American Journal of Computer Science and Technology
2018; 1(2): 39-43
http://www.sciencepublishinggroup.com/j/ajcst
doi: 10.11648/j.ajcst.20180102.11

A New Powerful Scheme Based on Self Invertible Stabilizer Multiplier Permutation to Find the Minimum Distance for large BCH Codes

Issam Abderrahman Joundan, Said Nouh, Abdelwahed Namir

Faculty of Sciences Ben M'sik, Hassan II University, Casablanca, Morocco

Email address:
joundan.fsb@gmail.com (I. A. Joundan), said.nouh@univh2m.ma (S. Nouh), a.namir@yahoo.fr (A. Namir)

To cite this article:
Issam Abderrahman Joundan, Said Nouh, Abdelwahed Namir. A New Powerful Scheme Based on Self Invertible Stabilizer Multiplier Permutation to Find the Minimum Distance for large BCH Codes. *American Journal of Computer Science and Technology*. Vol. 1, No. 2, 2018, pp. 39-43. doi: 10.11648/j.ajcst.20180102.11

Received: January 7, 2018; **Accepted**: January 17, 2018; **Published**: February 21, 2018

Abstract: In this paper, we present the powerful scheme ZSISMP (Zimmermann Self Invertible Stabilizer Multiplier Permutation) to attack the hardness of the minimum distance search problem of BCH codes. This scheme consists in evaluating the minimum distance of the reduced dimension sub code fixed by a Self Invertible Stabilizer Multiplier Permutation by Zimmermann algorithm. The proposed scheme ZSISMP is validated on all BCH codes of known minimum distance. A comparison with several known powerful techniques proves its efficiency in giving more accurate results in short time. The use of this efficient local search had yield to determine the error correcting capability of many BCH codes of length 1023 and 4095.

Keywords: Minimum Distance, Minimum Weight, BCH Codes, Designed Distance, Automorphism Group, Multiplier, Zimmermann Algorithm

1. Introduction

In telecommunication and storage systems, the fundamental problem is the reproduction at one point exactly or approximately the selected data at another point. An efficient solution of this problem is the use of error correcting codes. The error correcting codes improve the reliability of such communication, notably on channels that are subject to noise, by adding redundancy in data.

BCH codes are a family of cyclic codes, which are used in many applications, due to their powerful algebraic decoding algorithms and their error-correcting capability. The error-correcting capability of these codes is directly related to their minimum distance. However, the determination of this metric is difficult in general as pointed out by Charpin in [1] and remains an open problem in coding theory. For these codes, only a lower bound is known and the minimum distance is known only for some lengths and special cases [2-3-4-5-6-7-8]. In this paper, our work will focused on finding the minimum distance of large BCH codes.

The remainder of this paper is organized as follows: The next section presents the main related works. The section 3 presents the proposed scheme ZSISMP. The section 4 presents the main results. The conclusion and possible future directions of this research are outlined in section 5.

2. Related Works

Determining the minimum distance of BCH codes is an important, but difficult, problem. For these codes, only a lower bound is known but the true value is still unknown for large codes. For this reason, many researchers have explored several ways to attack the difficulty of the minimum distance search problem for large BCH Codes. This section summarizes the most important ones.

In [9], Augot, Charpin, and Sendrier presented an algebraic system constructed from Newton's identities. The existence of solutions to this system is a necessary condition to the existence of codewords of weight w in the code. The use of this method has finished the table of BCH codes of length 255. In [10], Augot and Sendrier found idempotent codewords of minimum weight for several primitive narrow-sense BCH codes.

In [11], Canteaut and Chabaud have developed a new probabilistic algorithm, based on the heuristic proposed by Stern [12], for finding minimum-weight words in a linear code.

The application of the proposed algorithm on narrow-sense BCH Codes of length 511 has determined the true minimum distance of some codes, however the table of BCH codes of length 511 is still open.

Zimmermann algorithm [13] is a general algorithm for computing the minimum distance of a linear code. It is implemented in GAP (package Guava) [14] over fields F_2 and F_3. It is also implemented, in Magma over any finite field. Zimmermann's algorithm is explained in detail in [15].

