

Improvement of Quantum Evolutionary Algorithm with a Functional Sized Population

Tayarani Mohammad and Akbarzadeh Toutounchi Mohammad Reza*

Abstract. This paper proposes a dynamic structured interaction among members of population in a Quantum Evolutionary Algorithms (QEA). The structured population is allowed to expand/collapse based on a functional population size and partial re-initialization of new members in the population. Several structures are compared here and the study shows that the best structure for QEA is the cellular structure which 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.

1 Introduction

Recently we proposed a ring structure sinusoid sized population for QEA (SRQEA) [1]. In this paper several structures and functions for the population of QEA are investigated to find the best population structure and function for the size of the population.. Size of the population is an effective parameter of the evolutionary algorithms and has a great role on the performance of EAs. Several researches investigate the effect of population size and try to improve the performance of EAs by controlling of the size of the population. A functional sized population GA with a periodic function of saw-tooth function is proposed in [2]. Reference [3] finds the best population size for genetic algorithms. Inspired by the natural features of the variable size of the population, [4] presents an improved genetic algorithm with variable population-size. In [5] an adaptive size for the population is proposed for a novel evolutionary algorithm. Reference [6] proposes a scheme to adjust the population size to provide a balance between exploration and exploitation. To preserve the diversity in the population in QEA, [7] proposes a novel diversity preserving operator for QEA.

Tayarani Mohammad
Azad University of Mashhad, Iran
e-mail: tayarani@ieee.org

Akbarzadeh Toutounchi Mohammad Reza
Ferdowsi University of Mashhad,
Departments of electrical engineering and computer engineering, Iran
e-mail: akbarzadeh@ieee.org

* Akbarzadeh Toutounchi Mohammad Reza is also currently with the departments of electrical engineering and computer engineering at Ferdowsi University of Mashhad, Iran.

This paper compares several structures for QEA to find the best structure for this algorithm and applies a novel operator that is functional sized population on this structure. Several functions for the population are proposed here to find the best function for the population. This paper is organized as follows: Section 2 finds the best structure for QEA, Section 3 proposes the functional sized population, Section 4 finds the best parameters for the proposed algorithm, Section 5 experiments the proposed algorithm on 14 numerical functions and finally Section 6 concludes the proposed operator.

2 Best Structure for QEA

Structure of evolutionary algorithms has a grate role on the performance and evolution process of the algorithms. This section tries to find the best structure for QEA. After finding the best structure for QEA, FSQEA is proposed that uses the best structure which is found in this section. The structures which are examined in this paper are shown in fig 1. The examined structures are ring, cellular, Btree, cluster, Grid, $K_{m,m}$, Ladder, Crossed ladder, Star, Random_{*h*} and the structure which is proposed in [9]. In star structure all the q-individuals are connected to each other and in Random_{*h*} in each iteration of algorithm each q-individual is connected to *h* q-individual randomly. To find the best structure for several examinations are performed on several problems. The problems which are used are Knapsack problem, Trap problem and 14 numerical benchmark functions. The detail of experiments is not analyzed here and only the overall results are discussed. For the dimension of 100, for the Knapsack problem and Trap problem is the cellular structure and the best structure for the numerical functions is the Random structure. For the numerical functions the best structure is Random₂ and Random₄ with 4 best results, after these structures, Random₆ places in third place with 3 best results. To find the best structure among these two structure (Cellular and Random₂) and the original structure of QEA which is proposed in [9] the experiments are performed for several dimensions. Other structures are not examined in this step because of the limitation of time and computational resources. Table 1 shows the experimental results on 19 benchmark functions for six dimensions. For the dimension of $m=25$, in 4 objective functions the best

