©2010 International Journal of Computer Applications (0975 - 8887)
Volume 1 – No. 22

An Adaptive Quantum Evolutionary Algorithm for Engineering Optimization Problems

Ashish Mani
U.S.I.C.
D.E.I., Dayalbagh
Agra, India

C. Patvardhan
Electrical Engineering Deptt.
D.E.I., Dayalbagh
Agra, India

ABSTRACT
Real world problems in engineering domain are typically constraint optimization problems. An Adaptive Quantum Evolutionary Algorithm for solving such problems is proposed in this paper. The proposed technique uses a novel qubits representation for search and optimization and uses feasibility rules for handling constraints. Moreover, it does not need stochastic ranking or niching or other methods for maintaining diversity. It does not even require mutation and local heuristics. The algorithm is tested on a standard set of four widely studied benchmark engineering design optimization problems. The results obtained are better than the existing state of the art approaches. The proposed algorithm is simple in concept and implementation, while being robust.

General Terms
Algorithms,

Keywords
Quantum, Evolutionary Algorithm, Engineering Optimization, Constraint Handling.

1. INTRODUCTION
Engineering optimization problems are mostly constrained optimization problems in continuous variables, which are formulated as follows:

Optimize $f(x)$, where $x = (x_1, x_2, \ldots, x_N) \in R^N$,
Such that: $x_{il} \le x_i \le x_{iu}$; x_i is i^{th} variable with x_{il} and x_{iu} as its lower and upper limits.
$g_j(x) \le 0$; g_j is j^{th} inequality constraint and $j = 1 \ldots p$.
$h_k(x) = 0$; h_k is k^{th} equality constraint and $k = 1 \ldots q$.

The objective function as well as inequality and equality constraints are often nonlinear, non-convex and non-differentiable. Such problems cannot be solved by traditional calculus-based methods and enumerative strategies [1]. Evolutionary Algorithms (EA) have been applied to solve such complex constrained optimization problems.

EAs are population based stochastic search and optimization techniques inspired by nature's laws of evolution. They are popular due to their simplicity and ease of implementation. However, they suffer from issues like premature convergence, slow convergence, stagnation and are sensitive to the choice of the crossover and mutation operators and parameters. Many efforts have been made by researchers to overcome such limitations by establishing a good balance between exploitation and exploration. A typical EA is designed using selection, crossover, mutation operators and local heuristic. The mutation operator is used for escaping from local minima i.e. improving exploration and local heuristic is used for increasing the convergence rate i.e. improving exploitation. However, all such attempts tend to use user-selectable parameters in their algorithm design. Thus, the balance struck between exploration and exploitation in a specific EA has a user bias rather than problem bias.

Quantum Evolutionary Algorithm (QEA) [2] was proposed to improve the balance between exploration and exploitation. QEAs are evolutionary algorithms inspired by the principles of Quantum Mechanics. They are developed by drawing some ideas from quantum mechanics and integrating them in the current framework of EAs. The important principles of Quantum mechanics are superposition, entanglement, interference and measurement [3]. The principles mostly utilized in designing QEAs are superposition and measurement [4] and have been used for improving diversity as reported in the literature. Another interesting observation is the use of single qubit (quantum analog of classical bit, and is governed by the principles of quantum mechanics) representation in almost all the efforts, thus ruling out use of any other principles of quantum mechanics such as entanglement.

This paper proposes to utilize two qubits representation instead of one qubit representation. This helps in utilizing entanglement and superposition principles of quantum computing for improving the search. Further, a parameter free adaptive quantum crossover operator inspired by the phase rotation has been designed to generate new population. The effect of using different strategies in the proposed crossover operator is studied in detail. The proposed QEA does not require mutation operator and local heuristic for avoiding premature convergence and improving convergence rate respectively. The QEA uses Feasibility Rules [5] for handling constraints as EAs essentially perform unconstrained search. Therefore, in order to effectively and efficiently handle constraints, Feasibility Rules strategy is employed, which is free from fine-tuning of any penalty parameters.

