Quantum Inspired Evolutionary Technique for Optimization of End Milling Process

Rajat Setia*, K. Hans Raj*, Suren N. Dwivedi**

(*Mechanical Engineering Department, Faculty of Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra, India)
(**Mechanical Engineering Department, University of Louisiana at Lafayette, U.S.A.)

ABSTRACT

In this paper an attempt is made to develop a new Quantum Inspired Evolutionary Technique (QIET) that is general, flexible and efficient in solving single objective constrained optimization problems. It generates initial parents using quantum seeds. It is here that QIET incorporates ideas from the principles of quantum computation and integrates them in the current frame work of Real Coded Evolutionary Algorithm (RCEA). It also incorporates Simulated Annealing (SA) in the selection process of Evolutionary Algorithm (EA) for child generation. In order to test this algorithm on domain specific manufacturing problems, Neuro-Fuzzy (NF) modeling of end milling process is attempted and the NF model is incorporated as a fitness evaluator inside the QIET to form a new variant of this technique, i.e. Quantum Inspired Neuro Fuzzy Evolutionary Technique (QINFET) and is effectively applied for process optimization of end milling process. The optimal process parameters obtained by QINFET correlates better than those reported in literature. The proposed methodology using QINFET is a step towards meeting the challenges posed in intelligent manufacturing systems and opens new avenues for parameter estimation and optimization.

Keywords – **End Milling, NF Modeling, QIET, QINFET.**

I. INTRODUCTION

The last two decades have witnessed tremendous growth in the application of stochastic search techniques. The primary reason for this is that these are well suited to the concurrent manipulation of models of varying resolution and structure. This is due to their ability to search non-linear space without gradient information or a prior knowledge relating to model characteristic. The most important stochastic search techniques that have been popular are Evolutionary Strategies (ES), Genetic Algorithms (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Immune Algorithm, Tabu Search (TS) and Quantum-inspired Evolutionary Algorithms (QIEA).

Many efforts have been made by researchers to overcome limitations of earlier algorithms such as slow and premature convergence by establishing a good balance between exploitation and exploration. One such effort resulted in the hybridization of Evolutionary Algorithms (EA) with other heuristics such as simulated annealing, local search, tabu search, hill climbing, dynamic programming, greedy random adaptive search procedure and quantum computing. This hybridization resulted in the improvement of performance in terms of convergence speed and quality of the solutions obtained by EA [1, 2].

Hans Raj et al. [3] have proposed a hybrid Evolutionary Computational Technique (ECT) by combining GA and SA. It is a hybrid scheme which incorporates a real–coded GA to provide multi-point search along with simulated annealing method to overcome local convergence and the problem of multiple minima. This technique provides more rapid and robust convergence on many function optimization problems. Two levels of competition are introduced between the strings in the population to ensure that only the better strings continue in the population. The concept of "Acceptance Number" is introduced to ensure that more computational effort is devoted to search in "better" regions of the search space. Constraints are handled by the use of the concept of penalty functions and by better coding.

This paper proposes a new variant of Real Coded Quantum Evolutionary Algorithm (RCQEA) using quantum computation principles to seed initial populations in the current framework of ECT as proposed by Hans Raj et al. [3], namely, QIET, which is more suitable than ECT for a wide range of real-world numerical optimization problems. To verify its effectiveness it has been applied to optimize end milling cutting conditions to obtain the best compromise between two critical machining-related values: surface roughness and machining time. Spindle speed, feed rate, radial depth of cut and tolerance were optimized, while one of the two key performance values was kept in the desired range and the other one was minimized.

The paper is organized as follows. The background material for quantum computation is described in section II. In section III QIET is described in detail. In section IV Neuro Fuzzy modeling of end milling process is detailed and in section V the new Quantum Inspired Neuro Fuzzy Evolutionary Technique is explained and applied to end milling process.

II. Introduction To Quantum Computation

A classical bit can only be in one of two states, 0 or 1 but according to the principles of quantum computation a qubit (or quantum bit) may be in the '1' state, in the '0' state, or in any superposition of the two. The state of a qubit can be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers that specify the probability amplitudes of the corresponding states. α^2 gives the probability that the qubit will be found in the '0' state and β^2 gives the probability that the qubit will be found in the '1' state. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1$$

The interesting part is that until the qubit is measured it is effective in both states. The probability of measuring the answer corresponding to an original 0 bit is α^2 and the probability of measuring the answer corresponding to an original 1 bit is β^2 [4, 5].

