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 11 (1): 67-75, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.67.75

Opposition-based Cooperative Coevolutionary Differential Evolution Algorithm With Gaussian Mutation for Simplified Satellite Module Optimization

[1]Yuan-Hui Wang, [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

Abstract: Layout problem of satellite module is a NP-hard problem with performance constraints. An opposition-based Cooperative Coevolutionary Differential Evolution algorithm with Gaussian mutation (OCCDEG) is developed here to solve this problem. In the proposed algorithm, the whole population was divided into several subpopulations. Cooperating with each other, these subpopulations evolve together to search the optimal solution. The cooperative coevolution decreases the complexity of problems and improves the convergence rate of the algorithm. The Opposition-Based Learning (OBL), employed for population initialization and evolution, enhances the exploration capability of the algorithm and improves the convergence rate of the algorithm. The Gaussian mutation operator enhances the local search of the algorithm. The experimental results from a layout design of simplified satellite module show that, compared with other algorithms in this paper, OCCDEG obtains the competitive computational efficiency and precision.

Key words: Differential evolution, opposition-based learning, gaussian mutation, coevolution, layout design, optimization, subpopulation

INTRODUCTION

Layout design of satellite module belongs to NP-hard problem, involving how to place a set of apparatuses, components and equipments in the limited space (satellite module) reasonably. With various constraints, the design aims at obtaining higher space utilization. The optimization functions of this kind of problems are discrete, non-linear and multi-modal. There are some shortcomings using evolutionary algorithm to solve layout optimization problems, such as slow convergence, being trapped in local optimum easily and large computational overhead. Therefore, in order to seek solutions, many researchers (Aladahalli *et al*., 2007; Braun *et al*., 1997; Cagan *et al*., 1998; Grignon and Fadel, 2004; Huang and Chen, 2006; Ning *et al*., 2004; Liu *et al*., 2011; Wang *et al*., 2009a; Wang *et al*., 2009b; Xiao *et al*., 2007; Zhang *et al*., 2008) focused on new evolution algorithms, improved evolution algorithms and hybrid algorithms consisting of heuristic and evolution algorithms.

This paper chooses Differential Evolution Algorithm (DE) as the basic algorithm to improve. DE was proposed by Storn and Price (1997). In the beginning of this century, DE gave rise to lots of research (Adeyemo and Otieno, 2009; Abbaspour and Samadzadegan, 2009; Ataei *et al*., 2009; Liu and Li, 2011; Otieno and Adeyemo, 2010; Peng and Wang, 2010; Soltani *et al*., 2008) for its global convergence and robustness. To improve machine intelligence algorithms, Tizhoosh (2005) proposed the concept of Opposition-Based Learning (OBL) and discussed some applications of OBL in machine learning algorithms. Rahnamayan *et al*. (2006) used the OBL to improve the convergence rate of the classical DE, proposing a new algorithm called opposition-based DE (ODE).

With the purpose of solving high-dimensional function optimization, Potter and Jong (1994) proposed the cooperative coevolutionary algorithm (CCEA) framework. The convergence and effectiveness of CCEA was proved by Jansen and Wiegand (2004) theoretically. Shi *et al*. (2005) applied Potter's framework to DE, proposing a Cooperative Coevolutionary DE (CCDE). His work shows that CCDE obtained a significant improvement in performance over the classical DE.

To address the layout optimization problem of Ref (Teng *et al*., 2001), Shi (2005) proposed a Cooperative Genetic Differential Evolution algorithm (CGDE), by adding Gaussian mutation to CCDE. With random

Corresponding Author: Yuan-Hui Wang, School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China Tel: 86-411-84709130 Fax: 86-411-84708812

disturbance caused by Gaussian mutation, the new algorithm obtained marked improvement in performance.

With the ideas in Ref (Shi, 2005; Shi *et al*., 2005; Tizhoosh, 2005), an Opposition-based Cooperative Co-Evolutionary Differential Evolution algorithm with Gaussian mutation (OCCDEG) was proposed here to solve layout design problem of simplified satellite module.

THE OCCDEG ALGORITHM

A brief overview of CCDE: The CCDE (Shi *et al*., 2005) decomposes the high-dimensional problem into low-dimensional subproblems and assigns each subproblem to a subpopulation. Each subpopulation is evolved by DE separately and all subpopulations are coevolved simultaneously. During the evolution, each subpopulation provides a cooperator for composing the whole solution to solve the problem.

There are two methods for choosing cooperators (Potter and Jong, 1994). Selecting the best individuals is the more commonly used method. It is also used in this paper: at the beginning of the evolution of subpopulation P_S (S is the subpopulation number), each subpopulation (except P_S) provides the best individual in current generation as a cooperator for the evaluation of P_S.

