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.
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):

\[ |ψ⟩ = α|0⟩ + β|1⟩ \]  

(1)

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

\[ |α|^2 + |β|^2 = 1 \]  

(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):

\[
q^t_i = \left[ \begin{array}{c} \alpha_{i,1}^t \\ \beta_{i,1}^t \\ \vdots \\ \alpha_{i,m}^t \\ \beta_{i,m}^t \end{array} \right]
\]

(3)

Where \(|\alpha_{i,k}^t|^2 + |\beta_{i,k}^t|^2 = 1\), \(k=1,2,\ldots,m\), \(m\) is the number of qubits, i.e., the string length of the qubit individual, \(t=1,2,\ldots,n\) 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):

\[
q^t_i = \left[ \begin{array}{c} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{array} \right]
\]

(4)

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

\[
q^t_i = \frac{1}{2\sqrt{6}} |000⟩ + \frac{1}{2\sqrt{6}} |010⟩ + \frac{1}{2\sqrt{6}} |100⟩ + \frac{1}{2\sqrt{6}} |110⟩ + \frac{1}{2\sqrt{3}} |101⟩ + \frac{1}{2\sqrt{3}} |011⟩ + \frac{1}{2} |111⟩
\]

(5)

Note that the square of above numbers are true probabilities, i.e. the above result means that the probabilities to represent the states \( |000⟩, |010⟩, |100⟩, |101⟩\) 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 \(U(θ)\) in this algorithm. The \(k\)-th qubit of the \(i\)-th quantum individual generation \(t\), \([α_{i,k}^t \ β_{i,k}^t]\) is updated as:

\[
\begin{pmatrix}
\alpha_{i,k}^{t+1} \\
\beta_{i,k}^{t+1}
\end{pmatrix} = \begin{pmatrix}
\cos(∂θ) & -\sin(∂θ) \\
\sin(∂θ) & \cos(∂θ)
\end{pmatrix} \begin{pmatrix}
\alpha_{i,k}^t \\
\beta_{i,k}^t
\end{pmatrix}
\]

(6)
Figure 2. Finding the best sinusoid function for Knapsack Problem, Trap Problem and Generalized Schwefel Problem, where \( \omega \) and \( \theta \) are 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 shown 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:

\[
\begin{align*}
    i' &= \begin{cases} 
        i-1 & i \neq 1 \\
        S & i = 1 
    \end{cases} \\
    i'' &= \begin{cases} 
        i+1 & i \neq S \\
        1 & i = S 
    \end{cases}
\end{align*}
\]

The proposed algorithm is described as below:

Procedure SRQEA

\[
t = 0
\]

1. initialize quantum population \( Q(0) \) by ring structure with the size of \( S(0)=S \)
2. make \( X(0) \) by observing the states of \( Q(0) \).
3. evaluate \( X(0) \).
4. for all binary solutions \( x_i^t \) in \( X(t) \) do
   begin
      5. find neighborhood set \( N_i \) in \( X(t) \).
      6. find binary solution \( x_i \) with best fitness in \( N_i \).
      7. save \( x \) in \( B_i \)
   end
8. while not termination condition do
   begin
      9. \( S(t) = S + A_S(\sin(\omega \theta)) \)
      10. if \( S(t)>S(t-1) \) create random \( q \)-individuals
      11. if \( S(t)<S(t-1) \) eliminate the \( q \)-individuals with worst observed fitness
      12. make \( X(t) \) by observing the states of \( Q(t-1) \)
      13. evaluate \( X(t) \)
      14. update \( Q(t) \) based on \( B_i \) and \( X(t) \) using Q-gates
      15. for all binary solutions \( x_i^t \) in \( X(t) \) do
         begin
            16. find neighborhood set \( N_i \) in \( X(t) \).
            17. select binary solution \( x_i \) with best fitness in \( N_i \).
            18. if \( x \) is fitter than \( B_i \), save \( x \) in \( B_i \)
         end
      end
end

<table>
<thead>
<tr>
<th>( X_i^t )</th>
<th>( B_i )</th>
<th>( f(x_i^t) \geq f(B_i) )</th>
<th>( \Delta \theta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>False</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>True</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>False</td>
<td>0.01\pi</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>True</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>False</td>
<td>-0.01\pi</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>True</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>False</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>True</td>
<td>0</td>
</tr>
</tbody>
</table>

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 \( B_i \)
The comprehensive description of the proposed algorithm is:

1. In the initialization step, the quantum-individuals \(q_i^0\) are located in a ring structured population with the size of:
   \[ S(0) = S \]

Where, \(S\) is the average population size.

