WAVELET PACKETS AND GENETIC ALGORITHMS

Jaakko Astola *Karen Egiazarian* *Heikki Huttunen*
Signal Processing Laboratory, Tampere University of Technology,
P.O.Box 553, FIN-33101, Tampere, FINLAND,
jta@cs.tut.fi, karen@cs.tut.fi, hehu@cs.tut.fi

ABSTRACT

This paper is devoted to the theoretical analysis of the fitness function in genetic algorithms using wavelet packet (WP) transforms. More specifically, WP transforms are used to calculate the average fitness value of a schema. Based on this one can decide whether a certain function is easy or hard for a genetic algorithm. The result is an extension of Bethke's work who discovered an efficient method for calculating schema average fitness values using the Walsh transform.

1. INTRODUCTION

Wavelet packet (WP) transforms (or, equivalently, tree-structured transforms) which are powerful extensions of wavelets and multiresolution analysis, have recently been subjected to a wide interest in the signal processing community. Based on tree-structure filter banks WPs offer a rich family of orthonormal bases, from which one can choose the "best" (under a certain criterion, like entropy-based) basis [8].

Genetic algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics. A simple GA is composed of three operators: reproduction, crossover and mutation [4]. An important problem of GA is to analyze how easy or hard will an objective (whence also fitness) function be for the GA.

As a search algorithm one can use GA for the selection of the "best" basis (under a certain criterion) among all WP bases. Such kind of problems occur e.g. in image coding applications [9].

In this matter the following question arise: what kind of role the WP transforms play in the analysis of GA?

First attempt in the application of orthogonal transforms to the analysis of genetic algorithms was made by Bethke [3], who discovered an efficient way for calculating schema average fitness values using the Walsh transform. The building blocks of GAs, which are combined to form optima or near optima, are short and low-order schemata with above-average fitness values. The comparison of the fitness averages written in terms of the Walsh transform coefficients and in terms of the identity transform coefficients (the canonical basis) showed that [5]:

- for Walsh transform the low-order schemata are specified with a short sum and the high-order schema are specified with a long sum,
- for identity transform the situation is opposite.

This makes the Walsh transform efficient for GA. But the Walsh transform is only one from the class of the Haar-wavelet packets.

How about other transforms, especially those based on rectangular basis functions?

In [7], an attempt was made to use the Haar transform with the GA's. It was based on the fact that in the case of Haar transform there is a reduction of the computation time compared to Walsh transform. Nevertheless, no general formula for schema average fitness in Haar basis, and no general analysis of the complexity (as the number of nonzero terms in the summation for schema average fitness) were presented in [7].

The main goal of this paper is to investigate the application of wavelet packet transforms to the analysis of genetic algorithms. We derive an analytical expression for GA average H-fitness matrix and average H-fitness cost vector (which are quantative measures of complexity of calculating schema average fitness values) corresponding to any basis H from the library of WP transforms.

2. THE SCHEMA THEOREM AND THE WALSH-SCHEMA TRANSFORM

2.1. The schema theorem

First we recall some concepts of GA, the fundamental theorem of GA - a schema theorem, and the Walsh-schema transform [3], [5] - [6].

We assume that GA processes n-bit strings, $\mathbf{x} = (x_{n-1}x_{n-2}...x_0)$, $x_i \in \{0,1\}$ corresponding to the decimal number $x = \sum_{i=0}^{n-1} x_i 2^i$.

A *schema* is a similarity subset containing strings with the defined similarity at some number of positions. For example, the subset $\{(001),(011)\}$ is the schema $(0*1)$, where $*$ is the "don't care" character. Thus, a schema is

$$\mathbf{s} = (s_{n-1}s_{n-2}...s_0),\ s_i \in \{0,1,*\}.$$

If we represent an n-bit string as a node of the binary n-cube, a schema is the covering of corresponding nodes by intervals. There are totally 3^n different schemata.