The rest of the paper is organized as follows. The design of proposed algorithm is presented in Section 2. The testing and results are discussed in Section 3. Section 4 concludes with a brief summary and direction of further work.

2. ALGORITHM
Feynman had originally proposed that Quantum Mechanical Systems can be used for computing purpose and can be a better alternative than classical Von Neumann computers for simulating quantum mechanical phenomena. Thus, Quantum

computing made its birth and gained popularity by the development of polynomial time Shor's factoring algorithm and Grover's algorithm for quick search in unsorted database [3]. However, these Quantum Algorithms can be implemented efficiently on Quantum Computers and not on classical computers. The development in hardware of quantum computer is still in its nascent stages with a number technological challenges to overcome before a quantum computer of significance can be commercially available. However, since the quantum computing paradigm is now widely believed to be more powerful than classical computing, the quantum inspiration can be used for improving classical stochastic algorithms. Quantum Evolutionary algorithms are such an effort in integrating the principles of quantum computing and Evolutionary Algorithms.

The smallest information element in quantum computer is a qubit, which is quantum analog of classical bit. The classical bit can be either in state 'zero' or in state 'one' whereas a quantum bit can be in a superposition of basis states in a quantum system. It is represented by a vector in Hilbert space with $|0\rangle$ and $|1\rangle$ as the basis states. The qubit can be represented by vector $|\psi\rangle$, which given by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (1)$$

where $|\alpha|^2$ and $|\beta|^2$ are probability amplitudes of the qubit to be in state $|0\rangle$ and $|1\rangle$ respectively and should satisfy the condition:

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2)$$

The qubits can store, in principal exponentially more information than classical bits. However, these qubits exist in quantum computing systems and are constrained by several limitations like they collapse to one of the basis states upon measurement and can be evolved by using unitary transformations. The simulation of qubits is inefficient on classical computers.

In QEA, the probabilistic nature of qubits has been widely used for maintaining diversity [4]. A single qubit is attached to the solution vector and the solution is obtained by taking measurement or collapsing in binary coded as well as real coded QEA. The qubits associated with the solution vector is also evolved by using quantum gate operators, which are influenced by phase rotation transformation used in Grover's Algorithm for searching unsorted database. Further, the past efforts have also used mutation operator and local heuristics [2].

An interesting feature of such implementation is no direct correspondence between the solution vector and qubits especially in case of real coded QEA [2]. The quantum rotation gates / operators also behave independent of information from the problem as well as the solution domain assuming that the quantum behavior would help in reaching the solution. However, it should not be forgotten that such algorithms are to be run on classical computer without simulating any quantum phenomena. Further, it can be argued that increasing the diversity by collapsing the solution qubit may affect the exploitation of the solution as the solution found in the next iteration even of a good candidate solution may end up being far worse due to probabilistic implementation.

This paper takes a different approach for designing the QEA by using not only qubit representation and associated superposition principle but also entanglement principal. Entanglement is one of the fundamental principles and if two qubits are entangled then performing any quantum operation on one of the qubits would affect the state of other qubit also i.e. there is a relation between the two qubits, which can be utilized for computation purposes. The proposed QEA uses two qubits per solution vector to utilize entanglement principle. An important point to remember is that the entanglement principle is being integrated in a classical algorithm, so the implementation would be classical. This QEA tries to overcome the limitations associated with classical EAs. The classical EA being a black box optimization technique has only objective function value as the domain information regarding a specific problem. This feedback is mostly used in the selection phase and not for directly controlling the crossover or mutation operators or even local heuristic. Therefore, the feedback through objective function value is not being utilized properly. The algorithm uses the second qubit, which stores the information regarding the objective function value of the solution vector. This provides information regarding the solution domain as well as problem function domain made available simultaneously. The information stored in first and second qubit are entangled to harness the power of the important entanglement principle. The first qubit influences the second qubit as the probability amplitude of the first qubit would determine the objective function value and hence the probability amplitude of second qubit. The second qubit influences the first qubit as the parameter free adaptive quantum rotation crossover operator used for evolving the first qubit uses the probability amplitude of the second qubit. Any operation performed on either of the two qubits would affect the other and so they are entangled in classical implementation.