The calculation of individual fitness: the jth individual of the *S*th subpopulation $X_{S,j}$ and cooperators from other subpopulations form a whole solution. The value of the solution calculated by evaluation function is the fitness of $X_{S,j}$.

In this paper, two subpopulations (P_1 and P_2) coevolve together. N is the total number of individuals. N_{p1}, N_{p2} are the numbers of individuals in subpopulations ($N_{p1}= N_{p2}= N/2$). P_1 and P_2 form the whole population P. P denotes the whole problem. Correspondingly, P_1 and P_2 denote the two subproblems. Each of them processes half variables of the problem. D denotes the dimension number of the whole problem. D_{p1} and D_{p2} denote the dimension number of a subproblem respectively ($D_{p1}= D_{p2}=D/2$).

Opposition-based optimization: Opposition-Based Optimization (OBO) means that the optimizer compares the current solution with its opposite solution and chooses the better one. The idea of OBL was proposed by Tizhoosh (2005) and then Rahnamayan *et al*. (2006) applied it to DE in order to improve the algorithm. According to Ref (Rahnamayan *et al*., 2006), here are the definitions of opposition-based vector and opposition-based optimization:

Opposition-based vector: Let $X(x_1, x_2, \ldots, x_D)$ be a vector in D-dimensional space, where $x_1, x_2, \ldots, x_D$ are real numbers and $x_i \in [a_i, b_i]$ ($i = 1, 2, \ldots, D$). The opposite-based vector of X is defined by $\bar{X}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D)$ where,

$$\bar{x}_i = a_i + b_i - x_i \tag{1}$$

Opposition-based optimization: Let $X(x_1, x_2, \ldots, x_D)$, a vector in D-dimensional space with $x_i \in [a_i, b_i]$($i =1, 2, \ldots, D$), be a candidate solution. Assume f (x) is fitness function which is used to measure candidate's optimality. $\bar{X}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D)$ is the opposite-based vector of X ($x_1, x_2, \ldots, x_D$). If $f(\bar{x}) \le f(X)$, then X can be replaced by $\bar{x}$; otherwise continue with X. Hence, the vector and its opposite-based vector are evaluated simultaneously to continue with the fitter one.

In this paper, opposition-based optimization is applied in two main steps of the algorithm (Rahnamayan *et al*., 2006).

Step 1: Subpopulation initialization:

- Initialize the subpopulation $P_{S,G}$ randomly (S is subpopulation number. G is subpopulation generation. Here G = 0)
- Create the opposite subpopulation $OP_{S,G}$. Each element of individuals in $OP_{S,G}$: $\bar{x}_{i,j} = a_j + b_j - x_{i,j}$ ($i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, D$), where N denotes the individual number in subpopulation $P_{S,G}$; D denotes the dimension number of subpopulation $P_{S,G}$; $x_{i,j}$ and $\bar{x}_{i,j}$ denote the jth dimension variable of ith individual in $P_{S,G}$ and $OP_{S,G}$, respectively
- Select N fittest individuals from $\{ P_{S,G} \cup OP_{S,G} \}$ as the original subpopulation

Here cooperators are chosen randomly from other subpopulations for the calculation of individual fitness in initialization. Comparing with the randomly generated subpopulation, fitter starting candidates can be obtained by utilizing OBO.

Step 2: Subpopulation evolution:

- Subpopulation $P_{S,G}$ generates new subpopulation $P_{S,G+1}$ by mutation, crossover and selection
- If Rand(0, 1) < J, create the opposite subpopulation $OP_{S,G+1}$. Each element of individuals in $OP_{S,G+1}$: $\bar{X}_{i,j} = MIN_j + MAX_j - x_{i,j}$ ($i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, D$), where Rand(0, 1) is the uniform distribution between 0 and 1; J is the probability of whether to create the opposite subpopulation; MIN_j and MAX_j denote the maximum and minimum values of the jth element of individuals in current subpopulation, respectively
- Select N fittest individuals from $\{ P_{S,G+1} OP_{S,G+1} \}$ as the new subpopulation

