http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (9): 1796-1803, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.1796.1803

Dual-system Cooperative Coevolutionary Differential Evolution Algorithm for Solving Nonseparable Function Optimization

[1]Feng-Zhe Cui, [1]Lei Wang, [3]Zhi-Zheng Xu, [1]Xiu-Kun Wang and [2,1]Hong-Fei Teng
[1]School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, People's Republic of China
[2]School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China
[3]School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China

Abstract: In recent years, researches on high-dimensional nonseparable function optimization have made progress. Approaches based on Potter's Cooperative Coevolutionary (CC) framework have achieved better results and aroused a great attention. However, the computational results are still unsatisfying for most Benchmark functions. Therefore, this study develops a dual-system (population) cooperative coevolutionary differential evolution (DCCDE) algorithm based on dual-system Evolutionary Algorithm (EA). This algorithm adopts a variable static grouping pattern and a improved Differential Evolution (DE) algorithm combined with simple crossover (SPX) local search strategy and modifies the migration pattern of the sub-individuals (not subpopulations) among the subsystems (subgroups of variables) in the dual-system. The test results of 20 Benchmark functions (including 17 nonseparable functions, dimension D = 1000) show that the proposed algorithm is better than other algorithms in computational accuracy.

Key words: Dual-system, cooperative coevolutionary differential evolution algorithm, nonseparable function, multimodal function, optimization

INTRODUCTION

The engineering background of nonseparable function optimization is one of the complicated coupling engineering system optimization problems. The nonseparable functions divide into partially nonseparable functions and fully nonseparable functions. The nonseparable functions and multimodal functions are no different in essentially, but only difference in the angle of view. The former is from the view of coupling relationship between variables; the latter is from multimodal perspective of the functions solution space structure. This kind of functions have more or less correlation (coupling) among variables, so currently the nonseparable functions, are the same as multimodal functions, have become a hotspot receiving wide attention. Especially, the fully nonseparable functions optimization problems are very difficult to solve.

The heuristic algorithms (such as PSO, DE, GA) are generally effective for nonseparable functions optimization problems with fewer dimensions (Zhang and Teng, 2009; Vesterstrom and Thomsen, 2004; Wang *et al*., 2013; Liu and Li, 2011; Gao *et al*., 2006). In recent years, CCEA (Potter CC) has attracted extensive attention and made encouraging progress in solving high-dimensional nonseparable function optimization problems. Potter and De Jong (1994) proposed a Cooperative Coevolutionary Genetic Algorithm (CCGA) and in 2000, developed the cooperative coevolutionary algorithms framework (CCEA). CCGA was used to solve the nonseparable Rosenbrock function. To overcome the difficulties presented by interacting variables, they modified the credit assignment algorithm. They adopted the best collaborator selection, the best and random collaborator selection to compare. The experimental results illustrate that the latter ways for evaluating individuals influence algorithm's performance. It is clear that the results influenced by the collaborator selection mechanism (Potter and De Jong, 2000). For a class of coupled functions, they explored an emergence of coadapted subcomponents to suggest that the evolution of species might need to be driven by more than the overall fitness of the ecosystem to produce good decompositions. Sofge *et al*. (2002) proposed a blended population approach for cooperative coevolutionary (BCCES). They combined Cooperative Coevolution Evolutionary Strategy (CCES) and standard Evolutionary Strategy (ES) in a single

Corresponding Author: Feng-Zhe Cui, School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, People's Republic of China Tel:+86 0411 84708823 Fax: +86 0411 84708412

