I J C T A, 9(28) 2016, pp. 79-86

A NOVEL APPROACH FOR E-LEARNING USING QPSO ALGORITHM

S. Selvakumara Samy* and Latha Parthiban**

Abstract: Particle swarm optimization has always played a major role in solving problems related to global optimization. A different PSO technique for the purpose of e-learning is proposed. Instead of following Newtonian principles for the PSO, the particles have quantum behavior, thus, leading to the use of quantum-based particle swarm optimization (QPSO). E-learning is one of the most rising technologies as of today. It has got several advantages than the classical approach of education. Students no more need to depend on their teachers entirely to know about a particular topic or their areas of interest. Moreover, industrial training has also become much easier due to e-learning. So, in order to get the maximum advantage of e-learning the student or a company employee should be able to select the set of courses that suit him best. Depending on the person's interests and also on the basis of the courses previously chosen by him/her, a recommendation system can be designed which will help them choose courses that suit his/her requirements. The optimization in this case can be done best using QPSO. However there are several other techniques like genetic algorithms that can be used for the same purpose. There are a few drawbacks of using genetic algorithms for this purpose. First of all, genetic algorithms require a lot of parameters to function properly and render the correct analysis. Secondly, genetic algorithms, at some points, are unable to guarantee that the global optimum solution has been found. For an e-learning system, mistakes have to be kept at a minimum rate. So, QPSO is a better approach considering these factors. QPSOs optimize a problem by going through several iterations, with the sole intent of reaching the best and optimized solution. The best thing about PSOs is that they are very easy to implement when compared to other similar algorithms. Moreover, QPSOs require very few parameters to function. This algorithm surpasses the performance of all existing e-learning portals in terms of speed and assures to achieve better levels of accuracy. Based on the understanding of all related concepts, explanations for the various aspects of the algorithm have been provided in the paper.

1. INTRODUCTION

The evolutionary PSO algorithm was introduced by Kennedy and Eberhart[1] in 1995. It is a global search strategy that can easily and efficiently handle arbitrary optimization problems. The idea of PSO algorithm is to imitate the interactions that take place between the individuals or members of a given biological swarm, say, swarms of bees, flocks of birds, schools of fishes, herds of animals, etc. to adapt to their environment, find the best food source and avoid predators by sharing information between them. Hence, this method has an evolutionary advantage.The concept is explained by taking the example of a flock of birds. Say, the birds(solution candidates) are allowed to fly in a specified area, looking for food. After a certain period of time (generations or iterations), all the birds will flock around the highest concentration of food in that particular area (global optimum). The current location of each bird is updated at every generation (or iteration) by using the information about local and global bests that have been provided by other birds. The process of arrival to the global optimum is ensured by the continuous interaction between the solution candidates. Compared to other evolutionary computing techniques, such as genetic algorithms [2], this technique has proved to be very efficient.

* Research scholar St.Peter's University, Avadi
** Head Incharge,Department of Computer Science , Pondicherry University CC.

A variation of PSO is its quantum based version, called QPSO [3]. All the particles in this algorithm are believed to have quantum behavior instead of classical Newtonian dynamics. So, the search procedure basically uses quantum motion. Compared to all the classical PSO techniques, when tested against a list of benchmarking functions, QPSO had a superior performance. In the testing cases, the population considered was very large. So, this algorithm can easily be used for the purpose of e-learning where the list of subjects or courses is very large [3]. The distinguishing feature about this particular algorithm is that it required very few parameters.

2. PSO, QPSO AND GENETIC ALGORITHM:

2.1 The Classical PSO:

These are the set of basic rules that govern the principles of PSO. Let us say, that our problem is N-dimensional.Genetic algorithm is discrete;hence, it can easily handle discrete variables. The design variables are encoded into bits of 0s and 1s. On the other hand, PSO is continuous. It has to be modified such that it can work on discrete variables. Considering a set of 'M' particles, the position vectors with respect to time are as follows:

$$x^m(t) = [x_1(t)\ x_2(t)\ \ldots x_n(t)]^T \tag{1}$$

Similarly, the velocity vector is as follows:

$$v^m(t) = [v_1(t)\ v_2(t)\ \ldots v_n(t)]^T \tag{2}$$

where, T is the transpose operator. '*m*' is an index ranging from 1 to M. As discussed previously, the idea of classical PSO is the information exchange about global and local best values. This can be achieved by the following equations:

$$v^m(t+\Delta t) = wv^m(t) + c_1\overbrace{[\Phi_1](\ P_L{}^m - x^m(t)) + c_2[\Phi_2]}^{\text{Particle memory influence}}\underbrace{(P_g - x^m(t))}_{\text{Swarminfluence}} \tag{3}$$

where Δt is the time step, the local best vector of the m^{th} particle is $P_L{}^m$ and the global best is P_g. There are two diagonal matrices.

