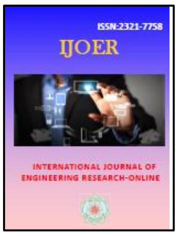
Impact Factor 5.8701 (ICI)
International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026
January
Special issue
ISSN: 2321-7758

Optimization of CNC Turning process using Classical and Quantum inspired Ant Colony Optimization algorithms: A Comparative Study

D. V. S. Lakshmi[1], D. V. S. S. S. V. Prasad[2], V. S. N. Ch. Dattu[3], G. Surya Rama[4]
[1]Lecturer in Physics, A.S.D. Government Degree College for Women (Autonomous), Kakinada, Andhra Pradesh, India; Email: dvslakshmiprasad@gmail.com
[2]Professor, Department of Mechanical Engineering, Aditya University, Surampalem–533437, Andhra Pradesh, India; Email: myselfdvsprasad@gmail.com
[3]Senior Assistant Professor, School of Nanotechnology, Jawaharlal Nehru Technological University Kakinada (JNTUK), Kakinada, Andhra Pradesh, India; Email: dathuthermal@gmail.com
[4]Assistant Professor, Department of Civil Engineering, University College of Engineering, JNTUK, Andhra Pradesh, India; Email: suryarama.g@gmail.com
Corresponding Author: D. V. S. Lakshmi (dvslakshmiprasad@gmail.com)

DOI: 10.33329/ijoer.14.S1.152



Abstract

In the era of digital manufacturing, automation is employed using sophisticated machines to enhance the product quality, minimizing the machining time and reducing production costs. CNC Turning process is a common process used in manufacturing industries, in which, speed, feed and depth of cut are the major input factors and surface roughness (Ra) and metal removal rate (MRR) are the major output characteristics. In this work, CNC Turning process is considered and polynomial regression (PR), support vector regression (SVR) and random forest (RF) techniques from classical machine learning (ML) are employed to model the process, based on the dataset prepared from the experimental observations. The experiments are conducted by considering the Aluminium 7075 alloy material. The best-fit ML technique is implemented to model the process. The validated model of the response, Ra is utilized in the process of optimization. Ant Colony Optimization Algorithm (ACO) and Quantum inspired Ant Colony Optimization Algorithm (QACO) are used to optimize the process parameters and the outcomes are compared. The study paves the way for quantum- accelerated intelligent machining systems.

Keywords: CNC Turning, Surface roughness, Machine learning, ACO, QACO.

1. Introduction

In the evolving landscape of digital manufacturing, the integration of automation and intelligent systems has become indispensable for achieving high-quality production, reduced machining time, and minimized operational costs. Among various machining processes, Computer Numerical Control (CNC) turning stands out due to its precision, repeatability, and suitability for a wide variety of materials and shapes, but its effectiveness depends heavily on choosing the right settings—specifically, the cutting speed, feed rate, and depth of cut. These factors play a major role in determining both the surface finish (Ra) and how quickly material is removed (MRR) — two critical performance indicators in the machining.

CNC turning is a machining process as shown in Fig. 1 that uses computer-controlled machines to create precision parts. A rotating work piece is shaped by a stationary cutting tool that moves along the X and Z axes. The movement of tool is controlled by computer control and allows to machine complex geometries with high accuracy. In general, turning process is used to produce cylindrical components like shafts, pins with close tolerance limits and fine finish and this process is used in great extent in the industries such as automotive, aerospace and medical devices.

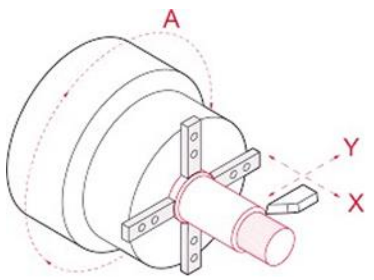


Fig. 1. CNC Turning process

It is a common practice to optimize the parameters of a machining process depends on empirical models or trial-and-error methods in conventional approach. These methods consume time and stuck at local minima. Data-driven modelling and optimization with the widespread application of machine learning techniques enables the prediction and optimization of parameters through the machine learning algorithms.

2. Related work

CNC turning is a commonly used process in the manufacturing industries and it impacts the productivity, quality and tool life significantly, if not optimized it. Ra and MRR are the two important output characteristics used to evaluate the performance of machining. Researchers explored empirical, statistical and machine learning based approaches to model and optimize the machining parameters.

