Knowledge-Based Systems 105 (2016) 38–47

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A hybrid evolutionary algorithm based on tissue membrane systems and CMA-ES for solving numerical optimization problems

Chuang Liu*, Linan Fan

School of Information Engineering, Shenyang University, Liaoning, China

ARTICLE INFO

Article history:
Received 29 December 2015
Revised 21 March 2016
Accepted 26 April 2016
Available online 3 May 2016

Keywords:
Tissue membrane system
CMA-ES
Numerical optimization problems
Membrane computing
Optimization

ABSTRACT

In this paper, a new hybrid algorithm is proposed to solve the single objective real-parameter numerical optimization problems, named as CETMS. The proposed CETMS is based on tissue membrane systems(TMS), and the evolution strategy with covariance matrix adaptation (CMA-ES) algorithm is employed to find the optimal solution in each cell of TMS. Some features of Tissue Membrane Systems, such as membrane structure, evolution mechanism and communication mechanism among cells, are introduced into CETMS. In addition, the optimal information of different cells can be shared by communication mechanism of TMS after the appointed cycle. The simulation experiments are conducted on thirty benchmark functions on the CEC14 test suite, which evaluate the performance of the proposed algorithm on solving single objective real-parameter numerical optimization problems. Numerical results show that the proposed CETMS has a very good performance in comparison with some of the state-of-the-art algorithms.

1. Introduction

Numerical optimization problems exist widely in different areas of science research and engineering practice. In the past decades, these optimization problems are solved by using the traditional mathematical methods. With increasing complexity of these optimization problems, the traditional mathematical methods cannot find the satisfactory solutions. Therefore, the effective optimization algorithms are needed to solve these kinds of optimization problems. One class of the optimization algorithms inspired by natural computing is effective method to solve these problems, such as genetic algorithm [6], evolutionary strategy [9], particle swarm optimization (PSO) [38], ant colony optimization (ACO) [5], artificial bee colony (ABC) [13], cat swarm optimization (CSO) [3], and firefly algorithm (FA) [40]. There algorithms have been proved that their solutions found are closer to the global optimum one. However, when applying these evolutionary algorithms to solve the real-world problems, some new mechanisms need to be introduced into them in order to obtain the desired results.

Membrane computing is proposed by Păun from the European Academy of Sciences in 1998 [23]. It is an abstract computational idea or model, which is inspired by the structure and functioning of living cells, such as processing chemical compounds in tissues or higher order structures. The device computational model of membrane computing is called as membrane system or P system. A membrane system, as we will see later on, is a distributed and parallel theoretical computing device and whose aim is mimicking the inner mechanism of the living cell [24,25]. Now, three kinds of membrane systems are proposed, which include cell-like membrane system, tissue-like membrane systems and neural-like membrane systems. The key difference among the three kinds of membrane systems that cell-like membrane system provides the hierarchical framework of membranes, and tissue-like membrane system contains multiple single-cell structures connected each other, and neural-like membrane system is an extend model of tissue-like membrane system which is based on the concept of spikes. Many of their variants may be found in literature [22,33,41,46,47]. Most of these membrane systems could tackle specific problems (i.e., optimization problems) in a feasible time. At presents, applications of the membrane system have been expanded to various fields, e.g., medicine, biology, linguistics, fuzzy reasoning, data clustering, image processing and ecology [2,4,21,26–30,34,45].

The membrane system in optimization is called as membrane algorithm. Membrane algorithm provides the opportunity to apply evolutionary algorithms or swarm intelligence in a parallel and distributed environment [32]. The study on membrane algorithm has been an increasing interest, because membrane algorithm can solve the challenging and complicated optimization problems. At present, many kinds of membrane algorithms have been proposed, most of them will be discussed as follows. Nishida proposed a

* Corresponding author.
E-mail address: chuang.liu@mail.dlut.edu.cn (C. Liu).

http://dx.doi.org/10.1016/j.knosys.2016.04.025

compound membrane algorithm to solve the traveling salesman problem [19]. Zhang et al. proposed a membrane-inspired approximate algorithm for traveling salesman problems, which implemented the rules of ant colony optimization in membrane systems. Huang et al. presented an optimization algorithm inspired by the membrane system to solve optimization problems [48]. Zhao et al. proposed the bio-inspired algorithm based on membrane systems (BIAMC) for solving both unconstrained and constrained problems [49,50]. Zhang et al. proposed a hybrid algorithm based on the quantum-inspired evolutionary approach and membrane systems to solve a well-known combinatorial optimization the knapsack problem [42,43]. Huang et al. presented a dynamic multi-objective optimization algorithm [12], which is inspired by membrane systems. Simulation results verify the effectiveness of the algorithm. Buiu et al. proposed a membrane controller based on membrane systems for mobile robots [1]. Xiao et al. proposed a membrane evolutionary algorithm for the DNA sequence design problem. The results of computer experiments are reported, in which the new algorithm is validated and out-performs certain known evolutionary algorithms for the DNA sequence design problem [39]. Liu et al. presented a novel algorithm based on the membrane system for solving multi-objective optimization problems. The proposed algorithm could quickly obtain the approximate Pareto front and satisfy the requirement of diversity of Pareto front [15]. In [31], the authors have studied the ways in which rules of particle swarm optimization have been implemented in membrane systems. In each of these membrane algorithms, modification is done either by using a new variant of PSO or by changing the way of intercommunication among the membranes. Each of these membrane based algorithms is designed for specific problems and hence is implemented on different problems. Xiao et al. proposed a dynamic membrane evolutionary algorithm to solve the DNA sequences design. The results of simulation experiments show that the proposed algorithm is valid and outperforms other evolutionary algorithms [37]. He et al. proposes an adaptively chosen parameters membrane algorithm to solve combinatorial optimization problems. Compared with the genetic algorithm, simulated annealing algorithm, neural net-work and a fine-tuned non-adaptive membrane algorithm, the proposed algorithm performs better than them [10]. Xiao et al. proposed a hybrid membrane evolutionary algorithm to solve constrained optimization problems. The simulation results show that the proposed algorithm is valid and outperforms the state-of-the-art algorithms [36]. Zhang et al. proposes a novel way to design a membrane system for directly obtaining the approximate solutions of combinatorial optimization problems without the aid of evolutionary operators [44]. Extensive experiments on knapsack problems have been reported to experimentally prove the viability and effectiveness of the proposed neural system. Niu et al. proposes a membrane algorithm based on ant colony optimization to solve the capacitated vehicle routing problem. Experimental results show that the proposed algorithm is better than other algorithms proposed in the previous literature [20].

To the best of our knowledge, membrane algorithms have been successfully applied to solve many kinds of optimization problems. However, the study on the membrane algorithm is less to solve the real-parameter single objective optimization problems. Therefore, our goal is to design a novel membrane algorithm to find the global optimal solutions of the optimization problems. Based on our previous work [7,8,15,16], a hybrid evolutionary algorithm based on tissue membrane systems and CMA-ES is proposed in order to focus on solving numerical optimization problems, named CETMS. Compared with our previous work, the proposed CETMS has two improvements. On the one hand, the proposed algorithm has the different membrane structure, which is based on the tissue membrane systems. The previously proposed algorithm is based on cell membrane systems. Since CETMS is based on a tissue-like membrane system, it inherits three ingredients of the membrane system, such as objects, membrane structure and reaction rules. An object in CETMS represents a candidate solution of optimization problems. The structure consists of multiple single-cell connected each other, which is conducive to the proposed algorithm for the parallel exploration of the solution space. On the other hand, in terms of evolutionary mechanisms, the proposed algorithm employees CMA-ES. CMA-ES is employed directly as reaction rules to evolve the candidate objects in each cell.

To evaluate the performance of CETMS, the simulation experiment will be executed on the set of benchmark functions provided for IEEE CEC 2014 special session and competition on single objective real-parameter numerical optimization [14]. In addition, this paper carries out experiments on the sensitivity analysis in comparison with some state-of-the-art algorithms which include L-SHADE [35], POBL-ADE [11], and OptBees [17] from the literature. The simulation results indicate that the proposed algorithm can balance of exploration and exploitation.

The main purposes of this paper are:

- To present CETMS, which is an algorithm based on the tissue membrane system for optimization in continuous spaces;
- and to evaluate its performance by applying it to all thirty minimization problems proposed for the IEEE 2014 Congress on Evolutionary Computation Competition on Real-Parameter Single Objective Optimization (CEC'2014), considering spaces of 10, 30, 50 and 100 dimensions;
- and to compared with some of state-of-the-art evolutionary algorithms, indicate the distinguishing feature of CETMS.

The rest of this paper is organized as follows. In Section 2, we briefly discuss on some concepts of a membrane system, a tissue membrane system, and CMA-ES. In Section 3, the description of the proposed CETMS is elaborated. In Section 4, the simulation results are evaluated on the benchmark test problems in comparison with some state-of-the-art evolutionary algorithms. Moreover, this section includes a sensitivity analysis for the proposed CETMS. Finally, Section 5 summarizes the concluding remarks and future work of this paper.

2. Related work

2.1. Tissue membrane system

A tissue Membrane system is proposed by Martin-Vide in 2003, which can simulate the inter-cellular communication with protein channels in a net of cells [18]. In the tissue membrane system, each cell has a finite state memory, which processes multisets of objects. These cells can communicate with their neighboring cells. Such cell nets are shown to be rather powerful, which can simulate a Turing machine. Fig. 1 shows an example of network membrane structure of a Tissue Membrane System. The structure consists of several cells in a current environment. Each cell has a certain number of protein channels connecting the other cells.

The general form of a tissue membrane system can be described as follows. The tissue membrane system has a network membrane structure. Multisets of objects are placed in the cells, which can be evolved according to the reaction rules of the current cell. It is noteworthy that the cells can communicate with the neighboring cells by borrowing protein channels. It is benefit that the information of different cells can be shared in the current environment. For further details about the tissue membrane system can be found in [18]. The model of the tissue membrane system with the degree $n \geq 1$ can be constructed in Eq. (1).

$$\prod = (O, \delta_1, \cdots, \delta_n, syn, i_{out}) \tag{1}$$

where,

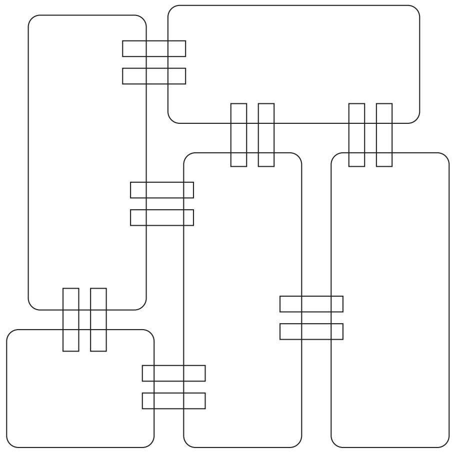

Fig. 1. An example of membrane structure of tissue membrane system.

