Synthesis of Boolean Functions using Information Theory

Arturo Hernández Aguirre[1],
Edgar C. González Equihua[1] and Carlos A. Coello Coello[2]

[1] CIMAT, Area de Computación, Callejón Jalisco s/n
Mineral de Valenciana, Guanajuato, Guanajuato 36240, MEXICO
artha,equihua@cimat.mx
[2] CINVESTAV-IPN, Evolutionary Computation Group
Depto. Ing. Eléctrica, Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco, México, D.F. 07300, MEXICO
ccoello@cs.cinvestav.mx

Abstract. In this paper, we propose the use of Information Theory as the basis of the fitness function for Boolean circuit design. Boolean functions are implemented by means of multiplexers and genetic programming. Entropy based measures such as Mutual Information and Conditional Entropy are investigated as tools for similarity measures between circuits. A comparison of synthesized (through evolution) and minimized circuits through other methods denotes the advantages of the Information-Theoretical approach.

1 Introduction

Entropy is a measure of disorder and the basis of Information Theory (IT) [15]. Shannon [13] suggested the use of information entropy as a measure of the amount of information contained within a message. Thus, entropy tells us that there is a limit in the amount of information that can be removed from a random process without having any information loss. For instance, in theory, music can be compressed (in a lossless form) and reduced up to its entropy limit. Further reduction is only possible at the expense of information lost.

The ID3 algorithm for the construction of classifiers (based on decision trees) is probably the best-known computer science representative that relies on entropy measures [12]. For ID3, an attribute is more important for concept classification if it provides greater "information gain" than the others.

IT was first used by Hartamann et al. [6] to transform decision tables into decision trees. Boolean function minimization through IT techniques has been approached by several authors [7, 8]. These methods are top-down, thus, the design strategy follows after a set of axioms in the knowledge domain. Luba et al. [10] address the synthesis of logic functions using a genetic algorithm and a fitness function based on conditional entropy. Their system needs heavy preprocessing of the search space (Shannon's expansion is applied to the target

Boolean function as to find subexpressions whose purpose is to guide the genetic search. Only after that, the genetic algorithm is started).

In this paper we use multiplexers and genetic programming (GP) for the synthesis of Boolean functions. We propose a fitness function driven by the Normalized Mutual Information between the target function and the evolved function. Our system works exclusively in a bottom-up fashion, thus no preprocessing of the search space is needed. The paper is organized as follows. Section 2 describes the problem statement, Section 3 introduces basic concepts of information theory used throughout the article. In Section 4 we show how entropy based methods will prevent convergence of any evolutionary method if not used correctly. In Section 5 we propose three fitness function based on normalized mutual information and conditional entropy. Section 6 is devoted to experiments, and we finish with conclusions and final remarks in Section 7.

2 Problem Statement

The design problem is the following: find the smallest circuit that implements a Boolean function specified by its truth table [2, 1, 4]. The design metric adopted in this case is the number of components in a 100% functional circuit. The process works at "gate-level" and the only component replicated is the binary multiplexer. A binary multiplexers implements the Boolean function $f = ax + a'y$, where a is the control and {x,y} the input signals. The use of multiplexers is a sound approach because: 1) they are universal generators of Boolean functions, and 2) any circuit in the population is the Shannon expansion of a Boolean function. The expansion takes the form of and-or sum of products (SOP) which are easily represented as decision trees. Therefore, circuits are encoded as trees and the approach follows the representation adopted by Genetic Programming. Leaves of the tree are only 1s and 0s (as in a decision tree), and the nodes are the variables of the Boolean function. Every variable of a node takes the place of the "pivot" variable used in the expansion.

