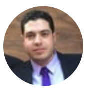
Quantum Crossover Based Quantum Genetic Algorithm for Solving Non-linear Programming

Conference Paper · May 2012

4 authors:


Amgad Monir
University of Deusto


N.A. El-Hefnawy
Menofia university, Egypt, Computers and infor…


Mahmoud Moustafa El-Sherbiny
Institute of Statistical Studies and Research (IS…


Mohiy M. Hadhoud
Minoufiya University

Quantum Crossover Based Quantum Genetic Algorithm for Solving Non-linear Programming

Amgad. M. Mohammed , N. A. Elhefnawy
Operations Research Dept., Menofia University
Menofia, Egypt
nancyabbas_1@hotmail.com
amgad.elsayed@ci.menofia.edu.eg

Mahmoud. M. El-Sherbiny
Operations Research Dept.,
Institute of Statistical Studies
& Research, Cairo, Egypt
m_sherbiny@yahoo.com

Mohiy M.Hadhoud
Information Technology Dept,
Menofia University,Menofia,Egypt
mmhadhoud@yahoo.com

***Abstract*—Quantum computing proved good results and performance when applied to solving optimization problems. This paper proposes a quantum crossover-based quantum genetic algorithm (QXQGA) for solving non-linear programming. Due to the significant role of mutation function on the QXQGA's quality, a number of quantum crossover and quantum mutation operators are presented for improving the capabilities of searching, overcoming premature convergence, and keeping diversity of population. For calibrating the QXQGA, the quantum crossover and mutation operators are evaluated using relative percentage deviation for selecting the best combination. In addition, a set of non-linear problems is used as benchmark functions to illustrate the effectiveness of optimizing the complexities with different dimensions, and the performance of the proposed QXQGA algorithm is compared with the quantum inspired evolutionary algorithm to demonstrate its superiority.**

Keywords-component (Quantum Computing; Quantum Evolutionary Algorithms; Quantum Crossover operator; Quantum Mutation operator; Convergence; Non-linear optimization)

I. INTRODUCTION

Research on quantum computing varies between working with quantum computers and working with hybrid intelligent algorithms to run onto classical computers. There are well-known quantum algorithms such as Shor's quantum factoring algorithm [1] and Grover's algorithm for searching in an unstructured database [2].Evolutionary algorithms which are essentially stochastic search methods have gained much attention and wide applications, used for intelligent optimization [3], used in designing sequential logic circuits [4], used for clustering [5].Later researches on merging quantum computing principles with evolutionary algorithms have appeared under the name of Quantum Inspired Evolutionary Algorithms [6], [7] .Quantum Genetic Algorithms (QGA) proved good results and performance when applied to applications such as solving combinatorial optimization problems [7] , blind source separation [8], feature selection algorithm [9] [10], quantum algorithms for handling probabilistic interval, and fuzzy uncertainty [11] , optimizing capacitated vehicle routing problem [12] . A unified framework and a comprehensive survey of recent work in quantum-inspired evolutionary algorithms and the differences between them can be found in [13],also merging quantum computing with particle swarm evolutionary algorithms proved good results in optimizations [14],[15] and have been applied in network routing [16] .The qubit representation for the elements of the population is a key point for the use of the quantum algorithm, a classical population can be generated by measuring the quantum population, and then its best elements are used to update the quantum population via quantum rotation gate [17] .Quantum computing inspires people with the new idea of using the quantum information representation and the quantum information processing mode to improve the performance of conventional intelligent algorithms. This paper focuses on evaluating the effects of different combinations based on quantum crossover operator and quantum mutation operator to propose a new algorithm QXQGA for overcoming on the complexities of non-linear optimization problems.

The organization of the remaining content is as follows: Section II presents the original principles of quantum genetic algorithm. Section III presents different quantum crossover and quantum mutation operators. In section IV the proposed algorithm is introduced. Section V explains parametric analysis and effectiveness of factors on solution quality. Section VI evaluates the proposed algorithm on numerical optimization to investigate its effectiveness. Finally, the conclusion and future work are reported in section VII.

II. PRELIMINARY KNOWLEDGE OF QGA

QGA is qubit based encoding chromosomes, such that each chromosome can be found in superposition of many states at quantum population and it represents only one state in classic population by applying measurement [17], [8]. According to the information of optimal individual in a population, quantum gates can lead chromosomes to update themselves.