The order $o(\mathbf{s})$ of a schema $\mathbf{s}$ is the number of fixed positions of similarity in the subset (the number of strings in this subset is $2^{n-o(\mathbf{s})}$). For example, $o(0*1) = 2$.

The length $\delta(\mathbf{s})$ of a schema is the distance between the outermost defining positions of a schema. For example, $\delta(0*1) = 2, \delta(00*) = 1$.

For a given GA problem we have a real-valued fitness function $f(g(\mathbf{x}))$, where $g(\mathbf{x})$ are decision variables. According to the *schema theorem*, (see,[4]) under reproduction, simple crossover, and mutation, the expected number of representatives m of a particular schema $\mathbf{s}$ is at least

$$m(\mathbf{s},t+1) \geq m(\mathbf{s},t)\frac{\tilde{f}(\mathbf{s})}{\bar{f}}[1 - p_c\frac{\delta(\mathbf{s})}{n-1} - p_m o(\mathbf{s})], \quad (1)$$

where $\tilde{f}(\mathbf{s})$ is the *schema average fitness* of the representatives of $\mathbf{s}$ in the current population, defined by

$$\tilde{f}(\mathbf{s}) = \frac{1}{|\mathbf{s}|}\sum_{\mathbf{x}\in\mathbf{s}} f(\mathbf{x}), \quad (2)$$

where $|\mathbf{s}|$ is the number of strings of the subset $\mathbf{s}$, $\overline{f}$ is the average fitness in the population, p_c and p_m are the crossover and the mutation probabilities, respectively, $\delta(\mathbf{s})$ is the length, and $o(\mathbf{s})$ is the order of the schema. This theorem says that a schema grows when it is short, of low-order, and has above-average fitness [5].

2.2. Walsh-schema transform

The Walsh functions, forming an orthogonal set of functions, are defined analytically by

$$w_i(\mathbf{y}) = \prod_{i=1}^{n} y_i^{j_i},\ y_i \in \{-1, 1\}, \qquad (3)$$

where j_i is the i^{th} bit in the binary representation of j, $0 \le j \le 2^n - 1$.

The schema average fitness (in the Walsh transform domain) can be written as [3], [5]:

$$\tilde{f}(\mathbf{s}) = \sum_{j \in J(\mathbf{s})} c_j w_j(\beta(\mathbf{s})),\ J(\mathbf{s}) = \{j : (\exists i) : (\mathbf{s} \subseteq \mathbf{s}_i(j))\}, \qquad (4)$$

where

$$\beta(s_i) = \begin{cases} 0, & \text{if } s_i = 0, * \\ 1, & \text{is } s_i = 1, \end{cases}$$

$$c_j = \frac{1}{2^n} \sum_{x=0}^{2^n-1} f(x) w_j(x).$$

The computation of (4) is called the *Walsh-schema transform* [5]. Note that the low-order schemata are specified with a short sum and the high-order schemata are specified with a long sum. With the schema-theorem this shows why the Walsh transform is effective for GA.

3. AVERAGE FITNESS TRANSFORM MATRIX AND COST VECTOR

Let now $\mathbf{f} = [f(0), ..., f(2^n - 1)]^T$ be the vector form of a fitness function, $\tilde{\mathbf{f}} = [\tilde{f}(0), ..., \tilde{f}(3^n - 1)]^T$ be the vector form of an average fitness function, H_n be any nonsingular $2^n \times 2^n$ matrix, and

$$\mathbf{g} = H_n \mathbf{f}. \qquad (5)$$

We define a $3^n \times 2^n$ matrix $A_n = A(H_n)$ (depending on the matrix H) such that

$$\tilde{\mathbf{f}} = A_n \mathbf{g}. \qquad (6)$$

We call such matrix $A_n = A(H_n)$ the *average H-fitness transform matrix*, and the transform (6) the *average H-fitness transform*.

It is easy to see that in the case $H_n = I_n$ (identity matrix of order 2^n)

$$A_n = A(I_n) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}^{\otimes n}, \qquad (7)$$

where $B^{\otimes n}$ is the n^{th} Kronecker power of B.