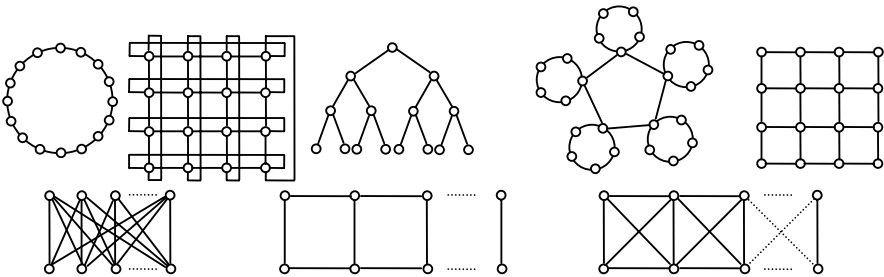


Fig. 1 The compared structures. The structures from top, left to bottom right are Ring, Cellular, Btree, Cluster, Grid, $K_{m,m}$, Ladder and Crossed-ladder

Table 1 The best Structure for QEA. The bold results are the best Ones. All the experiments are performed over 30 runs

	<i>m</i> =25	<i>m</i> =50	<i>m</i> =100	<i>m</i> =150	<i>m</i> =250	<i>m</i> =500	
Reference[9]	4	3	4	6	3	2	%0.19
Cellular	10	14	15	10	13	14	%0.67
Random ₂	5	2	0	3	3	3	%0.14

structure is the original structure of QEA, the cellular structure reaches the best results for 10 objective functions and Random₂ structure is the best structure for 5 objective functions. In %67 of the experiments the best structure for QEA is the Cellular structure, in %19 of experiments the best structure is the original structure of QEA and in %14 of experiments the best structure is Random₂. According to these experiments the best structure for QEA is the Cellular structure, so we use the Cellular structure for QEA in the reminder of this paper.

3 Functional Sized Population QEA (FSQEA)

Another approach to maintain the diversity of the population and improve the performance of the evolutionary algorithms is using a variable size for the population. In [1] a variable size population is proposed for QEA that improves the performance of QEA. They use a sinusoid function for the size of the population with partially reinitialization of the *q*-individuals. Here to improve the performance of QEA, and investigate the best structure and best function for the population of QEA this paper uses a functional population size for QEA. In addition to the sinusoid function, this paper uses some other functions for QEA; the functions are saw-tooth [2], inverse saw-tooth, triangular, sinusoid [1] and square functions. Fig. 1 shows the functions which are examined in this paper. The pseudo code of the proposed Functional Size QEA (FSQEA) is described as below:

Procedure FSQEA

begin

$t=0$

1. initialize quantum population $Q(0)$ with the size of $n(0) = \bar{n}$
2. make $X(0)$ by observing the states of $Q(0)$.
3. evaluate $X(0)$.
4. **for** all binary solutions x_i^0 in $X(t)$ **do**
 - begin**
 - 5. find neighborhood set N_i in $X(0)$.
 - 6. find binary solution x with best fitness in N_i
 - 7. save x in B_i
 - end**
8. **while not** termination condition **do**
 - begin**
 - $t=t+1$
9. $n(t) = f(t)$

```

10.  if  $n(t) > n(t-1)$  create random q-individuals
11.  if  $n(t) < n(t-1)$  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  $x'_i$  in  $X(t)$  do
      begin
16.    find neighborhood set  $N_i$  in  $X(t)$ .
17.    select binary solution  $x$  with best fitness in  $N_i$ 
18.    if  $x$  is fitter than  $B_i$  save  $x$  in  $B_i$ 
      end
end

```

The pseudo code of FSQEA is described as below:

1. In the initialization step, the quantum-individuals q_i^0 are located in a structured population. Then $[\alpha_i^0 \beta_i^0]^T$ of all q_i^0 are initialized with $1/\sqrt{2}$, where $i=1,2,\dots,n$ is the location of the q-individuals in the population, $k=1,2,\dots,m$, and m is the number of qubits in the 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 $X(0)=\{x_i^0 | i=1,2,\dots,n\}$ at generation $t=0$ by observing $Q(0)=\{q_i^0 | i=1,2,\dots,n\}$ states, where $X(t)$ at generation t is a random instant of qubit population and n is the size of population. Each binary instant, x_i^0 of length m , is formed by selecting each bit using the probability of qubit, either $|\alpha_{i,k}^0|^2$ or $|\beta_{i,k}^0|^2$ of q_i^0 . Observing the binary bit $x'_{i,k}$ from qubit $[\alpha_{i,k}^t \beta_{i,k}^t]^T$ performs as:

$$x'_{i,k} = \begin{cases} 0 & \text{if } R(0,1) < |\alpha_{i,k}^t|^2 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