evolutionary process, introduced a migration operator to allow populations to migrate from subpopulations of CCES to the population of ES. One population is consisted of subspecies and implemented as a Potter CCEA; the other is a traditional EA. Experimental results indicated that the blended population model outperformed the CCEA on nonseparable function problems (D = 2). Shi *et al*. (2005) presented the Cooperative Coevolutionary Differential Evolution (CCDE). CCDE adopted Potter's CC framework to improve the DE's performance. By splitting the solution vectors of DE, CCDE can partition a high-dimensional search space into smaller vectors. The subcomponents of a solution can be co-evolved by the multiple cooperating subspecies (or smaller vectors). The experimental results showed that, for non-separable problem (D = 100), CCDE outperformed the traditional DE and CCGA in performance. Yang *et al*. (2008) proposed a new cooperative coevolution framework (DECC-G) that could handle large scale nonseparable optimization problems. A random grouping scheme and an adaptive weighting strategy were used in problem decomposition to made the variables stronger coupled in the same group and the variables coupled as weak as possible in different groups. Experimental results indicated (D = 1000) that DECC-G could effectively solve nonseparable function optimization problems up to 1000 dimensions. Li and Yao (2012) developed a new cooperative co-evolutionary particle swarm optimization algorithm (CCPSO2) based on the cooperative co-evolutionary framework in order to solve large scale non-separable function optimization problem .The algorithm using new PSO model based on the combination of Cauchy and Gaussian mutation operator update particle personal best (Pbest) and neighborhood best position (Lbest), improved search ability and by using variable random dynamic grouping. The approach of variable random dynamic grouping is a kind of breakthrough research results. Standard test function set (Tang *et al*., 2007) test show that the CCPSO2 algorithm can effectively solve as high as 2000 dimensional non-separable function F7 and has good computational accuracy and robustness.

Although, the researches on solving nonseparable problems have made progress in recent years, most of the problems have not yet achieved the optimal solution. For most Benchmark functions, there even is several orders of magnitude away form the optimal solution.

In order to improve computational performance of algorithm in CC framework for high dimensional non-separable problems, we developed dual-system cooperative co-evolutionary differential evolution algorithm (DCCDE) to solve high dimensional non-separable function problem.

DUAL-SYSTEM COOPERATIVE COEVOLUTIONARY DIFFERENTIAL EVOLUTION

Basic idea: the DCCDE we develop is based on dual-system variable-grain cooperative coevolution algorithm (DVGCCEA), also called Oboe-CCEA (Teng *et al*., 2010). And the domain where the algorithm will apply is complex coupled system optimization problems (e.g., satellite layout optimization). Below is a brief description of the Oboe-CCEA. The Oboe-CCEA decomposes the original problem or system P (equivalent to non-separable function) into E subsystems PPe (e = 1,2,…,E) (equivalent to decomposing function variables into E groups) and then duplicates system P (copy) as systems A and B, each includes sub-systems (AAe or BBe), respectively. System A is a virtual Potter CC, while system B is an authentic one. The two systems evolve parallelly. System A is optimized globally (all-in-one) and the optimization of system B is achieved by its sub-systems BBe optimizing parallelly. And system A adopts a coarse-to-fine variable-grain strategy in order to reduce the time brought by the dual-system. It is the sub-individual migration between subsystems AAe and BBe on a group level, rather than the individual migration between systems A and B on a whole level, that improves the diversity in a population (the 'subsystem' here differs from the 'subpopulation' in literature Sofge *et al*. (2002). System B adopts the implicit coordination mechanism. The difference between Oboe-CCEA and traditional CCEA is that BBe are evaluated by system A rather than system B. Oboe-CCEA in literature (Teng *et al*., 2010) did better in solving satellite-module layout optimization, but it did not touch upon nonseparable function optimization problems.

A dual-system cooperative coevolutionary differential evolution with human-computer cooperation (HDCCDE) is developed in Literature (Zhang *et al*., 2012). HDCCDE incorporates human-computer cooperation based on Oboe-CCEA, namely, artificial individual is added in the optimization process of system A.

In this study, DCCDE employs the dual-system CCEA architecture in Oboe-CCEA (Teng *et al*., 2010). The differences between DCCDE and Oboe-CCEA lie in:

- DCCDE does not use the variable-grain strategy, because we mainly focus on functions optimization problems rather than complex engineering system optimization problems
- DCCDE decomposes the original problem P by static variables grouping, while Oboe-CCEA decomposes P according to the physical structure of engineering systems

- Systems A and B of DCCDE use the improved DE algorithm, which is combined with a simple cross (SPX) (Tsutsui *et al*., 1999) local search strategy, in order to improve the convergence
- Systems A and B evolve parallelly. System A optimizes on a global level and the optimization of system B is achieved by the cooperative coevolutionary among its subsystems BBe. AAe in system A are the virtual subsystems corresponding to the subsystems BBe in system B. Individuals migration between systems A and B is achieved by the sub-individuals migration between AAe and BBe. It can improve the diversity of a population. The sub-individuals migration process between AAe and BBe is the same as the one in Oboe-CCEA

In this study, we focus on dual-system CC coordination mechanism and improving the space searching ability of EA, in order to enhance the ability of DCCDE to solve nonseparable functions.

