Proceedings of the 6th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China, April 16-18, 2006 (pp119-124)

Time Variant Genetic Algorithm Assisted Code Estimation for Spread Spectrum Systems

VAHID REZA ASGHARI
Department of Electrical Engineering
K.N.Toosi University of Technology
Electrical Faculty, IRAN

MEHRDAD ARDEBILIPOUR
Department of Electrical Engineering
K.N.Toosi University of Technology
Electrical Faculty, IRAN

Abstract: - In the context of spectrum surveillance, a new method to recover the code of spread spectrum signal is presented, while the receiver has no knowledge of the transmitter's spreading sequence. As previous work, a conventional Genetic algorithm (GA) was used to recover spreading code. Although genetic algorithms (GAs) are well known for their robustness in solving complex optimization problems, but nonetheless, by increasing the length of the code, we will often lead to an unacceptable slow convergence speed. To solve this problem we introduce Time Variant Genetic Algorithm (TV-GA) into code estimation in spread spectrum communication system. In searching process for code estimation, the TV-GA algorithm has the merits of rapid convergence to the global optimum, without being trapped in local suboptimum, and good robustness to noise. In this paper we describe how to implement TV-GA as a component of a searching algorithm in code estimation. TV-GA boasts a number of advantages due to the use of mobile agents. Some of them are: Scalability, Fault tolerance, Adaptation, Speed, Modularity, Autonomy, and Parallelism. These properties make TV-GA very attractive for spread spectrum code estimation. They also make TV-GA suitable for a variety of other kinds of channels. Our results compare between Time Variant Genetic algorithm (TV-GA) and conventional Genetic algorithms (GA), and also show time variant Genetic algorithm performance in code estimation process.

Key-Words: - Code estimation, Time Variant Genetic Algorithm (TV-GA), Spread spectrum communications.

1 Introduction

Although spread spectrum communications were initially developed for military applications, they are now widely used for commercial ones, especially for code division multiple access (CDMA), or global positioning systems (GPS) [1]. They are mainly used to transmit at low power without being interfered by jamming, to other users or to multi path propagation. The spread spectrum techniques are useful for secure transmitter, because the receiver has to know the sequence used by the transmitter to recover the transmitter data [2]–[3].

Our purpose is to determine the spreading sequence automatically, whenever the receiver has no knowledge of the transmitter's code. As previous work [4], we used conventional Genetic Algorithm (GA), to recover spreading code. GAs have been used to learn complex behaviors characterized by sets of sequential decision rules and we used them for their robustness in solving complex optimization problem, nonetheless, by increasing the length of the code, we will often lead to an unacceptable slow convergence speed. Hence, we have introduced a new method, which is Time Variant Genetic Algorithm (TV-GA), into code estimation in spread spectrum communication system. In searching process for code estimation, the TV-GA algorithm has the merits of rapid convergence to the global optimum results, and good robustness to noise. In this paper, we describe how to implement TV-GA as a component of a searching algorithm in code estimation. TV-GA boasts a number of advantages due to the use of mobile agents. Some of them are: Scalability, Fault tolerance, Adaptation, Speed, Modularity, Autonomy, and parallelism. These properties make TV-GA very attractive for spread spectrum code estimation. They also make TV-GA suitable for a variety of other kinds of channels.

The code estimation performance of the proposed algorithm is examined by computer simulations. The performance measure of interest in this paper is the mean-squared error (MSE) for the code estimation.

The paper is organized as follows. Section two describes the technique of direct sequence spread spectrum (DS-SS) and explains the difficulty to recover the data in an unfriendly context. Section three describes the system model used in this paper. Sections four and five describe the TV-GA used to implement our proposed code estimator. Our simulation results are presented in section six. Section seven concludes the paper.

2 DS-SS Technique

In order to spread the signal power over a broadband channel, the direct sequence spread spectrum (DS-SS) technique consists in multiplying the information signal with a periodic pseudo-noise sequence.

