Evolutionary Algorithms Based Speed Optimization of Servo Motor in Optical Disc Systems

Radha Thangaraj[1], Millie Pant[1] and Ajith Abraham[2]
[1]*Indian Institute of Technology Roorkee, India*
[2]*Norwegian University of Science and Technology, Norway*
t.radha@ieee.org, millifpt@iitr.ernet.in, ajith.abraham@ieee.org

Abstract

Evolutionary Algorithms are inspired by biological and sociological motivations and can take care of optimality on rough, discontinuous and multimodal surfaces. During the last few decades, these algorithms have been successfully applied for solving numerical bench mark problems and real life problems. This paper presents the application of two popular Evolutionary Algorithms (EA); namely Particle Swarm Optimization (PSO) and Differential Evolution (DE) for optimizing the average bit rate of an optical disc servo system. Two optimization models are considered in the present study subject to the various constraints due to servo motor. The results obtained by PSO and DE are compared with the experimental and the design results given in the literature. Simulation results clearly show the superior performance of PSO and DE algorithms.

1. Introduction

The Electrical Engineering community has shown a significant interest in optimization for many years [1] – [4]. In particular, there has been a focus on global optimization of numerical, real- valued problems for which exact and analytical methods do not apply. During the last few decades, many general-purpose optimization algorithms have been proposed for finding optimal solutions, some of which are; Evolution strategies [5], evolutionary programming [6], Genetic algorithms (GA) [7], Particle Swarm Optimization (PSO) [8] and Differential Evolution (DE) [9]. These algorithms are also known as Evolutionary Algorithms (EAs) or Nature Inspired Algorithms because they follow simple rules of nature. These algorithms have also become popular because of their advantages over the traditional optimization techniques (decent method, quadratic programming approach, etc). They have been successfully applied to a wide range of engineering optimization problems [10] - [14] etc. In this study we investigate the performance of PSO and DE for optimizing the average bit rate of an optical disc servo system.

Servo motor is one of the most sophisticated motion control devices in electric motors. In CD-ROM or DVD ROM, the objective function mainly consists of maximizing the average bit transfer rate subject to various constraints due to servo motor, control circuits decoding electronics etc. In the present article we have taken a popular yet complex problem of CD/ DVD ROM systems where the objective is to optimize the speed of the servo motor. We considered two cases of optimization (i) average bit rate in seeking and (ii) average bit rate of zoned CLV. The mathematical models are taken from Jung and Sheu [15]. A preliminary version of this study was already presented by Pant et al [16], but in the present study more elaborated analysis is given.

The structure of the paper is as follows; Section 2 briefly describes the mathematical models of the optimization problems. Section 3, gives a general introduction to PSO and DE algorithms. In Section 4, the penalty approach for handling constraints is discussed. Section 5 gives the parameter settings and numerical results; finally this paper concludes with section 6.

.

2. Mathematical Model of the Problem

With the boost in CD/ DVD ROM market, the demand of smaller access times and higher data rates are also increasing. The design of a CD servo system employs a constant linear velocity (CLV) strategy in which the disc is rotated at a varying rotation speed to maintain a synchronized velocity between the pickup head and track across the disc radius [17], [18]. Several adaptive speed algorithms for CD ROM systems are proposed in [19]. The objective of the present paper is to observe the effectiveness of PSO and DE algorithms for solving two different optimization models to maximize the average bit rate, subject to various constraints imposed by the servo motor, control circuits, decoding electronics and other limitation factors. The model of the design of the speed profile has been adopted from Jung and Sheu [15].

2.1. Speed Profile in Seeking

One advantage of adaptive speed control in optical disc servo system is reduction of seek time and hence access time during seek motion. Typically there are two simultaneous control activities involved in seek motion. The spindle motor is controlled to adjust the speed and the sledge motor (as well as the voice coil) is commanded to move the optical head to the desired position. If the speed profile is well designed the system is able to read out data for processing once the pickup head is in its target position. Otherwise, the access time will be increased.