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

3. Each binary instant x_i^0 is evaluated to give some measure of its objective. 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_i^0 in $X(0)$ are found and the best solution among N_i is stored in B_i . In the structured proposed algorithm each individual is the neighbor of itself that is x_i belongs to neighborhood set N_i . B_i is the best possible solution, which the q-individual q_i^t has 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 function of the iteration number. In this step, $n(t)$, the size of the population in iteration t , is calculated as a function. The functions that used in this paper are:

$$\text{Saw-tooth [3]: } n(t) = \text{Round} \left[\bar{n} - \frac{2A}{T-1} \left(t - T \times \text{Round} \left(\frac{t-1}{T} - 1 \right) \right) \right]$$

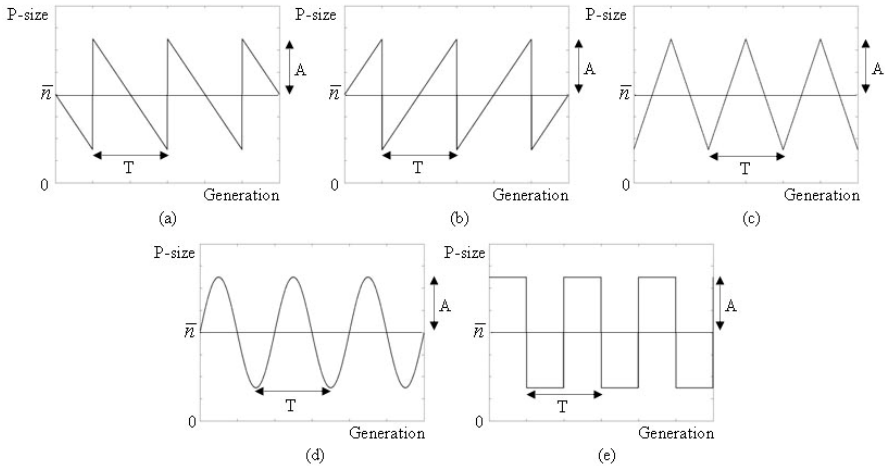


Fig. 2 a) The functions which are used for the population size. a) saw-tooth b) inverse saw-tooth c) triangular d) sinusoid e) square. T is the period of the functions, A is the amplitude and P -size is the size of the population in generation t

$$\text{Inverse Saw tooth: } n(t) = \text{Round} \left[\bar{n} + \frac{2A}{T-1} \left(t - T \times \text{Round} \frac{t-1}{T} - 1 \right) \right]$$

$$\text{Triangular: } n(t) = \text{Round} \left[\bar{n} - A + 2A \max \left(\min \left(\frac{2 \text{mod}(t, T)}{T}, 2 - \frac{2 \text{mod}(t, T)}{T} \right), 0 \right) \right]$$

$$\text{Sinusoid: } n(t) = \text{Round} \left[\bar{n} + A \sin \left(\frac{2\pi}{T} t \right) \right]$$

$$\text{Square: } n(t) = \text{Round} \left[\bar{n} + A - 2A \times \text{Round} \left(\frac{\text{mod}(t, T)}{T} \right) \right]$$

Where $n(t)$ is the size of the population in generation t , \bar{n} is the average size of the population, A is the amplitude of the periodic function of population size, T is the period of the functional population, $\text{Round}(\cdot)$ is the round function (rounds its input to nearest integer), and $\text{mod}(\cdot, \cdot)$ is modulus after division function. Fig 3 shows the functions which are used in this paper. The best values for T and A are found in the following of this section.