Let us consider $b(t)$ the information signal

$$b(t) = \sum_{n=-\infty}^{+\infty} b_n \, p(t - nT_b) \tag{1}$$

Where $b_n = \pm 1$ with equal probability and $p(t)$ is a rectangular pulse of duration T_b.

Let us note y, the PN sequence of length k,

$$y = y_0, y_1, \Lambda \;, y_{k-1} \tag{2}$$

The transmitter signal $\hat{y}_n$ is the product of both waveforms. If we consider a direct sequence spread spectrum system without noise,

$$\hat{y}_n = b_n y \tag{3}$$

We assume the receiver knows this sequence and can dispread the signal using a correlator

$$\langle \hat{y}_n, y \rangle = \langle b_n y, y \rangle = b_n \langle y, y \rangle = b_n k \tag{4}$$

According to the properties of PN sequences [5], the data information is then recovered.

However it becomes more challenging when the receiver doesn't know exactly the code used by the transmitter.

Let us note $\tilde{y}$ a sequence similar to y, but not exactly the same. Then using a correlator with $\tilde{y}$, we get

$$\langle \hat{y}_n, \tilde{y} \rangle = \langle b_n y, \tilde{y} \rangle = b_n \langle y, \tilde{y} \rangle \tag{5}$$

According to the properties of PN sequence, $\langle y, \tilde{y} \rangle$ is low [5] and then we can not recover the data information.

3 System Description

Typically direct sequence spread spectrum systems use binary or quadrature phase shift keying (BPSK or QPSK) data modulation. Usually the PN sequence is a binary maximal length sequence or a Gold sequence [3].

Although in this method, we can estimate different PN sequences, but here we consider a BPSK data modulation, spread by a Gold sequence. The baseband noise is assumed to be additive, white, Gaussian, and centered.

An interesting method to estimate spreading code is illustrated in [6]. It takes profit of blind identification techniques available for multiple FIR channels. Also In [4], a conventional Genetic algorithm (GA) was used to estimate PN sequence. In this method which is based on Time Variant Genetic algorithm (TV-GA), we improve the speed of convergence to the global optimum.

4 Time Variant GA overview

Time variant Genetic algorithm has its roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolution strategies [7]. Time variant Genetic algorithm comprises a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed [8].

Time variant Genetic algorithm comprise can be used to solve many of the same kind of problems as genetic algorithms (GAs) [8]. This optimization technique does not suffer, however, from some of GA's difficulties; interaction in the group enhances rather than detracts from progress toward the solution. Further, a time variant GA system has a memory, which the genetic algorithm does not have. Change in genetic populations results in destruction of previous knowledge of the problem, except when elitism is employed, in which case usually one or a small number of individuals retain their "identities". In TV-GA, individuals who fly past optima are tugged to return toward them; knowledge of good solutions is retained by all particles [9].

Time variant GA is also similar to Swarm intelligence which appears in biological swarms of certain insect species. It gives rise to complex and often intelligent behavior through complex interaction of thousands of autonomous swarm members [10]. The main principle behind these interactions is called stigmergy, or communication through the environment. An example is pheromone laying on trails followed by ants. Pheromone is a potent form of hormone that can be sensed by ants as they travel along trails. It attracts ants and therefore ants tend to follow trials that have high pheromone concentrations. This causes an autocatalytic reaction, i.e., one that is accelerated by itself. Ants attracted by the pheromone will lay more of the same on the same trail, causing even more ants to be attracted [10].