- O is a finite non-empty alphabet of objects;
- $syn \subseteq \{1, 2, \cdots, n\} \times \{1, 2, \cdots, n\}$ (synapses among cell);
- $i_{out} \in 1, 2, \cdots, n$ indicates the output cell;
- $\delta_1, \cdots, \delta_n$ are cells, of the form

$$\delta_i = (Q_i, s_{i,0}, w_{i,0}, P_i), 1 \le i \le n, \tag{2}$$

where,
- Q_i is a finite set (of states);
- $s_{i,\,0} \in Q_i$ is the initial state;
- $w_{i,\,0} \in O*$ is the initial multiset of objects;
- P_i is a finite set of rules.

A tissue membrane system as above is said to be cooperative if it contains at least a rule $sw \to s'w'$ such that $|w| > 1$, and non-cooperative in the opposite case. The objects which appear in the left hand multiset w of a rule $sw \to s'w'$ are sometimes called impulses, while those from w' are also called excitations.

Any m-tuple of the form $(s_1w_1, \ldots, s_mw_m)$, with $s_i \in Q_i$ and $w_i \in E*$, for all $a \le i \le m$, is called a configuration of $\prod$; $(s_{1,0}w_{1,0}, \ldots, s_{m,0}w_{m,0})$ is the initial configuration of $\prod$.

During any transition, some cells can do nothing: if no rule is applicable to the available multiset of objects in the current state, then a cell waits until new objects are sent to it from its ancestor cells. It is also worth noting that each transition lasts one time unit, and that the work of the net is synchronized, the same clock marks the time for all cells.

A sequence of transitions among configurations of the tissue membrane system $\prod$ is called a computation of $\prod$. A computation which ends in a configuration where no rule in no cell can be used, is called a halting computation.

2.2. CMA-ES

CMA-ES is a kind of the effective evolutionary optimization strategy based on the derandomized evolution strategy with covariance matrix adaptation [9]. The covariance matrix adaptation is employed, which not only reduces the randomness of evolutionary strategy, but significantly accelerates its convergence rate. In addition, the concept of cumulative evolution path is introduced into CMA-ES, which can improve the overall performance of CMA-ES.

The process of the CMA-ES algorithm is described as follows.

Step 1) $g = 0$

Step 2) $\langle x \rangle_{\mu}^{(g)}$, which represents the center of mass of the μ individuals of the generation g, is initialized. $I_{sel}^{(g)}$ is the set of indices of the selected individuals of generation g with$|I_{sel}^{(g)}| = \mu$. The covariance matrix $C^{(g)}$ of the random vectors $B^{(g)}D^{(g)}z_k^{(g+1)}$ is a symmetrical positive $n \times n$ matrix. The columns of the orthogonal matrix $B^{(g)}$ represent normalized eigenvectors of the covariance matrix. $D^{(g)}$ is a diagonal matrix whose elements are the square roots of the eigenvalues of $C^{(g)}$. $\delta^{(g)} = 0.5$ is the global step size. The evolution path $p_c^{(g)}$ and $p_\delta^{(g)}$ are initialized as the zero vector.

Step 3) the λ offspring of generation $g + 1$ are computed by

$$x_k^{(g+1)} = \langle x \rangle_{\mu}^{(g)} + \delta^{(g)} \times N(0, C^{(g)}), k = 1, \cdots, \lambda \tag{3}$$

Step 4) the λ offspring of the population is evaluated

Step 5) the μ individual is chosen from the current population, and the $\langle x \rangle_{\mu}^{(g+1)}$ is recalculated, and $p_c^{(g+1)}$, $p_\delta^{(g+1)}$, $C^{(g+1)}$ and $\delta^{(g+1)}$ are updated

Step 6) the execution of CMA-ES is terminated until the end condition is satisfied. Otherwise, Step 3 is continued.

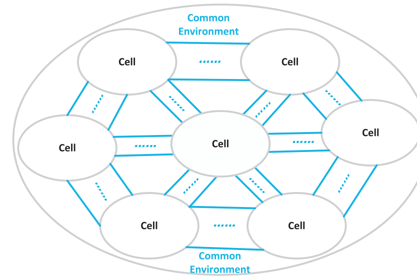

Fig. 2. The framework of the tissue-like membrane system.

3. The proposed algorithm

In this section, the description of a hybrid evolutionary algorithm based on a tissue membrane system is given. The tissue membrane system consists of a special network structure, multisets of objects, reaction rules. The network structure consists of several of cells. Each cell has its own multisets of objects. Theses multisets of objects can be evolved by reaction rules. After the appointed generation, these cells communicate with each other by using multisets of objects through protein channels. In the proposed algorithm, the objects in each cell represent the candidate solutions of optimization problems. CMA-ES is employed as the reaction rules to evolve objects in different cells. Reaction rules consist of evolutionary operators and communication rules of tissue membrane systems. These features are very useful to develop a hybrid approach to improve its solving performance.

The framework of the tissue-like membrane system is shown in Fig. 2, where ovals represent the cells and arrows indicate the channels. The tissue-like membrane system that we introduce for defining our optimization algorithm can be described in detail as follows.

The pseudo-code of the proposed algorithm is given in the Algorithm 1.

3.1. Objects

In tissue membrane systems, an object is an abstract representation of the atom, molecule and other chemical substance. In the proposed algorithm, the object represents a candidate solution of the optimization problem. Under meeting the constraints condition, the object is initialized using the decimal encode according to the formula (4).

$$x_{i,j} = x_j^l + (x_j^u - x_j^l) \times r \tag{4}$$

where $1 \le i \le N$, N represents the number of objects. $1 \le j \le D$, D denotes the dimension of an object. $x_{i,\,j}$ is the information of

Algorithm 1 The pseudo-code of the proposed algorithm.

Input: The parameters of the proposed algorithm are initialized, including the number of cells NC, the number of objects in each cell, each object within its boundaries
Output: The best object is found from the different cells.
for $i = 1$; $i < NC$; $i++$ **do**
  The objects of the different cells are initialized according to the formula 4.
  The fitness of these objects is calculated according to the objective functions of the optimization problems.
end for
while End Condition **do**
  for $i = 1$; $i < NC$; $i++$ **do**
    Execute the Algorithm 2;
    Recoding the best object of the current cell by invoking communication rule
  end for
  The good objects from the different cells is compared, and the best object is found according to the fitness of the optimization problems.
  Sending the best object to the different cells by invoking communication rule
end while

the j-th in i-th object. x_j^l represent the j-th lower boundary, and x_j^u represent the j-th upper boundary. r denotes a random function which can generate the number on the interval (0, 1).

3.2. Membrane structure

Since CETMS is based on the tissue membrane system, it inherits the same network structure from the tissue membrane system. The structure consists of several cells. In the experiments, the number of cells is set to 5. In order to simplify the implementation of the structure, the structure of CETMS is defined as the inner region of a cell that does not contain the cell. Each cell can be seen as an evolution unit. In the proposed algorithm, the initialization of objects is implemented in each cell. The initialized objects are evolved in the inner region of own cell according to executing the reacting rules. After executing several generation, some good objects can be generated by executing reaction rules in the different cells. These good objects are sent into the common environments of all cells. The best object can be found by comparing the fitness of these objects. The worst object in each cell is replaced with the best object from the common environments, which helps other objects to move toward the direction of the global optimal solution.

The structure of CETMS is benefit to improve the search efficiency of the proposed algorithm, which is suit to solve the complexity real-world numerical optimization problems.

3.3. Reaction rules

Reaction rules are inspired by the way of handling the compound in the cell. In the proposed algorithm, the reaction rules can be implemented by CMA-ES, which can evolve the objects toward the direction of the global optimal solution. The CMA-ES algorithm is implemented in each cell. The pseudo-code of CMA-ES is described in the Algorithm 2.

Communication rules not only send the best object from the different cells to the common environments of all cells, but send the common best object to the different cells. The rules are benefit to share the information of candidate solutions among cells.

Algorithm 2 The pseudo-code of CMA-ES.

The μ objects is chosen from the current cell;
The λ new objects of generation $g+1$ are computed by

$$x_k^{(g+1)} = \langle x \rangle_\mu^{(g)} + \delta^{(g)} \times N(0, C^{(g)}), k = 1, \cdots, \lambda$$

The λ new offspring is evaluated according to the optimization problems;
The $\langle x \rangle_\mu^{(g+1)}$ is recalculated, and $p_c^{(g+1)}$, $p_\delta^{(g+1)}$, $C^{(g+1)}$ and $\delta^{(g+1)}$ are updated;
The execution of CMA-ES is terminated until the end condition is satisfied. Otherwise, the above-mentioned steps of CMA-ES is executed repeatedly.

4. Experimental results

In this section, the performance of the proposed algorithm is evaluated in solving the CEC2014 competition on real-parameter optimization problems [14]. Section 4.1 will discuss the CEC2014 benchmark set. Section 4.2 will describe the experimental condition when the simulation is run. Section 4.4 will discuss the experimental results on 30 groups of benchmark test functions in IEEE CEC2014.

4.1. Description of CEC2014 benchmark set

To assess the performance of the proposed algorithm on single objective real-parameter numerical optimization, thirty benchmark functions on the CEC2014 test suite are employed in the following experiments. These functions are used to conduct the performance analysis of the proposed algorithms. All these benchmark functions are evaluated as the minimization problems. More details about the definition of these functions can be found in the literature [14].

