An Extended Mind Evolutionary Computation Model for Optimizations

Jing Jie 1,2, Chongzhao Han 1, Jianchao Zeng 2

1 School of Electronic and Information Engineering, Xi’an Jiaotong University
710049, Xi’an City, China
Jjing277@sohu.com
2 Division of System Simulation & Computer Application, Taiyuan University of Science & Technology
030024, Taiyuan City, China

Abstract. The paper makes an analysis on the simulated mechanisms of mind evolutionary computation (MEC) firstly and proposes an extended computation model for MEC (EMEC). EMEC manipulates the search based on the behavior space and the information space. All operations in the behavior space are processed based on groups that symbolize the solution area, while the operations in the information space are done based on the billboards that are used to record the evolutionary information. All components of EMEC are formulated in details, including the similar-taxis operation, the cooperation operation, and a simulated-annealing-based dissimilation operation (SADO). EMEC emphasizes on the share and the guide of the information in the search, and gets a performance superior to the simple MEC. The proposed EMEC was performed on some well-known benchmark problems. The experimental results show EMEC is a robust global optimization algorithm and can alleviate the premature convergence validly.

Keywords: mind evolutionary computation, similar-taxis, dissimilation, simulated annealing, and global optimization.

1 Introduction

Evolutionary algorithm (EA) has proved a viable and valid technique for search, machine learning and optimization problems [1,2]. Much research has been done to develop and improve EA[3-5]. However, there are still some complaints towards EA about its problems such as the premature convergence, the complex parameters control and the high computation costs etc…Recently some studies have focused on analyzing the social behavior of certain biologic systems (for example, a flock of birds, a school of fish or a colony of ants, etc…), and attempted to look for new and simple simulated computation models to overcome the inherent flaws lying in EA [6-
Those studies have brought on some swarm-intelligence-based algorithms emerging in the family of evolutionary algorithm. Two notable algorithms are particle swarm optimization (PSO) [6] and ant colony optimization (ACO)[7], which have proved some viable technologies to solve optimization problems.

Mind evolutionary computation (MEC)[10] is such an algorithm whose architecture is specifically designed to confront the intrinsic flaws in genetic algorithm (GA). Though based on the concepts of “population” and “evolution” in GA, it’s different from GA essentially. Inspired by man’s mind action attributes under certain social environment, MEC comes up with “similar-taxis” and “dissimilation” to replace “crossover” and “reproduction” in GA, and firstly solves problems by simulating the process of human mind evolution. Through the similar-taxis and the dissimilation acting alternately, MEC can achieve the balance between the exploration and the exploitation to some extent. Moreover, MEC designs billboards to record the evolutionary information that will in turn guide the evolution. Now MEC has been applied to lots of optimization problems successfully, such as the complex numerical optimization, the combination optimization, machine learning[10-12], etc. The relative studies have proved MEC outperforms GA in the convergent speed and the optimization performance. However, the main literatures on MEC only show some preliminary research, little study is done to analyze its social simulated mechanism and make its framework more rigid and substantial. Moreover, the dissimilation operation in MEC needs to be improved to guarantee its global convergent ability and contribute to overcome the premature problems.

Based on the analysis of the simulated mechanisms about MEC, the paper develops an extended computation model for MEC (EMEC). EMEC emphasizes on the share and the guide of the information in the search, and manipulates the search based on the behavior space and the information space. All operations in the behavior space are processed based on groups that symbolize the solution area, while the operations in the information space are done based on the billboards that are used to record the evolutionary information. Considering the exchange of information between different groups can contribute to the evolution of the mind in the whole society, the cooperation operation is introduced into MEC and become an essential operation just like the similar-taxis operation and the dissimilation operation. In order to overcome the premature convergence, EMEC introduces the simulated annealing into the dissimilation process and develops a simulated-annealing-based dissimilation operation (SADO). SADO starts the search from the local optimum explored by the similar-taxis operation, which makes full use of the local search ability of similar-taxis and the global search ability of SA. Through the introduction of the SA, SADO can make the search easy to escape from the explored local optimum and improve the global convergent ability of MEC.

