INT J COMPUT COMMUN, ISSN 1841-9836
9(2):227-242, April, 2014.

Dynamic Behavior Analysis of Membrane-Inspired Evolutionary Algorithms

G. Zhang, J. Cheng, M. Gheorghe

Gexiang Zhang*, Jixiang Cheng
School of Electrical Engineering,
Southwest Jiaotong University
Chengdu, 610031, P.R. China
*Corresponding author: zhgxdylan@126.com

Marian Gheorghe
Department of Computer Science,
The University of Sheffield,
Regent Court, Portobello Street, Sheffield, S1 4DP, UK

Abstract: A membrane-inspired evolutionary algorithm (MIEA) is a successful instance of a model linking membrane computing and evolutionary algorithms. This paper proposes the analysis of dynamic behaviors of MIEAs by introducing a set of population diversity and convergence measures. This is the first attempt to obtain additional insights into the search capabilities of MIEAs. The analysis is performed on the MIEA, QEPS (a quantum-inspired evolutionary algorithm based on membrane computing), and its counterpart algorithm, QIEA (a quantum-inspired evolutionary algorithm), using a comparative approach in an experimental context to better understand their characteristics and performances. Also the relationship between these measures and fitness is analyzed by presenting a tendency correlation coefficient to evaluate the importance of various population and convergence measures, which is beneficial to further improvements of MIEAs. Results show that QEPS can achieve better balance between convergence and diversity than QIEA, which indicates QEPS has a stronger capacity of balancing exploration and exploitation than QIEA in order to prevent premature convergence that might occur. Experiments utilizing knapsack problems support the above made statement.

Keywords: Membrane computing, membrane-inspired evolutionary algorithm, dynamic behavior, quantum-inspired evolutionary algorithm; knapsack problem.

1 Introduction

Membrane computing, initiated by Păun in 1998 [1], focuses on the investigation of the models, called membrane systems or P systems, abstracted from the structure and the functioning of the living cell as well as from the cooperation of cells in tissues, organs, and other populations of cells. Thompson Institute for Scientific Information, ISI, listed the seminal paper as a fast breaking record and this area as an emerging research front in computer science in 2003, and thereby membrane computing becomes a branch of natural computing and has developed very fast into a vigorous scientific discipline [2–6].

Aiming at investigating the interactions between membrane computing and evolutionary computation, membrane-inspired evolutionary algorithms (MIEAs) are considered as a class of hybrid optimization algorithms, which use the concepts and principles of meta-heuristic search methodologies and the hierarchical or network structures of P systems, and to some extent, some of the rules of P systems [7,8]. A MIEA is regarded as a successful paradigm extending P system models with capabilities that make them amenable to real-world applications [9]. Due to a very wide range of applications of meta-heuristic search methodologies, such as genetic algorithms and tabu search, there is a very promising perspective of applying P systems to solve various complex and difficult engineering problems.

In recent years, many investigations referred to MIEAs. The first version of membrane algorithms was designed with a nested membrane structure (NMS) and a local search heuristic for solving travelling salesman problems, which are well-known NP-hard optimization problems [10]. An approach combining NMS and genetic algorithms was presented and tested by using six benchmark functions [11]. In the preceding work, we proposed a MIEA, called a quantum-inspired evolutionary algorithm based on P systems (QEPS), which incorporates a one-level membrane structure (OLMS) and a quantum-inspired evolutionary algorithm (QIEA) [12]. A well-known NP-complete optimization problem, knapsack problem, was used to carry out extensive experiments, which show that QEPS achieves better solutions than its counterpart QIEA and OLMS has an advantage over NMS. In [7, 13–15], QEPS and its modified versions were presented to solve various problems, such as radar emitter signal analysis and image processing. In [16] and [17], DNA sequences design was optimized by designing a MIEA based on crossover and mutation rules and a dynamic MIEA combining the fusion and division rules of P systems with active membranes and search strategies of differential evolution (DE) and particle swarm optimization (PSO), respectively. In [18], a memory mechanism was considered in the design of MIEAs. In [19], a hybrid MIEA was presented by combining OLMS with PSO to solve constrained optimization problems. In [8], a MIEA was proposed by using the network membrane structure of a tissue P system with five cells to organize five representative DE variants.

However, since MIEAs were initiated in 2004, a question has been asked many times by researchers from the areas of membrane computing and evolutionary computation. The question refers to the role played by P systems in MIEAs, that is, what advantages do P systems bring to MIEAs? This is also a critical and tough question that has been haunting many researchers in the field of membrane computing. So the motivation of this work is to try to some extent to find an appropriate answer for this question.

In this study, we propose the analysis of the dynamic characteristics of MIEAs by using a set of population diversity and convergence measures to comparatively investigate the evolving processes of QEPS and its counterpart algorithm, QIEA. Due to the difficulty in theoretically reasoning about MIEAs, we mainly focus on the experimental analysis. Also we present a tendency correlation coefficient to analyze the relationship between the population diversity and convergence measures and fitness to understand the importance of each measure and to provide suggestions on how to improve the performance of MIEAs with respect to population diversity and convergence. Furthermore, experiments conducted on knapsack problems are presented.

The rest of this study is organized as follows. Section 2 gives a brief description of QIEA and QEPS, which is helpful to understand the dynamic behavior analysis of MIEAs expounded in Section 3. Specific examples follow in Section 4 to verify the analysis presented in the preceding section. Section 5 concludes this work.

2 QIEA and QEPS

2.1 QIEA

Inspired by quantum computing, Han and Kim [20] proposed a novel evolutionary algorithm, called QIEA, for a classical computer. QIEA consists of three main components: quantum-inspired bit (Q-bit) representation, a probabilistic observation and a quantum-inspired gate (Q-gate) [21]. In QIEA, a genotypic gene is represented by using a Q-bit defined by a pair of numbers $(\alpha,\ \beta)$ denoted as $[\alpha\ \beta]^T$, where $|\alpha|^2$ and $|\beta|^2$ are probabilities that the observation of the Q-bit will render a '0' or '1' state. A string of Q-bits is applied to represent a Q-bit individual. The connection between genotypic representation (Q-bit representation) and phenotypic individuals (binary solutions) is established by the probabilistic observation. The Q-gate is used to produce offspring. Generally speaking, QIEA is composed of the following steps:

(i) Initialization: a population $Q(t)$ with n Q-bit individuals is generated, $Q(t)=\{\boldsymbol{q}_1^t, \boldsymbol{q}_2^t, \cdots, \boldsymbol{q}_n^t\}$ at generation t (here $t = 0$), where $\boldsymbol{q}_i^t$ $(i = 1, 2, \cdots, n)$ is an arbitrary individual in $Q(t)$, which is represented as

$$\boldsymbol{q}_i^t = \begin{bmatrix} \alpha_{i1}^t|\alpha_{i2}^t|\cdots|\alpha_{il}^t \\ \beta_{i1}^t|\beta_{i2}^t|\cdots|\beta_{il}^t \end{bmatrix}, \tag{1}$$

where l is the number of Q-bits, i.e., the string length of the Q-bit individual.