2.1 Proposed QEA

The first set of qubits $|\psi_{1i}\rangle$ stores the current value of the ith variable as amplitude α_{1i} whose value [0, 1]. The upper and lower limits of variables are scaled between 0 and 1. The amplitude β is not stored as it can be computed from equation 2. Therefore, the number of qubits per quantum register QR_1 is equal to the number of variables. The quantum register stores the qubits. The number of quantum registers has also been made a function of the number of variables in the specific problem. Thus, giving it a problem bias rather than user bias. The number of QR_1 is 100 times the number of variables. The number of QR_1 is kept high in order to utilize the benefit of superposition and quantum rotation in EA. The structure of QR_1 is shown below:

$QR_{11} = [\alpha_{111}, \alpha_{112} \ldots \alpha_{11n}]$

$QR_{12} = [\alpha_{121}, \alpha_{112} \ldots \alpha_{12n}]$

…………………………………

$QR_{1100N} = [\alpha_{1100n1}, \alpha_{1100n2} \ldots \alpha_{1100nn}]$

The second set of qubits ψ_{2i} stores the ranked and scaled objective function value of the ith solution vector as amplitude α_{2i} whose value [0, 1]. The fittest vector's objective function value is assigned 1 and the worst vector is assigned value 0. The rest of the solution vectors' objective function value is ranked and assigned in between the 0 and 1. Another alternative was normalizing the solution vector's objective function value between 0 and 1 and assigning it to α_{2i}. However, it was found that amplitude amplification due to quantum phase rotation led to disruptive rotation around the fittest vector.

Quantum Rotation based Adaptive Cross-over (QRACO): Quantum gates are used for evolving the qubits in quantum computing paradigm. Quantum phase rotation gate has been used in Grover's algorithm for amplitude amplification in searching the marked element in an unsorted database. Most of the efforts have used rotation gates for evolving the qubits. A quantum rotation based adaptive and parameter tuning free cross-over operator is designed for the QEA. The second qubits's amplitude is used for determining the angle of rotation for evolving the first qubit. The following equation is used for the purpose:

$$\psi_{1i}(t+1) = \psi_{1i}(t) + f(\psi_{2i}(t), \psi_{2j}(t))*(\psi_{1j}(t) - \psi_{1i}(t)) \quad (3)$$

where t is iteration number, ψ_{1j} can be the best solution vector or any other randomly or deterministically selected solution vector. All solution vectors are rotated towards best solution vector when ψ_{1j} is the best solution. When the solution vector is randomly or deterministically picked then inferior solution is rotated towards better solution. In case of best solution, it is rotated away from the inferior solution. Therefore, the rotation crossover operator balances the exploration and exploitation and converges the solution vector adaptively towards global optima by using three strategies viz. Rotation towards Best (R-I), Rotation away from worse (R-II) and Rotation Towards Better (R-III).

The function $f(\psi_{2i}(t), \psi_{2j}(t))$ controls gross and fine search. Presently, $f(\psi_{2i}(t), \psi_{2j}(t))$ generates a random number either between α_{2i} and α_{2j} or $|\alpha_{2j}|^2$ and $|\alpha_{2i}|^2$. The value $|\alpha_{2j}|^2 - |\alpha_{2i}|^2$ is generally smaller than $\alpha_{2j} - \alpha_{2i}$, thus the later is used for gross search and former for fine search. The salient feature of the new quantum rotation crossover operator is that it adaptively changes each variable in the solution vector and at the same time is problem driven rather than being an arbitrary choice of the user. Fixed rotation was also attempted by using $|\alpha_{2j}|^2 - |\alpha_{2i}|^2$ and $\alpha_{2j} - \alpha_{2i}$, but failed as it reduces the diversity and spoils the balance between exploration and exploitation.