The paper begins with a brief introduction to MEC in Section 2. Section 3 provides a detailed analysis about the simulated mechanism of MEC and develops the extended version of MEC. All components of EMEC are formulated in Section 4, including the similar-taxis operation, the cooperative operation, and the simulated-annealing-based dissimilation operation. Section 5 presents some experiments and makes some performance analysis of EMEC. Finally, some concluding remarks are drawn in Section 6.
2 Mind Evolutionary Computation

2.1 The Architecture of MEC

The first version of MEC is developed by Cheng-yi Sun in 1998 which is named as the simple mind evolutionary algorithm (SMEC) [10]. Its architecture is illustrated in the following Fig. 1.

Obviously, the population of MEC consists of several groups surviving around the environment. Those groups are divided into superior groups and temporary groups according to their evolutionary action during an evolution. Each group owns a local billboard and a set of individuals. Every individual is given a score and the score is the main information to guide the evolution. In order to trace the local and global competition, MEC provides two kinds of billboards (a local one and a global one) to record the evolutionary information drawn by the knowledge abstractor.

![Fig. 1. The architecture of the simple MEC](image)

2.2 The Evolutionary Process of MEC

Inspired by the development of human mind, MEC designs two operations to solve the optimization problems. One is similar-taxis operation; the other is dissimilation
operation. The similar-taxis operation conducts local search and makes “exploitation” in a local search area, while the dissimilation operation conducts global search and makes “exploration” in the whole search space. The evolutionary process is described as the follows:

A. Initialization of groups:
To gain a solution of an optimization problem, MEC initializes the population with $S$ superior groups and $T$ temporary groups randomly. Each group consists of $m$ individuals bred by the uniform distribution in the solution space.

B. Similar-taxis and local competition:
Once the similar-taxis operation begins, the individuals will compete with each other to find the local optimum in every group. In a new generation, all individuals are scattered around the winner by the normal distribution and the score of each individual is computed to search for a new winner. If the score of one individual is higher than that of the old winner, it will be regarded as the new winner of the group, and the information about the new winner will be recorded to the local billboard. The process mentioned above doesn’t stop until the group is mature.

C. Dissimilation and global competition:
During each stage, global competition never stops among all groups. If the score of certain temporary group is higher than that of any mature superior one, the temporary group will take place of the superior group. Likewise, if the score of certain mature temporary group is lower than that of any superior one, the temporary group will be discarded. At this time, the dissimilation operation begins. A new temporary group will be produced at random in the solution space to replace the discarded groups. After that, the new group will go through the similar-taxis operation and the local search will continue.

Repeating the above process till the score of the superior group is so high that it is impossible to further increase, then the algorithm is regarded convergent and the winner of the superior group just is the global optimum.

Obviously, MEC take advantage of the similar-taxis and the dissimilation to make the local search and the global search alternately, so MEC tends to get the balance between “exploration” and “exploitation” through the two operations. MEC can be regarded as a “free” computing model. Here, “free” means the model is open and extensible. The first version of MEC(SMEC) develops two new concepts to describe the search, but it doesn’t offer any formulations about the operations in details. Moreover, SMEC pays little attention to the disposals of the evolutionary information and doesn’t make full use of the memory mechanism of MEC. In order to improve SMEC to fit more complex problems, it’s necessary for us to make an analysis about the simulated mechanismlying in MEC.

3 The Simulated Mechanism of MEC

According to the descriptions above, the architecture of MEC is obviously similar to the construction of human society, and the algorithm just emulates the simple mode of human mind. In MEC, every individual can be regarded as an intelligent agent surviving in certain group, who can “think” and “reason”. Through “considering”, it
can perceive the social pressure in different directions and has a desire to get an uninterrupted development. In order to obtain a higher identification from the society, it always learns from the superiors or the winners around itself continuously to improve its score (similar-taxis). At the same time, all individuals in a group always own a same belief in order to develop their group. With the score of every individual increasing, the level of the group gets an improvement in the whole society. When the score of every individual in a group are the same as or near to the score of the winner, the group and its winner will make a creation or learn from the other groups to get some new advanced methods and experience to obtain a chance for a new development in the whole society (dissimilation).