(ii) Observation: a probabilistic observation is used to produce binary solutions $P(t)$, $P(t)=\{\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \cdots, \boldsymbol{x}_n^t\}$, by observing the states of $Q(t)$, to be specific, a binary bit 0 or 1 is obtained in terms of the probability, either $|\alpha_{ij}^t|^2$ or $|\beta_{ij}^t|^2$ of $\boldsymbol{q}_i^t$, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, l$. Thus a Q-bit individual with l Q-bits results in a binary solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, n)$ with l binary bits.

(iii) Evaluation: the binary solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, n)$ in $P(t)$ is evaluated thus obtaining its fitness. Additionally the best solution among $P(t)$ is stored.

(iv) Offspring generation: Q-gates are performed on Q-bit individuals in $Q(t)$ to produce their corresponding individuals at the next generation. For example, the j-th Q-bit in the i-th Q-bit individual $\boldsymbol{q}_i^t$, $j = 1, 2, \cdots, l$, $i = 1, 2, \cdots, n$ is updated by applying the current Q-gate $\boldsymbol{G}_{ij}^t(\theta)$. QIEA uses a quantum rotation gate as a Q-gate; this is given by

$$\boldsymbol{G}_{ij}^t(\theta) = \begin{bmatrix} \cos\theta_{ij}^t & -\sin\theta_{ij}^t \\ \sin\theta_{ij}^t & \cos\theta_{ij}^t \end{bmatrix}, \tag{2}$$

where θ_{ij}^t is an adjustable Q-gate rotation angle.

(v) Termination condition: the maximal number of evolutionary generations or the maximal number of function evaluations could be utilized to stop the algorithm. If the termination condition is satisfied, the algorithm will stop and output the final results, otherwise, the generation number increases by 1, i.e., $t = t + 1$, and the algorithm goes back to Step (ii).

2.2 QEPS

In the process of investigating the interactions between P systems and evolutionary algorithms, we presented a MIEA, QEPS [12], which was designed with the hierarchical framework of a cell-like P system, the objects consisting of Q-bits and classical bits, the rules made up of Q-gate evolutionary rules in QIEA and evolution rules in P systems. QEPS uses OLMS, where the skin membrane contains m elementary membranes defining m regions. Q-bits, organized as a Q-bit individual in a proper way, are treated as multisets of objects. Classical bits, obtained from their corresponding Q-bits by using a probabilistic observation, are arranged as a binary string and are dealt with also as multisets of objects. In QEPS, a binary string corresponds to a solution of a problem. The set of rules are responsible for evolving the system and selecting the best fit Q-bit individuals. All the objects and rules are appropriately placed in the membrane structure.

More precisely the P system-like framework consists of

(i) a membrane structure $[[\,]_1, [\,]_2, \cdots, [\,]_m]_0$ with $m{+}1$ regions contained in the skin membrane, denoted by 0;

(ii) an alphabet that consists of all possible Q-bits and classical bits;

(iii) a set of terminal symbols, $T = \{0, 1\}$;

(iv) initial multisets $w_0 = \lambda$,

$$w_1 = \boldsymbol{q}_1\boldsymbol{q}_2 \cdots \boldsymbol{q}_{n_1},$$
$$w_2 = \boldsymbol{q}_{n_1+1}\boldsymbol{q}_{n_1+2} \cdots \boldsymbol{q}_{n_2},$$
$$\cdots$$
$$w_m = \boldsymbol{q}_{n_{(m-1)}+1}\boldsymbol{q}_{n_{(m-1)}+2} \cdots \boldsymbol{q}_{n_m},$$

where $\boldsymbol{q}_i$, $1 \leq i \leq n$, is a Q-bit individual; n_j, $1 \leq j \leq m$, is the number of individuals in w_j; $\sum_{j=1}^{m} n_j = n$, where n is the total number of individuals in this computation;

(v) rules which are classified as

- a evolution rules in each of the compartments 1 to m which are transformation-like rules updating a Q-bit individual according to the current Q-gate;
- b observation rules which make binary solutions from Q-bit individuals;
- c communication rules which send the best fit individual binary representation from each of the m elementary membranes into the skin membrane and then the overall best binary representation from the skin membrane to each elementary membrane.

3 Dynamic Behavior Analysis

This section analyzes the dynamic behaviors of MIEAs in the process of evolution from two perspectives, the population diversity and convergence. Six diversity and four convergence measures are introduced to comparatively exhibit the evolutionary behaviors of QEPS and QIEA. We start from population diversity analysis and then turn to convergence analysis. Finally, a tendency correlation coefficient is proposed to evaluate the relationship between diversity and convergence measures and the quality of solutions.

3.1 Population Diversity Analysis

Population diversity is crucial for a population-based search method to prevent premature convergence toward local optima. Diversity measures are used to evaluate the levels and types of varieties of individuals in a population [22]. In this subsection, six diversity measures are considered for QEPS and QIEA, and they are respectively

(1) D_{qbw}: Q-bit distance between the best and worst Q-bit individuals corresponding to the best and worst fitness values in a population, respectively. D_{qbw} is described as

$$D_{qbw} = \frac{1}{m}\sum_{j=1}^{m}\left||a_{bj}|^2 - |a_{wj}|^2\right| \tag{3}$$

where $|a_{bj}|^2$ and $|a_{wj}|^2$ are probabilities of the j-th Q-bit in the best and worst Q-bit individuals, respectively; m is the number of Q-bits in a Q-bit individual. $0 \leq D_{qbw} \leq 1$. A larger value of D_{qbw} gives a hint of larger distance between the best and worst Q-bit individuals.