Then \([\alpha_i^0, \beta_i^0]\) of all \(q_i^0\), are initialized with \(\sqrt{\frac{2}{S}}\), 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 \(q_i^0\) represents the linear superposition of all possible states with equal probability.

2. This step makes a set of binary instants
   \[ X(0) = \{x_i^0| i = 1,2,..,S\} \]
   at generation \(t=0\) by observing
   \[ Q(0) = \{q_i^0| i = 1,2,..,S\} \]
   states, where \(X(t)\) at generation \(t\) is a random instant of qubit population and \(S\) is the size of ring.
   Each binary instant, \(x_i^0\) of length \(m\), is formed by selecting each bit using the probability of qubit, either \(|\alpha_i^0|^2\) or \(|\beta_i^0|^2\) of \(q_i^0\).

   Observing the binary bit \(x_{i,k}^t\) from qubit \([\alpha_i^t, \beta_i^t]\)

   performs as:
   \[
   x_{i,k}^t = \begin{cases} 
   0 & \text{if } R(0,1) < |\alpha_{i,k}^t|^2 \\
   1 & \text{otherwise}
   \end{cases}
   \tag{6}
   
   Where \(R(\cdot,\cdot)\) is a uniform random number generator.

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

   4. Each binary instant \(x_i^t\) is evaluated to give some measure of its fitness. In this step, the fitness of all binary solutions of \(X(0)\) are evaluated.

   5. Each binary instant \(x_i^t\) is evaluated to give some measure of its fitness. In this step, the fitness of all binary solutions of \(X(0)\) are evaluated.

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

   7. 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 \(\omega\) and \(A\) are the angle frequency and the amplitude of the sinusoid size population respectively. The best values for \(\omega\) 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 \(S(t)\).

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 \(\omega\) 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>
<thead>
<tr>
<th>PROBLEM</th>
<th>(A)</th>
<th>(\omega)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwefel 2.26</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Ackley</td>
<td>0.5</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Griewank</td>
<td>0.3</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Penalized 1</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Penalized 2</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Michalewicz</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Goldberg</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Sphere Model</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Schwefel 2.22</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>(A)</th>
<th>(\omega)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schwefel 2.21</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Ackley</td>
<td>0.5</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Griewank</td>
<td>0.3</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Penalized 1</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Penalized 2</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Michalewicz</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Goldberg</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Sphere Model</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
<tr>
<td>Schwefel 2.22</td>
<td>0.7</td>
<td>(\frac{\pi}{100})</td>
</tr>
</tbody>
</table>
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 Penalty 1** and **Knapsack Repair 1**.