Definition 1. Boolean Residue The residue of a Boolean function $f(x_1, x_2, \ldots, x_n)$ with respect to a variable x_j is the value of the function for a specific value of x_j. It is denoted by f_{x_j}, for $x_j = 1$ and by $f_{\bar{x}_j}$ for $x_j = 0$.

$$f = \bar{x}_j f|_{\bar{x}_j} + x_j f|_{x_j} \tag{1}$$

The pivot variable is x_j. For instance, for the function $f(a, b, c) = a'b'c + a'bc' + ab'c'$, the residue of the expansion over variable a is:

$$f(a, b, c) = a'F(a = 0) + aF(a = 1) = a'(b'c + bc') + a(b'c')$$

Therefore, pivot variable a takes the control of the multiplexer and the two residues form the inputs, as shown in Figure 1.

Further expansion of the functions at the mux inputs yields the complete tree of muxes implementing the target function.

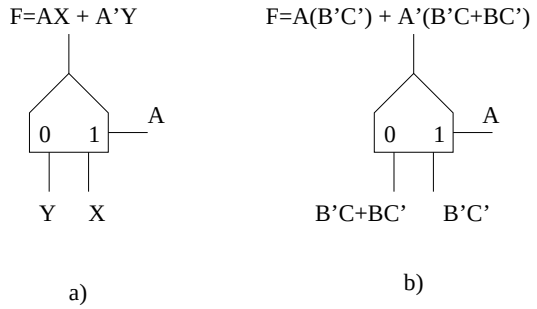

Fig. 1. a) the binary multiplexer, b) the Shannon expansion using a multiplexer

3 Basic concepts of IT

Uncertainty and its measure provide the basis for developing ideas about Information Theory [5]. The most commonly used measure of information is Shannon's entropy.

Definition 2. Entropy The average information supplied by a set of k symbols whose probabilities are given by $\{p_1, p_2, \ldots, p_k\}$, can be expressed as,

$$H(p_1, p_2, \ldots, p_k) = -\sum_{s=1}^{k} p_k log_2 p_k \tag{2}$$

The information shared between the transmitter and the receiver at either end of the communication channel is estimated by its Mutual Information,

$$MI(T;R) = H(T) + H(R) - H(T,R) = H(T) - H(T|R) \tag{3}$$

The conditional entropy $H(T|R)$ can be calculated through the joint probability, as follows:

$$H(T|R) = -\sum_{i=1}^{n}\sum_{j=1}^{n} p(t_i r_j) log_2 \frac{p(t_i r_j)}{p(r_j)} \tag{4}$$

An alternative expression of mutual information is

$$MI(T;R) = \sum_{t\in T}\sum_{r\in R} p(t,r) log_2 \frac{p(t,r)}{p(t)p(r)} \tag{5}$$

Mutual information, Equation 3, is the difference between the marginal entropies $H(T) + H(R)$, and the joint entropy $H(T,R)$. We can explain it as a measure of the amount of information one random variable contains about another random variable, thus it is the reduction in the uncertainty of one random variable due to the knowledge of the other [5].

Conditional entropy is used in top-down circuit minimization methods [3], and also in evolutionary approaches [10, 9].

Mutual information is not an invariant measure between random variables because it contains the marginal entropies. Normalized Mutual Information is a better measure of the "prediction" that one variable can do about the other [14].

$$NMI(T;R) = \frac{H(T) + H(R)}{H(T,R)} \tag{6}$$

Normalized Mutual Information has been used in image registration with great success [11].

Example: We illustrate these concepts by computing the Mutual Information between two Boolean vectors F and C, shown in Table 1. Variable C is an argument of the Boolean function $F(A,B,C) = AB + BC$. We aim to estimate the description the variable C can do about variable F, that is, $MI(F;C)$.