Constraint handling is one of the main issues in constrained optimization. The choice of the technique tends to have serious impact on the quality of the algorithm as EAs are generically unconstrained optimization algorithms. Constraint handling has been implemented using Feasibility rules discussed in [6]. It is free from fine-tuning of penalty parameters.

2.2 Constraint Handling

Constraint handling is one of the main issues in constrained optimization as the choice has serious impact on the quality of the algorithm. Constraint handling is implemented using Deb's Feasibility rules [6]:

i. If both solutions are feasible, the one with better objective function value wins.

ii. If one solution is feasible and the other infeasible, the feasible one wins.

iii. If both solutions are infeasible, the one with lower constraint violation wins.

Feasibility Rules is again parameter free but reduces the diversity of the population [7]. In order to improve diversity, EAs using Feasibility Rules often incorporate niching and other associated techniques. However, the proposed QEA does not require any such technique. Tournament selection is used for selecting the next generation of solution vector by comparing with their respective best evolved vector.

2.3 Pseudo code for Proposed QEA

The Pseudo code for Proposed AQEA is given below:

```
Random_Initialize (QR1);
while (!termination_criteria)  {
        f(x) = Compute_fitness(QR1);
        QR2 = Rank_Scale(f(x));
        QRACO(QR1, QR2);
        Tournament_Selection(QR1);  }
Print(Result);
```

3. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed algorithm has been tested on four standard benchmark engineering optimization problems, which have been widely used for testing similar constraint handling optimization algorithms.

The proposed algorithm has been implemented in 'C' programming language on an IBM Workstation with Pentium-IV 2.4 GHz processor, 2GB RAM and Windows XP platform. Thirty independent runs have been performed for each problem in each experiment. The testing has been performed for determining the stability of the proposed algorithm. The stability is determined by analyzing statistically the quality of the solutions produced for each problem in thirty independent runs. The efficiency is determined by the number of function evaluations and by plotting the convergence graph.

The first problem (P-1) is designing of a welded beam for minimum cost, subject to some constraints [10]. The objective is to find the minimum fabrication cost, considering four design variables and constraints of shear stress, bending stress in the beam, buckling load on the bar, and end deflection on the beam.

The second problem (P-2) is designing a compressed air storage tank with a working pressure of 3,000 psi and a minimum volume of 750 ft^3. A cylindrical vessel is capped at both ends by hemispherical heads. Using rolled steel plate, the shell is made in two halves that are joined by two longitudinal welds to form a cylinder. The objective is to minimize the total cost, including the cost of the materials forming the welding [10]. The design variables are: thickness, thickness of the head, the inner radius and the length of the cylindrical section of the vessel. The variables x_1 and x_2 are discrete values, which are integer multiples of 0.0625 inch.

The third problem (P-3) is designing of the speed reducer [11] and is concerned with the face width, module of teeth, number of teeth on pinion, length of the first shaft between bearings, length of the second shaft between bearings, diameter of the first shaft, and diameter of the first shaft (all variables are continuous except number of teeth on pinion that is integer). The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shaft.

The fourth problem (P-4) is minimizing the weight of a tension/compression spring, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. There are three design variables: the wire diameter, the mean coil diameter, and the number of active coils [10].

The experiments have been performed to gain better insight into the working of the proposed algorithm by analyzing the QRACO in four different configurations. The first is standard configuration (SC-I) as described in section 2 and the results are given in Table 1. The second configuration (SC-II) uses deterministic rotation instead of random rotation in standard configuration and the results are given in Table 2. The results in Table 1 and 2 show that random rotation is a better design option than deterministic rotation. The third configuration (SC-III) uses R-I and R-II of standard configuration i.e. random exploration is turned off and the results are given in Table 3. The results in Table 1 and 3 show that exploration part of the QRACO is performing its job. The fourth configuration (SC-IV) uses deterministic rotation and R-I and R-II of standard configuration and result is given in Table 4. This test was performed to validate the choice of each component in designing QRACO and the results as predicted were much worse than in former studies.