(2) D_{qa}: average Q-bit distance of all Q-bit individuals in a population. D_{qa} is defined as

$$D_{qa} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left\{ \frac{1}{m} \sum_{k=1}^{m} \left| |\alpha_{ik}|^2 - |\alpha_{jk}|^2 \right| \right\} \tag{4}$$

where $|a_{ik}|^2$ and $|a_{jk}|^2$ are probabilities of the k-th Q-bit in the i-th and j-th Q-bit individuals, respectively; m is the number of Q-bits in a Q-bit individual; n is the number of individuals in a population. D_{qa} is the average value of the Q-bit distance between $n(n-1)$ pairs of Q-bit individuals. $0 \leq D_{qa} \leq 1$. A larger value of D_{qa} suggests a larger distance between each pair of Q-bit individuals in a population.

The two diversity measures above are obtained in Q-bit space, so they can be regarded as genotypic diversity measures for QEPS and QIEA. In what follows we will introduce four phenotypic diversity measures: Hamming distance between the best and worst binary individuals (D_{hbw}) in a population, mean Hamming distance of all binary individuals (D_{hm}) in a population, and two diversity measures based on dispersion statistical measures including the diversity between chromosomes (D_{bc}) and the diversity between the alleles (D_{ba}) [22].

(3) D_{hbw} and D_{hm} are depicted as

$$D_{hbw} = \frac{1}{m} \sum_{i=1}^{m} (x_{bi} \oplus x_{wi}) \tag{5}$$

$$D_{hm} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left\{ \frac{1}{m} \sum_{k=1}^{m} (x_{ik} \oplus x_{jk}) \right\} \tag{6}$$

where x_{bi} and x_{wi} are the i-th bits in the best and worst binary solutions, respectively; m is the number of bits in a binary solution; n is the number of individuals in a population; the symbol $\oplus$ is exclusive OR operator; x_{ik} and x_{jk} are the k-th bits in the i-th and j-th binary solutions, respectively. D_{hbw} and D_{hm} vary between 0 and m. Larger values of D_{hbw} and D_{hm} indicate more varieties between the best and worst binary individuals, and each pair of binary individuals in a population, respectively.

(4) D_{bc} and D_{ba} are defined as

$$D_{bc} = \frac{1}{n-1} \left(\frac{\sum_i S_i^2}{L} - \frac{S^2}{L * n} \right) \tag{7}$$

$$D_{ba} = \frac{1}{L-1} \left(\frac{\sum_j S_j^2}{n} - \frac{S^2}{L * n} \right) \tag{8}$$

where n is the population size; L is the length of a chromosome; S is the sum of genes ‘1’; S_i and S_j are the sum over a row i and the sum over a column j, respectively. D_{bc} and D_{ba} have the following properties [23]:

(i) If the population is homogeneous in either 0 or 1, D_{bc} and D_{ba} equals zero;

(ii) When all the chromosomes in the population are identical, D_{bc} is zero and D_{ba} holds a constant value that is dependent on how many genes ‘1’ are in a chromosome.

In what follows we use knapsack problems, which are described in Section 4.1, to show the changes of the six population diversities in the evolution. Figures 1–6 illustrate comparisons of

QEPS and QIEA for three knapsack problems with 400, 600 and 800 items. Each subfigure in Figs.1–6 provides results of 30 independent random runs (green solid lines for QEPS and cyan solid lines for QIEA) and the mean values over 30 runs (black bold solid lines for QEPS and black bold dash-dot lines for QIEA). The values of D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc} and D_{ba} in Figs.1–6 are obtained by setting population size to 20 and the maximal number 20000 of function evaluations (NoFE) as the stopping condition.

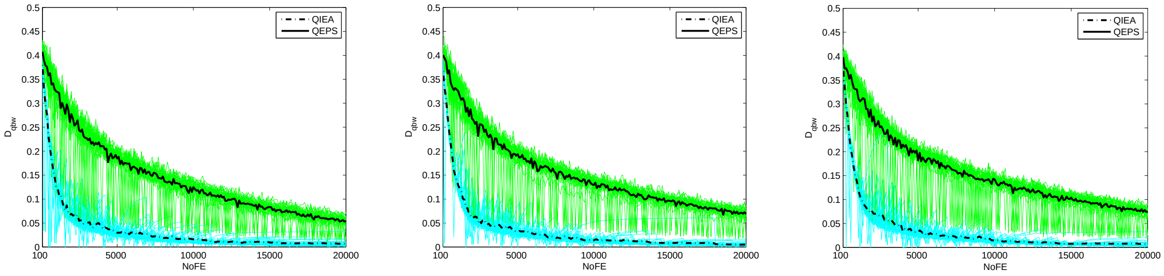
From the results, shown in Figs.1–6, about the comparisons of population diversity between QEPS and QIEA, we can draw the following conclusions:

(i) The three subfigures, which correspond to the respective knapsack problems with 400, 600 and 800 items, in each of Figs.1–6, show respectively consistent trends for QEPS and QIEA, which indicates the reasonableness of the six diversity measures to a certain degree.

(ii) Figures 1–2 show that QEPS can maintain better population diversity than QIEA in Q-bit space. The two algorithms have approximate values, of about 0.4, for D_{hbw} and D_{hm} in the initial states. As NoFE mounts up, the values of D_{hbw} and D_{hm} of Q-bit individuals in QEPS decrease gradually to around 0.5 at NoFE 20000, whereas the values in QIEA rapidly fall below 0.5 at NoFE 3000 and approximate 0 at NoFE 20000.

