

# A New Quantum Tunneling Particle Swarm Optimization Algorithm for Training Feedforward Neural Networks

**Geraldine Bessie Amali. D**

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India - 632014  
E-mail: geraldine.amali@vit.ac.in

**Dinakaran. M**

School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India - 632014  
E-mail: dinakaran.m@vit.ac.in

Received: 19 November 2017; Accepted: 24 April 2018; Published: 08 November 2018

**Abstract**—In this paper a new Quantum Tunneling Particle Swarm Optimization (QTPSO) algorithm is proposed and applied to the training of feedforward Artificial Neural Networks (ANNs). In the classical Particle Swarm Optimization (PSO) algorithm the value of the cost function at the location of the personal best solution found by each particle cannot increase. This can significantly reduce the explorative ability of the entire swarm. In this paper a new PSO algorithm in which the personal best solution of each particle is allowed to tunnel through hills in the cost function analogous to the Tunneling effect in Quantum Physics is proposed. In quantum tunneling a particle which has insufficient energy to cross a potential barrier can still cross the barrier with a small probability that exponentially decreases with the barrier length. The introduction of the quantum tunneling effect allows particles in the PSO algorithm to escape from local minima thereby increasing the explorative ability of the PSO algorithm and preventing premature convergence to local minima. The proposed algorithm significantly outperforms three state-of-the-art PSO variants on a majority of benchmark neural network training problems.

**Index Terms**—Particle Swarm Optimization algorithm, Quantum Tunneling, Artificial Neural Networks, Global Optimization, Nelder Mead, Feedforward Neural Networks.

## I. INTRODUCTION

Artificial Neural Networks (ANNs) are computational models inspired by the human brain and have been successfully applied to a wide range of pattern recognition, function approximation, classification and nonlinear regression problems [1]. A feedforward neural network is an inter connection of neurons arranged in layers. It consists of an input layer, an output layer and zero or more hidden layers. The input layers and the

subsequent layers are connected by links with weights. The strength of the link depends on the weight of the link. The performance of the neural network depends on the number of neurons in each layer and also the weights [1].

Training a neural network involves using an optimization algorithm to compute the weights thereby minimizing the mean squared error between the targets and actual network outputs [2]. Traditionally gradient descent based optimization schemes have been used to train neural networks. Due to the local and deterministic nature of the search performed by gradient based schemes the problem of convergence to a local minima near the initial random starting point is always an issue. Further the ANN error function is nonconvex and highly multimodal and finding the global minimum is a NP complete problem and exact solution is not practical [3]. Thus heuristic global optimization algorithms that compute a good local minimum for neural network training are of interest.

## II. RELATED WORK

In “Iris recognition using artificial neural networks” [4] the authors trained a feedforward neural network with Backpropagation (BP) algorithm for iris recognition. Backpropagation algorithm is a deterministic local search algorithm that converges to a local minimum close to the randomly chosen initial solution. The cost function is the average of the difference between the target and the neural network’s output over all the training samples. BP works in two phases. In the first phase, a forward pass calculates the network’s output for the randomly chosen weights. The error function then determines the error obtained by the network for the chosen weights and biases. The calculated error is then propagated backwards from the output layer to the hidden layers in the second phase. The weights are then adjusted to reduce the error [5, 6]. Since BP uses simple gradient descent to train ANNs it is very efficient but suffers from two

fundamental limitations. The learning time taken by the neural network increases with the increase in the size of the data. Since the weight update is proportional to the gradient in BP it does not perform well in the flat regions of the search space where the gradient is almost zero [7]. Biologically inspired optimization algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the other hand perform a random search and have the ability to escape from local minimum and locate a good approximation to the global minimum [8]. Thus training ANNs with stochastic global optimization algorithms which explore multiple local minima like GA and PSO can be advantageous because of the possibility of locating deep local minima compared to deterministic gradient based algorithms.

GA is inspired by natural evolution and employs selection, crossover and mutation operations to refine an initial random population [9]. It is stochastic and derivative free and can be applied to both continuous and discrete optimization problems [10]. Different variants of the GA have been used to calculate the weights of the feedforward neural network. GA with crossover was used to train ANN in "Improvement of Real-valued Genetic Algorithm and Performance Study" by [11]. A new distributed GA reinforced by perceptron learning rule was used in "Design architectures and training of neural networks with a distributed genetic algorithm" [12], whereas a variation of GA known as soft algorithm which combines GA with backpropagation was used to train ANNs by Adawy et.al., in "A SOFT-backpropagation algorithm for training neural networks"[6]. Application of a parallel implementation of the GA for training feedforward neural networks was explored in [13]. The limitations of GA include slow convergence to the optimum solution; use of complex constructs such as crossover, mutation and selection and the use of large number of adjustable parameters that determine the performance of the algorithm.

The PSO algorithm which is inspired by the social behaviour of a flock of birds was used for training neural networks by Gudise et al. in "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks" [14]. PSO algorithm is stochastic and derivative free algorithm like GA. However, it does not use complex constructs such as crossover, selection and mutation. It involves interactions between a population of agents with their environment as well as each other [15]. The interaction of agents are constrained using very simple rules. The interaction between the particles results in complex behaviour and emergence of a collective intelligence and problem solving ability.

The performance of the PSO algorithm was compared with the backpropagation algorithm on benchmark nonlinear function approximation tasks by Gudise et.al. It was found that the PSO was the faster of the two algorithms for function approximation tasks as it required smaller computational effort to attain the same error goal. PSO algorithm itself has certain limits in terms of convergence, precision and parameter selection [14]. The

algorithm converges prematurely to a local minimum thereby produced lower precision. Therefore in "A novel hybrid Evolutionary Algorithm based on PSO and AFSA for feedforward neural network training" [16] Chen proposed an algorithm called Artificial Fish Swarm Algorithm (AFSA) - PSO parallel hybrid evolutionary algorithm (APPHE) for training feedforward neural networks. The authors tested the performance of the proposed APPHE algorithm with the Levenberg-Marquardt Back propagation (LMBP) algorithm by training the neural network for iris data classification. The neural network trained using the APPHE algorithm did better than LMBP in terms of faster convergence to the good local minimum and accuracy of the result [17].