$$[\Phi_i] = \text{diag}(\varphi_1{}^i, \varphi_2{}^i,, \ldots\ \varphi_N{}^i)\ , i = 1,2 \tag{5}$$

The independent variable present in these matrices are uniformly distributed between 0 and 1. '*w*' is the factor of inertia (generally, inertia factor range : 0.4 to 1.4); c_1 is the cognitive factor (generally, range : 1.5-2) and c_2 is the social factor(generally, range: 2-2.5). Therefore, the equation (3) can be simplified.

$$v^m(t + \Delta t) = wv^m(t) + [\Phi](P^m - x^m(t)) \tag{6}$$

where,

$$\begin{aligned} P_m\ =\ & c_1\ diag\left(\frac{\varphi_1^1}{c_1\varphi_1^1 + c_2\varphi_1^2}, \frac{\varphi_2^1}{c_1\varphi_2^1 + c_2\varphi_2^2}, \ldots \frac{\varphi_N^1}{c_1\varphi_N^1 + c_2\varphi_N^2}\right) P_L^M \\ & + c_2\ diag\left(\frac{\varphi_1^2}{c_1\varphi_1^1 + c_2\varphi_1^2}, \frac{\varphi_2^2}{c_1\varphi_2^1 + c_2\varphi_2^2}, \ldots \frac{\varphi_N^2}{c_1\varphi_N^1 + c_2\varphi_N^2}\right) P_g \end{aligned} \tag{7}$$

$$[\Phi] = diag\left(c_1\varphi_1^1 + c_2\varphi_1^2,\ c_1\varphi_2^1 + c_2\varphi_2^2, \ldots c_1\varphi_N^1 + c_2\varphi_N^2\right) \tag{8}$$

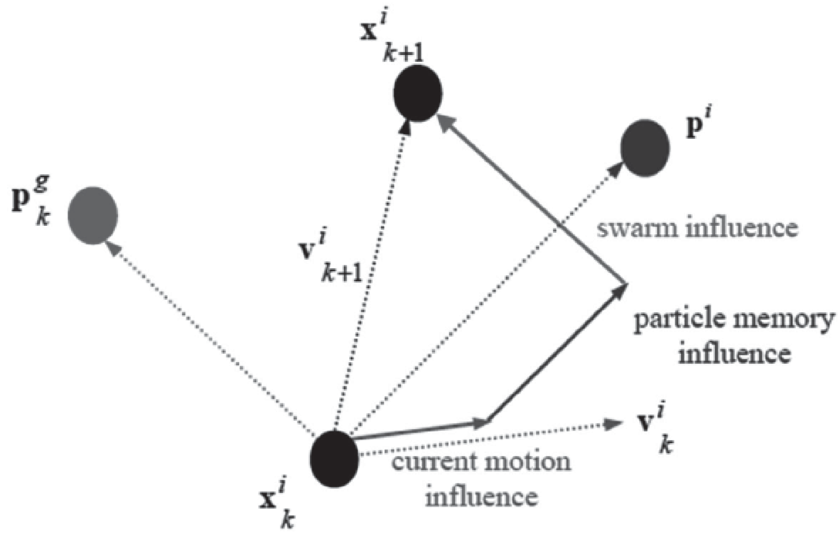

Figure 1. Depiction of the velocity and position updates in Particle Swarm Optimization. [9]

Nomenclature for Figure (1):

- p_k^g stands for 'which particle has the best global value in the given swarm'.
- p^i stands for 'best position of each particle over time in current and all previously undergone moves'
- v_k^i stands for 'search direction for this iteration'.
- v_{k+1}^i stands for 'search direction for the next iteration'.

For a PSO algorithm to obtain full coverage [4], all the particles have to reach towards the location P^m ,equ. (7). A maximum velocity, V_{max} is to be introduced in each of the '*n*' dimensions, so that the particles tent to stay inside the boundary of the domain.