The artificial intelligence Simulated Annealing presented in [16], Tabu Search [17], Hill-Climbing [18], Genetic Algorithm [18-19], Ant Colony Optimization [20], Metropolis Algorithm [21], was shown to be useful to attack the difficulty of the minimum distance search problem for BCH Codes. In [22], Aylaj and Belkasmi improve the classical Simulated Annealing presented in [16]. This improvement has yield to a fast convergence of the Simulated Annealing by reducing the number of iterations, as well as obtaining good results in comparison with the previous works presented in [17-18-19-21].

Unlike classical techniques based on exhaustive or partial enumeration of codewords, Berrou in [23] has presented an efficient approach based on the notion of Error Impulse response of a Soft-In decoder. This approach consists in adding to all-zero codeword a level of noise and considering the minimum distance as the smallest level of noise from which the Soft-In decoder fails in correction. The authors in [19], continue to improve this method, by injecting errors in many positions and proposed the Multiple Impulse Method. In [24], the authors proposed an efficient method, by applying the MIM method on some sub code randomly extracted from the considerer BCH code. The proposed method MIM-RSC, has allowed an efficient local search and therefore finding the true minimum distance of some BCH codes of length 1023 and 2047 as well as obtaining good results in comparison with the previous works presented in [17-18-19-20-21-22].

3. The Proposed Scheme

It is well known that for BCH ($n=2^m-1$, δ) codes, the multiplier permutations defined on $\{0, 1,..., n-1\}$ by μ_{2^k} : $i \rightarrow 2^k i \pmod n$ with $1 \leq k \leq m-1$ are stabilizers. From these stabilizers, we take only a Self Invertible stabilizer if it exist and by using a mathematical tool, we find the sub code fixed by this involution and then we evaluate the minimum distance by using the famous Zimmermann algorithm. This section presents the proposed scheme for finding the lowest weight in BCH codes.

For finding the minimum distance of BCH codes. The proposed scheme works as follows:

Inputs:

- A generator matrix G of BCH ($n=2^m-1$, k, δ)
- -A self-invertible permutation σ from $\{\mu_2^i, 1 \leq i \leq m\}$

Begin

$d \leftarrow n-k+1$

Step 1:

Find the sub code SC fixed by σ

Step 2:

Find the estimated minimum distance d of SC by using the Zimmermann algorithm.

Output:

- d as estimated minimum distance of BCH (n, k, δ)

4. Results and Discussions

This section presents a validation of the proposed method on BCH codes of known minimum distance and its application for finding the minimum distance of BCH codes of unknown minimum distance. This section presents also a comparison between the proposed scheme and previous work on minimum distance for BCH codes.

All results have been done using a simple configuration machine: Intel (R) Core (TM) i3-4005U CPU @1.70GHz RAM 4GO and are made by running the considered algorithm in 1day for each code.

4.1. Validation of the Proposed Scheme

It is well known that the smallest primitive narrow-sense BCH code whose minimum distance is greater than its designed distance is BCH (127, 43, 29) and it is the only one for this length. It is known also, that All the narrow-sense primitive binary BCH codes of length 255 have their minimum distance equal to their designed distance except BCH (255, 63, 61), and BCH (255, 71, 59). The both last result have been proved in [9], by using the Newton's identities.

In order to validate the proposed method, it is applied on all BCH codes of known minimum distance presented in table 1. the obtained results show that the minimum weight found by the proposed method is equal to the true value of the minimum distance of all BCH codes of length up to 255. Therefore, the proposed method is validated for BCH codes of lengths up to 255.

4.2. Comparison of the Proposed Scheme with Zimmermann Algorithm

A comparison between the proposed scheme with Zimmermann algorithm, on some BCH codes are made. The table 2 summarizes the obtained results. These results demonstrate that the proposed scheme outperform greatly the famous Zimmermann algorithm.

4.3. Comparison of the Proposed Scheme with MIM-RSC Method

The table 3 presents a comparison between the proposed scheme and MIM-RSC method [24]. It shows that the proposed scheme greatly passes the MIM-RSC method.