A. *Qubit Encode*

Qubit is the smallest information unit in quantum computing, it represents a two - state quantum system.

Compared with classical-bit, the state of quantum-bit $|\psi\rangle$ can be any linear superposition state between "0" and "1" state,

namely:

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

The system will give "0" state when the whole probability is $|\alpha|^2$ and will give "1" state when the whole probability is $|\beta|^2$ [18], also normalization condition must be satisfied such that

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

B. Representation Of Single Chromosome

One of the most important issues when designing the GA lies on its solution (chromosome) representation. In order to construct a direct relationship between the problem domain and the QXQGA, the proposed coding scheme (chromosome) represented by qubit encoding structure such that every chromosome contains M-qubits can be found in a superposition of 2^M states at the same time, but will represent single state after collapsing.

- Representation Example

A Q-bit chromosome as a string of m-qubits as described in [17], [13]

$$\left[\left[\begin{array}{c|c|c|c} \alpha\ 1 & \alpha\ 2 & \alpha\ 3 & \ldots\ldots\ldots\ \alpha\ m \\ \beta\ 1 & \beta\ 2 & \beta\ 3 & \beta\ m \end{array}\right]\right]$$

- Numerical Example

$$\left|\begin{array}{c|c|c} 1/\sqrt{2} & \sqrt{3}/2 & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/2 & 1/\sqrt{2} \end{array}\right| \qquad (3)$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1$ and i=1, 2, 3 .this represents a linear probabilistic superposition of 2^3 states.

$$-\frac{\sqrt{3}}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle + \frac{1}{4}|011\rangle - \frac{\sqrt{3}}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle + \frac{1}{4}|111\rangle \qquad (4)$$

The above system can output states $|000\rangle$, $|001\rangle$,$|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, $|111\rangle$ with probability of 3/16, 3/16, 1/16, 1/16, 3/16, 3/16, 1/16 and 1/16 respectively. Also single gene can contain multiple qubits depending on application as in [8].

C. Quantum Rotational gate

QGA more efficient than GA, since updating of population in GA depends on blind operators such as crossover and mutation which may diverge the generated offspring from our near optimal solution. Applying rotational gate in QGA faces the previous challenge since the role of rotational gate is updating qubits of worst chromosomes in the same direction of better ones. Equation (5) is used as Q-gate in our proposed algorithm with Lookup table of θi in Table I.

$$\cup(\theta i) = \begin{bmatrix} \cos\theta i & -\sin\theta i \\ \sin\theta i & \cos\theta\ i \end{bmatrix} \qquad (5)$$

Where θi is rotation angle in equation (6)

$$\theta i = s(\alpha_i, \beta_i) * \Delta\theta i \qquad (6)$$

$s(\alpha_i, \beta_i)$ is the sign of θi that determines the direction , $\Delta\theta i$ is the magnitude of rotation gate [7].

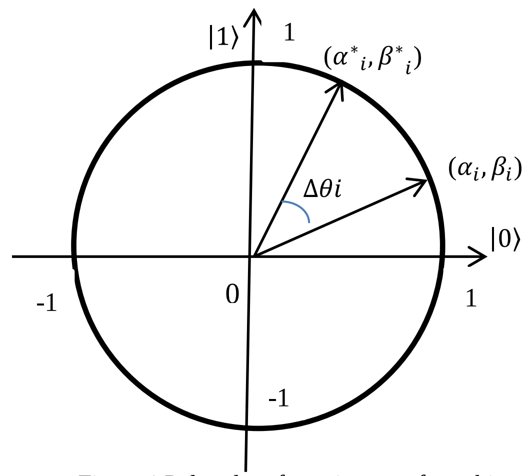


Figure.1.Polar plot of rotation gate for qubit

TABLE I. LOOKUP TABLE OF θi

x_i	b_i	f(x)≥f(b)	Δθi	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i>0$	$\alpha_i\beta_i<0$	$\alpha_i=0$	$\beta_i=0$
0	0	false	0	0	0	0	0
0	0	true	0	0	0	0	0
0	1	false	0	0	0	0	0
0	1	true	.08pi	1	-1	0	±1
1	0	false	0	0	0	0	0
1	0	true	.08pi	-1	1	±1	0
1	1	false	0	0	0	0	0
1	1	true	0	0	0	0	0

$$\begin{bmatrix} \alpha^*{}_i \\ \beta^*{}_i \end{bmatrix} = U(\theta i)\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos\theta i & -\sin\theta i \\ \sin\theta i & \cos\theta\ i \end{bmatrix}\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \qquad (7)$$

The new updated α^* and β^* can be found by equation (7) as shown in Fig.1

D. Basic structure of QGA

The original existence of QGA was without any crossover and mutation operators [6], [7]. For simple flowchart of QGA refer to [19].