Regression based approaches like polynomial regression [1], support vector regression [2], random forest [3] from machine learning were applied widely to predict the parameters of the machining processes. Conventional genetic algorithm and teaching-learning based optimization algorithms used in in different manufacturing processes [4]. Quantum machine learning with its algorithms is gaining popularity to explore the advantages of quantum computation. Techniques like Quantum Support Vector Machines (QSVMs) and quantum kernel estimation have demonstrated potential in classification and regression tasks [5], even though constrained by quantum hardware limitations. Grover's algorithm, a quantum search algorithm, is especially notable for providing a quadratic speedup in finding optimal solutions in unstructured datasets, making it a promising candidate for optimization tasks in manufacturing [6]. Few researchers attempted to optimize logistics and supply chain management with quantum optimization [7].

The idea of incorporating quantum computing principles into evolutionary algorithms was first proposed by Han et al. [8, 9]. In their approach, individuals are represented using quantum bits, or qubits. These qubits are then modified using quantum gates and other quantum operators. Interestingly, each quantum individual can represent multiple classical individuals at once which is the concept of superposition.

This allows for a broader search space and introduces diversity through quantum-based encoding. Quantum inspired algorithms are gaining superiority and popularity over classical evolutionary algorithms for solving complex engineering and combinatorial problems such as travelling salesman problem

International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026
January

[10, 11], knapsack problem [12], filter design problem [13], numerical optimization problem [14], network design problem [15], multi cast routing problem [16], flow shop scheduling problem [17], power system optimization [18], training of fuzzy neural networks [19]. Researchers started exploring hybrid approaches that combine classical ML models with quantum inspired algorithms for improved performance. This kind of approach in manufacturing processes remain unexplored. This work aimed to bridge the gap between classical and quantum inspired approaches for modelling and optimizing the parameters of CNC turning process.

3. Methodology

In this work, it is proposed to adopt both the classical and quantum inspired approaches as shown in Fig. 2. Initially, the experiments are conducted by considering the Aluminium 7075 alloy material on Sinumerik, CNC Turning center and dataset is prepared. PR, SVR and RF techniques from classical machine learning are implemented to model the CNC Turning process parameters. Further, GA is implemented to optimize the process parameters with the validated models and quantum inspired genetic algorithm is also implemented and the results are compared. The comparative analysis highlights the feasibility and potential advantages of quantum-assisted manufacturing and this study paves the way for quantum-accelerated intelligent machining systems.

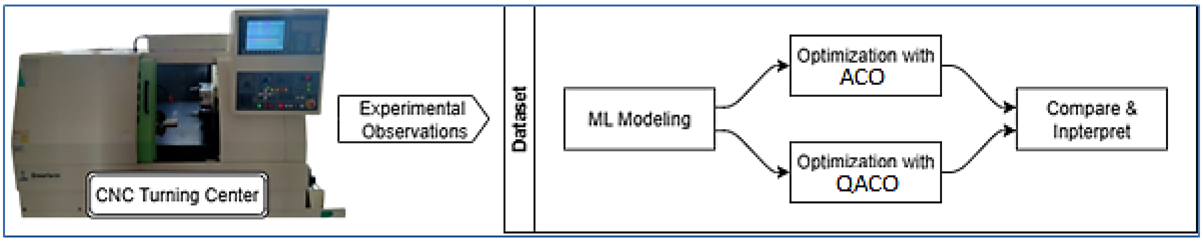


Fig. 2. Methodology adopted

3.1. Experimentation

A variety of techniques have been applied to different machining processes in previous research. In this study, CNC turning operations were carried out using a Sinumerik 8280 CNC Lathe Machine, as shown in Fig. 3.1. The material used was a round bar of Aluminium 7075 (75 x 33 mm), chosen for its widespread use in the tooling and manufacturing industries due to its excellent dimensional stability and strong resistance to wear and abrasion. The machined work piece is shown in Fig. 3.2. Data collected from these experiments were used to build machine learning models. The input factors considered were spindle speed (s), feed rate (f), and depth of cut (d), while the output variable was surface roughness (Ra).