A	**B**	**C**	**F=AB+BC**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Table 1. Function $F = AB + BC$ used to compute MI(F;C)

We use Equations 3 and 4 to calculate $MI(F;C)$. Thus, we need the entropy $H(F)$ and the conditional entropy $H(F|C)$.
Entropy requires the discrete probabilities $p(F=0)$ and $p(F=1)$ which we find by counting their occurrences

$$H(F) = -(\frac{5}{8}log_2\frac{5}{8} + \frac{3}{8}log_2\frac{3}{8}) = 0.9544$$

The conditional entropy, Equation 4, uses the joint probability $p(f_i, c_j)$, which can be estimated through conditional probability, as follows: $p(f,c) = p(f)p(c|f)$. Since either vector F and C has two possible values, the discrete joint distribution has four entries, as follows:

$$p(F=0, C=0) = p(f=0)p(c=0|f=0) = \frac{5}{8} \times \frac{3}{5} = 0.375$$

$$p(F=0,C=1) = p(f=0)p(c=1|f=0) = \frac{5}{8} \times \frac{2}{5} = 0.25$$
$$p(F=1,C=0) = p(f=1)p(c=0|f=1) = \frac{3}{8} \times \frac{1}{3} = 0.125$$
$$p(F=1,C=1) = p(f=1)p(c=1|f=1) = \frac{3}{8} \times \frac{2}{3} = 0.25$$

Now we can compute the conditional entropy by using Equation 4. The double summation produces four terms (since $n = 2$):

$$H(F|C) = -(\frac{3}{8}log_2\frac{3}{4} + \frac{1}{4}log_2\frac{1}{2} + \frac{1}{8}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{2})$$
$$H(F|C) = 0.9056$$

Therefore, $MI(F;C) = H(F) - H(F|C) = 0.9544 - 0.9056 = 0.0488$.

4 Entropy and Circuits

Entropy has to be carefuly applied to the synthesis of Boolean functions. Let's assume any two Boolean functions, $F1$ and $F2$, and a third $F3$ which is the one's complement of F2, then $F3 \neq F2$.

$$H(F2) = H(F3)$$

Also, Mutual Information shows a similar behavior.

$$MI(F1,F2) == MI(F1,F3)$$

The implications for Evolutionary Computation are important since careless use of mutual information can nullify the system's convergence. Assume the target Boolean function is T, then $MI(T,F2) = MI(T,F3)$, but only one of the circuits implementing $F2$ and $F3$ is close to the solution since their Boolean functions are complementary. A fitness function based on mutual information will reward both circuits with the same value, but one is better than the other. Things could worsen as evolution progresses because mutual information increases when the circuits get closer to the solution, but in fact, two complementary circuits are then given larger rewards. The scenario is one in which the population is driven by two equally strong attractors, hence convergence is never reached.

The fitness function of that scenario is as follows. Assume T is the target Boolean function (must be seen as a truth table), and C is output of any circuit in the population. Fitness function is either the maximization of mutual information or minimization of the conditional entropy term. This is,

$$badfitnessfunction\#1 = MI(T,C) = H(T) - H(T|C)$$

The entropy term $H(T)$ is constant since this is the expected target vector. Therefore, instead of maximizing mutual information the fitness function can minimize the conditional entropy,

$$bad fitness function\#2 = H(T) - H(T|C)$$

5 Fitness Function based on Normalized Mutual Information

So far, we have described the scenario where the population is driven by a fitness function based on the sole mutual information. We now propose two new fitness functions based on entropy. Let's assume a target Boolean function of m atributes $T(A_1, A_2, \ldots, A_m)$, and the circuit Boolean function C of the same size. We propose and report experiments using the two following fitness functions (higher fitness means that a better solution has been found).

$$fitness = (Length(T) - Hamming(T, C)) \times NMI(T, C) \tag{7}$$

$$fitness1 = \sum_{i=1}^{m} \frac{fitness}{NMI(A_i, C)} \tag{8}$$

$$fitness2 = \sum_{i=1}^{m} fitness \times NMI(A_i, C) \tag{9}$$

$$fitness3 = (Length(T) - Hamming(T, C)) \times (10 - H(T|C)) \tag{10}$$

Fitness1, Equation 7, is driven by $NMI(T, C)$ and adjusted by the factor $Length(T) - Hamming(T, C)$. This factor tends to zero when T and C are far in Hamming distance, and tends to $Length(T)$ when T and C are close in Hamming distance. The effect of the term is to give the correct rewarding of the NMI to a circuit C close to T. Equation 7 is designed to remove the convergence problems described in the previous section. Fitness1 and Fitness2, Equations 8 and 9, combine the NMI of T and C with NMI of C and the attributes A_k of the target function. Thus, fitness1 and fitness2 pretend to use more information available in the truth table in order to guide the search. Fitness3 is based on conditional entropy and it uses the mentioned factor to supress the reproduction of undesirable trees. Since conditional entropy has to be minimized we use the factor $10 - H(T|C)$ in order to miximize fitness.