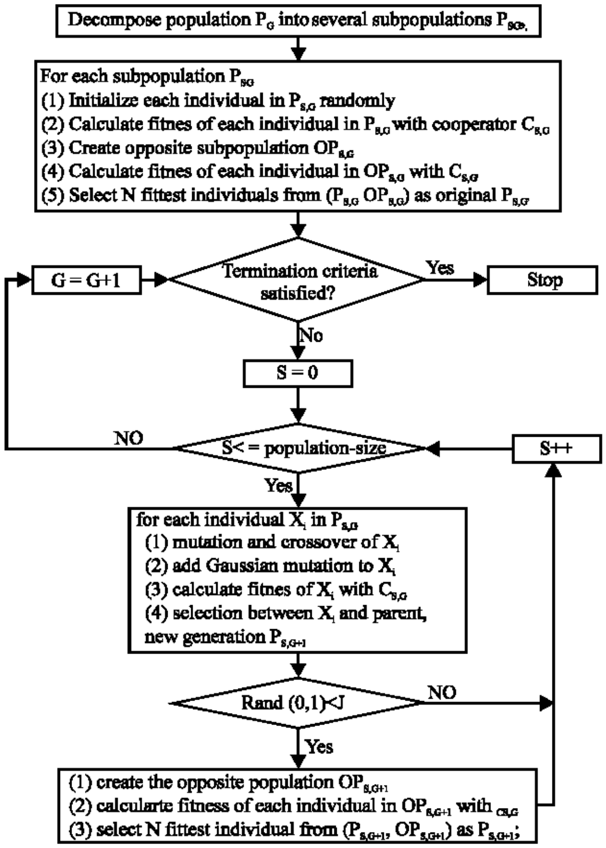

Fig. 1: The flow chart of OCCDEG

Here, the calculation of individual fitness of subpopulation $P_{S,G+1}$ and $OP_{S,G+1}$ is the same as CCDE described previously.

The operation of Gaussian mutation: With the same strategy Shi (2005), during the evolution, the Gaussian mutation is added to the each element of trial individual $X'_i = [x'_{i,1}, x'_{i,2},\ldots,x'_{i,D}](i = 1, 2, \ldots, N)$ which is generated by mutation and crossover of classical DE:

$$x''_{i,j} = x'_{i,j} + N(0,\sigma_j), j = 1,2,\ldots,D \tag{2}$$

where, $N(0, \sigma_j)$ denotes Gaussian random number; σ_j denotes the mutation step-size of jth element of X'_i.

The flow chart of OCCDEG: According to the flow chart of ODE (Rahnamayan *et al*., 2006), the flow chart of OCCDEG is as Fig. 1:

LAYOUT OPTIMIZATION OF SIMPLIFIED SATELLITE MODULE

Application instance: This application instance is cited from Teng *et al*. (2001). The layout problem of a simplified satellite module is to locate objects within the rotating frustum module as shown in Fig. 2a. All the objects are simplified as cuboids or cylinders. Its sectional view is shown in Fig. 2b. This problem is a three-dimensional packing problem with coupling variables and performance constraints (dynamical non-equilibrium constraints).

For this instance, parameters are as follows: ω = 40 r/min, R_0 = 700 mm, R_b=1000 mm, R_t = 600 mm, H_t = 1400 mm, H_0 = 800 mm, t = 70 mm. There are 19 objects to locate on the bearing plate. Among them, 6 objects are pre-fixed. Others are movable objects which contains 12 cuboids and 1 cylinder. The edges of cuboids are parallel to the axes. The permissible values of dynamical non-equilibrium force and dynamical non-equilibrium moment are 10 N and 20 N·m. The initial data of objects is in Table 1 (Huo *et al*., 2005).

Experimental setup: According to above instance, the mathematical model is the same as that of Ref (Teng *et al*., 2001). The flow chart of OCCDEG is shown in Fig. 1. Based on the CCEA framework, the layout problem (whole population) on the bearing plate is decomposed into two subproblems (subpopulation). The up base and down base of the bearing plate correspond to one

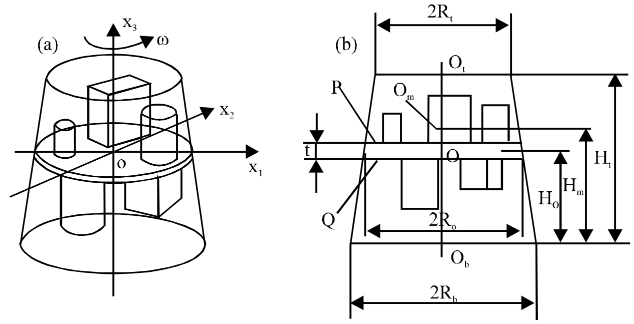

Fig. 2: The schematic diagram of locating objects within a rotating frustum module (Teng *et al*., 2001)

Table 1: Initial data of layout objects

	Number																		
Parameter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Shape	R	R	R	R	R	R	R	R	R	R	R	R	C	R	R	R	R	R	R
Removability (Y/N)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
Mass (kg)	1.8	10.7	1.1	3.0	1.1	12.7	4.0	7.0	8.0	4.5	2.5	2.0	2.0	21.0	1.2	3.5	3.0	14.0	0.8
Length/Radius (mM)	190	302	140	135	140	302	250	250	200	174	180	160	150	444	444	140	140	296	296
Breadth/Radius (mM)	185	210	130	110	130	210	205	200	170	96	150	150	150	475	475	211	211	475	475
Height (m)	92.5	297	95	100	95	317	145	270	240	158	142	170	361	350	400	201	180	400	400
x (mM)	-130	133.5	430	420	430	163.5	-155	-130	130	302.5	430	256	0	-282.6	-282.6	430	430	161.4	161.4
y (mM)	350	355.5	340	-100	-265	-355	-355	-300	-410	-340	-111	375	390	0	0	105	105	0	0
z (mM)	-81.25	183.5	82.5	85	-82.5	193.5	-107.5	170	155	-114	-106	-120	-215.5	210	-235	135.5	-125	235	-235