Table 1. Results with standard configuration (SC-I)

Prob-#	P-1	P-2	P-3	P-4
Best	1.724852	6059.714	2996.348	0.012665
Median	1.724854	6061.203	2996.348	0.012666
Mean	1.725497	6072.980	2996.348	0.012667
S.D.	0.002706	15.491	0.000000	2.24E-06
Worst	1.738919	6090.708	2996.348	0.012672

Table 2. Results with deterministic rotation (SC-II)

Prob-#	P-1	P-2	P-3	P-4
Best	1.724852	6059.714	2996.348	0.012667
Median	1.777314	6410.100	3018.266	0.012711
Mean	1.968457	6572.043	3020.834	0.012769
S.D.	0.377568	0535.804	20.03446	0.000147
Worst	3.320852	7710.367	3055.462	0.01316

Table 3. Results with limited exploration (SC-III)

Prob-#	P-1	P-2	P-3	P-4
Best	1.728963	6210.053	3008.528	0.012671
Median	1.975894	7054.347	3046.714	0.01317
Mean	2.038391	7194.338	3061.622	0.013132
S.D.	0.319265	686.4489	49.14603	0.000358
Worst	3.34185	8510.765	3183.587	0.014096

Table 4. Experimental Results (SC-IV)

Prob-#	P-1	P-2	P-3	P-4
Best	1.942917	6214.444	3036.677	0.012928
Median	2.429257	9008.829	3072.224	0.013405
Mean	2.453762	8978.313	3097.475	0.014623
S.D.	0.462394	1958.365	58.18803	0.001898
Worst	3.838626	13017.60	3208.92	0.018834

Further, Table 5 presents experimental results when standard BLX-α crossover operator with α = 0.3 is used instead of QRACO. The results in Table 1 and 5 show that the QRACO is an improvement over an efficient BLX-α crossover operator.

Table 5: Results with BLX-α crossover operator

Prob-#	P-1	P-2	P-3	P-4
Best	1.726097	6133.133	2996.348	0.012671
Median	1.907752	6778.268	2996.348	0.012868
Mean	1.945073	6771.723	2996.348	0.012902
S.D.	0.148081	196.2644	0.000	0.000187
Worst	2.376852	7332.842	2996.348	0.013296

Table 6 presents comparison of the proposed Adaptive RQEA (ARQEA) with existing state of art algorithms like RQIEA [4], ECPSO [8] and DTS [9] on best solution on problems P-1, P-2 and P-4. The results of Table 1 and Table 6 show that the proposed algorithm named ARQEA is far better than the earlier proposed RQIEA, which is a quantum inspired EA and is also better than ECPSO and DTS.

Table 6: Comparison of known algorithms (Best Solution)

Prob-#	P-1	P-2	P-4
ARQEA	**1.7248520**	**6059.714**	**0.012665**
RQIEA[4]	1.7513172	6088.568	0.012680
ECPSO [8]	1.7248600	6059.714	0.012669
DTS [9]	1.7282260	6059.946	0.012681

Figure 1 shows the convergence graphs for Problem P-1of the four configurations of the proposed ARQEA (median of Table 1 to 4) and BLX-α version (median of Table 5) indicated by T-1 to T-5 respectively on the graph. The proposed ARQEA is efficient as takes around 20K function evaluations to reach near optima.

Figure 2 shows the convergence graphs for Problem P-2of the four configurations of the proposed ARQEA (median of Table 1 to 4) and BLX-α version (median of Table 5) indicated by T-1 to T-5 respectively. The proposed ARQEA is efficient as takes around 75K function evaluations to reach the vicinity of optima.

Figure 3 shows the convergence graphs for Problem P-3 of the four configurations of the proposed ARQEA (median of Table 1 to 4) and BLX-α version (median of Table 5) indicated by T-1 to T-5 respectively. The proposed ARQEA is efficient as takes around 110K function evaluations to reach the vicinity of optima.