The explorative ability of the classical particle swarm optimization (PSO) algorithm is reduced because the value of the cost function at the location of the personal best solution found by each particle cannot increase. In this paper a new PSO algorithm inspired by the quantum tunneling in which the personal best solution of each particle is allowed to tunnel through hills in the cost function analogous to the quantum tunneling effect in Quantum Physics is proposed. The introduction of the quantum tunneling effect increases the explorative ability of the PSO algorithm and prevents premature convergence to local minima.

The paper is organized as follows: first the mathematical formulation of the problem is presented, secondly heuristic algorithms for global nonlinear optimization are discussed, thirdly a new PSO algorithm with enhanced exploration ability inspired by the tunneling phenomenon in Quantum Physics is proposed and finally the performance of the proposed algorithm is compared to three state-of-the-art PSO algorithms on benchmark neural network function approximation problems.

### III. PROBLEM FORMULATION

A feedforward neural network is a biologically inspired computation model that consists of a large number of processing nodes called neurons which are interconnected in layers [1, 2]. In a feedforward neural network, the information flows in one direction and does not support feedback. Therefore it is referred to as a feedforward neural network. Every neuron in a layer is connected with all the neurons in the previous layer. These interconnections have different strengths called weights. The weights help in encoding the knowledge of the neural network. The weights of these interconnections can be adjusted by a learning algorithm in order to match the desired output (target). A multi-layer perceptron with adequate number of neurons and one hidden layer can approximate any nonlinear function [1]. Fig. 1 shows the architecture of a feedforward neural network with one hidden layer, two inputs and one output neuron. In supervised learning the input pattern is given to the neurons in the input layer along with the expected output [2]. Each neuron computes the product of the interconnection weights and the input and adds it with the

bias and presents the output to a transfer function. The output generated by the neural network is compared with the target output.

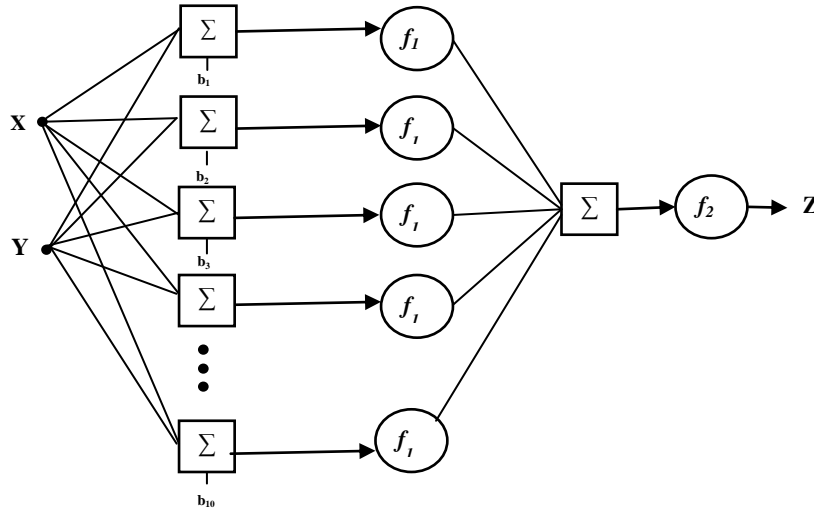


Fig.1. A feedforward ANN with one hidden layer, two inputs and one output neuron.  $f_1$  and  $f_2$  are tansig and purelin activation functions respectively.

In supervised learning, the learning is done with input and output pairs in the form of  $(X^i, Y^i)$ , Where  $i = 1$  to  $N$ , ‘N’ the number of training samples,  $X^i$  is the input vector  $X$  for the  $i^{th}$  training sample and  $Y^i$  is the target output for the  $i^{th}$  training sample. Let  $h(X^i)$  be the neural network output for an input  $X^i$  and  $\Theta$  is the activation function. The network’s output is presented in (1)

$$h(X^i) = \theta(\sum_{j=1}^k W_{ij} X_j + b_j) \tag{1}$$

where  $W_{ij}$  is the weight of the interconnection between  $i^{th}$  neuron and the  $j^{th}$  neuron;  $b_j$  is the bias of the  $j^{th}$  neuron. The mean square error can be calculated as follows

$$E(W) = \frac{\sum_{i=1}^N (Y^i - h(X^i))^2}{N} \tag{2}$$

Equation (2) defines the cost function that measures the difference between the ANN outputs and targets for a given data set. The weights of the ANN are determined by minimizing the cost function with respect to the weights. Because each neuron applies a nonlinear function to its weighted input to calculate its output the overall function  $h(x)$  calculated by the neural network and hence the error function  $E(W)$  are highly nonlinear. Thus the problem of training the weights of an ANN is a nonlinear global optimization problem of great importance. In this paper a new heuristic global optimization algorithm is proposed and applied to the problem of training feedforward neural networks.

#### IV. GLOBAL OPTIMIZATION

The goal in optimization is to minimize or maximize a real valued function by choosing the best solution from an available set of candidate solutions [18]. Solutions that satisfy the constraints are termed as feasible solutions. In the case of minimization of the ANN cost function the goal is to determine the value of the variables (weights) which minimizes the function [11, 19]. An optimization problem can be formulated as follows

$$\begin{aligned} &\min f(x) \\ &\text{Subject to } x \in \Omega \end{aligned}$$

Where  $f : R^n \rightarrow R$  is the function that is to be minimized and  $n$  is the dimensionality of the vector  $x$ . The set  $\Omega$  is a subset of  $R^n$ . The optimization problem above can be viewed as finding the vector  $x^*$  from the domain  $\Omega$  such that that  $f(x^*) \leq f(x)$ . A point  $x^* \in \Omega$  is a local minimum of  $f$  over  $\Omega$  if there exists  $\epsilon > 0$  such that  $f(x^*) \leq f(x)$  for all  $x \in \Omega \setminus \{x^*\}$  and  $\|x - x^*\| < \epsilon$ . On the other hand a point A point  $x^* \in \Omega$  is a global minimum of  $f$  over  $\Omega$  if  $f(x^*) \leq f(x)$  for all  $x \in \Omega$ . In general, it is only practical to compute a good local minimum as the problem of nonlinear global optimization is known to be NP-complete.

Optimization problems can be unimodal or multimodal [20]. A function is unimodal if it has one local minimum which is also the global minimum. Multimodal functions on the other hand may contain numerous local minima which makes the problem of locating the global minimum either very difficult or impossible. In many instances optimization algorithms get stuck in local minima without converging to the global minimum. Thus only

good local minimum can be computed in general. However a majority of important problems like training a neural network involves multimodal global optimization. Hence, heuristic random search algorithms inspired by biology like PSO and GA which can compute approximate optimal solutions are of interest [3].