Variable grouping: In this study, variable static grouping is employed to solve fully nonseparable functions. We set a fixed number of variables groups and randomly distribute the variables into each group. The grouping principle is that the coupled relationships (usually strong correlation) of variables in the same group are kept the same as in original system and the variables in different groups are independent of each other. To illustrate the variables grouping, we take the partially nonseparable function F_{14} as an example. F14 is shown as follows:

$$\begin{aligned} F_{14} &= \sum_{k=1}^{D/m} F_{rot\_elliptic}[x(P_{(k-1)*m+1} : P_{k*m})] \\ F_{rot\_elliptic}(x) &= \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2 \end{aligned} \quad (1)$$

where, D is the dimension of variable; assume that E is the number of group and m = D/E is the dimension of variables in each group; P is a D dimensional random position vector. The decision variables of the kth group $x(P_{(k-1)*m+1}:P_{k*m})$ are shown in Fig. 1. For example, when D = 100, E = 5, thus m = 20.

Coordination mechanism of DCCDE and information exchange between systems A and B: The implicit overall coordination mechanism in CCEA relates to collaborator selection. Some common methods for collaborators selection are: random individual selection, multiple individuals selection, the best individual selection (Wiegand *et al*., 2001) and archive method (Panait *et al*., 2006). In this study, system B decomposes into several subsystems BBe, which should maintain the coordination consistency in the evolution process if there are coupled relationships among subsystems BBe.

The coordination mechanism we improve based on the Oboe-CCEA is described as follows. Before migrating from BBe (variable groups) to AAe (variable groups), the elite sub-individuals $X^{k^*}_{BBe}$ should be evaluated. The approach we adopt is that: according to the best collaborator selection, the collaborators $\bar{X}^k_{BBe}$ is selected from the remaining (E-1) subsystems (variable groups) in system B to constitute the complete individuals $X^k_{BBe} = \{X^{k^*}_{BBe}, \bar{X}^k_{BBe}\}$, which is evaluated by system A rather than system B in traditional CCEA. If the migration criteria is satisfied, $X^{k^*}_{BBe}$ and $\bar{X}^k_{BBe}$ migrate to the corresponding AAe to replace the worst individuals $X^{k,\ worst}_{AAe}$ and the X^k_{BB} in system A will iterate for m generations by the survival of the fittest. It is noteworthy that there are general not one single elite individual, but rather several subpopulations, migrating from BBe to AAe. The process that the elite individuals $X^{k^*}_{AAe}$ in AAe migrate to BBe is idem. It is worth stressing that the individual migrations between systems A and B are achieved by the sub-individual migrations between AAe and BBe in order to increase the diversity of the population. Systems A and B adopt the elitist preserving strategy and utilize their synergies to increase the

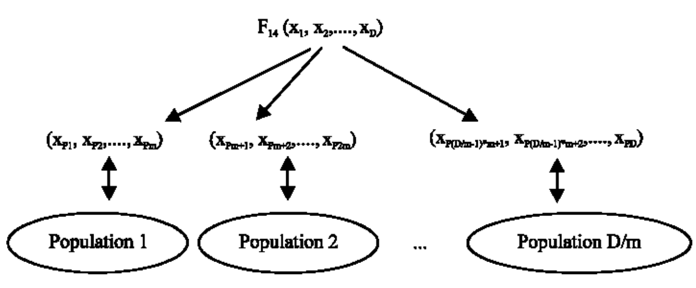

Fig. 1: The decomposition of function F_{14}

convergence of the algorithm. Finally system B approximates system A and system A approximates original system P (original problem).