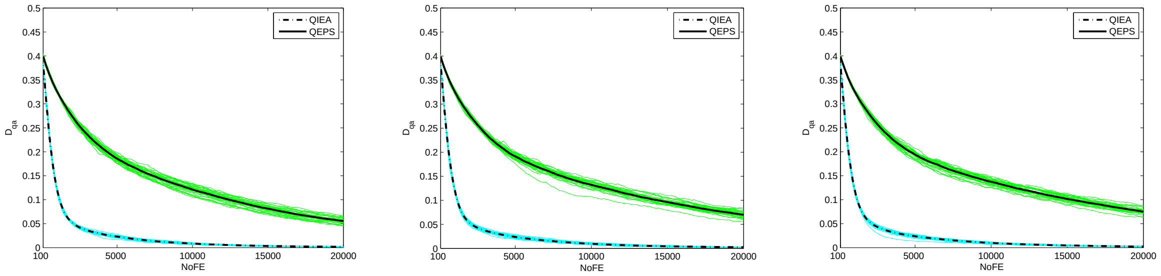
(iii) The Hamming distance D_{hbw} between the best and worst binary individuals and the mean Hamming distance D_{hm} of all binary individuals in Figs.3–4 show a clear picture of QEPS having a greater potential to preserve the population diversity than QIEA. More specifically, QEPS and QIEA have almost the same initial values and decreasing values for changes of D_{hbw} and D_{hm} with respect to NoFE, but QEPS maintains a much higher level of population diversity than QIEA throughout the evolutionary process. The six subfigures in Figs.3–4 also demonstrate that QIEA loses population diversity too fast and very quickly goes down to a small value close to 0 when NoFE reaches values around 10000, which implies that the individuals in QIEA become nearly identical and therefore lose exploration capability. When NoFE increases to 20000, QEPS still has one-fourth of initial values of D_{hbw} and D_{hm}.

(iv) D_{bc} and D_{ba}, two diversity measures based on dispersion statistical measures in Figs.5–6, also clearly illustrate that QEPS has better population diversity than QIEA through the whole process. D_{bc} values of QIEA rapidly fall down to 0 and D_{ba} values of QIEA rapidly rise to and stay at a higher steady level, while QEPS maintains more varieties in the population, even when the algorithm stops at values of NoFE close to 20000.



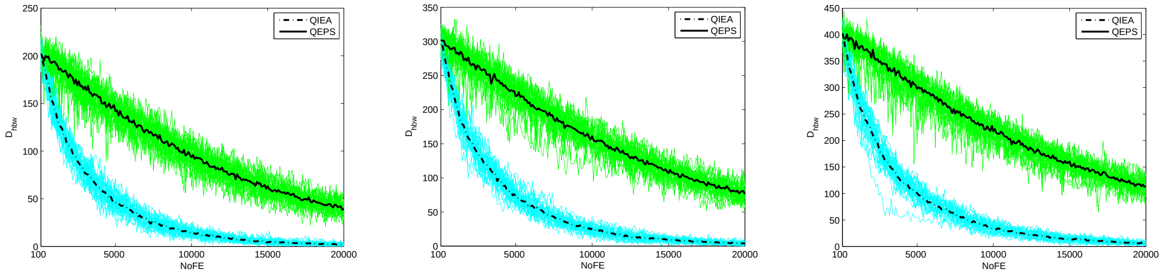
(a) 400 items (b) 600 items (c) 800 items

Figure 1: D_{qbw} of QEPS and QIEA with items 400, 600 and 800.



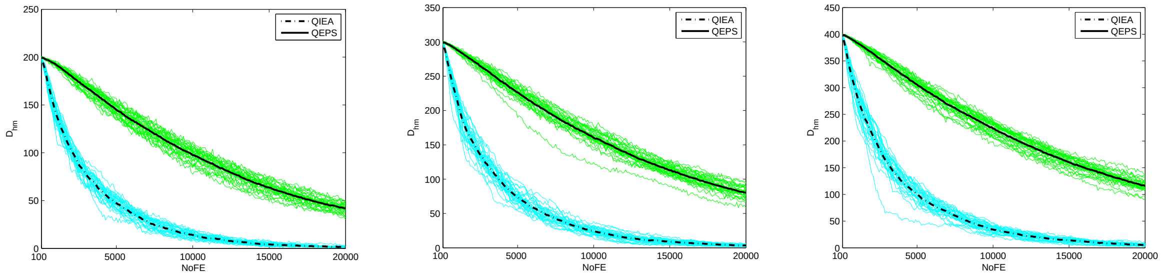
(a) 400 items (b) 600 items (c) 800 items

Figure 2: D_{qa} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items (b) 600 items (c) 800 items

Figure 3: D_{hbw} of QEPS and QIEA with items 400, 600 and 800.



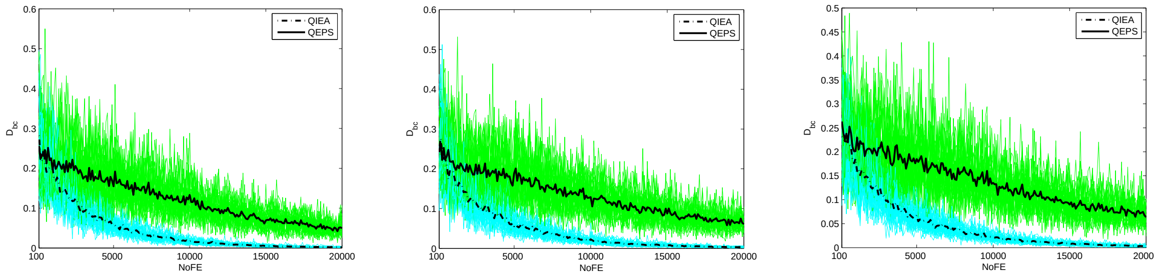
(a) 400 items (b) 600 items (c) 800 items

Figure 4: D_{hm} of QEPS and QIEA with items 400, 600 and 800.

The loss of population diversity means that the algorithm will fail to further explore the solution space. In the following description, we will go further to analyze the convergence performance of membrane algorithms.

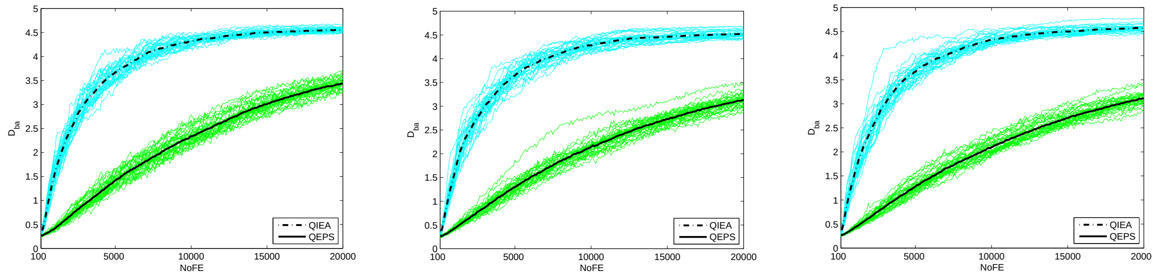
3.2 Convergence Analysis

Convergence is very important for a meta-heuristic search method as it shows the speed of the method in finding a satisfactory solution to an optimization problem. In this subsection, the convergence behavior of MIEAs is observed by presenting four measures: best Q-bit individual convergence (C_{qb}), average Q-bit individual convergence (C_{qa}), the best fitness convergence (C_{fb})



(a) 400 items (b) 600 items (c) 800 items

Figure 5: D_{bc} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items (b) 600 items (c) 800 items

Figure 6: D_{ba} of QEPS and QIEA with items 400, 600 and 800.

and the average fitness convergence (C_{fa}).

