Using a Genetic Algorithm for Optimizing the Functional Decomposition of Multiple-Valued Functions

T. Kalganova[†], E. Kochergov[‡], N. Strechen[‡]
[†]Belarussian State University of Informatics and Radioelectronics,
Phone: +375172 49 19 81, Fax: +375172 49 51 06,
E-Mail: jack@expert.belpak.minsk.by or pottosina@risq.belcaf.minsk.by
[‡]Institute on Problems of Criminology, Criminalistics and Forensic Expertise
Phone: +375172 23 95 54; Fax: +375172 26 72 79,
E-Mail: jack@expert.belpak.minsk.by

ABSTRACT: The genetic algorithm which determines the good functional decomposition of multiple-valued logic functions is presented. The algorithm expands the range of searching for a best decomposition, providing the optimal column multiplicity. The possible solutions are evaluated using the gain of decomposition for multiple-valued function.

1. INTRODUCTION

In recent years, increasing attention has been given to multiple-valued (MV) circuits. This is because of their ability to reduce the number of interconnections in complicated digital circuits, to increase the information content per unit chip area, and to increase the speed of information transmission [Shousha 1995].

The combinational circuits based on the look-up table type FPGA (field programmable gate array), MV PLA (multiple-valued programmable logic array), FPD (field programmable devices), PAL (programmable logic array), T-gate (multiplexer) and etc. can be designed by using the functional decomposition technique. This technique is to break the many variable function into several subfunctions in terms of decreasing the number of variables. Each of these subfunctions can be implemented independently, and are relatively easier to design.

The classical functional decomposition method contains the two main phase:

1. Finding the best partition of X that provides the minimum number of subfunctions, where X is the set of input variables.
2. Encoding the subfunctions using the minimum number of r-valued variables

The optimization of these phases leads to decreasing the size of combinational circuit. In this paper we search the best partition of X. Note that the success of functional decomposition depends on the partition of X. Moreover, the decomposition methods for r-valued functions and their systems have been poorly investigated. Therefore the problem to find the best partition of X is actual and requires the further researches especially for r-valued functions and their systems.

The search for a fast algorithm to determine the best decomposition of switching (binary) functions is studies by T. Sasao [Sasao 1993]. He presented an algorithm based on testing the necessary condition of function decomposability, using reduced ordered binary decision diagrams. By performing of this algorithm we can intend the best partition of X with minimum column multiplicity. Note that this algorithm does not examine dependence of the variable permutations on the column multiplicity for given partition of X. This drawback can be removed by using genetic algorithm (GA).

The objective of this research is to use such optimization technique as GA to search the best partition of X for a r-valued function. The fitness function determines the best partition for specific variable's permutation.

The multiple-terminal multiple-valued decision diagrams (MTMDD) are used instead of decomposition tables. It allows to investigate (n-1) different partition of X for one MTMDD.

Thus this paper addressed the problem to find the best decomposition without considering the all decomposition variants. For this purpose the genetic algorithm is applied. The distinctive feature of this algorithm is that the possible solutions are coded by real numbers. It permits to increase the GA convergence.

The rest of paper is organized as follows. In section 2 we start with describing the basic notations and definitions. We then discussed the genetic algorithm and its properties. The experimental results is explained in Section 4. the Section 5 draws a conclusion.

2. DEFINITIONS AND NOTATIONS

We will use the following notation throughout.

X : the set of n r-valued variables $\{x_1,x_2,...,x_n\}$, where n is the number of elements of X.
$|X|$: the number of elements of the set X.
X_1 : the bound set of variables, $|X_1| = n_1$;
X_2: the free set of variables, $|X_2| = n_2$;

Definition 1. Let X is the set of r-valued variables. $\{X_1, X_2\}$ is a *partition of* X when $X_1 \cap X_2 = \varnothing$ and $X_1 \cup X_2 = X$.

Definition 2. An r-valued function $f(X)$ is said to have a *generalized disjunctive decomposition* with respect to X_1 if there exist r-valued functions $h_1,h_2,...,h_k$ and g such that (Figure 2-1)

$$f(X)=g(h_1(X_1), h_2(X_1),..., h_k(X_1), X_2) \quad (1)$$

The decomposition is said to be *trivial* if n_1 is 1 or n. A function that has a nontrivial generalized or simple disjunctive decomposition is said to be *decomposable*.

Definition 3. The number of different column patterns in the decomposition table is called as a *column multiplicity* of the decomposition, and denoted by μ.

Note that the column multiplicity defines the type of disjunctive decomposition. An r-valued function $f(X)$ is said to have a simple disjunctive decomposition if μ is less or equal to r [Abugharieh, Lee 1995; Kalganova 1996] and a generalized disjunctive decomposition if μ is less or equal to r^{n_1}, where n_1 is the number of elements in set X_1 [Kalganova 1996].

