A Sinusoid Size Ring Structure Quantum Evolutionary Algorithm

M.-H. Tayarani-N.
Bahar Institute of Higher Education
Mashhad, Iran
tayarani@ieee.org

M.-R. Akbarzadeh-T.
Ferdowsi University of Mashhad
Mashhad, Iran
akbarzadeh@ieee.org

***Abstract* — This paper proposes a dynamic ring architecture of interaction among members of population in a Quantum Evolutionary Algorithms (QEA). The ring is allowed to expand/collapse based on a sinusoidal population size and partial re-initialization of new members in the population. The study shows that the ring structure can be an efficient architecture for an effective Exploration/Exploitation tradeoff, and the partial re-initialization of the proposed algorithm can improve the diversity of the algorithm. The proposed approach is tested on Knapsack Problem, Trap Problem as well as 14 numerical optimization functions. Experimental results show that the proposed Structure consistently improves the performance of QEA.**

***Keywords*—Quantum Evolutionary Algorithms, Ring structure population, dynamic structure.**

I. INTRODUCTION

One of the most important parameters of Evolutionary algorithms is the population size. A large population size has a better search performance but more computational complexity, while small population size has a better computational complexity with less searching efficiency. Reference [1] studies the optimal population size of genetic algorithms based on a defined partial function. An adaptive population size for a distributed genetic algorithm is proposed in [2] for video segmentation. Reference [3] proposes a variable population size for genetic algorithms with periodic reinitialization that follows a saw-tooth scheme. In [4] a competitive frequency-based methodology is proposed to explore the least proper population size for GA. Reference [5] presents the mathematics formula for researching the optimal population size of Partheno-Genetic Algorithm. In [6], a novel dynamic population size is proposed. In this approach the initial population size is initialized randomly, and during the evolution, the size of population is tuned by a mechanism called "suppression". A new scheme for adjusting the population size in genetic algorithm is proposed in [7] that provides a balance between exploration and exploitation. This new schema is used for cell placement in the chip design process. A variable population size and a hybridization of PSO with GA are proposed in [8] which can improve the performance of GA and PSO.

Variable population size schema is also used in Particle Swarm Optimization. Considering the small effect of numbers of particles on PSO performance, [9] proposes a method which finds the best population size for PSO to make a tradeoff between optimization precision and optimization speed.

One of the major parameters of evolutionary algorithms is the structure of the population. A graph based evolutionary algorithm is proposed in [10] in which the individuals are located on the nodes of a graph structured population. The effect of variable population structure on Particle Swarm Optimization is investigated in [11]. Random graphs and their performance on several criteria are compared in their work.

Reinitialization operator is another approach for improving the diversity and hence performance of the evolutionary algorithms. In order to maintain the diversity of population, [12] proposes a genetic algorithm called Diversity Control GA which uses a reinitialization schema. For optimization in dynamic environments, [13] proposes two strategies for population re-initialization when a change in the environment is detected. The first strategy is the prediction of the new optimum according to previous searches. The second proposed strategy is to perturb the current population with a Gaussian noise with variance which is estimated according to previous changes. A PSO algorithm for finding the shortest path in graphs is used in [14], which is improved with using a reinitialization method and a local search operator.

Quantum Evolutionary Algorithm (QEA) is an approach in which chromosomes are coded after quantum states of electrons in a probabilistic fashion. The resulting architecture is highly suitable to preserve diversity, i.e. each chromosome consists of m Q-bits that is equivalent to 2^m states. In quantum informatics, the basic carrier of information is not a bit but a quantum system with two states such as in an atom, an ion or a photon with two polarized directions, or the Q-bit. A Q-bit is in a linear superposition state and is used to specify the amplitudes of two states. In [17, 18] quantum-inspired evolutionary algorithms are investigated for a class of combinatorial optimization problems in which quantum rotation gates act as update operators. This quantum rotation gate is also used in a novel parallel quantum GA for hierarchical ring model and infinite impulse response (IIR) digital filter design [19]. Reference [20] proposes a quantum evolutionary algorithm for multi-objective optimization and quantum rotation gate.