The method used to measure Ra is illustrated in Fig. 3.3. To identify the best-performing ML model, PR, SVR, and RF techniques were applied and compared based on their statistical performance. Before training the models, the dataset was checked for normal distribution. A normal probability plot for Ra was plotted and is shown in Fig. 4. It can be observed from Fig. 4 that all the data points are normally distributed.

Figure 3.1 Sinumerik 8280 CNC Lathe machine

International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026 January

3.2. Work piece

3.3. Ra measurement

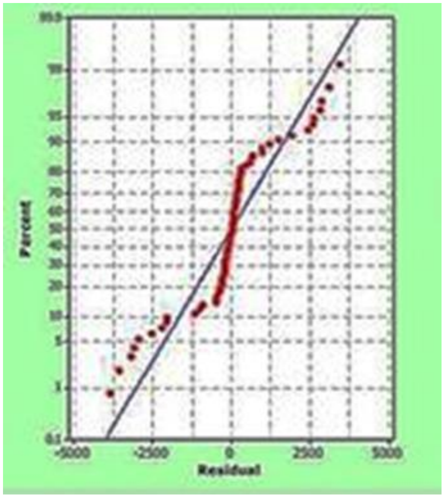


Fig. 3.4. Normal Probability plot for Ra

Machine learning models are used in the optimization process to fine-tune the machining parameters with the help of optimization algorithms. These methods are implemented using Python, by varying the split ratio of training-to-testing data. Standard split of 80:20 is used at first then lower ratio (75: 25) is considered but performance was not satisfactory. Higher split ratios are considered and their statistical performance is summarized in Table 1. As shown in the table 1, Polynomial Regression (PR) emerges as the most suitable technique with split ratio 95:05 for this dataset, achieving R-squared values above 95% for surface roughness (Ra) and showing minimal prediction errors.

Therefore, PR with a 95:05 training-to-testing data split was used to develop the machine learning models. PR was applied to the experimental data focusing on the output response, and a detailed analysis was performed. The regression coefficients for each individual variable and their interactions were calculated. As shown in Table 1, the R-squared value for surface roughness (Ra) is 0.9516, meaning the model explains about 95.16% of the variation in the data. The resulting ML model for Ra is presented in equation (1).

Table 1. Performance of ML techniques considered

Method	**Split ratio of the dataset is 80:20**		
	Ra		
	MSE	MAE	R2
PR	0.3506	0.4560	0.2958
SVR	0.3581	0.5595	0.2806
RF	0.3107	0.4648	0.3758
Method	**Split ratio of the dataset is 90:10**		
	MSE	MAE	R2
PR	0.1662	0.2684	0.5806
SVR	0.2171	0.4654	0.4523
RF	0.1366	0.3100	0.6553
Method	**Split ratio of the dataset is 95:05**		
	MSE	MAE	R2
PR	**0.0027**	**0.0519**	**0.9516**
SVR	0.2418	0.4895	-3.3059
RF	0.0366	0.1912	0.3483

$$Ra = 0.3799 + 0.00039s - 0.1951f - 1.4964d + 0.0000001185s^2 + 0.00053sf + 0.00005631sd - 0.318f^2 + 1.4715fd + 2.7384d^2 \quad (1)$$

International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026
January

3.2. Optimization of turning process parameters

Optimization is the process of finding the best values for a system's parameters to meet all design goals while keeping costs as low as possible. In this study, Equation (1) is used to carry out the optimization.

3.2.1. Ant Colony Optimization Algorithm

Ant Colony Optimization (ACO) algorithm is a nature-inspired, meta-heuristic that mimics the foraging behavior of real ant colonies to solve complex optimization problems. In ACO, a population of artificial ants collaboratively constructs solutions by moving through a problem space based on probabilistic decision rules.

These decisions are guided by two key factors: pheromone trails, which represent accumulated experience from previous ants, and heuristic information, which reflects problem-specific knowledge. As ants build solutions, their quality is evaluated using an objective function. Pheromone levels on solution components are then updated through evaporation and deposition mechanisms, where evaporation prevents premature convergence and deposition reinforces high-quality solutions. Over successive iterations, promising solution paths receive stronger pheromone reinforcement, increasing the likelihood of being selected by future ants. This iterative learning process enables ACO to balance exploration and exploitation effectively, leading to near-optimal solutions for combinatorial and continuous optimization problems in engineering applications. The pseudocode of ACO is given in exhibit 1.

```
Procedure Ant Colony Optimization:
  Initialize necessary parameters and pheromone trials;
  while not termination do:
     Generate ant population;
     Calculate fitness values associated with each ant;
     Find best solution through selection methods;
     Update pheromone trial;
  end while
end procedure
```