Note that the matrix $A(I_n)$ plays the same role as the interval splicing matrix A_n (used for extracting all intervals of the n-cube [1]) and, in fact, coincides with it.

Since $\tilde{\mathbf{f}}$ is independent of H_n we have from (6), (5) and (7):

$$\tilde{\mathbf{f}} = A(H_n) H_n \mathbf{f} = A(I_n)\mathbf{f} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}^{\otimes n} \mathbf{f},$$

therefore

$$A(H_n) H_n = A(I_n),$$

and

$$A(H_n) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}^{\otimes n} H_n^{-1}. \qquad (8)$$

For the comparison of simplicity of the representation (6) for different matrices H_n we introduce an *average H-fitness cost vector* $\mathbf{r} = r(\mathbf{g}) = [r(0), ..., r(2^n - 1)]^T$, where $r(\alpha(s))$ shows the average number of nonzero terms $g(s)$ in the representation (6), where

$$\alpha(s) = \begin{cases} 0, & \text{if } s = 0, 1 \\ 1, & \text{if } s = * \end{cases} \qquad (9)$$

The H-average fitness cost vector gives a quantative measure of complexity of calculating schema average fitness values.

4. RECTANGULAR WAVELET PACKETS AND FITNESS AVERAGE MATRICES AND COST VECTORS

4.1. Rectangular wavelet packets and fitness average matrices

The Haar wavelet packet of the Haar-like unitary transforms $H_n^{(P)}$ corresponds to the tree-structured filter banks (P is a binary tree) with the synthesis filters pair (lowpass and highpass) defined by (see eg. [8]):

$$G_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}),\ G_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}). \qquad (10)$$

An arbitrary pruned tree structure P of the full binary tree of depth n will give a family of Haar-like unitary bases $\{H_n^{(P)}\}$. The extreme cases are the following: If P is the trivial tree (only the root) then $H_n^{(P)}$ is the identity transform, if P is an octave-band tree (iterating only on the lowpass sections) then $H_n^{(P)}$ is the Haar transform, and if P is the full tree then $H_n^{(P)}$ is the Walsh transform.

Below we will give an explicit formula for the average fitness transform matrix corresponding to an arbitrary basis from the Haar wavelet packet.

Let us first derive an analytic expression for the matrix $[H_n^{(P)}]^{-1}$ (which is the inverse of the matrix $H_n^{(P)}$ corresponding to a pruned tree P of the full binary tree of depth n).

For this aim we code each node at the kth ($0 < k \le n$) level of the tree P (the root is on the level 0 and not coded) by a $(0,1)$-vector of length k. Let some nonterminal node be coded by the binary vector $\mathbf{c}$. The descendants of this node will have the following codes: $\mathbf{c}0$ (the left one) and $\mathbf{c}1$ (the right one). To the codes of all terminal nodes which are not on the last, nth, level we add from the right the "don't care" character $*$.

Let P have t terminal nodes with corresponding codes

$$\{\mathbf{c}^{(i)}\},\ \mathbf{c}^{(i)} = (c_1^{(i)}, ..., c_n^{(i)}), \qquad (11)$$

where $c_j^{(i)} \in \{0, 1, *\}$, $i = 1, ..., t$.

Proposition 1. Let P be a pruned tree of the regular tree of depth n and have t terminal nodes with their codes given by (11). The inverse transform matrix $[H_n^{(P)}]^{-1}$ for the Haar-like transform corresponding to the tree P will have the following form:

$$[H_n^{(P)}]^{-1} = 2^{-\frac{n}{2}} \left[2^{\frac{n_1}{2}} I_{n_1} \otimes \left(\bigotimes_{j=1}^{k_1} \begin{pmatrix} 1 \\ -1 \end{pmatrix}^{\alpha(c_j^{(1)})} \right) \cdots \right.$$

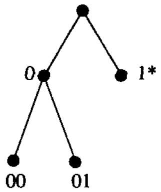

Figure 1. A pruned binary tree with corresponding codes of nodes

$$\cdots 2^{\frac{n_t}{2}} I_{n_t} \otimes \left(\bigotimes_{j=1}^{k_t} \begin{pmatrix} 1 \\ -1 \end{pmatrix}^{\alpha(c_j^{(t)})} \right) \Bigg], \qquad (12)$$

where I_k is the identity matrix of order 2^k, $n_i = n - o(\mathbf{c}^{(i)})$, $o(s)$ is the order of a schema s, $\alpha(c)$ is defined by (9), $\begin{pmatrix} 1 \\ -1 \end{pmatrix}^0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ -1 \end{pmatrix}^1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$.