Time variant Genetic algorithm boasts a number of advantages due to the use of mobile agents and stigmergy. These are:

1. Scalability: Population of the agents can be adapted according to spreading code size.
2. Fault tolerance: TV-GA processes do not rely on a centralized control mechanism. Therefore the loss of a few bits or frames does not result in catastrophic failure, but rather leads to graceful, scalable degradation.
3. Adaptation: Agents can change, die or reproduce, according to the length of the code changes. But here, we supposed the length of the code is constant.
4. Speed: Changes in the systems can be modified very fast.
5. Modularity: Agents act independently of other codes of users. It can be used for multiuser systems.
6. Autonomy: Little or no human supervision is required.
7. Parallelism: Agent operations are inherently parallel.

These properties make time variant Genetic algorithm very attractive for spread spectrum code estimation.

5 TV-GA Technique In The Code Estimation

The TV-GA algorithm has proved to be very effective in solving global optimization for multidimensional problems in static, noisy, and continuously changing environments [11]. We introduced for the first time the GA technique into spread spectrum code estimation in our previous work [4], and now, we use time variant GA technique, which has some properties does not exist in conventional GA technique.

In reality, TV-GA and GA techniques are too similar and by making some changes to conventional GA's algorithm, you have your time variant Genetic algorithm. At the beginning, the time variant Genetic algorithm randomly initializes a population of individuals (called chromosomes). Each particle represents a single intersection of spreading code. The particles evaluate their position relative to a goal at every iteration. In each iteration, every particle of the code sequence adjusts its trajectory toward its own previous best position, and toward the previous best position attained by any member of its topological neighborhood. If any particle's position is close enough to the goal function, it is considered as having found the global optimum and the recurrence is ended. Generally, there are two kinds of topological neighborhood structure, corresponding to the global version of time variant GA (GTV-GA), and local neighborhood structure, corresponding to the local version of time variant GA (LTV-GA). For the global neighborhood structure, the whole population is considered as the neighborhood, while for the local neighborhood structure, some smaller number of adjacent members in subpopulation is taken as the neighborhood [12].

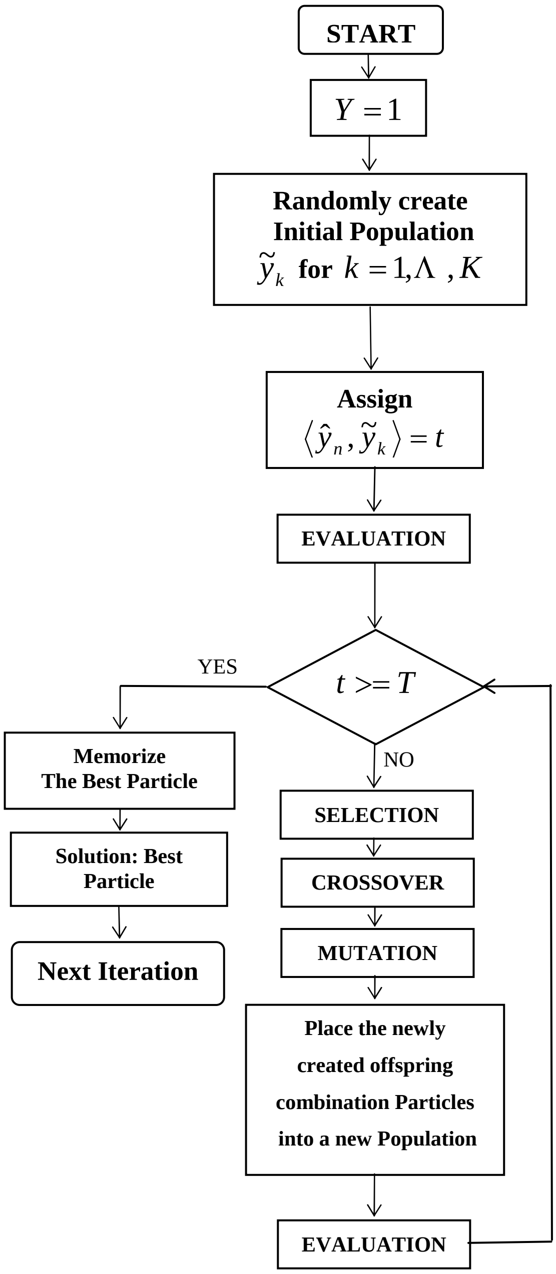

Figure 1. Flowchart depicting the structure of the proposed time variant Genetic algorithm (TV-GA) used to code estimation.