III. QUANTUM CROSSOVER AND QUANTUM MUTATION

In order to improve the quality of proposed QXQGA algorithm, in this section four types of crossover operators and two types of mutation operators are presented and implemented to make eight combinations from them.

- The common feature applied to all crossover operators is crossover probability of .7 that

> *if (random number < .7)*
>
> *then allow genes to cross*
>
> *else*
>
> *keep parent chromosomes without change*

- Before crossover takes place we select two parents randomly and their index should not be same

A. *single-point crossover operator*

A simple crossover which allows qubits to be crossed between mutated parents after a single crossover point is generated randomly. The single-point quantum crossover operator is represented in Fig.2, also equation (2) must be satisfied after applying this operator.

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Parent one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	**Parent Two**
0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0 1.0000	0.9859 0.1672	0.9204 0.391	**Offspring one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Offspring Two**

Figure.2. Single- point Quantum Crossover Operator

B. *Two-point crossover operator*

In this operator we choose randomly two crossover points, but they must be different, then only qubits between these points are exchanged and other qubits kept without change. The two-point quantum crossover operator is represented in Fig.3, also equation (2) must be satisfied.

C. *Multi-point crossover operator*

Randomly m-crossover points will be chosen with unique positions. The qubits in even areas are exchanged and qubits in odd areas kept without change. The multi-point quantum crossover operator is represented in Fig.4, also known as double-point crossover [20].

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Parent one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	**Parent Two**
0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0 1.0000	0.9859 0.1672	0.0783 0.9969	**Offspring one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0.646 0.7633	0.2757 0.9613	0.9204 0.391	**Offspring Two**

Figure 3. Two- point Quantum Crossover Operator

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Parent one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	**Parent Two**
0.9098 0.4151	1.0000 0	−0.7278 0.6858	0 1.0000	0.9859 0.1672	0.0783 0.9969	**Offspring one**
0.274 0.9617	0.1121 0.9937	−0.7762 0.6305	0.646 0.7633	0.2757 0.9613	0.9204 0.391	**Offspring Two**

Figure 4. Multi- point Quantum Crossover Operator

D. *Quantum Interference crossover operator*

In this operator all individuals in a population interferes with each other to form new offsprings. It provides more diversity in population as proved in [21]. The quantum interference crossover operator is represented in Fig. 5.where all genes with same color will compose single quantum chromosome. For explaining refer to [21].

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Parent one**
0.274 0.9617	1.0000 0	−0.7762 0.6305	0 1.0000	0.9859 0.1672	0.9204 0.391	**Parent Two**
0.8893 0.4573	0.9531 0.3025	0.9835 0.1808	0.6446 0.7645	0.7525 0.6586	0.3130 0.9498	**Parent Three**
0.8594 0.5113	0.7462 0.6657	0.6733 0.7394	0.3124 0.9499	0.9839 0.1789	0.5145 0.8575	**Parent Four**

Figure 5. Quantum Interference Crossover Operator

1) Quantum Inversion Mutation Operator

After applying crossover our mutation operator is applied to achieve both exploration and exploitation. The role of mutation operator is changing the structure of qubit in quantum chromosome. We change $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$. The quantum inversion mutation operator is represented in Fig.6.

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Before mutation**
0.9098 0.4151	0.1121 0.9937	0.6858 −0.7278	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**After mutation**

Figure 6. Quantum Inversion Mutation Operator

2) Left and right Quantum swap mutation operator

In this operator two qubits chosen randomly left qubit and right qubit , where classical binary bit related to left qubit differs from classical binary bit of right qubit ,then their $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$ are changed . The left and right quantum swap mutation operator is represented in Fig.7.