The benchmark set, which is provided by CEC2014, is proposed to evaluate the performance of the evolutionary algorithms in real-parameter optimization problems. The test suite consists of 30 benchmark functions, which includes four kinds of functions. More specifically, Functions from F_1 to F_3 are unimodal. Simple Multimodal Functions consists of thirteen functions from F_4 to F_{16}. Considering that in the real-world optimization problems, different subcomponents of the variables may have different properties, therefore functions from F_{17} to F_{22} are proposed as the hybrid functions. In this set of hybrid functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents. The remaining eight functions of the test suite are composition functions which combine multiple test problems into a complex landscape. Each function of these test suite is a minimization problem. Different from CEC2013, each function has a shift data for CEC2014. For convenience, the same search ranges are defined for all test functions, which is $[-100, 100]^D$.

According to the guidelines requirements of CEC2014 benchmark competition, all experimental algorithms are performed. More specifically, when the solution found by the experimental algorithm is smaller than 10^{-8}, the error is set to 0. The dimension number(D) of these test functions is set to 10, 30, 50, and 100, respectively. The maximum evaluated times is set to $D \times 10,000$ on each run. Each problem is evaluated at 51 times. Moreover, five statistical metrics are designed, such as Best, Worse, Median, Mean, and Std [14]. These metrics can be employed to evaluate the solving performance of these various algorithms.

Table 1
Algorithm complexity.

Dimension	T0	T1	$\hat{T}2$	($\hat{T}2$-T1)/T0
D = 10	0.119722	0.621780	8.4514	65.3983
D = 30		0.806945	9.7243	74.4838
D = 50		1.248416	10.9553	81.0785
D = 100		3.130678	18.233975	126.1531

4.2. Experimental conditions

All experiments are run in Windows 7 enterprise version under the hardware environment of Intel Pentium dual-core 2.93 GHZ and 16 GB RAM. The proposed algorithm is implemented using matlab2015.

According to the guidelines requirements of CEC2014 benchmark competition, all experimental algorithms are performed [14]. More specifically, when the error between the optimal solution and the solution found by the experimental algorithm is smaller than 10^{-8}, the result is set to 0. The dimension number(D) of these test functions is set to 10, 30 50 and 100, respectively. To reduce the randomness of the algorithm, each test function is executed independently 51 times. The maximum evaluated times is set to $D \times 10,000$ on each run. The solution error measure, defined as $f(x') - f(x^*)$, is employed to evaluate the performance of the proposed algorithm, where x^* is the global optimum of the benchmark function and x' is the best solution attained by the algorithms in the experiment.

4.3. Complexity of the proposed algorithm

In this section, the algorithm complexity for the proposed CETMS is described. Table 1 shows the algorithm complexity on different dimensions, including 10, 30, 50 and 100 dimensions. As defined in [14], $T0$ is the time calculated by running the following test problem in the Algorithm 3:

Algorithm 3 Test problem

for $i = 1$; $i < 1000000$; $i++$ **do**
  $x = 0.55 + (double)i$;
  $x = x + x; x = x/2$;
  $x = x * x; x = sqrt(x)$;
  $x = log(x); x = exp(x); x = x/(x + 2)$;
end for

$T1$ is the time to execute 200,000 evaluations of benchmark function F18 with a certain dimension D. The complete computing time for the proposed algorithm with 200,000 evaluations of the same D dimensional F_{18} is $T2$. $\hat{T}2$ is the mean value for $T2$ with 5 runs ($\hat{T}2 = Mean(T2)$). The complexity of the proposed algorithm is presented on 10, 30, 50 and 100 dimensions in Table 1, which shows the algorithm complexities relationship with dimension.

According to Table 1, it is easy to conclude that more computational cost is required with the increasing number of dimensions for the benchmark functions.

4.4. Statistical results of the proposed algorithm

In this section, the proposed algorithm is run independently 51 times, and five statistical metrics are calculated, such as Best, Worse, Median, Mean, and Std [14]. These metrics can be used to analysis the performance of the proposed algorithm. The maximum number of objective function evaluations is $D \times 10,000$. Table 2 and Table 3 show the computational results of the proposed algorithm in solving the CEC2014 competition on real-parameter optimization problems for dimension 10, 30, 50, and 100. Each column shows Best, Worse, Median, Mean, and Std of the error value between the true optimal value and the best fitness values found by the algorithm in each run. Max and Min represent the maximum and minimum fitness values of the algorithm over the 51 runs, respectively. Median denotes the median of the result fitness values over the 51 runs. Mean represents the average value of the result fitness values over the 51 runs. Std is the corresponding standard deviation. It is worth noting that the error values smaller than $1.00e-8$ has been set as zero.

From the results obtained in Table 2 for dimension 10, and 30, the proposed algorithm could perform good results in unimodal problems from F_1 to F_3. While for multimodal function F_4, F_6, F_7, F_8, F_{12}, F_{13}, F_{14}, the error value is still smaller than $1.00e-01$. While for all the hybrid functions, the error value is still small than $1.00e-01$ from F_{17} to F_{22} for dimension 10. The rest of the functions, named the composition functions, the best error value is still larger than $1.00e+02$ except F_{26}, F_{27} for dimension 10.

Similarly, Table 3 for dimension 50, and 100 presents the statistical results. From the comparisons in terms of the unimodal problems, the proposed algorithm could perform good results from F_1 to F_3. For multimodal functions, the error value of the proposed algorithm is smaller than $1.00e+00$ on F_7, F_{12}, F_{13}, F_{14}. While for all the hybrid functions, the error value is still small than $1.00e+03$ from F_{17} to F_{22}. The rest of the functions, named the composition functions, the best error value is still larger than $1.00e+04$ from F_{23} to F_{30}.

In summary, the proposed algorithm has obtained good results on the most of functions from low dimension to high dimension. Compared with solving unimodal functions, the proposed algorithm is difficult to solve multimodal functions, hybrid functions, and composition functions.

4.5. Comparison of CETMS with some state-of-the-art proposed algorithms

In this section, the proposed CETMS is compared with three state-of-the-art metaheuristic algorithms, including L-SHADE [35], POBL-ADE [11] and OptBees [17], CMA-ES [9].

- L-SHADE is a variant of the Success-History based Adaptive Differential Evolution(SHADE), which continually decreases the population size according to a linear function [35]. On the CEC2014 test suite, the solving performance of L-SHADE is quite competitive with the state-of-the-art evolutionary algorithms.
- POBL-ADE is a variant of the Adaptive Differential Evolution(ADE), which focuses a set of partial opposite points of an estimate by a partial opposition-based learning schema [11]. Simulation results over the CEC2014 test suite demonstrate the effectiveness and improvement of the POBL-ADE compared with ADE.
- OptBees is an algorithm inspired by the processes of collective decision-making by bee colonies [17]. The robustness of OptBees is proved by conducting the CEC2014 test suite.

In order to evaluate the performance of the proposed CETMS, four groups of simulation are executed on 10, 30, 50, and 100 dimensions over the CEC2014 test suite. All experimental algorithms are run independently 51 times on each test problem, and two statistical metrics are calculated, such as Mean and Std. The best mean results among the comparative algorithms on each problem are shown in bold.Tables 4–7 respectively present the experimental results on unimodal, multimodal, hybrid and composition functions.

Table 4 shows the experimental results of all algorithms on 10 dimension. On the first unimodal group, the proposed CETMS, L-SHADE and CMA-ES can obtain the best Mean and Std. However, POBL-ADE and OptBees only can get the not bad results on F_3.

Table 2
The statistical results of CETMS on the CEC2014 benchmarks for $D = 10, 30$ dimensions.

Fun	**10D**					**30D**				
	Best	Worse	Median	Mean	Std	Best	Worse	Median	Mean	Std
F1	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F2	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F3	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F4	0.00e+0	3.47e+1	0.00e+0	6.95e+0	1.55e+1	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F5	1.84e+1	2.01e+1	2.00e+1	1.95e+1	7.64e-1	1.99e+1	2.10e+1	2.00e+1	2.01e+1	3.49e-1
F6	0.00e+0	8.94e-1	0.00e+0	1.78e-1	4.00e-1	0.00e+0	6.14e+0	1.63e+0	1.86e+0	1.77e+0
F7	9.85e-3	6.39e-2	3.44e-2	3.10e-2	2.18e-2	0.00e+0	1.25e+1	0.00e+0	6.53e-1	2.65e+0
F8	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	6.96e+0	2.20e+1	1.29e+1	1.31e+1	3.32e+0
F9	3.97e+0	1.09e+1	5.96e+0	7.36e+0	2.95e+0	1.98e+0	2.38e+1	1.19e+1	1.24e+1	5.20e+0
F10	1.24e-1	6.82e+0	3.12e-1	2.22e+0	2.97e+0	2.64e+2	6.01e+3	1.90e+3	2.04e+3	1.26e+3
F11	5.85e+1	5.45e+2	2.70e+2	3.00e+2	2.14e+2	2.40e+2	5.07e+3	2.43e+3	2.51e+3	1.06e+3
F12	1.20e-1	2.28e-1	1.69e-1	1.69e-1	4.06e-2	0.00e+0	1.29e-2	1.64e-3	2.65e-3	3.15e-3
F13	1.78e-2	8.36e-2	4.61e-2	4.78e-2	2.46e-2	1.93e-2	1.18e-1	4.97e-2	5.32e-2	1.98e-2
F14	5.35e-2	1.47e-1	1.01e-1	1.05e-1	3.59e-2	4.92e-1	8.54e-1	4.99e-1	5.15e-1	5.65e-2
F15	4.40e-1	7.65e-1	6.81e-1	6.31e-1	1.34e-1	2.53e+0	5.79e+2	3.65e+0	2.04e+1	8.36e+1
F16	1.45e+0	2.11e+0	2.02e+0	1.86e+0	2.83e-1	9.23e+0	1.48e+1	1.37e+1	1.34e+1	1.10e+0
F17	1.33e+1	1.67e+2	1.59e+2	1.31e+2	6.65e+1	4.24e+2	2.25e+3	1.60e+3	1.48e+3	4.41e+2
F18	2.34e-1	6.17e+0	1.01e+0	1.88e+0	2.41e+0	7.10e+1	2.63e+2	1.53e+2	1.57e+2	4.71e+1
F19	2.38e-1	3.60e-1	2.65e-1	2.93e-1	5.40e-2	4.03e+0	1.32e+1	7.37e+0	7.42e+0	1.97e+0
F20	8.64e-4	2.05e+0	9.29e-2	4.74e-1	8.87e-1	2.15e+0	3.39e+2	7.35e+1	1.13e+2	1.07e+2
F21	9.70e-3	1.75e+1	3.39e-1	3.80e+0	7.70e+0	2.64e+2	2.06e+4	6.45e+2	1.07e+3	2.81e+3
F22	1.03e-1	4.91e-1	2.26e-1	2.94e-1	1.83e-1	2.25e+1	6.43e+2	1.63e+2	2.08e+2	1.42e+2
F23	3.29e+2	3.29e+2	3.29e+2	3.29e+2	0.00e+0	3.15e+2	3.15e+2	3.15e+2	3.15e+2	3.44e-13
F24	1.09e+2	1.17e+2	1.12e+2	1.12e+2	3.21e+0	2.00e+2	2.39e+2	2.27e+2	2.29e+2	6.53e+0
F25	1.14e+2	1.36e+2	1.19e+2	1.23e+2	8.74e+0	2.02e+2	2.10e+2	2.03e+2	2.03e+2	1.47e+0
F26	1.00e+2	1.00e+2	1.00e+2	1.00e+2	2.08e-2	1.00e+2	2.27e+2	1.00e+2	1.07e+2	2.21e+1
F27	1.64e+0	2.91e+0	2.19e+0	2.21e+0	4.98e-1	3.00e+2	5.10e+2	3.67e+2	3.76e+2	4.95e+1
F28	3.56e+2	3.69e+2	3.60e+2	3.62e+2	6.41e+0	8.39e+2	1.44e+3	8.98e+2	9.33e+2	1.25e+2
F29	2.20e+2	2.31e+2	2.24e+2	2.24e+2	4.20e+0	1.86e+2	9.41e+6	7.22e+2	1.56e+6	3.42e+6
F30	4.83e+2	5.18e+2	5.05e+2	5.03e+2	1.41e+1	5.71e+2	6.41e+3	2.84e+3	2.92e+3	1.11e+3