Maximize

$$B_{avg} = \frac{2\pi B_0}{Lpv_o}\frac{n-1}{12(R_{out}-R_{in})}.$$
$$\{\omega_1[3R_1^4 - 4R_1^3R_2 + 3R_2^4] +$$
$$\sum_{i=2}^{n-1}\omega_i[R_{i+1}^4 - 4R_{i+1}R_i^3 + 6R_i^4 - 4R_{i-1}R_i^3 + R_{i-1}^4]$$
$$+\omega_n[R_{n-1}^4 - 4R_n^3R_{n-1} + 3R_n^4]\} \qquad (1)$$

Where

$$L = \frac{\pi(R_{out}^2 - R_{in}^2)}{p} \qquad (2)$$

$$R_i = R_{in} + ((R_{out} - R_{in})/(n-1))(i-1) \qquad (3)$$

The objective function is a linear function with unknown vector ω.

Subject to the constraints:

$$\omega_i \le \omega_{max} \quad i = 1,2,....,n \qquad (4)$$

$$N_{min}v_o \le R_j\omega_i e^{-(\beta|Rj - Ri|)/Jvh}$$
$$+ R_j\omega_j[1 - e^{-(\beta|R_j - R_i|)/Jv_h}] \le N_{max}v_o \qquad (5)$$

2.2. Zoned Constant Linear Velocity Control (CLV)

In zoned CLV control, the disc area is partitioned into m zones $R_1 = R_{in}$, R_2, R_3… R_m. Zoned CLV assumes that within each zone the linear velocity or over speed factor is fixed. If Ni denotes the over speed factor in the i^{th} zone, then the rotation speed is given as

$$\omega(r) = \frac{N_i v_0}{r} \quad \text{for} \quad R_i \le r \le R_{i+1} \qquad (6)$$

The objective function for maximizing the average bit rate is given as

$$B_{avg} = \frac{2\pi B_0}{Lpv_0}\sum_{i=1}^{m}\int_{Ri}^{Ri+1}\omega(r)r^2dr$$
$$= \frac{\pi B_0}{Lp}\sum_{i=1}^{m}N_i(R_{i+1}^2 - R_i^2) \qquad (7)$$

Subject to the constraints

$$N_i v_0 \le \omega_{max}R_i \quad \forall i \qquad (8)$$

$$N_{min} \le N_i \le N_{max} \quad \forall i \qquad (9)$$

$$\frac{N_i v_0}{r}\left[-\alpha\frac{N_i v_0}{r^2} + \beta\right] \le \frac{K_m}{R_\alpha}u_{max} \quad \forall i \qquad (10)$$

$$N_{min} \le N_j + (\frac{r_j}{r_i}N_i - N_j)e - (\beta / Jv_h)|r_i - r_j|$$
$$\le N_{max} \qquad (11)$$

Table 1 Design Parameters

Parameter	Value	Unit
Rin	25	mm
Rout	58	mm
B_0	150	KB/sec
p	1.6	µm
v_0	1.3	m/sec
v_h	12	mm/sec
K_m	0.0062	Nm/A
K_b	0.0062	Nm/A
R_a	5	Ω
D	5.0×10^{-6}	Kg m^2
N_{min}	12	-
N_{max}	24	-
ω_{max}	7500	rpm

3. Particle Swarm Optimization and Differential Evolution

PSO and DE algorithms may be termed as general purpose algorithms for solving optimization problems. Both of these methods are assisted with special operators that are based on some natural phenomenon. These algorithms are iterative in nature and in each iteration the operators are invoked to reach to optimal (or near optimal) solution. Pseudo codes of all the algorithms used in this study are given in Appendix A. A brief description of PSO and DE are given in the following subsections:

3.1. Particle Swarm Optimization

PSO was proposed by Kennedy and Eberhart in 1995 [8]. It is inspired by the complex socio cooperative behavior displayed by various species like flocks of birds and shoals of fish. In PSO, the members of the swarm or the particles are placed in the parameter space of a particular problem, and each particle evaluates the fitness at its current location. The movement of each particle in space is determined by the history of its own fitness and also by the fitness of its neighbors. It then moves through the parameter space with a velocity determined by the locations and processed fitness values of those other members, along with some random perturbations. The members of the swarm that a particle can interact with are called its social neighborhood. Together the social neighborhoods of all particles form a social network of PSO.