According to the analysis above, it’s easy to know the simulated mechanism of MEC mainly derives from the mind modes of a cluster of individual agents. Here, the mind modes simulated by MEC mainly include the similar-taxis and the dissimilation of mind. Besides that, the information is also an important factor should be emphasized because the information always has a great influence on the change of mind. During the development of the social, all groups should develop a cooperative relationship and get a mutual improvement based on the shared information.

Owing to the viewpoints above, an extended version of MEC (EMEC) is developed in the paper. Its configuration is divided into behavior space and information space. The behavior space involves such operations as the similar-taxis, the cooperation and the dissimilation operations, while information space involves a series of disposals of evolutionary information. The knowledge abstractor just provides a bridge between the two spaces. The framework of EMEC is described in Fig.2.

![Fig. 2. The Framework of EMEC](image-url)
In the following sections, the main components in EMEC will be described in detail based on a kind of non-linear programming problem. The problem can be described as the following:

$$\min f(X), X \in S \subseteq \mathbb{R}^n; S = [a_i, b_i]_{i=1}^n.$$  \hspace{1cm} (1)

Where, $f(X)$ is a unimodal or multimodal function while $X$ represents any one solution vector match with the function; $S$ means the solution space.

4 Main Components in EMEC

4.1 Groups and Billboards

MEC manipulates the search based on multi-groups. During the search for an optimum, each group is regarded as a mobile objective wandering in the solution space. Each group consists of $m$ individuals (m candidate solutions), and all individuals will be evaluated according to the objective function and be given a score. The individual with the highest score will be chosen as the winner, and every group will be evaluated according to the score of its winner.

Once a group is born, it will go through three stages: initialization, evolution and maturity. When the score of a group no longer changes, the group will be regarded mature. During the evolution, a group tends to be near to another group gradually, till the two groups collide with each other in the end. Under that situation, the two groups will be regarded as two assimilative groups and must be combined into a group.

In MEC, every group is offered with a local billboard that provides all individuals with a chance to share the valid information. The information about the winner often is regarded as important evolutionary information to indicate the trajectory of the group. The global billboard is designed to record the information about the global search and the change of the environment, which is often obtained by integrating the information on the local billboard, generally including the current location of each group, the state of each group (immature, mature or assimilative), the winner of each group and some field knowledge on the objective problem.

During the search for a global optimum, the information on the local billboards and global billboard will be updated at intervals. All of information are open to every group and shared among all groups, which offer a guide for the search and can enhance the search efficiency validly.

4.2 The Similar-taxis Operation

Similar-taxis is an exploiting operation that occurs in all groups in parallel. As far as the numerical optimization is concerned, the similar-taxis operation can be formulated as the following computation model:
\[
\begin{align*}
X^k_{id}(t + 1) &= X^k_{id}(t) + \sigma^k_{id}(t + 1)N_i(0, 1) \\
\sigma^k_{id}(t + 1) &= \rho \sigma^k_{id}(t)
\end{align*}
\] (2)

Supposed every individual in MEC is described as a multi-dimension vector, \(X^k_{id}\) represents the \(d\)th dimension of \(i\)th individual in the \(k\)th group and \(\sigma^k_{id}\) is its learning step; \(X^k_{wd}\) is the winner’s vector in the \(k\)th group; \(N_i(0, 1)\) is a random number obeying the normal distribution; \(\rho\) is defined as a “cognitive” parameter, which will modify the learning step according to the individual’s perception on the learning pressure coming from the winner.

According to equation (2), every individual in a group improves its score through learning from the winner with a certain learning step. During the search, the winner of the group is regarded as a “banner” and followed by all other individuals, all individuals tend to make a “conscious” learning and go on a local search in a local neighborhood of the winner. If the score of an individual is higher than that of the winner, it will replace the old one and be chosen as the new winner of the group. Once the winner changes, the other individuals will turn their learning direction immediately and begin a new search around the winner. Under the guide of the winner, all individuals can rapidly develop themselves and converge to a local optimum after several iterations.