4.4. Results of the Proposed Scheme for Some Large BCH Codes

In order to find the minimum distance of some large BCH codes, the proposed scheme is applied by using a simple machine of the configuration given above. The obtained

results are given in the table 4 so that d_f represent the minimum distance found by our scheme. This table shows the height capacity of the proposed technique to find a minimum weight codeword.

Table 1. *Validation of the proposed scheme for BCH codes of length up to 255.*

BCH (n, k, δ)	True value of minimum distance	d (ZSISMP)	BCH (n, k, δ)	True value of minimum distance	d (ZSISMP)
BCH (15, 11, 3)	3	3	BCH (255, 171, 23)	23	23
BCH (15, 7, 5)	5	5	BCH (255, 163, 25)	25	25
BCH (15, 5, 7)	7	7	BCH (255, 155, 27)	27	27
BCH (63, 57, 3)	3	3	BCH (255, 147, 29)	29	29
BCH (63, 51, 5)	5	5	BCH (255, 139, 31)	31	31
BCH (63, 45, 7)	7	7	BCH (255, 131, 37)	37	37
BCH (63, 39, 9)	9	9	BCH (255, 123, 39)	39	39
BCH (63, 36, 11)	11	11	BCH (255, 115, 43)	43	43
BCH (63, 24, 15)	15	15	BCH (255, 107, 45)	45	45
BCH (63, 18, 21)	21	21	BCH (255, 99, 47)	47	47
BCH (63, 16, 23)	23	23	BCH (255, 91, 51)	51	51
BCH (63, 10, 27)	27	27	BCH (255, 87, 53)	53	53
BCH (63, 7, 31)	31	31	BCH (255, 79, 55)	55	55
BCH (255, 247, 3)	3	3	BCH (255, 71, 59)*	61	61
BCH (255, 239, 5)	5	5	BCH (255, 63, 61)*	63	63
BCH (255, 231, 7)	7	7	BCH (255, 55, 63)	63	63
BCH (255, 223, 9)	9	9	BCH (255, 47, 85)	85	85
BCH (255, 215, 11)	11	11	BCH (255, 45, 87)	87	87
BCH (255, 207, 13)	13	13	BCH (255, 37, 91)	91	91
BCH (255, 199, 15)	15	15	BCH (255, 29, 95)	95	95
BCH (255, 191, 17)	17	17	BCH (255, 21, 111)	111	111
BCH (255, 187, 19)	19	19	BCH (255, 13, 119)	119	119
BCH (255, 179, 21)	21	21	BCH (255, 9, 127)	127	127

Table 2. *Comparison between the proposed scheme and Zimmermann algorithm for some BCH codes of length 1023.*

BCH (n, k, δ)	d (Zimmermann)	Run Time of Zimmermann (s)	d (ZSISMP)	Total Run Time of ZSISMP (s)
BCH (1023, 848, 37)	47	9198.411	37	474.001
BCH (1023, 838, 39)	51	8951.867	39	705.476
BCH (1023, 828, 41)	55	5030.891	41	7347.036
BCH (1023, 818, 43)	61	7723.827	43	913.925
BCH (1023, 808, 45)	63	5582.260	45	46246.594
BCH (1023, 798, 47)	67	21379.687	47	61864.161

Table 3. *Comparison between the proposed scheme and MIM-RSC method for some BCH codes of lengths 1023.*

BCH (n, k, δ)	d (MIM-RSC)	d (ZSISMP)	BCH (n, k, δ)	d (MIM-RSC)	d (ZSISMP)
BCH (1023, 1013, 3)	3	3	BCH (1023, 953, 15)	15	15
BCH (1023, 1003, 5)	5	5	BCH (1023, 943, 17)	17	17
BCH (1023, 993, 7)	7	7	BCH (1023, 933, 19)	21	19
BCH (1023, 983, 9)	9	9	BCH (1023, 923, 21)	25	21
BCH (1023, 973, 11)	11	11	BCH (1023, 913, 23)	29	23
BCH (1023, 963, 13)	13	13			

Table 4. *True minimum weights of some BCH codes of length 1023, 4095 found by the proposed Scheme.*