For a D-dimensional search space the position of the i^{th} particle is represented as $X_i = (x_{i1}, x_{i2},..x_{iD})$. Each particle maintains a memory of its previous best position $P_i = (p_{i1}, p_{i2}\ldots p_{iD})$ and a velocity $V_i = (v_{i1}, v_{i2},\ldots v_{iD})$ along each dimension . At each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector are given by:

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (12)$$

$$x_{id} = x_{id} + v_{id} \quad (13)$$

The first part of equation (12) represents the inertia of the previous velocity, the second part is tells us about the personal thinking of the particle and the third part represents the cooperation among particles and is therefore named as the social component [20]. Acceleration constants c_1, c_2 [21] and inertia weight ω [22] are predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

3.2. Differential Evolution

Differential evolution (DE) is an Evolutionary Algorithm (EA) proposed by Storn and Price in 1995 [9]. DE is similar to other EAs particularly Genetic Algorithms (GA) [23] in the sense that it uses the same evolutionary operators like selection recombination and mutation like that of GA. However the significant difference is that DE uses *distance* and *direction* information from the current population to guide the search process. The performance of DE depends on the manipulation of *target vector* and *difference vector* in order to obtain a *trial vector.*

A general DE variant may be denoted as DE/X/Y/Z, where X denotes the vector to be mutated, Y specifies the number of difference vectors used and Z specifies the crossover scheme which may be binomial or exponential. For the more details the interested reader may please refer to [24]. In this study, the mutation strategy DE/rand/1/bin [9] is considered. It is also known as the classical version of DE and is perhaps the most frequently used version of DE. DE works as follows: First, all individuals are initialized with uniformly distributed random numbers and evaluated using the fitness function provided. Then the following will be executed until maximum number of generation has been reached or an optimum solution is found.

In a D-dimensional search space, for each target vector $x_{i,g}$, a mutant vector is generated by

$$v_{i,g+1} = x_{r_1,g} + F * (x_{r_2,g} - x_{r_3,g}) \quad (14)$$

where $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, which are different from each other and also different from the running index i. F (>0) is a scaling factor which controls the amplification of the differential vector $(x_{r_2,g} - x_{r_3,g})$.

Once mutation phase is complete, crossover is introduced in order to increase the diversity of the perturbed parameter vectors. The parent vector is mixed with the mutated vector to produce a trial vector $u_{ji,g+1}$,

$$u_{j,i,g+1} = \begin{cases} v_{j,i,g+1} & if \quad rand_j \leq Cr \vee j = k \\ x_{j,i,g} & otherwise \end{cases} \qquad (15)$$

where j , k ∈ {1, 2,……, D}; k is a random parameter index, chosen once for each i . $rand_j \in [0,1]$; Cr is the crossover constant takes values in the range [0, 1].

Finally selection takes place where a tournament is held between the target vector and trial vector and the one with better fitness function is allowed to enter the next generation. In this way individuals in a new generation are as good as or better than the individuals in the previous generation.

4. Penalty Approach for Handling Constraints

Many real-world optimization problems are solved subject to sets of constraints. The search space in Constrained Optimization Problems (COPs) consists of two kinds of solutions: feasible and infeasible. Feasible points satisfy all the constraints, while infeasible points violate at least one of them. Therefore the final solution of an optimization problem must satisfy all constraints.

In this paper, the two algorithms PSO and DE handle the constraints using the concept of penalty functions. In the penalty function approach, the constrained problem is transformed into an unconstrained optimization algorithm by penalizing the constraints and building a single objective function, which is minimized using an unconstrained optimization algorithm.

That is,

$$F(x) = f(x) + \lambda \ p(x) \qquad (16)$$

Where $$p(x_i, t) = \sum_{m=1}^{n_g + n_h} \lambda_m(t) p_m(x_i) \qquad (17)$$

$$p_m(x_i) = \max\{0, g_m(x_i)^\alpha\}$$

if $m \in [1,......, n_g]$ (inequality)

$$p_m(x_i) = |h_m(x_i)|^\alpha$$

if

$m \in [n_g + 1,......, n_g + n_h]$ (equality)

with α a positive constant, representing the power of the penalty. The inequality constraints are considered as g(x) and h(x) represents the equality constraints. n_g and n_h denotes the number inequality and equality constraints respectively. λ is the constraint penalty coefficient.

5. Experimental Settings and Results

In this section we analyze the numerical results obtained after applying the PSO and DE algorithms to the optimization models given in Section 2. It can be seen that both the problems are constrained in nature. For handling constraints, penalty approach is used. We would like to point out that the authors in [15] have suggested the use of Linear Programming approach for solving the speed profile in seeking and semi-infinite approach (SIA) for solving CLV control. However in the present article we used PSO and DE for solving both the cases.