2.2 Quantum based Particle Swarm Optimization:

As discussed earlier, all particles in QPSO are allowed to move following quantum-mechanical rules instead of Newtonian mechanics. Considering the classical environment, all the particles move towards the most optimum location, here, P^m, as shown in equation (7). The optimization process *attracts* the particles towards this position. This process of *attraction* leads to finding the global optimum. Now, coming to quantum mechanics, the basic equation governing it is the time-dependent Schrödinger Equation.

$$j\hbar \frac{\partial}{\partial t}\Psi(r,t) = \hat{h}(r)\Psi(r,t) \tag{9}$$

Here, $\hat{H}$is the time-independent Hamiltonian operator

$$\hat{H}(r) = \frac{\hbar}{2m}\Delta^2 + V(r) \tag{10}$$

Here, ħ is the Planck's constant, the mass of the particle is represented by *m* and V(*r*) is the distribution of potential energy. The wave function, Ψ(*r*,*t*), is unknown. This function does not have a direct physical meaning. The square of its amplitude is a measure of the motion of the particle. Once we impose this normalization, we can say that

$$\iiint |\Psi(r,t)|^2 \, dxdydz = 1.0 \tag{11}$$

In this case, the integration is performed over the entire space.We will apply a set of constraints such that what type of subjects the person likes, what are his fields of interest, what are the courses that he has taken up in the past, etc. These act as the attractive potential field[3] and these conditions will gradually*attract* the particles towards the position defined by equation (7). The provided conditions will generate *bound states* [7].

Let us assume we have a 1-D problem, where a particle is moving in the dimension *r*. Considering , p is the average best '*p*' given by equation (7), $r = x - p$. Now, *r* should approach zero for coverage of all particles. An attractive potential field centered at zero needs to be applied. Basically, any potential well can do the task; however, *delta-well* is the simplest of all. [7], [8]

$$V(r) = -\gamma\delta(r) \tag{12}$$

Where, γ is a positive number that is proportional to the depth of the potential well, which is infinite at origin and zero at any other point.

Using the principle of separation of variables, the time dependence of the wave function is separated from the spatial dependency [7]. On substituting the separated form in equation (8), we get

$$\Psi(r, t) = \Psi(r)^{e^{-jEt/\hbar}} \tag{13}$$

where E is the energy of the particle. $\Psi(r)$ can be found by solving the following time-independent SchrödingerEquation.

$$\left\{-\frac{\hbar^2}{2m}\Delta^2 + V(r)\right\}\Psi = E\Psi \tag{14}$$

Hence, we get time–independent solutions. These solutions are known as stationary states. Superposition of Eigen-solutions helps to form non-stationary states.

2.3 Genetic algorithm

Genetic algorithms can be implemented in many ways. The design variables in this case are known as chromosomes. GA can work with the combination of continuous and discrete parameters in a given problem. The design variables are assigned with values that are between lower and upper bounds.

GA and PSO basically start in the same way, i.e. a random set of populations of designs are set to evolve at every generation (or iteration). Now, GA has three basic operators that help in the propagation of its generations. The "Selection" operator follows the principle of "Survival of the fittest". The "Crossover" operator is analogous to mating in biological populations. This operator integrates the good surviving features into the upcoming generations. "Mutation" operator forbidsthealgorithm from getting trapped in the local minima. It helps in changing the characteristics of the population.

3. QPSO AND E-LEARNING:

A self-learning process is which one individual learns about almost any topic through the use of internet or other communication network [10]. E-learning focuses on the process of interaction between the learners and the onesproviding the educational contents, thus, distinguishing it from the process of Distance learning.

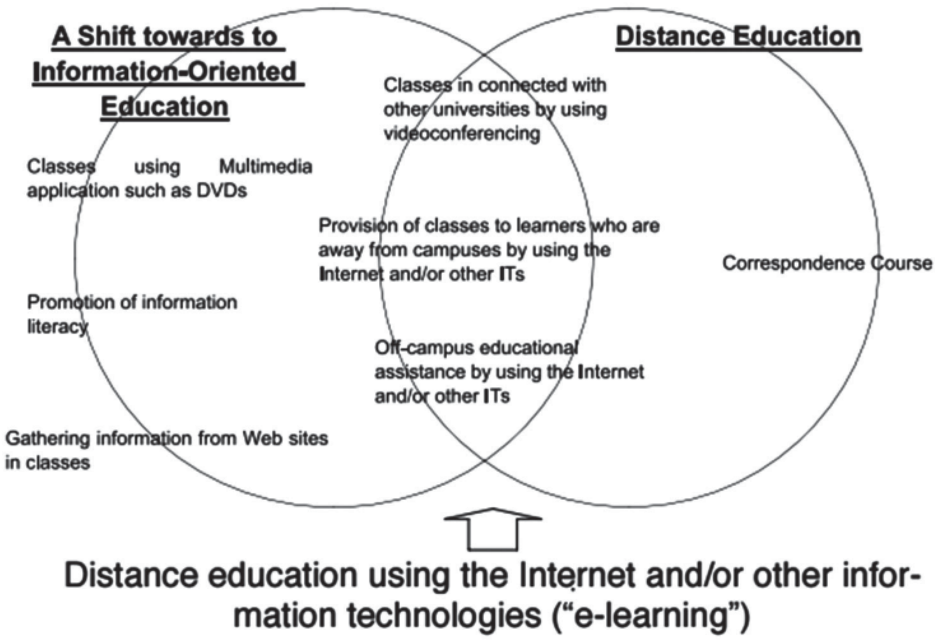

Figure 2. Depiction of e-learning v/s distance education [10]