## V. PARTICLE SWARM OPTIMIZATION ALGORITHM

Eberhart and Kennedy proposed the PSO algorithm inspired by the swarm behavior of a flock of birds and a school of fishes [15]. It has been shown to outperform other global optimization algorithms like the GAs [15, 16]. PSO algorithms starts with a randomly generated population referred to as a swarm which move around the search space [21, 22]. Each generated particle computes its best solution. The particle moves in the direction of the best solution found by it and also the best solution found by the entire swarm of particles. The personal best solution is referred to as  $pbest$  and the best solution determined by the swarm is termed as  $gbest$ .

In the PSO algorithm every generated particle moves towards the best position found by that particle so far and the best position found by the entire group swarm of particles [15]. The formula for the PSO algorithm is given in (3) and (4).

$$Vel^i(j+1) = Vel^i(j) + \Phi_{1d}(P^i(j) - X^i(j)) + \Phi_{2d}(G_d(j) - X^i(j)) \quad (3)$$

$$X^i(j+1) = X^i(j) + Vel^i(j+1) \quad (4)$$

where  $Vel^i = [Vel_1^i, Vel_2^i \dots Vel_N^i]$  denotes the velocity of the particle  $i$ ;  $X^i = [X_1^i, X_2^i \dots X_N^i]$  denotes the position of particle  $i$ ;  $\Phi_{1d}$  and  $\Phi_{2d}$  are random numbers which are uniformly distributed  $P^i = [P_1^i, P_2^i \dots P_N^i]$  is the best position determined by the particle  $i$ ,  $G = [G_1, G_2 \dots G_N]$  is the best position discovered by the whole group of particles. Dimension of the search space is denoted by  $N$  and  $j$  is the iteration number. The global best solution of the swarm is updated with the position  $X_i$  if the value of  $f(X_i) < f(G)$  [23, 24].

Numerous variants of the classical PSO algorithm have been proposed and applied to various problems. A variant of the PSO called CLPSO (Comprehensive Learning Particle Swarm Optimization Algorithm) was proposed by Liang et al [25]. In CLPSO a particle not only learns from its own experience but also on the best experiences of all the particles which helps the swarm from premature convergence to a local minimum [26]. Although the algorithm performed well in multi modal problems, it did not provide good solutions for unimodal problems. Another variant of the PSO algorithm is the Dynamic Neighbourhood Learning Particle Swarm Optimizer (DNLPSO) proposed by Nasi et al. In this variant each particle learns from its experience and also from the

experience of a dynamically changing neighbourhood. Learning from a dynamically changing neighbourhood helps in diversifying the learning experience of each particle and prevents premature convergence [27]. However, since the particles learn from the neighbourhood which is chosen at random, the particles might not be guided correctly. The DNLPSO algorithm is used for comparison in this paper.

### A. Improving accuracy with local search

Stochastic algorithms like PSO are computationally expensive because they perform a random walk in the search space to explore multiple local minima and hence tend to converge slowly. On the other hand deterministic local search algorithms are computationally efficient but perform a local search [8]. Thus approaches that combine the advantages of global and local search algorithms are of interest [3]. The Nelder Mead Simplex method is a derivative free local optimization technique used to minimize or maximize an objective function in an unconstrained multidimensional space [28]. In the case of a bi-variable function, the simplex forms a triangle and the technique performs a pattern search which computes function values at every vertex. The vertex which has the largest value in the case of a minimizing problem will be reinstated by a new vertex. A new triangle is formed and the search continues. The method results in a series of triangles, for which the values of the function at every vertex progressively reduces. The triangle size is then progressively decreased and the coordinates of the least points are determined. The NM-Simplex method is computationally efficient and robust [28, 29]. In this paper Nelder-Mead Simplex method is used to refine the solution found by the PSO for the ANN training problem.

## VI. A NEW QUANTUM TUNNELING PARTICLE SWARM OPTIMIZATION ALGORITHM (QTPSO)

The global exploration ability of the classical PSO algorithm is limited because the personal best solution found by each particle always moves down the surface of the cost function towards smaller values resulting in a greedy search heuristic. Since the personal best solution always moves downhill it will always converge towards a nearby local minima. In this paper the personal best solution of each particle is allowed to tunnel through peaks in the cost function surface analogous to tunneling of quantum mechanical particles.

In classical mechanics a particle cannot cross a potential barrier if its total energy is less than that of the barrier. However, in quantum mechanics a particle without sufficient energy can still cross the potential barrier with a small probability. Fig. 2 depicts the quantum tunneling effect in QTPSO. The particle is allowed to tunnel through the peaks in the cost function surface. Thus the personal best solution can tunnel through peaks in the cost function to reach even lower values and avoid getting trapped in local minima. If the cost  $C(x_p)$  at the newly computed position  $x_p$  is less

than  $C(x)$ , the cost at the current position  $x$ , the new position is accepted with a probability 1. However, if  $C(x_p)$  is greater than  $C(x)$  the new position is not discarded but accepted with a probability that depends on the difference between the costs and distance between the two positions. Analogous to quantum tunneling it is proposed that the tunneling probability decrease with respect to barrier height and length. The probability of accepting the new position  $x_p$  is given in (5).

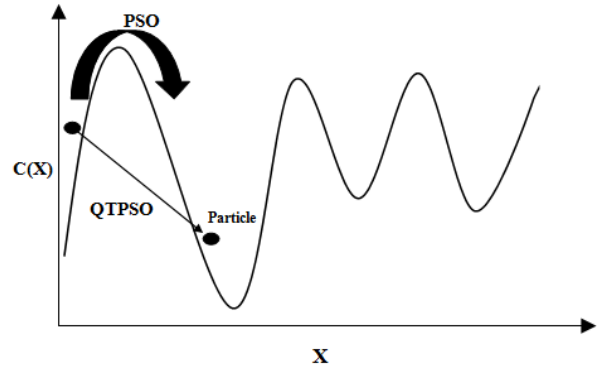


Fig.2 The quantum tunneling effect in QTPSO. The particle tunnels through the peaks in the cost function surface.