This paper proposes a Sinusoid size Ring structured Quantum Evolutionary Algorithm (SRQEA) with a partial reinitialization of population in the form of a sinusoid function. In the proposed, random q-individuals are inserted in the ring structured population during the expansion cycles of population; while the inferior q-individuals are eliminated

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

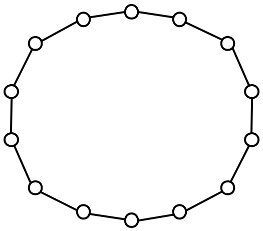


Figure 1. The ring structure of the proposed algorithm

during the shrinking cycles of decreasing population. The ring interaction structure restricts the interactions among quantum individuals to only their (left and right) neighbors, and hence better diversity of the population is maintained. The variable size of the population with partial reinitialization has a fine effect on diversity preserving. This paper also tries to find the best parameters of sinusoid function for population size for the tested problems. The proposed algorithm is tested on Knapsack Problem, Trap Problem as well as 14 numerical benchmark functions for various dimensions of the problems. Experimental results show that the proposed algorithm consistently improves the performance of QEA.

This paper is organized as follows. Section II describes the QEA representation. In Section III the ring structure for QEA is proposed. Section IV proposes the sinusoid population size for ring structured QEA. Finding the best angle frequency for sinusoid function of population size for each problem is performed in section V. In Section VI the proposed algorithm is evaluated on some benchmark functions and finally the proposed algorithm is concluded in section VII.

II. QEA

QEA is inspired from the principles of quantum computation, and its superposition of states is based on qubits, the smallest unit of information stored in a two-state quantum computer. A qubit could be either in state "0" or "1", or in any superposition of the two as described in (1):

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Where α and β are complex numbers, which denote the corresponding state's appearance probability, following below constraint:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

This probabilistic representation implies that if there is a system of m qubits, the system can represent 2^m states simultaneously. At each observation, a qubits quantum state collapses to a single state as determined by its corresponding probabilities.

A. Representation

QEA uses a novel representation based on the above concept of qubits. Consider *i-th* individual in *t-th* generation defined as an *m*-qubit as (3):

$$q_i^t = \begin{bmatrix} \alpha_{i,1}^t & \alpha_{i,2}^t & \ldots & \alpha_{i,k}^t & \ldots & \alpha_{i,m}^t \\ \beta_{i,1}^t & \beta_{i,2}^t & & \beta_{i,k}^t & & \beta_{i,m}^t \end{bmatrix} \tag{3}$$

Where$|\alpha_{i,k}^t|^2 + |\beta_{i,k}^t|^2 = 1$, k=1,2,…,m , m is the number of qubits, i.e., the string length of the qubit individual, i=1,2,…,m, n is the population size and t is generation number of the evolution. Since a qubit is a probabilistic representation, any superposition of states is simultaneously represented. If there is, for instance, a three-qubits (m = 3) individual such as (4):

$$q_i^t = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{\sqrt{2}}{\sqrt{3}} & \frac{\sqrt{3}}{2} \end{bmatrix} \tag{4}$$

Alternatively, the possible states of the individual can be represented as:

$$\begin{aligned} q_i^t = &\tfrac{1}{2\sqrt{6}}|000\rangle + \tfrac{1}{2\sqrt{2}}|001\rangle + \tfrac{1}{2\sqrt{3}}|010\rangle + \tfrac{1}{2}|011\rangle \\ &+ \tfrac{1}{2\sqrt{6}}|100\rangle + \tfrac{1}{2\sqrt{2}}|101\rangle + \tfrac{1}{2\sqrt{3}}|100\rangle \\ &+ \tfrac{1}{2}|111\rangle \end{aligned} \tag{5}$$

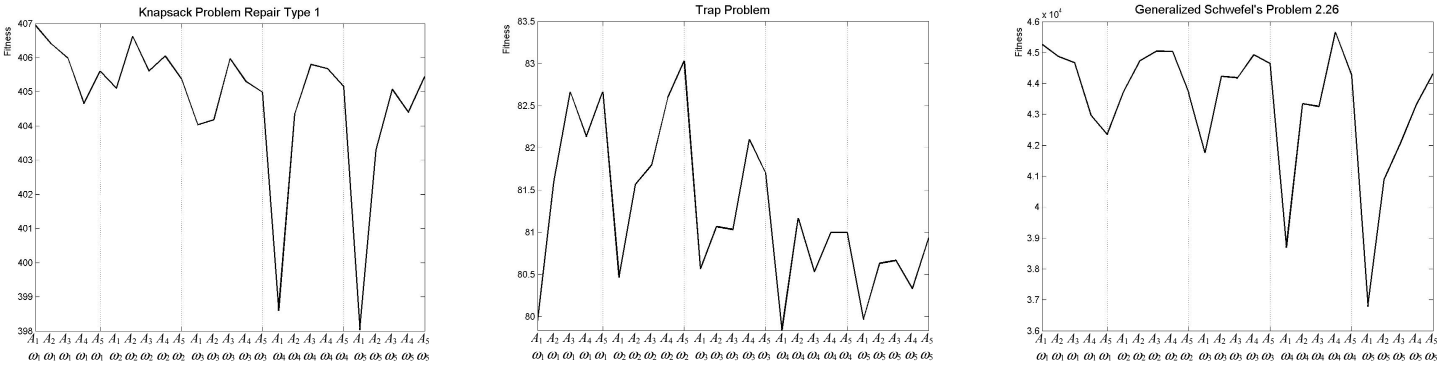
Note that the square of above numbers are true probabilities, i.e. the above result means that the probabilities to represent the states $|000\rangle, |001\rangle, |100\rangle, |010\rangle$ are 1/24, 1/8, 1/24 and 1/12 respectively. Consequently, the three-qubits system of (4) could carry all eight states information at the same time.