0.9098 0.4151	0.1121 0.9937	−0.7278 0.6858	0.646 0.7633	0.2757 0.9613	0.0783 0.9969	**Before mutation**
0.9098 0.4151	0.2757 0.9613	−0.7278 0.6858	0.646 0.7633	0.1121 0.9937	0.0783 0.9969	**After mutation**

Figure 7. Left and right Quantum swap mutation operator

IV. THE PROPOSED ALGORITHM

Step 1: Initialize generation number t=1, create quantum population Qp(t) with N-individuals and M-qubit genes.

Step 2: collapse each quantum individual , form classic binary-represented population Cp(t) and evaluate each individual ,with recording the best one in ***b.***

Step 3: If stopping condition satisfied, then output the best solution and stop, otherwise go on the following steps.

Step 4: If local stagnation occurred, overcoming it by applying multi-point quantum crossover with crossover probability = 1 and applying quantum inversion mutation with mutation probability =1 , otherwise go to Step 5.

Step 5: Apply two-point quantum crossover with crossover probability =.7 and apply quantum inversion mutation with mutation probability =1.

Step 6: collapse the quantum system before applying rotational gate, since the new resulted quantum population after crossover and mutation may contain chromosome with fitness value better than the reserved one.

Step 7: Apply quantum rotational gate (table.I) on Qp(t) to obtain Qp(t+1).

Step 8: Collapsing each individual on Qp(t+1) to obtain Cp(t+1), and evaluating each individual to take a decision about updating the reserved best individual ***b*** or not.

Step 9 : increase generation counter t=t+1, and go to step 3.

Simple flowchart of the proposed algorithm presented in fig 9.

V. PARAMETRIC ANALYSIS

In this section the best combination between crossover operators and mutation operators will be chosen from the implemented eight ones to solve well known benchmark functions, and because the objective functions have different scales ,they couldn't be used directly. We will use Relative Percentage Deviation (RPD) for each combination [22] .

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} * 100 \qquad (8)$$

Where Alg_{sol} represents the value of objective function at specific combination and Min_{sol} represents the minimum objective function value during all different combinations for the same problem. After converting the objective values to RPDs, the mean RPD is calculated for each combination. The tested benchmark function shown in table.II, different combination abbreviations shown in table.III and Tables from IV to VII *(originally its one table with size 8*16, split into four tables due to formatting)* shows that two-point crossover operator with quantum inversion operator have minimum RPD value. And this combination is employed in next section to be tested against other algorithm to prove superiority of (QXQGA).

TABLE II. TESTED BENCHMARK FUNCTIONS

F	Mathematical Representation	Range
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	$-5 \le x_i \le 5$
F2	$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$-50 \le x_i \le 50$
F3	$f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	$-100 \le x_i \le 100$
F4	$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \le x_i \le 600$

TABLE III. ABBREVIATIONS

Functions		Crossover Operators	
F1	DeJong(Sphere)	X1	Single-point Crossover
F2	Rosenbrock	X2	Two-point Crossover
F3	Rastrigin	X3	Multi-point Crossover
F4	Griewangk	X4	Quantum Interference Crossover
		Mutation Operators	
		M1	Quantum Inversion Mutation
		M2	Left &Right Quantum Swap Mutation

TABLE IV.RPD OF DIFFERENT COMBINATIONS (DC) FOR FIRST 4 COLUMNS

DC	Dimension 5			
	A	*B*	*C*	*D*
	F1	*F2*	*F3*	*F4*
X1M1	0.00%	712.09%	0.00%	1.39%
X1M2	0.00%	35.50%	10.31%	2.54%
X2M1	0.00%	11.69%	0.00%	1.83%
X2M2	0.00%	106.14%	9.02%	2.33%
X3M1	0.00%	0.00%	1.29%	0.00%
X3M2	15.79%	136.11%	12.89%	5.53%
X4M1	15.79%	9.08%	1.29%	0.88%
X4M2	15.79%	156.68%	12.89%	7.12%

