Gate-level Synthesis of Boolean Functions using Information Theory Concepts

Arturo Hernández Aguirre
Center for Research in Mathematics, Department of Computer Science
Mineral de Valenciana, Guanajuato, 36240, México
artha@cimat.mx

Carlos Coello Coello
CINVESTAV-IPN Computer Science Section
México, D.F. 07300, México
ccoello@cs.cinvestav.mx

Abstract

In this paper we apply information theory concepts to evolutionary Boolean circuit synthesis. We discuss the schema destruction problem when simple conditional entropy is used as fitness function. The design problem is the synthesis of Boolean functions by using the minimum number of binary multiplexers. We show that the fitness landscape of normalized mutual information exhibits better characteristics for evolutionary search than the landscape of simple mutual information. A comparison of minimum evolved circuits shows the potential of information theory concepts.

1 Introduction

Information theory originated from the studies of Claude Shannon on the transmission of messages over communication channels. A message is only a string of symbols with some meaning in some particular domain. For digital communications and many other areas, removing or identifying redundant information from a message is part of the procedure to improve channel capacity. In early 20th century several researchers had agreed in the relationship between the information carried by a message and the minimum or shortest code that represents such the message. A way to understand information was proposed by Shannon [14], who 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 (a message) without 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. After Shannon, entropy is the measure of disorder and the basis of Information Theory (IT) [16]. Information theory (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 [8, 9]. These methods are top-down, thus, the design strategy follows after a set of axioms in the knowledge domain. Luba et al. [11] address the synthesis of logic functions using a genetic algorithm and a fitness function based on simple conditional entropy.

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 [13]. For ID3, an attribute is more important for concept classification if it provides greater "information gain" than the others.

In this paper we use multiplexers and genetic programming (GP) for the synthesis of gate-level Boolean functions. This means that GP working at a gate-level representation will try to produce the circuit that implements the Boolean function. We propose a fitness function that by measuring the Normalized Mutual Information, drives the search towards circuits that maximize the similarity 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 fitness function works in two stages. The goal of the first stage is the generation of 100 % functional circuits. The goal of the second stage is the minimization of the number of components of those circuits. Therefore, two fitness functions are used during the evolutionary process. 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 is a 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. The Shannon expansion of the function $f(a, b, c) = a'b'c + a'bc' + ab'c'$, is the residue of the expansion over variable *a*:

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

The procedure could continue till no further expansion is possible. In Figure 1 we show the result of the expansion performed over variables "b-c-a" (in that order). Note that the variable over which the expansion is performed takes the control of the multiplexer, and the two residues become inputs. Different variable ordering result in different size circuits, thefore, in principle the use of an evolutionary method to deal with the combinatorial problem is sound.

Further expansion of the "residual" functions yields the complete tree of muxes implementing the target function.

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 [11, 10].

Studholme [15] proposed normalized mutual information as an invariant measure for image registration problems. His approach improves mutual information since he shows his normalized version has better characteristics for measuring the shared information between two images at different angles and area of overlapping. We investigate this issue in Section 4

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

Example: We illustrate these concepts by computing the Mutual Information between two Boolean vectors F and C,

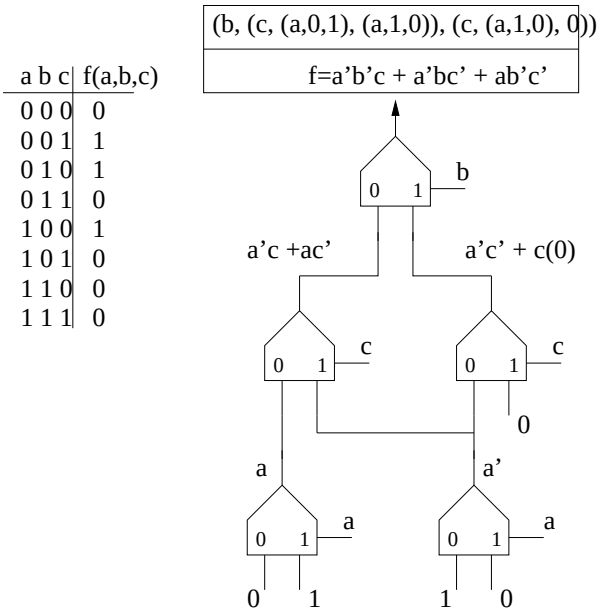

Figure 1. Shannon expansion of function and equivalent circuit using multiplexers

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)

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, $MI(F; C)$.