Evolutionary computing with the qubit representation has a better characteristic of diversity than classical approaches since it can represent superposition of states. Only one qubit individual such as (4) is enough to represent eight states, whereas in classical representation eight individuals are needed. Additionally, along with the convergence of the quantum individuals, the diversity gradually fades away and the algorithm converges.

B. Quantum Gates Assignment

The common mutation is a random disturbance of each individual, promoting exploration while also slowing convergence. Here, the quantum bit representation can be simply interpreted as a biased mutation operator. Therefore, the current best individual can be used to steer the direction of this mutation operator, which will speed up the convergence. The evolutionary process of quantum individual is completed through the step of "update $Q(t)$." A crossover operator, quantum rotation gate, is described below. Specifically, a qubit individual q^t_i is updated using the rotation gate $U(\theta)$ in this algorithm. The k-th qubit of the i-th quantum individual generation t, $[\alpha_{i,k}^t \quad \beta_{i,k}^t]^T$ is updated as:

$$\begin{bmatrix} \alpha_{i,k}^{t+1} \\ \beta_{i,k}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_{i,k}^t \\ \beta_{i,k}^t \end{bmatrix} \tag{6}$$



(a) (b) (c)

Figure 2. Finding the best sinusoid function for Knapsack Problem, Trap Problem and Generalized Schwefel Problem, where $\omega_1 \ldots \omega_5=\left(\frac{\pi}{25},\frac{\pi}{50},\frac{\pi}{100},\frac{\pi}{200},\frac{\pi}{300}\right)$, $A_1 \ldots A_5=(0.1,0.3,0.5,0.7,0.9)$.

Where $\Delta\theta$ is rotation angle and controls the speed of convergence and determined from Table I. Reference [17] shows that these values for $\Delta\theta$ have better performance.

III. A RING STRUCTURE FOR QEA

One of the main parameters of evolutionary algorithms is the structure of the algorithm. Ring structured Evolutionary Algorithms are structured evolutionary algorithms in which the individuals are located in a ring structured population and each individual only interacts with its neighbors. In a ring environment, the connections among neighbors help the algorithm to exploit possible solutions of the algorithm, and the overlapped small neighborhoods help algorithm to explore the search space. So the advantage of the ring structure is that the fitness and genotype diversity in the population is preserved for a long number of generations. This paper uses the ring structure for QEA for two reasons. Firstly, the small connectivity among the individuals in the ring structured population preserves the diversity in the population. Secondly, the sinusoid variable sizing of the population can be trivially applied to a ring structure population. The ring structure of the proposed algorithm is showed in Fig 1.

In the ring structured population each individual in the population is connected to two neighbors. Suppose that the size of the ring structured population is S. The neighbors of the individual q_i are then defined as:

$$N_i = \{q_i, q_{i'}, q_{i''}\}$$

Where:

$$i' = \begin{cases} i-1 & i \neq 1 \\ S & i = 1 \end{cases}, \quad i'' = \begin{cases} i+1 & i \neq S \\ 1 & i = S \end{cases}$$