TABLE V. RPD OF DIFFERENT COMBINATIONS (DC) FOR SECOND 4 COLUMNS

DC	Dimension 10			
	E	*F*	*G*	*H*
	F1	*F2*	*F3*	*F4*
X1M1	0.00%	287.65%	0.00%	0.62%
X1M2	342.11%	530.35%	28.51%	4.38%
X2M1	10.53%	72.95%	1.10%	0.12%
X2M2	486.84%	434.13%	37.54%	3.00%
X3M1	42.11%	252.55%	0.00%	0.00%
X3M2	518.42%	184.78%	22.06%	4.53%
X4M1	26.32%	0.00%	1.10%	0.00%
X4M2	428.95%	202.21%	31.42%	3.27%

TABLE VI. RPD OF DIFFERENT COMBINATIONS (DC) FOR THIRD 4 COLUMNS

DC	Dimension 50			
	I	*J*	*K*	*L*
	F1	*F2*	*F3*	*F4*
X1M1	7.84%	3.94%	5.99%	0.00%
X1M2	203.44%	301.69%	109.05%	161.64%
X2M1	7.26%	14.85%	3.96%	6.29%
X2M2	198.47%	434.85%	106.71%	155.51%
X3M1	0.00%	0.00%	0.00%	8.90%
X3M2	173.86%	415.67%	114.59%	211.64%
X4M1	29.82%	20.52%	2.15%	24.36%
X4M2	234.07%	401.54%	149.68%	243.00%

TABLE VII. AVERAGE RPD FOR ALL 16 COLUMNS FROM (A-P)

DC	Dimension 100				Mean RPD
	M	*N*	*O*	*P*	
	F1	*F2*	*F3*	*F4*	
X1M1	4.35%	13.58%	3.56%	3.28%	6.14%
X1M2	35.80%	84.04%	56.83%	46.51%	12.36%
X2M1	1.65%	0.00%	3.44%	10.54%	0.00%
X2M2	50.64%	68.54%	64.55%	48.40%	14.09%
X3M1	0.00%	4.88%	0.00%	0.00%	1.12%
X3M2	36.09%	70.49%	55.91%	47.51%	12.86%
X4M1	9.62%	26.93%	6.39%	13.83%	0.29%
X4M2	50.24%	91.66%	65.03%	74.13%	13.83%