**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 PENALTY 1** AND **KNAPSACK REPAIR 1**.

<table>
<thead>
<tr>
<th></th>
<th><strong>QEA</strong></th>
<th><strong>SRQEA</strong></th>
<th><strong>QEA</strong></th>
<th><strong>SRQEA</strong></th>
<th><strong>QEA</strong></th>
<th><strong>SRQEA</strong></th>
<th><strong>QEA</strong></th>
<th><strong>SRQEA</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schwefel</strong></td>
<td>3.47e-07</td>
<td>3.36e-07</td>
<td>4.74e-07</td>
<td>2.55</td>
<td>5.74e-07</td>
<td>6.24e-07</td>
<td>7.94e-07</td>
<td>4.88e-07</td>
</tr>
<tr>
<td><strong>Rastrigin</strong></td>
<td>-1.99e+00</td>
<td>2.27e+00</td>
<td>1.46e+00</td>
<td>1.07e+00</td>
<td>6.83e+00</td>
<td>4.49e+00</td>
<td>5.44e+00</td>
<td>1.96e+00</td>
</tr>
<tr>
<td><strong>Ackley</strong></td>
<td>-1.72e+00</td>
<td>0.06e+00</td>
<td>-1.69e+00</td>
<td>0.05e+00</td>
<td>-1.76e+00</td>
<td>0.12</td>
<td>-1.17e+00</td>
<td>0.03e+00</td>
</tr>
<tr>
<td><strong>Griewank</strong></td>
<td>-3.83e+00</td>
<td>6.51</td>
<td>-19.96e</td>
<td>2.19</td>
<td>-156.94e</td>
<td>16.51</td>
<td>-108.99e</td>
<td>5.33</td>
</tr>
<tr>
<td><strong>Penalized 1</strong></td>
<td>-1.69e+00</td>
<td>2.15e+00</td>
<td>8.85e+00</td>
<td>1.14e+00</td>
<td>-5.94e+00</td>
<td>4.78e+00</td>
<td>-4.47e+00</td>
<td>2.01e+00</td>
</tr>
<tr>
<td><strong>Penalized 2</strong></td>
<td>-5.83e+00</td>
<td>4.37e+00</td>
<td>2.13e+00</td>
<td>1.82e+00</td>
<td>-1.45e+00</td>
<td>1.68e+00</td>
<td>-1.03e+00</td>
<td>7.58e+00</td>
</tr>
<tr>
<td><strong>Michalewicz</strong></td>
<td>22.84</td>
<td>2.55</td>
<td>33.14</td>
<td>2.08</td>
<td>36.07</td>
<td>4.28</td>
<td>52.51</td>
<td>2.65</td>
</tr>
<tr>
<td><strong>Goldberg</strong></td>
<td>4.01e+00</td>
<td>2.59</td>
<td>51.71</td>
<td>2.20</td>
<td>76.76</td>
<td>4.69</td>
<td>94.58</td>
<td>2.71</td>
</tr>
<tr>
<td><strong>Sphere Model</strong></td>
<td>-4.32e+00</td>
<td>5.46e+00</td>
<td>-2.27e+00</td>
<td>2.93e+00</td>
<td>-1.65e+00</td>
<td>1.83e+00</td>
<td>-1.20e+00</td>
<td>5.85e+00</td>
</tr>
<tr>
<td><strong>Schwefel 2.21</strong></td>
<td>1.76e+00</td>
<td>5.86</td>
<td>-168.17</td>
<td>5.18</td>
<td>-190.94</td>
<td>22.86</td>
<td>-189.84</td>
<td>5.33</td>
</tr>
<tr>
<td><strong>Dejong</strong></td>
<td>-2.47e+00</td>
<td>6.02e+00</td>
<td>-8.02e+00</td>
<td>1.75e+00</td>
<td>-3.18e+00</td>
<td>4.71e+00</td>
<td>-1.68e+00</td>
<td>1.75e+00</td>
</tr>
<tr>
<td><strong>Rosenbrock</strong></td>
<td>-1.06e+00</td>
<td>3.34e+00</td>
<td>-3.71e+00</td>
<td>6.00e+00</td>
<td>-5.06e+00</td>
<td>8.39e+00</td>
<td>-2.77e+00</td>
<td>2.46e+00</td>
</tr>
<tr>
<td><strong>Kennedy</strong></td>
<td>-1.22e+00</td>
<td>1.07</td>
<td>8.96e+00</td>
<td>8.48e+00</td>
<td>-20.99</td>
<td>5.42</td>
<td>-9.07e+00</td>
<td>0.14</td>
</tr>
<tr>
<td><strong>KR1</strong></td>
<td>415.29</td>
<td>6.26</td>
<td>428.69</td>
<td>3.29</td>
<td>893.58</td>
<td>18.54</td>
<td>960.13</td>
<td>11.96</td>
</tr>
<tr>
<td><strong>KR2</strong></td>
<td>420.88</td>
<td>5.22</td>
<td>423.27</td>
<td>0.95</td>
<td>101.23</td>
<td>13.60</td>
<td>1036.52</td>
<td>3.89</td>
</tr>
<tr>
<td><strong>KP1</strong></td>
<td>579.22</td>
<td>8.06</td>
<td>591.16</td>
<td>3.47</td>
<td>1217.66</td>
<td>55.27</td>
<td>1357.41</td>
<td>12.79</td>
</tr>
<tr>
<td><strong>KP2</strong></td>
<td>399.64</td>
<td>5.86</td>
<td>411.3</td>
<td>3.03</td>
<td>923.82</td>
<td>21.42</td>
<td>1001.99</td>
<td>8.86</td>
</tr>
<tr>
<td><strong>TRAP</strong></td>
<td>72.17</td>
<td>4.19</td>
<td>82.37</td>
<td>1.81</td>
<td>154.8</td>
<td>8.42</td>
<td>186.87</td>
<td>5.29</td>
</tr>
</tbody>
</table>

**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 each of the experiments is set to 20 ($S = 20$), 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 $\omega$ and $\Delta$ 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.

1. **Trap problem**

Trap problem is defined as:

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

Where $N$ is the number of traps and $\text{Trap}$.
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 $f(x)$:

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

This paper considered:

$$w_i = R(1,v), \quad p_i = R(1,v)$$

Where $R(\cdot,\cdot)$ 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],
- 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**