III. Quantum Inspired Evolutionary Technique (QIET)

In QIET the idea is to seed the initial population with a quantum approach in the framework of ECT. The algorithm concentrated its search only in the permissible regions of the search space using penalty approach. QIET search technique starts out with a guess of N grandparents, chosen at random in the search space. Initially each grandparent generates a number of quantum parents.

Number of quantum parents is chosen to be 10. Larger population sizes might yield further improvement in the results obtained but would entail higher computational effort. All the variables are encoded as floating point numbers. The method selects grandparents as random numbers in the range (0, 1) for each element of chromosome. Considerably distant points in the solution space are generated as grandparents to avoid any domination of particular schemata during the initialization process. Initially all grandparents generate an equal number of quantum parents. The advantage of using a quantum seeded generation is that a number of quantum parents are generated with each grandparent without using GA. The idea is taken from the point of view that in a parental string any value generated will either be bigger or smaller than another randomly generated number, which is known as probability of finding this string value in a particular state. If the string value is less than the random probability it is retained as such else it is changed as:

$$\sqrt{[1-(String\,Value)^2]}$$

This idea is in accordance with the principles of quantum computation as described in section II, which states that the probability of a qubit to be in any state satisfies the condition $|\alpha|^2 + |\beta|^2 = 1$. Thus in a single pass numerous quantum parents can be generated with a single grandparent. Each quantum parent is checked over its functional value and constraints violation. From these quantum parents further parents are selected. A quantum parent is made a parent only when it clears a criterion of the sum of penalties for all the constraints violated. Thus the total number of parents selected varies for each iteration and also varies with a particular run of the program. Now these parents are sent into ECT which is combined GA/SA and is used for further child generation.

Initially all parents generate an equal number of children given by $m(i) = M$. A reasonable value of M is taken as 10. A higher value of M results in a more exhaustive search with a corresponding increase in computational effort. The total number of children in a generation is fixed and is given by:

$$TC = \sum_{i=1}^{N} m(i)$$

For each parent i, mates are selected from the other parents at random and cross-over is applied to generate $m(i)$children.

For each family a blend cross-over operator (BLX-α) based on the theory of interval schemata is employed in the study. BLX-α operates by randomly picking a point in the range $(p_1 - \alpha\,(p_2 - p_1),\ p_2 + \alpha\,(p_2 - p_1))$ where p_1 and p_2 are two parent points and $p_1<p_2$. In a number of test problems BLX-0.5 performed better than the BLX operators with any other α values and has, therefore, been used. Mutation is not employed.

The best child (with minimum objective value) out of the children generated from the same parent is found. The best child then competes with its parents to survive in the next generation. If the best child is better than its parent, it is accepted as a parent in the next generation. If the best child is worse than its parent then Boltzmann criterion is applied before the child be accepted.

As in SA, the selection of temperatures is such that initially the probability of acceptance of a bad move, i.e. when the best child is worse than the parent is high (approximately 1) but as the temperatures are successively lowered through a cooling schedule this probability is decreased until, at the end, the probability of accepting a bad move is negligible (approximately 0). Logarithmic cooling schedule is adopted in this work. Such a strategy enables the technique to seek the global optimum without getting stuck in any local optimum. The initial and final temperatures are calculated as follows:

A bad move is accepted according to the Boltzmann Criterion. Initially the probability of accepting a bad move is approximately one i.e.

$exp\ (-\Delta X_{average}/T_1)=0.99$

and finally $exp\ (-\Delta X_{average}/T_{MAXIT})=0.0001$

Therefore, $T_1=-\Delta X_{average}/log\ (0.99)$

$T_{MAXIT} = -\Delta X_{average}/log\ (0.0001)$

where T_1 is the initial temperature, T_{MAXIT} is the final temperature, $\Delta X_{average}$ is the average difference between the objectives X for any two neighborhood points in the search space. This average is calculated over a number of chromosomes.

The number of children that are generated in the next generation is proportional to a parameter called the acceptance number. This number provides a measure of the goodness of solutions in the vicinity of the current parent. The number is computed by sampling the search space around the current parent and counting the number of good samples out of the total samples as per steps 13 to 15 of the pseudo-code. This strategy enables the algorithm to focus search on the better regions of the search space.