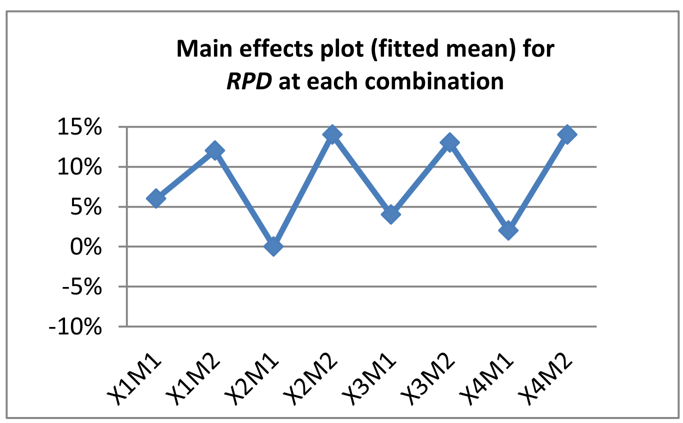


Figure 8. Main effects plot (fitted mean) for RPD at each combination

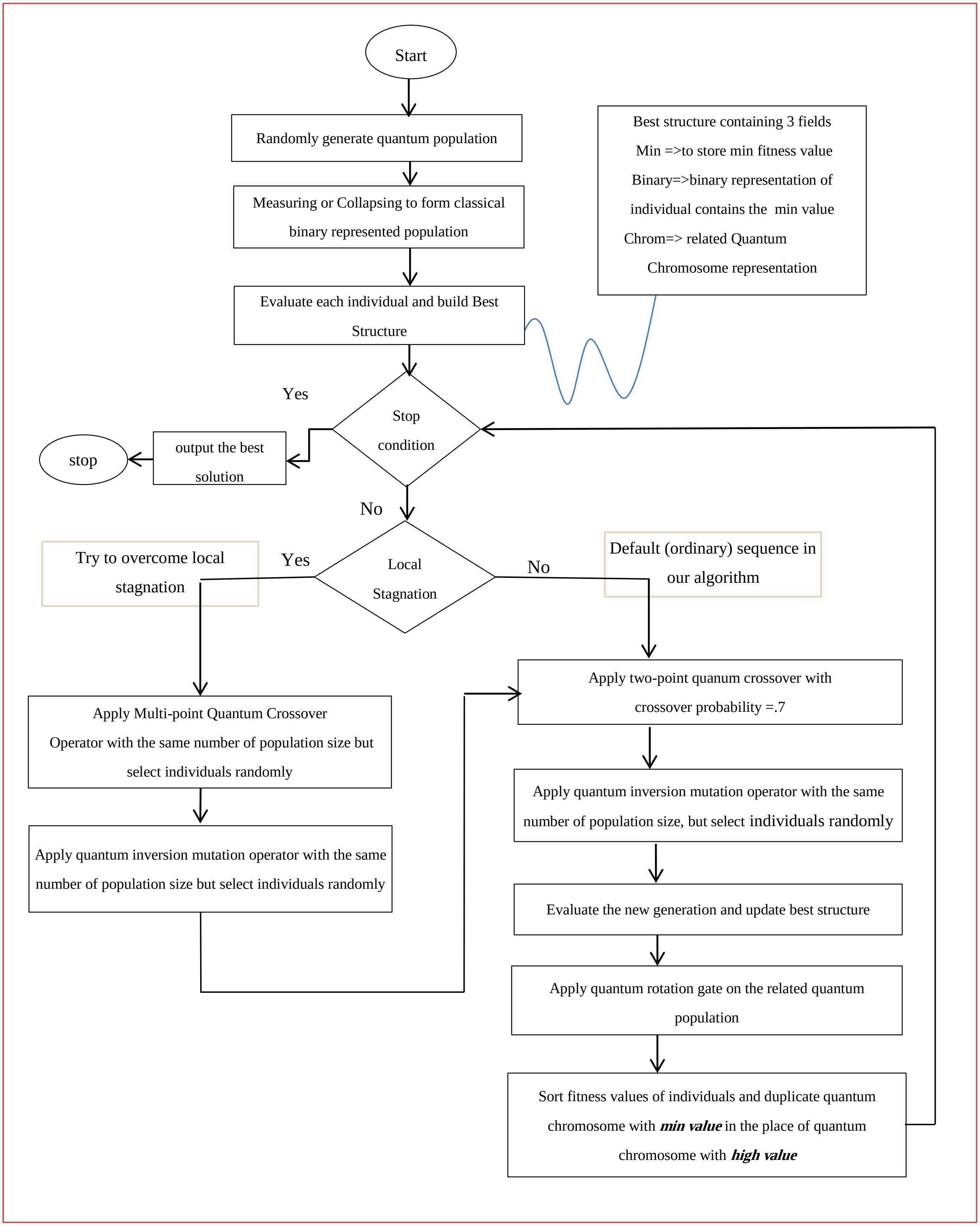


Figure 9. Simple flowchart of the proposed algorithm

Fig 8 shows that two-point quantum crossover operator with quantum inversion mutation takes rank 1, and quantum interference crossover with quantum inversion mutation operator comes in the second rank and both give better results when compared with previous quantum-inspired evolutionary algorithm.

VI. Numerical Test Results

From the previous section the best combination has been selected and employed in the proposed algorithm (QXQGA) to solve Benchmark functions in table.II. Then our results compared with quantum inspired- evolutionary algorithm (QIEA) found in [23] .

TABLE VIII. DEJONG RESULTS

DeJong Results			
[-5,5]	**Best**	**Mean**	**S.D.**
Dimension : 5			
QXQGA	0.0019	0.0019	6.50e-019
QIEA	0.0021	0.0012	0.0007
Dimension : 10			
QXQGA	0.0038	0.0045	0.0017
QIEA	0.0378	0.0090	0.0216
Dimension : 50			
QXQGA	0.7851	1.5669	0.3507
QIEA	3.427	0.289	2.562
Dimension : 100			
QXQGA	13.8916	22.5258	3.9353
QIEA	24.953	2.077	21.434