The detail of process for implementing the GTV-GA can be found in [12]. In the global neighborhood structure, each particle's search is influenced by the best position found by any member of the entire population.

Proceedings of the 6th WSEAS International Conference on Multimedia Systems & Signal Processing, Hangzhou, China, April 16-18, 2006 (pp119-124)

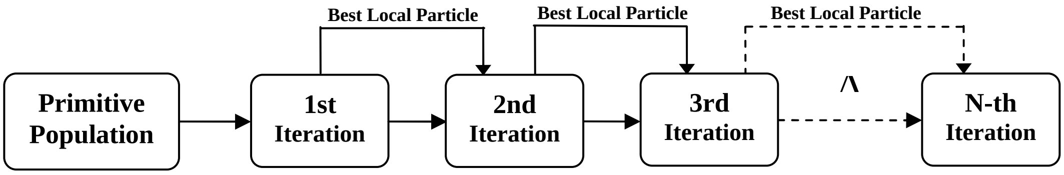

Figure 2. Flowchart depicting the structure of the iterations

In contrast, each particle in the local neighborhood structure is influenced only by parts of the adjacent members. Therefore, the LTV-GA has fewer opportunities to be trapped in suboptimum than the GTV-GA.

Generally, the larger the number of particles adopted in TV-GA, the fewer the opportunities to be trapped in suboptimum, but the greater the time spent searching for the global optimum. In our experiment, 40 particles are used in LTV-GA, which is a balance between the accuracy required in searching for the global optimum and time consumed. This procedure, whose flowchart is shown in Fig. 1, is iterated a predefined number of consecutive particles.

5.1 Initialization

Initialization of the TV-GA is performed at the so-called $(y = 1)$st generation for the first signaling interval, as seen in Fig. 1, by creating p number of candidate solutions, or particles in TV-GA parlance. For the others iteration, we just use the population of previous iteration. The set of p particles is known as a generation, and p is known as the population size. These particles represent the unknown variables of interest, which in this case are the estimated PN sequence. Hence, each particle will contain k elements corresponding to the length of the PN sequence.

5.2 Evaluation

Associated with the p th combination particle is a so-called figure of merit — more commonly known in TV-GA as the fitness value — which has to be evaluated, as seen in Fig. 1. The fitness value, denote by $f[\hat{y}_n, \tilde{y}_k]$ for $k = 1,\Lambda, K$ is computed by substituting the elements of both the transmitted string and the k th candidate solution into the objective function or crosscorrelation of (5).

5.3 Selection

The exploitative property of TV-GA (and also conventional GA) is derived from two operators referred to as selection and crossover [9]. The crossover operation will be explained in the next subsection. Let us refer to the elements that constitute the optimal solution as good particles. Any other elements are referred to as bad particles. For example, if the optimal solution constitutes a particle containing all $+1$ elements, then any $+1$ in a particle will be a good element while any -1 in the particle will be a bad element.

Intuitively, particles having a high fitness value in the sense of (5) will contain more good elements and hence should be exploited further. At the same time, particles having a low fitness value should be discarded. As shown in Fig. 1, following the evaluation, our population of particles is sorted according to their fitness value. Then, the particles which are located at the top level of sorted population will be memorized and used for subsequent exploitation and exploration of the solution space.

5.4 Crossover and Mutation

Crossover and mutation are two different operators which produce one or more new particles. Crossover applies to one or more parents and exchange particle elements (good or bad) with equal probability (p_c) between two different particles (offspring), will constitute the new population of the next generation. In Fig. 1, the mutation operation refers to the alteration of the value of each particle in the offspring with a probability denoted by p_m.