During the similar-taxis process, \(\sigma^k_{id}\) and \(\rho\) are two important factors that influence the searching efficiency of the similar-taxis operator. Here, we can look \(\sigma^k_{id}\) as a factor relative to the individual’s learning pressure coming from the winner. Generally, the lower the score of an individual is (compared with the one of the winner), the bigger learning pressure the individual receives, then the individual will learn from the winner with a bigger learning step. With the individual’s score improving, the individual is near to the winner gradually, so the learning step will be shrunken down with the learning pressure reducing. As is shown in equation (2), the learning step \(\sigma^k_{id}\) is adjusted by \(\rho\), so the value of \(\rho\) should be elaborated to choose and match with the adaptive change of the learning step.

4.3 Cooperation among the Groups

It’s well known that the share and the spread of the information among the whole social is a key that prompts the development of technology and science greatly. Through the similar-taxis operation, every group would get a development independently and be bound to converge to a local optimum. At the same time, the group must be open and tends to receive the influence coming from the other groups to get a great improvement, just as an innovation always emerges in a kind of social culture when another one intervenes in or gives an impact on it. Considering the above viewpoints, it is necessary to bring the cooperation operation into MEC.

During the cooperation, every group will exchange the information with a superior one and make a change. The operation will be manipulated as the follows:
Here, $X^k_w$ is the winner of a certain group, and $X^*_w$ is the winner of another superior group which has a higher score than that of $X^k_w$. $X^*_w$ represents a new seed. If $X^*_w$ is superior to $X^k_w$, $X^*_w$ will replace $X^k_w$ to be the new seed of the group and guide the individuals turning to a more potential area. Otherwise, the local search will continue and make an “exploitation” deeply around $X^k_w$.

According to equation (3), every group tends to learn from a superior one with an adaptive learning step, and the learning step mainly depends on the distance between two winners. Through the cooperative learning, every group gets more chances to turn to a potential area, which can improve the search rate of similar-taxis to some extent.

### 4.4 Dissimilation Operation with SA

The dissimilation operator is defined to conduct global search and make “exploration” in the whole search space. The main task for the operation is to search more potential area for the similar-taxis operation and prevent the search from being trapped in certain local optimum.

Once a group is mature or assimilative during the evolution, the group will be discarded and the dissimilation operation will begin. At that time, a new group must be produced in the solution space to take place of the discarded one. The production of a new group depends on a new better seed (winner) appearing. In SMEC, the new seed mainly produced by the uniform distribution in the whole solution space that is referred as a random dissimilation operator (RDO). Though RDO can guarantee the algorithm converge to the global optimum in theory, it owns a lower efficiency because the operation discards all information captured in the search. In order to improve the search efficiency, the new seed should be chosen based on the evolutionary information. According to the concept of the dissimilation, other valid global methods can be used to develop the operation.

Simulated annealing (SA) is a robust global optimization algorithm [15-18] and the motivation of the method lies in the physical process of annealing, in which a solid is heated to a liquid state and then gradually cooled until another solid-state with minimal inner energy is reached. The process of annealing aims to reach the lowest energy state just as the process of a search for a minimum of an optimization problem. As far as an optimization problem is concerned, SA begins the search with a random point and jumps to a new point with a finite step. Later, the new point will be evaluated according to the objective problem. If the new point is superior to the old one, it will be accepted naturally and a new step of search will begin with it. During the search, some inferior points for the objective problem may also be accepted with a probability determined by the metropolis criteria. Just because it can accept the inferior point occasionally, SA tends to escape from the local optima and converge to a global optimum. SA has proved a valid global optimization algorithm in Theory.
However, the performance of SA mainly depends on an accurate annealing schedule that always is difficult to make [15], including an appropriate initial temperature \(T_0\), the modification to the temperature parameter \(T\), etc… If the annealing plan isn’t reasonable, SA will lose its global search ability. Moreover, as a single-point-based algorithm, SA always spends lots of time and computation costs to converge to a global optimum and has a lower convergent efficiency.