TABLE IX. ROSENBROCK RESULTS

Rosenbrock Results			
[-50,50]	**Best**	**Mean**	**S.D.**
Dimension : 5			
QXQGA	14.8264	26.2515	20.9383
QIEA	68.9335	44.1549	14.1392
Dimension : 10			
QXQGA	38.8656	253.5741	283.0384
QIEA	786.7354	362.0550	80.5617
Dimension : 50			
QXQGA	1.2962e+05	6.4386e+05	3.6653e+05
QIEA	8.751e+5	2.245e+5	3.901e+5
Dimension : 100			
QXQGA	2.6847e+07	4.7622e+07	1.1822e+07
QIEA	3.50e+7	8.47e+6	1.88e+7

TABLE X. RASTRIGIN RESULTS

Rastrigin Results			
[-100,100]	**Best**	**Mean**	**S.D.**
Dimension : 5			
QXQGA	20.7335	22.3089	1.8556
QIEA	18.687	4.579	7.285
Dimension : 10			
QXQGA	44.2471	49.4364	3.1886
QIEA	47.485	10.104	25.542
Dimension : 50			
QXQGA	804.6647	1106.7	206.6056
QIEA	1479.3	190.5	866.9
Dimension : 100			
QXQGA	7700.9	10604	1413.1
QIEA	10599.0	858.7	8244.5

TABLE XI. GRIEWANGK RESULTS

Griewangk Results			
[-600,600]	**Best**	**Mean**	**S.D.**
Dimension : 5			
QXQGA	0.7067	0.7234	0.0184
QIEA	0.398	0.092	0.182
Dimension : 10			
QXQGA	1.0135	1.0151	0.0040
QIEA	0.803	0.078	0.638
Dimension : 50			
QXQGA	3.8152	6.1671	1.2831
QIEA	10.06	1.35	5.65
Dimension : 100			
QXQGA	53.0803	96.3253	17.6807
QIEA	173.36	21.76	131.43

Both QXQGA and QIEA are run 30 times. The first column in tables from VIII to XI lists the minimal (overall best) objective value found during the 30 runs. The second and third column lists the mean and standard deviation (S.D) for the best value found in each of the 30 runs. The results performed with population size=50, number of generations=200 and crossover probability =.7 which are the same conditions and initial experimental arguments in [23] . Results prove superiority of QXQGA against quantum inspired evolutionary algorithm and the ability of the proposed algorithm to solve complex problems with different large scale dimensions.

VII. CONCLUSIONS

In this paper, a quantum crossover based quantum genetic algorithm (QXQGA) has been proposed for solving complex non-linear programming problems. QXQGA controls the search direction via updating the population by using a multi-point quantum crossover operator. Eight combinations of quantum crossover and quantum mutation operators have been implemented and analyzed using relative percentage deviation (RPD) to choose the most effectiveness combination for used with the proposed QXQGA algorithm. A set of well-known benchmark functions are used for comparing the performance of the QXQGA algorithm and the quantum inspired evolutionary algorithm to demonstrate. Due to achieving exploration and exploitation in search space, overcoming premature convergence and presenting diversity in population the proposed QXQGA algorithm can be considered as a best alternative search evolutionary algorithm for solving the non-linear optimization problems. Future work will be held on adaptive crossover and mutation operators depending on the fitness value of chromosomes, also changing in quantum rotational gate will be considered.

REFERENCES

[1] P. W.Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science* , Los Alamitos, CA, 1994, pp. 124—134.

[2] L. K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack," in *Phys. Rev. Lett. 79(2)*, 1997, pp. 325–328.

[3] L. Wang., "Intelligent Optimization Algorithms with Applications," in *Tsinghua University & Springer Press*, Beijing, 2001.

[4] B. ALI A., E. A. ALMAINI and T. KALGANOVA, "Evolutionary Algorithms and Their Use in the design of sequential logic circuits," *Genetic Programming and Evolvable Machines*, 5(1), pp. 11-29, March 2004.

[5] Eduardo R. Hruschka, Ricardo J. G. B. Campello,Alex A. Freitas and André C. P. L. F. de Carvalho, "A Survey of Evolutionary Algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 39(2), pp. 1-26, March 2009.

[6] A. Narayanan and M. Moore, "Quantum inspired genetic algorithmss," *IEEE International Conference on Evolutionary Computation (ICEC'96)*, Nagoya, Japan, 1996, pp. 41-46.

[7] K. H. Han and J H. Kim, "Quantum—inspired evolutionary algorithm for a class of combinatorial Optimization," *IEEE Transactions on Evolutionary Computation*, 2002, pp. 580—593.