6 Experiments

In the following experiments we find and contrast the convergence of our GP system for the three fitness functions defined above.

6.1 Experiment 1

Here we design the following (simple) Boolean function:

$$F(a,b,c,d) = \sum(0,1,2,3,4,6,8,9,12) = 1$$

We use a population size of 300 individuals, $p_c = 0.35$, $p_m = 0.65$, and we run our algorithm for 100 generations. The optimal solution has 6 nodes, thus we find the generation in which the first 100% functional solution appears, and the generation number where the optimal is found. The problem was solved 20 times for each fitness function.

Table 2 shows the results of these experiments.

Event	Gen. at fitness1	Gen. at fitness2	Gen. at fitness3
100% Functional	13 ± 5	14 ± 7	18 ± 6
Optimum Solution	30 ± 7	30 ± 10	40 ± 20

Table 2. Generation number where the first 100% functional circuit is found, and the generation where the optimum is found, for three fitness functions

6.2 Experiment 2

The next test function is:

$$F(a,b,c,d,e,f) = ab + cd + ef$$

In this case, we use a population size of 600 individuals, $p_c = 0.35$, $p_m = 0.65$, and we stop after 200 generations. The optimal solutions has 14 nodes. Each problem was solved 20 times for each fitnesss function.

Table 3 shows the results of these experiments.

Event	Gen. at fitness1	Gen. at fitness2	Gen. at fitness3
100% Functional	39 ± 12	40 ± 11	50 ± 12
Optimum Solution	160 ± 15	167 ± 15	170 ± 20

Table 3. Generation number where the first 100% functional circuit is found, and the generation where the optimum is found, for three fitness functions

6.3 Experiment 3

The last problem is related to partially specified Boolean functions [1]. With this experiment we address the ability of the system to design Boolean functions with "large" number of arguments and specific topology. For this, we have designed a synthetic problem were the topology is preserved when the number of variables increases.

Boolean functions with $2k$ variables are implemented with $(2 * 2k) - 1$ binary muxes *if* the truth table is specified as shown in Table 4.

ABCD	F(ABCD)
0 0 0 0	0
0 0 0 1	1
0 0 1 0	1
0 1 0 0	1
1 0 0 0	1
0 1 1 1	1
1 0 1 1	1
1 1 0 1	1
1 1 1 0	1
1 1 1 1	0

Table 4. Partially specified Boolean function of Example 3 needs $(2 * 2k) - 1$

We ran experiments for $k = 2, 3, 4$, thus 4,8, and 16 variables and we have contrasted these results with the best known solutions for this problem (reported in [1]). For completeness, all previous results are reported together with the results of the new experiments in Table 5, where we use the three fitness functions (Equations 8,9,10).

k	variables	size	Avg(previous)	Avg(fitness1)	Avg(fitness2)	Avg(fitness3)
2	4	7	60	60	60	60
3	8	15	200	190	195	194
4	16	31	700	740	731	748
5	32	63	2000	2150	2138	2150

Table 5. Generation number where the first 100% functional circuit is found, and the generation where the optimum is found, for three fitness functions

All parameters are kept with no change for similar experiments, average is computed for 20 runs. The previous experiments use a fitness function based on Hamming distance between the current solution of an individual and the target solution of the truth table. One important difference is the percentage of

correct solution found. Previously we reported that in 90% of the runs we found the solution (for the case of fitness based on Hamming distance). For the three fitness functions based on entropy we found the solution in 99% of the runs.