Figure 4 shows the convergence graphs for Problem P-4of the four configurations of the proposed ARQEA (median of Table 1 to 4) and BLX-α version (median of Table 5). The proposed ARQEA is again efficient as takes around 25K function evaluations to reach the vicinity of optima.

4. CONCLUSIONS AND FUTURE WORK

Constrained Optimization is an important problem in engineering domain for which a new adaptive quantum inspired evolutionary algorithm is proposed. The algorithm uses two qubit representation instead of one and utilizes the quantum entanglement and superposition principles hitherto not tapped. It does not require mutation or local heuristic for improving solution quality but still provides better solutions than the state of the art approaches. Future work would involve in-depth analyses to understand the working of the proposed algorithm.

5. ACKNOWLEDGMENTS

We are extremely grateful to Most Revered Chairman, Advisory Committee on Education, Dayalbagh, for continued guidance and support in our every endeavor.

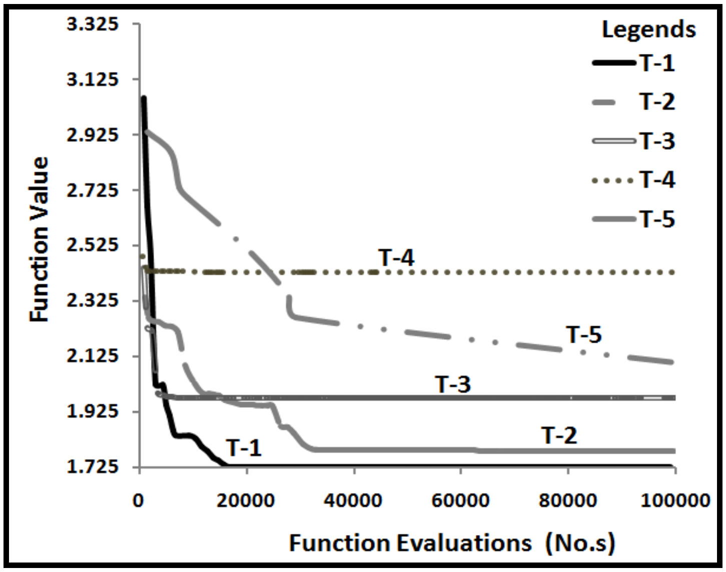

Fig 1: Convergence Graphs (Problem P-1)

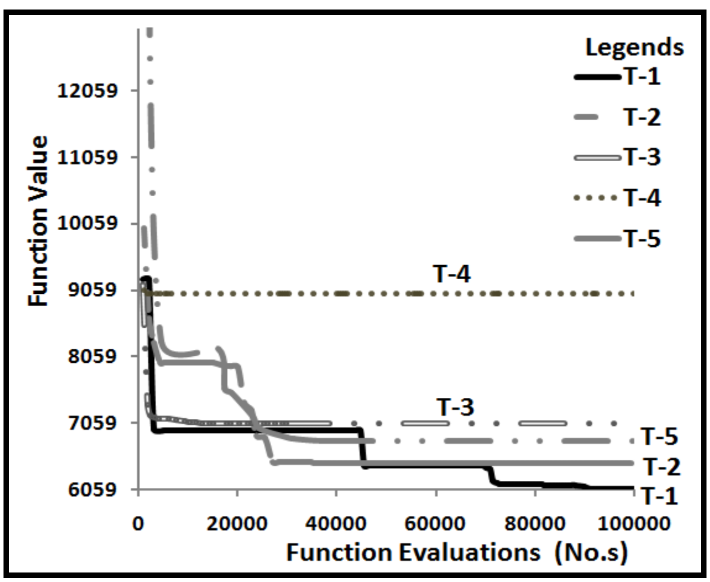

Fig 2: Convergence Graphs (Problem P-2)