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:

$$\begin{aligned}
p(F = 0, C = 0) = &\quad p(f = 0)p(c = 0|f = 0) = \tfrac{5}{8} \times \tfrac{3}{5} = 0.375 \\
p(F = 0, C = 1) = &\quad p(f = 0)p(c = 1|f = 0) = \tfrac{5}{8} \times \tfrac{2}{5} = 0.25 \\
p(F = 1, C = 0) = &\quad p(f = 1)p(c = 0|f = 1) = \tfrac{3}{8} \times \tfrac{1}{3} = 0.125 \\
p(F = 1, C = 1) = &\quad p(f = 1)p(c = 1|f = 1) = \tfrac{3}{8} \times \tfrac{2}{3} = 0.25
\end{aligned}$$

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

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

Therefore, $MI(F; C) = H(F) - H(F|C) = 0.9544 - 0.9056 = 0.0488$. In fact, for the the three arguments of the example function we calculate that $MI(F; A) = MI(F; B) = MI(F; C)$. Also, $NMI(F; A) = NMI(F; B) = NMI(F; C) = 1.0256$. Although no function argument seems to carry more information than the others for the truth table of this example, next we show that the fitness landscape spawn by these measures differs in shape, and that NMI seems more suitable for searching.

4 Entropy and Circuits

Entropy has to be carefuly applied to the synthesis of Boolean functions. 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 behaviour.

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

The implications for Evolutionary Computation are important since careless use of mutual information can anulate 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 would evolve towards 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 prefered over the other. Things could go worst as evolution progresses because the mutual information increases when the circuits are closer to the solution. In fact, two complementary circuits are given larger rewards. The scenario is one in which the population is pulled by two equally strong attractors whose best solution are complementary individuals. When these individuals are selected for reproduction they destroy their building blocks hence convergence is never reached. The events just described are similar to the TwoMax problem and the Ising Model problem of Hoyweghen et.al. [7]. They propose to treat the problem as a multimodal function optimization, so a crowding method (niching) is used in order to allow the evolution of the population towards two attractors. In this article we do not wish to know both complementary solutions, therefore, we propose the use of the Hamming distance of an individual as a way to bias the search towards only one of the attractors (see next section).
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 only minimize the conditional entropy,

$$badfitnessfunction\#2 = H(T|C) \qquad (7)$$

We called *bad* to these fitness functions based on mutual information because we were not able to find a solution with them. Although mutual information has been described as the "common" information shared by two random processes, the search space is not amenable for evolutionary computation. In Figure 2 we show this search space over mutual information for all possible combinations with two binary strings of 8 bits (shown in decimal). The area shown corresponds to about $\frac{1}{4}$ ($[1, 150] \times [1, 150]$) of the whole search space of ($[1, 254] \times [1, 254]$) (the values 0 and 255 were not used).

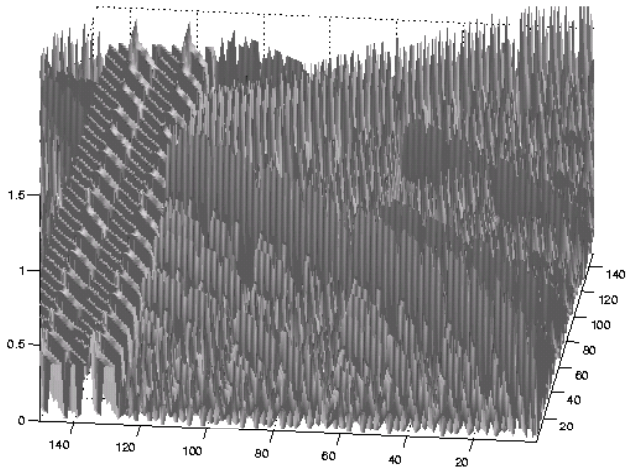

Figure 2. The search space of Mutual Information

The mutual information space, clearly full of spikes, does not favors the area of common information. For any two equal vectors, their Mutual Information lies on the line at 45^o (over points $\{(1,1),(2,2),(3,3)\ldots(n,n)\}$). In the next Section we continue this discussion and design fitness functions whose *landscape* seems more promisory for exploration.

5 Fitness Function based on Normalized Mutual Information

So far we have described the poor scenario where the search is driven by a fitness function based on the sole mutual information. We claim that fitness functions based on Normalized Mutual Information (NMI) should improve the performance of the genetic programming algorithm because of the form of the NMI landscape. This is shown in Figure 3 for two 8-bit vectors (as previous case). Note on the figure how the search space becomes more regular, and more important, notice the appearance of the *wall* at 45^o where both strings are equal.