For highly constrained problems, infeasible solutions may occupy a relatively big portion of the population. The penalty technique is perhaps the most common technique used to handle infeasible solutions in the constrained optimization problems. In essence, this technique transforms the constrained problem into an unconstrained problem by penalizing infeasible solutions, in which a penalty term is added to the objective function for any violation of the constraints. The major concern is how to determine the penalty term so as to strike a balance between the information preservation (keeping some infeasible solutions) and the selective pressure (rejecting some infeasible solutions), and avoid both under penalty and over penalty. There are no general guidelines on designing penalty function. Constructing an efficient penalty function is quite problem dependent. The same has been incorporated in the QIET algorithm in evaluating the objective function i.e.

$$Evol(X) = f(X) + \beta \times D(X)$$

where, β is the problem dependent constant, and $D(X)$ is the difference measure for constraint violation. The result of such an approach is that the infeasible strings have much worse objective function values and are eliminated from the population whereas the strings with better objective function values survive and contribute more to the evolution of better solutions.

The number of grandparents taken depends upon what is the criterion for the selection of parents from the quantum parents. The relaxation gives us choice to start with small number of grandparents. It is seen that more constrained the criterion for the selection of parents is, the better is the convergence. The initial generation of the quantum parent ensures that parents with a better fitness value are sent into GA to produce further children. This gives GA a better convergence towards the optimum solution. The results show that the convergence is very fast. The various features explained above have been combined together to develop an optimization algorithm and is represented succinctly in the form of pseudo-code given below:

1. Generate random initial grandparent strings.
2. Generate a random probability.
3. If any of the string values is less than the random probability it is retained as such.
4. Otherwise it is changed as $\sqrt{[1-(String\,Value)^2]}$.
5. Initialize T_1 & T_{MAXIT} with N parent string.
6. For each parent i , generate $m(i)$ children using crossover
7. Find the best child for each parent (1st level of competition).
8. Select the best child as the parent for the next generation. For each family accept the best child as the parent for the next generation if

   $$Y_1 < Y_2 \ \ OR \ \exp[(Y_2 - Y_1) \ / \ T] \ \geq \rho$$

   where
   Y_1 is the objective value of the best child
   Y_2 is the objective value of its parent
   T is the temperature co-efficient
   ρ is a random number uniformly distributed between 0 and 1.
9. Repeat step 10 to Step 13 for each family
10. Count = 0
11. Repeat step 11for each child: Go to step13
12. Increase count by 1, if

    $((Y_1<Y_2)\ \exp\ ((Y_{LOWEST}-Y_1)/T) \geq \rho)$

    where Y_1 is the objective value of the child
    $Y2$ is the objective value of its parent
    Y_{LOWEST} is the lowest objective value ever found
    T is the current temperature
    ρ is a random number uniformly distributed between 0 and 1.
13. Acceptance number of the family is equal to count (A)
14. Sum up the acceptance number of all the families (S)
15. For each family i, calculate the number of children to be generated in the next generation according to the following formula $m\ (i) = (TC \times A) / S$
    where, TC is the total number of children generated by all the families.
16. Decrease the temperature.
17. Repeat Step 6 to Step 16 until a certain number of iterations has been reached.

IV. Neuro Fuzzy Modeling of End Milling Process

Intelligent manufacturing systems require intelligent models that can help the manufacturer to meet the customer demands with existing infrastructure. The rising demand for precision and quality in manufacturing necessitates that vast amounts of manufacturing knowledge be incorporated in manufacturing systems. Neuro Fuzzy modeling of end milling process is attempted in this section. Surface finish in end milling depends upon a number of variables such as cutting speed, feed rate, spindle speed, radial depth of cut, tolerance etc. The relative effect of these variables on surface roughness and machining time is quite considerable. A complex relationship exists between these process parameters and hence there is a need to develop intelligent models which can capture this complex interrelationship and enable fast computation of the average surface roughness and machining time based on process parameters.

Tansel et al. [6] have experimentally measured the surface roughness and the machining time at various test conditions. Aluminum block having 30x30x90mm dimensions was machined at three stages. The first two stages, rough and semi finish cut were the same for the entire part. A flat end mill with a 12mm diameter was used for rough cutting. The depth of cut was 1.5mm. 3D spiral tool motions were performed with 3mm stopovers at 2500mm/min feed rate and 5000rpm spindle speed. The rough cutting continued until 0.6 mm thick material was left on the desired final surface. A ball end mill with a 12mm diameter was used at the second stage to machine the material with a 0.3mm depth of cut. The step over, feed rate and spindle speed were 3mm, 700mm/min, and 3000rpm, respectively. After the second stage, 0.3 mm thick material was left on the desired part surface. The finishing cut (Third stage) was performed with a ball end mill with 10mm diameter.