Both QEPS and QIEA use Q-bit individuals to construct a population. Thus we can apply the best Q-bit individual convergence and the average Q-bit individual convergence in a population to observe how much Q-bits approach 0 or 1 in the searching process. Their definitions are as follows.

(i) The best Q-bit individual convergence is described as

$$C_{qb} = \frac{1}{m}\sum_{j=1}^{m} \max\left\{|\alpha_{bj}|^2, |\beta_{bj}|^2\right\} \tag{9}$$

where $[\alpha_{bj}\ \beta_{bj}]^T$ is the j-th Q-bit in the best Q-bit individual corresponding to the best fitness in a population; m is the number of Q-bits in a Q-bit individual. $0.5 \le C_{qb} \le 1$.

(ii) The average Q-bit individual convergence is depicted as

$$C_{qa} = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{1}{m}\sum_{j=1}^{m} \max\left\{|\alpha_{ij}|^2, |\beta_{ij}|^2\right\}\right\} \tag{10}$$

where $[\alpha_{ij}\ \beta_{ij}]^T$ is the j-th Q-bit in the i-th Q-bit individual in the population with n individuals; m is the number of Q-bits in a Q-bit individual. $0.5 \le C_{qa} \le 1$.

C_{qb} and C_{qa}, calculated in the Q-bit space, can be regarded as genotypic convergence measures. C_{qb} and C_{qa} have not a direct relationship to the quality of solutions. Therefore we employ the other two convergence measures, the best fitness and the average fitness, to observe the convergence rates of solutions.

(iii) The description of C_{fb} and C_{fa} is given as follows

$$C_{fb} = \max_{i=1}^{n} f_i(x) \tag{11}$$

$$C_{fa} = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{12}$$

where $f_i(x)$ is the fitness of the i-th individual. Equation 11 is listed based on a maximum optimization. For a minimum problem, C_{fb} is to find the minimal fitness among n solutions.

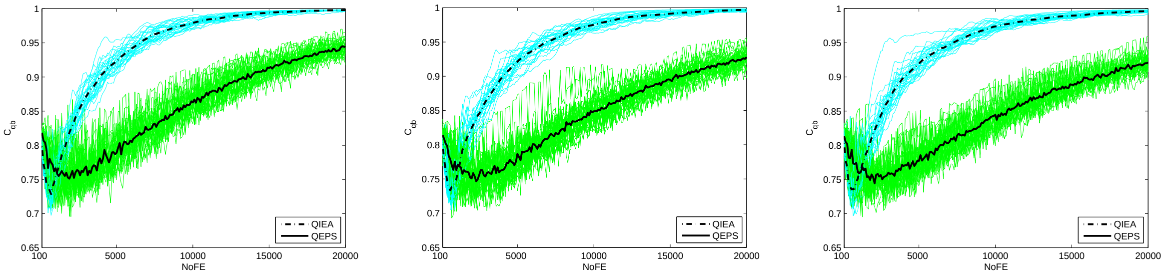
We still apply the three knapsack problems with 400, 600 and 800 items to observe the convergence performances of QEPS and QIEA. The population size, NoFE and independent runs are assigned as 20, 20000 and 30, respectively. The changes of C_{qb}, C_{qa}, C_{fb} and C_{fa} are shown in Figs.7–10, where each subfigure provides the results of 30 independently random runs (green solid lines for QEPS and cyan solid lines for QIEA) and the mean values over 30 runs (black bold solid lines for QEPS and black bold dash-dot lines for QIEA). The comparisons between QEPS and QIEA, shown in Figs.7–10, give us the following hints.

(i) The three subfigures corresponding to the respective knapsack problems with 400, 600 and 800 items, in each of Figs.7–10, show respectively similar changes for QEPS and QIEA.

(ii) It can be seen from the results shown in Figs.7–8 that C_{qb} and C_{qa} have similar tendencies, to be specific, QIEA converges much faster in Q-bit space than QEPS and quickly arrives at the maximal value 1, which implies that no further improvement of solutions in QIEA can be gained at the second half of evolutionary processes. The drastic convergence easily makes QIEA trapped in local extrema and consequently a premature end of the evolutionary process appears. On the contrary, C_{qb} and C_{qa} of QEPS go up much slower than those corresponding to QIEA with respect to NoFE and finally mount up to around 0.9 for values of NoFE in the region of 20000, which suggests that the solutions can be further improved if more NoFE is provided.

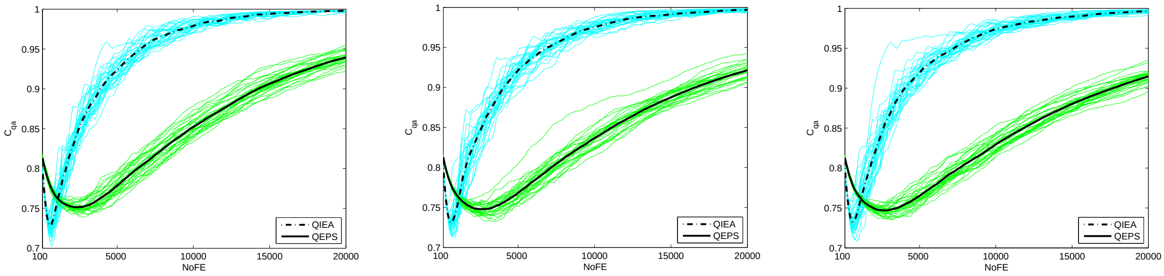
(iii) In Figs.9–10, QIEA has faster increases of C_{fb} and C_{fa} than QEPS and then stays at a relatively flat level after a certain NoFE, while QEPS goes through a slower start than QIEA and then rapidly goes beyond QIEA and keeps an ascending trend. Thus QEPS obtains better solutions than QIEA. The observations in Figs.9–10 can also be derived from the results in Figs.7–8. Additionally, it is worth noting, according to the six subfigures of Figs.9–10, that QEPS has better performance than QIEA in terms of the consistency of the results obtained for 30 independent runs when mean values are considered. This suggests that QEPS has better robustness properties than QIEA.