Table 3
The statistical results of CETMS on the CEC2014 benchmarks for $D = 50, 100$ dimensions.

Fun	**50D**					**100D**				
	Best	Worse	Median	Mean	Std	Best	Worse	Median	Mean	Std
F1	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F2	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F3	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0
F4	0.00e+0	1.01e+2	9.81e+1	7.13e+1	4.43e+1	0.00e+0	2.37e+2	1.98e+2	1.79e+2	4.13e+1
F5	1.99e+1	2.11e+1	2.00e+1	2.01e+1	3.35e-1	1.99e+1	2.13e+1	2.00e+1	2.00e+1	3.27e-1
F6	0.00e+0	1.17e+1	4.39e+0	4.23e+0	2.60e+0	2.10e+0	2.18e+1	9.90e+0	1.01e+1	4.88e+0
F7	0.00e+0	1.04e+1	0.00e+0	4.59e-1	1.99e+0	0.00e+0	4.93e+0	0.00e+0	1.93e-1	9.66e-1
F8	1.69e+1	4.09e+1	2.38e+1	2.53e+1	5.52e+0	3.58e+1	8.38e+1	5.67e+1	5.96e+1	1.22e+1
F9	1.39e+1	4.78e+1	2.28e+1	2.45e+1	8.14e+0	4.57e+1	8.00e+1	6.26e+1	6.24e+1	8.68e+0
F10	1.12e+3	6.91e+3	3.86e+3	3.91e+3	1.26e+3	5.64e+3	1.25e+4	9.10e+3	9.15e+3	1.67e+3
F11	7.40e+2	7.77e+3	4.28e+3	4.29e+3	1.58e+3	4.68e+3	1.56e+4	1.08e+4	1.06e+4	1.89e+3
F12	5.96e-5	6.37e-3	1.42e-3	1.51e-3	1.10e-3	2.21e-4	1.96e-3	6.61e-4	7.48e-4	3.70e-4
F13	5.19e-2	2.59e-1	1.01e-1	1.11e-1	4.21e-2	1.32e-1	3.26e-1	1.94e-1	2.03e-1	4.11e-2
F14	2.65e-1	8.23e-1	5.01e-1	5.00e-1	6.68e-2	2.30e-1	3.37e-1	2.86e-1	2.84e-1	2.44e-2
F15	4.35e+0	1.61e+2	6.00e+0	1.00e+1	2.24e+1	7.58e+0	3.36e+2	1.22e+1	2.27e+1	4.76e+1
F16	1.81e+1	2.46e+1	2.32e+1	2.28e+1	1.51e+0	4.35e+1	4.86e+1	4.72e+1	4.69e+1	1.13e+0
F17	1.23e+3	3.69e+3	2.11e+3	2.16e+3	5.40e+2	3.44e+3	6.80e+3	5.35e+3	5.31e+3	6.98e+2
F18	1.50e+2	4.07e+2	2.61e+2	2.65e+2	6.47e+1	3.48e+2	8.48e+2	5.87e+2	5.97e+2	1.03e+2
F19	7.43e+0	2.20e+1	1.17e+1	1.21e+1	3.14e+0	3.31e+1	1.22e+2	1.10e+2	1.07e+2	1.62e+1
F20	9.23e+0	7.22e+2	3.64e+2	3.51e+2	1.53e+2	3.88e+2	1.01e+3	6.88e+2	7.12e+2	1.60e+2
F21	3.77e+2	2.25e+3	1.37e+3	1.36e+3	4.22e+2	2.06e+3	4.56e+3	3.16e+3	3.17e+3	5.47e+2
F22	3.45e+1	1.00e+3	3.54e+2	3.68e+2	1.90e+2	1.71e+2	1.68e+3	6.93e+2	7.68e+2	3.65e+2
F23	3.44e+2	3.44e+2	3.44e+2	3.44e+2	2.87e-13	3.48e+2	3.48e+2	3.48e+2	3.48e+2	7.66e-11
F24	2.72e+2	2.77e+2	2.75e+2	2.75e+2	1.14e+0	3.84e+2	4.25e+2	3.90e+2	3.90e+2	5.90e+0
F25	2.05e+2	2.18e+2	2.07e+2	2.08e+2	2.77e+0	2.18e+2	2.45e+2	2.27e+2	2.28e+2	6.01e+0
F26	1.00e+2	3.33e+2	1.00e+2	1.75e+2	1.02e+2	1.00e+2	3.86e+2	2.00e+2	2.09e+2	6.65e+1
F27	3.33e+2	7.22e+2	4.51e+2	4.54e+2	7.34e+1	3.59e+2	8.76e+2	5.79e+2	5.91e+2	1.07e+2
F28	1.08e+3	1.58e+3	1.27e+3	1.27e+3	8.63e+1	2.20e+3	3.42e+3	2.41e+3	2.51e+3	2.83e+2
F29	5.48e+2	4.09e+7	7.91e+2	2.26e+6	9.16e+6	7.05e+2	7.92e+2	7.28e+2	7.41e+2	2.83e+1
F30	7.96e+3	1.48e+4	9.18e+3	9.81e+3	1.73e+3	6.07e+3	1.11e+4	8.87e+3	8.80e+3	1.08e+3

Table 4
The experimental results of all algorithms on 10 dimension.