Definition 4. Let $\{X_1, X_2\}$ be a partition of X. Let $n_1=|X_1|$ and $n_2=|X_2|$. $\gamma=min(r^{n_1}, r^{r^{n_2}})/\mu$ is the gain of the decomposition for a given partition.

3. GENETIC ALGORITHM FOR *r*-VALUED VARIABLE PARTITION PROBLEM

GAs provide a stochastic search technique inspired by principles of natural genetics and evolution [Holland 1975]. They operate through a simulated evolution process on a population of string structures, each of which encoded a possible solution in the search space. Each string structures is called chromosome and a set of chromosomes is referred as a population.

Evolution of population involved two basic steps: (1) a selection mechanism that implements a survival of the fittest strategy and (2) genetic recombination of the selected high-fittest chromosomes to produce offspring for the new generation. Recombination is effected through the two genetic operators of crossover and mutation.

GAs seem to provide a highly effective balance between exploitation (using known good partial solutions) and exploration (trying new solutions) [Miller, Luchian; Bradbeer; Barclay 1994]. First we see how a solution can be mapped to a string. Then we discuss the basic operators applied in searching for an optimum partition of X.

The variable partition problem leads itself to a straightforward representation by means of GA techniques. Indeed, as required for a classical GA approach, a partition of X is easily encoded by a string of integer numbers of length n, where n is the number of variables on which the r-valued function depends. But we solve the coding problem by nontraditional way. Each gene b_i of chromosome $\boldsymbol{b}=(b_1,...,b_i,...,b_n)$ takes real value from the domain. The less value of b_i corresponds to the variable with the less index, where b_i is the gene of chromosome. For an n variable r-valued function the search space (the set of all possible partitions of X) has $\sum_{i=1}^{n-1} C_n^i \big/ 2$ elements.

3.1 FITNESS FUNCTION

The fitness function depends on the gain index γ. This is obtained by generalizing the decomposition algorithm based on the reduced ordered binary decision diagrams analysis to a r-valued function. Let us consider main principles to find the generalized disjunctive decomposition for an n-variable r-valued function. The algorithm consists of two phases:

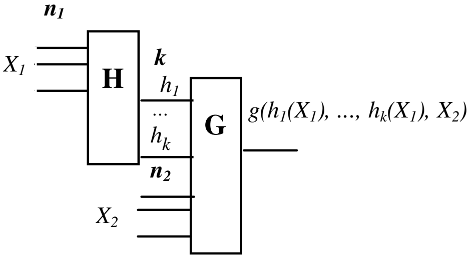

Figure 2-1: Circuit diagram of $f(X)=g(h_1(X_1), h_2(X_1),..., h_k(X_1), X_2)$.

1) A MTMDD for given function is generated.

2) This MTMDD is tested for good decomposition.

Theorem. Let $\{X_1, X_2\}$ be a partition of X. Suppose that the reduced ordered MTMDD for $f(X)$ is partitioned into two blocks such as shown in Figure 3-1. Let t be the number of the nodes in the lower block that are adjacent to the boundary of two blocks, and μ be the column multiplicity of decomposition $f(X)=g(h_1(X_1),..., h_k(X_1),X_2))$. Then t=μ.

Each MTMDD is tested for a good decomposition using the gain γ the specific properties of which is shown in Table 1. As we can see the partition with maximal gain is useful. In our GA the gain index γ determines the evaluation function.

Table 1 **THE SPECIFIC PROPERTIES OF GAIN γ.**

Value of γ	Condition of decomposition existence	Notes
$r<\gamma$	$k<n_1$	This condition determines the existence of generalized disjunctive decomposition.
$1<\gamma\leq r$	k=1	This condition determines the existence of simple disjunctive decomposition.
$\gamma=r^{n_1}$	k is absent	The value of function $f(X)$ does not depends on the variables of X_1. In this case we can design circuit implemented only r-valued function $f(X_2)$. The number of inputs in MV circuit is reduced.
$\gamma=r^{n_1-1}$	k=1	The “don’t cares” inputs for the design of G are absent.
$1<\gamma<min(r^{n_1}, r^{r^{n_2}})$, but ratio γ/r is real number.	$k<n_1$	The number of the outputs of H cannot be reduced. However, we can design H such that it does not produce particular output combinations. Such combinations can be used as the “don’t cares” for the design of G.
$1<\gamma<min(r^{n_1}, r^{r^{n_2}})$, but ratio γ/r is integer number.	$k<n_1$	The same like in previous row, but output combinations of H applied as the “don’t cares” for the design of G are absent.
γ=1	$k=n_1$	Because the purpose of the functional decomposition is to obtain the subfunctions with fewer variables, this decomposition is not applied in practice.
$0<\gamma\leq 1$	$k>n_1$	The number of outputs of H can be smaller than n_1. Because the goal of the functional decomposition is to get the subfunctions with fewer variables, this decomposition is not used in practice.