Proof follows from the definition (10) of the Haar filter pair and the construction of tree codes.

As an example, the transform matrix corresponding to the tree P shown in Fig. 1, is represented analytically by

$$[H_n^{(P)}]^{-1} = \frac{1}{2}\left[I_{n-2} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad I_{n-2} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right.$$

$$\left. \sqrt{2} I_{n-1} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right].$$

Proposition 2. Let P be a binary tree as in Proposition 1. The average H-fitness transform matrix $A(H_n)$ corresponding to the Haar-like transform $H_n = H_n^{(P)}$ (defined by the tree P) will have the following form:

$$A(H_n) = 2^{-\frac{n}{2}} \left[2^{\frac{n_1}{2}} A(I_{n_1}) \otimes \left(\bigotimes_{j=1}^{k_1} v(c_j^{(1)}) \right) \cdots \right.$$

$$\left. \cdots 2^{\frac{n_t}{2}} A(I_{n_t}) \otimes \left(\bigotimes_{j=1}^{k_t} v(c_j^{(t)}) \right) \right], \qquad (13)$$

where $A(I_n)$ is defined by (7), $v(0) = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$, $v(1) = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$, and all other items as in the proposition 1.

Proof. Using proposition 1, formula (8) and simple properties of the Kronecker product, we obtain an analytical expression for $A(H_n)$. Vectors $v(0)$ and $v(1)$ are obtained by $v(0) = A(I_1) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $v(1) = A(I_1) \begin{pmatrix} 1 \\ -1 \end{pmatrix}$.

Using (8) it is easy to derive the Walsh-schema transform (vector-matrix version of (4)), or, equivalently, the Walsh-average fitness transform:

$$\tilde{\mathbf{f}} = A(W_n)\mathbf{g},$$

where

$$A(W_n) = 2^{-n/2} \left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right]^{\otimes n}$$

$$= 2^{-n/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 2 & 0 \end{pmatrix}^{\otimes n} \qquad (14)$$

and

$$\mathbf{g} = W_n \mathbf{f},$$

$W_n = 2^{-\frac{n}{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes n}$ is the Walsh transform matrix.

Similarly, using (13) one can derive other schema transforms, i.e. the Haar-like average fitness transforms.

Let us change the basis functions of Haar wavelets to another rectangular basis, that forms the class of nonorthogonal Reed-Muller wavelet packets. The name comes from the fact that in the extreme case (i.e. when we have the full binary tree decomposition) this construction leads to the Reed-Muller (or conjunctive) transform [1]. In the case of the octave-band tree it leads to oblique wavelets (see, [2]). Analysis and synthesis pairs of filters for the corresponding filter bank are:

$$H_0(z) = 1; H_1(z) = z - 1; G_0(z) = 1 + z^{-1}; G_1(z) = z^{-1}.$$

First, we give an analytic expression for the inverse matrix $[K_n^{(P)}]^{-1}$ of the Reed-Muller packet matrix $K_n^{(P)}$ corresponding to a pruned tree P of the binary tree of depth n.

Proposition 3. Let P be a pruned tree as defined in Proposition 1. The matrix $[K_n^{(P)}]^{-1}$ corresponding to the decomposition tree P will have the following form:

$$[K_n^{(P)}]^{-1} = \left[I_{n_1} \otimes \left(\bigotimes_{j=1}^{k_1} \mathbf{h}_{\alpha(c_j^{(1)})} \right) \cdots \right.$$

$$\left. \cdots I_{n_t} \otimes \left(\bigotimes_{j=1}^{k_t} \mathbf{h}_{\alpha(c_j^{(t)})} \right) \right], \qquad (15)$$

where $\mathbf{h}_i$ is the ith column of the inverse Reed-Muller matrix $K_1^{-1} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$, and all other notations are similar to those in Proposition 1.