Exploration ability of the improved heuristic algorithm: Exploration ability of an algorithm is mainly reflected in two aspects of diversity and convergence. DCCDE fully utilized the migration information between AAe and BBe to increase the diversity of the system and employs improved DE as the solving algorithm. In the improved DE, the individuals have good diversity in the early stage of evolution, so their exploration abilities are stronger; but with the increasing evolution generations, the differences among individuals decrease and the convergence rate slow in the later stage. The improved DE is an algorithm that combines the traditional DE with a simplex crossover (SPX) (Tsutsui *et al*., 1999) local search strategy. And the SPX local search strategy will be implemented every few generations in the optimization process. What we should note here is that a proper search length is very important for SPX local search strategy. A small length is unuseful to improve the search quality, while a longer length may bring additional unnecessary evaluations, which takes a lot of time. Considering the function complexity and the allowable maximum evaluation times, we set the maximum search times of SPX local search strategy T = D/5 (D is variable dimension). In addition, in order to avoid premature convergence and save evaluation times, SPX local search stops once a better fitness is obtained.

In the classical DE algorithm, the CR and F generally take fixed values. CR relates to the nature and complexity of problems and F relates to the convergence rate. The two are closely related to the considered problems and different problems have different optimal control parameters CR and F. Therefore, several algorithms that can adaptively adjust the control parameters of DE in the evolution process have been proposed in many literatures, such as SaDE (Qin *et al*., 2009), jDE (Brest *et al*., 2006). In this study, the adaptive strategy is also employed. We set CR = 0.5 (1+rand (0, 1), where rand (0,1) is a uniformly distributed random number between 0 and 1. F takes values by the following rules (Bdack *et al*., 1991): the success rate of mutations should be maintained at 1/5. If it is greater than 1/5, F increases; Otherwise, F decreases. Hence:

$$F^{t+k} = \begin{cases} C_d \times F^t, \text{if } p_s^t < 1/5 \\ C_i \times F^t, \text{if } p_s^t > 1/5 \\ F^t \quad\quad, \text{if } p_s^t = 1/5 \end{cases} \tag{2}$$

where, F^t and F^{t+k} are mutation factors in tth and (t+k)th generations, respectively, Cd = 0.82, Ci = 1/Cd, p_s^t is the success rate of mutations and measured over intervals of k trials.

DCCDE algorithm procedure: According to above algorithm description, the pseudo-code of DCCDE algorithm and improved DE algorithm are shown in Algorithm 1 and 2, respectively.

Algorithm 1: The pseudo-code of DCCDE:

```
begin
(1) decompose system P into E subsystems PPe;
(2) duplicate P to A and B respectively, that is the corresponding subsystems Aae and BBe
(3) for each Xai of A do
        (3.1) random initial population the individual Xai of A system;
        (3.2) evalute the individual fittness of f (Xai);
        end for
(4) for each BBe do
        (4.1) random initial subpopulation BBe;
        (4.2) evalute the individual fittness of f (XBBe);
        end for
(5) mutate facotr F = 1, 2, cross rate CR = 0, 9;
(6) while terminal condition do
        (6.1) A system use differential improve evolution algorithm (IDE) to evolve;
(6.2) B system employ CCDE to evolve;
(6.3) If the condition of mifration is met
        migrate the subindvidiuals between AAe and BBe;
        end if
 end while
end
```

Algorithm 2: The pseudo-code of Improved DE (IDE):

```
begin
(1) for each Xai in A do
        (1.1) slect three different random individuals from A population XA1, XA2, XA3;
            (1.2) if rand (0,1)<CR
            Ui = XA1+F×(XA2-XA3);
            if f (Ui) better than f (Xai);
            Xi = Ui;
          end if
        end if
  end for
(2) if the criteria of local search in met
    using local search algorithm SPX;
        end if
(3) if the frequency of successful mutations p<0.2
        F = 0.82×F;
        else if p>0.2
        F = 1/0.82×F;
        end if
(4) CR = 0.5×(1+rand (0,1));
end
```

NUMERICAL SIMULATION

Test functions: In this study, we use 20 representative Benchmark functions presented by literature (Tang *et al*., 2010) to test the proposed DCCDE algorithm.