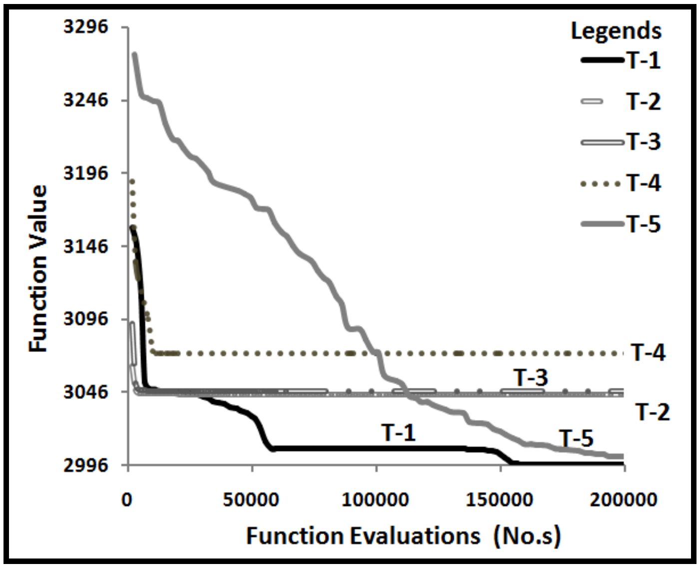

Fig 3: Convergence Graphs (Problem P-3)

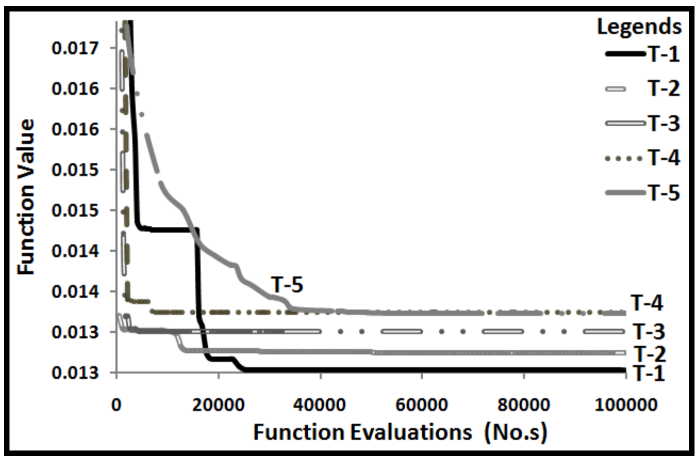

Fig 4: Convergence Graphs (Problem P-4)

6. REFERENCES

[1] Michalewicz, Z. 1996 Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin.

[2] Han, K.H., Kim, J.H. 2004 Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, Hε Gate and Two Phase Scheme. IEEE Trans. Evo. Co. 8, 2, 156—168.

[3] Nielsen M.A., Chuang I.L. 2006 Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

[4] Alfares F.S., Esat I.I. 2007 Real-coded Quantum Inspired Evolutionary Algorithm Applied to Engg. Optimiz. Probs. In: 2nd Int. Symp. On LAFMVV, 169-176, IEEE C S.

[5] Herrera F., Lozano M. Sanchez A.M. 2003 A Taxonomy for the crossover operator for Real-coded Genetic Algorithms. Int. J. Intelligent Systems, 18, 309--338.

[6] Deb, K. 2000 An efficient constraint handling method for genetic algorithms. J. Comput. Methods Appl. Mech. Eng., 186, 2/4, 311--338.

[7] Coello, C.A.C. 2002 Theoretical and numerical constraint handling techniques used with evolutionary algorithms, J. Comput. Methods Appl. Mech., 191, 1245--1287.

[8] He, Q., Wang, L., Huang, F. 2008 Nonlinear Constrained Optimization by Enhanced Co-evolutionary PSO. Proc. IEEE CEC, pp. 1436-1441.

[9] Coello, C.A.C., Montes, E.F. 2002 Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. J. Adv. Engrg. Informatics, 16, 192--203.

[10] Golinski, J. 1973 An Adaptive Optimization System Applied to Machine Synthesis. Mech. Mach. Theory, 8, 4, 419--436.