Finishing cut continued until the desired surface was obtained. The surface roughness of the machined surface was measured by using a Mitatoyu Surftest 301 portable surface roughness tester. The surface roughness was measured three times at 10 different regions for each cutting condition and average was calculated.

The ranges of the cutting parameters are presented in table 1 and the sample experimental values by Tansel et al. and estimated values by neuro fuzzy model are shown in table 2.

Cutting Speed (m/min)	Feed Rate (mm/tooth)	Radial Depth of cut (mm)	Tolerance (mm)
74-123	0.07-0.12	0.1-0.3	0.01-0.001

Table 1 Range of cutting parameters

No	Cutting Speed (m/min)	Feed rate (mm/tooth)	Radial depth of cut (mm)	Tolerance (mm)	Experimental values by Tansel et al. [6]		Estimated values using Neuro Fuzzy Model	
					Average surface roughness (μm)	Machining Time (min)	Average surface roughness (μm)	Machining Time (min)
1	74	0.07	0.1	0.001	0.26	64	0.26	64.00
2	98.5	0.07	0.1	0.001	0.31	47	0.31	47.00
3	123	0.07	0.1	0.001	0.27	45	0.27	45.00
4	74	0.095	0.1	0.001	0.32	48	0.32	48.00
5	98.5	0.095	0.1	0.001	0.36	36	0.35	35.99
6	123	0.095	0.1	0.001	0.85	29	0.85	29.00
7	74	0.12	0.1	0.001	0.48	39	0.48	39.00
8	98.5	0.12	0.1	0.001	0.37	28	0.37	27.99
9	123	0.12	0.1	0.001	1.58	24	1.58	23.99
10	74	0.07	0.2	0.001	0.36	32	0.35	31.99
11	98.5	0.07	0.2	0.001	0.59	24	0.59	23.99
12	123	0.07	0.2	0.001	0.52	19	0.52	18.99
13	74	0.095	0.2	0.001	0.51	23	0.51	23.00
14	98.5	0.095	0.2	0.001	0.53	17	0.53	17.00
15	123	0.095	0.2	0.001	0.81	15	0.81	15.00

Table 2 Sample experimental and estimated values for given average surface roughness and machining time

Neuro-fuzzy inference system under consideration has four inputs as shown in figures 1 and 2 viz. cutting speed, feed rate, radial depth of cut, tolerance and one output machining time and average surface roughness each. The overall output is expressed as linear combinations of the consequent parameters. The output f can be written as:

$$f = \sum_{i=1}^{81} \overline{w}_i f_i$$

$$f = \sum_{i=1}^{81} (\overline{w}_i \alpha) p_i + (\overline{w}_i \beta) q_i + (\overline{w}_i \nu) r_i + (\overline{w}_i \gamma) s_i + (\overline{w}_i) t_i$$

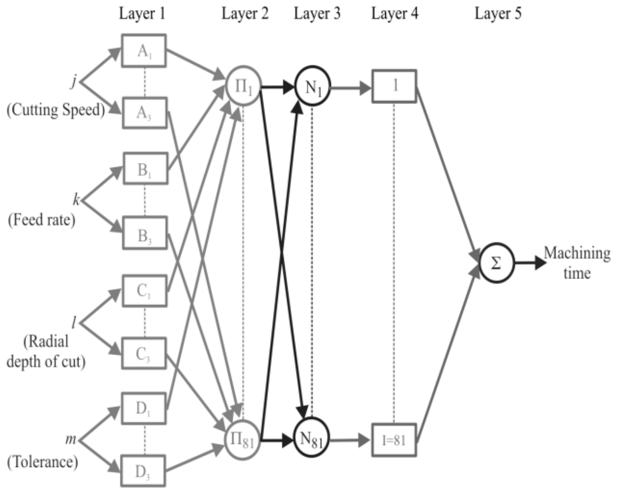

Fig. 1 A four input and one output (machining time) neuro-fuzzy network model for end milling.