Definition 1 (Tang *et al.*, 2010): A function f(x) is separable if:

$$\arg\min_{(x_1,\cdots,x_n)} f(x_1,\cdots,x_n) = \left(\arg\min_{x_1} f(x_1,\cdots),\cdots,\arg\min_{x_n} f(\cdots,x_n)\right) \quad (3)$$

Definition 2 (Tang *et al.*, 2010): A nonseparable function f(x) is called m-nonseparable function if at most m of its parameters xi are not independent. A nonseparable function f(x) is called fully-nonseparable function if any of its two parameters xi is dependent with each other.

According to the above two definitions, the 20 high-dimensional functions in (Tang *et al*., 2010) can divide into 3 categories: (1) separable functions F1-F3; (2) partially nonseparable functions F4-F18; (3) fully-nonseparable functions F19-F20.

The experiment is to compare the performance of DCCDE with the performance of CCDE Shi *et al*. (2005), DECC-G (Yang *et al*., 2008) and SDENS (Wang *et al*., 2010) in solving nonseparable function optimization problems. And the evaluation index is computational accuracy.

Experimental setup: The experimental parameters are set as follows. For the 20 test functions in (Tang *et al*., 2010), the variable dimension was set to 1000, the population size of systems A and B was set to 30, the population size of subsystems BBe and AAe was set to 30 and the number of subpopulations (variable groups) was set to 5. We allocated the 1000 variables into the 5 subpopulations. The control parameters CR and F were initialized with 0.9 and 1.2, respectively. In this study, the DE/rand/1/exp strategy was adopted by DCCDE. The maximum number of fitness evaluations (MAX_FES) is calculated by:

$$MAX\_FES = 3000\times D$$

where, D is the number of dimensions. In order to eliminate the influence brought by random operation in the initialization on the performance, every algorithm ran 25 times on each test function. For each test function, the averaged results of 25 independent runs were recorded. The function error for a solution x is defined as:

$$\Delta f(x) = f(x)-f(x^*) \quad (4)$$

where, x* is the global optimum.

Experimental results and analysis: The computational results of the 25 independent runs of DCCDE on the 20 Benchmark functions are listed in Table 1. The averages of function error Δf(x) in 25 independent runs of DCCDE, CCDE, DECC-G and SDENS are shown in Table 2. The MAX_FES in CCDE, DECC-G and SDENS were set to (3×10^6). Because DCCDE algorithm adopts dual-system architecture, therefore, MAX_FES in each system of DCCDE was set to (1.5×10^6) for fair.

From the experimental results of the 20 Benchmark functions in Table 2, we can find that DCCDE apparently outperforms SDENS.

Compared with CCDE, DCCDE performs better on most of the test functions, specifically, on the fully-nonseparable function F19. This owes to the Potter CC framework and sub-individuals migration pattern in dual-system of DCCDE. The coordination mechanism in DCCDE reflects the interactions between the sub-individuals in one variable group (subsystems) and the sub-individuals in other variable groups and it also provides a potential environmental pressure, which guides the evolution direction. The coordination consistency of evolution among subpopulations in CCDE is maintained by the collaborator selection. For weak coupled problems (or separable problems), the traditional coordination mechanism does well in guiding each subpopulation toward the optimal solutions, while for the strong coupled problems (or fully-nonseparable problems), it does poorly. After the sub-individuals in BBe migrating to AAe, system A optimizes on global level, thus the coupling relationships in original problem can be considered on the system level. After m iterations with the elitist strategy, system A migrate the global optimal solutions in AAe backwards to the corresponding BBe to conduct the evolution of each subsystem in system B.

Compared with CCDE, for the m-nonseparable functions, DCCDE performed worse than CCDE on F7 and F12. And the reason may be that the variable groupings for the two functions are improper. For F18 and F20, there is no significant difference between the two. Most importantly, DCCDE performed significantly better than CCDE on the remaining m-nonseparable functions. In a word, the proposed DCCDE algorithm achieved better results on the 12 out of the 17 m-nonseparable functions.

Compared with DECC-G, DCCDE performed better on 11 out of 17 functions (F5, F6, F7, F8, F10, F11, F13, F16, F18, F19 and F20) according to Table 2. DECC-G is a DECC with single system, adopts a new dynamic random grouping mechanism to increase the chance of allocating the strong coupled variables into the same subpopulation (variable group). Thus, it improves the ability of CCEA for solving nonseparable problems by using adaptive weighted strategy.

