Permutation codes: a new upper bound for M(7,5)

Roberto Montemanni, János Barta, and Derek H. Smith

***Abstract*—**This paper deals with permutation codes. These codes have a main application in error correction in telecommunications. An algorithm based on combinatorial optimization concepts such as branch and bound, and graph theoretical concepts such as graph isomorphism, is discussed. The new theoretical result $M(7,5) \leq 122$, obtained by this approach, is finally disclosed.

***Index Terms*—**Branch and bound algorithms, Permutation codes, Upper bounds.

I. Introduction

Permutation codes have received remarkable attention in the literature [1], [2], [3], [4], [5], [6], [7]. This happened because of their application to powerline communications when M-ary Frequency-Shift Keying modulation is used [8], [9], [10], [11].

A permutation code is formally a set of permutations (codewords in our context) in the symmetric group S_n of all permutations on n elements. Parameter n defines the length of the code. The error correcting strength of a permutation code is proportional to the *minimum Hamming distance* of the code. The Hamming distance δ between two codewords is the number of elements that differ in the two permutations. The minimum distance d is defined as the minimum δ taken over all pairs of distinct permutations of the code. We will denote with (n,d) such a permutation code.

Redundancy in an encoding is minimized if the number of codewords is as large as possible. Thus if we denote with $M(n,d)$ the maximum number of codewords in an (n,d) permutation code, it is important to determine $M(n,d)$, or if this is not possible to produce good lower and upper bounds for this quantity. Many best-known lower bounds can be found is in [12]. Other results based on isomorphism arguments have been presented in [13] and [14], while in [15] a study on the structure of optimal codes has been presented.

In this paper the method originally discussed in [14] is described, and a novel theoretical result obtained by the method is disclosed.

II. Problem Description

A permutation of the n-tuple $x_0 = [0, 1, \dots, n-1] \in N^n$ is a *codeword* of length n and Ω_n denotes the set of all codewords of length n. Any subset Γ of Ω_n is a *permutation code* of length n. The main problem can now be stated as:

Definition 1. *Given a code length n and a Hamming distance d, the* maximum permutation code problem *consists of the determination of a code* $\Gamma \subseteq \Omega_n$ *with minimum distance d and the maximum possible number of codewords.*

Example 1. *The problem (6,5) is to determine a maximal code of length n = 6 with minimum distance d = 5. As reported in [14] the optimal solution of this problem is M(6,5) = 18. One of the many possible optimal (6, 5) codes is* Γ = {[012345], [021453], [034512], [045231], [102534], [130425], [153240], [205143], [243015], [251304], [310254], [324105], [341520], [425310], [432051], [450132], [504321], [513402]}.

III. A Branch and Bound Approach

In this section, the algorithm originally presented in [14] is sketched. The interested reader is referred to [14] for full details of the method.

A. Structure of the search-tree node

The search-subtree rooted at search-tree node t will be denoted as $SubT(t)$. Each node t is identified by the following elements:

- $in(t)$: a list of permutations that have to appear in all the solutions associated with $SubT(t)$;
- $feas(t)$: a list of permutations that are feasible according to the list of forced permutations $in(t)$, and to pruning and reduction explained in Sections III.D and III.E;
- $lb(t)$: a lower bound for the number of permutations in the optimal solutions associated with the search-tree nodes in $SubT(t)$. The calculation of the lower bound will be described in Section III.G;
- $ub(t)$: an upper bound for the number of permutations in the optimal solutions associated with the search-tree nodes in $SubT(t)$. The calculation of the upper bound will be described in Section III.F.

B. Initialization and branching strategy

Parameters *BestLB* and *BestUB*, containing initial lower and upper bounds for the problem are provided as input to the algorithm. These initial values will be updated during the execution of the algorithm with any improved values that should be found. A random permutation p is selected (all permutations are equivalent at this stage for symmetry considerations) and the root r of the search-tree is the node initialized with $in(r) = \{p\}$, $lb(r) = BestLB$, $ub(r) = BestUB$ and $feas(r) = \{i \in \Omega_n : \delta(i, p) \geq d\}$. Initially, r will be the only node contained in S, the dynamically updated set of active search-tree nodes to be examined ($S = \{r\}$), referred to as *open nodes* from now on. The set of *closed nodes* C is initially empty. This set will contain the search-tree nodes that have already been processed by the algorithm, and will be used by pruning and reduction techniques described in Section III.E.

An open node t from the set S is expanded at each iteration of the algorithm (details will be disclosed in Section III.C). Technically, node t is decomposed into the associated

Manuscript received October 18, 2014.

R. Montemanni and J. Barta are with the University of Applied Sciences of Southern Switzerland (SUPSI), 6928 Manno, Switzerland (e-mail: {roberto.montemanni, janos.barta}@supsi.ch).