R: Cuboid. C: Cylinder. Y: Object is removable. N: Object is immovable. There are some errors in Table 1 of Huo *et al.* (2005), and correction as follows: the length and breadth of No. 16 and No.17 are 140, 211 and 140, 211; the coordinates z of No.1, No.5, No.7, No.8 and No.9 are -81.25, -82.5, -107.5, 170 and 155

subproblem respectively. According to the centroid coordinate z, objects are divided into two subpopulations (P_1 or P_2).

The fitness function, which is used to evaluate the layout scheme, can be described as:

$$\Phi(X) = w_1\lambda_1 F(X) + w_2\lambda_2 M(X) + w_3\lambda_3 S(X) + w_4\lambda_4 R(X) \quad (3)$$

where, w_1, w_2 and w_4 denote the weight coefficients of subfunctions; w_3 denotes the punishment coefficient of constraints; λ_1, λ_2, λ_3, λ_4 are the normalization factors of subfunctions and constraints. Here, the layout design of the satellite module is converted to an optimization problem without behavioral constraints using punishment coefficients method. For this problem, the value of $\Phi(X)$ is smaller, the layout scheme is better.

Gaussian mutation used here is N(0,1). The mutation factor f and probability factor J are set as follows:

$$f = \begin{cases} 0.95 \times (\text{Gmax} - \text{G}) / \text{Gmax}, & \text{if } (f > 0.5) \\ 0.5, & \text{else} \end{cases} \quad (4)$$

$$J = 0.6 \times (\text{Gmax} - \text{G}) / \text{Gmax} \quad (5)$$

where, Gmax denotes the maximum of generation; G denotes the number of current generation.

Formula (4) shows f decreases with the increase of G. In early stage of evolution, larger f could enhance the exploration capability of the algorithm. In late evolution, smaller f conduces to the convergence of the algorithm. Also, formula (5) shows J decreases with the increase of G. Larger J could enhance the exploration capability of the algorithm, but it hinders the convergence of the algorithm in late evolution.

Termination criteria: Preset the value of maxNum. maxNum denotes the maximum number of $\Phi(X)$ evaluations. When the evaluation number reaches maxNum, the evolution terminates. The experiment is determined to be success, if the layout scheme satisfies the constraints and is less than the permissible values of F and M. Otherwise, the experiment is a failure.

`In order to compare the performance of algorithms, OCCDEG, DE (Storn and Price, 1997), DEG (Chen *et al*., 2008), CCDE (Shi *et al*., 2005) and CGDE (Shi, 2005) are all used to optimize the instance above. All of algorithms are real-coded and implemented in ANSI C language. The platform is Microsoft Visual Studio 2008. The CPU and memory of experimental computer are Core2 Duo 2.20 GHz and 2 GB. The following parameters of each algorithm are the same. Strategy: rand/1/bin; crossover factor CR=0.9; the coefficients (set empirically) in $\Phi(X)$:

$w_1\lambda_1$= 0.2, $w_2\lambda_2$= 0.2, $w_3\lambda_3$= 0.2, $w_4\lambda_4$= 1.05; population size: N = 100, subpopulation size: $N_{P1} = N_{P2}$ = 50; max Num = 1×10^5. Each experiment was run 50 times with different random seeds.

EXPERIMENTAL RESULTS AND DISCUSSION

Table 2 shows dynamical non-equilibrium force (F for short), dynamical non-equilibrium moment (M for short), enveloping circle radius (envelop objects, R for short) and overlap area (among objects and the module wall) of optimal layout scheme calculated by each algorithm. Table 3 lists the statistic results of each algorithm independently run 50 times respectively.