Exhibit 1. Pseudocode of ACO

3.2.2. Quantum inspired Ant Colony Optimization Algorithm

Quantum inspired Ant Colony Optimization (QACO) is an advanced meta-heuristic algorithm that combines the collective foraging behavior of classical ant colony optimization with concepts borrowed from quantum computing, such as superposition and probabilistic state representation. In this, solution components are encoded using qubits, which exist in a superposition of states, allowing the algorithm to represent multiple potential solutions simultaneously. Artificial ants construct solutions by observing or collapsing these qubits into discrete states, guided by both pheromone information and quantum probability amplitudes. The quality of each constructed solution is evaluated using objective function, and the best-performing solutions are identified. Unlike classical ACO, QACO updates its search process through quantum rotation gates, which adjust qubit probabilities toward promising regions of the solution space.

Simultaneously, pheromone trails are updated through evaporation and deposition mechanisms to reinforce high-quality solutions. This hybrid learning strategy enhances exploration, reduces premature convergence, and improves convergence speed, making QACO particularly effective for complex, high-dimensional engineering optimization problems and the pseudocode of ACO is given in exhibit 2.

International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026 January

```
Procedure Quantum inspired Ant Colony Optimization Algorithm
Initialize parameters: population size, pheromone evaporation rate, rotation angle
Initialize Pheromone trails, Qubit population in a state of superposition (usually 1/√2)
while not termination do:
        Generate ant population:
        Each ant constructs a solution by observing (collapsing) the Qubits into discrete binary
        states to choose its path;
        Calculate fitness values:
        Evaluate the objective function for each ant's constructed path;
        Find best solution:
        Compare current iteration ants with the global best;
        Select the "Best-so-far" ant to guide the quantum update;
        Update Quantum and Pheromone states:
        Apply Quantum Gates to the Qubit population to shift probabilities toward the best
        solution;
        Apply pheromone evaporation:
        Deposit new pheromone based on the best solution found;
    end while
return Best global solution;
end procedure
```

Exhibit 2. Pseudocode of Quantum inspired ACO

Ant Colony Optimization (ACO) and Quantum inspired ACO (QACO) serve complementary roles depending on problem complexity and resource availability. Classical ACO excels in moderate-sized problems where decision variables and constraints are manageable, providing an effective balance between exploration and exploitation with transparent pheromone-based mechanisms. Its simplicity, interpretability, and low computational requirements make it suitable for small-to-medium scheduling, routing, and process optimization, particularly in resource-limited environments like embedded systems or real-time applications. In contrast, QACO is designed for large, complex search spaces with high dimensionality and non-linear interactions. By leveraging quantum-inspired representations, it explores multiple regions simultaneously, maintains solution diversity, and avoids local minima through probabilistic superposition and interference mechanisms. Although more complex to implement, QACO delivers superior solution quality, faster convergence, and greater robustness, making it ideal for high-precision, research-driven, or large-scale optimization tasks where performance outweighs simplicity.

3.2.3. Formulation of optimization problem

The statement of optimization problem is given below:

Optimization problem : Single-objective optimization problem Objective : Minimize Ra

Input parameters : $500 \geq s \leq 1500$

$0.5 \geq f \leq 1.5$

$0.1 \geq d \leq 0.5$

4. Results and discussion

CNC Turning experimentation is conducted and the experimental observations are turned into a potential dataset. Machine learning techniques namely, PR, SVR and RF are implemented with different training-and-test data split ratios. PR is emerged as best-fit technique based on the statistical performance and the same is implemented to model the CNC turning process. The single objective optimization problem is formulated to optimize the CNC turning process parameters using ACO and QACO. Equation (1) is utilized as objective

International Journal of Engineering Research-Online
A Peer Reviewed International Journal
ISSN: 2321-7758 http://www.ijoer.in editorijoer@gmail.com
Vol.14., S1, 2026
January

function and both the algorithms are implemented using Python code and executed on Google Colab platform.