7 Final remarks and conclusions

A fitness function using only conditional entropy was tested with no success at all. We believe this is a clear indication of a fitness function that does not take into account the properties of entropy. In general, the three fitness functions work quite well, all of them found the optimum in most cases, thus comparable to other fitness functions based on Hamming distances. Entropy based measures seem hard to adapt to Evolutionary Computation since the entropy of evolutionary systems is not well understood (after "creationists" would say evolution is imposible because entropy would not allow the development of a system). The final remark is that the convergence time and the quality of results produced is comparable with the many experiments we have done before in this area. Based on the results shown in Tables 2 and 3 we would give some advantage to normalized mutual information over simple mutual information because it is less biased. Results from Table 5 could imply that mutual information is able to capture "that" relationship between the data that the sole Hamming distance can not convey to the population.

Acknowledgements

The first author acknowledges partial support from CONACyT project No. I-39324-A. The second author acknowledge support from CONACyT through a scholarship to complete the Master in Science program at CIMAT The third author acknowledges support from CONACyT project No. NSF-CONACyT 32999-A.

References

1. Arturo Hernández Aguirre, Bill P. Buckles, and Carlos Coello Coello. Evolutionary synthesis of logic functions using multiplexers. In C. Dagli, A.L. Buczak, and et al., editors, *Proceedings of the 10th Conference Smart Engineering System Design*, pages 311–315, New York, 2000. ASME Press.
2. Arturo Hernández Aguirre, Carlos Coello Coello, and Bill P. Buckles. A genetic programming approach to logic function synthesis by means of multiplexers. In Adrian Stoica, Didier Keymeulen, and Jason Lohn, editors, *Proceedings of the First NASA/DoD Workshop on Evolvable Hardware*, pages 46–53, Los Alamitos, California, 1991. IEEE Computer Society.
3. V. Cheushev, S. Yanushkevith, and et al. Information theory method for flexible network synthesis. In *Proceedings of the IEEE 31st. International Symposium on Multiple-Valued Logic*, pages 201–206. IEEE Press, 2001.

4. Carlos Coello Coello and Arturo Hernández Aguirre. Design of combinational logic circuits through an evolutionary multiobjective optimization approach. *Artificial Intelligence for Engineering, Design, Analysis and Manufacture*, 16(1):39–53, January 2002.
5. T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991.
6. C.R.P. Hartmann, P.K. Varshney, K.G. Mehrotra, and C.L. Gerberich. Application of information theory to the construction of efficient decision trees. *IEEE Transactions on Information Theory*, 28(5):565–577, 1982.
7. A.M. Kabakcioglu, P.K. Varshney, and C.R.P. Hartmann. Application of information theory to switching function minimization. *IEE Proceedings, Part E*, 137:387–393, 1990.
8. A. Lloris, J.F. Gomez-Lopera, and R. Roman-Roldan. Using decision trees for the minimization of multiple-valued functions. *International Journal of Electronics*, 75(6):1035–1041, 1993.
9. T. Luba, C. Moraga, S. Yanushkevith, and et al. Application of design style in evolutionary multi-level network synthesis. In *Proceedings of the 26th EUROMICRO Conference Informatics:Inventing the Future*, pages 156–163. IEEE Press, 2000.
10. T. Luba, C. Moraga, S. Yanushkevith, and et al. Evolutionary multi-level network synthesis in given design style. In *Proceedings of the 30th IEEE International Symposium on Multiple valued Logic*, pages 253–258. IEEE Press, 2000.
11. Frederik Maes, André Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, April 1997.
12. J.R. Quinlan. Learning efficient classification procedures and their application to chess games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Springer, Berlin, Heidelberg, 1983.
13. Claude E. Shannon. A Mathematical Theory of Information. *Bell System Technical Journal*, 27:379–423, July 1948.
14. C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32:71–86, 1999.
15. W. Weaver and C. E. Shannon. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois, 1949.