The test results show that DCCDE simultaneously outperformed CCDE and SDENS on 7 out of 17

Table 1: Experimental results of DCCDE of 25 independent runs for F1-F20, with dimension D = 1000

Function	Best	Median	Worst	Mean	Std.
F_1	3.66E-15	4.66E-15	6.97E-15	5.02E-15	1.16E-15
F_2	2.30E+02	2.42E+02	2.69E+02	2.45E+02	1.29E+01
F_3	1.67E-08	1.88E-08	2.06E-08	1.90E-08	1.37E-09
F_4	7.34E+12	1.17E+13	1.23E+13	1.04E+13	2.04E+12
F_5	1.67E+08	2.44E+08	2.74E+08	2.24E+08	3.90E+07
F_6	9.12E-06	1.11E-05	1.61E-05	1.18E-05	2.61E-06
F_7	1.07E+05	1.42E+05	2.69E+05	1.63E+05	6.00E+04
F_8	4.22E+07	4.29E+07	4.46E+07	4.31E+07	8.79E+05
F_9	1.59E+08	1.86E+08	2.41E+08	1.93E+08	3.02E+07
F_{10}	1.94E+03	2.16E+03	2.80E+03	2.27E+03	3.36E+02
F_{11}	6.46E-06	7.59E-06	1.22E-05	8.97E-06	2.51E-06
F_{12}	3.67E+01	4.93E+01	8.92E+01	5.72E+01	2.10E+01
F_{13}	4.03E+02	4.52E+02	5.59E+02	4.64E+02	5.73E+01
F_{14}	5.18E+08	5.86E+08	6.45E+08	5.82E+08	4.82E+07
F_{15}	4.69E+03	5.10E+03	5.36E+03	5.07E+03	2.42E+02
F_{16}	1.34E-05	1.59E-05	1.79E-05	1.59E-05	1.67E-06
F_{17}	9.69E+02	9.89E+02	1.09E+03	1.01E+03	5.08E+01
F_{18}	8.00E+02	8.54E+02	9.11E+02	8.62E+02	4.72E+01
F_{19}	5.55E+05	5.88E+05	1.23E+06	8.27E+05	3.56E+05
F_{20}	9.82E+02	9.83E+02	9.84E+02	9.83E+02	5.29E-01

F_1-F_3: Separable functions, F_4-F_{18}: Partially nonseparable functions and F_{19}-F_{20}: Fully nonseparable functions

Table 2: Comparison between CCDE and DECC-G, SDENS and DCCDE of 25 independent runs on function F1-F20, with dimension D = 1000

	CCDE	DECC-G	SDENS	DCCDE	
Function	Median	Median	Median	Median	Std.
F_1	1.67E-09(3)	8.81E-12(2)	5.73E-06(4)	5.02E-15(1)	1.16E-15
F_2	1.70E+01(1)	4.42E+02(3)	2.21E+03(4)	2.45E+02(2)	1.29E+01
F_3	3.94E-08(3)	3.30E-08(2)	2.70E-05(4)	1.90E-08(1)	1.37E-09
F_4	1.54E+14(4)	2.29E+12(1)	5.11E+12(2)	1.04E+13(3)	2.04E+12
F_5	3.28E+08(4)	2.45E+08(3)	1.18E+08(1)	2.24E+08(2)	3.90E+07
F_6	9.98E-05(2)	8.77E-03(4)	2.02E-04(3)	1.18E-05(1)	2.61E-06
F_7	9.97E+03(1)	1.10E+07(3)	1.20E+08(4)	1.63E+05(2)	6.00E+04
F_8	5.47E+07(3)	6.14E+07(4)	5.12E+07(2)	4.31E+07(1)	8.79E+05
F_9	1.24E+09(4)	1.41E+07(1)	5.63E+08(3)	1.93E+08(2)	3.02E+07
F_{10}	3.29E+03(3)	2.48E+03(2)	6.87E+03(4)	2.27E+03(1)	3.36E+02
F_{11}	3.90E-05(2)	3.52E-02(3)	2.21E+02(4)	8.97E-06(1)	2.51E-06
F_{12}	1.19E+01(1)	7.87E+01(2)	4.13E+05(4)	5.72E+01(3)	2.10E+01
F_{13}	7.87E+02(3)	5.50E+02(2)	2.19E+03(4)	4.64E+02(1)	5.73E+01
F_{14}	2.56E+09(4)	2.91E+07(1)	1.88E+09(3)	5.82E+08(2)	4.82E+07
F_{15}	6.86E+03(3)	3.88E+03(1)	7.32E+03(4)	5.07E+03(2)	2.42E+02
F_{16}	4.38E-05(2)	4.01E-01(3)	4.08E+02(4)	1.59E-05(1)	1.67E-06
F_{17}	4.15E+02(2)	1.03E+02(1)	1.08E+06(4)	1.01E+03(3)	5.08E+01
F_{18}	8.59E+02(1)	1.80E+03(3)	3.08E+04(4)	8.62E+02(2)	4.72E+01
F_{19}	1.31E+08(4)	1.14E+06(3)	8.80E+05(2)	8.27E+05(1)	3.56E+05
F_{20}	9.82E+02(1)	3.33E+03(4)	9.90E+02(3)	9.83E+02(2)	5.29E-01