There are certain parameters associated with PSO and DE which require a proper setup for the optimum performance of the algorithm. The population size is fixed at 30 for both the algorithms. Inertia weight w is taken as linearly decreasing, which starts at 0.9 and ends at 0.4. The acceleration constants c_1, c_2 are fixed at 2.0. The DE parameter Cr = 0.5 and F = 0.5. For each algorithm, the stopping criteria is to terminate the search process when one of the following conditions is satisfied: (1) the maximum number of generations is reached (assumed 500 generations), (2) $| f_{max} - f_{min} | < 10^{-4}$ where f is the value of objective function. Since PSO and DE are stochastic techniques, more than one run is required to ascertain the final solution. In the present article, PSO and DE algorithms are executed 30 times and the best value throughout the run is recorded.

Table 2 Results of PSO and DE for *Seeking*

Number of Segments	Bit rate (KB/sec)	
	PSO	DE
5	4194.126987	4194.145646
6	4227.906719	4227.771129
7	4298.238376	4298.22215
8	4382.221168	4382.221215
9	4478.242506	4478.24225
10	4582.023881	4582.02379

The numerical results of both the problems are given in Table 2 – 5. Figures 1 – 4 show the performance of PSO and DE algorithms for Seeking and Zoned CLV. From the numerical results of Table 2, we can see that PSO and DE gave more or less same results in terms of objective function values i.e. in terms of Bit rate. But if we compare the average number of generations and time taken by the algorithms for seeking then DE performs better than

PSO in 3 cases out of 6 cases tried. In the remaining 3 cases PSO gave better results than DE. Likewise for zoned CLV, the results given by PSO are better than DE in 3 test cases out of 4 cases. If we compare the results of PSO and DE for both the problems (Seeking and Zoned CLV) with the results in the literature [15] then both the algorithms perform better than the results quoted in the literature.

Table 3 Average number of generations and run time taken by PSO and DE for *Seeking*

Number of Segments	PSO		DE	
	Generation	Run time (sec)	Generation	Run time (sec)
5	500+	0.8	268	0.5
6	500+	1.4	300	0.7
7	500+	1.7	324	1.0
8	257	1.1	313	1.3
9	268	1.5	337	1.8
10	276	1.8	370	2.5

Table 4 Results of PSO and DE for *Zoned CLV*

Number of Zones	Bit rate (KB/sec)		
	PSO	DE	Results in [15]
5	2829.854007	2821.483415	2741.313
6	2887.411097	2880.843368	2770.959
7	2912.814069	2913.969309	2775.275
8	2946.866655	2943.485306	2793.640

Table 5 Comparison Results of PSO and DE

Speed Profile	PSO	DE	Results in [15]
Zoned CLV (Bit rate KB/sec)	2887.411097	2880.843368	2770.959
Seeking	4227.906719	4227.771129	2885.414

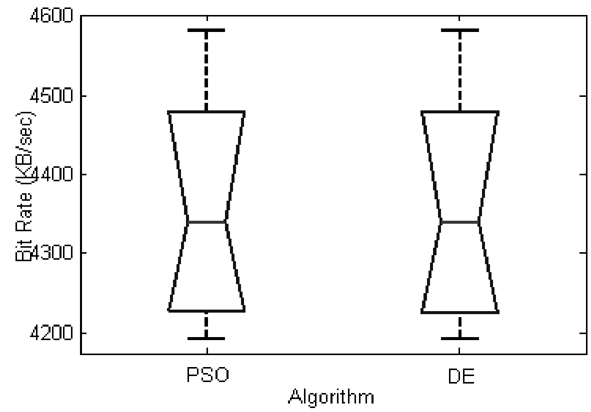

Figure 1 Comparison of PSO and DE in terms of Bit rate for Seeking

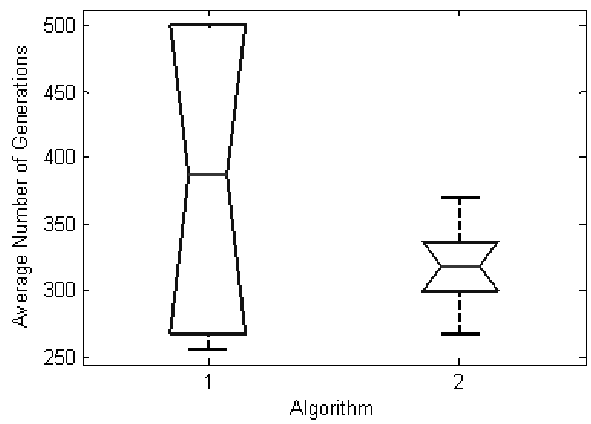

Figure 2 Comparison of PSO and DE in terms of Average number of generations for Seeking