As Table 2 shows, compared with DE (Storn and Price, 1997), DEG (Chen *et al*., 2008), CCDE (Shi *et al*., 2005) and CGDE (Shi, 2005), the OCCDEG obtained the best layout scheme according to the criteria. The dynamical non-equilibrium Force (F) and Moment (M) of OCCDEG decrease by 78.3 and 54.6% than CGDE. The average performance of each algorithm is compared in Table 3. With the comparison between DE and DEG and comparison between CCDE and CGDE, we can see that the introduction of Gaussian mutation improves the solution accuracy of DE and CCDE. Because the random disturbances caused by Gaussian mutation enhance the local search capability of algorithm. Besides, the average value of fitness function of OCCDEG decreases by 1.14% than CGDE; the average radius of enveloping circle of OCCDEG decreases by 1.20% than CGDE; the average dynamical non-equilibrium moment of OCCDEG decreases by 2.06% than CGDE; the average overlap area of OCCDEG approximates to CGDE. Moreover the performance of OCCDEG is more stable for its smaller standard deviation than CGDE. But the average dynamical non-equilibrium force of OCCDEG is inferior to CGDE and the success rate of OCCDEG is lower than CGDE and CCDE.

Figure 3 shows the curve of the best individual fitness during the evolution for OCCDEG, DE (Storn and Price, 1997), DEG (Chen *et al*., 2008), CCDE(Shi *et al*., 2005) and CGDE (Shi, 2005). It can be seen that the convergence rates of cooperative coevolution algorithms (CCDE, CGDE and OCCDEG) are larger than non-coevolution algorithms (DE and DEG). It is the evidence of that the divide and conquer strategy of CCEA framework decreases the complexity of the

Table 2: The best results of each algorithm for criteria of the layout scheme

Algorithm	R(mm)	F(N)	M(N·m)	Overlap area (mm^2)
DE (Storn and Price, 1997)	557.698	0.769	6.925	0
DEG (Chen *et al*., 2008)	557.698	0.387	7.245	0
CCDE (Shi *et al*., 2005)	557.698	0.033	2.030	0
CGDE (Shi, 2005)	557.698	0.023	3.617	0
OCCDEG	557.698	0.005	1.642	0

Notes: the calculation time of each algorithm in table 2. is less than 10 seconds for maximum number (maxNum) of Ô(*X*) evaluations

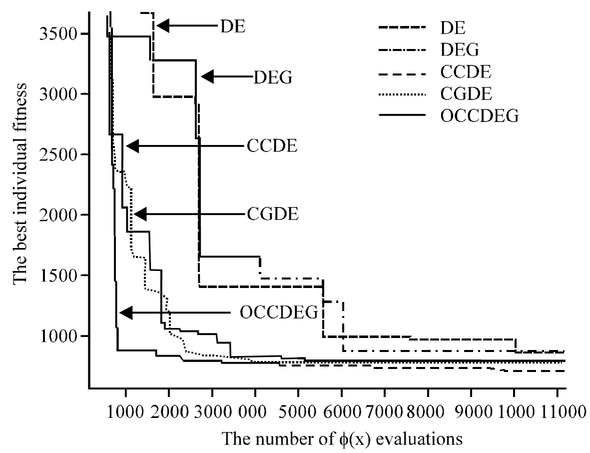

Fig. 3: The curve of the best individual fitness (the smaller the better)

Table 3: The statistic results of each algorithm for running 50 times

	Values of Φ (X)				R (mm)				Overlap area (mm^3)			
Algorithm	Mean	Best	Worst	SD	Mean	Best	Worst	SD	Mean	Best	Worst	SD
DE (Storn and Price, 1997)	692.180	591.727	810.121	60.478	648.859	557.698	759.196	55.120	0.921	0	20.829	3.753
DEG (Chen *et al*., 2008)	667.954	589.209	792.521	58.243	628.356	557.698	745.561	54.924	0.707	0	26.201	3.726
CCDE (Shi *et al*., 2005)	617.137	585.996	681.351	32.432	585.542	557.698	646.632	31.511	0.013	0	0.299	0.046
CGDE (Shi, 2005)	613.253	586.311	680.484	35.593	582.171	557.698	646.670	29.558	0.012	0	0.487	0.070
OCCDEG	606.255	586.371	680.533	24.687	575.168	557.698	646.645	23.916	0.022	0	0.478	0. 071

	F (N)				M (N·m)				
Algorithm	Mean	Best	Worst	SD	Mean	Best	Worst	SD	Success rate
DE (Storn and Price, 1997)	42.287	0.219	162.725	35.314	11.180	0.906	42.583	7.528	0.160
DEG (Chen *et al*., 2008)	29.392	0.229	101.577	24.390	10.803	0.958	27.079	6.063	0.180
CCDE (Shi *et al*., 2005)	2.942	0	41.662	8.236	8.634	0.709	16.375	3.827	0.740
CGDE (Shi, 2005)	1.815	0	36.162	5.983	8.041	0.735	15.434	3.527	0.880
OCCDEG	3.747	0	34.557	9.252	7.875	0.302	13.709	3.114	0.680

