International Symposium on Information Theory and its Applications, ISITA2008
Auckland, New Zealand, 7-10, December, 2008

An Efficient SAT Encoding of Circuit Codes

Yury CHEBIRYAK† and Daniel KROENING‡

† Computer Systems Institute
ETH Zurich
8092 Zurich, Switzerland
E-mail: yury.chebiryak@inf.ethz.ch

‡ Computing Laboratory
Oxford University
Oxford, OX1 3QD, UK
E-mail: kroening@comlab.ox.ac.uk

Abstract

Circuit codes in hypercubes are generalized snake-in-the-box codes and are used in analog-to-digital conversion devices. The construction of the longest known circuit codes is based on either an exhaustive search or an algorithm that restricts the search to the codes with periodic coordinate sequences. In this paper, we describe an efficient SAT encoding of circuit codes, which enabled us to obtain new circuit codes.

1. INTRODUCTION

In 1958, W. H. Kautz brought attention to the *snake-in-the-box* problem—finding a binary code that has unit distance between adjacent code words and minimum distance two between all other code words [14]. The search for snakes is motivated by the theory of error-correcting codes (as the vertices of a solution to the snake or coil in the box problems can be used as a Gray code that can detect single-bit errors), electrical engineering, computer network topologies [1], Systems Biology [9], etc. Approaches to find long snakes range from studies of mathematical constructions (e.g. binary necklaces [17]) and certain patterns in lower dimensions [20, 19] to genetic algorithms [1, 2, 22].

R. C. Singleton generalized the concept of snake-in-the-box codes to circuit codes with a parameter *spread* [21]. A circuit code of spread δ has unit distance between adjacent code words, and minimum distance δ between code words δ apart in the ordered sequence. For example, the circuit codes with the spread $\delta = 2$ are the coil-in-the-box, and the codes with $\delta = 1$ and 2^n distinct code words are the Hamiltonian cycles of the n-cube. Singleton then presented constructions for circuit codes for spreads up to 7.

Circuit codes are useful in correcting and limiting errors in analog-to-digital conversion (see [16]). The longer the code, the greater the accuracy of the system (while the greater the spread, the greater the error-detection capability). Therefore, determining the length of the longest n-dimensional circuit code of spread δ is of interest [8, 23].

V. Klee showed the construction of a code with even spread δ by extending a code of spread δ using a code of spread $\delta-1$ [15]. K. Deimer described a method for finding a circuit code of spread δ and length $L-k$ in the n-cube from a circuit code in dimension $n+1$ of spread $\delta+1$ and length L [4]. Here k is the number of times a certain transition is taken (this transition number is then removed from the transition sequence). It is not evident that finding such a code (of higher spread and length, in higher dimension) is easier than the target circuit code. Paterson and Tuliani presented a construction method based on binary necklaces [17], generalizing ideas for obtaining single-track circuit codes of [7]. In earlier work, we improved lower bounds and proved optimality of circuit codes for 14 different sets of parameters (n, δ) [24]. The approach uses a SAT-solver and is not limited to specific values of a spread.

In this paper, we present new (longer than previously known) circuit codes that we have obtained using a novel efficient propositional satisfiability encoding.

2. CIRCUIT CODES

Consider an ordered sequence C of L binary code words $W_0, W_1, \ldots, W_{L-1}$. Let $d_H^{k,m}$ denote the Hamming distance between code words W_k and W_m:

$$d_H^{k,m} = \left|\{\ i \mid W_k[i] \neq W_m[i]\}\right| ,$$

where $W_k[i]$ denotes the i-th bit of k-th code word, and let d_C be the cyclic distance between code words in the sequence [12]:

$$d_C^{k,m} = \min\{|k-m|,\ L-|k-m|\} .$$

A *path* in a hypercube is a sequence of code words, in which consecutive elements have unit Hamming dis-

This research is supported in part by an award from IBM Research and by ETH Research Grant TH-19 06-3.

tance. In a *cycle*, the first and the last code words also have unit Hamming distance:

$$\forall k \in \{0, 1, \ldots, L\} : \\ d_C^{k,k+1 \bmod L} = 1 \Longrightarrow d_H^{k,k+1 \bmod L} = 1 \, .$$

The definitions of circuit codes by Paterson and Tuliani [17] and by Preparata and Nievergelt [18] differ slightly, but were proven equivalent by L. Haryanto [12].

