International Journal of Hybrid Information Technology
Vol.8, No.4 (2015), pp. 147-156
http://dx.doi.org/10.14257/ijhit.2015.8.4.17

Research on Improved Differential Evolution Algorithm based on Hybrid Multi-strategy and its Application

Bin Gao, Jing-Hua Zhu and Wen-chang Lang

Wenzhou Key Laboratory of Material Processing and Die & Mould Technology, Wenzhou Vocational & Technical College, Wenzhou 325035, PR China

Abstract

In order to improve the problem of premature convergence and computational efficiency of traditional differential evolution algorithm in solving high-dimensional problems, an improved differential evolution (HMSDE) algorithm based on combing elite synergy strategy, multi-population strategy and dynamic adaptive strategy is proposed in this paper. In the proposed HMSDE algorithm, the population is dynamically divided into multi-populations in order to keep the diversity of the population, elite synergy strategy is used to achieve information exchange among different sub-populations, and dynamic adaptive strategy is used to dynamically control the parameter values of scaling factor and crossover factor in order to improve the stability and robustness of the HMSDE algorithm. In order to test the performance of the HMSDE algorithm, a set of 10 benchmark functions are selected in here. The results show that the HMSDE algorithm takes on remarkable optimized ability, faster convergence speed and higher search accuracy. And the HMSDE algorithm can avoid the premature convergence and outperforms several state-of-the-art performances.

Keywords: *Differential evolution; Hybrid multi-strategy; Elite synergy; Dynamic adaptive control; Complex function*

1. Introduction

Optimization is an applied technology in the fields of industry, agriculture, defense, transportation, and management and so on, which explores the best values for solving complex problem that can take under specified conditions [1]. The aims are to enable an objective function to generate the minimum or maximum value. So there proposed a lot of direct or heuristics methods to solve these complex optimization problems [2]. For example, the well known direct search methods or stochastic methods such as genetic algorithm (GA) [3-4], simulated annealing method (SA) [5], particle swarm optimization (PSO) [6], differential evolutionary algorithm [7-8], ant colony optimization algorithm [9], artificial bee colony algorithm [10], and so on in the past few years.

Differential evolution (DE) algorithm is proposed by Storn and Price in 1995, is a very popular evolutionary algorithm and exhibits remarkable performance in a wide variety of problems from diverse fields. The DE algorithm is a population-based and stochastic global optimizer. The DE algorithm achieves the optimization search by performing the mutation operator operation, crossover operation and selection operation at each generation among the current individuals. The DE algorithm can randomly, parallel and efficiently implement the global optimization. Like other evolutionary algorithms, the DE algorithm exists some deficiencies, such as premature convergence, local optimum and low search efficiency and so on. In recent years, some strategies and methods are proposed in order to choose trial vector generation strategies and control parameter settings during the last decade [11-16]. Fan and Lampinen [11] proposed a trigonometric mutation operator to accelerate convergence of the DE algorithm. Their mutation operator can be viewed as a local search operator, since it moves the new trial vector towards the

ISSN: 1738-9968 IJHIT

direction provided by the best one of three individuals chosen for mutation. Feoktistov and Janaqi [12] classified mutation operators into four categories according to the way they use the objective function values. Babu and Angira [13] proposed a modification to original DE that enhances the convergence rate without compromising on solution quality. The modified differential evolution (MDE) algorithm utilizes only one set of population as against two sets in original DE algorithm at any given point of time in a generation. Ronkkonen *et al.* [14] suggested that NP should be between 2D and 4D, $F \in [0.4, 0.95]$ with F =0.9 being a good trade-off between convergence speed and robustness, and $CR \in [0, 0.2]$ for separable functions, $CR \in [0.9, 1.0]$ for multimodal and non-separable functions. Das *et al.* [15] introduced two schemes to adapt the scaling factor F in the DE. One scheme varies F in a random manner, and the other one linearly reduces the value of F from a preset maximal value to a minimal one. Teo [16] investigated the population sizing problem via self-adaptation and proposed two different approaches, one adopts an absolute encoding strategy for NP, and the other adopts a relative encoding strategy for NP.