As an example, the transform matrix for the tree in Fig. 1, is:

$$[K_n^{(P)}]^{-1} = \left[I_{n-2} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad I_{n-2} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right.$$

$$\left. I_{n-1} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right].$$

Proposition 4. Let P be a binary tree specified in Proposition 1. The average H-fitness transform matrix $A(K_n)$ corresponding to the Reed-Muller-like transform $K_n^{(P)}$ (defined by the tree P) will have the following form:

$$A(K_n) = \left[A(I_{n_1}) \otimes \left(\bigotimes_{j=1}^{k_1} w(c_j^{(1)}) \right) \cdots \right.$$

$$\left. \cdots 2^{\frac{n_t}{2}} A(I_{n_t}) \otimes \left(\bigotimes_{j=1}^{k_t} w(c_j^{(t)}) \right) \right], \qquad (16)$$

where $A(I_n)$ is defined by (7), $w(0) = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$, $w(1) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$, and all other items as in Proposition 1.

Proof follows from Proposition 3 and formula (8). Vectors $w(0)$ and $w(1)$ are obtained by $w(0) = A(I_1)\begin{pmatrix}1\\-1\end{pmatrix}$, $w(1) = A(I_1)\begin{pmatrix}0\\1\end{pmatrix}$.

We have the following average-fitness matrix for the Reed-Muller transform:

$$A(K_n) = \left[\begin{pmatrix}1 & 0\\0 & 1\\1 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\-1 & 1\end{pmatrix}\right]^{\otimes n} = \begin{pmatrix}1 & 0\\-1 & 1\\0 & 1\end{pmatrix}^{\otimes n} \quad (17)$$

Comparing this with the Walsh average-fitness matrix (14) one can see that there is no scaling factor 2 for low-order schemata, and it has a much shorter sum for specification of high-order schemata.

In order to compare complexities of calculation of schema average fitness values we analyze average fitness cost vectors corresponding to different rectangular wavelet packet transforms.

5. RECTANGULAR WAVELET PACKETS AND FITNESS AVERAGE COST VECTORS

Recall that each orthogonal Haar-like transform corresponds to a pruned tree structure. Fixing a pruned tree G we will construct a pruned tree $G' = [G, j]$ just by adding branches to the jth node (i.e. node with the code $j = (j_1, j_2, ..., j_p), p < n$) of G.

Proposition 5. An average H-fitness cost vector $\mathbf{r}(G')$ corresponding to the tree $G' = [G, j]$ can be obtained from an average H-fitness cost vector $\mathbf{r}(G)$ corresponding to G according to the following procedure:

$$\mathbf{r}(G') = \mathbf{r}(G) + \mathbf{r}'(j),$$

where, as an initialization step, the average I_n-fitness cost vector $\mathbf{r}(.)$ (i.e. for the trivial tree) has the form $\mathbf{r}(.) = (1 \;\; 2)^{\otimes n}$, and

$$\mathbf{r}'(j) = \bigotimes_{i=0}^{p-1} (1 \;\; \bar{j}_{p-i}) \otimes (1 \;\; 2)^{\otimes(n-p-1)} \otimes (1 \;\; -1),$$

$j = (j_1, j_2, ..., j_p)$, $(j_k \in \{0,1\})$, $\overline{j_k}$ is the negation of j_k, $k = 1, ..., p$, and $(1 \;\; 2)^{\otimes a}$ is the ath Kronecker power of $(1 \;\; 2)$.