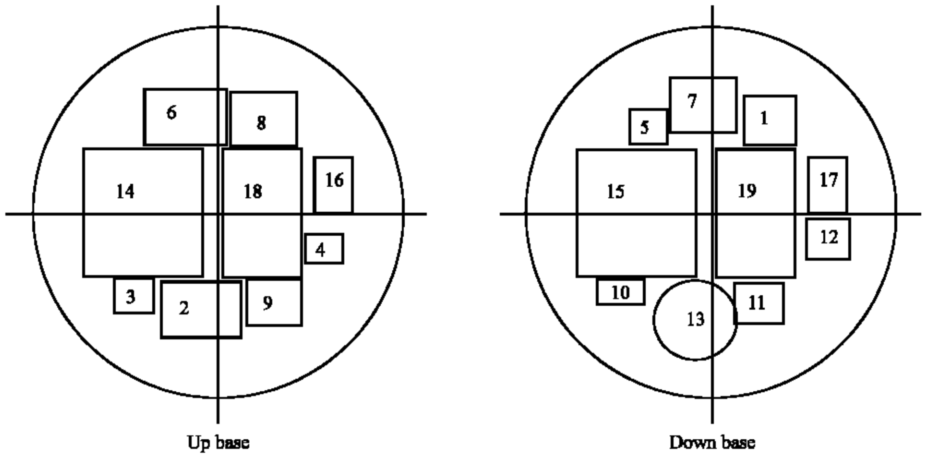

Fig. 4: The best layout scheme of DE

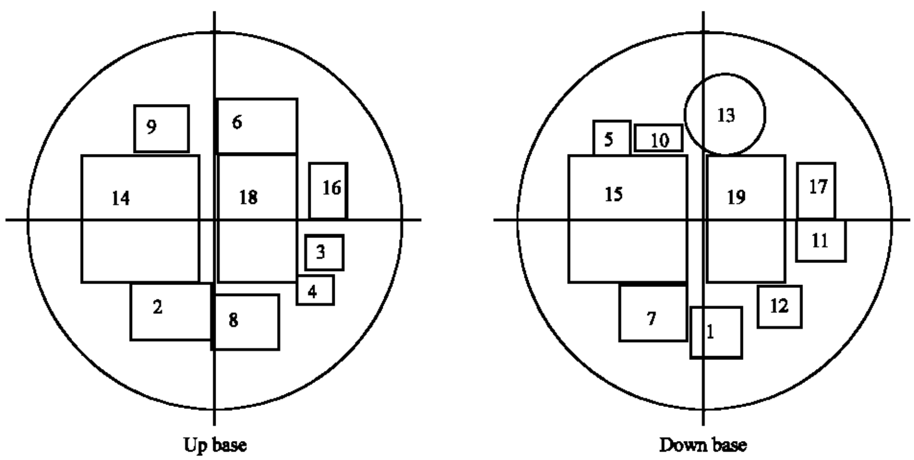

Fig. 5: The best layout scheme of DEG

problem and promotes the convergence of algorithms. For OCCDEG and CGDE, at the beginning of evolution, the convergence rate of OCCDEG is larger than CGDE. This shows that the OBL enhances the exploration ability of algorithm. With the opposite individuals, the diversity of exploration directions of whole population is increased. So the algorithm can find more potential regions. In late stage of evolution, with a narrower range of exploration the opposite individuals play the role of local search. So the exploitation capability of the algorithm is enhanced to some extent. It should be noted that, in early evolution, the individual elimination rules of the OBO would decrease the diversity of whole population. That may lead the population into a trap of premature and then experiment fails.

According to the results and comparisons above, it's obvious that the decomposition strategy of CCEA framework is suitable for solving such decomposable problems. So the cooperative coevolution algorithm obtains better results (compare CCDE (Shi *et al*., 2005) with DE (Storn and Price, 1997). Besides, the random disturbance brought by Gaussian mutation enhances the local search of algorithms and promotes algorithms to find more excellent layout design (compare CGDE (Shi, 2005) with CCDE (Shi *et al*., 2005)). What's more, with the help of OBL, the algorithm could achieve the goal more quickly and obtain a better result (compare OCCDEG with CGDE (Shi, 2005)). However, the possibility of premature brought by OBL needs to be considered. The strategies used in OCCDEG also are promising to be used for improving other evolutionary algorithms for the layout optimization problems.

The best layout schemes of all algorithms are as follows. Figure 4 shows the best layout scheme of DE.