THE PROPOSED QUANTUM TUNNELING PARTICLE SWARM OPTIMIZATION ALGORITHM (QTPSO)

```

For i = 1 to N
  Initialize  $x^{(i)}$  in  $[x_{min}, x_{max}]^D$  randomly with uniform probability
  Initialize  $x^{(i)}, v$  in  $[v_{min}, v_{max}]^N$  randomly with uniform probability

Let  $x^{gbest} = \arg \min_i C(x^i)$ 
Let  $E := E_{max}$ 
For r = 1 to  $I_{max}$ 
  For p = 1 to N
     $x^{(p)}, v := \chi(x^{(p)}, v + c_1 r_1 (x_{pbest}^{(p)} - x^{(p)}) + c_2 r_2 (x^{(gbest)} - x^{(p)}))$ 
     $x^{(p)}, v := \max(x^{(p)}, v, v_{max})$ 
     $x^{(p)}, v := \min(x^{(p)}, v, v_{min})$ 
     $x^{(p)} := x^{(p)}, v + x^{(p)}$ 
  do
    If  $C(x^{(p)}) < C(x_{pbest}^{(p)})$ 
       $x_{pbest}^{(p)} := x^{(p)}$ 
    Else
       $P(x_{pbest}^{(p)} := x^{(p)}) := e^{\frac{-(C(x^{(p)}) - C(x_{pbest}^{(p)})) \|x^{(p)} - x_{pbest}^{(p)}\|}{E}}$ 
  do
    Decrease E
     $E := E_{max} \frac{(r - I_{max})}{(1 - I_{max})}$ 

  Let  $x^0 := x^{gbest}$ 

Perform Nelder Mead Simplex with  $x^0$  as the initial point and store the result in x
Return x
    
```

If  $C(x^{(p)}) < C(x_{pbest}^{(p)})$

$$P(x \leftarrow x_p) = 1$$

Else

$$P(x \leftarrow x_p) = e^{\frac{-((C(x_p) - C(x)) \|x_p - x\|)}{E}} \tag{5}$$

Where  $P$  is the probability that the  $x_p$  will be accepted

as the new position;  $x$  is the current position and  $x_p$  is the new position computed by PSO algorithm;  $C(x)$  is the cost function evaluated at  $x$ ;  $E$  is the Tunneling field.

To encourage exploration,  $E$  is kept large in the beginning, therefore the probability of transition to the new location is close to one. As the search progresses,  $E$  is reduced linearly to zero to avoid losing good solutions already computed. Finally the global best solution found by the PSO is used as the initial solution for the Nelder-Mead Simplex local search algorithm to refine the best solution found by the PSO. This is done to avoid wasting

computational effort doing a random search around the global minimum once it is approximately located. The nomenclature used in the proposed QTPSO algorithm is presented in Table 1.

Table 1. Nomenclature used in QTPSO

Symbol	Description
<b>N</b>	Size of the swarm
$\mathbf{x}^{(i)}$	Position of the $i^{\text{th}}$ particle in the swarm
<b>D</b>	Dimension of the search space
$\mathbf{vel}^{(i)}$	Velocity of the $i^{\text{th}}$ particle
$\mathbf{pbest}^{(i)}$	Personal best solution of the $i^{\text{th}}$ particle
<b>gbest</b>	Global best solution

VII. RESULTS AND DISCUSSION

The performance of the QTPSO algorithm was compared with three heuristic global optimization algorithms on benchmark feedforward neural network function approximation tasks. The architecture of the feedforward network consisted of two inputs, 10 hidden tansigmoidal hidden layer neurons and one linear output neuron. This feedforward network was trained to learn the 2D De Jong test suite of functions using different heuristic global optimization algorithms and the proposed

QTPSO algorithm. The De Jong test suite of benchmark functions that are used to compare the performance of the proposed QTPSO and the PSO variants are listed in Table 2. The De Jong test suite provides common challenges that the global optimization algorithms face, such as multiple local minima, flat regions and narrow ridge [5]. The performance of the QTPSO algorithm was compared with the classical PSO algorithm and hybrid PSO algorithms such as Nelder Mead Particle Swarm Optimization Algorithm (NMPSO), Dynamic Multi Swarm Particle Swarm Optimization Algorithm (DNLPSO). The mean square error obtained between the neural network’s output trained using the various algorithms and the ideal output was chosen as the performance metric in this paper.

Table 2. De Jong’s test suite of functions

Function	Definition
<b>Sphere</b>	$f_1(x, y) = x^2 + y^2$
<b>Rastrigin</b>	$f_2(x, y) = 20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)$
<b>Griewangk</b>	$f_3(x, y) = \frac{(x^2 + y^2)}{4000} - (\cos(x) * \cos(\frac{y}{\sqrt{2}})) + 1$
<b>Ellipsoid</b>	$f_4(x, y) = x^2 + 2 * y^2$
<b>Rosenbrock</b>	$f_5(x, y) = 100 * (y - x^2)^2 + (x - 1)^2$

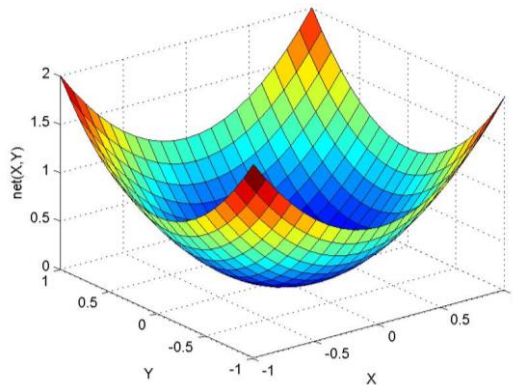


Fig.3. Plot of the ideal Sphere Test Function

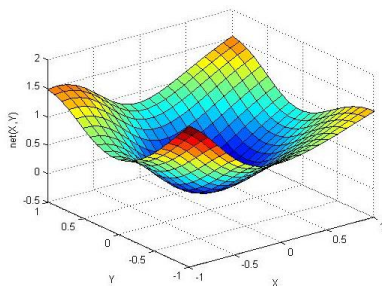


Fig.4. Approximation of sphere function learnt by an ANN trained with classical PSO algorithm.

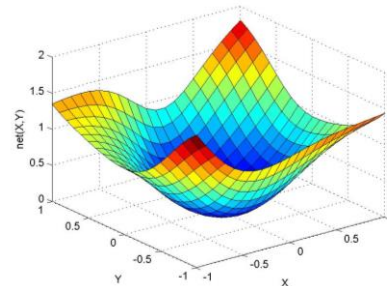


Fig.5. Approximation of sphere function learnt by an ANN trained with NMPSO algorithm.