10. If $n(t)$, the size of the population in iteration t is greater than $n(t-1)$, 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 $n(t)$.

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

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

Table 2 The best parameters for the proposed FSQEA. The best results are the best ones

	Saw-Tooth			Inverse-Saw			Sinusoid			Square			Triangular			QEA	
	A	T	Best	A	T	Best	A	T	Best	A	T	Best	A	T	Best	A	Best
KR1	0.4	25	406.7	0.2	100	407.51	0.4	100	407.23	0.2	250	407.67	0.1	50	407.18	387.74	
KR2	0.9	100	412.62	0.4	100	412.8	0.2	50	413.05	0.4	500	412.59	0.4	100	412.72	407.43	
KP1	0.2	50	556.69	0.1	50	556.69	0.2	50	556.69	0.2	25	556.69	0.2	50	556.69	517.66	
KP2	0.4	50	406.44	0.4	25	407.56	0.2	25	407.19	0.4	250	407.49	0.1	25	405.35	388.88	
Trap	0.2	250	82.6	0.4	500	83.7	0.2	500	83.7	0.2	500	84.4	0.1	500	84.2	79.737	
f_1	0.2	100	44932	0.6	100	47227	0.9	100	47678	0.4	100	45464	0.2	100	45973	32471	
f_2	0.2	100	-1420	0.4	100	-1274	0.2	50	-1398	0.2	100	-1419	0.2	250	-1374	-2281	
f_3	0.2	250	-17.07	0.4	100	-16.88	0.2	50	-16.98	0.2	25	-17.00	0.4	100	-17.00	-17.24	
f_4	0.2	500	-22.85	0.2	250	-17.44	0.4	100	-21.22	0.2	250	-21.33	0.4	100	-20.74	-47.744	
f_5	0.2	100	-1.0e5	0.4	500	-78259	0.4	500	-94385	0.4	250	-3.8e4	0.4	100	-9.0e4	-2.05e5	
f_6	0.2	100	-22786	0.4	250	-18903	0.4	250	-21103	0.2	100	-2.2e4	0.4	250	-2.1e4	-49138	
f_7	0.2	250	32.38	0.4	250	35.40	0.4	100	32.36	0.4	500	32.45	0.2	50	33.33	19.33	
f_8	0.2	250	50.17	0.4	250	53.64	0.4	250	52.28	0.2	100	50.67	0.2	100	51.36	37.49	
f_9	0.2	500	-2.5e5	0.4	250	-1.94e5	0.2	100	-2.27e5	0.2	100	-2.3e5	0.2	100	-2.3e5	-5.69e5	
f_{10}	0.2	250	-3.55	0.4	500	-2.99	0.2	100	-3.33	0.2	250	-3.38	0.4	100	-3.3591	-5.5741	
f_{11}	0.2	25	-162.19	0.2	500	-158.75	0.2	100	-161.56	0.2	250	-159	0.2	500	-163.13	-143.63	
f_{12}	0.2	250	-7.1e6	0.4	500	-6.22e6	0.4	500	-7.8e6	0.2	100	-7.7e6	0.4	500	-7.1e6	-2.54e7	
f_{13}	0.2	100	-39280	0.4	250	-31939	0.2	500	-37418	0.2	500	-36826	0.4	100	-34912	-1.10e5	
f_{14}	0.6	25	-0.0057	0.6	25	-0.004	0.6	25	-0.009	0.9	50	-0.001	0.9	25	-0.0058	-1.13	

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 $x^i_i (i=1,2,\dots,S)$ in the population.
16. Finding the neighbors of the binary solution located on the location i .
17. Find the best possible solution in the neighborhood N_i , and store it to x .
18. If x is fitter than B_i , store x to B_i .

The proposed functions for the population have two cycles. One cycle is increasing the size of population. In the increasing cycle, the new quantum individuals are created and inserted in the population. 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.