Definition 1 (Circuit code [13])**.** *A length L, spread δ circuit code in the n-cube (or $(n,\ L,\ \delta)$-CC) is a cyclic path C of L binary n-tuples $W_0, W_1, \ldots, W_{L-1}$ with the property that for all $k, m \in \{0, 1, \ldots, L-1\}$,*

$$d_H^{k,m} < \delta \Longrightarrow d_C^{k,m} < \delta \, . \tag{1}$$

3. PROPOSITIONAL SAT ENCODING

A SAT solver determines whether a propositional formula, given in conjunctive normal form (CNF), is satisfiable. If it is, the solver provides a satisfying assignment to the variables in the formula.

In this section, we describe our encoding of a search for circuit codes into a propositional SAT formula in detail. Then, we improve it using an observation about the circuit codes' structure and obtain new codes.

3.1. The Satisfiability Problem

Let $V = \{x_0, x_1, \ldots, x_{t-1}\}$ be a set of Boolean variables. A *literal* is a variable x_i or its negation $\neg\, x_i$. Let ϕ be a propositional formula over the variables in V. The propositional satisfiability (SAT) problem [10] is to determine whether there exists an assignment of truth values to the variables in V such that the formula ϕ evaluates to **true**.

3.2. Encoding of Circuit Codes

Our goal is to construct a formula with satisfying assignments corresponding to the coordinates of the nodes forming a spread-δ circuit code of L code words in the n-cube. For this purpose, we define $n{\cdot}L$ Boolean variables denoted by $W_i[j]$, where $i \in \{0, \ldots, L-1\}$ and $j \in \{0, \ldots, n-1\}$. The Boolean vector W_i denotes the coordinates of node number i of the code, where $W_i[0]$ corresponds to the right-most bit of the coordinates of node W_i.

For a sequence of nodes $W_0, W_1, \ldots, W_{L-1}$, we encode the following constraints:

1. To form a cycle in an n-cube, the neighbouring nodes of a sequence must be adjacent. The adjacency is expressed using the Hamming distance:

$$\phi^{cycle} := \bigwedge\nolimits_{k=0}^{L-1} \left(d_H^{k,k+1 \bmod L} = 1\right) . \tag{2}$$

2. The formula (1) suggests to pick pairs of codewords W_k and W_m, that are at least δ apart in the sequence[1] and require their Hamming distances to be at least δ:

$$\phi^{\delta} := \bigwedge_{0 \le k < m < L} \left(d_C^{k,m} \ge \delta \Rightarrow d_H^{k,m} \ge \delta\right) . \tag{3}$$

The propositional formula

$$\phi^{CC} := \phi^{cycle} \wedge \phi^{\delta} \tag{4}$$

encodes an $(n,\ L,\ \delta)$-CC. A satisfying assignment of ϕ^{CC} contains the coordinates of some circuit code with these parameters.

We encoded formulae (2) and (3) using *once-twice* chains and bitonic sorting networks respectively (for details, see [3]) and obtained new circuit codes in [24].

3.3. A More Efficient Encoding

One can reduce the number of variables and clauses by a factor of two.

Consider nodes W_k and W_m together with neighbours of W_m: W_{m-1} and W_{m+1}. If W_k is at least δ apart in the sequence from each of these three nodes, we have to require that their Hamming distances are at least δ each (see Figure 1).

Suppose that k and m are of the same parity, i.e. the distance $d_C^{k,m}$ is even. Then, by 2-colourability of a hypercube [11], the Hamming distance is also even. If the value of the spread is odd, we can strengthen the property (3) to $d_H^{k,m} \ge \delta + 1$, which implies the separability condition for W_k and neighbours of W_m (because Hamming distances for them may decrease only by one). In Eq. (3) we can therefore reduce the number of constraints by about one half due to redundancy.