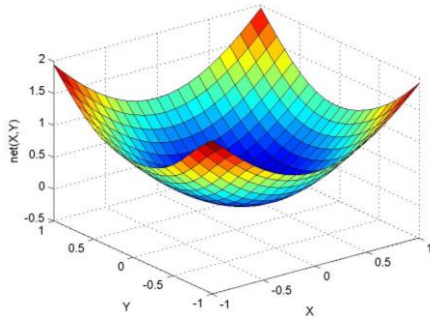


Fig.6. Approximation of sphere function learnt by an ANN trained with QTPSO algorithm.

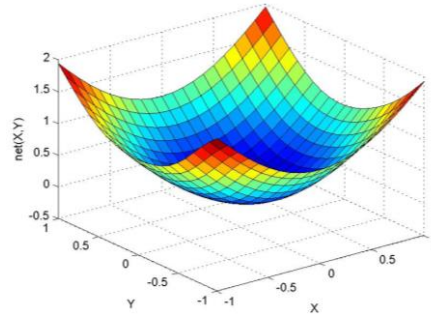


Fig.7. Approximation of sphere function learnt by an ANN trained with classical DNLPSO algorithm.

**B. BENCHMARK PROBLEM 2**

$$f_2(x, y) = 20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)$$

Benchmark problem 2 considers approximating the Rastrigin function. Rastrigin function is highly

multimodal and tests the ability of the global optimization algorithm to escape from local minima. A plot of the ideal Rastrigin function is in fig. 8. Fig. 9 indicates that the approximation of the Rastrigin function learnt by an ANN trained with the QTPSO algorithm is superior to the other algorithms.

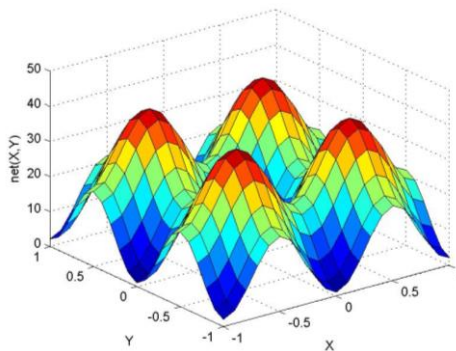


Fig.8. Plot of the ideal Rastrigin Test Function

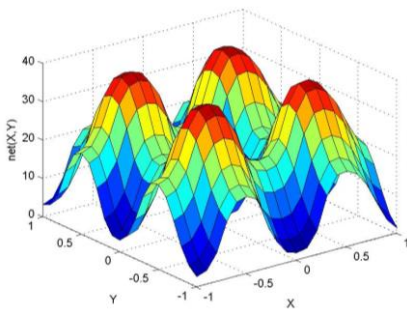


Fig.9. Approximation of rastrigin function learnt by an ANN trained with QTPSO algorithm

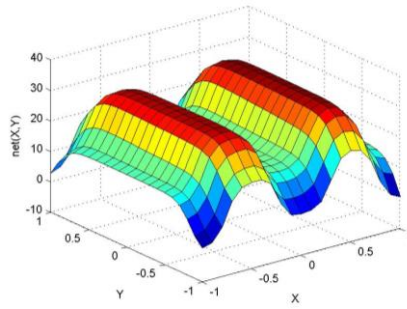


Fig.10. Approximation of rastrigin function learnt by an ANN trained with classical PSO algorithm

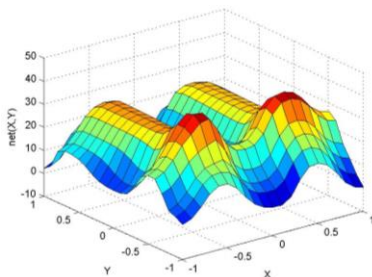


Fig.11. Approximation of rastrigin function learnt by an ANN trained with NMPSO algorithm

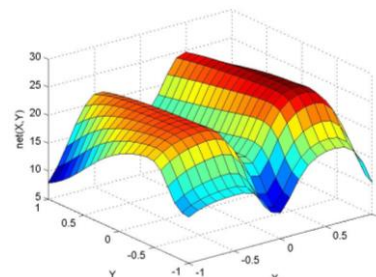


Fig.12. Approximation of rastrigin function learnt by an ANN trained with classical DNLPSO algorithm

C. BENCHMARK PROBLEM 3

$$f_3(x, y) = \frac{(x^2 + y^2)}{4000} - (\cos(x) * \cos(\frac{y}{\sqrt{2}})) + 1$$

Benchmark problem 3 considers the approximating the Griewangk function. The function is highly multimodal and tests the algorithm’s ability to explore the search

space and escape local minima.

A plot of the ideal Griewangk test function is shown in fig. 13. The approximation of the Griewangk test function learnt by the neural networks trained with different global optimization algorithms is in figs.14 to 17. Fig. 16 indicates that the approximation of the Griewangk function learnt by an ANN trained with the QTPSO algorithm is superior to the approximation learnt with PSO, NMPSO and DLNPSO algorithms.

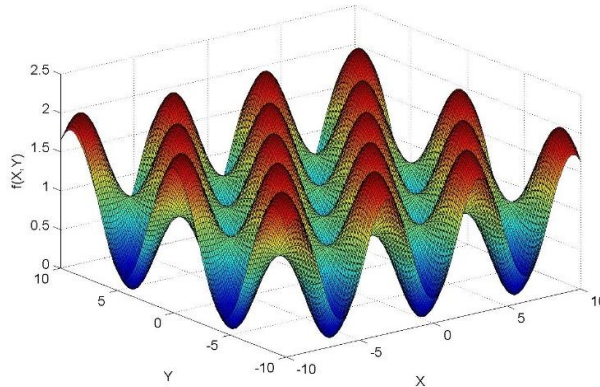


Fig.13. Plot of the ideal Griewangk Test Function

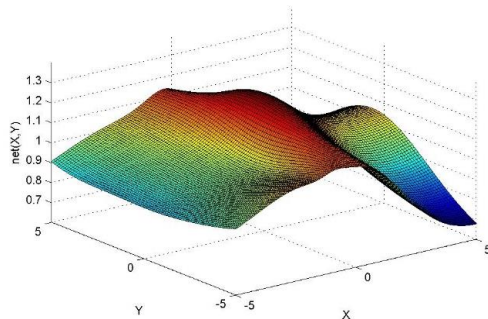


Fig.14. Approximation of Griewangk function learnt by an ANN trained with classical PSO algorithm