D.H. Smith is with the University of South Wales, Pontypridd CF371DL, United Kingdom (e-mail: derek.smith@southwales.ac.uk).

http://dx.doi.org/10.15242/IAE.IAE1214003

subproblems as follows. One new search-tree node t_p is created for each permutation $p \in feas(t)$ in such a way that $in(t_p) = in(t) \cup \{p\}$ and the new set $feas(t_p)$ is determined consequently, also taking into account the reduction and pruning rules described in Section III.D. Sets *S* and *C* are finally updated: $S = S\backslash\{t\}$, $C = C \cup \{t\}$. For each new node t_p the values of $lb(t_p)$ and $ub(t_p)$ are calculated as described in Sections III.F and III.G. In the case that the pruning test (see the appropriate subsection) is positive, the new node t_p is pruned and $C = C \cup \{t_p\}$, otherwise $S = S \cup \{t_p\}$.

If $\min_{t\in S}(ub(t)) \leq BestUB$, it means that the global upper bound of the residual open problems has been improved, and *BestUB* can be updated to $\min_{t\in S}(ub(t))$. The value of *BestLB* is updated each time an improving incumbent heuristic solution is found. Notice that when *BestLB* is updated, all the open nodes $u \in S$ are inspected and pruned in case $ub(u) \leq BestLB$. Formally, $S = S\backslash\{u\}$ and $C = C \cup \{u\}$ in such a case.

The exit criterion for the branch and bound algorithm is based on the cardinality of set S: when $|S| = 0$ the computation stops.

C. Selection of the search-tree node to expand

Nodes are expanded in a depth-first fashion, with nodes at a same level of the search-tree visited in the same order they have been created. This component is different from the equivalent one of the method originally proposed in [14], and it has been changed to increase the overall efficiency of the algorithm.

D. Pruning strategy

Some rules useful to identify and prune dominated search-tree nodes while generating a new search-tree node *t* are described. They are based on the concept of *isomorphism* for graphs [16]. Two graphs $G = \{V_G, E_G\}$ and $H = \{V_H, E_H\}$ are said to be *isomorphic* if a bijection $f: V_G \rightarrow V_H$ exists, such that $(i,j) \in E_G$ if and only if $(f(i), f(j)) \in E_H$. In this case we will write $G \cong H$

Definition 2. *The* graph induced *by a search-tree node* t *is defined as* $G_t^I = \{V_t^I, E_t^I\}$, *with* $V_t^I = \{i \in \Omega_n \mid \delta(i,j) \geq d \; \forall j \in in(t)\}$[1] *and* $E_t^I = \{\{i,j\} | i,j \in V_t^I, \delta(i,j) \geq d\}$.

The following definition is at the basis of the pruning technique.

Definition 3. *If the graph* G_t^I *induced by the search-tree node t is isomorphic to the graph* G_u^I *induced by the search-tree node u, with* $|in(u)| = |in(v)|$, *we will say that node t is isomorphic to node u and write* $t \cong u$.

The next result allows one of two isomorphic nodes to be pruned from the branch and bound tree.

Theorem 1 (Montemanni et al. [14])**.** *If a new search-tree node t is such that* $t \cong u$ *with* $u \in S \cup C$, $|in(t)| = |in(u)|$, *then the node t can be classified as* dominated *and moved to set C* ($S = S\backslash\{t\}$ *and* $C = C \cup \{t\}$).

[1] The vertex set V_t^I was erroneously defined as $V_{in(t)}^{old} = \Omega_n \setminus in(t)$ in [14] instead of the correct definition reported here. The correct definition had been however used in [14] to derive the theoretical results and everywhere in the implementation.

E. Reduction strategy

Some rules useful to reduce the size of *feas*(*t*) while generating a new search-tree node *t* are discussed. During the branching of a node *t*, all potential new nodes obtained by expanding the set *feas*(*t*) with each possible permutation are considered, as described before.

Proposition 1 (Montemanni et al. [14])**.** *While creating a new search-tree node u obtained by adding* $p_u \in feas(t)$ *into in*(*u*), *a permutation* $p_k \in feas(t)$, *such that* $k \cong v$, $v \in S \cup C$, $|in(k)| = |in(v)|$, *can be taken out of feas*(*u*) ($feas(u)\backslash\{p_k\}$).