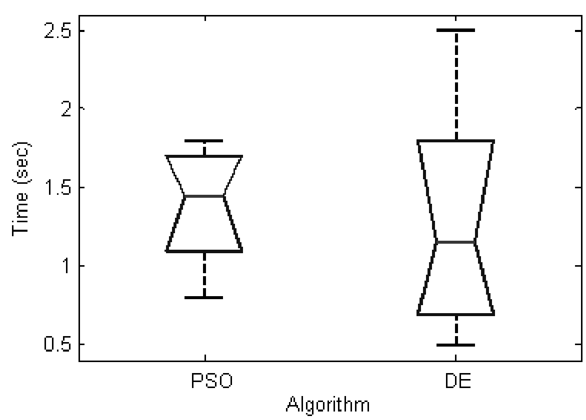

Figure 3 Comparison of PSO and DE in terms of total time for Seeking

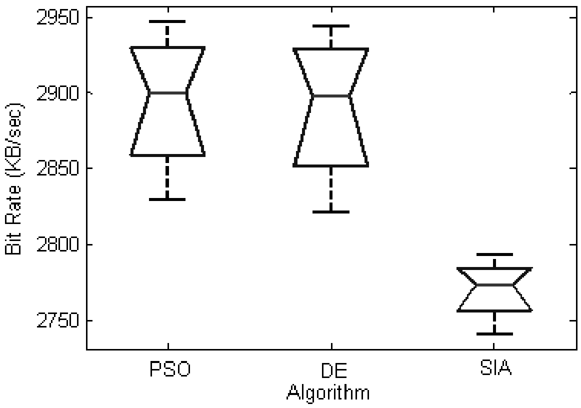

Figure 4 Comparison results of Zoned CLV

6. Conclusion

This paper discusses the use of PSO and DE algorithms for maximizing the average bit rate of optical disc servo systems. Two cases are considered. The first one being a linear model of optimizing the average bit rate in seeking and the second case being zoned CLV control. The objective function considered in both the cases is to maximize the average bit rate subject to various constraints due to Servo motor. Simulation results clearly show the superior performance of PSO and DE which enhanced the speed of data transfer (KB/sec) by programming optimum speed in the servo motor. However we would like to add that we have used a basic version of PSO and DE whereas much advanced versions are available in literature. It is very much possible that using the more sophisticated versions, the performance can be further improved. Besides PSO there are other general purpose algorithms like Genetic Algorithms, Evolutionary Programming, Ant Colony Optimization etc. which may be used for comparison.

7. Appendix

Pseudo codes of algorithms used in this study:

Pseudo code for Particle Swarm optimization

Step1: Initialization.
For each particle i in the population:
Step1.1: Initialize X[i] with Uniform distribution.
Step1.2: Initialize V[i] randomly.
Step1.3: Evaluate the objective function of X[i], and assigned the value to fitness[i].
Step1.4: Initialize P_{best}[i] with a copy of X[i].
Step1.5: Initialize Pbest_fitness[i] with a copy of fitness[i].
Step1.6: Initialize P_{gbest} with the index of the particle with the least fitness.
Step2: Repeat until stopping criterion is reached:
For each particle i:
Step 2.1: Update V[i] and X[i] according to equations (1) and (2).
Step2.2: Evaluate fitness[i].
Step2.3: If fitness[i] < Pbest_fitness[i] then P_{best}[i] =X[i], Pbest_fitness[i] =fitness[i].
Step2.4: Update P_{gbest} by the particle with current least fitness among the population.

Pseudo code for Differential Evolution

Initialize the population
Calculate the fitness value for each particle
Do
For i = 1 to number of particles
Do mutation, Crossover and Selection
End for.
Until stopping criteria is reached.