In the case of the data string, the mutation process simply inverts the bit value of the element concerned from $+1$ to -1 or vice versa. Then these offspring are later made a new generation which can select as parents.

The TV-GA algorithm is terminated if there is no improvement in the maximum fitness value of the population, in each iteration. Hence we terminate the algorithm of Fig. 1 after $\langle \hat{y}_n, \tilde{y}_k \rangle$ is coming upper than a threshold (T). By adjusting the value of T, the bit error rate (BER) performance of the time variant GA-based code estimator can be controlled. In each iteration, as showed in Fig. 2, the TV-GA uses the previous best particles which were memorized in previous iteration. In this algorithm the rate of convergence and adaptation is increased.

6 Simulation Results

In this section, our simulation results are presented in order to demonstrate the performance of the proposed code estimator. A summary of the various parameters that are used in our simulations is shown in Table 1. The channel noise was assumed to be additive, white, Gaussian, centered and real. The data rate (R_b) and the number of chips per bit (p) were assumed to be known by the receiver. The PN sequence was used with a processing gain of $p = 31$.

TABLE 1
SUMMARY OF VARIOUS PARAMETERS USED IN OUR SIMULATIONS

Symbol	Description
R_b	Data rate
p	Population size
p_c	Crossover probability
p_m	Mutation probability
Y	Number of generation per iteration
k	Number of Particles in each Population

In order to give an impression of how the TV-GA manages to estimate the transmitted code over the course of iterations given a population of randomly generated possible solutions at the beginning, the best fitness value of particles in our population in some iterations is shown in Fig. 3 at $\xi / N_0 = -5\text{db}$.

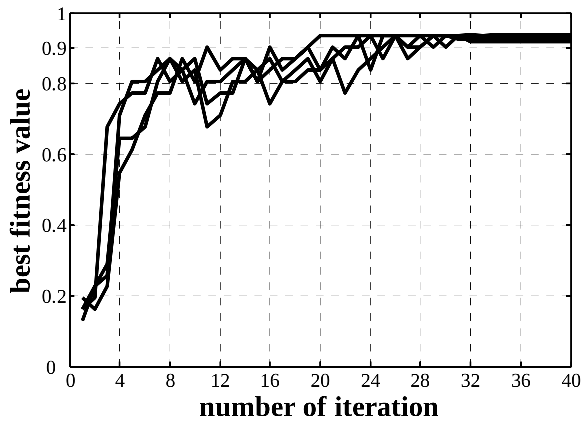

Fig. 3. Best fitness values of the proposed time variant GA-based estimator for three different try, over forty iterations, $p = 100$, $Y = 10$, and $SNR = -5dB$, $p_c = 0.2$, and $p_m = 0.1$.

As we have mentioned in section five, the time variant Genetic algorithm will efficiently identify the areas in the solution space, where the optimal solution might be located. Fig. 3 shows that the entire final searched fitness values in any code estimation process exceed 0.94 for LTV-GA used as the optimization algorithm. Furthermore, entire the fitness values reach 0.9 within about thirty iterations for LTV-GA.

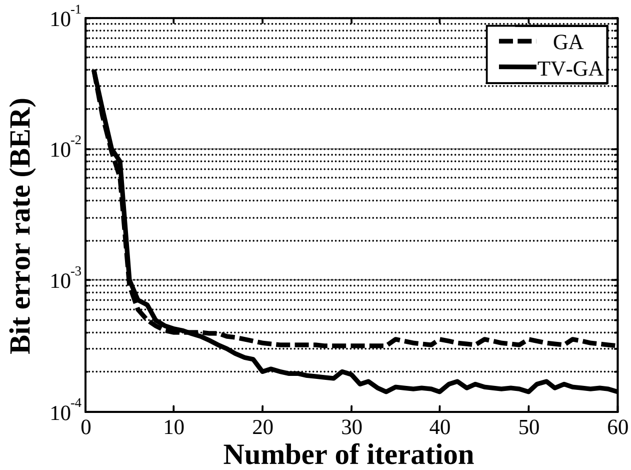

Fig. 4. BER performance of the proposed PSO-based code estimator in compare with GA-based, over 60 iterations where, $p = 100$, $Y = 10$, $p_c = 0.2$, and $p_m = 0.1$.