4 Finding the Best Parameters

As it seen in Fig. 1, the proposed functions have some parameters that are A , the amplitude and T the period of the functions. In order to find the best values for these parameters some experiments are performed. Fig. 1 shows the finding of the best parameters for the proposed FSQEA for Knapsack Problem penalty type 1 and Generalized Schwefel Function 2.26. The best parameters for the Knapsack problem, Trap Problem and 14 numerical benchmark functions are found similar

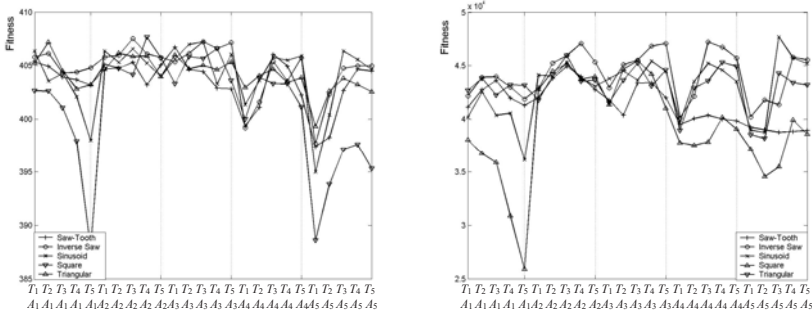


Fig. 3 Parameter setting of FSQEA for T and A for (a) Knapsack Problem Penalty 1 (b) Generalized Schwefel Function 2.26 for several functions for the population. The parameters are set to $T_1 \dots T_5=(25,100,250,500,1000)$ and $A_1 \dots A_5=\bar{n} \times (0.1,0.2,0.4,0.6,0.9)$

to the Fig 2. The best parameters and the best functions for the size of the population are summarized in Table 2. According to Table 2 the Inverse Saw-Tooth function has the best results for 11 benchmark objective functions, the Square for 3 benchmark functions, sinusoid for 2 functions, saw-tooth function for 1 benchmark function, and Triangular with no objective function, so the best function for the size of the population is Inverse Saw-Tooth function. Only for one objective function the best result is reached by original version of QEA and the proposed algorithm improves the performance of QEA for most of the objective functions. In order to find the best parameters for the proposed algorithm Table 3 shows the median and standard deviation of the best parameters for 5 proposed functions. According to this table the best amplitude for the proposed functions in 0.2 and best period T is 100.

Table 3 Median and Standard deviation of the best parameters for the proposed FSQEA

	A		T	
	Mean	Std	Mean	Std
Saw-Tooth	0.2	0.184	100	144
Inverse-Saw	0.4	0.124	250	177
Sinusoid	0.2	0.18	100	179
Square	0.2	0.17	250	168
Triangular	0.2	0.18	100	161

5 Experimental Results

The proposed FSQEA is compared with the original version of QEA and DPCQEA[8]. In order to compare the algorithms with their best parameters, the best parameters for each algorithm are found independently. The experimental results are performed for several dimensions ($m=25, 50, 100, 250, 500$) of Knapsack Problem, Trap Problem and 14 numerical benchmark functions. The

Table 4 Experimental results on 14 numerical benchmark functions for $m=100$ and $m=250$. the results are averaged over 30 runs. Ttest shows Ttest between the results of each algorithm with FSQEA. The best results are the best ones