The proposed algorithm is described as below:

```
Procedure SRQEA
begin
      t=0
    initialize quantum population Q(0) by
      ring structure with the size of $S(0)=\overline{S}$
    make X(0) by observing the states of
      Q(0).
    evaluate X(0).
    for all binary solutions $x^0_i$ in X(t) do
      begin
        find neighborhood set $N_i$ in X(0).
        find binary solution x with best
          fitness in $N_i$
        save x in $B_i$
      end
    while not termination condition do
      begin
        t=t+1
        $S(t)=\overline{S}+A\overline{S}(\sin(\omega t))$
        if S(t)>S(t-1) create random q-
          individuals
        if S(t)<S(t-1) eliminate the q-
          individuals with worst observed
          fitness
        make X(t) by observing the states of
          Q(t-1)
        evaluate X(t)
        update Q(t) based on $B_i$ and X(t) using
          Q-gates
        for all binary solutions $x^t_i$ in X(t) do
          begin
            find neighborhood set $N_i$ in X(t).
            select binary solution x with best
              fitness in $N_i$
            if x is fitter than $B_i$ save x in $B_i$
          end
      end
end
```

TABLE I. LOOKUP TABLE OF $\Delta\theta$. $f(x_i^t)$ IS THE FITNESS OF BINARY SOLUTION x_i^t AND $f(B_i)$ IS THE FITNESS OF B_i

$x^t_{i,k}$	$b_{i,k}$	$f(x_i^t) \geq f(B_i)$	$\Delta\theta$
0	0	*False*	0
0	0	*True*	0
0	1	*False*	0.01π
0	1	*True*	0
1	0	*False*	-0.01π
1	0	*True*	0
1	1	*False*	0
1	1	*True*	0

The comprehensive description of the proposed algorithm is:

1. In the initialization step, the quantum-individuals q^0_i are located in a ring structured population with the size of:

$$S(0) = \bar{S}$$

Where, $\bar{S}$ is the average population size.

Then $[\alpha^0_{ik} \quad \beta^0_{ik}]^T$ of all q^0_i are initialized with $1/\sqrt{2}$, where i=1,2,…,S is the location of the q-individuals in the ring structured population, k=1,2,...,m, and m is the number of qubits in the q-individuals. This implies that each qubit individual q^0_i represents the linear superposition of all possible states with equal probability.

2. This step makes a set of binary instants $X(0) = \{x^0_i \mid i = 1,2,..., S\}$ at generation t=0 by observing $Q(0) = \{q^0_i \mid i = 1,2,..., S\}$ states, where $X(t)$ at generation t is a random instant of qubit population and S is the size of ring. Each binary instant, x^0_i of length m, is formed by selecting each bit using the probability of qubit, either $|\alpha^0_{i,k}|^2$ or $|\beta^0_{i,k}|^2$ of q^0_i. Observing the binary bit $x^t_{i,k}$ from qubit $[\alpha^t_{i,k} \quad \beta^t_{i,k}]^T$ performs as:

$$x^t_{i,k} = \begin{cases} 0 & if \quad \mathrm{R}(0,1) < |\alpha^t_{i,k}|^2 \\ 1 & \quad \text{otherwise} \end{cases} \qquad (6)$$

Where $\mathrm{R}(\cdot,\cdot)$ is a uniform random number generator.

3. Each binary instant x^0_i is evaluated to give some measure of its fitness. In this step, the fitness of all binary solutions of X(0) are evaluated.

4,5,6,7. In these steps the neighborhood set N_i of all binary solutions x^0_i in X(0) are found and the best solution among N_i is stored in B_i. B_i is the best possible solution, which the q-individual q^t_i and its neighbors had reached.

8. The while loop is terminated when the termination condition is satisfied. Termination condition here is when maximum number of iterations is reached.

9. In the proposed algorithm, the size of the population is a sinusoid function of iteration number. In this step, S(t), the size of the population in iteration t is calculated as:

$$S(t) = \bar{S} + A\bar{S}(\sin(\omega t))$$

Where, $\bar{S}$ is the average size of the population, t is the iteration number and ω and A are the angle frequency and the amplitude of the sinusoid size population respectively. The best values for ω and A is found in section V.

10. If S(t), the size of the population in iteration t is greater than S(t-1), it means that the size of the population is increased. So creating random q-individuals, until the size of ring structured population be equal to S(t).

11. If S(t), the size of the population in iteration t is smaller than S(t-1), eliminate the q-individuals which have the worst observed solution, until the size of ring structured population reaches S(t).