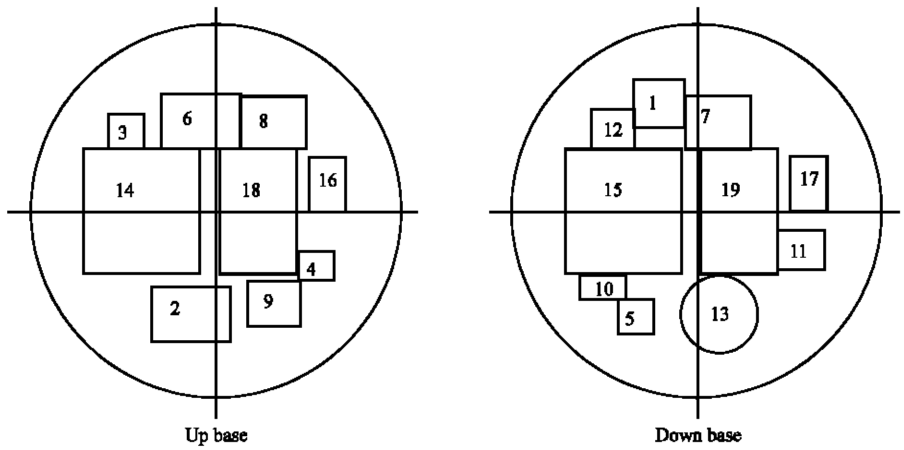

Fig. 6: The best layout scheme of CCDE

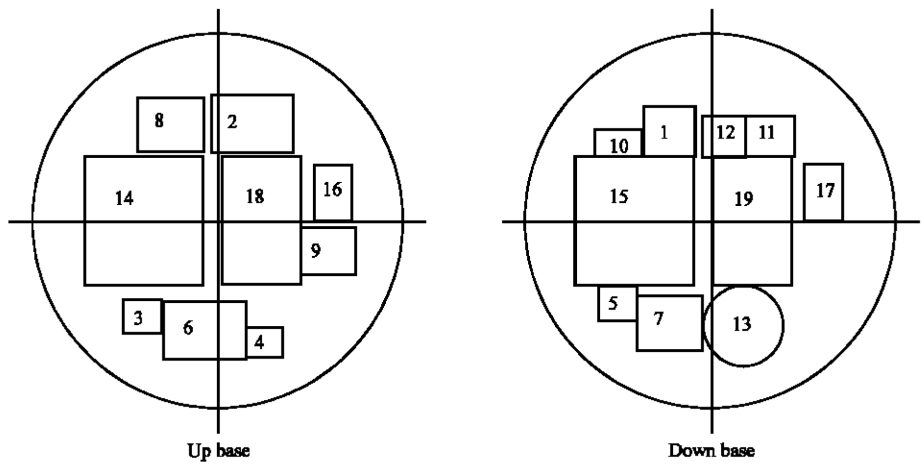

Fig. 7: The best layout scheme of CGDE

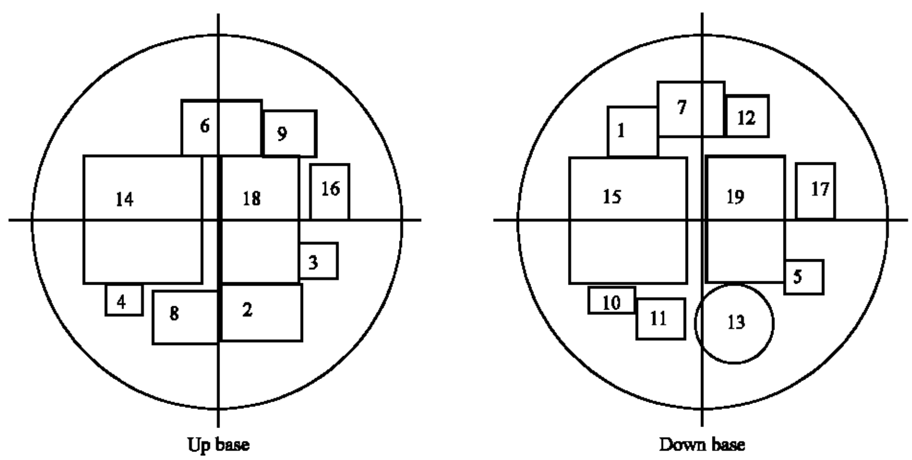

Fig. 8: The best layout scheme of OCCDEG

Figure 5 shows the best layout scheme of DEG. Figure 6 shows the best layout scheme of CCDE. Figure 7 shows the best layout scheme of CGDE. Figure 8 shows the best layout scheme of OCCDEG.

CONCLUSION

This paper proposed the OCCDEG algorithm for layout optimization of simplified satellite module. The experimental results show that, compared with other algorithms in this paper, OCCDEG obtained the better calculation results. The strategies (CCEA framework, OBL and Gaussian mutation) adopted by this paper effectively improve the performance of the algorithm. It can be inferred that OCCDEG is a promising algorithm for the decomposable layout optimization problem.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of P. R. China (Grant Nos. 51005034 and 50975033).

REFERENCES

Abbaspour, R.A. and F. Samadzadegan, 2009. Optimum crop planning using multi-objective differential evolution algorithm. J. Applied Sci., 9: 3804-3812.

