwise not cycle-equivalent edges of $H(n,d)$. How many pairwise not cycle-equivalent edges are there in $H(n,d)$?

In the answer to this question appears an interesting link between permutation Hamming graphs and the well-known partition function of integer numbers (for details see [1]).

Definition 7.1 Let n be a positive integer. The sequence $(a_1, a_2, \ldots, a_t)$ of non-increasing positive integers is called a partition of n if $\sum_{i=1}^{t} a_i = n$. The partition function $p(n)$ indicates the number of possible partitions of the integer n.

As an example $p(4) = 5$, because $n = 4$ has 5 partitions: (4), (3,1), (2,2), (2,1,1) and (1,1,1,1). For $n = 1, 2, \ldots$ the first values of the partition function are 1, 2, 3, 5, 7, 11, 15, 22, 30, 42.

The link between permutations and partitions is given by the fact that the cycle lengths of a permutation of n elements form a partition of the integer n. As an example, the cycle lengths of the permutation of 6 elements $\pi = (051)(24)(3)$ form the partition (3,2,1) of $n = 6$. Therefore, the number of not cycle-equivalent permutations of n elements corresponds to the value of the partition function $p(n)$. This value takes into account all possible Hamming distances from 0 (n fixpoints) to n (no fixpoints). Now it is not difficult to deduce a formula for the total number of pairwise not cycle-equivalent edges of $H(n,d)$.

Proposition 7.1 *Let $c(n,d)$ be the total number of pairwise not cycle-equivalent edges of $H(n,d)$. It holds $c(n,d) = p(n) - p(d-1)$.*

Proof In the Hamming graph $H(n,d)$ there are only Hamming distances between the values d and n. In other terms all permutations with more than $n - d$ fixpoints have to be discarded. Since the cycle lengths of permutations with at least $n - d + 1$ fixpoints depend only on the remaining $d - 1$ free components, the number of possible not cycle-equivalent permutations with at least $n - d + 1$ fixpoints is equal to $p(d-1)$. It follows that the number of pairwise not cycle-equivalent edges of $H(n,d)$ is equal to the difference $p(n) - p(d-1)$. □

In the case of $H(6,5)$, for instance, the number of pairwise not cycle-equivalent edges is $c(6,5) = p(6) - p(4) = 11 - 5 = 6$. The 6 possible partitions are (6),

Table 1 Non-equivalent derangements and cycle lengths for $n \leq 7$

n	$c(n,n)$	Cycle lengths	Permutation
2	1	(2)	(01)
3	1	(3)	(012)
4	2	(4)	(0123)
		(2,2)	(01)(23)
5	2	(5)	(01234)
		(3,2)	(012)(34)
6	4	(6)	(012345)
		(4,2)	(0123)(45)
		(3,3)	(012)(345)
		(2,2,2)	(01)(23)(45)
7	4	(7)	(0123456)
		(5,2)	(01234)(56)
		(4,3)	(0123)(456)
		(3,2,2)	(012)(34)(56)

(4,2), (3,3), (2,2,2), (5,1) and (3,2,1). Note that the last two partitions contain a cycle length 1, which corresponds to a fixpoint, that is, to edges with Hamming length 5. It is clear that by removing the cycle length 1, we obtain the partitions (5) and (3,2), which correspond to the two possible types of derangements, i.e. fixpoint-free permutations, with 5 elements.

Table 1 shows all possible derangements of n elements, which are mutually not cycle-equivalent for $n = 1, \ldots, 7$. According to Proposition 7.1, their number is equal to $c(n,n) = p(n) - p(n-1)$, i.e. the difference between two consecutive terms of the partition function $p(n)$. The third column of Table 1 reports the possible partitions of n without 1-cycles and the last column shows a representative of the corresponding class of derangements.

The result that $H(6,5)$ has 6 classes of not cycle-equivalent edges can be easily derived also from Table 1, as there are 4 classes of derangements with $n = 6$ and 2 classes with $n = 5$. Even in the case of larger graphs like $H(7,4)$ and $H(7,5)$ the number of edge classes remains low: $c(7,4) = p(7) - p(3) = 12$ and $c(7,5) = p(7) - p(4) = 10$.

From Theorem 5.1 and Lemma 6.1, we know that cycle-equivalent edges of $H(n,d)$ generate isomorphic residual graphs. However, this property can not be generalized to pairs of not cycle-equivalent edges, even if they have the same Hamming length. Table 2 shows a list of counterexamples for $n = 4$, 5 and 6, which could be extended also to larger values of n. For each Hamming graph $H(n,d)$ a representative of each equivalence class of edges has been fixed and the corresponding residual graph R has been computed. The columns of Table 2 report the graph $H(n,d)$, the cycle lengths of the fixed edge and the order $|V_R|$, the number of edges $|E_R|$ and the clique number $\omega(R)$ of the residual graph R.