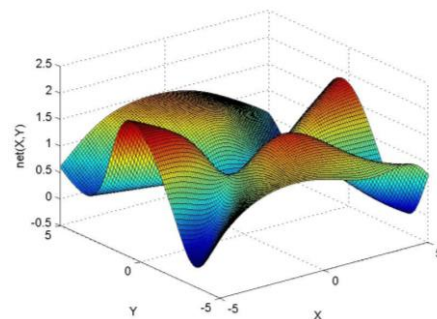


Fig.15. Approximation of Griewangk function learnt by an ANN trained with NMPSO algorithm

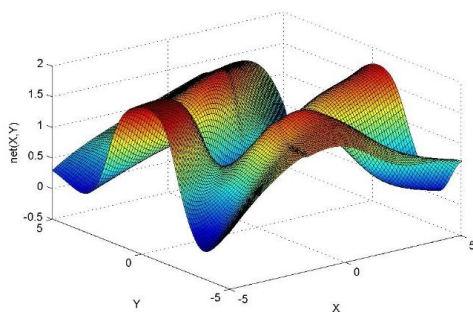


Fig.16. Approximation of Griewangk function learnt by an ANN trained with QTPSO algorithm

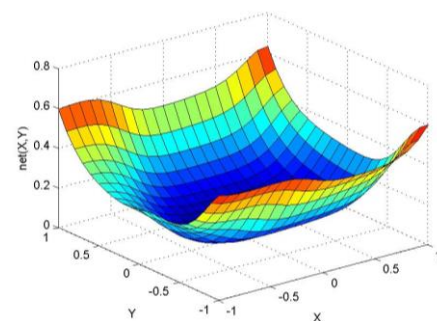


Fig.17. Approximation of Griewangk function learnt by an ANN trained with DNLPSO algorithm

D. BENCHMARK PROBLEM 4

$$f_4(x, y) = x^2 + 2 * y^2$$

Benchmark problem 4 considers approximating the Ellipsoid function. The function is continuous and unimodal. A plot of the ideal Ellipsoid function is shown

in fig. 18. The approximation of the Ellipsoid test function learnt by the neural networks trained with different global optimization algorithms is in figs. 19 to 22. Fig. 22 indicates that the approximation of the Ellipsoid function learnt by an ANN trained with the QTPSO algorithm is superior to the approximation learnt with PSO, NMPSO and DLNPSO algorithms.



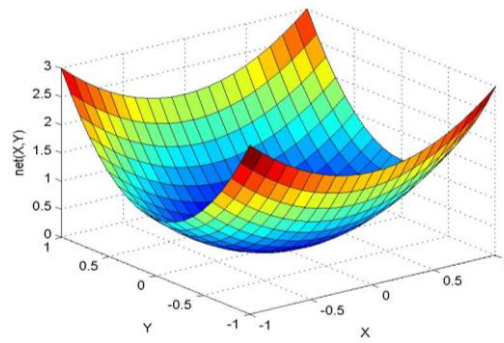


Fig.18. Plot of the ideal Ellipsoid Test Function

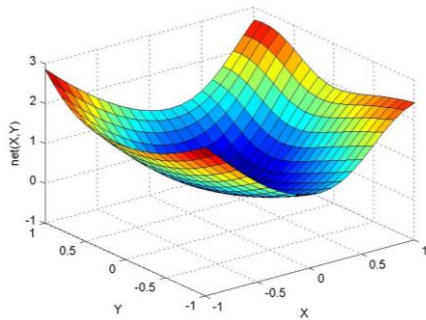


Fig.19. Approximation of Ellipsoid function learnt by an ANN trained with classical PSO algorithm

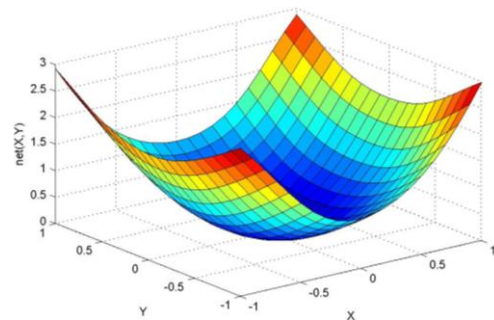


Fig.20. Approximation of Ellipsoid function learnt by an ANN trained with NMPSO algorithm

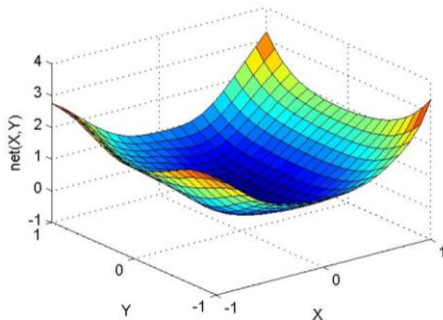


Fig.21. Approximation of Ellipsoid function learnt by an ANN trained with DNLPSO algorithm

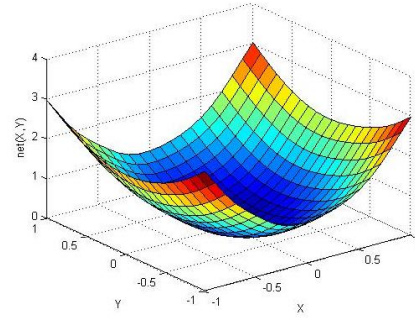


Fig.22. Approximation of Ellipsoid function learnt by an ANN trained with QTPSO algorithm

### E. BENCHMARK PROBLEM 5

$$f_5(x, y) = 100*(y - x^2)^2 + (x - 1)^2$$

Benchmark problem 5 considers the approximating the Rosenbrock's function. Rosenbrock function has a very narrow ridge. Algorithms that are not able to discover good directions underperform in this problem. An ideal

plot of the function is shown in Fig. 23. The approximation of the Rosenbrock's test function learnt by the neural networks trained with different global optimization algorithms is in figs. 24 to 27. Fig. 27 indicates that the approximation of the Rosenbrock's function learnt by an ANN trained with the QTPSO algorithm is superior to the approximation learnt with PSO, NMPSO and DNLPSO algorithms.