We propose three new fitness functions based on Normalized Mutual Information (Equation 6) and report

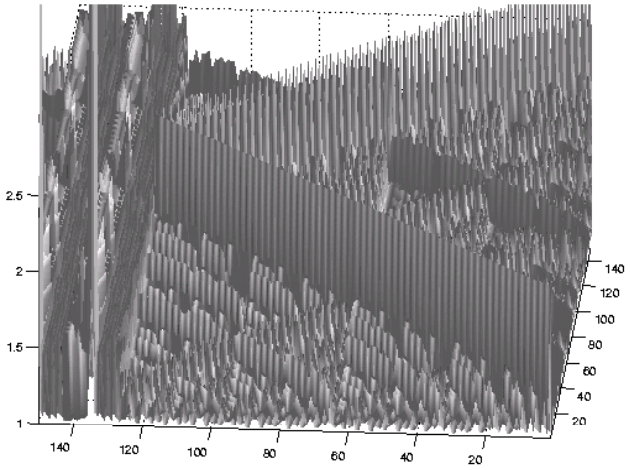

Figure 3. The search space of Normalized Mutual Information

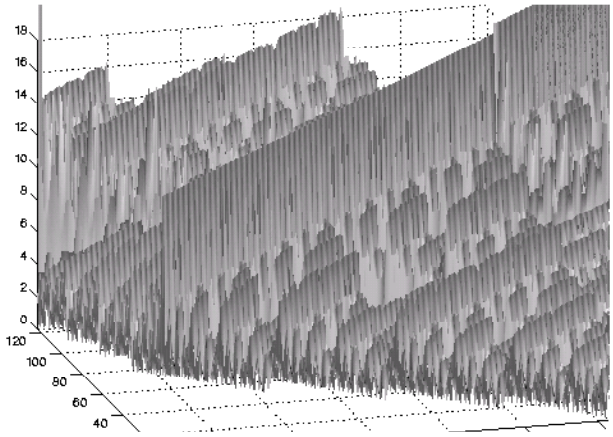

Figure 4. Fitness landscape of: $f = (Length(T) - Hamming(T,C)) \times NMI(T,C)$

experiments using the next three fitness functions (higher fitness means better).

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. In the following, we propose variations of the basic fitness function of Equation 8, and discuss the intuitive idea of their (expected) behavior.

$$fitness = (Length(T) - Hamming(T,C)) \times NMI(T,C) \quad (8)$$

We tested Equation 8 in the synthesis of several problems and the results were quite optimistic. Thus, based on this primary equation we designed the following fitness functions. In Figure 4 we show the *fitness landscape* of Equation 8.

$$fitness1 = \sum_{i=1}^{m} \frac{fitness}{NMI(A_i, C)} \quad (9)$$

$$fitness2 = \sum_{i=1}^{m} fitness \times NMI(A_i, C) \quad (10)$$

$$fitness3 = (Length(T) - Hamming(T,C)) \times (10 - H(T|C)) \quad (11)$$

The function fitness, Equation 8, 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 8 is designed to remove the convergence problems described in the previous section.Fitness1 and Fitness2, Equations 9 and 10, combines NMI of T and C with NMI of C and the attributes A_k of the target function. Thus, fitness1 and fitness2 pretends 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 maximize fitness. Equations 9 and 7 use the conditional entropy term, nevertheless, only Equation 9 works fine. As a preliminar discussion regarding the design of the fitness function, the noticeable difference is the use of Hamming distance to guide the search towards the aforementioned *optimum wall* of the search space. The Hamming distance destroys elements of the population on one side of the wall, and favors the other side. Thus, there is only one attractor in the search space.

6 Experiments

In the following experiments we find and contrast the convergence of our GP system for the three fitness functions defined above. Initial population is created randomly but allowing the deep of the tree at most the number of variables of the Boolean function.

6.1 Experiment 1

Here we design the following (simple) Boolean function:

Event	Fitness1	Fitness2	Fitness3
100% Funct.	13 ± 5	14 ± 7	18 ± 6
Opt. Soltn.	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

Event	Fitness1	Fitness2	Fitness3
100% Funct.	39 ± 12	40 ± 11	50 ± 12
Opt. Solutn.	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

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

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.

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.

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.

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 $(2 * 2k) - 1$

k	vars	size	Prev.	Fit1	Fit2	Fit3
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

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

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 above results are reported together with the results of the new experiments in Table 5, where we use the three fitness functions (Equations 9,10,11).

All parameters are kept with no change for similar experiments, average is computed for 20 runs. Former 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. We have reported [2, 1] 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 find the optimum in most cases (with some higher probability than in previous experiments), 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. We introduced a combination of Hamming distance and information theory measures in the fitness function to guide the population towards only one attractor. 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. The comparison of the fitness landscapes is already encouraging and favors normalized mutual information, so more experiments are being performed in this direction.

Acknowledgements

The first author acknowledges partial support from CONCyTEG project No. 03-02-K118-037. The second 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] Van Hoyweghen, D. Goldberg, and B. Naudts. From twomax to the ising model: Easy and hard symmetrical problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2002*, pages 626–633, San Francisco, CA, 2002. Morgan Kaufmann Publishers.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Claude E. Shannon. A Mathematical Theory of Information. *Bell System Technical Journal*, 27:379–423, July 1948.

[15] 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.

[16] W. Weaver and C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, 1949.