BCH (n, k, δ)	d_f	RUN TIME OF STEP1 (S)	Run Time of Step2 (s)	Total Run Time (s)
BCH (1023, 1013, 3)	3	47	0.405	47.405
BCH (1023, 1003, 5)	5	32	0.561	32.561
BCH (1023, 993, 7)	7	33	0.701	33.701
BCH (1023, 983, 9)	9	34	0.780	34.780
BCH (1023, 973, 11)	11	44	0.889	44.889
BCH (1023, 963, 13)	13	38	0.748	38.748
BCH (1023, 953, 15)	15	38	0.826	38.826
BCH (1023, 943, 17)	17	40	0.811	40.811
BCH (1023, 933, 19)	19	43	0.920	43.920
BCH (1023, 923, 21)	21	46	0.935	46.935
BCH (1023, 913, 23)	23	46	0.935	46.935
BCH (1023, 903, 25)	25	46	0.982	46.982
BCH (1023, 893, 27)	27	47	1.232	48.232
BCH (1023, 883, 29)	29	51	1.809	52.809
BCH (1023, 873, 31)	31	56	1.107	57.107
BCH (1023, 863, 33)	33	59	13.368	72.368

BCH (n, k, δ)	d_f	RUN TIME OF STEP1 (S)	Run Time of Step2 (s)	Total Run Time (s)
BCH (1023, 858, 35)	35	57	26.941	83.941
BCH (1023, 848, 37)	37	64	410.001	474.001
BCH (1023, 838, 39)	39	55	650.476	705.476
BCH (1023, 828, 41)	41	77	7270.036	7347.036
BCH (1023, 818, 43)	43	75	838.925	913.925
BCH (1023, 808, 45)	45	72	46174.594	46246.594
BCH (1023, 798, 47)	47	78	61786.161	61864.161
BCH (4095, 4083, 3)	3	381	11.902	392.902
BCH (4095, 4071, 5)	5	308	13.993	321.993
BCH (4095, 4059, 7)	7	350	16.207	366.207
BCH (4095, 4047, 9)	9	472	19.140	491.140
BCH (4095, 4035, 11)	11	474	21.028	495.028
BCH (4095, 4023, 13)	13	516	21.761	537.761
BCH (4095, 4011, 15)	15	633	23.431	656.431
BCH (4095, 3999, 17)	17	548	24.616	572.616
BCH (4095, 3987, 19)	19	629	27.486	656.486
BCH (4095, 3975, 21)	21	732	29.764	761.764

5. Conclusion and Perspectives

In this paper, we have proposed a new efficient scheme to find the minimum distance for large BCH codes. The experimental results show that the proposed scheme outperforms several known powerful techniques. In the perspectives, we will apply this powerful scheme to construct good large cyclic codes, and adapt this scheme for other linear codes.

References

[1] P. Charpin, Open problems on cyclic codes, in: V. S. Pless, W. C. Human, R. A. Brualdi (Eds.), Handbook of Coding Theory, Part 1: Algebraic Coding, Elsevier, Amsterdam, The Netherlands, 1998 (Chapter 11).

[2] C. Ding, X. Du, Z. Zhou, "The Bose and minimum distance of a class of BCH codes," IEEE Trans. Inf. Theory 61 (5) (2015) 2351–2356.

[3] C. Ding, "Parameters of several classes of BCH codes," IEEE Trans. Inf. Theory 61 (10) (2015) 5322–5330.

[4] Rajesh Kumar Narula, O. P. Vinocha and Ajay Kumar, "A class of non-binary BCH code of Bose and minimum distance." Journal of Mathematical and Computational Science (2016), No. 6, 1169-1176.

[5] Cunsheng Ding, Cuiling Fan, Zhengchun Zhou "The dimension and minimum distance of two classes of primitive BCH codes" *Finite Fields and Their Applications*, Volume 45, May 2017, Pages 237-263.

[6] Hao Liu, Cunsheng Ding, Chengju Li, "Dimensions of three types of BCH codes over GF (q)", *Discrete Mathematics, 340 (2017) 1910–1927*.

[7] Shuxing Li "The Minimum Distance of Some Narrow-Sense Primitive BCH Codes". SIAM Journal on Discrete Mathematics, 2017, Vol. 31, No. 4: pp. 2530-2569.