Here, SA is introduced into MEC to develop a SA-based dissimilation operation (SADO). SADO makes full use of the local search ability of similar-taxis and the global search ability of SA, and develops a kind of hybrid global optimization algorithm. SADO starts the search from the local optima captured by the similar-taxis operation, and makes the search easy to escape from the explored local optima. In SA, “exploitation” and “exploration” mainly are controlled by the temperature parameter. Generally, SA begins the search with a higher temperature and makes a global search in the forepart, while a higher temperature just means a bigger accepting probability to an inferior solution. With the search continuing, the temperature is reduced slowly, the accepting probability of an inferior solution is also decreased, so the search tends to make a local search at the end. Because SADO mainly take advantage of the global search ability of SA, a simple modification to the temperature parameter is enough to guarantee the search escaping from the local points, which greatly reduce the dependence of the performance on the temperature parameter in SA.

When a new point produced by SA, the new point will be evaluated according to the information on the global billboard. If the new point is different from all captured points recorded on the global billboard, it will become a new seed. At the same time, a new group will be born to replace the discarded one and the similar-taxis operation will be conducted on it.

4.5 The Pseudocode for EMEC

According to the descriptions above, we can get the pseudocode of EMEC described in the Fig.3.
5 Numerical Experiments and Results

5.1 Benchmark Functions

Numerical experiments were conducted to test the effectiveness and efficiency of the EMEC. Nine benchmark functions were selected in our experiments, which are very popular in the relevant studies to evaluate an optimization algorithm. The benchmark functions are formulated as the follows:

\[ f_1 = \sum_{i=1}^{N} x_i^2 \]  \hspace{1cm} (4)

\[ f_2 = \sum_{i=1}^{N} x_i^4 + \text{random} \ [0,1) \]  \hspace{1cm} (5)

\[ f_3 = \sum_{i=1}^{N} |x_i| + \prod_{i=1}^{N} x_i \]  \hspace{1cm} (6)
\[ f_4 = \sum_{i=1}^{N} [100 (x_i^2 - x_i)^2 + (1 - x_i^2)] \]  
(7)

\[ f_5 = \sum_{i=1}^{N} x_i^2 - 10 \cos(2\pi x_i) + 10 \]  
(8)

\[ f_6 = \frac{1}{4000} \sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \]  
(9)

\[ f_7 = \sum_{i=1}^{N} -x_i \sin(\sqrt{|x_i|}) \]  
(10)

\[ f_8 = \frac{1}{N} \sum_{i=1}^{N} (x_i^4 - 16x_i^2 + 5x_i) \]  
(11)

\[ f_9(x) = (x_1 - 10x_2)^2 + 5(x_3 - x_4)^2 
+ (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 
+ 40[\sin^2(kx_1) + \sin^2(kx_2) 
+ \sin^2(kx_3) + \sin^2(kx_4)] \]  
(12)

The benchmark functions above can be divided into two categories according to their complexities. \( f_1 \sim f_4 \) are unimodal functions with different unique features, which are relatively easy to optimize, but the difficulty increases with the dimension growing. \( f_5 \sim f_9 \) are multimodal functions with many local optima and they are very difficult to solve for many optimization algorithms. Some basic information of the test functions in experiments is listed in table 1.

<table>
<thead>
<tr>
<th>Test function</th>
<th>Solution space</th>
<th>Global minimum</th>
<th>Dimensions of function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>([-100,100]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>([-1.28,1.28]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>([-10,10]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>([-30,30]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>([-5.12,5.12]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>([-600,600]^N)</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>([-500,500]^N)</td>
<td>-12569.5</td>
<td>30</td>
</tr>
<tr>
<td>( f_8 )</td>
<td>([-5,5]^N)</td>
<td>-78.3323</td>
<td>100</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>([-1,1]^N)</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
5.2 Experimental Analysis of the Parameter $\rho$

The first set of experiments is designed mainly to analyze the influence of the parameter $\rho$ on the efficiency of the similar-taxis operation. In the experiments, MEC is used to solve some functions above only through the similar-taxis with different $\rho$. Here the parameters are set as follows: the size of group is 50 and the max generation is limit to 400; the error precision is 0.001; the initial learning step $\sigma_k^1(0) = (b - a) / 10$. The mean results based on $f_6 (k=3)$ are shown in the following Table.2. $f_6$ is a complex optimization function whose global optimum encircled by lots of near-optima, especially, the number of the near-optima will increase with the coefficient $k$ enhancing.