The smallest counterexample can be found in $H(4,4)$. This graph with 24 vertices has only 2 classes of cycle-equivalent edges: the edges with the cycle lengths (4)

Table 2 Residual graphs of $H(n, d)$ for different fixed 2-cliques ($n \leq 6$)

Graph	Fixed 2-clique	$\|V_R\|$	$\|E_R\|$	$\omega(R)$
$H(4, 4)$	(4)	2	1	2
	(2,2)	4	2	2
$H(5, 5)$	(5)	13	18	3
	(3,2)	12	12	3
$H(6, 4)$	(6)	608	169910	118
	(5,1)	613	172, 668	118
	(4,2)	608	169, 866	108
	(4,1,1)	616	174, 382	118
	(3,3)	610	171, 045	118
	(3,2,1)	610	170, 970	108
	(2,2,2)	608	169, 940	118
	(2,2,1,1)	618	175, 497	118
$H(6, 5)$	(6)	368	46, 781	16
	(4,2)	388	52, 338	16
	(3,3)	368	46, 816	16
	(2,2,2)	376	49, 149	16
	(5,1)	388	52, 392	16
	(3,2,1)	368	46, 760	16
$H(6, 6)$	(6)	80	731	4
	(4,2)	80	736	4
	(3,3)	82	774	4
	(2,2,2)	80	776	4

generate a residual graph with 2 vertices and a single edge, whereas the edges with the cycle lengths (2,2) produce a residual graph with 4 vertices and 2 edges. For all graphs reported in Table 2, the representatives of the edge classes generate residual graphs with not coinciding numbers of vertices and edges. This fact confirms that the residual graphs of not cycle-equivalent edges are in general not isomorphic. In the last column, the maximum value of the clique number $\omega(R)$ is $\omega(H(n, d)) - 2$, because the number of fixed vertices is $k = 2$. In most cases, the maximum value is reached; however, it is worth to notice that $H(6, 4)$ has 2 classes of edges with a suboptimal $\omega(R)$ value, which implies that a maximum clique of $H(6, 4)$ can not include any of the edges of these 2 classes. Such additional constraints may reduce significantly the complexity of the clique problem.

7.3 Computational Tests

The reduction approach described in Sect. 6 can now be applied to the cases $k = 1$ and $k = 2$ analysed in Sects. 7.1 and 7.2. The objective is to compare the running times of the original problem with the sum of the running times of the reduced instances. The algorithm for the generation of the reduced instances has been encoded in ANSI

C and the maximum clique problems have been solved by means of the clique solver MoMC (Mixed ordering MaxClique Solver) (see [13] for details). All tests reported in this section have been obtained on a computer equipped with an Intel Core *i*5 1.6 GHz processor and 8 GB of memory. Detailed results of the computational tests are available from the corresponding author upon request.

Table 3 shows the results of the computational experiments performed on 5 Hamming graphs $H(n, d)$ with $n = 6, 7$. The clique number of each graph is reported in square brackets under the name of the graph. The number of fixed vertices k assumes the values 0, 1 or 2. The case $k = 0$ corresponds to the full-size problem. For $k = 0, 1$ there is a unique clique problem to be solved, while for $k = 2$ the number of reduced problems is given by the value $c(n, d)$, according to Proposition 7.1. The following columns of the table report the minimum, respectively maximum order of the residual graphs $R(K_0)$, the minimum, respectively maximum CPU time (in sec) for the solution of a single reduced clique problem and the cumulative time for the solution of the entire set of reduced instances. The entry *tmax* indicates that the maximum allowed CPU time $tmax = 10800$ sec has been reached before closing the instance.

The collected data provide useful information about the reduction approach. First of all, the order of the residual graph shows that the reduction is more effective when the difference between n and d is small. For instance, 2 fixed nodes in $H(7, 7)$ lead to a 89% reduction of order, whereas in $H(6, 4)$ the gain is only 15%. The reason for this difference is the lower density of graphs, which allow only few Hamming distances. Usually, the graphs with the best reduction rates show also the best improvements in terms of CPU time. It is the case of $H(6, 5)$, which can not be solved within $tmax$ as a full-sized problem, but with 1 fixed node it becomes solvable in 479.26 sec and with 2 fixed nodes in only 48.71 sec. Similarly, $H(7, 7)$ requires 2 h 36 min as a full problem, but only 9.98 sec with $k = 1$ and 0.20 sec with $k = 2$.

On the other hand, in the case of $H(6, 4)$ it turns out that fixing one vertex is more effective than fixing two. However, both options provide a drastic save in CPU time. In the 2-clique case it is interesting to remark that the 2 instances with suboptimal clique numbers were considerably more time consuming than the others. For $H(7, 6)$ the best option is $k = 1$ solved in 310.68 sec. The $tmax$ entry for $k = 2$ is due to 4 instances, which could not be solved within the allowed time. The solutions of these instances are likely to be suboptimal because the lower bounds obtained are far from the optimum.