The ACO algorithm implemented using the parameters, ant population=50, maximum number of iterations=1500 and the convergence plot is presented in Fig. 7. The QACO is implemented using the parameters, ant population=80, maximum number of iterations=1500, rotation angle=0.05 and the convergence plot is presented in Fig. 8. It is observed from Fig. 7 and 8 that objective function is converged between 1000-1050 generations in ACO whereas the same is converged in QACO around 850th generation.

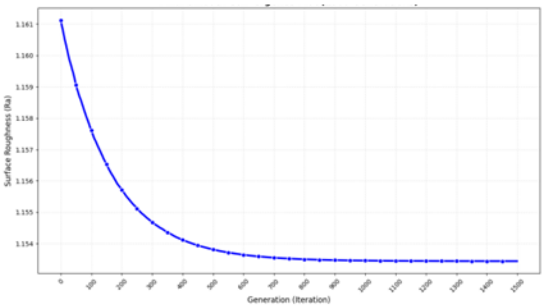


Fig. 7. Convergence plot for Ra with ACO

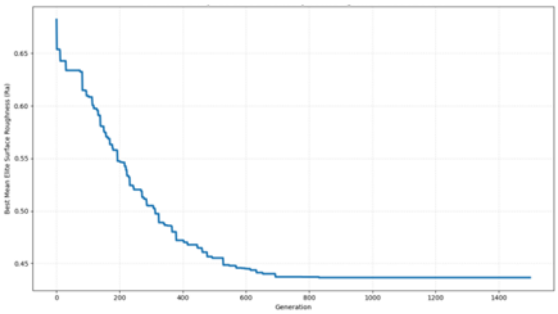


Fig. 8. Convergence plot for Ra with QACO

4.1. Comparison of ACO and Quantum inspired ACO

The comparative analysis is carried out while implementing conventional as well as quantum inspired approaches of optimization when optimizing the machining process parameters and the observations are presented below:

4.1.1. Convergence plots comparison and analysis

The convergence behavior of the two algorithms reveals a significant difference in both the final optimized surface roughness (Ra) and the efficiency of the search process. The ACO algorithm demonstrates a smooth, asymptotic descent, starting from an initial Ra of approximately 1.1610 and stabilizing at a final value of 1.1544. In contrast, QACO achieves a much lower (superior) surface roughness, starting its elite mean around 0.68 and converging to a final value of approximately 0.4360. The objective function value with ACO converges beyond 1000 iterations while with QACO converges between 650 and 750 generations—the QACO algorithm proves to be far more effective for this specific optimization task, as it finds a global minimum that is significantly lower than that achieved by the classical ACO.

When both the algorithms compared, it is observed that algorithms converge at different optimized values of surface roughness. QACO significantly outperforms ACO in terms of convergence speed, computational efficiency and optimization dynamics. This makes QACO particularly advantageous for turning applications, where minimizing machining time while achieving excellent surface finish is crucial. The ability of QACO to reach optimal solutions with less number of generations and minimum surface roughness underscores the potential of classical ACO as a more effective optimization technique for process optimization problems.

4.1.2. Algorithmic differences

The primary difference between the two approaches lies in their search mechanisms and parameter representation. Classical ACO implementation uses a continuous gradient-based approach where ants generate perturbations to find a local gradient, and the solution is updated using a momentum-based fractional method with a fixed decay rate. On the other hand, QACO algorithm utilizes quantum-inspired principles, representing pheromone levels through qubits and employing a quantum rotation gate for updates. This allow the QACO to maintain a balance between exploration and exploitation more effectively, utilizing polynomial decay for learning rates and noise to avoid local optima, which ultimately leads to the significantly better optimization results observed in the surface roughness values.

4.1.3. Outcome differences in CNC Turning context

Aspect	Classical Algorithm (ACO)	Quantum inspired Algorithm (QACO)
Final Surface Roughness (Ra)	Achieved slowly over many generations	Achieved rapidly with higher precision
Optimization Efficiency	Requires high computation and setup time	Low computation time; ideal for quick adjustments
Convergence Stability	Prone to slight oscillations and less stable	Smooth, monotonic, and highly stable convergence
Risk of Local Optima	Higher risk; prone to stagnation in search	Lower risk; Qubit rotation avoids premature convergence
Resource Efficiency	Moderate; higher energy/tool use due to long tuning	High; quick tuning improves tool and energy efficiency
Industrial Suitability	Primarily suited for offline optimization	Excellent for both offline and real-time/adaptive control
Repeatability	Moderate; sensitive to initial parameter tuning	High; consistently demonstrates stable optimization

5. Conclusion

In this work, CNC Turning process is considered for optimization and PR, SVR and RF techniques from classical ML are employed to model the process, based on the dataset prepared from the experimental observations. The best-fit ML technique, PR is implemented to model the process and regression analysis is carried out to determine the regression coefficients. The validated models of both the responses are utilized in the process of optimization. ACO and Quantum inspired ACO algorithms are used to optimize the process parameters and the outcomes are compared.

