INFORMATION TECHNOLOGY JOURNAL
Opposition-based Cooperative Coevolutionary Differential Evolution Algorithm

With Gaussian Mutation for Simplified Satellite Module Optimization

Yuan-Hui Wang, Xiu-Kun Wang and 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 (OCCDEBG) 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, OCCDEBG 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; Gignon 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; Solanet 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 Wiegard (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: 86-411-84799136 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 (OCDEG) was proposed here to solve the layout design problem of simplified satellite module.

THE OCDEG 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 $j$th individual of the $S$th subpopulation $X_{i_S}$ and cooperators from other subpopulations form a whole solution. The value of the solution calculated by evaluation function is the fitness of $X_{i_S}$.

In this paper, two subpopulations ($P_1$ and $P_2$) coevolve together. $N$ is the total number of individuals. $N_{i_1}$, $N_{i_2}$ are the numbers of individuals in subpopulations ($N_i = N/2$). $P_1$ 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 a subproblem. $D_{i_1}$ and $D_{i_2}$ denote the dimension number of a subproblem respectively ($D_{i_1}=D_{i_2}$=$D/2$).

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

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

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. $\bar{X}(x_1, x_2, \ldots, x_D)$ is the opposite-based vector of $X(x_1, x_2, \ldots, x_D)$. If $f(\bar{X}) < f(X)$, 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 $P_{i_0}$ randomly ($S$ is subpopulation number, $G$ is subpopulation generation. Here $G = 0$)
- Create the opposite subpopulation $OP_{i_0}$: Each element of individuals in $OP_{i_0}$: $\tilde{x}_j = a_j + b_j - x_j (i = 1, 2, \ldots, N_j; j = 1, 2, \ldots, D)$, where $N_j$ denotes the individual number in subpopulation $P_{i_0}$; $D$ denotes the dimension number of subpopulation $P_{i_0}$; $x_j$ and $\tilde{x}_j$ denote the $j$th dimension variable of $i$th individual in $P_{i_0}$ and $OP_{i_0}$ respectively
- Select $N$ fittest individuals from $\{P_{i_0}, OP_{i_0}\}$ 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 $P_{i_0}$ generates new subpopulation $P_{i_0+1}$ by mutation, crossover and selection
- If $Rand(0, 1) < J$, create the opposite subpopulation $OP_{i_0+1}$: Each element of individuals in $OP_{i_0+1}$: $\tilde{x}_j = MIN + MAX \cdot x_j (i = 1, 2, \ldots, N_j; j = 1, 2, \ldots, D)$, where $Rand(0, 1)$ is the uniform distribution between 0 and 1; $J$ is the probability of whether to create the opposite subpopulation; $MIN$ and $MAX$ denote the maximum and minimum values of the $j$th element of individuals in current subpopulation, respectively
- Select $N$ fittest individuals from $\{P_{i_0+1}, OP_{i_0+1}\}$ as the new subpopulation
Fig. 1: The flow chart of OCCDEG

Here, the calculation of individual fitness of subpopulation $P_{S0+1}$ and $OP_{S0+1}$ 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 each element of trial individual $X'_i = [x'_{i1}, x'_{i2}, ..., x'_{iN}] (i = 1, 2, ..., N)$ which is generated by mutation and crossover of classical DE:

$$x'_{ij} = x_{ij} + \sigma \cdot N(0, 1) \quad j = 1, 2, ..., D$$

(2)

where, $N(0, \sigma)$ denotes Gaussian random number, $\sigma_j$ denotes the mutation step-size of $j$th 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: $\omega = 40$ r/min, $R_e = 700$ mm, $R_s = 1000$ mm, $R_i = 600$ mm, $H_e = 1400$ mm, $H_s = 800$ 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
The fitness function, which is used to evaluate the layout scheme, can be described as:

$$
\phi(X) = w_1 f_1(X) + w_2 f_2(X) + w_3 f_3(X) + w_4 f_4(X)
$$

where, \( w_i \), \( w_j \), and \( w_k \) denote the weight coefficients of subfunctions; \( w_i \) denotes the punishment coefficient of constraints; \( \lambda_1 \), \( \lambda_2 \), \( \lambda_3 \), and \( \lambda_4 \) 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 \( \mathcal{N}(0,1) \). The mutation factor \( f \) and probability factor \( J \) are set as follows:

$$
f = \begin{cases} 
0.95 \times \frac{G_{max} - G}{G_{max}}, & \text{if } f > 0.5 \\
0.5, & \text{else}
\end{cases}
$$

$$
J = 0.6 \times \frac{G_{max} - G}{G_{max}}
$$

where, \( G_{max} \) 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 \( f \) conducive 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 Core 2 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 \( \Phi(X) \):
$w_1 \lambda_1 = 0.2$, $w_2 \lambda_2 = 0.2$, $w_3 \lambda_3 = 1.05$; population size: $N = 100$, subpopulation size: $N_p = N_q = 50$; max Num $\approx 1 \times 10^7$. 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 CCDE, 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**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean (mm)</th>
<th>Best (mm)</th>
<th>Worst (mm)</th>
<th>SD (mm)</th>
<th>Overlap area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE (Storn and Price, 1997)</td>
<td>557.698</td>
<td>557.698</td>
<td>557.698</td>
<td>0.132</td>
<td>739,06</td>
</tr>
<tr>
<td>DEG (Chen et al., 2008)</td>
<td>557.698</td>
<td>557.698</td>
<td>557.698</td>
<td>0.132</td>
<td>739,06</td>
</tr>
<tr>
<td>CCDE (Shi et al., 2005)</td>
<td>557.698</td>
<td>557.698</td>
<td>557.698</td>
<td>0.132</td>
<td>739,06</td>
</tr>
<tr>
<td>CGDE (Shi, 2005)</td>
<td>557.698</td>
<td>557.698</td>
<td>557.698</td>
<td>0.132</td>
<td>739,06</td>
</tr>
<tr>
<td>OCCDEG</td>
<td>557.698</td>
<td>557.698</td>
<td>557.698</td>
<td>0.132</td>
<td>739,06</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$\Phi$ (X)</th>
<th>R (mm)</th>
<th>Overlap area (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Best</td>
<td>Worst</td>
</tr>
<tr>
<td>DE (Storn and Price, 1997)</td>
<td>692.180</td>
<td>591.727</td>
<td>810.121</td>
</tr>
<tr>
<td>DEG (Chen et al., 2008)</td>
<td>667.954</td>
<td>589.209</td>
<td>792.521</td>
</tr>
<tr>
<td>CCDE (Shi et al., 2005)</td>
<td>617.137</td>
<td>585.996</td>
<td>681.351</td>
</tr>
<tr>
<td>CGDE (Shi, 2005)</td>
<td>613.253</td>
<td>586.311</td>
<td>680.484</td>
</tr>
<tr>
<td>OCCDEG</td>
<td>606.255</td>
<td>588.371</td>
<td>680.533</td>
</tr>
</tbody>
</table>

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