12. Observing the binary solutions X(t) from Q(t).

13. Evaluating the binary solutions X(t).

14. The quantum individuals are updated using Q-gate.

15. The "**for**" loop is for all binary solutions x^t_i (i=1,2,…,S(t)) in the population.

16. Finding the neighbors of the binary solution x_i.

17. Find the best possible solution in the N_i, and store it to x.

18. If x is fitter than B_i, store x to B_i.

The proposed sinusoid size population has two cycles. The first cycle is increasing the size of population. In the increasing cycle, the new quantum individuals are created and inserted in the ring. Creating new random quantum individuals increases the diversity of the population and improves the exploration performance of the algorithm. The other cycle is the decreasing cycle. In this cycle, the worst quantum individuals of the population are eliminated. This treatment improves the exploitation of the algorithm by exploiting the best solutions and ignoring the inferior ones. This means that the proposed algorithm has two cycles: exploration cycle and exploitation cycle.

IV. Finding the Best Sinusoid Functions for Population Size

This paper proposes a sinusoid sized population for QEA; but the question is which amount of angular frequency ω and amplitude A of the sinusoid function is the best. For evaluation of the proposed algorithm, 14 numerical functions, Knapsack Problem and Trap Problem are used. This section tries to find the best sinusoid function for the size of the population for each benchmark functions. Fig. 2 shows the effect of the angular

TABLE II. THE BEST PARAMETERS FINED FOR SRQEA. KP1 MEANS KNAPSACK PENALTY TYPE 1 AND KR1 MEANS KNAPSACK REPAIR TYPE1

PROBLEM	A	ω	PROBLEM	A	ω
Schwefel 2.26	0.7	$\frac{\pi}{200}$	Schwefel 2.21	0.9	$\frac{\pi}{25}$
Rastrigin	0.7	$\frac{\pi}{50}$	Dejong	0.7	$\frac{\pi}{50}$
Ackley	0.5	$\frac{\pi}{50}$	Rosenbrock	0.9	$\frac{\pi}{100}$
Griewank	0.3	$\frac{\pi}{25}$	Kennedy	0.1	$\frac{\pi}{300}$
Penalized 1	0.7	$\frac{\pi}{100}$	KP1	0.1	$\frac{\pi}{25}$
Penalized 2	0.7	$\frac{\pi}{100}$	KP2	0.1	$\frac{\pi}{100}$
Michalewicz	0.7	$\frac{\pi}{100}$	KR1	0.7	$\frac{\pi}{50}$
Goldberg	0.7	$\frac{\pi}{50}$	KR2	0.1	$\frac{\pi}{50}$
Sphere Model	0.7	$\frac{\pi}{50}$	TRAP	0.9	$\frac{\pi}{50}$
Schwefel 2.22	0.7	$\frac{\pi}{50}$			

TABLE III. EXPERIMENTAL RESULTS ON THE KNAPSACK PROBLEM, TRAP PROBLEM AND FOURTEEN NUMERICAL FUNCTION OPTIMIZATION PROBLEMS. THE NUMBER OF RUNS WAS 50. **MEAN** AND **STD** REPRESENT THE MEAN AND STANDARD DEVIATION OF BEST ANSWER FOUND FOR 50 RUNS RESPECTIVELY. KP1 AND KR1 MEANS KNAPSACK PENALTY1 AND KNAPSACK REPAIR1