Fun	CETMS Mean$_{Std}$	L-SHADE [35] Mean$_{Std}$	POBL-ADE [11] Mean$_{Std}$	OptBees [17] Mean$_{Std}$	CMA-ES [9] Mean$_{Std}$
F1	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	1.6e+4$_{3.7e+4}$	7.8e+2$_{6.9e+2}$	**0.0e+0**$_{0.0e+0}$
F2	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	2.2e+3$_{3.2e+3}$	9.8e-3$_{6.3e-2}$	**0.0e+0**$_{0.0e+0}$
F3	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	5.7e-4$_{1.9e-3}$	9.2e-1$_{2.9e+0}$	**0.0e+0**$_{0.0e+0}$
F4	6.9e+0$_{1.5e+1}$	2.9e+1$_{1.3e+1}$	2.5e+1$_{1.4e+1}$	2.6e+0$_{2.8e+0}$	**1.5e-1**$_{7.8e-1}$
F5	1.9e+1$_{7.6e-1}$	**1.4e+1**$_{8.8e+0}$	1.9e+1$_{3.6e+1}$	1.9e+1$_{1.2e-4}$	2.0e+1$_{6.6e-1}$
F6	1.7e-1$_{4.0e-1}$	**1.8e-2**$_{1.3e-1}$	1.0e+0$_{7.8e-1}$	3.0e+0$_{1.2e+0}$	1.0e+1$_{4.1e+0}$
F7	3.1e-2$_{2.1e-2}$	**3.0e-3**$_{6.5e-3}$	1.6e-1$_{1.9e+1}$	1.5e-1$_{1.3e-1}$	1.0e-2$_{8.6e-3}$
F8	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	7.8e+0$_{3.8e+0}$	**0.0e+0**$_{0.0e+0}$	1.3e+2$_{4.6e+1}$
F9	7.3e+0$_{2.9e+0}$	**2.3e+0**$_{8.4e-1}$	7.6e+0$_{4.0e+0}$	2.0e+1$_{7.5e+0}$	1.6e+2$_{6.9e+1}$
F10	2.2e+0$_{2.9e+0}$	**8.6e-3**$_{2.2e-2}$	1.5e+3$_{1.1e+2}$	2.1e+2$_{1.0e+2}$	1.6e+3$_{4.3e+2}$
F11	3.0e+2$_{2.1e+2}$	**3.2e+1**$_{3.8e+1}$	2.0e+2$_{1.4e+2}$	3.9e+2$_{1.6e+2}$	1.6e+3$_{4.7e+2}$
F12	1.6e-1$_{4.0e-2}$	**6.8e-2**$_{1.9e-2}$	2.6e-1$_{5.8e-2}$	1.3e-1$_{7.8e-2}$	2.1e-1$_{2.9e-1}$
F13	**4.7e-2**$_{2.4e-2}$	5.2e-2$_{1.5e-2}$	1.3e-1$_{4.6e-2}$	4.1e-1$_{1.8e-1}$	1.3e-1$_{6.8e-2}$
F14	1.0e-1$_{3.5e-2}$	**8.1e-2**$_{2.6e-2}$	2.6e-1$_{1.2e-1}$	3.6e-1$_{1.9e-1}$	4.4e-1$_{6.5e-2}$
F15	6.3e-1$_{1.3e-1}$	**3.7e-1**$_{6.9e-2}$	7.1e-1$_{2.4e-1}$	2.4e+0$_{1.2e+0}$	1.1e+0$_{4.1e-1}$
F16	1.8e+0$_{2.8e-1}$	**1.2e+0**$_{3.0e-1}$	1.4e+0$_{5.2e-1}$	2.6e+0$_{3.9e-1}$	4.6e+0$_{2.9e-1}$
F17	1.3e+2$_{6.6e+1}$	**9.8e-1**$_{1.1e+0}$	2.5e+2$_{1.6e+2}$	6.8e+2$_{9.4e+2}$	6.2e+2$_{2.8e+2}$
F18	1.8e+0$_{2.4e+0}$	**2.4e-1**$_{3.1e-1}$	3.3e+1$_{3.3e+1}$	3.3e+1$_{4.5e+1}$	1.0e+2$_{8.6e+1}$
F19	2.9e-1$_{5.4e-2}$	**7.7e-2**$_{6.4e-2}$	2.0e+0$_{1.0e+0}$	9.3e-1$_{3.7e-1}$	3.2e+0$_{1.1e+0}$
F20	4.7e-1$_{8.8e-1}$	**1.8e-1**$_{1.8e-1}$	1.2e+1$_{1.1e+1}$	8.9e+0$_{2.2e+1}$	1.1e+2$_{1.0e+2}$
F21	3.8e+0$_{7.7e+0}$	**4.1e-1**$_{3.1e-1}$	1.0e+2$_{1.1e+2}$	5.7e+1$_{6.2e+1}$	5.7e+2$_{5.7e+2}$
F22	2.9e-1$_{1.8e-1}$	**4.4e-2**$_{2.8e-2}$	3.0e+1$_{3.3e+1}$	1.7e+1$_{7.4e+0}$	2.7e+2$_{2.0e+2}$
F23	3.2e+2$_{0.0e+0}$	3.3e+2$_{0.0e+0}$	3.2e+2$_{2.6e-4}$	**2.7e+2**$_{1.0e+2}$	3.2e+2$_{3.3e-13}$
F24	**1.1e+2**$_{3.2e+0}$	1.1e+2$_{1.1e+2}$	1.2e+2$_{2.4e+1}$	1.3e+2$_{9.9e+0}$	2.4e+2$_{2.2e+2}$
F25	**1.2e+2**$_{8.7e+0}$	1.3e+2$_{4.0e+1}$	1.8e+2$_{2.6e+1}$	1.4e+2$_{1.3e+1}$	2.0e+2$_{4.7e+0}$
F26	**1.0e+2**$_{2.0e-2}$	**1.0e+2**$_{1.6e-2}$	1.0e+2$_{4.9e-2}$	1.0e+2$_{1.9e-1}$	1.8e+2$_{1.0e+2}$
F27	**2.2e+0**$_{4.9e-1}$	5.8e+1$_{1.3e+2}$	2.5e+2$_{1.6e+2}$	7.4e+0$_{2.4e+0}$	3.8e+2$_{7.8e+1}$
F28	3.6e+2$_{6.4e+0}$	3.8e+2$_{3.2e+1}$	4.2e+2$_{5.5e+1}$	**3.0e+2**$_{1.9e-1}$	1.5e+3$_{1.4e+3}$
F29	2.2e+2$_{4.2e+0}$	2.2e+2$_{4.6e-1}$	3.5e+5$_{9.4e+5}$	2.1e+2$_{2.2e+1}$	**2.0e+2**$_{1.3e+1}$
F30	5.0e+2$_{1.4e+1}$	4.6e+2$_{1.3e+1}$	6.3e+2$_{1.3e+2}$	**3.8e+2**$_{7.9e+1}$	4.7e+2$_{1.9e+2}$

Table 5
The experimental results of all algorithms on 30 dimension.

Fun	CETMS Mean$_{Std}$	L-SHADE [35] Mean$_{Std}$	POBL-ADE [11] Mean$_{Std}$	OptBees [17] Mean$_{Std}$	CMA-ES [9] Mean$_{Std}$
F1	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	1.6e+4$_{1.2e+4}$	8.5e+4$_{3.0e+5}$	**0.0e+0**$_{0.0e+0}$
F2	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	3.1e+2$_{7.5e+2}$	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$
F3	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	8.4e-3$_{3.7e-2}$	5.3e+4$_{1.6e+5}$
F4	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	6.3e+1$_{2.6e+1}$	1.2e+1$_{1.3e+1}$	3.1e-1$_{1.0e+0}$
F5	**2.0e+1**$_{3.4e-1}$	**2.0e+1**$_{3.7e-2}$	**2.0e+1**$_{5.1e-2}$	**2.0e+1**$_{1.0e-5}$	2.0e+1$_{7.2e-1}$
F6	1.8e+0$_{1.7e+0}$	**1.4e-7**$_{9.9e-7}$	5.1e+0$_{1.6e+0}$	1.6e+1$_{3.4e+0}$	3.2e+1$_{8.3e+0}$
F7	6.5e-1$_{2.6e+0}$	**0.0e+0**$_{0.0e+0}$	2.3e-2$_{2.3e-2}$	3.7e-2$_{3.8e-2}$	1.2e-3$_{3.4e-3}$
F8	1.3e+1$_{3.3e+0}$	**0.0e+0**$_{0.0e+0}$	5.5e+1$_{1.1e+1}$	0.0e+0$_{0.0e+0}$	4.4e+2$_{8.2e+1}$
F9	1.2e+1$_{5.2e+0}$	**6.8e+0**$_{1.5e+0}$	8.4e+1$_{9.0e+0}$	1.3e+2$_{3.2e+1}$	6.5e+2$_{1.5e+2}$
F10	2.0e+3$_{1.2e+3}$	**1.6e-2**$_{1.6e-2}$	2.1e+3$_{4.9e+2}$	1.0e+3$_{2.5e+2}$	5.1e+3$_{6.2e+2}$
F11	2.5e+3$_{1.0e+3}$	**1.2e+3**$_{1.8e+2}$	3.8e+3$_{3.5e+2}$	2.7e+3$_{5.6e+2}$	5.0e+3$_{7.6e+2}$
F12	**2.6e-3**$_{3.1e-3}$	1.6e-1$_{2.3e-2}$	9.5e-1$_{1.3e-1}$	1.8e-1$_{6.1e-2}$	4.6e-2$_{2.9e-2}$
F13	**5.3e-2**$_{1.9e-2}$	1.2e-1$_{1.7e-2}$	2.8e-1$_{6.1e-2}$	5.6e-1$_{1.4e-1}$	3.3e-1$_{2.0e-1}$
F14	5.1e-1$_{5.6e-2}$	2.4e-1$_{3.0e-2}$	**2.2e-1**$_{4.2e-2}$	3.9e-1$_{2.3e-1}$	5.3e-1$_{2.2e-1}$
F15	2.0e+1$_{8.3e+1}$	**2.1e+0**$_{2.5e-1}$	7.7e+0$_{1.0e+0}$	1.2e+1$_{6.9e+0}$	3.6e+0$_{9.1e-1}$
F16	1.3e+1$_{1.1e+0}$	8.5e+0$_{4.6e-1}$	**1.0e+1**$_{4.5e-1}$	**1.0e+1**$_{6.9e-1}$	1.4e+1$_{4.2e-1}$
F17	1.4e+3$_{4.4e+2}$	**1.9e+2**$_{7.5e+1}$	1.1e+3$_{4.1e+2}$	2.7e+4$_{4.0e+4}$	2.0e+3$_{4.7e+2}$
F18	1.5e+2$_{4.7e+1}$	**5.9e+0**$_{2.9e+0}$	1.1e+2$_{3.8e+1}$	1.9e+2$_{4.7e+2}$	2.8e+2$_{1.3e+2}$
F19	7.4e+0$_{1.9e+0}$	**3.7e+0**$_{6.8e-1}$	8.8e+0$_{1.2e+1}$	7.8e+0$_{1.8e+0}$	9.4e+0$_{1.9e+0}$
F20	1.1e+2$_{1.0e+2}$	**3.1e+0**$_{1.5e+0}$	3.8e+1$_{2.2e+1}$	8.5e+2$_{7.7e+2}$	2.0e+4$_{9.5e+4}$
F21	1.0e+3$_{2.8e+3}$	**8.7e+1**$_{9.0e+1}$	3.8e+2$_{1.9e+2}$	1.7e+4$_{1.8e+4}$	1.0e+3$_{3.1e+2}$
F22	2.0e+2$_{1.4e+2}$	**2.8e+1**$_{1.8e+1}$	2.3e+2$_{8.1e+1}$	2.3e+2$_{9.2e+1}$	4.2e+2$_{2.3e+2}$
F23	**3.1e+2**$_{3.4e-13}$	3.2e+2$_{0.0e+0}$	3.1e+2$_{1.1e-7}$	3.1e+2$_{6.6e-2}$	3.1e+2$_{3.5e-13}$
F24	**2.2e+2**$_{6.5e+0}$	**2.2e+2**$_{1.1e+0}$	**2.2e+2**$_{7.4e+0}$	2.3e+2$_{5.4e+0}$	3.1e+2$_{2.4e+2}$
F25	**2.0e+2**$_{1.4e+0}$	**2.0e+2**$_{5.0e-2}$	**2.0e+2**$_{3.2e+0}$	**2.0e+2**$_{1.6e-1}$	**2.0e+2**$_{2.3e-3}$
F26	**1.0e+2**$_{2.2e+1}$	**1.0e+2**$_{1.6e-2}$	1.3e+2$_{4.9e+1}$	**1.0e+2**$_{1.7e-1}$	1.1e+2$_{5.9e+1}$
F27	3.7e+2$_{4.9e+1}$	**3.0e+2**$_{0.0e+0}$	4.2e+2$_{4.6e+1}$	4.0e+2$_{9.7e-1}$	4.5e+2$_{1.2e+2}$
F28	9.3e+2$_{1.2e+2}$	8.4e+2$_{1.4e+1}$	9.1e+2$_{1.6e+2}$	**4.3e+2**$_{1.5e+1}$	4.6e+3$_{3.2e+3}$
F29	1.5e+6$_{3.4e+6}$	7.2e+2$_{5.1e+0}$	3.3e+5$_{2.4e+6}$	2.1e+2$_{1.1e+0}$	**2.0e+2**$_{1.7e+0}$
F30	2.9e+3$_{1.1e+3}$	1.2e+3$_{6.2e+2}$	1.2e+3$_{5.1e+2}$	**5.9e+2**$_{9.8e+1}$	9.5e+2$_{3.3e+2}$

Table 6
The experimental results of all algorithms on 50 dimension.