Adeyemo, J.A. and F.A.O. Otieno, 2009. Multi-objective differential evolution algorithm for solving engineering problems. J. Applied Sci., 9: 3652-3661.

Aladahalli, C., J. Cagan and K. Shimada, 2007. Objective function effect based pattern search-theoretical framework inspired by 3D component layout. J. Mech. Design, 129: 243-254.

Ataei, A., M.H. Panjeshahi and M. Gharaie, 2009. A new algorithm for optimum design of mechanical draft wet cooling towers. J. Applied Sci., 9: 561-566.

Braun, R.D., I.M. Kroo and A.A. Moore, 1997. Collaborative approach to launch vehicle design. J. Spacecr Rockets, 34: 478-486.

Cagan, J., D. Degentesh and S. Yin, 1998. A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout. Comput. Aided Design, 30: 781-790.

Chen, W., Y.J. Shi and H.F. Teng, 2008. An improved differential evolution with local search for constrained layout optimization of satellite module. Adv. Intel. Comput. Theor. Appl. Aspects Artificial Intel., 5227: 742-749.

Grignon, P.M. and G.M. Fadel, 2004. A GA based configuration design optimization method. J. Mech. Design, 126: 6-15.

Huang, W. and M. Chen, 2006. Note on: An improved algorithm for the packing of unequal circles within a larger containing circle. Comput. Ind. Eng., 50: 338-344.

Huo, J.Z., G.Q. Li, H.F. Teng and Z.G. Sun, 2005. Human-computer cooperative ant colony/genetic algorithm for satellite module layout design. Chinese J. Mech. Eng., 41: 112-116.

Jansen, T. and R.P. Wiegand, 2004. The cooperative coevolutionary (1+1) EA. Evol. Comput., 12: 405-434.

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

Liu, Z.C., X.F. Lin, Y.J. Shi and H.F. Teng, 2011. A micro genetic algorithm with cauchy mutation for mechanical optimization design problems. Inform. Technol. J., 10: 1824-1829.

Ning, L., L. Fei, S. Debao and H. Chang, 2004. Particle swarm optimization for constrained layout optimization. Fifth World Congr. Intel. Control Automation, 3: 2214-2218.

Otieno, F.A.O. and J.A. Adeyemo, 2010. Strategies of differential evolution for optimum cropping pattern. Trends Applied Sci. Res., 5: 1-15.

Peng, L. and Y. Wang, 2010. Differential evolution using uniform-quasi-opposition for initializing the population. Inform. Technol. J., 9: 1629-1634.

Potter, M.A. and K.A.D. Jong, 1994. A cooperative coevolutionary approach to function optimization. Parallel Problem Solving Nat. III, 866: 249-257.

Rahnamayan, S., H.R. Tizhoosh and M.M.A. Salama, 2006. Opposition-based differential evolution algorithms. Proceeding of the IEEE Congress on Evolutionary Computation, July 16-21, 2006, Vancourver, Canada, pp: 2010-2017.

Shi, Y., H. Teng and Z. Li, 2005. Cooperative co-evolutionary differential evolution for function optimization. Lecture Notes Comuput. Sci., 3611: 1080-1088.

Shi, Y.J., 2005. The cooperative co-evolutionary differential evolution and its application for complex layout optimization. Dalian University of Technology, Dalian, China.

Soltani, M.A., A. Karimi, M.R. Bazargan-Lari and E. Shirangi, 2008. Stochastic multi-purpose reservoir operation planning by scenario optimization and differential evolutionary algorithm. J. Applied Sci., 8: 4186-4191.

Storn, R. and K. Price, 1997. Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces. J. Global Optimization, 11: 341-359.

Teng, H.F., S.L. Sun, D.Q. Liu and Y.Z. Li, 2001. Layout optimization for the objects located within a rotating vessel-a three-dimensional packing problem with behavioral constraints. Comput. Operat. Res., 28: 521-535.

Tizhoosh, H., 2005. Opposition-based learning: A new scheme for machine intelligence. Proc. Int. Conf. Comput. Intel. Modeling Control Autom, I: 695-701.

Wang, L.J., X.L. Sun and Z.K. Shi, 2009a. New evolutionary algorithm applying to a type of facility location problem. Inform. Technol. J., 8: 605-609.

Wang, Y.S., H.F. Teng and Y.J. Shi, 2009b. Cooperative co-evolutionary scatter search for satellite module layout design. Eng. Comput., 26: 761-785.

Xiao, R.B., Y.C. Xu and M. Amos, 2007. Two hybrid compaction algorithms for the layout optimization problem. BioSystems, 90: 560-567.

Zhang, B., H.F. Teng and Y.J. Shi, 2008. Layout optimization of satellite module using soft computing techniques. Applied Soft Comput., 8: 507-521.