$m=100$								
	FSQEA		PDCQEA			QEA		
	Mean	Std	Mean	Std	Ttest	Mean	Std	Ttest
KR1	429.79	3.66	396.62	12.27	1.76e-7	417.07	6.62	4.74e-5
KR2	414.48	0.48	434.08	3.07	1.06e-13	412.51	1.64	0.002
KP1	562.13	0.92	519.54	3.19	0	546.82	11.29	0.0004
KP2	418.15	1.94	398.61	1.80	6.88e-15	406.91	4.39	7.31e-7
Trap	81.1	0.99	82	1.05	0.06	72.2	3.55	4.79e-7
f_1	45889	1558.8	38937	2687.6	1.34e-6	34437	3984.1	1.08e-7
f_2	-1287.8	97.97	-1684.8	183.4	1.04e-5	-2096.3	199.45	9.89e-10
f_3	-16.89	0.14	-16.92	0.12	0.65	-17.19	0.09	2.91e-5
f_4	-30.08	5.24	-32.48	3.15	0.23	-39.60	8.48	0.007
f_5	-1.43e5	14756	-1.46e5	15450	0.62	-1.67e5	16807	0.004
f_6	-23017	2742.1	-32625	4038.4	7.15e-6	-36949	4918.7	3.36e-7
f_7	31.17	2.53	22.69	1.40	2.93e-8	22.04	2.46	1.84e-7
f_8	50.57	1.75	46.38	3.76	0.005	38.19	3.24	3.58e-9
f_9	-2.46e5	23972	-3.62e5	35152	8.79e-8	-4.55e5	68039	3.39e-8
f_{10}	-4.2161	0.29	-4.34	0.94	0.67	-5.21	0.68	0.0005
f_{11}	-169.84	7.76	-172.19	4.15	0.41	-176.72	3.78	0.02
f_{12}	-1.33e7	2.53e6	-1.72e7	2.20e6	0.002	-2.56e7	9.20e6	0.0007
f_{13}	-49229	5885	-69515	7918.2	4.09e-6	-1.08e5	35412	5.60e-5
f_{14}	-0.098	0.065	-0.0005	0.0005	0.0002	-1.29	1.26	0.008
$M=250$								
	FSQEA		PDCQEA			QEA		
	Mean	Std	Mean	Std	Ttest	Mean	Std	Ttest
KR1	985.45	15.85	1003.8	6.55	0.003	919.75	16.89	4.63e-8
KR2	1010.1	5.03	1045.1	8.73	2.02e-9	975.36	18.00	1.47e-5
KP1	1252.6	17.40	1361.2	24.44	1.07e-9	1173	30.19	1.02e-6
KP2	994.46	10.31	183.1	4.04	8.21e-10	942.99	20.90	1.60e-6
Trap	198.9	1.45	183	4.04	8.2e-10	157.4	9.54	6.64e-11
f_1	76292	2939	66920	4602	3.71e-5	55844	5845.3	1.07e-8
f_2	-5118.9	304.75	-6360	264.68	1.36e-8	-6511	360.74	2.59e-8
f_3	-17.39	0.11	-17.21	0.02	7.61e-5	-17.62	0.11	0.0002
f_4	-134.53	10.16	-132.79	6.83	0.65	-154.42	14.04	0.002
f_5	-5.60e5	20743	-5.54e5	25772	0.58	-6.05e5	43232	0.007
f_6	-1.09e5	5039	-1.26e5	3670	4.92e-8	-1.44e5	8269.3	1.05e-9
f_7	52.24	4.39	38.38	1.85	3.25e-8	39.43	3.02	5.18e-7
f_8	93.51	5.31	94.27	5.95	0.76	73.50	5.02	7.91e-8
f_9	-1.26e6	93119	-1.42e6	75573	0.0005	-1.64e6	1.34e5	8.91e-7
f_{10}	-5.95	0.30	-5.97	0.36	0.91	-6.52	0.46	0.004
f_{11}	-189.69	2.35	-190.31	1.77	0.51	-192.03	1.36	0.014
f_{12}	-2.44e8	2.45e7	-2.64e8	2.36e7	0.08	-3.09e8	3.59e7	0.0002
f_{13}	-3.1e5	35376	-4.15e5	31888	2.19e-6	-4.81e5	72311	3.21e-6
f_{14}	-7.53	1.94	-0.11	0.13	4.77e-10	-20.38	6.19	6.6e-6

average population size of all algorithms for all of the experiments is set to 25; termination condition is set for a maximum of 1000 generations and the structure of population is considered as cellular[8]. Due to statistical nature of the optimization algorithms, all results are averaged over 30 runs and Ttest analysis is performed on results. The parameter of QEA is set to $\Delta\theta=0.01\pi$ (reference [9])

shows this is the best parameter for QEA), the parameters of FSQEA are set to the best parameters found in previous section and the best parameters of DPCQEA are set to the best parameters proposed in [9].