F_1-F_3: Separable functions, F_4-F_{18}: Partially nonseparable functions and F_{19}-F_{20}: Fully nonseparable functions that the results of DECC-G and SDENS are cited from literatures Yang *et al.* (2008) and Wang *et al.* (2010) and the results of CCDE and DCCDE are worked out in this study, The number in the bracket denotes sort order, The number 1 denotes best, the number 4 denotes worst

nonseparable Benchmark functions (D = 1000) and outperformed DECC-G on 11 out of the 17 functions.

In addition, DCCDE is compared with BCCES. BCCES was tested on 6 functions (D = 2) in literature Sofge *et al.* (2002) and there is only a convergent curve without any specific data Table. F1, F2, F3 in the convergent curve are comparable. The results show that DCCDE outperformed BCCES on F1, F3 and for F2, there is no obvious difference between the two algorithms. It indicates that, for 2-D functions, the dual-system (dual population) algorithm is better than the BCCES in literature Sofge *et al.* (2002), where the high-dimensional functions were not involved.

ACKNOWLEDGMENT

This study was supported by National Natural Science Foundation (Grant No. 50975039 and 51005034) and National Ministries Project (Grant No. A0920110001) of P.R. China.

CONCLUSION

On the basis of Oboe-CCEA, we develop DCCDE based on dual-system CC framework. In the dual-system of DCCDE, system A is a virtual Potter CC, while system B is the authentic Potter CC. We mainly focus on variable static grouping, dual-system CC framework and its coordination mechanism and improving the space searching ability of heuristic algorithms.

The reasons why DCCDE can improve the diversity of populations are: (a) a dual-system Potter CC framework and a new migration pattern of the subpopulations (or sub-individuals) between AAe and BBe are adopted; (b) the improved DE is employed, therefore the searching ability in the early stage of optimization is good and the synergistic effect among subsystems BBe is played. The reasons why DCCDE can improve the convergence are: (a) a variable static grouping strategy that uses fixed number of groups to group the variables randomly is used to keep the coupled relationships of variables in a variable group the same as they are in original problem (original system) and each variable group (subsystem) is independent with others; (b) due to the improved DE, the searching ability in the later stage of optimization is good; (c) system A optimizes on the global level and system B optimizes based on Potter CC, synchronously, the elitist preserving strategy is employed, thus system B approximates system A, which approximates system P (the original problem).

The test results show that, for most of the 17 nonseparable Benchmark functions (D = 1000), the proposed DCCDE is better than other algorithms in computational accuracy. However, the proposed variable grouping pattern is improper for some functions; thereby our work will focus on the variable dynamic grouping pattern in the near future.

REFERENCES

Bdack, T., F. Hoffmeister and H.P. Schwefel, 1991. A survey of evolution strategies. Proceedings of the 4th International Conference on Genetic Algorithms, July 13-16, 1991, San Diego, CA., USA., pp: 2-9.

Brest, J., S. Greiner, B. Boskovic, M. Mernik and V. Zumer, 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Trans. Evolut. Comput., 10: 646-657.