F. Upper bound

The set of codewords Ω_n can be split into *n* subsets $W_0, W_1, \ldots, W_{n-1}$ in such a way that for a fixed value $k \in \{0,1,\ldots,n-1\}$ the subset W_i is defined as $W_i = \{x \in \Omega_n : x(k) = i\}$. Equivalently, the subset W_i contains all codewords with the *k*-th component having the value *i*. Since the partition is obtained by fixing the value of one component of the codewords, it is clear that the sets $W_i s$ are isomorphic to Ω_{n-1}. Furthermore, as the sets W_is form a partition of Ω_n, it is well-known that an upper bound of *M(n,d)* can be obtained by adding the upper bounds on the subsets W_i:

Theorem 2 (Deza and Vanstone [18]).

$$M(n,d) \leq n \cdot M(n-1,d) \quad (1)$$

The partitioning procedure described in Theorem 2 can be carried out on any subset of Ω_n. At each search-tree node *t* the algorithm generates a partition $T_0, T_1, \ldots, T_{n-1}$ of the set *feas*(*t*), such that $T_i = \{x \in feas(t) : x(k) = i\}$ and a partition $Q_0, Q_1, \ldots, Q_{n-1}$ of the set *in*(*t*), such that $Q_i = \{x \in in(t) : x(k) = i\}$. For each subset T_i an upper bound $UB(T_i)$ is calculated using the *maximum clique problem* solver proposed in [19] (see also [20]), which is run for 30 seconds on every subproblem.

The following result describes an upper bound obtained by specializing the result of Theorem 2 to the subproblem associated with a search-tree node *t*.

Proposition 2 (Montemanni et al. [14]).

$$\sum_{i=0}^{n-1} |Q_i| + \min\{UB(T_i); M(n-1,d) - |Q_i|\} \quad (2)$$

is a valid upper bound for the search-subtree rooted at the search-tree node t.

G. Lower bound

The lower bound originally presented in [14] can be used to have a full exact method. However we observe that for the novel theoretical results reported in Section IV, the branch and bound method will be used merely as a checker for the lower bound *LBbest* initially passed as a parameter to the method, with no need for a lower bound procedure within the algorithm.

IV. A New Theoretical Result: $M(7,5) \leq 122$

In the study presented in this paper, subproblems of (7,5) are considered, where a position of the permutations is restricted to values from a given set $F \subset \{0,1,\ldots,n-1\}$.

Definition 4. *We refer to the problem* $(n,d)|_{|F|}$ *as the subproblem of* (*n*,*d*) *where a position of the code is restricted to values from a given set* $F \subset \{0,1,\ldots,n-1\}$.

Notice that only the cardinality of F is of interest, since different sets of values with the same cardinality generate equivalent problems due to symmetry.

From previous studies (e.g. [14], [15]) it is known that $M(7, 5)|_1 = 18$ and $M(7, 5)|_2 = 36$. In this work we will focus on $M(7, 5)|_3$, aiming at improving the known upper bound of 53.

Proposition 3.

$$M(7,5)|_3 \leq 52 \quad (3)$$

Proof: The result was proven by the algorithm described in this paper, with an initial lower bound *LBbest* = 52. The algorithm was coded in ANSI C, *Nauty 2.5* [17] was used to identify isomorphisms, and the executable was run on a Dual AMD Opteron 250 2.4GHz/4GB RAM computer (only one core has been used at a time). A total of 4 832 search-tree nodes have been visited in approximately 33 days of computation to close the problem. ■

The result of Proposition 3 has an interesting implication that leads to the most important result of this paper. We first need the following result.

Proposition 4.

$$If\ \sum_{i=1}^{z}|F_i| = n\ then\ M(n,d) \leq \sum_{i=1}^{z} M(n,d)|_{|F_i|}$$

Proof: The result is a trivial generalization of that of Theorem 2. ■

Proposition 5.

$$M(7,5) \leq 122 \quad (4)$$

Proof: The bound is easily obtained by using the result of Proposition 3 inside the inequality of Proposition 4. With $|F_1| = |F_2| = 3$ and $|F_3| = 1$ we have $M(7,5) \leq 52 + 52 + 18 = 122$ ■

V. Conclusion

A branch and bound approach for permutation codes that has been recently appeared in the literature has been summarised, and a previously unknown theoretical result for the permutation code (7,5), obtained with this method, has been disclosed.

References

[1] I.F. Blake, *Permutation codes for discrete channels*, IEEE Transactions on Information Theory 20(1), 138–140, 1974. http://dx.doi.org/10.1109/TIT.1974.1055142

[2] M. Bogaerts, *New upper bounds for the size of permutation codes via linear programming*, The Electronic Journal of Combinatorics 17(#R135), 2010.

[3] W. Chu, C.J. Colbourn and P. Dukes, *Constructions for permutation codes in powerline communications*, Designs, Codes and Cryptography 32, 51–64, 2004. http://dx.doi.org/10.1023/B:DESI.0000029212.52214.71

[4] P. Dukes and N. Sawchuck, *Bounds on permutation codes of distance four*, Journal of Algebraic Combinatorics 31, 143–158, 2010. http://dx.doi.org/10.1007/s10801-009-0191-2

[5] P. Frankl and M. Deza, *On maximal numbers of permutations with given maximal or minimal distance*, Journal of Combinatorial Theory Series A 22, 352–260, 1977. http://dx.doi.org/10.1016/0097-3165(77)90009-7

[6] I. Janiszczak and R. Staszewski, *An improved bound for permutation arrays of length 10*, http://www.iem.uni-due.de/preprints/IJRS.pdf (accessed October 16th 2014).