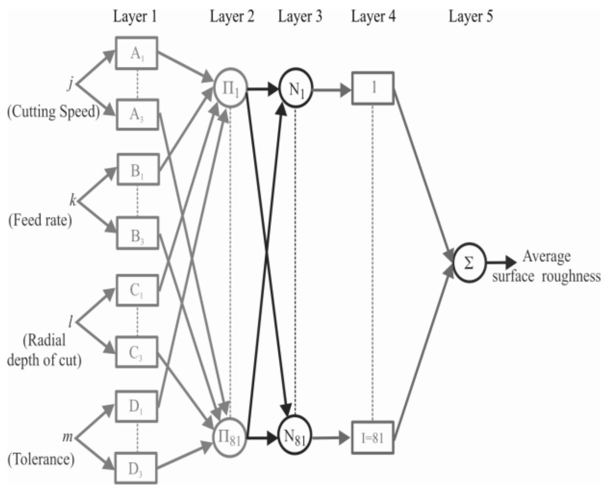

Fig. 2 A four input and one output (average surface roughness) neuro-fuzzy network model for end milling.

This is linear in the consequent parameters. The forward pass of the learning algorithm continues up to nodes at layer 4 and consequent parameters are determined by the method of least squares. In the backward pass, the error signal propagates backward to update the premise parameters by gradient descent.

The close agreement of the experimental values reported by Tansel et al. and the computed values after training NF model in table 2 clearly indicates that the model can be used for predicting the values in the range of parameters under consideration and is suitable to act as function approximator in QIET. The model is very fast and the time taken for prediction is negligible. The training information of Neuro-fuzzy model is shown in table 3.

Number of nodes:	193
Number of linear parameters:	405
Number of nonlinear parameters:	36
Total number of parameters:	441
Number of training data pairs:	81
Number of checking data pairs:	40
Number of fuzzy rules:	81

Table 3 Training parameters of NF architecture for end milling process.

V. Quantum Inspired Neuro-Fuzzy Evolutionary Technique (QINFET) and it's Application To End Milling Process

In this section end milling process is chosen to demonstrate the effectiveness of the hybrid approach formulated by integrating Neuro-Fuzzy (NF) network models, genetic algorithms (GA) and simulated annealing (SA) for process optimization to form a novel hybrid technique namely Quantum Inspired Neuro-Fuzzy Evolutionary Technique (QINFET). The optimization is performed using the Quantum Inspired Evolutionary Technique (QIET) algorithm which requires that the fitness function is easily computable for the method to be computationally tractable [7]. A NF model is used to provide the fitness function value in the QIET. Thus QINFET uses neuro-fuzzy network model in tandem with Quantum Inspired Evolutionary Technique (QIET) in determining the optimal process parameters. The NF model intelligently determines the average surface roughness and machining time for a given set of input process parameters. Once the NF model is ready it is incorporated in the QIET algorithm for fitness evaluation while finding optimal values. This integration of NF model enables fast computation of fitness function which is the primary requirement for successful implementation of the evolutionary optimization. This approach of using a Neuro-Fuzzy in QIET is quite similar to that of meta-model. The cutting conditions of end milling process were optimized using the QINFET to obtain the best compromise between two critical machining-related values: surface roughness and machining time. Spindle speed, feed rate, radial depth of cut and tolerance were optimized, while one of the two key performance values was kept in the desired range and the other one was minimized. QINFET generated a series of alternatives for the user. The results demonstrated the compromise between the machining time and estimated surface roughness. When the minimization of the surface roughness requested, QINFET selected high cutting speed and very small feed rate. To minimize the machining time, very high cutting speed and the feed rate were selected. The surface roughness deteriorated in these cases. The tendency of the estimations of the QINFET agreed with the theoretical expectations.

The optimal parameters were found after 50 runs of QINFET algorithm for average surface roughness and machining time. Table 4 and 5 shows comparison of optimized results between those reported by Tansel et al. and QINFET algorithm.

The optimization results using QINFET as indicated in tables below show a close agreement with the optimized results reported by Tansel et al. [6] using Genetically Optimized Neural Network System (GONNS). The Quantum Inspired Neuro Fuzzy Evolutionary Technique (QINFET) is a flexible and versatile technique that can be used for intelligent modeling and optimization of process parameters.

VI. CONCLUSION

In this paper Quantum Inspired Evolutionary Technique (QIET) is presented. The technique has been carefully designed with various features that enable it to seek the near global optimum rapidly without getting stuck in the local optima. The algorithm allows a natural coding of design variables by considering continuous variables.