The convergence and population diversity are often conflicting features for population-based search methods. Rapid convergence usually results in a fast loss of population diversity, whereas better varieties of individuals produce more possibilities to improve solutions. The dynamic behaviors can be observed from the changes of population diversity, shown in Figs.1–6, and convergence performance in Figs.7–10. The diversity and convergence analysis above indicate that QEPS can achieve a better trade-off between convergence and diversity than QIEA, i.e., better balance between exploration and exploitation than QIEA. This better balance of these two essential features of any evolutionary approach is the principal explanation of the fact that QEPS achieves high quality solutions, better than QIEA if NoFE is large enough. For example, NoFE is greater than 10000 for the knapsack problems with 400, 600 and 800 items, which corresponds to 100 evolutionary generations.



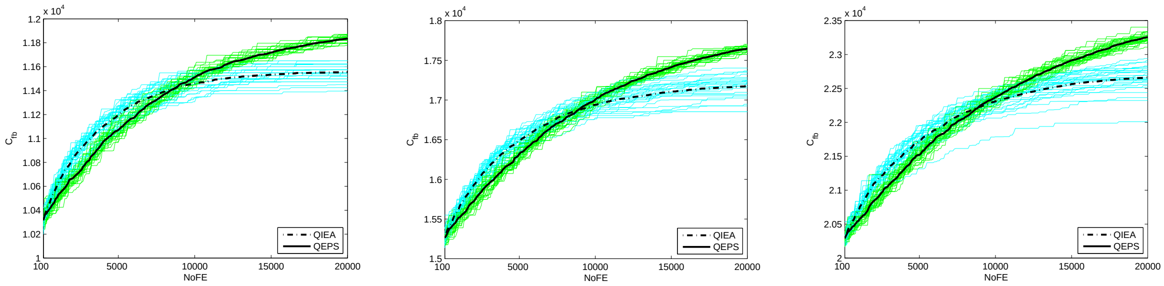
(a) 400 items (b) 600 items (c) 800 items

Figure 7: C_{qb} of QEPS and QIEA with items 400, 600 and 800.



(a) 400 items (b) 600 items (c) 800 items

Figure 8: C_{qa} of QEPS and QIEA with items 400, 600 and 800.

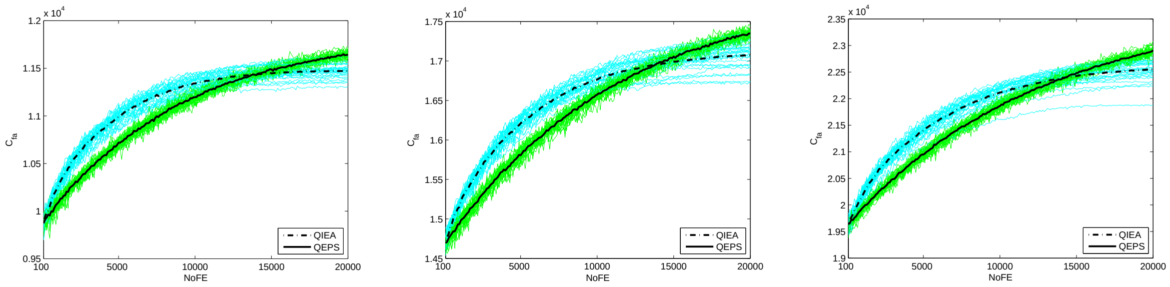


(a) 400 items (b) 600 items (c) 800 items

Figure 9: C_{fb} of QEPS and QIEA with items 400, 600 and 800.

3.3 Correlation between Measures and Fitness

The goal of MIEAs is to find the optimal solution of an optimization problem, so the relationship between the diversity and convergence measures and fitness is very important for improving



(a) 400 items

(b) 600 items

(c) 800 items

Figure 10: C_{fa} of QEPS and QIEA with items 400, 600 and 800.

the algorithm performance with respect to diversity and convergence. In this subsection, we introduce a tendency correlation coefficient to evaluate the importance of the eight measures: D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc}, D_{ba}, C_{qb} and C_{qa}.

The tendency correlation coefficient of two sequences, $S_1 = (s_1^1, s_2^1, \cdots, s_L^1)$ and $S_2 = (s_1^2, s_2^2, \cdots, s_L^2)$, is defined as

$$\rho = \frac{\sum_i (\Delta S_1 \cdot \Delta S_2)}{\sqrt{\sum_i \Delta {S_1}^2 \cdot \sum_i \Delta {S_2}^2}} \tag{13}$$

where $\Delta S_1 = (\Delta s_1^1, \Delta s_2^1, \cdots, \Delta s_{L-1}^1)$ and $\Delta S_2 = (\Delta s_1^2, \Delta s_2^2, \cdots, \Delta s_{L-1}^2)$, where Δs_k^1 and Δs_k^2 $(k = 1, 2, \cdots, L-1)$ are respectively

$$\Delta s_k^1 = s_{k+1}^1 - s_k^1 \tag{14}$$

$$\Delta s_k^2 = s_{k+1}^2 - s_k^2 \tag{15}$$

The tendency correlation coefficient ρ varies in the range between -1 and 1. The maximal value 1 and minimal value -1 mean that the two sequences S_1 and S_2 have identical and completely opposite tendencies, respectively. Thus a larger absolute value of ρ indicates a stronger tendency correlation.

We use the tendency correlation coefficient to analyze the relationship between each of the eight measures, D_{qbw}, D_{qa}, D_{hbw}, D_{hm}, D_{bc}, D_{ba}, C_{qb} and C_{qa}, and each of two kinds of fitness, C_{fb} and C_{fa}. Experimental results are listed in Table 1, where each datum is calculated by using the statistical results of three knapsack problems with 400, 600 and 800 items in Sections 3.1–3.2.