[8] Shuxing Li, Cunsheng Ding, Maosheng Xiong, and Gennian Ge, "Narrow-Sense BCH Codes Over GF (q) With Length $n=q^m-1/q-1$," IEEE Transactions on Information Theory (Volume: 63, Issue: 11, Nov. 2017).

[9] Daniel Augot, Pascale Charpin, and Nicolas Sendrier "Studying the Locator Polynomials of Minimum Weight Codewords of BCH Codes" *IEEE TRANSACTIONS ON INFORMATION THEORY*, VOL 38, NO. 3, MAY 1992.

[10] D. Augot and N. Sendrier, "Idempotents and the BCH bound," *IEEE Trans. Inform. Theory*, vol. 40, pp. 204–207, 1994.

[11] A. Canteaut, and F. Chabaud. "A new algorithm for finding minimum-weight words in a linear code: Application to primitive narrow-sense BCH codes of length 511", *IEEE Trans. Inform. Theory*, IT-44 (1), pp 367-378, Jan. 1998.

[12] J. Stern, "A method for finding codewords of small weight," in Coding Theory and Applications, G. Cohen and J. Wolfmann, Eds. New York: Springer-Verlag, 1989, pp. 106–113.

[13] Zimmermann K.-H., Integral Hecke Modules, Integral Generalized Reed-Muller Codes, *and Linear Codes* Technische Universitat HamburgHarburg, Tech. Rep. 3-96, 1996.

[14] The GAP Group. "GAP–Groups, Algorithms, and Programming, Version 4.7.9". 2015. http://www.gap-system.org.

[15] Grassl, M. Searching for linear codes with large minimum distance. Discovering mathematics with Magma, pp. 287–313, Algorithms Comput. Math., 19, Springer, Berlin, 2006.

[16] M. Zhang, F. Ma, Simulated annealing approach to the minimum distance of error-correcting codes, Int. J. Electron. 76 (1994) 377–384.

[17] J. A. Bland, D. J. Baylis, A tabu search approach to the minimum distance of error-correcting codes, Int. J. Electron. 79 (1995) 829–837.

[18] J. Wallis and K. Houghten, "A Comparative Study of Search Techniques Applied tothe Minumum Distance of BCH Codes," *Conference on Artificial Intelligence and Soft Computing*, Banff, 17-19 July 2002.

[19] Askali M., Azouaoui A., Nouh S., Belkasmi M. (2012) On the Computing of the Minimum Distance of Linear Block Codes by Heuristic Methods, *International Journal of Communications, Network and System Sciences*, 5 (11), 774-784.

[20] J. A. Bland. Local search optimisation applied to the minimum distance problem. *Advanced Engineering Informatics*, 21, 2007.

[21] Ajitha Shenoy K. B, Somenath Biswas, Piyush P. Kurur, "Performance of metropolis algorithm for the minimum weight code word problem", Genetic and Evolutionary Computation Conference, 2014.

[22] Bouchaib AYLAJ and Mostafa BELKASMI, "New Simulated Annealing Algorithm for Computing the Minimum Distance of Linear Block Codes". *Advances in Computational Research*, indexed Google Scholar ISSN: 0975-3273, E-ISSN: 0975-9085, Volume 6, Issue 1, pp.-153-158, 2014.

[23] C. Berrou, S. Vaton, M. Jezequel and C. Douillard, "Computing the Minimum Distance of Linear Codes by the Error Impulse Method," Proceedings of IEEE Globecom, Taipei, 17-21 November 2002, pp. 10-14.

[24] S. NOUH, I. A. Joundan, B. Aylaj, M. Belkasmi, A. Namir "New Efficient Scheme Based on Reduction of the Dimension in the Multiple Impulse Method to Find the Minimum Distance of Linear Codes", International Review on Computers and Software IRECOS, Vol 11, No 9 (2016) pages 742-751.

[25] W. C. Huffman and V. Pless. Fundamentals of error-correcting codes. Cambridge University Press, Cambridge, 2003.

[26] R. C Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3:68–79, mars 1960.

[27] A. Hocquenghem. Codes correcteurs d'erreurs. Chiffres, 2:147–156, sept 1959.