Proof follows from the fact that the transform matrix $H_{G'}$ corresponding to the tree $G' = [G, j]$ can be constructed from the matrix H_G by $H_{G'} = H_G D_j$, where D_j is the block-diagonal matrix $D_j = diag(B_0, ..., B_{2^p-1})$ and $B_j = [I_{n-p-1} \otimes (1\;1)^T \;\; I_{n-p-1} \otimes (1\;\;-1)^T]$ and other $B_k, k \neq j$ are the identity matrices I_{n-p}.

In Figure 2 a recurrent construction of the average H_3-fitness cost vector $\mathbf{r}$ for the Haar wavelet packet transforms is shown.

Similar construction for $\mathbf{r}(H)$ can be done also for the Reed-Muller wavelet packets.

Analyzing vectors $\mathbf{r}(H)$ one can see that the minimum number of terms for specifying the low-order schemata (corresponding to the kth elements of the vector $\mathbf{r}(H)$, $\mathbf{k} = (k_1, ..., k_n)$, such that $w_H(\mathbf{k})$, the Hamming weights of $\mathbf{k}$, are high) among the Haar wavelet packet transforms and Reed-Muller wavelet packet transforms is achieved for the Reed-Muller transform (since in this case $r_k = 1.5^{n-w_H(\mathbf{k})}$).

Identity	1	2	2	4	2	4	4	8	+
-	1	-1	2	-2	2	-2	4	-4	+
1	1	0	-1	0	2	0	-2	0	
0	1	1	-1	-1	2	2	-2	-2	+
11	1	0	0	0	-1	0	0	0	
01	1	1	0	0	-1	-1	0	0	
10	1	0	1	0	-1	0	-1	0	
00	1	1	1	1	-1	-1	-1	-1	+
Walsh	8	4	4	2	4	2	2	1	Sum of all the rows from the table
Haar	4	3	4	2	5	3	5	1	Sum of the rows marked by "+"

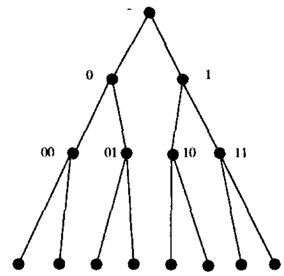

Figure 2. Recurrent construction of an average H-fitness cost vector $\mathbf{r}$ for the Haar wavelet packet transforms

6. CONCLUSION

Some results on the average fitness function in genetic algorithms using wavelet packets have been reported in this paper. The main attentioned focused on the wavelet packet transforms generated from the Haar basis, and the Reed-Muller basis. The octave band decomposition for last case is corresponding to the so-called oblique wavelets investigated in [2].

Simple vector-matrix relations were used for the specifications of an average H-fitness transform for any nonsingular matrix H. Analytical expressions for the average H-fitness transform matrices are obtained.

Comparing the results for the average fitness functions we find that the Reed-Muller-like transforms required shorter sums for low order schemata than the Haar-like transforms.

REFERENCES

[1] S. Agaian, J. Astola, K. Egiazarian, *Binary Polynomial Transforms and Nonlinear Digital Filters*, Marcel Dekker, Inc., 1995.

[2] A. Aldroubi, "Oblique Multiwavelet bases: examples", *SPIE Proceedings*, vol. 2825, 1996, pp. 54-64.

[3] A.D. Bethke, *Genetic algorithms as function optimizers*, Doctoral dissertation, University of Michigan, 1980.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Inc., 1989,

[5] D.E. Goldberg, "Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction", *Complex Systems*, vol. 3, 1989, pp. 129-152.

[6] J.H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.

[7] S. Khuri, "Walsh and Haar Functions in Genetic Algorithms", *In Proceedings of the 1994 ACM Symposium on Applied Computing (SAC'94)*, March 1994, Phoenix AZ, ACM Press, 1994. Genetic Algorithms and Optimization Track.

[8] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, 1995.

[9] Z. Xiong, K. Ramchandran, M.T. Orchard, and K. Asai, "Wavelet Packets-based Image Coding using Joint Space-Frequency Quantization", *In Proceedings of ICIP-94*, vol. 3, Nov. 13-16, 1994, Austin, Texas.