It can be seen from experimental results in Table 1 that the three diversity measures, D_{qa}, D_{hm} and D_{ba}, have stronger tendency correlation with fitness than the other five measures. So the improvement of D_{qa}, D_{hm} and D_{ba} could be a promising way to enhance the algorithm performance. In the column of D_{ba}, positive tendency correlation coefficients show that D_{ba} and fitness have similar trends, which implies that a larger value of D_{ba} may correspond to a better fitness. While in the columns of D_{qa} and D_{hm}, negative values demonstrate that the two population diversity measures have contrary trends with fitness, which might mean that smaller values of D_{qa} and D_{hm} could result in better solutions.

Table 1: Correlation between fitness and diversity, convergence measures.

Measures			D_{qbw}	D_{qa}	D_{hbw}	D_{hm}	D_{bc}	D_{ba}	C_{qb}	C_{qa}
QIEA	400 items	C_{fb}	-0.7335	-0.7804	-0.7706	-0.9251	-0.4812	0.9180	0.1627	0.1707
		C_{fa}	-0.6894	-0.7174ĄĄ	-0.7004	-0.8643	-0.6312	0.8500	0.1579	0.1606
	600 items	C_{fb}	-0.6878	-0.6976	-0.7737	-0.9064	-0.3604	0.8921	0.2663	0.2412
		C_{fa}	-0.6548	-0.6818	-0.7792	-0.8940	-0.4993	0.8764	0.2591	0.2493
	800 items	C_{fb}	-0.6812	-0.6664	-0.8153	-0.9086	-0.4291	0.8969	0.2914	0.2865
		C_{fa}	-0.6491	-0.6794	-0.8174	-0.9171	-0.4466	0.9035	0.2436	0.2749
QEPS	400 items	C_{fb}	-0.3260	-0.8679	-0.3008	-0.7910	-0.1042	0.7715	-0.0007	0.0061
		C_{fa}	-0.2004	-0.6672	-0.2155	-0.6895	-0.4590	0.6175	0.0220	0.0246
	600 items	C_{fb}	-0.3004	-0.8260	-0.3192	-0.8121	-0.0638	0.7960	0.0522	0.0427
		C_{fa}	-0.2300	-0.7009	-0.2148	-0.7635	-0.3809	0.7067	0.0231	0.0691
	800 items	C_{fb}	-0.2531	-0.8121	-0.2878	-0.8472	-0.0524	0.8356	0.1060	0.0573
		C_{fa}	-0.2057	-0.7123	-0.2370	-0.7767	-0.3477	0.7336	0.0575	0.0766

4 Examples

In this section, a knapsack problem is described and more experiments are conduced to further verify the observations in the preceding section.

4.1 Knapsack Problem

A knapsack problem can be described as the process of selecting from among various items those that are most profitable, given that the knapsack has limited capacity [12, 20, 21]. The knapsack problem is to select a subset from the given number of items so as to maximize the profit $f(x)$:

$$f(x) = \sum_{i=1}^{k} p_i x_i \tag{16}$$

Subject to

$$\sum_{i=1}^{k} w_i x_i \leq C_k \tag{17}$$

where k is the number of items; p_i is the profit of the i-th item; w_i is the weight of the i-th item; C_k is the capacity of the given knapsack; and x_i is 0 or 1. This paper uses strongly correlated sets of unsorted data: w_i =uniformly random [1, 50], $p_i = w_i + 25$. The average knapsack capacity C_k is applied.

$$C_k = \frac{1}{2} \sum_{i=1}^{k} w_i \tag{18}$$

4.2 Experiments and Results

In this subsection, we carry out the experiments on 15 knapsack problems with 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800 and 3000 items to compare the performance of QEPS and QIEA. In the experiments, QEPS and QIEA use 20 individuals and 30 independent runs are performed for each case. QEPS uses the OLMS, where 15 elementary membranes and the maximal number 9 of iterations for each elementary membrane are considered, according to the investigation in [12]. The stopping condition for QEPS and QIEA is set as follows: 20000 NoFE for the first four knapsack problems; 30000 NoFE for the three knapsack problems with 1000, 1200 and 1400 items; 40000 NoFE for the four knapsack problems with 1600, 1800, 2000 and 2200; 60000 NoFE for the last four knapsack problems. The best, worst and average solutions over 30 independent runs are recorded and listed in Table 2. We also provide the results, shown in Table 3, for each problem when NoFE is one-tenth of the prescribed value.

Several facts can be obtained from the results in Tables 2 and 3.

(i) The results in Table 3 show that QIEA achieves better results than QEPS when NoFE is only one-tenth of the prescribed value for each of the 15 knapsack problems.

(ii) The statistical data in Table 2 show that QEPS is superior to QIEA in terms of the quality of solutions. Even the worst solution of each problem obtained by QEPS is better than the best one provided by QIEA.

(iii) It seems that there is a conflict between the two conclusions in (i) and (ii). Actually these conclusions can be explained by using the analysis of population diversity and convergence in Section 3. The advantage of QIEA over QEPS in Table 3 comes from the fact that QIEA has a faster convergence speed and shows more rapid changes of population diversity than QEPS. Slower changes can also be regarded as a better balance, between diversity and convergence, leading to improved searching capabilities for QEPS, which results in the superiority of QEPS over QIEA, as shown in Table 2. Moreover, these experimental results provide sufficient details explaining the behavior of QEPS and QIEA, as stated in Section 3.

Table 2: Experimental results for QEPS and QIEA, where NoFE for each problem is prescribed.

Items	QIEA			QEPS		
	Best	Average	Worst	Best	Average	Worst
200	5885	5786	5359	5959	5945	5909
400	11650	11553	11396	11873	11837	11778
600	17403	17173	16851	17702	17647	17575
800	22940	22659	22010	23403	23257	23109
1000	28673	28333	27954	29531	29373	29198
1200	34399	33984	33424	35441	35292	35061
1400	39560	39149	38488	40886	40722	40364
1600	45277	44864	44423	47242	47018	46672
1800	50784	50163	49506	52772	52600	52395
2000	56453	55879	55129	58775	58543	58065
2200	61645	61175	59820	64513	64230	63680
2400	66683	65984	64981	70402	70015	69726
2600	72546	71992	71497	76621	76245	75296
2800	77511	76734	75924	81918	81486	80683
3000	83294	82608	82020	88207	87657	87044

5 Conclusion

The dynamic behavior analysis of MIEAs is very illustrative for a better understanding of the role played by P systems in the context of evolutionary algorithms. This paper discusses

Table 3: Experimental results for QEPS and QIEA, where NoFE for each problem is one-tenth of the prescribed value.