Table 2. The influence of the parameter $\rho$ on the performance of the similar-taxis operation. The mean results on $f_6$ were averaged over 50 runs. Here “Mean CGen”indicates the mean convergent generations in 50 runs, and “success times” means the times of the convergence successfully over 50 runs.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>0.45</th>
<th>0.50</th>
<th>0.55</th>
<th>0.6</th>
<th>0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean CGen</td>
<td>400</td>
<td>232</td>
<td>198</td>
<td>76</td>
<td>20</td>
</tr>
<tr>
<td>Success times</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>36</td>
<td>50</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.75</td>
<td>0.8</td>
<td>0.9</td>
<td>0.95</td>
<td>1.0</td>
</tr>
<tr>
<td>Mean CGen</td>
<td>24</td>
<td>30</td>
<td>60</td>
<td>124</td>
<td>400</td>
</tr>
<tr>
<td>Success times</td>
<td>50</td>
<td>50</td>
<td>46</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2. shows the statistic results about the mean convergent generation and the times of the successful convergence over 50 runs when the parameter $\rho$ varies from 0.45 to 1.0. Based on the results in Table.2., we obtain two relation curves between the parameter $\rho$ and the two statistic, with the hope to find out some advice on the choice of the parameter $\rho$.

Fig.4. The relation curve between $\rho$ and the mean convergent generation
According to the Fig.4 and the Fig.5, it’s easy to know that the parameter $r$ influences the efficiency of the similar-taxis operation greatly. As far as $f_9$ is concerned, the similar-taxis operation can obtain a good search result at a faster search speed when $r$ is between 0.6 and 0.9, while the best performance can be obtained when the parameter $r$ is near to 0.7. The same experiments have been done on the other benchmark functions, the results indicate $0.6 < r < 0.9$ may be a good choice for the similar-taxis operation.

### 5.3 Experimental Analysis of the Cooperation Operation

The second set of experiments is designed aiming to analyze the influence of the cooperation operation on the performance of MEC. We have observed the experimental results and made a comparison analysis between the simple MEC (SMEC) and MEC with the cooperation operation (MECCO). In the experiments, the parameters are set as follows: the number of groups is 5; the size of group is 20; the max generation is limited to 1000 and the cognitive parameter $r = 0.8$. The statistic results are shown in the Table 3.

The statistic results in the Table 3, clearly show the MEC with the cooperation operation outperforms the simple MEC in the global performance. Under the limit of the same terminal generation and the same error precision, MECCO has a higher probability to converge to the global optimum successfully with a faster search speed on $f_1$, $f_4$, $f_6$, when compared with SMEC. Considering $f_5$ and $f_9$ are two multimodal functions that are difficult to optimize generally, MECCO has a slower convergent speed on $f_5$ and $f_9$ but has a higher convergent probability than SMEC. Maybe the result can be explained by the following reason: Through the share and exchange of information among groups, the cooperation operation just increases the diversity of the population to some extent. It’s well known that a higher diversity always contributes to overcoming the premature convergence, especially for the multimodal
and complex optimization problems. All the results indicate the cooperation operation can improve the global search ability of MEC with small costs.

**Table 3.** Performance comparison between SMEC and MECCO. The mean results were averaged over 50 runs. The dimension of \( f_1, f_5, f_6 \) is 10 and the coefficient \( k \) of \( f_9 \) is 10. “Mean CGen” indicates the mean convergent generations over 50 runs, and “success rate” means the rate of the times of the successful convergence over 50 runs.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Algorithms</th>
<th>Mean CGen</th>
<th>Success rate</th>
<th>Error precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>SMEC</td>
<td>165</td>
<td>100%</td>
<td>0.000001</td>
</tr>
<tr>
<td></td>
<td>MECCO</td>
<td>128</td>
<td>100%</td>
<td>0.000001</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>SMEC</td>
<td>478</td>
<td>96%</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>MECCO</td>
<td>456</td>
<td>100%</td>
<td>0.01</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>SMEC</td>
<td>668</td>
<td>90%</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>MECCO</td>
<td>725</td>
<td>96%</td>
<td>0.0001</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>SMEC</td>
<td>256</td>
<td>100%</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>MECCO</td>
<td>214</td>
<td>100%</td>
<td>0.01</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>SMEC</td>
<td>316</td>
<td>88%</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>MECCO</td>
<td>321</td>
<td>96%</td>
<td>0.001</td>
</tr>
</tbody>
</table>