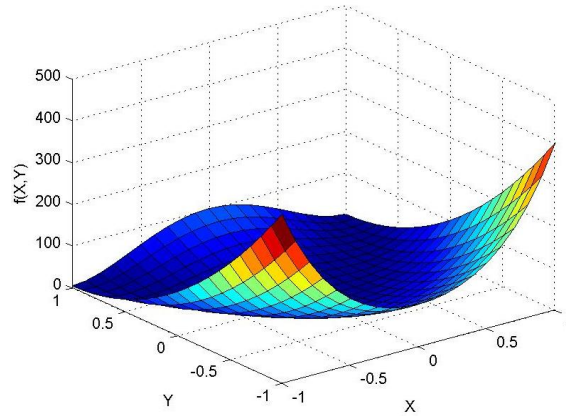


Fig.23. Plot of the ideal Rosenbrock's Test Function

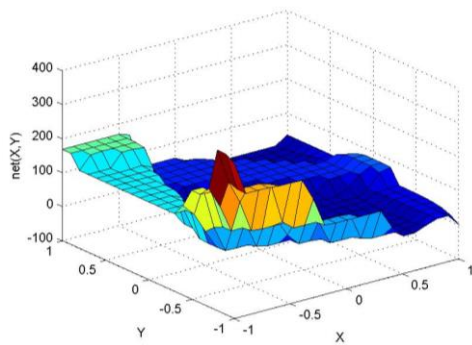


Fig.24. Approximation of Rosenbrock's function learnt by an ANN trained with classical PSO algorithm

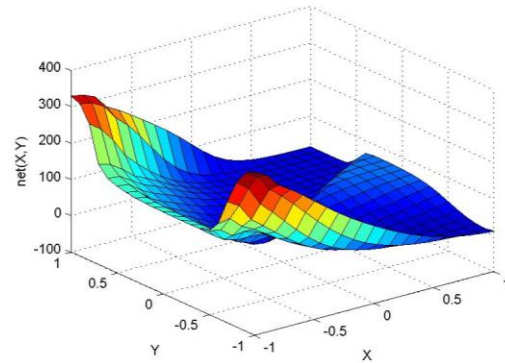


Fig.25. Approximation of Rosenbrock's function learnt by an ANN trained with NMPSO algorithm

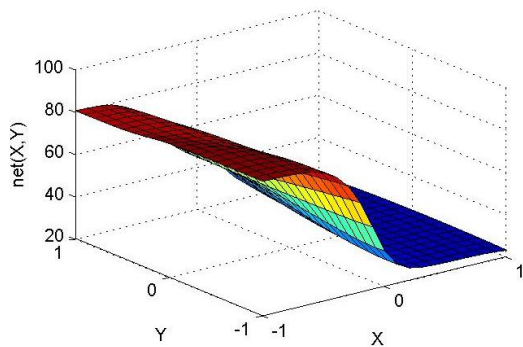


Fig.26. Approximation of Rosenbrock's function learnt by an ANN trained with DNLPSO algorithm

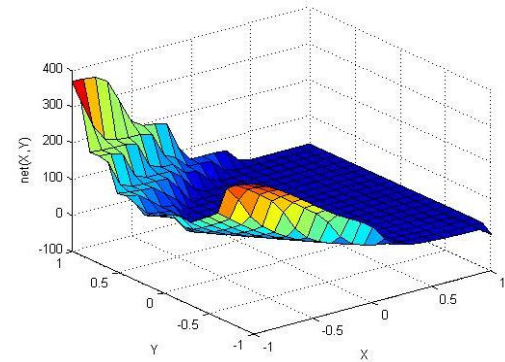


Fig.27. Approximation of Rosenbrock's function learnt by an ANN trained with QTPSO algorithm

Table 3 presents the mean square function approximation error for a feedforward ANN trained with different PSO variants. Table 3 clearly indicates that the QTPSO algorithm proposed in this paper significantly outperforms the classical PSO and state-of-the-art PSO algorithm variants on benchmark ANN training problems.

Table 3. Comparison of Mean Square Function Approximation Errors

Test Function	PSO	NMPSO	DNLPSO	QTPSO
SPHERE	0.006	0.001	0.0035	1.68E-04
Rastrigin	0.094	0.005	34.5732	9.89e-04
Griewangk	0.237	0.055	9.25e-04	2.29e-04
Ellipsoid	0.016	0.001	0.0090	1.70e-03
Rosenbrock	1.4e+03	967.6	4.11e+03	468.24

### VIII. CONCLUSION

In this paper a new quantum tunneling PSO algorithm was proposed and applied to the problem of training feedforward neural networks to learn the challenging De Jong's suite of benchmark functions. The problem of the personal best solution in the PSO algorithm getting trapped in local minima is overcome by allowing the personal best solution to tunnel through peaks in the cost function analogous to the tunneling phenomenon in Quantum Mechanics. In this paper the final solution is also refined with local search to avoid wasting computational effort performing stochastic search once the global minimum is approximately located. Thus

inclusion of the quantum tunneling effect allows many local minima to be explored and local search allows each local minimum to be accurately calculated. Simulation results on benchmark feedforward ANN function approximation problems indicate that the QTPSO algorithm proposed in this paper outperforms the three state-of-the-art heuristic global optimization algorithms. Future work can explore the effect of allowing the individual particles to also tunnel or take random jumps according to different probability distributions to further improve the explorative ability of the QTPSO algorithm. The use of low cost GPU hardware to reduce execution time of different PSO algorithms can also be explored.