Fun	CETMS Mean$_{Std}$	L-SHADE [35] Mean$_{Std}$	POBL-ADE [11] Mean$_{Std}$	OptBees [17] Mean$_{Std}$	CMA-ES [9] Mean$_{Std}$
F1	**0.0e+0**$_{0.0e+0}$	1.2e+3$_{1.5e+3}$	6.5e+3$_{2.9e+3}$	1.3e+5$_{9.7e+4}$	**0.0e+0**$_{0.0e+0}$
F2	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	1.3e+1$_{1.4e+1}$	1.0e-3$_{2.9e-3}$	**0.0e+0**$_{0.0e+0}$
F3	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	5.2e-1$_{5.4e-1}$	1.7e+2$_{1.7e+2}$	1.2e+5$_{3.7e+5}$
F4	7.1e+1$_{4.4e+1}$	5.9e+1$_{4.6e+1}$	1.0e+2$_{3.9e+1}$	3.8e+1$_{3.4e+1}$	**6.2e-1**$_{1.4e+0}$
F5	**2.0e+1**$_{3.3e-1}$	**2.0e+1**$_{4.6e-2}$	2.1e+1$_{4.1e-2}$	**2.0e+1**$_{2.5e-5}$	2.1e+1$_{6.4e-1}$
F6	4.2e+0$_{2.6e+0}$	**2.6e-1**$_{5.2e-1}$	1.9e+1$_{3.6e+0}$	3.0e+1$_{4.5e+0}$	5.9e+1$_{9.0e+0}$
F7	4.5e-1$_{1.9e+0}$	**0.0e+0**$_{0.0e+0}$	2.8e-2$_{4.1e-2}$	2.3e-2$_{3.9e-2}$	3.8e-4$_{1.9e-3}$
F8	2.5e+1$_{5.5e+0}$	**0.0e+0**$_{0.0e+0}$	1.4e-2$_{4.1e-3}$	**0.0e+0**$_{0.0e+0}$	7.3e+2$_{1.4e+2}$
F9	2.4e+1$_{8.1e+0}$	**1.1e+1**$_{2.1e+0}$	2.1e+2$_{2.9e+1}$	2.4e+2$_{4.8e+1}$	1.1e+3$_{2.1e+2}$
F10	3.9e+3$_{1.2e+3}$	**1.2e-1**$_{4.1e-2}$	4.9e+2$_{1.5e+2}$	1.8e+3$_{3.4e+2}$	8.4e+3$_{1.0e+3}$
F11	4.2e+3$_{1.5e+3}$	**3.2e+3**$_{3.3e+2}$	7.2e+3$_{6.0e+2}$	5.1e+3$_{6.8e+2}$	8.2e+3$_{9.0e+2}$
F12	**1.5e-3**$_{1.1e-3}$	2.2e-1$_{2.8e-2}$	1.8e+0$_{2.0e-1}$	1.6e-1$_{4.6e-2}$	3.1e-2$_{1.3e-2}$
F13	**1.1e-1**$_{4.2e-2}$	1.6e-1$_{1.8e-2}$	4.2e-1$_{5.6e-2}$	6.0e-1$_{1.1e-1}$	6.7e-1$_{2.7e-1}$
F14	5.0e-1$_{6.6e-2}$	3.0e-1$_{2.5e-2}$	**2.8e-1**$_{3.3e-2}$	4.7e-1$_{2.3e-1}$	8.9e-1$_{5.4e-1}$
F15	1.0e+1$_{2.2e+1}$	**5.2e+0**$_{5.1e-1}$	2.3e+1$_{3.0e+0}$	2.6e+1$_{9.1e+0}$	6.3e+0$_{1.3e+0}$
F16	2.2e+1$_{1.5e+0}$	**1.7e+1**$_{4.8e-1}$	2.0e+1$_{4.1e-1}$	1.8e+1$_{8.7e-1}$	2.3e+1$_{5.2e-1}$
F17	2.1e+3$_{5.4e+2}$	**1.4e+3**$_{5.1e+2}$	3.0e+3$_{1.2e+3}$	3.2e+4$_{3.5e+4}$	3.1e+3$_{5.7e+2}$
F18	2.6e+2$_{6.4e+1}$	**9.7e+1**$_{1.4e+1}$	4.1e+2$_{8.3e+2}$	6.8e+2$_{1.0e+3}$	4.9e+2$_{1.4e+2}$
F19	1.2e+1$_{3.1e+0}$	**8.3e+0**$_{1.8e+0}$	1.6e+1$_{3.2e+0}$	1.5e+1$_{2.7e+0}$	1.5e+1$_{2.5e+0}$
F20	3.5e+2$_{1.5e+2}$	**1.4e+1**$_{4.6e+0}$	1.8e+2$_{4.2e+1}$	1.9e+3$_{1.4e+3}$	1.9e+5$_{6.4e+5}$
F21	1.3e+3$_{4.2e+2}$	**5.2e+2**$_{1.5e+2}$	1.4e+3$_{2.8e+2}$	6.0e+4$_{4.1e+4}$	1.8e+3$_{4.4e+2}$
F22	3.6e+2$_{1.9e+2}$	**1.1e+2**$_{7.5e+1}$	4.8e+2$_{1.3e+2}$	7.5e+2$_{2.0e+2}$	7.9e+2$_{3.0e+2}$
F23	3.4e+2$_{2.8e-13}$	3.4e+2$_{4.4e-13}$	3.4e+2$_{1.1e-3}$	**3.3e+2**$_{2.8e-1}$	**3.3e+2**$_{6.2e-13}$
F24	2.7e+2$_{1.1e+0}$	2.8e+2$_{6.6e-1}$	**2.5e+2**$_{6.4e+0}$	2.6e+2$_{3.2e+0}$	3.3e+2$_{2.2e+2}$
F25	**2.0e+2**$_{2.7e+0}$	2.1e+2$_{3.6e-1}$	**2.0e+2**$_{1.4e-5}$	**2.0e+2**$_{6.0e-1}$	**2.0e+2**$_{3.3e-2}$
F26	1.7e+2$_{1.0e+2}$	1.0e+2$_{1.4e+1}$	1.8e+2$_{3.4e+1}$	**1.0e+2**$_{1.2e-1}$	**1.0e+2**$_{1.8e-1}$
F27	4.5e+2$_{7.3e+1}$	**3.3e+2**$_{3.0e+1}$	8.5e+2$_{9.6e+1}$	1.1e+3$_{2.1e+2}$	6.0e+2$_{1.6e+2}$
F28	1.2e+3$_{8.6e+1}$	**1.1e+3**$_{2.9e+1}$	2.2e+3$_{6.5e+2}$	4.4e+2$_{3.8e+1}$	5.0e+3$_{6.0e+3}$
F29	2.2e+6$_{9.1e+6}$	7.9e+2$_{2.4e+1}$	2.5e+5$_{1.2e+6}$	2.3e+2$_{2.3e+0}$	**2.1e+2**$_{2.6e+0}$
F30	9.8e+3$_{1.7e+3}$	8.7e+3$_{4.1e+2}$	1.1e+4$_{2.3e+3}$	**8.1e+2**$_{1.2e+2}$	1.4e+3$_{3.2e+2}$

Table 7
The experimental results of all algorithms on 100 dimension.