**5.4 Experimental Analysis of EMEC**

In order to evaluate the performance of the EMEC in a whole, we designed the third set of experiments. The results are recorded in the following tables and its performances are compared with other global optimizations, including FEP(Fast Evolutionary Programming)[13], OGA/Q(An Orthogonal Genetic Algorithm with Quantization)[14], PSO(Choke Swarm Optimization)[9] and ESA(Enhanced simulated annealing)[17]. All comparisons are made based on two performance parameters: the mean number of evaluations and the mean best value. The parameters in EMEC are set as follows: the number of groups is 5 and the size of group is 20; The initial learning step \( a(0) = (b - a) / 10 \); The cognitive parameter \( p \in [0.5, 0.95] \). During the SADO, the initial annealing temperature \( T_0 = f(X_w) \) (here \( X_w \) is the winner of the discarded group); the parameter \( T \) is modified by the equation: \( T_{k+1} = \beta T_k \) (0.8 < \( \beta < 1 \)).

**Table 4.** Performance comparison between EMEC and FEP based on Mean number of evaluations and Mean best value. All results were averaged over 50 runs.

<table>
<thead>
<tr>
<th>Test func</th>
<th>Mean number of evaluations</th>
<th>Mean best value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FEP</td>
<td>EMEC</td>
</tr>
<tr>
<td>( f_1 )</td>
<td>150,000</td>
<td>15,893</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>300,000</td>
<td>30,265</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>200,000</td>
<td>25,414</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>200,000</td>
<td>78,618</td>
</tr>
</tbody>
</table>
Table 5. Performance comparison between EMEC and OGA/Q based on Mean number of evaluations and Mean best value. All results were averaged over 50 runs.

<table>
<thead>
<tr>
<th>Test func</th>
<th>OGA/Q Mean number of evaluations</th>
<th>Mean best value</th>
<th>EMEC Mean number of evaluations</th>
<th>Mean best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>112,559 15,893</td>
<td>0</td>
<td>7.694x10^3</td>
<td></td>
</tr>
<tr>
<td>$f_2$</td>
<td>112,652 30,265</td>
<td>6.301x10^3</td>
<td>8.835x10^3</td>
<td></td>
</tr>
<tr>
<td>$f_3$</td>
<td>112,612 25,414</td>
<td>0</td>
<td>7.502x10^1</td>
<td></td>
</tr>
<tr>
<td>$f_4$</td>
<td>167,863 78,618</td>
<td>7.520x10^1</td>
<td>6.834x10^1</td>
<td></td>
</tr>
<tr>
<td>$f_5$</td>
<td>224,710 106,589</td>
<td>0</td>
<td>8.566x10^3</td>
<td></td>
</tr>
<tr>
<td>$f_6$</td>
<td>134,000 89,496</td>
<td>0</td>
<td>5.294x10^2</td>
<td></td>
</tr>
<tr>
<td>$f_7$</td>
<td>302,166 150,438</td>
<td>-12569.45</td>
<td>-12569.42</td>
<td></td>
</tr>
<tr>
<td>$f_8$</td>
<td>245,930 165,116</td>
<td>-78.30002</td>
<td>-78.11692</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Performance comparison between EMEC and PSO based on Mean number of evaluations and Mean best value. All results were averaged over 50 runs.