Items	QIEA			QEPS		
	Best	Average	Worst	Best	Average	Worst
200	5574	5476	5359	5496	5395	5306
400	10964	10836	10687	10744	10666	10548
600	16004	15928	15775	15914	15721	15549
800	21234	21096	20860	21230	20839	20686
1000	26755	26571	26309	26501	26274	26124
1200	32148	31925	31707	31849	31635	31460
1400	36990	36832	36553	36754	36482	36179
1600	42918	42588	42352	42634	42270	42038
1800	47878	47559	47353	47428	47121	46788
2000	53376	53068	52803	53001	52654	52433
2200	58646	58335	58102	58271	57957	57622
2400	63441	63195	62819	63226	62806	62328
2600	69342	69077	68676	69196	68755	68420
2800	74051	73613	73269	73524	73207	72833
3000	79630	79236	78743	79361	78886	78483

for the first time significant aspects of developmental processes occurring in MIEAs using a comparative approach in an experimental context, whereby a set of population diversity and convergence measures were introduced and analyzed. This work not only provides some answers to the difficult and intriguing question of what are the benefits of using P systems in evolutionary algorithms, but also suggests possible improvements for MIEAs. On the basis of this work, a promising research might start with a multi-objective optimization framework utilizing concepts and principles of P systems in a systematic and consistent way.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61170016, 61373047), the Program for New Century Excellent Talents in University (NCET-11-0715) and SWJTU supported project (SWJTU12CX008)

Bibliography

[1] Păun, Gh. (1998); Computing with membranes, *J. Comput. Syst. Sci.*, ISSN: 0022-0000, 61: 108-143.

[2] Păun, Gh.; Rozenberg, G.; Salomaa, A. (2010); The Oxford Handbook of Membrane Computing, *Oxford University Press.*

[3] Pan, L.; Păun, Gh.; (2009); Spiking neural P systems with anti-spikes, *Int J Comput Commun*, ISSN: 1841-9836, 4: 273-282.

[4] Zhang, X.; Wang, J.; Pan, L. ; (2009); A note on the generative power of axon P systems, *Int. J. Comput. Commun.* ISSN: 1841-9836, 4: 92-98.

[5] Lu, C.; Zhang, X.; (2010); Solving vertex cover problem by means of tissue P systems with cell separation, *Int J Comput Commun*, ISSN: 1841-9836, 5: 540-550.

[6] Zhang, X.; Luo, B.; Pan, L.; (2012); Small universal tissue P systems with symport/antiport rules, *Int J Comput Commun*, ISSN: 1841-9836, 7: 173-183

[7] Zhang, G.X.; Liu, C.X.; Rong, H.N. (2010)Łť Analyzing radar emitter signals with membrane algorithms, *Math. Comput. Model*, ISSN: 0895-7177, 52: 1997-2010.

[8] Zhang, G.X.; Cheng, J.X.; Gheorghe, M.; Meng, Q. (2013); A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, *Appl. Soft Comput*, ISSN:1568-4946, 13:1528-1542.

[9] Păun, Gh.; Pérez-Jiménez, M.J. (2006); Membrane computing: brief introduction, recent results and applications, *Biosystems*, ISSN: 0303-2647, 85: 11-22.

[10] Nishida, T.Y. (2004); An application of P-system: A new algorithm for NP-complete optimization problems, *Proc. of WMSCI*, 109-112.

[11] Huang, L.; He, X.; Wang, N.; Xie, Y. (2007); P systems based multi-objective optimization algorithm, *Prog. Nat. Sci.*, 17: 458-465.

[12] Zhang, G.X.; Gheorghe, M.; Wu, C.Z. (2008); A quantum-inspired evolutionary algorithm based on P systems for knapsack problem, *Fund. Inform.*, ISSN: 0169-2968, 87: 93-116.

[13] Cheng, J.X.; Zhang, G.X.; Gheorghe, M.; Zeng, X.X. (2011); A novel membrane algorithm based on differential evolution for numerical optimization,*Int. J. Unconv. Comput.*, ISSN: 1548-7199, 7: 159-183.

[14] Zhang, G.X.; Gheorghe, M.; Li, Y.Q. (2012); A membrane algorithm with quantum-inspired subalgorithms and its application to image processing, *Nat. Comput.*, ISSN: 1567-781, 11: 701-717.

[15] Zhang, G.X.; Zhou, F.; Huang, X.L.; Cheng, J.X.; Gheorghe, K.; Ipate, F.; Lefticaru, R. A novel membrane algorithm based on particle swarm optimization for solving broadcasting problems, *J. Univers. Comput. Sci.*, ISSN: 0948-695X, 18: 1821-1841.

[16] Xiao, J.H.; Zhang, X.Y.; Xu, (2012); J. A membrane evolutionary algorithm for DNA sequences design in DNA computing. *Chinese Sci. Bull.*, ISSN: 1001-653, 57:698-706.

[17] Xiao, J.H.; Jiang, Y.; He, J.J.; Cheng, Z. (2013); A dynamic membrane evolutionary algorithm for solving DNA sequences design with minimum free energy, *MATCH Commun. Math. Comput. Chem.*, ISSN: 0340-6253, 70: 971-986.

[18] He, J.J.; Xiao, J.H.; Shi, X.L.; Song, T. (2013); A membrane-inspired algorithm with a memory mechanism for knapsack problems, *J. Zhejiang U.-SCI. C*, ISSN: 1869-1951, 14: 612-622.

[19] Xiao, J.H.; Huang, Y.F.; Cheng, Z; He, J.J.; Niu, Y.Y. (2014); A hybrid membrane evolutionary algorithm for solving constrained optimization problems. *Optik*, ISSN: 0030-4026, 125: 897-902.

[20] Han, K.H; Kim, J.H. (2002); Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE T. Evolut. Comput.*, ISSN: 1089-778X, 6: 580-593.

[21] Zhang, G.X. (2011); Quantum-inspired evolutionary algorithms: a survey and empirical study, *J. Heuristics*, ISSN: 1381-1231, 17: 303-351.

[22] Burke, E.K.; Gustafson, S.; Kendall, G. (2004); Diversity in genetic programming: an analysis of measures and correlation with fitness, *IEEE T. Evolut. Comput.*, ISSN 1089-778X, ISSN 1089-778X, 8: 47-62.

[23] Herrera, F.; Lozano, M. (1996); Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, 95-125.