Fig. 4 characterizes the BER performance of proposed estimator in compare with a receiver which uses conventional genetic algorithm (GA) method to estimate the spreading code [4]. It can be seen that the bit error rate (BER) performance of time variant GA-based code estimator is better than conventional GA-based one. It is because in conventional GA-based code estimator, we try to find the spreading code which has the best fitness value, just in first iteration. Hence, it takes a lot of time [4]. In contrast, in time variant GA-based code estimator, step by step, we find the spreading code over the course of iterations. In fact, at each iteration, we try to find the code which has the best local fitness value. Hence, it is faster compared with conventional GA-based algorithm.

7 Conclusion

For the first time, we have introduced the time variant Genetic algorithm into spread spectrum code estimator, which showed the desirable features of rapid convergence to the global optimum without being trapped in local suboptimum and robustness to noise. Time variant Genetic algorithm is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions. Much further research remains to be conducted on this new concept. The goal in developing it has been to use this system in fading channels.

Acknowledgment

The authors are grateful for the constructive critique by the anonymous reviewers.

References:

[1] D. Thomas Magill, Francis D. Natali, Gwyn P. Edwards, “Spread Spectrum Technology for Commercial Applications,” *Proceeding of the IEEE*, vol. 82, April. 1994, pp. 572–584.

[2] Raymond. L. Picholtz, Doland L. Schilling, Laurence B. Milstein, “Theory of Spread Spectrum Communications – A Tutorial,” *IEEE Transactions on Communications*, vol. COM-30, May. 1982, pp. 855–884.

[3] John G. Proakis, *Digital communication*, Third Edition, Mac Graw Hill International Editions, 1995.

[4] V. R. Asghari and M. Ardebilipour, “Spread Spectrum Code Estimation by Genetic Algorithms,” *International Journal on Signal Processing*, vol. 1, Dec. 2004, pp. 30-33.

[5] Dilip V. Sarwate, Michael B. Pursley, “Crosscorrelation Properties of Pseudo-random and Related Sequences,” *Proceeding of the IEEE*, vol. 68, May. 1980, pp. 593–619.

[6] Michail K. Tsatsanis, Georgios B. Giannakis, “Blind Estimation of Direct Sequence Spread Spectrum Signals in Multipath,” *IEEE Transactions on Signal Processing,* vol. 45, May. 1997, pp. 1241–1252.

[7] T. Baeck, “Generalized convergence models for tournament and (mu,lambda)-selection,” *Proc. of the Sixth International Conf. on Genetic Algorithms*, pp. 2-7, Morgan Kaufmann Publishers, San Francisco, CA, 1995.

[8] M. Potter, K. De Jong, and J. Grefenstette, “A coevolutionary approach to learning sequential decision rules,” *Proc. of the Sixth International Conf. on Genetic Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, 1995, pp. 366-372.

[9] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *Proc. 6th Int. Symp. Micro Machine Human Sci.*, 1995, pp. 39–43.

[10] E. Bonabeau, M. Dorigo, and G. Théraulaz, *Swarm intelligence: from natural to artificial systems*, Oxford University Press, 1999.

[11] H. Plotkin, *Darwin Machines and the Nature of Knowledge*, Harvard University Press, Cambridge, MA, 1994.

[12] K. S. Tang, K. F. Man, S. Kwong, “GA approach to time-variant delay estimation,” *Proc. Int. Conf. on Control and Information*, Hong Kong, 1995, pp. 173-175.