<table>
<thead>
<tr>
<th>Test func</th>
<th>PSO Mean number of evaluations</th>
<th>Mean best value</th>
<th>EMEC Mean number of evaluations</th>
<th>Mean best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>250,000 15,893</td>
<td>11.175</td>
<td>7.694x10^3</td>
<td></td>
</tr>
<tr>
<td>$f_2$</td>
<td>250,000 78,618</td>
<td>47.1354</td>
<td>8.566x10^3</td>
<td></td>
</tr>
<tr>
<td>$f_3$</td>
<td>250,000 106,589</td>
<td>0.4498</td>
<td>5.294x10^2</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Performance comparison between EMEC and ESA based on Mean number of evaluations and Mean best value. All results were averaged over 50 runs.

<table>
<thead>
<tr>
<th>Test func</th>
<th>ESA Mean number of evaluations</th>
<th>Mean best value</th>
<th>EMEC Mean number of evaluations</th>
<th>Mean best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_4$</td>
<td>188,227 78,618</td>
<td>17.10</td>
<td>6.384x10^3</td>
<td></td>
</tr>
</tbody>
</table>
Table 6. and Table 7. compare EMEC with PSO and ESA based on several test functions. The results show that EMEC has a performance superior to the two algorithms.

According to the comparisons above, it’s easy to know EMEC can converge to a global optimum or near to optimum with certain accuracy in general, and has a higher search efficiency with a lower computation cost.

6 Conclusions

The paper makes an analysis on the simulated mechanism of mind evolutionary computation firstly and proposes an extended computation model for MEC. EMEC manipulates the search based on the behavior space and the information space, and emphasizes on the share and the guide of the information in the search. In order to improve the performance of SMEC, EMEC introduces the cooperation operation into MEC and develops a simulated-annealing-based dissimilation operation (SADO). A series of experiments have been designed to evaluate the performance of EMEC based on some benchmark functions. The relative experimental results show that the cooperation operation can contribute to improving the convergent efficiency of MEC through the share and exchange of information among groups, while SADO makes full use of the global search ability of SA to alleviate the premature convergence validly. All the studies indicate EMEC is a robust global optimization algorithm with high search efficiency in numerical optimizations. The following work will try to apply EMEC to others complex optimization problems, such as the multi-objective optimizations and the combined optimizations. EMEC is anticipated to become a generalized optimization algorithm.

Acknowledgements

This paper is supported by the National Natural Science Foundation (No 60174002) and Shanxi Youth Science Foundation (No 20031031) (Chinese). The authors of the paper express great acknowledgments for the supports.

References

4. Kazarlis S. A., Papadakis S. E., Theocharis J.B., Petridis V., Microgenetic algorithms as 
generalized hill-climbing operators for GA optimization. IEEE Transactions on 
Evolutionary Computation, Vol.5, No.3, June 2001: 204-217
5. Potter, M. and De Jong, K. Cooperative coevolution: An architecture for evolving 
6. Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proc. IEEE Int'l. Conf. on 
7. M.Dorigo, V.Maniezzo, and A.Coloni, “ The ant system: Optimization by a colony of 
8. Clerc, M. The swarm and the queen: towards a deterministic and adaptive particle swarm 
opimization. Proc. the Congress on Evolutionary Computation, Washington DC, USA 
(1999)1951-1957
9. P.J. Angeline, Evolution and Optimization versus particle swarm optimization: 
Systems.(1998) 355-359
Optimization Problems, Proc.3rd World Congress on Intelligent Control and Automation, 
Hefei, P.R. China. (2000)126-128
12. Jing Jie and Jianchao Zeng, “ Mind Evolutionary Computation Based on Guiding 
15. W.L. Goffe, G.D. Ferrier, J. Rogers, Global optimization of statistical functions with 
annealing: an experimental evaluation; Part I, Graph partitioning. Operations 
17. P.Siarry, G.Berthiau, F.Durbin and J.Haussy, Enhanced simulated annealing for globally 
minimizing functions of many-continuous variables, ACM Trans. Math. Software, 
vol.23, June (1997) 209-228
18. S. L.Rosen and C.M. Harmonosky, An improved simulated annealing simulation 
optimization method for discrete parameter stochastic systems, Computers & Operations 
Research 32.(2003) 343-358