3.4. Evaluation

[1] With such a formulation, for codes with higher spreads there are fewer of pairs to consider (given that dimension and length are fixed), hence an encoding of these codes requires fewer variables and clauses. This is advantageous as performance of existing approaches decays with increasing spread (e.g., the construction in [17] uses special kinds of binary necklaces and finding them for higher spreads is hard).

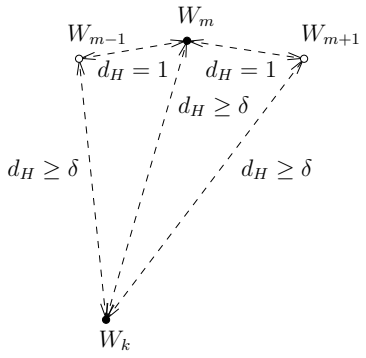

Figure 1: Constraints on code words to form a circuit code.

We generalize this observation, by modifying the Eq. (3) as follows:

$$\phi^{\delta'} := \bigwedge\nolimits_{0 \leq k < m < L} \Big(((d_C^{k,m} \bmod 2 \neq \delta \bmod 2) \\ \wedge\ d_C^{k,m} \geq \delta) \Rightarrow d_H^{k,m} \geq \delta + 1 \Big) . \quad (5)$$

The modified encoding of a $(10, 84, 3)$-CC takes 49.9% less variables and clauses. The runtime[2] for this instance is decreased by 51.5%.

Using the efficient encoding we obtained 11 new circuit codes (see Table 1).

References

[1] D. Casella and W. Potter, "Using evolutionary techniques to hunt for the snakes and coils," *The 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2499–2505, 2005.

[2] D. A. Casella, "New lower bounds for the snake-in-the-box and the coil-in-the-box problems: Using evolutionary techniques to hunt for snakes and coils," Master's thesis, The University of Georgia, 2005.

[3] Y. Chebiryak and D. Kroening, "Towards a classification of Hamiltonian cycles in the 6-cube," *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, vol. 4, pp. 57–74, 2008.

[4] K. Deimer, "Some new bounds for the maximum length of circuit codes." *IEEE Transactions on Information Theory (IT)*, vol. 30, no. 5, pp. 754–756, 1984.

[2]For our experiments, we use the MiniSat, written by Eén and Sörensson [6]. MiniSat provides interfaces for incremental solving and All-SAT. The current version uses preprocessing techniques from QBF-Solvers [5]. All experiments were carried out on a PC with an Intel Xeon (3.0-GHz, 4-GB RAM, running Linux) with a timeout of 24 h.

[5] N. Eén and A. Biere, "Effective preprocessing in SAT through variable and clause elimination," in *Theory and Applications of Satisfiability Testing (SAT)*, ser. LNCS, vol. 3569. Springer, 2005, pp. 61–75.

[6] N. Eén and N. Sörensson, "An extensible SAT-solver," in *Theory and Applications of Satisfiability Testing (SAT)*, ser. LNCS. Springer, 2004, vol. 2919, pp. 502–518.

[7] T. Etzion and K. G. Paterson, "Near optimal single-track gray codes." *IEEE Transactions on Information Theory (IT)*, vol. 42, no. 3, pp. 779–789, 1996.

[8] A. A. Evdokimov, "Maximal length of circuit in a unitary n-dimensional cube," *Mathematical Notes*, vol. 6, no. 3, pp. 642–648, 1969.

[9] L. Glass, "Combinatorial aspects of dynamics in biological systems," in *Statistical mechanics and statistical methods in theory and applications*, U. Landman, Ed. Plenum, 1977, pp. 585–611.

[10] J. Gu, P. Purdom, J. Franco, and B. Wah, "Algorithms for the satisfiability (SAT) problem: a survey," *DIMACS*, pp. 19–152, 1997.

[11] F. Harary, J. P. Hayes, and H.-J. Wu, " A survey of the theory of hypercube graphs," *Comput. Math. Appl.*, vol. 15, pp. 277–289, 1988.