[7] H. Tarnanen, *Upper bounds on permutation codes via linear programming*, European Journal of Combinatorics 20, 101–114, 1999. http://dx.doi.org/10.1006/eujc.1998.0272

[8] C.J. Colbourn, T. Kløve and A.C.H. Ling, *Permutation arrays for powerline communication and mutually orthogonal latin squares*, IEEE Transactions on Information Theory 50, 1289–1291, 2004. http://dx.doi.org/10.1109/TIT.2004.828150

[9] A.J. Han Vinck, *Coded modulation for power line communications*, A.E.Ü. International Journal Electronics and Communications 54(1), 45– 49, 2000.

[10] S. Huczynska, *Powerline communications and the 36 officers problem*, Philosophical Transactions of the Royal Socicety A. 364, 3199–3214, 2006. http://dx.doi.org/10.1098/rsta.2006.1885

[11] N. Pavlidou, A.J. Han Vinck, J. Yazdani and B. Honary, *Powerline communications: state of the art and future trends*, IEEE Communications Magazine 41(4), 34–40, 2003. http://dx.doi.org/10.1109/MCOM.2003.1193972

[12] D.H. Smith and R. Montemanni, *A new table of permutation codes*, Design, Codes and Cryptography 63(2), 241–253, 2012. http://dx.doi.org/10.1007/s10623-011-9551-8

[13] I. Janiszczak, W. Lempken, P.R.J. Östergård and R. Staszewski, *Permutation codes invariant under isometries*, Designs, Codes and Cryptography, to appear. http://dx.doi.org/10.1007/s10623-014-9930-z

[14] R. Montemanni, J. Barta and D.H. Smith, *Permutation codes: a branch and bound approach*, Proceedings of the International Conference on Pure Mathematics, Applied Mathematics, Computational Methods – PMAMCM, 86-90, 2014.

[15] J. Barta, R. Montemanni and D.H. Smith, *A branch and bound approach to permutation codes*, Proceedings of the IEEE Second International Conference of Information and Communication Technology - ICOICT, 187–192, 2014.

[16] B.D. McKay, *Practical graph isomorphism*, Congressum Numerantium 30, 45–87, 1981.

[17] B.D. McKay and A. Piperno, *Practical graph isomorphism, II*, Journal of Symbolic Computation 60, 94–112, 2014. http://dx.doi.org/10.1016/j.jsc.2013.09.003

[18] M. Deza and S.A. Vanstone, *Bounds for permutation arrays*, Journal of Statistical Planning and Inference 2, 197–209, 1978. http://dx.doi.org/10.1016/0378-3758(78)90008-3

[19] P.R.J. Östergård, *A fast algorithm for the maximum clique problem*, Discrete Applied Mathematics 120, 197–207, 2002. http://dx.doi.org/10.1016/S0166-218X(01)00290-6

[20] P.R.J. Östergård, *A new algorithm for the maximum-weight clique problem*, Nordic Journal of Computing 8(4), 424–436, 2001.

Roberto Montemanni is Professor of Advanced Algorithms at the University of Applied Sciences of Southern Switzerland (SUPSI), where he works since 2001. He is active as a Senior Researcher at the Dalle Molle Institute for Artificial Intelligence (IDSIA), Switzerland.

He obtained a Laurea degree in Computer Science from the University of Bologna, Italy in 1999 and a Ph.D. in Applied Mathematics from the University of Glamorgan, U.K. in 2002.

His main research interests are in mathematical programming modeling and heuristic methods for optimization problems arising in telecommunications, transportations and coding theory.

János Barta studied mathematics at the Swiss Federal Institute of Technology of Zurich (ETHZ).

From 1998 to 2003 he worked as a researcher and lecturer at the Zurich University of Applied Sciences (ZHAW). Since 2003 he is a lecturer at the University of Applied Sciences of Southern Switzerland (SUPSI) and since 2004 is an associate researcher with the Dalle Molle Institute for Artificial Intelligence (IDSIA).

His main research interests are combinatorial and robust optimization.

Derek H. Smith is Emeritus Professor of Mathematics at the University of South Wales (formerly the University of Glamorgan), where he worked from 1971 to 2014.

He has B.Sc. and Ph.D. degrees from the University of Southampton, U.K. and a D.Sc. degree from the University of Glamorgan.

His research interests include radio frequency assignment, graph theory, data compression, network reliability studies, coding theory and biological applications of coding theory.