3.2 CROSSOVER

Crossover is the main genetic operator. It operates on three randomly selected chromosomes (parents) and generates offsprings. Thus, new three offspring are obtained and are replaced the parents in the breeding pool.

The sense of crossover operator is preserved: the obtained chromosomes have better values then parents. This property is essential convention of GA convergence that is to get the optimal solution corresponding to the maximum of fitness function. If this particular feature is not taken into consideration, than the search of optimal solution will be

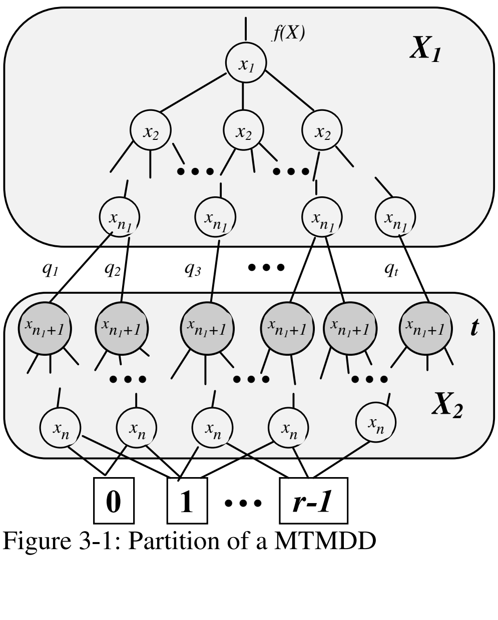

Figure 3-1: Partition of a MTMDD

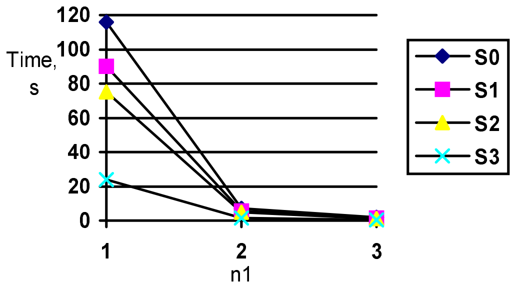

Figure 4-1: Dependence the execution time on the number of X_1, n_1 (n=6, r=4)

depends on the mutation, i.e. GA would be degenerate into random search.

Crossover creates new chromosomes (offsprings) on the basis of three initial ones (parents). Using crossover operator, calculation of new chromosomes is fulfilled by analogy with simplex-method [Bersini, Seront 1992]. Three chromosomes are randomly selected from current population at the chromosomes swapping. The number of parents is determined by α coefficient called as crossover speed.

Assume that certain current population where each chromosome has two genes exists. Select three chromosomes called as parent by random way. Chromosomes from this triad have to be arranged in accordance with values of fitness-function. Interpret initial chromosomes as points A, B, C on surface (Figure 3-2), where values of genes are values of coordinate. In accordance with crossover operator the forth point (D) is derived by building the median CE from "the worst" point (C) and continuation of median ED. First length of ED is equal CE. ED is further decreased at reaching the global extreme.

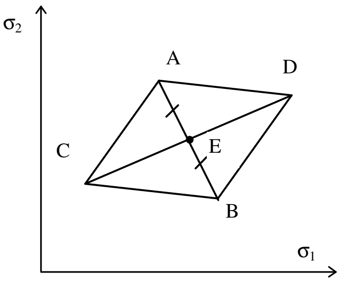

Figure 3-2: Graphical interpretation of crossover

If the number of genes is equal *n*, than described transforms are executed on the basis of mathematical expression:

$$p_i = \frac{x_i+y_i}{2} + \delta\left(\frac{x_i+y_i}{2} - z_i\right) \qquad i=\overline{1,n}$$

where x_i, y_i, z_i are the *i*-th genes of three initial chromosomes arranged in decreasing order of fitness-function. They are correspond to the points A, B, C on Figure 3-2, p_i is the *i*-th gene of new chromosomes performed by crossover (it corresponds to the point D), δ is coefficient called as crossover force.

The analysis of local extreme is envisaged in given algorithm. The sense of this procedure is following. If fitness-function of chromosomes from new generation does not increase, than a conclusion about reaching the global extreme or about falling into "trap" of local extreme is made. In order to go out from local extreme it needs to use the additional procedures, for example "shaking" of population [Michalewicz 1992]. However such approach requires high temporal expenditures (it needs to modeling the evolution for hundred populations). If improvement of fitness-function is lacked, two procedures are used. First one is to use "intense" mutation of current population. In given case, parents as well as offsprings mutate. Second one is based on changing the selection operator. Here new population is formed from the fittest chromosomes as well as from chromosomes changing by random way. Experimental investigations show that suggested procedure allows to go out from the local extreme and to continue the search of global extreme.