	$m=100$				$m = 250$				$m=500$				$m=1000$			
	QEA		SRQEA		QEA		SRQEA		QEA		SRQEA		QEA		SRQEA	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN	STD	QEA	STD	QEA	STD	QEA	STD	QEA	STD
Schwefel [16]	3.47×10^{4}	3.36×10^{3}	4.74×10^{4}	2.55	5.74×10^{4}	6.24×10^{3}	7.94×10^{4}	4.88×10^{3}	8.75×10^{4}	8.47×10^{3}	1.17×10^{5}	8.37×10^{3}	1.25×10^{5}	1.86×10^{4}	1.81×10^{5}	1.81×10^{4}
Rastrigin [16]	-1.99×10^{3}	2.27×10^{2}	-1.40×10^{3}	1.07×10^{2}	-6.83×10^{3}	4.49×10^{2}	-5.44×10^{3}	1.96×10^{2}	-1.57×10^{4}	7.23×10^{2}	-1.36×10^{4}	3.69×10^{2}	-3.54×10^{4}	1.38×10^{3}	-3.20×10^{4}	4.51×10^{2}
Ackley [16]	-17.24	0.068	-16.99	0.053	-17.62	0.12	-17.29	0.032	-17.97	0.11	-17.42	0.07	-18.34	0.089	-17.58	0.094
Griewank [16]	-38.53	6.51	-19.96	2.19	-156.94	16.51	-108.99	5.33	-3.82×10^{2}	2.59×10^{1}	-3.06×10^{2}	1.00×10^{1}	-8.79×10^{2}	3.95×10^{1}	-7.66×10^{2}	1.26×10^{1}
Penalized 1 [16]	-1.69×10^{5}	2.15×10^{4}	-8.85×10^{4}	1.14×10^{4}	-5.94×10^{5}	4.78×10^{4}	-4.47×10^{5}	2.01×10^{4}	-1.45×10^{6}	8.28×10^{4}	-1.19×10^{6}	3.33×10^{4}	-3.19×10^{6}	1.07×10^{5}	2.86×10^{6}	4.92×10^{4}
Penalized 2 [16]	-3.83×10^{4}	4.37×10^{3}	-2.13×10^{4}	1.82×10^{3}	-1.45×10^{5}	1.08×10^{4}	-1.03×10^{5}	7.58×10^{3}	-3.65×10^{5}	1.73×10^{4}	-2.89×10^{5}	8.17×10^{3}	-8.15×10^{5}	3.22×10^{4}	-7.11×10^{5}	1.58×10^{4}
Michalewicz [15]	22.84	2.55	33.14	2.08	36.07	4.28	52.51	2.65	53.23	6.55	74.11	3.78	70.94	9.09	103.28	5.17
Goldberg [3]	40.01	2.59	51.71	2.20	76.76	4.69	94.58	2.71	127.96	6.22	153.76	3.67	2.29×10^{2}	10.79	2.56×10^{2}	6.11
Sphere Model [16]	-4.32×10^{5}	5.46×10^{4}	-2.27×10^{5}	2.93×10^{4}	-1.65×10^{6}	1.83×10^{5}	-1.20×10^{6}	5.85×10^{4}	-4.22×10^{6}	2.67×10^{5}	-3.39×10^{6}	1.14×10^{5}	-9.91×10^{6}	4.50×10^{5}	-8.45×10^{6}	2.05×10^{5}
Schwefel 2.22 [16]	-4.90	0.54	-3.39	0.19	-6.41	0.39	-5.32	0.17	-7.45	0.25	-6.56	0.18	-8.18	0.21	-7.50	0.12
Schwefel 2.21 [16]	-176.56	5.86	-168.17	5.18	-190.94	2.26	-189.84	1.78	-195.83	1.03	-195.62	1.04	-198.33	0.39	-198.07	0.67
Dejong [15]	-2.47×10^{7}	6.02×10^{6}	-8.02×10^{6}	1.75×10^{6}	-3.18×10^{8}	4.71×10^{7}	-1.68×10^{8}	1.75×10^{7}	-1.79×10^{9}	1.77×10^{8}	-1.22×10^{9}	6.76×10^{7}	-9.25×10^{9}	6.25×10^{8}	-7.15×10^{9}	3.17×10^{8}
Rosenbrock [3]	-1.06×10^{5}	3.43×10^{4}	-3.71×10^{4}	6.00×10^{3}	-5.06×10^{5}	8.39×10^{4}	-2.77×10^{5}	2.46×10^{4}	-1.37×10^{6}	1.39×10^{4}	-9.81×10^{5}	3.77×10^{4}	-3.49×10^{6}	2.29×10^{5}	-2.76×10^{6}	1.03×10^{5}
Kennedy [3]	-1.22	1.07	-8.9×10^{-4}	8.44×10^{-4}	-20.99	5.42	-0.07	0.14	-74.82	12.09	-0.83	0.57	-227.30	21.05	-4.80	1.98
KR1	415.29	6.26	428.69	3.29	893.58	18.54	960.13	11.96	1742.02	30.71	1861.89	25.56	3140.68	44.05	3350.98	40.79
KR2	420.88	2.52	423.27	0.95	1011.23	13. 60	1036.12	3.99	1904.88	36.06	1967.35	23.99	3613.54	60.16	3739.79	37.16
KP1	579.22	8.06	591.16	3.47	1217.66	35.27	1357.41	12.79	2161.42	59.23	2417.01	51.09	3898.92	114.63	4370.67	96.63
KP2	399.64	5.86	411	3.03	923.82	21.42	1001.99	8.86	1701.52	36.87	1846.33	23.18	3340.36	46.39	3582.66	43.67
TRAP	72.17	4.19	82.37	1. 81	154.8	8.42	186.87	5.29	283.7	15.27	343.37	14.53	531.43	27.41	663.8	35.99

frequency and amplitude on the performance of the proposed algorithm for Generalized Schwefel's Problem 2.26, Generalized Rastrigin's Function and Ackley's function (see appendix). In Fig. 2 the vertical axis is the best fitness averaged over 50 runs and the horizontal axis is the parameters of the SRQEA. Finding the best parameters for all of the other benchmark functions is the same as in Fig 2. The best values for ω and A for all benchmark functions is summarized in Table II.