## REFERENCES

- [1] A. Roy and M.M. Noel, "Design of a high speed line following robot that smoothly follows tight curves", *Computers and Electrical Engineering*, vol. 56, pp. 732-747, 2016.
- [2] F .M. Ham and I. Kostanic., *Principles of Neurocomputing for Science and Engineering*, vol.1, McGraw-Hill Companies, New York, N.Y, 2001.
- [3] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum", *Applied Soft Computing*, vol. 12 (1), pp. 353-359, 2012.
- [4] F. N. Sibai, H. I. Hosani, R.. M. Naqbi, S. Dhanhani and S. Shehhi., "Iris Recognition using artificial neural networks", *Expert Systems with Applications*, vol. 38 (5), pp. 5940–5946, 2011.
- [5] K. Smith and J. Gupta., "Continuous Function Optimisation via Gradient Descent on a Neural Network Approximation Function", In: *Proc. Of International Work Conference Artificial Neural Networks*, Berlin, Hiedelberg, pp. 741-748, 2001.
- [6] E. Adawy, "A SOFT-backpropagation algorithm for training neural networks", In: *Proc. Of the Nineteenth National Radio Science Conference*, Alexandria, Egypt, pp. 397-404, 2002.
- [7] M. Gori and A. Tesi, "On the problem of local minima in backpropagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14 (1), pp.76-86, 1992.
- [8] V.M. Nakarajan and M. M. Noel, "Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion", *Applied Soft Computing*, vol. 38, pp.771-787, 2016.
- [9] G. Amali and M. Dinakaran., "Solution of the nonlinear least squares problem using a new gradient based genetic algorithm", *ARPJ journal of engineering and applied sciences*, vol. 11 (21), pp. 12876-12881, 2016.
- [10] D. Whiteley, T. Starkweather and C. Bogart, "Genetic Algorithms and neural Networks: Optimizing Connections and Connectivity", *Parallel Computing*, vol. 14, pp. 347-361.
- [11] R. E. Zi-wu and S. A. Ye, "Improvement of Real-valued Genetic Algorithm and Performance Study", *Acta Electronica Sinica.*, vol. 2, pp. 0-17, 2007.
- [12] Olikier, M. Furst and O. Maimon "Design architectures and training of neural networks with a distributed genetic algorithm", In: *Proc. Of IEEE International Conference on Neural Networks*, San Francisco, USA, pp.199-202, 1993.
- [13] S. G. Mendivil, O. Castillo and P. Melin, "Optimization of artificial neural network architectures for time series prediction using parallel genetic algorithms", *Soft Computing for Hybrid Intelligent System*, vol. 154, pp. 387-399, 2008.
- [14] Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks", In: *Proc. of the 2003 IEEE in Swarm Intelligence Symposium*, Indianapolis, USA, pp.110-117, 2003.
- [15] R.C.Eberhart and Y.Shi, "Comparison between Genetic Algorithms and Particle Swarm Optimization", *Evolutionary Programming VII, Lecture Notes in Computer Science*, vol.1447, pp. 611-616, 1998.
- [16] X. Chen, J. Wang, D. Sun and J. Liang, "A novel hybrid Evolutionary Algorithm based on PSO and AFSA for feedforward neural network training", In: *Proc. Of 4<sup>th</sup> International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, pp. 1-5, 2008.
- [17] J. Chen, J. Zheng, P. Wu, L. Zhang, and Q. Wu., "Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems", *Expert Systems with Applications*, vol. 86, pp. 208-223, 2017.
- [18] M.M.Noel and T.C.Janette, "A new continuous optimization algorithm based on sociological model", In: *Proc. of American Control Conference*, Portland, USA, pp. 237-242, 2005.
- [19] A. Conde., A. Arriandiaga, J.A Sanchez, E. Portillo, S. Plaza and I. Cabanes, "High-accuracy wire electrical discharge machining using artificial neural networks and optimization techniques". *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 24-38, 2018.
- [20] G.P. Singh and A. Singh, "Comparative study of krill herd, firefly and cuckoo search algorithms for unimodal and multimodal optimization", *International journal of intelligent systems and applications*, vol. 6(3), pp. 35-49, 2014.
- [21] G. Amali and V. Vijayarajan, "Accurate solution of benchmark linear programming problems using hybrid particle swarm optimization (PSO) algorithms", *International journal of applied engineering research*, vol.10 (4), pp. 9101-9110, 2015.
- [22] Y. Zhang., D. W. Gong, X. Y. Sun and Y. N. Guo, Y., "A PSO-based multi-objective multi-label feature selection method in classification". *Scientific Reports*, vol. 7, pp. 1-12, 2017.
- [23] J.J. Liang and P.N. Suganthan PN, "Dynamic multi-swarm particle swarm optimizer", In : *Proc. of Swarm Intelligence Symposium*, Pasadena, USA, pp. 124-129, 2005.
- [24] D.G.B. Amali and M. Dinakaran, "A review of heuristic global optimization based artificial neural network training approaches", *International journal of pharmacy and technology*, vol. 8(4), pp. 21670-21679, 2016.
- [25] S. Jiang, K.S. Chin, K, L. Wang, G. Qu. and K. L. Tsui, "Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department", *Expert Systems with Applications*, vol. 82, pp. 216-230, 2017.
- [26] J.J. Liang, A.K. Qin, P.N. Suganthan and S. Baskar., "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions". *IEEE transactions on evolutionary computation*, vol. 10 (3), pp. 281-295, 2006.
- [27] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder and P. N, Suganthan ., "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization", *Information Sciences*, vol. 209, pp.16-36,

2012.

- [28] W. Chagra, H. Degachi and Ksouri, "Nonlinear model predictive control based on Nelder Mead optimization method", *Nonlinear Dynamics*, pp. 1-12, 2017.
- [29] E. Zahara and Y. Kao, "Hybrid Nelder Mead simplex search and particle swarm optimization for constrained engineering design problems", *Expert Systems with Applications*, vol. 36 (2), pp. 3880-3886, 2009.

### Authors' Profiles



**Geraldine Bessie Amali** received her M.Tech. degree in Computer Science and Engineering from VIT University in 2014 where she received gold medal for graduating at the top of her batch. She also has a university rank in Master of Computer Applications from Bharathidasan University. She is currently working as an Assistant

Professor at VIT University and has more than 7 years experience teaching computer science. Her research interests include machine learning and biologically inspired optimization algorithms.



**Dr. Dinakaran M** has completed his B.Tech (IT), M.Tech (IT-Networking) in Vellore Institute of Technology and Ph.D in Anna University, Chennai, Tamil Nadu, India. He worked in TATA Consultancy Services as Assistant System Manager from September 2006 to July 2009. Currently he

is working as Associate Professor and Head of the Department in School of Information Technology and Engineering, VIT University, Vellore. He has published 25 articles in various International Conferences and Journals. His research interests are mobile networks, image retrieval and machine learning.

**How to cite this paper:** Geraldine Bessie Amali, D, Dinakaran, M, "A New Quantum Tunneling Particle Swarm Optimization Algorithm for Training Feedforward Neural Networks", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.10, No.11, pp.64-75, 2018. DOI: 10.5815/ijisa.2018.11.07