Fun	CETMS Mean$_{Std}$	L-SHADE [35] Mean$_{Std}$	POBL-ADE [11] Mean$_{Std}$	OptBees [17] Mean$_{Std}$	CMA-ES [9] Mean$_{Std}$
F1	**0.0e+0**$_{0.0e+0}$	1.7e+5$_{5.7e+4}$	3.0e+3$_{8.1e+2}$	2.9e+5$_{1.0e+5}$	**0.0e+0**$_{0.0e+0}$
F2	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	1.5e+1$_{1.1e+1}$	1.0e+1$_{2.9e+1}$	**0.0e+0**$_{0.0e+0}$
F3	**0.0e+0**$_{0.0e+0}$	**0.0e+0**$_{0.0e+0}$	1.7e+1$_{1.1e+1}$	7.5e+2$_{7.8e+2}$	2.2e+5$_{7.7e+5}$
F4	1.7e+2$_{4.1e+1}$	1.7e+2$_{3.1e+1}$	3.2e+2$_{4.6e+1}$	1.4e+2$_{5.4e+1}$	**8.5e-1**$_{1.6e+0}$
F5	**2.0e+1**$_{3.2e-1}$	2.1e+1$_{3.1e-2}$	2.1e+1$_{2.4e-2}$	**2.0e+1**$_{2.4e-5}$	2.1e+1$_{5.1e-1}$
F6	1.0e+1$_{4.8e+0}$	**8.7e+0**$_{2.3e+0}$	8.6e+1$_{6.5e+0}$	7.1e+1$_{7.7e+0}$	1.2e+2$_{1.1e+1}$
F7	1.9e-1$_{9.6e-1}$	**0.0e+0**$_{0.0e+0}$	1.9e-2$_{3.0e-2}$	5.8e-3$_{8.0e-3}$	2.9e-4$_{1.4e-3}$
F8	5.9e+1$_{1.2e+1}$	1.1e-2$_{7.4e-3}$	5.0e-1$_{4.2e-2}$	**0.0e+0**$_{0.0e+0}$	1.4e+3$_{1.4e+2}$
F9	6.2e+1$_{8.6e+0}$	**3.4e+1**$_{5.0e+0}$	1.9e+2$_{2.5e+1}$	6.6e+2$_{9.7e+1}$	2.0e+3$_{3.0e+2}$
F10	9.1e+3$_{1.6e+3}$	**2.6e+1**$_{5.8e+0}$	1.3e+3$_{7.4e+1}$	4.2e+3$_{4.2e+2}$	1.6e+4$_{1.1e+3}$
F11	**1.0e+4**$_{1.8e+3}$	1.1e+4$_{5.6e+2}$	**1.0e+4**$_{1.0e+3}$	1.2e+4$_{1.1e+3}$	1.6e+4$_{1.3e+3}$
F12	**7.4e-4**$_{3.7e-4}$	4.4e-1$_{4.7e-2}$	9.4e-1$_{6.6e-2}$	2.3e-1$_{5.0e-2}$	2.2e-2$_{7.5e-3}$
F13	**2.0e-1**$_{4.1e-2}$	2.4e-1$_{2.1e-2}$	5.1e-1$_{5.0e-2}$	5.8e-1$_{8.1e-2}$	1.0e+0$_{2.9e-1}$
F14	2.8e-1$_{2.4e-2}$	**1.2e-1**$_{7.3e-3}$	**1.2e-1**$_{9.0e-3}$	2.2e-1$_{2.2e-2}$	1.5e+0$_{8.4e-1}$
F15	2.2e+1$_{4.7e+1}$	**1.6e+1**$_{1.2e+0}$	2.9e+1$_{2.9e+0}$	6.5e+1$_{1.8e+1}$	1.5e+1$_{2.2e+0}$
F16	4.6e+1$_{1.1e+0}$	**3.9e+1**$_{4.8e-1}$	4.5e+1$_{6.0e-1}$	4.0e+1$_{1.1e+0}$	4.7e+1$_{7.7e-1}$
F17	5.3e+3$_{6.9e+2}$	**4.4e+3**$_{7.1e+2}$	1.0e+4$_{3.6e+3}$	1.0e+5$_{6.7e+4}$	6.3e+3$_{7.3e+2}$
F18	5.9e+2$_{1.0e+2}$	**2.2e+2**$_{1.7e+1}$	7.9e+2$_{6.5e+2}$	1.5e+3$_{2.1e+3}$	1.0e+3$_{3.1e+2}$
F19	1.0e+2$_{1.6e+1}$	9.6e+1$_{2.3e+0}$	9.3e+1$_{2.7e+1}$	5.2e+1$_{1.5e+1}$	**3.4e+1**$_{5.5e+0}$
F20	7.1e+2$_{1.6e+2}$	**1.5e+2**$_{5.2e+1}$	3.2e+2$_{6.8e+1}$	1.0e+4$_{4.2e+3}$	4.1e+5$_{6.4e+5}$
F21	3.1e+3$_{5.4e+2}$	**2.3e+3**$_{5.3e+2}$	5.0e+3$_{2.6e+3}$	3.1e+5$_{1.5e+5}$	3.6e+3$_{5.6e+2}$
F22	**7.6e+2**$_{3.6e+2}$	1.1e+3$_{1.9e+2}$	1.4e+3$_{3.7e+2}$	2.0e+3$_{3.3e+2}$	1.3e+3$_{4.6e+2}$
F23	3.4e+2$_{7.6e-11}$	3.5e+2$_{2.8e-13}$	**2.2e+2**$_{5.4e+1}$	3.4e+2$_{9.2e-1}$	3.4e+2$_{2.5e-12}$
F24	3.9e+2$_{5.9e+0}$	3.9e+2$_{2.9e+0}$	**2.6e+2**$_{5.4e+1}$	3.4e+2$_{1.0e+1}$	5.7e+2$_{5.5e+2}$
F25	**2.2e+2**$_{6.0e+0}$	**2.0e+2**$_{4.0e-13}$	2.0e+2$_{6.6e-4}$	2.0e+2$_{1.1e+0}$	**2.0e+2**$_{4.9e-3}$
F26	2.0e+2$_{6.6e+1}$	2.0e+2$_{6.2e-13}$	2.0e+2$_{1.6e-2}$	**1.0e+2**$_{6.8e-2}$	1.9e+2$_{2.9e+1}$
F27	5.9e+2$_{1.0e+2}$	**3.8e+2**$_{3.3e+1}$	1.9e+3$_{2.0e+2}$	2.1e+3$_{1.6e+2}$	1.0e+3$_{2.3e+2}$
F28	2.5e+3$_{2.8e+2}$	2.3e+3$_{4.6e+1}$	7.4e+3$_{1.3e+3}$	**6.1e+2**$_{4.0e+1}$	2.6e+4$_{1.2e+4}$
F29	7.4e+2$_{2.8e+1}$	8.0e+2$_{7.6e+1}$	1.2e+4$_{3.2e+4}$	2.7e+2$_{3.2e+0}$	**2.3e+2**$_{4.2e+0}$
F30	8.8e+3$_{1.0e+3}$	8.3e+3$_{9.6e+2}$	7.8e+3$_{1.6e+3}$	2.8e+3$_{2.4e+0}$	**2.6e+3**$_{4.1e+2}$

On the second multimodal group, the proposed CETMS can find the best results on F_8, F_{13}. L-SHADE obtains the good results except $F4$, F_{13}. OptBees can execute the best results on F_8. CMA-ES can get the best results on F_4. On the third hybrid group, L-SHADE gets the best results on all test functions. On the last composition group, the proposed CETMS from F_{24} to F_{27} attains the best results. OptBees gets the best results from F_{23}, F_{28}, F_{30}. CMA-ES can obtain the best results on F_{29}. By observing the simulation results of Table 4, the proposed CETMS not only get some best results on 10-dimensional problems, but attain the suboptimal solutions on the rest of these problems. Compared with CMA-ES, the proposed CETMS has not good results on F_4, F_7, F_{29}.

As seen in Table 5, the statistical results of all experimental algorithms on 30-dimensional problems are calculated. On the first unimodal group, the proposed CETMS and L-SHADE can obtain the best Mean and Std. However, POBL-ADE only can get the best results on F_3. OptBees only can get the best results on F_2. CMA-ES can find the best results on F_1, F_2. On the second multimodal group, the proposed CETMS can find the best results on F_4, F_5, F_{12}, F_{13}. L-SHADE obtains the good results from $F4$ to F_{11}. POBL-ADE can find the best results on $F5$, F_{14}, $F_{15}F_{16}$. OptBees can execute the best results on F_5, F_{16}. On the third hybrid group, L-SHADE gets the best results on all test functions. On the last composition group, the proposed CETMS from F_{23} to F_{26} attains the best results. L-SHADE attains the best results from F_{24} to F_{27}. POBL-ADE can find the best results on F_{24}, F_{25}. OptBees gets the best results from F_{25}, F_{26}, F_{28}, F_{30}. CMA-ES gets the best results on F_{25}, F_{29}. By observing the simulation results of Table 5, the proposed CETMS can attain some best results on 30-dimensional problems. CMA-ES is better than the proposed CETMS on F_7, F_{15}, F_{25}, F_{29}, F_{30}. The solving performance of OptBees and POBL-ADE get the improvement on 30-dimensional problems.

Table 6 shows the experimental results of all algorithms on 50-dimensional problems. On the first unimodal group, the proposed CETMS can obtain the best Mean and Std. However, L-SHADE gets the best results on F_2, F_3. POBL-ADE and OptBees cannot get the best results on this group. CMA-ES can get the best results on F_1, F_2. On the second multimodal group, the proposed CETMS can find the best results on F_5, F_{12}, F_{13}. L-SHADE obtains the good results from F_5 to F_{11}, and F_{15}, F_{16}. POBL-ADE only get the best result on F_{14}. OptBees can execute the best results on F_5, F_8. CMA-ES can obtain the best results on F_4. On the third hybrid group, L-SHADE gets the best results on all test functions. On the last composition group, the proposed CETMS on F_{25} attains the best results. L-SHADE gets the best results on F_{27}, F_{28}. POBL-ADE gets the best results on F_{24}, F_{25}. OptBees gets the best results on F_{23}, F_{25}, F_{26}, F_{30}. CMA-ES gets the best results on F_{23}, F_{25}, F_{26}, F_{29}. By observing the simulation results of Table 6, the proposed CETMS can keep the stable solving performance. Compared with CMA-ES, the proposed algorithm has not good results on F_4, F_7, F_{15}, F_{23}, F_{26}, F_{29}, F_{30}.

The last experimental simulation is executed on 100 dimensional problems, the experimental results of all algorithms are shown in the Table 7. On the first unimodal group, the proposed CETMS can obtain the best results on all unimodal functions. L-SHADE can obtain the best results on F_2, F_3. However, POBL-ADE and OptBees cannot find the best results on these three problems. On the second multimodal group, the proposed CETMS can find the best results on F_5, F_{11}, F_{12}, F_{13}. L-SHADE obtains the good results on F_6, F_9, F_{10}, F_{14}, F_{15}, F_{16}. POBL-ADE gets the best results on F_{11}, F_{14}. OptBees can execute the best results on F_5, F_8. CMA-ES attains the best results on F_4. On the third hybrid group, the proposed CETMS can find the best results on F_{22}. L-SHADE get the best results on F_{17}, F_{18}, F_{20}, F_{21}. OptBees cannot execute the best results on this group. CMA-ES gets the best results on F_{19}. On the last composition group, the proposed CETMS cannot attain the best result on this group. L-SHADE obtains the good results from F_{25}, F_{27}. POBL-ADE gets the best results on F_{23}, F_{24}. OptBees gets the best results F_{26}, F_{28}. CMA-ES can find the best results on F_{25}, F_{29}, F_{30}. By observing the simulation results of Table 7, the proposed CETMS not only get some best results on 100-dimensional problems, but attain the suboptimal solutions on the rest of these problems. CMA-ES is better than the proposed CETMS on F_4, F_7, F_{15}, F_{19}, F_{25}, F_{26}, F_{29}, F_{30}.

In summary, the proposed CETMS attains the best results from low dimension to high dimension over the most of CEC2014 benchmarks suit. Even though CETMS cannot get the best results on some test problems, it can get the suboptimal results compared with all experimental algorithms. The proposed CETMS is superior to CMA-ES on the most benchmark problems with different dimension. Compared with other methods, CETMS can get the best results on the unimodal test problems with the different dimensions. On the rest of test problems, the proposed algorithm still keeps the stable solving performance. The stable solving performance is due to the introduction of tissue membrane system. More specifically, multiple cells can get more opportunities for find the global optimal solution. Moreover, after the certain generation, the optimal solutions from different cells can be exchanged. It is useful to the share information among multiple cells. Therefore, the appropriate combination of tissue membrane system and CMA-ES can produce a better capability to balance exploration and exploitation, which are two contradictory factors directly related to the performance of an optimization algorithm. The proposed CETMS is capable of handling these problems effectively.