8. References

[1] G D. H. Kim, "GA-PSO based vector control of indirect three phase induction motor," Applied Soft Computing, Vol. 7, No.2, 2006, pp. 601-611.

[2] B. Pryymak, et al., "Neural network based flux optimization using a model of losses in induction motor drives," Mathematics and computers in simulation, Vol. 71, 2006, pp. 290-298.

[3] S. Ghozzi , K. Jelassi, X. Roboam, "Energy optimization of induction motor drives," IEEE conference on Industrial Technology (ICIT), 2004, pp. 602-610.

[4] S. Lim, K. Nam., "Loss minimization control scheme for induction motors," IEE proc. Electr. Power appli., Vol. 151, No. 4, 2004, pp. 385-397.

[5] I. Rechenberg, "Evolution Strategy: Optimization of Technical systems by means of biological evolution", Fromman-Holzboog, 1973.

[6] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through a simulation of evolution", In M. Maxfield, A. Callahan and L. J. Fogel, editors, Biophysics and Cybernetic systems. Proc. of the 2nd Cybernetic Sciences Symposium, pp. 131 – 155, Spartan Books, 1965.

[7] Holland, J. H., "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence," Ann Arbor, MI: University of Michigan Press.

[8] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization," IEEE International Conference on Neural

Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, pp. IV: 1942-1948.

[9] R. Storn and K. Price, “Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces”, Technical Report, International Computer Science Institute, Berkley, 1995.

[10] M. Pant, R. Thangaraj and V. P. Singh, “Efficiency Optimization of Electric motors: A Comparative Study of Stochastic Algorithms”, World Journal of Modeling and Simulation, Vol. 4(2), 2008, pp.140 – 148.

[11] M. Pant, R. Thangaraj and A. Abraham, “Optimization of a Kraft Pulping System: Using Particle Swarm Optimization and Differential Evolution”, 2nd Asia Int. Conf. on Modeling and Simulation (AMS’08), Malaysia, IEEE Computer Society Press, USA, 2008, pp. 637 – 641.

[12] M. Pant, R. Thangaraj and A. Abraham, “Optimal Tuning of PI Speed Controller in PMSM: A Comparative Study of Evolutionary Algorithms”, Journal of Electrical Engineering, Vol. 2(1), 2008, pp. 36 – 43.

[13] M. Pant, R. Thangaraj and V. P. Singh, “Optimization of Mechanical Design Problems Using Improved Differential Evolution Algorithm”, Int. Journal of Recent Trends in Engineering, Vol. 1(5), 2009, pp. 21 – 25.

[14] D. H. Kim, “GA-PSO based vector control of indirect three phase induction motor,” Applied Soft Computing, Vol. 7, No.2, 2006, pp. 601-611.

[15] J-C. Juang and J-S. Sheu, “On Adaptive Speed Design in Optical Disc Servo Systems”, IEEE Trans. on Control Systems Technology, Vol. 8(6), 2001, pp. 971 – 978.

[16] M. Pant, R. Thangaraj and V. P. Singh, “Speed Optimization of Servo Motor in Optical Disc Systems Using Particle Swarm Optimization”, POWERCON 2008 & 2008 IEEE Power India Conference, India, 2008, pp. 1 – 4.

[17] K. C. Pohlmann, “The Compact Disc Handbook”, A-R Editions., 1992.

[18] J. Watkinson, “The Art of Data Recording: Focal Tress”, 1994.

[19] S. G. Stan and J. L. Bakx, “Adaptive-speed algorithms for CD-ROM systems,” IEEE Trans. Consumer Electron., vol. 42, no. 1, 1996, pp. 43–51.

[20] J. Kennedy, “The Particle Swarm: Social Adaptation of Knowledge”, IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), IEEE Service Center, (Piscataway, NJ, 1997), pp. 303-308.

[21] R.C. Eberhart, Y. Shi, “Particle Swarm Optimization: developments, Applications and Resources”, In Proc. of IEEE Int. Conference on Evolutionary Computation, 2001, pp. 81 -86.

[22] Y. Shi, R.C. Eberhart, “A Modified Particle Swarm Optimizer”, IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, 1998, pp. 69 – 73.

[23] D. Goldberg, “Genetic Algorithms in Search Optimization and Machine Learning”, Addison Wesley Publishing Company, Reading, Massachutes, 1986.

[24] http://www.icsi.Berkley.edu/~storn/code.html