[8] Yang Junan , Li Bin and Zhuang Zhenquan, "Research of Quantum Genetic Algorith and its application in blind source separation," *Journal of Electronics*, 20(1), pp. 62-68, Jan 2003.

[9] Gexiang Zhang , Laizhao Hu and Weidong Jin, "Resemblance Coefficient and a Quantum Genetic Algorithm for Feature Selection," *Discovery Science*, Verlag Berlin Heidelberg, 2004, pp. 155-168.

[10] Gexiang Zhang , laizhao Hu and Weidong Jin, "Quantum Computing Based Machine Learning Method and Its Application in Radar Emitter Signal Recognition," *Modeling Decisions for Artificial Intelligence*, Verlag Berlin Heidelberg, 2004, pp. 1-12.

[11] Mark Martinez, Luc Longpre, Vladik Kreinovich, Scott A. Starks and Hung T. Nguyen "Fast Quantum Algorithms for Handling Probabilistic, Interval, and Fuzzy Uncertainty," *Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'03*, Chicago, Illinois, 2003, pp. 395-400.

[12] Jing-Ling Zhang, Yan-Wei Zhao,Dian-Jun Peng and Wan-Liang Wang, "A Hybrid Quantum-Inspired Evolutionary Algorithm for Capacitated Vehicle Routing Problem," *ICIC 2008*, Verlag Berlin Heidelberg, 2008, pp. 31-38.

[13] Gexiang Zhang, "Quantum-inspired evolutionary algorithms: a survey and empirical study ," *Journal of Heuristics*, 17(3), pp. 303-351, June 2010.

[14] S. N. Omkar ,Rahul Khandelwal, T. V. S. Ananth,G. Narayana Naik and S. Gopalakrishnan, "Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Systems with Applications*, 36(8), pp. 11312-11322, Oct. 2009.

[15] Zhang Zhisheng, "Quantum-behaved particle swarm optimization algorithm for economic load dispatch of power system," *Expert Systems with Applications*, 37(2), pp. 1800-1803, Mar. 2010.

[16] Jun Sun ,Wei Fang,Xiaojun Wu ,Zhenping Xie and Wenbo Xu , "QoS multicast routing using a quantum-behaved particle swarm optimization algorithm," *Engineering Applications of Artificial Intelligence*, 24(1), pp. 123-131, February 2011.

[17] Shude Zhou and Zengqi Sun, "A New Approach Belonging to EDAs: Quantum-Inspired Genetic Algorithm with Only One Chromosome," *Advances in Natural Computation*, Verlag Berlin Heidelberg, 2005, pp. 141 – 150.

[18] Eleanor.R and Wolfgang P, "An introduction to quantum computing for non-physicists," *ACM Computing Surveys (CSUR)*, 32(3), pp. 300-335, Sept 2000.

[19] Jia-Chu Lee, Whei-Min Lin, Gwo-Ching Liao and Ta-Peng Tsao, "Quantum genetic algorithm for dynamic economic dispatch with valve-point effects and including wind power system," *International Journal of Electrical Power & Energy Systems*, 33(2), pp. 189–197, Febuary 2011.

[20] Qun Niu ,Fang Zhou and Taijin Zhou, "Quantum Genetic Algorithm for Hybrid Flow Shop Scheduling Problems to Minimize Total Completion Time," *Life System Modeling and Intelligent Computing*, Verlag Berlin Heidelberg, 2010, pp. 21–29.

[21] Hongwei Dai and Cunhua Li, "Improved Quantum Interference Crossover-based Genetic Algorithm and Its Application," *First International Conference on Intelligent Networks and Intelligent Systems*, 2008, pp. 35 - 38.

[22] Khalid M. Altassan, Mahmoud M. El-Sherbiny and Ahmed D. Abid, "A Novel Artificial Immune Algorithm for Solving Fixed Charge Transportation Problems," *2nd International conference on Innovation and Information Management*, China, 2012, pp. 1-7.

[23] Kai Fan ,Anthony Brabazon, Conall O'Sullivan and Michael O'Neill, "Quantum-Inspired Evolutionary Algorithms for Financial Data Analysis," *Applications of Evolutionary Computing*, Verlag Berlin Heidelberg, 2008, pp. 133–143.