E-learning is basically being used in all places. Previously, e-learning was only used by students if they wanted to know more about some particular topic. However, as e-learning gained momentum, it was no more limited to the school or university students. Today, e-learning is being used by people working in companies for the purpose of training of the employees. Now, the basic problem that every individual faces is that they get confused while selecting the course. Basically, in an e-learning portal there are a lot of courses that are provided. So, confusion is obvious. So, there should be some procedure that will help the candidate to recommend a proper course that is best suited for the candidate. Also, it should also be kept in mind that the particular candidate should also be notified about the trending courses that are present in the e-learning portal. This process can be done best by the use of a QPSO algorithm.

3.1 The QPSO algorithm [11]:

1. A choice of a suitable potential well needsto be done. They can be delta-well, harmonic oscillator, etc.
2. Then, the SchrödingerEquation needs to be solved.
3. Also, we need to find out the probability density function of the position of the particle. Here, each and every course is a separate particle.
4. The wave function needs to be collapsed into a desired region using suitable measurement methods. This desired region is computed using the data provided by the candidate. This data includes details about the specific candidate.
5. Apply a pseudo-code for this particular operation.

The cognitive parameter (c_1) and the social parameter (c_2) do not appear in the application of QPSO. In most cases, these factors are considered to be equal. Values of c_1 and c_2 have been used in many cases but they have yielded poor performance.A pseudo-code is shown for QPSO algorithm. The candidate solutions here are the courses.

3.2 Pseudo-code for QPSO:

```
Initialize x^m, P_local^m, P_global
Do i = 1, N_itr
    Do m = 1, N_population
            UpdateP_local^m andP_global
            φ1 = rand (0,1), φ2 = rand (0,1)
            P = (φ1P_local^m + φ2P_global)/(φ1+ φ2)
            u = rand (0,1)
            L = L(g, u, | x^m – p|)
            If rand(0,1) > 0.5
                    x = p + f(L,u)
            else
                    x = p + f(L,-u)
            end if
end do
end do
```

This is the pseudo-code that is required to implement a QPSO in an e-learning module. This algorithm will help us to attain the optimum solution easily without any hindrances.

4. COMPARISON WITH GA:

This implementation can also be done using the Genetic algorithm [5]. However, implementing the same with QPSO is much easier and time saving. In many cases, it has been seen that this algorithm is not able to operate successfully when it encounters a real world problem. In such cases, it leads to an acceptable answer but it is not computationally efficient. Objective functions can be divided into epistatic and non-epistatic when working with GA. However, if the function is non-epistatic (it can be solved by GA), the N-dimensional function can be divided into N-single-dimensional functions and then solved, which is complex. In such cases, QPSO is much easier to implement. If the function is epistatic, GA is not able to function properly and often results in errors.

Let us consider a few simulations [6]. We will use binary-encoding. The length of encoding of every variable is considered to be 15 bits. N=2 was set for all benchmark functions, i.e. 1, 2. During the simulation, each experiment was run 200 times.

The first function that is used is the Griewank function. Here, x_i is in the interval [-600, 600]. The global minimum value for this particular function is 0. The corresponding global optimum is $x_{opt} = (x_1, x_2, \ldots, x_n) = (100,100, \ldots ,100)$. From Fig. (3), it can be easily understood that all classic algorithms have fast convergence rate but all of them trap into local convergence. QPSO on the other hand has a more accurate convergence rate, thus, overcoming the disadvantage of local convergence.

The second function that is used is the Rastrigin function. Its value is 0 at global minimum solution. The global optimum is $x_{opt} = (x_1, x_2 \ldots x_n) = (0, 0, \ldots, 0)$. Here, x_i is in the interval [-5.12, 5.12]. The most complex part of finding optimal solutions to this function is that an optimization algorithm can easily be trapped in a local optimum on its way to the global optimum.From Fig. (4), it can be seen that QPSO has a very accurate convergence value than the other 3 algorithms.