8 Conclusions

This paper proposes an approach to reduce the complexity of maximum clique problems associated with permutation Hamming graphs. The main contributions of the article are a classification of the edges based on their cycle representation and a better understanding of the automorphism group of permutation Hamming graphs. These results enable to infer the solution of the maximum clique problem associated with a permutation Hamming graph by solving a set of maximum clique problems smaller than the original one. Computational tests confirm that in most cases a speed up is obtained by applying the reduction approach.

Table 3 Running times of the reduction algorithm and of the full-size clique problem

Graph	k	# clique problems	$\lvert V_{R(K_0)}\rvert$ min	$\lvert V_{R(K_0)}\rvert$ max	clique_time min	clique_time max	Cumulative time
$H(6, 4)$	0	1	720	720	97.76	97.76	97.76
[120]	1	1	664	664	3.24	3.24	3.24
	2	8	608	618	0.73	4.93	15.05
$H(6, 5)$	0	1	720	720	tmax	tmax	tmax
[18]	1	1	529	529	479.26	479.26	479.26
	2	6	368	388	3.17	13.25	48.71
$H(6, 6)$	0	1	720	720	0.60	0.60	0.60
[6]	1	1	265	265	0.02	0.02	0.02
	2	4	80	96	0.01	0.01	0.04
$H(7, 6)$	0	1	5040	5040	832.85	832.85	832.85
[42]	1	1	3709	3709	310.68	310.68	310.68
	2	8	2608	2734	137.82	tmax	tmax
$H(7, 7)$	0	1	5040	5040	9359.68	9359.68	9359.68
[7]	1	1	1854	1854	9.98	9.98	9.98
	2	4	578	580	0.05	0.05	0.20

Acknowledgements The authors wish to thank Emanuele Delucchi for useful discussions on the automophism group of permutation Hamming graphs.

Funding Open access funding provided by SUPSI - University of Applied Sciences and Arts of Southern Switzerland

References

1. Andrews, G.E.: The Theory of Partitions. Cambridge University Press, Cambridge (1984)
2. Barta, J., Montemanni, R.: Hamming graphs and permutation codes. In: Proceedings of the 4th International Conference on Mathematics and Computers in Sciences and in Industry, pp. 154–158. IEEE Press (2017)
3. Barta, J., Montemanni, R.: Permutation Codes, Hamming Graphs and Turán Graphs. In: Proceedings of the 3rd International Conference on Applied Physics, System Science and Computers, Lecture Notes in Electrical Engineering, vol. 574, pp. 111–118. Springer Nature Switzerland (2019)
4. Barta, J., Montemanni, R., Smith, D.H.: A branch and bound approach to permutation codes. In: Proceedings of the IEEE 2nd International Conference of Information and Communication Technology—ICOICT, pp. 187–192 (2014)
5. Bereg, S., Levy, A., Sudborough, I.H.: Constructing permutation arrays from groups. Des. Codes Cryptogr. **86**(5), 1095–1111 (2018)
6. Bogaerts, M.: New upper bounds for the size of permutation codes via linear programming. Elect. J. Combin. **17**, #R135 (2010)
7. Chu, W., Colbourn, C.J., Dukes, P.: Constructions for permutation codes in powerline communications. Des. Codes Cryptogr. **32**, 51–64 (2004)
8. Colbourn, C.J., Kløve, T., Ling, A.C.H.: Permutation arrays for powerline communication and mutually orthogonal latin squares. IEEE Trans. Inf. Theory **50**(6), 1289–1291 (2004)
9. Deza, M., Vanstone, S.A.: Bounds for permutation arrays. J. Stat. Plan. Infer. **2**, 197–209 (1978)
10. El Rouayheb, S., Georghiades, C.N.: Graph theoretic methods in coding theory. In: Cohen, L., Poor, H., Scully, M. (eds.) Classical. Semi-classical and Quantum Noise. Springer, New York (2012)
11. Frankl, P., Deza, M.: On maximal numbers of permutations with given maximal or minimal distance. J. Combin. Theory Ser. A **22**, 352–360 (1977)
12. Han Vinck, A.J.: Coded modulation for power line communications. AEÜ Int. J. Elect. Commun. **54**(1), 45–49 (2000)
13. Li, C.M., Jiang, H., Manya, F.: On minimization of the number of branches in branch and bound algorithms for the maximum clique problem. Comput. Oper. Res. **84**, 1–15 (2017)
14. Montemanni, R., Barta, J., Smith, D.H.: The design of permutation codes via a specialized maximum clique algorithm. In: Proceedings of the 2nd International Conference on Mathematics and Computers in Science and in Industry, pp. 298–301. IEEE Press (2015)
15. 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)
16. Smith, D.H., Montemanni, R.: A new table of permutation codes. Des. Codes Cryptogr. **63**(2), 241–253 (2012)
17. Smith, D.H., Montemanni, R., Perkins, S.: The use of an exact algorithm within a tabu search maximum clique algorithm. Algorithms **13**(10), 253 (2020)
18. Tarnanen, H.: Upper bounds on permutation codes via linear programming. Eur. J. Comb. **20**, 101–114 (1999)