[12] L. Haryanto, "Constructing Snake-in-the-box codes and families of such codes covering the hypercube," Ph.D. dissertation, Delft University of Technology, January 2007.

[13] A. P. Hiltgen and K. G. Paterson, "Single-track circuit codes," *IEEE Transactions on Information Theory (IT)*, vol. 47, no. 6, pp. 2587–2595, 2001.

[14] W. H. Kautz, "Unit distance error checking codes," *IRE Trans. on Electronic Computers*, vol. 7, pp. 179–180, 1958.

[15] V. Klee, "A method for constructing circuit codes," *J. ACM*, vol. 14, no. 3, pp. 520–528, 1967.

[16] ——, "The use of circuit codes in analog-to-digital conversion," *Graph Theory and its Applications*, pp. 121–132, 1970.

[17] K. G. Paterson and J. Tuliani, "Some new circuit codes." *IEEE Transactions on Information Theory (IT)*, vol. 44, no. 3, pp. 1305–1309, 1998.

Table 1: New Circuit codes

n	L	δ	Transition sequence
10	86	3	461328167a3869a3495769a4629832173497531a576154785279348759 1a754a615a8624973526758a1687
11	68	4	546138b5497b6521b79a5b14985b279543268597b213879245a836b29a6872a4b629
12	98	4	18523b95a641b89a175321bc4238b5928cba91c83b59c16b748391ca596342b53a6174a381ba58cb23891a b237c4a82bca
12	56	5	8ca57968a473b91a547218ca5b143a79b463ca54732c8ab31279a342
13	50	6	1b264a98b56da39715bc49a18b246ad5b1c4a987d3cba86d5c
14	66	6	85d473c2b517e94362d795384c719da46731b542961d8349e157c843b615294bac
15	82	6	d4c61b9de3f7bad436785ab23e18579a4d8f295e48c19da68eb94236df1e9267ba5d293f4bde8369a2
16	106	6	gdbc146g7dec234gdfebc157af48637ce519b7dcgf94a52b713f92db63afc827b1ef8379b6f48dgbe6c9854 be1c7ag452b368f5492
16	72	7	b951g74e2b81d5c4ef238a147d92561efb48276gf31d47a26935d7efac4b9de8a2713d6e
17	90	7	472h8g1f37e2dg5h94c762eaf8cg697bh3fgdc4273659e1fdag259b73fha5e968b274ce9d561fc7ae981gc5b69
17	64	8	fc4g7e813f9h5gca7bfehd4g56738eh9fcg4213e5fb94dg7ehf38bac9ge5682b

[18] F. Preparata and J. Nievergelt, "Difference-preserving codes," *IEEE Transactions on Information Theory (IT)*, vol. 20, no. 5, pp. 643–649, September 1974.

[19] D. Rajan and A. Shende, "Maximal and reversible snakes in hypercubes," in *24th Annual Australasian Conference on Combinatorial Mathematics and Combinatorial Computing*, 1999.

[20] K. G. Scheidt, "Searching for patterns of snakes in hypercubes," *Proceedings of the second annual CCSC on Computing in Small Colleges Northwestern conference*, pp. 168–176, 2000.

[21] R. C. Singleton, "Generalized snake-in-the-box codes," *IEEE Transactions on Electronic Computers*, vol. EC-15, no. 4, pp. 596–602, 1966.

[22] D. Tuohy, W. Potter, and D. Casella, "Searching for Snake-in-the-Box codes with evolved pruning models," in *Int. Conf. on Genetic and Evolutionary Methods*, 2007, pp. 3–9.

[23] A. D. Wyner, "Note on circuits and chains of spread k in the n-cube," *IEEE Trans. Comput.*, vol. C-20, no. 4, pp. 474–474, April 1971.

[24] I. Zinovik, D. Kroening, and Y. Chebiryak, "Computing binary combinatorial Gray codes via exhaustive search with SAT-solvers," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1819–1823, April 2008.