Gao, H., B. Feng, Y. Hou, B. Guo and L. Zhu, 2006. Adaptive SAGA based on mutative scale chaos optimization strategy. Inform. Technol. J., 5: 524-528.

Li, X. and X. Yao, 2012. Cooperatively coevolving particle swarms for large scale optimization. IEEE Trans. Evol. Comput., 16: 210-224.

Liu, Y. and S. Li, 2011. A new differential evolutionary algorithm with neighborhood search. Inform. Technol. J., 10: 573-578.

Panait, L., S. Luke and J.F. Harrison, 2006. Archive-based cooperative coevolutionary algorithms. Proceedings of the 8th Annual Conference on GENETIC and Evolutionary Computation, July 08-12, 2006, Washington, USA., pp: 345-352.

Potter, M.A. and K.A. De Jong, 1994. A cooperative coevolutionary approach to function optimization. Proceedings of the International Conference on Evolutionary Computation and the 3rd Conference on Parallel Problem Solving from Nature, October 9-14, 1994, Jerusalem, Israel, pp: 249-257.

Potter, M.A. and K.A. De Jong, 2000. Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evol. Comput., 8: 1-29.

Qin, A.K., V.L. Huang and P.N. Suganthan, 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evolut. Comput., 13: 398-417.

Shi, Y.J., H.F. Teng and Z.Q. Li, 2005. Cooperative co-evolutionary differential evolution for function optimization. Proceedings of the 1st International Conference on Advances in Natural Computation, August 27-29, 2005, Changsha, China, pp: 1080-1088.

Sofge, D., K.A. De Jong and A. Schultz, 2002. A blended population approach to cooperative coevolution for decomposition of complex problems. Proceedings of the 2002 Congress on Evolutionary Computation, Volume 1, May 12-17, 2002, Honolulu, HI., USA., pp: 413-418.

Tang, K., X. Yao, P.N. Suganthan, C. MacNish, Y.P. Chen, C.M. Chen and Z. Yang, 2007. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, pp: 1-18.

Tang, K., X. Li, P.N. Suganthan, Z. Yang and T. Weise, 2010. Benchmark functions for the CEC'2010 special session and competition on large scale global optimization. Proceedings of the IEEE Congress on Evolutionary Computation, July 18-23, 2010, Barcelona, Spain, pp: 1-23.

Teng, H.F, Y. Chen, W. Zeng, S. Yan-jun and H. Qing-Hua, 2010. A dual-system variable-grain cooperative coevolutionary algorithm: Satellite-module layout design. IEEE Trans. Evol. Comput., 14: 438-455.

Tsutsui, S., M. Yamamura and T. Higuchi, 1999. Multi-parent recombination with simplex crossover in real coded genetic algorithms. Proceedings of the Genetic and Evolutionary Computation Conference, June 12-14, 2009, China, pp: 657-664.

Vesterstrom, J. and R. Thomsen, 2004. A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems. Proceedings of the Congress on Evolutionary Computation, Volume 2, June 19-23, 2004, Portland, OR., USA., pp: 1980-1987.

Wang, C.X., C.H. Li, H. Dong and F. Zhang, 2013. An efficient differential evolution algorithm for function optimization. Inform. Technol. J., 12: 444-448.

Wang, H., Z. Wu, S. Rahnamayan and D. Jiang, 2010. Sequential DE enhanced by neighborhood search for large scale global optimization. Proceedings of the IEEE Congress on Evolutionary Computation, July, 18-23, 2010, Barcelona, Spain, pp: 1-7.

Wiegand, R.P., W.C. Liles and K.A. De Jong, 2001. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. Proceedings of the Genetic and Evolutionary Computation Conference, August 6, 2001, China, pp: 1235-1242.

Yang, Z., K. Tang and X. Yao, 2008. Large scale evolutionary optimization using cooperative coevolution. Inform. Sci., 178: 2985-2999.

Zhang, Y.N. and H.F. Teng, 2009. Detecting particle swarm optimization. Concurrency Comput. Pract. Exp., 21: 449-473.

Zhang, Z.H., Y.J. Shi and H.F. Teng, 2012. Dual-system cooperative coevolutionary differential evolution with human-computer cooperation for satellite module layout optimization. J. Convergence Inform. Technol., 7: 346-355.