5. Conclusions

This paper presents a new hybrid algorithm called CETMS, which incorporates with CMA-ES in Tissue Membrane Systems. In CETMS, each cell has a CMA-ES algorithm to find the approximate optimal solution. The superiority of the structure is that multiple cells can get more opportunities for find the global optimal solution. Moveover, after some generation executed in each cell, the optimal solutions from different cells are sent to the common environment, which achieve a tradeoff between exploration and exploitation. The optimal solution in the common environment is sent to the different cells, which can accelerate finding the optimal solution. In order to verify the performance of CETMS, a set of numerical benchmark functions are tested in the experiments.

Compared to L-SHADE, POBL-ADE, and OptBees, CETMS achieves more accurate solutions on the majority of test functions. Although L-SHADE performs better than CETMS, CETMS is simpler and has less control parameters to be tuned. Moreover, both CETMS and L-SHADE obtain similar performance on most of test functions except hybrid problems. In addition, CETMS is competitive to POBL-ADE, and OptBees on most of the test functions. Experimental results confirm that the structure of tissue membrane system is useful to improve the solving performance of CETMS. It helps to switch the search behaviors of cells and balance exploration and exploitation. By the ensemble of multi-CMA-ES, CETMS obtains a higher probability to improve the quality of candidate solutions.

For future research, many variant versions of CETMS can be made by redefining new rules, or introducing new membrane structure. Moreover, CETMS will be extended for solving constrained, and multiobjective optimization problems. Finally, CETMS will be applied to solve some problems in machine learning, such as clustering and classification.

References

[1] C. Buiu, C. Vasile, O. Arsene, Development of membrane controllers for mobile robots, Inf. Sci. 187 (2012) 33–51.

[2] J.M. Cecilia, J.M. Garcia, G.D. Guerrero, M.A. Martinez-del Amor, M.J. Perez-Jimenez, M. Ujaldon, The gpu on the simulation of cellular computing models, Soft Comput. 16 (2) (2012) 231–246.
[3] S. Chu, P. Tsai, J. Pan, Cat swarm optimization, in: PRICAI 2006: Trends in Artificial Intelligence, Springer, 2006, pp. 854–858.
[4] G. Ciobanu, G. Paun, M.J. Paerez-Jimaenez, Applications of Membrane Computing, Springer, 2006.
[5] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, Comput. Intell. Mag. IEEE 1 (4) (2006) 28–39.
[6] J. Gu, X. Gu, M. Gu, A novel parallel quantum genetic algorithm for stochastic job shop scheduling, J. Math. Anal. Appl. 355 (1) (2009) 63–81.
[7] M. Han, C. Liu, Endpoint prediction model for basic oxygen furnace steel-making based on membrane algorithm evolving extreme learning machine, Appl. Soft Comput. 19 (2014) 430–437.
[8] M. Han, C. Liu, J. Xing, An evolutionary membrane algorithm for global numerical optimization problems, Inf. Sci. 276 (2014) 219–241.
[9] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[10] J. He, J. Xiao, Z. Shao, An adaptive membrane algorithm for solving combinatorial optimization problems, Acta Math. Sci. 34 (5) (2014) 1377–1394.
[11] Z. Hu, Y. Bao, T. Xiong, Partial opposition-based adaptive differential evolution algorithms: Evaluation on the cec 2014 benchmark set for real-parameter optimization, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 2259–2265.
[12] L. Huang, I.H. Suh, A. Abraham, Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants, Inf. Sci. 181 (11) (2011) 2370–2391.
[13] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[14] J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore, 2013.
[15] C. Liu, M. Han, X. Wang, A multi-objective evolutionary algorithm based on membrane systems, in: Advanced Computational Intelligence (IWACI), 2011 Fourth International Workshop on, IEEE, 2011, pp. 103–109.
[16] C. Liu, M. Han, X. Wang, A novel evolutionary membrane algorithm for global numerical optimization, in: Intelligent Control and Information Processing (ICICIP), 2012 Third International Conference on, IEEE, 2012, pp. 727–732.
[17] R.D. Maia, L.N. de Castro, W.M. Caminhas, Real-parameter optimization with optbees, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 2649–2655.
[18] C. Martin-Vide, G. Paun, J. Pazos, A. Rodriguez-Paton, Tissue P systems, Theor. Comput. Sci. 296 (2) (2003) 295–326.
[19] T.Y. Nishida, An approximate algorithm for np-complete optimization problems exploiting p systems, in: Proc. Brainstorming Workshop on Uncertainty in Membrane Computing, 2004, pp. 185–192.
[20] Y. Niu, S. Wang, J. He, J. Xiao, A novel membrane algorithm for capacitated vehicle routing problem, Soft Comput. 19 (2) (2015) 471–482.
[21] L. Pan, C. Martin-Vide, Solving multidimensional 0–1 knapsack problem by p systems with input and active membranes, J. Parallel Distrib. Comput. 65 (12) (2005) 1578–1584.
[22] L. Pan, M.J. Pérez-Jiménez, Computational complexity of tissue-like p systems, J. Complexity 26 (3) (2010) 296–315.
[23] G. Paun, Computing with membranes, J. Comput. Syst. Sci. 61 (1) (2000) 108–143.
[24] G. Paun, G. Rozenberg, A guide to membrane computing, Theor. Comput. Sci. 287 (1) (2002) 73–100.
[25] G. Paun, G. Rozenberg, A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, Inc., UK, 2010.
[26] H. Peng, J. Wang, M.J. Pérez-Jiménez, Optimal multi-level thresholding with membrane computing, Digital Signal Process. 37 (2015) 53–64.
[27] H. Peng, J. Wang, M.J. Pérez-Jiménez, A. Riscos-Núñez, An unsupervised learning algorithm for membrane computing, Inf. Sci. 304 (2015) 80–91.
[28] H. Peng, J. Wang, M.J. Pérez-Jiménez, P. Shi, A novel image thresholding method based on membrane computing and fuzzy entropy, J. Intell. Fuzzy Syst. 24 (2) (2013) 229–237.
[29] H. Peng, J. Wang, P. Shi, M.J. Pérez-Jiménez, A. RISCOS-NÚÑEZ, An extended membrane system with active membranes to solve automatic fuzzy clustering problems, Int. J. Neural Syst. (2015), doi:10.1142/S0129065716500040.
[30] H. Peng, J. Wang, P. Shi, A. Riscos-Núñez, M.J. Pérez-Jiménez, An automatic clustering algorithm inspired by membrane computing, Pattern Recognit. Lett. 68 (2015) 34–40.
[31] G. Singh, K. Deep, Hybridization of p systems and particle swarm optimization for function optimization, in: Proceedings of the Third International Conference on Soft Computing for Problem Solving, Springer, 2014, pp. 395–401.
[32] G. Singh, K. Deep, A.K. Nagar, Cell-like p-systems based on rules of particle swarm optimization, Appl. Math. Comput. 246 (2014) 546–560.
[33] B. Song, M.J. Pérez-Jiménez, L. Pan, Efficient solutions to hard computational problems by p systems with symport/antiport rules and membrane division, Biosystems 130 (2015) 51–58.
[34] B. Song, T. Song, L. Pan, A time-free uniform solution to subset sum problem by tissue p systems with cell division, Math. Struct. Comput. Sci. (2015) 1–16, doi:10.1017/S0960129515000018.
[35] R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, IEEE, 2014, pp. 1658–1665.
[36] J. Xiao, Y. Huang, Z. Cheng, J. He, Y. Niu, A hybrid membrane evolutionary algorithm for solving constrained optimization problems, Opt. Int. J. Light Electron Opt. 125 (2) (2014) 897–902.
[37] J. Xiao, Y. Jiang, J. He, Z. Cheng, A dynamic membrane evolutionary algorithm for solving dna sequences design with minimum free energy, MATCH Commun. Math. Comput. Chem. 70 (3) (2013) 971–986.
[38] J. Xiao, B. Liu, Y. Huang, Z. Cheng, An adaptive quantum swarm evolutionary algorithm for partner selection in virtual enterprise, Int. J. Prod. Res. 52 (6) (2014) 1607–1621.
[39] J. Xiao, X. Zhang, J. Xu, A membrane evolutionary algorithm for dna sequence design in dna computing, Chinese Sci. Bull. 57 (6) (2012) 698–706.
[40] X. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio Insp. Comput. 2 (2) (2010) 78–84.
[41] G. Zhang, M. Gheorghe, L. Pan, M.J. Perez-Jimenez, Evolutionary membrane computing: a comprehensive survey and new results, Inf. Sci. 279 (2014) 528–551.
[42] G. Zhang, M. Gheorghe, C. Wu, A quantum-inspired evolutionary algorithm based on p systems for knapsack problem, Fund. Inf. 87 (1) (2008) 93–116.
[43] G. Zhang, C. Liu, M. Gheorghe, Diversity and convergence analysis of membrane algorithms, in: Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on, IEEE, 2010, pp. 596–603.
[44] G. Zhang, H. Rong, F. Neri, M.J. Perez-Jimenez, An optimization spiking neural p system for approximately solving combinatorial optimization problems, Int. J. Neural Syst. 24 (05) (2014) 1440006.
[45] X. Zhang, Y. Liu, B. Luo, L. Pan, Computational power of tissue p systems for generating control languages, Inf. Sci. 278 (2014) 285–297.
[46] X. Zhang, B. Luo, L. Pan, Small universal tissue p systems with symport/antiport rules, Int. J. Comput. Commun. Control 7 (1) (2012) 173–183.
[47] X. Zhang, S. Wang, Y. Niu, L. Pan, Tissue p systems with cell separation: attacking the partition problem, Sci. China Inf. Sci. 54 (2) (2011) 293–304.
[48] Y. Zhang, L. Huang, A variant of p systems for optimization, Neurocomputing 72 (4) (2009) 1355–1360.
[49] J. Zhao, N. Wang, A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling, Comput. Chem. Eng. 35 (2) (2011) 272–283.
[50] J. Zhao, N. Wang, Hybrid optimization method based on membrane computing, Ind. Eng. Chem. Res. 50 (3) (2011) 1691–1704.