Range is selected for	Critical Parameters				Optimized operating conditions – the minimized critical parameter is underlined							
Machining time (min)	Machining time (min)		Surface Roughness (µm)		Cutting speed (m/min)		Feed rate (mm/tooth)		Radial depth of cut (mm)		Tolerance(mm)	
	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET
7.3-65	54.98	41.22	0.14	0.14	89.50	84.39	0.07	0.07	0.1	0.1	0.01	0.01
7.3-10	9.99	9.00	0.34	0.25	88.64	88.58	0.12	0.12	0.27	0.30	0.001	0.001
7.3-20	15.96	15.30	0.207	0.206	86.26	85.62	0.08	0.10	0.3	0.23	0.001	0.001

Table 4 Comparison of Optimization Results obtained by Tansel et al. and QINFET
(Minimization of Surface Roughness)

Range is selected for	Critical Parameters				Optimized operating conditions – the minimized critical parameter is underlined							
Surface Roughness (µm)	Machining time (min)		Surface Roughness (µm)		Cutting speed (m/min)		Feed rate (mm/tooth)		Radial depth of cut (mm)		Tolerance(mm)	
	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET	Tansel	QINFET
0.2-1.58	7.17	7.00	1.01	0.50	122.99	115.77	0.12	0.12	0.3	0.3	0.001	0.001
0.2-0.50	8.68	8.11	0.5	0.51	97.92	96.28	0.12	0.11	0.29	0.29	0.001	0.001
0.2-0.80	7.39	7.01	0.68	0.61	123	104.80	0.12	0.12	0.3	0.3	0.01	0.01

Table 5 Comparison of Optimization Results obtained by Tansel et al. and QINFET
(Minimization of Machining Time)

This may give designer more flexibility in the optimization problems. This technique is further generalized by incorporating provision to embed Neuro-Fuzzy models as fitness evaluators to create Quantum Inspired Neuro Fuzzy Evolutionary Technique (QINFET). Subsequently this new technique is applied to process optimization of end milling process and the results are presented. The proposed design scheme helps to achieve the desired level of control needed to avoid costly production problems and ensures economical production of quality products. QINFET demonstrates promise in optimizing complex industrial processes pertaining to intelligent manufacturing systems for achieving energy and material saving, quality improvement in the end product.

ACKNOWLEDGEMENTS

We gratefully acknowledge the inspiration and guidance provided by Most Revered Chairman, Advisory Committee on Education, Dayalbagh. We also acknowledge financial support for this research from UGC, AICTE and ADRDE.

References

[1] K Ganesh and M Punniyamoorthy, Optimization of continuous —time production planning using hybrid genetic algorithms—simulated annealing, *The International Journal of Advanced Manufacturing Technology,vol.26, no.1-2*, 2005, 148–154.

[2] G Nallakumarasamy, P Srinivasan, K Venkatesh Raja, and R Malayalamurthi, Optimization of operation sequencing in CAPP using simulated annealing technique (SAT), *International Journal of Advanced Manufacturing Technology, vol. 54, no. 5-8,* 2011, 721–728.

[3] K Hans Raj, R S Sharma, G S Mishra, A Dua, C Patvardhan, An evolutionary computational technique for constrained optimisation of engineering design, *Institution of Engineers (India) Journal, vol. 86*, 2005, 121-128.

[4] K H Han and J H Kim, Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization, *IEEE Transactions on Evolutionary Computation, vol. 6, No. 6.*, 2002, 671-682.

[5] K H Han and J H Kim, Quantum-inspired evolutionary algorithm with a new termination criterion, H gate, and two-phase scheme, *IEEE Transactions on Evolutionary Computation, vol. 8,* 2004, 156-169.

[6] I N Tansel, B Ozelik , W Y Bas, P Chen, D Rincon, S Y Yang, A Yenilmeg, Selection of optimal cutting conditions by using GONNS, *International Journal of Machine Tools & Manufacture, 46,* 2006, 26-35.

[7] K Hans Raj, S Rahul, S Rajat, A Swarup, J Rochak, A Quantum Seeded Evolutionary Computational Technique for Constrained Optimization in Engineering Design, *Proceedings in the XXXI National Systems Conference, MIT, Manipal vol.1*, 2007, 876-881.