V. EXPERIMENTAL RESULTS

Here Knapsack Problem, Trap Problem and 14 numerical benchmark functions are used for testing the proposed algorithm (see Appendix). The size of population for all of the experiments is set to 20 ($\overline{S}$ =20), and maximum generation termination condition is used equal to 2000. All results are averaged over 50 runs. The value of $\Delta\theta$ is considered as Table I and the values of ω and A is set to their best values found in section IV (see Table II). The structure of SRQEA is a ring structure with sinusoid size population and the structure of the QEA is considered as the structure proposed by [17]. The experimental results are shown in Table III. In this paper the proposed algorithm is compared with the conventional structure of QEA as proposed in [17] with global migration of 100 and local migration of 30. The time complexity of the proposed algorithm is equal to the QEA because the average size of population for SRQEA is considered equal to QEA.

In all of the experiments, the proposed structure improves the performance of QEA. The least improvement is in Knapsack problem Repair type 1, and the best improvement is in Schwefel problem 2.21. Additionally, the standard deviation of the results in SRQEA is generally smaller than QEA. The improvements of the proposed algorithm become more substantial as the dimensions of the problem at hand increase.

VI. CONCLUSION

This paper proposes a sinusoid sized ring structure for quantum evolutionary algorithm that aims to improve the diversity and performance of QEA. One of the main tools for improving the performance of the evolutionary algorithms is finding a good architecture for these algorithms. Cellular, parallel and ring structures are three kinds of these structures which are powerful for making a tradeoff between exploration and exploitation in EAs. The structure of the EAs has two roles in the evolution process, first the limitation of the relationship between the individuals allows the algorithm to better explore the search space, second the connectivity between the individuals make it possible to exploit the existing solutions. We believe that either cellular or ring graph structure could potentially be used for QEA. It is still an open question that which structure is the best one.

APPENDIX

In this section two combinatorial optimization problems, Trap problem and Knapsack problem, and 14 function optimization problems are discussed to evaluate the proposed SRQEA.

i. Trap problem

Trap problem is defined as:

$$f(x)=\sum_{i=0}^{N-1}\text{Trap}(x_{5i+1},x_{5i+2},x_{5i+3},x_{5i+4},x_{5i+5}) \qquad (7)$$

Where N is the number of traps and

$$\text{Trap}(x) = \begin{cases} 4 - \text{ones}(x), & \text{if ones}(x) \leq 4 \\ 5 & \text{if ones}(x) = 5 \end{cases} \tag{8}$$

Where the function “ones” returns the number of ones in the binary string x. Trap problem has a local optimum in $(0,0,0,0,0)$ and a global optimum in $(1,1,1,1,1)$.

ii. Knapsack problem

Knapsack problem is a well-known combinatorial optimization problem which is in class of NP-hard problems [17]. Knapsack problem can be described as selecting various items x_i (i=1,2,…,m) with profits p_i and weights w_i for a knapsack with capacity C. Given a set of m items and a knapsack with capacity C, select a subset of the items to maximize the profit *f(x)*:

$$f(x) = \sum_{i=1}^{m} p_i x_i \quad , \quad \sum_{i=1}^{m} w_i x_i \leq C \, .$$

This paper considered:

$$w_i = \mathrm{R}(1, v) \, , \; p_i = \mathrm{R}(1, v)$$

Where $\mathrm{R}(\cdot,\cdot)$ is a uniform random number generator and v=10.

The use of QEA for solving Knapsack problem is described in [17].

iii. Numerical Functions

There are some benchmark numerical functions for testing the optimization algorithms. Here 14 benchmark functions are used for testing the algorithms:

Generalized Schwefel's Problem 2.26 [16], Generalized Rastrigin's Function [16], Ackley's function [16], Generalized Griewank Function [16], Generalized Penalized Function 1 [16], Generalized Penalized Function 2 [16], Michalewicz Function [15], Goldberg & Richardson Function [3], Sphere Model [16], Schwefel's Problem 2.22 [16], Schwefel's Problem 2.21 [16], Dejong Function 4 [15], Rosenbrock Function [3], and Kennedy multimodal function generator [3].