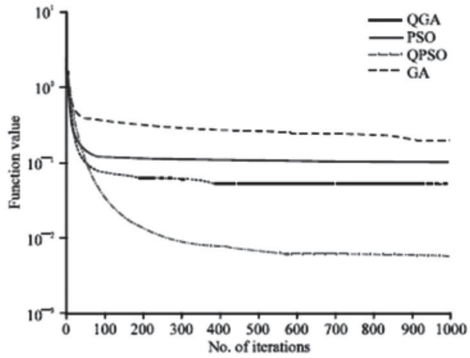

Figure 3. The performance of four algorithms using Griewank function [6]

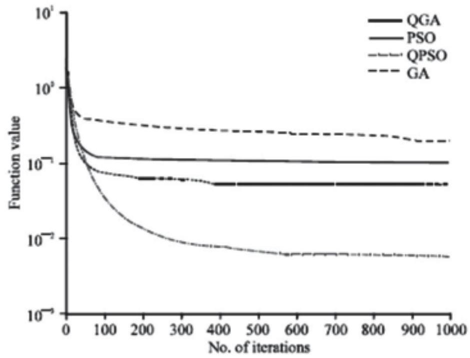

Figure 4. The performance of four algorithms using Rastrigin function [6]

5. CONCLUSION:

QPSO is one of the best algorithms for discrete optimization problems. This algorithm is used for the purpose of e-learning. Each of the particles represents a node. The candidate has to provide information about him. This information is used by the QPSO algorithm to understand what the basic interests of a particular candidate are. Based on that information, the student will be shown a set of courses that are meant for him and will help him in future. This process of optimization is done using QPSO. This process can also be done using several optimization algorithms. Some of the famous algorithms that can be used are classical PSO, genetic algorithms, etc. In this paper, it has been shown that out of all the algorithms QPSO is the one that is the easiest to implement. Not only that, QPSO has the maximum efficiency and also provides with the correct result every time, unlike GA, where we do not get the correct answer every time. Considering all the benchmark functions, QPSO seems to be the best optimization algorithm that can be applied to every e-learning portal. Moreover, it has an average time complexity. It is advantageous from each and every possible angle.

References

1. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE Conf. Neural Networks IV, Piscataway, NJ, 1995.
2. D. Boeringer and D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," IEEE Trans. Antennas Progat., vol. 52, no.3, pp. 771-779, Mar. 2004.
3. Jun Sun, Bin Feng, and WenboXu, "Particle swarm optimization with particles having quantum behavior," in Proc. Cong. Evolutionary Computation, CEC2004, vol. 1, pp. 325-331, June 2004.
4. M. Clerc and J. Kennedy, "The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space", *IEEE Trans. Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, Feb. 2002.
5. SajadJafari, Mansour R. Darabad, S. M. Reza and H. Golpayegani, "Is Genetic Algorithm As Efficient As Supposed?"
6. HongyuanGao, Jinlong Cao and Ming Diao,"A Simple Quantum-inspired Particle Swarm Optimization and its Application".*Information Technology Journal, 10: 2315-2321*.2011.
7. F. S. Levin, An introduction to quantum theory, Cambridge University Press, 2002.
8. P. A. Lindsay, Quantum mechanics for electrical engineers, McGraw-Hill, 1967.

9. Rania Hassan, BabakCohanim, Olivier de Weck, “A Comparison of Particle Swarm Optimization and the Genetic Algorithm”

10. Keiko Watanabe,”A Study on needs for e-learning – Through the analysis of national survey and case studies”, Progress in informatics, No. 2, pp. 77-86, (2005).

11. Said Mikki and Ahmed A. Kishk, “Quantum Particle Swarm Optimization for Electromagnetics”

12. Harnett, D., *Statistical Methods*, 3rd ed., Addison-Wesley, 1982, pp.247-297.

13. Hassan, R., and Crossley, W., “Variable Population-Based Sampling for Probabilistic Design Optimization with a Genetic Algorithm,” AIAA-2004-0452, *42nd Aerospace Sciences Meeting*, Reno, NV, January 2004.

14. I. De Falco, A. Della Cioppa, E. Tarantino, Mutation-based genetic algorithm: performance evaluation, Applied Soft Computing , 1, 285-299 (2002)

15. O. Hrstka, A. Kucerova, Improvements of real coded genetic algorithms based on differential operators preventing premature convergence, Advances in Engineering Software , 35, 237-246 (2004)

16. Y. D. Kwon, S. B. Kwon, S. B. Jin, J. Y. Kim, Convergence enhanced genetic algorithm with successive zooming method for solving continuous optimization problems, Computers and Structures, 81, 1715-1725 (2003)