3.3 MUTATION

Mutation operator transforms a chromosome by means of randomly changing one or some gene's values from the following range [0..1]. Mutation is used for a part of chromosomes from new population. The number of mutated chromosomes is defined by β coefficient called as mutation speed.

Size of population is increased at the crossover and mutation execution. Here the current population contains both parents and offsptings. Decreasing the size of population is executed by selection operator that forms new population by choice of the fittest chromosomes.

4. EXPERIMENTAL RESULS

The purpose of experimental investigations is to sign the minimum partition of *X* for *r*-valued function.

The GA success depends on the correct choose of the size of populations crossover speed and the number of population. The structure of GA had been suggested in [Kochergov 1996]. Thus GA parameters are selected as in [Kochergov 1996], taking into consideration the computer characteristics such as the size of main memory and CPU time.

The time needed to evaluate the fitness function is investigated too.

Figure **4-1** illustrate the dependence the evaluation time on the number of variables in X_1. S0, S1, S2 and S3 are randomly generated *r*-valued functions. The time complexity is exponentially decreased when we take into consideration the size of X_1. Comparison the execution time of the proposed algorithm with standard decomposition as shown in Figure 4-2. Let us assume that the time complexity of standard decomposition method is 100%. Figure 4-2 illustrate that the time needed to calculate the fitness function by proposed method is decreased as well as the size of X_1 is increased.

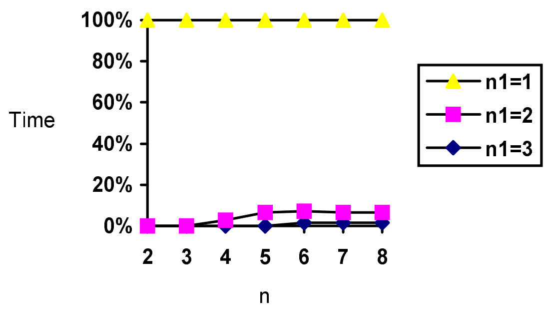

Figure 4-2: Comparison the execution time of the proposed algorithm with standard decomposition (100%) (r=4)

5. CONCLUSION

In this paper, we have presented a new technique to optimize the partition of *X* which leads to reducing the size of synthesized *r*-valued combinational circuit. The distinctive feature of suggested algorithm is that a fast algorithm to find the best partition is applied as one of the part of GA. A possible solutions are encoded by real number instead of integer. This aspect provides the GA convergence. Each chromosome is evaluated using a fast algorithm to find the best position. So, two techniques are incorporated that allows to determine the global optimum. We applied a permutation of input variables to obtain the best results because the result of similar algorithms without permutation does not gives the global optimum. Experiments on large randomly generated circuits and other benchmarks obtained show that the procedure is very effective in finding the best partition.

The future investigations direct into generalized distinguished approach to the system of *r*-valued functions. It permits to research the performance of new approach on other standardized large output benchmark problem.

6. REFERENCES

Abugharieh, S.B.; S.C. Lee 1995. A Fast Algorithm for the Disjunctive Decomposition of *m*-Valued Functions. Part 1. The Decomposition Algorithm. IEEE Transactions on Computers, pp. 118-125.

Bersini, H.; Seront G. 1992. In search of a good crossover between evolution and optimization, in Parallel Problem Solving from Nature 2, Manderick, R. Manner Eds., Amsterdam: Elsevier, pp. 479-488.

Holland J.H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: University of Michigan Press.

Kalganova, T. 1996. Functional Decomposition Methods for Multiple-valued Logic Functions and its System. 3rd International Conference on Application of Computer Systems, Szczecin, Poland, pp.75-82.

Kochergov, E.1996. Optimization of Decision Acceptance in Person Identification System due to Handwritten. 2nd International Conference on the New Information Technologies in Education, Minsk,Belarus,Vol.1, pp.371-380.

Michalewicz, Z. 1992. Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlar, Berlin, Heidelberg.

Miller, J.; Luchian H.; Bradbeer P.; Barclay P. 1994. Using the Genetic Algorithm to Optimizing Fixed Polarity Reed-Muller Expansions of Boolean Functions. Int. J. Electronics, Vol. 76, No. 4, pp. 601-609

Sasao, T. 1993. FPGA Design by Generalized Functional Decomposition. Logic Synthesis and Optimization. Kyushu institute of technology, Iizuka, Japan.

Shousha, A.H. 1995. Switched - Current CMOS Ternary Logic Circuits. Int. J. Electronics, Vol. 79, No. 5, pp. 617-625.