These functions have some local minima and a global minimum. We used them for maximization process, so we used –*f*(x) as fitness function.

References

[1] Dawei Li, Li Wang, “A study on the optimal population size of genetic algorithm,” *the 4th World Congress on Intelligent Control and Automation*, 2002.

[2] Se Hyun Park, Eun Yi Kim and Beom-Joon Cho, “Genetic Algorithm-Based Video Segmentation with Adaptive Population Size,” *Proceeding of 25th* DAGM *Symposium Magdeburg, Germany, September, Springer-Verlag* 2003.

[3] V. K. Koumousis, C. P. Katsaras. "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," *IEEE Trans. Evol. Comput.* vol. 10, pp. 19–28, 2006.

[4] A. Kaveh, M. Shahrouzi. “A hybrid ant strategy and genetic algorithm to tune the population size for efficient structural optimization,” *Emerald Journal of Engeneering Computations,* Vol. 24 pp 237-254.

[5] Yong, He, “Optimal Population Size for Partheno-Genetic Algorithm,” *IEEE Chinese Control Conference*, 2007.

[6] Qin Jun Kang Li-Shan, “A novel dynamic population based evolutionary algorithm for revised multimodal function optimization problem,” *IEEE Fifth World Congress on Intelligent Control and Automation*, 2004.

[7] Al Qasim, Rose Eldos and Taisir, “Population Sizing Scheme for Genetic Algorithm”, *IEEE, International Conference on Computer Systems and Applications*, 2007.

[8] Shi, X.H. Wan, L.M. Lee, H.P. Yang, X.W. Wang, L.M. Liang, Y.C. “An improved genetic algorithm with variable population-size and a PSO-GA based hybrid evolutionary algorithm,” *IEEE International Conference on Machine Learning and Cybernetics*, 2003.

[9] Fei Ma Xue-bo Chen, “Application of Varying Population Size Particle Swarm Optimization Algorithm to AGC of Power Systems,” *IEEE The Sixth World Congress on Intelligent Control and Automation*, 2006.

[10] Kenneth Mark Bryden, Daniel A. Ashlock, Steven Corns, and Stephen J. Willson, “Graph-Based Evolutionary Algorithms,” *IEEE Trans. Evol. Comput*. vol. 10, no 5, pp. 550–567, 2006.

[11] Kennedy, J. Mendes, R, “Population structure and particle swarm performance,” *IEEE Proceedings of the Congress on Evolutionary Computation*, 2002.

[12] Hisashi Shimodaira, “Methods for Reinitializing the Population to Improve the Performance of a Diversity-Control-Oriented Genetic Algorithm,” *IEICE TRANSACTIONS on Information and Systems* Vol.E84-D No.12 pp.1745-1755, 2001.

[13] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff and Edward Tsang, “Prediction-Based Population Re-initialization for Evolutionary Dynamic Multiobjective Optimization,” *Springer, bookchapter, Evolutionary Multi-Criterion Optimization*, pp. 832-846, 2007.

[14] Mohemmed, A.W. Sahoo, N.C, “Particle Swarm Optimization Combined with Local Search and Re-Initialization for Shortest Path Computation in Networks,” *IEEE Swarm Intelligence Symposium*, pp. 266 – 272, 2007.

[15] A.-R. Khorsand, M.-R. Akbarzadeh-T. “Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm”. IEEE International Conference on Systems, Man and Cybernetics, 2005.

[16] W. Zhong, J. Liu, M. Xue, L. Jiao, "A Multi-agent Genetic Algorithm for Global Numerical Optimization," *IEEE Trans. Sys, Man and Cyber.* vol. 34, pp. 1128–1141, 2004.

[17] K. Han, J. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Trans. on Evolutionary Comput*, vol., no. 6, 2002.

[18] K.H Han, K.H Park, C.H Lee, J.H Kim, “Parallel quantum-inspired genetic algorithm for combinatorial optimization problem,” *Proceeding of the* 2001 *IEEE congress on Evolutionary Computation* Seoul, Korea, 2001.

[19] G Zhung, W Jin, L Hu, “Novel parallel quantum GAs,” *Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies*, PDCAT,'2003.

[20] G Zhung, W Jin, L Hu, “Quantum evolutionary algorithm for multi-objective optimization,” *Proceedings of the 2003 IEEE International Symposium on Intelligent Control Houston*, Texas October 5-8.2003.