Table.4 summarizes the experimental results on QEA, DPCQEA and FSQEA for Knapsack Problem, Trap Problem and 14 benchmark functions (The results for some dimensions are not summarized in Table.4 because of small space of the paper). As it seen in Table 4, FSQEA has the best results. The Ttest between the results of FSQEA and QEA is too small (in average about 10^{-5}) and it means that the results of Table 4 has high amount of validity.

6 Conclusion

This paper proposes a Functional Sized population QEA with a cellular structure. Before proposing the functional sized QEA, the best structure for QEA is found that is the cellular structure. After finding the best structure for QEA, the FSQEA is proposed with the Cellular structure. The proposed FSQEA has several parameters that are investigated in this paper. Finally the experimental results are performed on the proposed algorithm and the improvement that is shown and the small amount of Ttest shows that the proposed algorithm reaches the best results with high validity. The time complexity of the proposed FSQEA is equal to original version of QEA because the average size of the population for FSQEA is equal to QEA and the number of function evaluations for both of algorithms is equal.

The objective functions which are used here are f_1 :Schwefel 2.26 [6], f_2 :Rastrigin [6], f_3 :Ackley [6], f_4 :Griewank [6], f_5 :Penalized 1 [6], f_6 :Penalized 2 [6], f_7 :Michalewicz [7], f_8 :Goldberg [2], f_9 :Sphere Model [6], f_{10} :Schwefel 2.22 [6], f_{11} :Schwefel 2.21 [6], f_{12} :Dejong [7], f_{13} :Rosenbrock [2], and f_{14} :Kennedy [2].

References

1. Tayarani-N, M.-H., Akbarzadeh-T, M.R.: A Sinusoid Size Ring Structure Quantum Evolutionary Algorithm. In: IEEE International Conference on Cybernetics and Intelligent Systems Robotics, Automation and Mechanics (2008)
2. Koumousis, V.K., Katsaras, C.P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. IEEE Trans. Evol. Comput. 10, 19–28 (2006)
3. Wang, D.L.: A study on the optimal population size of genetic algorithm. In: Proceedings of the 4th World Congress on Intelligent Control and Automation (2002)
4. 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 PSOGA based hybrid evolutionary algorithm. In: International Conference on Machine Learning and Cybernetics (2003)
5. Li-Shan, Q.J.K.: A novel dynamic population based evolutionary algorithm for revised multimodal function optimization problem. In: Fifth World Congress on Intelligent Control and Automation (2004)

6. Zhong, W., Liu, J., Xue, M., Jiao, L.: A Multi-agent Genetic Algorithm for Global Numerical Optimization. *IEEE Trans. Sys, Man and Cyber.* 34, 1128–1141 (2004)
7. Khorsand, A.-R., Akbarzadeh-T, M.-R.: Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm. In: *IEEE International Conference on Systems, Man and Cybernetics* (2005)
8. Tayarani-H, M.-H., Akbarzadeh-T, M.-R.: A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms. In: *IEEE World Conference on Computational Intelligence* (2008)
9. Han, K., Kim, J.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. on Evolutionary Computation* 6(6) (2002)

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.

Trap problem is defined as:

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

Where N is the number of traps and

$$\text{Trap}(x) = \begin{cases} 4 - \text{ones}(x), & \text{if } \text{ones}(x) \leq 4 \\ 5 & \text{if } \text{ones}(x) = 5 \end{cases} \quad (3)$$

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 $(1,1,1,1,1)$.

Knapsack problem is a well-known combinatorial optimization problem which is in class of NP-hard problems [7]. Knapsack problem can be described as selecting various items x_i ($i=1,2,\dots,m$) 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:

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

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

The use of QEA for solving Knapsack problem is described in [7].