Traditional approach takes a bit long time to process than Quantum inspired approach which took very less time. Quantum inspired ACO outperforms classical ACO in terms of convergence speed; optimization quality; stability of solution and found quantum inspired ACO is more effective for minimizing surface roughness (Ra) while being computationally efficient. The study paves the way for quantum-accelerated intelligent machining systems.

References

[1]. Rao, S. S. (2009). *Engineering optimization: Theory and practice* (4th ed.). Wiley.

[2]. Suriyanarayanan, P., Suresh, R. K., & Narayanan, S. (2013). Support vector regression for modeling surface roughness in CNC turning. *International Journal of Advanced Manufacturing Technology, 67*(5–8), 1101–1112.

[3]. Livieris, V. N., Kalles, D., & Pintelas, A. (2016). Predicting surface roughness in CNC machining using ensemble learning. *Procedia CIRP, 46*, 463–466.

[4]. Rao, R. V. (2012). Teaching–learning-based optimization algorithm for solving unconstrained and constrained optimization problems. *Engineering Optimization, 44*(12), 1447–1462.

[5]. Schuld, M., & Petruccione, F. (2018). *Supervised learning with quantum computers*. Springer.

[6]. Grover, L. K. (1996). A fast quantum mechanical algorithm for database search.

In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)* (pp. 212–219).

[7]. Li, A. C. Y., Bharti, K., & Kyaw, T. H. (2021). Quantum optimization for logistics and supply chain management. *IEEE Transactions on Quantum Engineering, 2*, 1–14.

[8]. Han, K. H., & Kim, J. H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation, 6*(6), 580–593.

[9]. Han, K. H., & Kim, J. H. (2004). Quantum-inspired evolutionary algorithm with a new termination criterion, Hε gate, and two-phase scheme. *IEEE Transactions on Evolutionary Computation, 8*(2), 156–169.

[10]. Narayanan, A., & Moore, M. (1996). Quantum-inspired genetic algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (pp. 61–66).

[11]. Talbi, H., Draa, A., & Batouche, M. (2004). A new quantum-inspired genetic algorithm for solving the travelling salesman problem. In *Proceedings of the 14th International Conference on Computer Theory and Applications*, Alexandria, Egypt.

[12]. Han, K., & Kim, J. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In *Proceedings of the IEEE Congress on Evolutionary Computation* (Vol. 2, pp. 1354–1360).

[13]. Zhang, G. X., Li, N., Jin, W. D., & Hu, L. Z. (2006). Novel quantum genetic algorithm and its applications. *Frontiers of Electrical and Electronic Engineering in China, 1*(1), 31–36.

[14]. Abs da Cruz, A., Vellasco, M., & Pacheco, M. (2006). Quantum-inspired evolutionary algorithm for numerical optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* (pp. 2630–2637).

[15]. Xing, H., Ji, Y., Bai, L., Liu, X., Qu, Z., & Wang, X. (2009). An adaptive-evolution-based quantum-inspired evolutionary algorithm for QoS multicasting in IP/DWDM networks. *Computer Communications, 32*(6), 1086–1094.

[16]. Xing, H., Liu, X., Jin, X., Bai, L., & Ji, Y. (2009). A multi-granularity evolution-based quantum genetic algorithm for QoS multicast routing problem in WDM networks. *Computer Communications, 32*(2), 386–393.

[17]. Li, B. B., & Wang, L. (2007). A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37*(3), 576–591.

[18]. Vlachoglannis, J. G. (2008). Quantum-inspired evolutionary algorithm for real and reactive power dispatch. *IEEE Transactions on Power Systems, 23*(4), 1627–1636.

[19]. Zhao, S., Xu, G., Tao, T., & Liang, L. (2009). Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks. *Computers & Mathematics with Applications, 57*(11–12), 2009–2015.