These improved DE algorithms overcome the premature convergence, local optimum and low search efficiency for solving complex problem. But these improved DE algorithms have not yet systematically exploited in DE algorithm design, exists slow exploitation ratio, different selecting parameters. The convergence speed is further strengthened. So In order to further improve the DE algorithm, elite synergy strategy, multi-population strategy and dynamic adaptive strategy are introduced in to the DE algorithm in order to propose an improved differential evolution (HMSDE) algorithm based on hybrid multi-strategy in this paper.

The rest of this paper is organized as follows. Section 2 introduces the differential evolution algorithm. Section 3 presents an improved differential evolution algorithm named HMSDE algorithm. Section 4 compares our experimental results with the recent algorithms that have been used to solve the benchmark test functions. Finally, the conclusions are discussed in Section 5.

2. Differential Evolution

Differential Evolution (DE) is a relatively recent heuristic algorithm. It is a simple yet powerful population-based, direct search algorithm with the generation-and-test feature for solving complex optimization problems in the continuous domains using real-valued parameters. This algorithm can create new candidate solutions by combining parent individual and several other individuals in the same population. The parent is replaced by a candidate only if the candidate has better fitness. It has been shown that this algorithm is not only very effective, but also very speed and robust for obtaining a minimal variability of results from one generation to another generation. The DE algorithm has mutation operation, crossover operation and selection operation. It takes advantage of the weighted and random vector to mutate vectors in each iteration. The DE algorithm is evolving until the optimal solution is obtained or the iteration achieves the given maximum value.

The different variants of DE algorithm will be classified by the notation $DE/\alpha/\beta/\delta$. The α is a method to choose parent chromosome. The β is a number of difference vectors. The δ is the recombination mechanism. In actual application, some mutation and selection strategies are proposed in order to improve the convergence speed and solving accuracy for standard DE algorithm. There are several schemes that are proposed[10]. In many literatures, the $DE/rang/1/bin$ is often selected most commonly to realize the variants and is also employed in this paper. The pseudo-code of the $DE/rang/1/bin$ scheme is shown in Table 1.

Table 1. The Pseudo-code of the Standard DE Algorithm

The DE algorithm with $DE/rand/1/bin$ scheme
Generate the initial population P
Compute the fitness value for each individual in P
while the halting criterion is not met **do**
for $i=1$ to NP **do**
Select randomly uniform $r_1 \neq r_2 \neq r_3 \neq i$
$j_{rand} = rnd\,\text{int}(1, D)$
for $j=1$ to D **do**
if $rndreal_j[0,1) < CR \; or \; j == j_{rand}$ **then**
$U_i(j) = X_{r_1}(j) + F \times (X_{r_2}(j) - X_{r_3}(j))$
else
$U_i(j) = X_{i_1}(j)$
end if
end for
end for
for $i=1$ to NP **do**
Compute the individual value U_i
if U_i is better value than P_i **then**
$P_i = U_i$
end if
end for
end while

In the Table 1, NP is the size of parent population P , D is the number of variable dimensionality, CR is the probability of crossover operator, F is the mutation scaling factor, $X_i(j)$ is the j^{th} variable of the solution X_i , U_i is the individual, $rndreal_j[0,1)$ is a uniformly distributed random real number in [0,1), $rnd\,\text{int}(1, D)$ is a uniformly distributed random integer number between 1 and n.

1. Initial Population

The key parameters of the DE algorithm are initialized in first. These parameters include the size of population(NP), the probability of crossover operator(CR),the mutation scaling factor(F),the number of iteration(T).The upper bound($x_{j\max}$) and lower bound($x_{j\min}$) of each variable are initialized by using the uniformly distributed probability, i.e., $x_i = (x_1, x_2, \cdots, x_D)$. The initial population is obtained:

$$x_{ji}(0) = x_{j\min} + Rand(0,1) \times (x_{j\max} - x_{j\min}) \; (j = 0,1,2,\cdots,D) \tag{1}$$

2. Mutation

There are many differential strategies, which are proposed to achieve mutation for each target vector. The each target vector is composed of different individuals (x_{g2}, x_{g3}). For each target vector x_i , a mutant vector v_i is defined by the following express:

$$v_i = x_{g1} + F \cdot (x_{g2} - x_{g3}) \tag{2}$$

where $i = 1,2,3,\cdots,$ $g1, g2, g3 \in [1, D]$ are mutually different random integer number, which are come from the running index i. The mutation scaling factor(F) is a real number and constant factor, $F \in [0,2]$. It is used to control the amplification of the differential value $(x_{g2} - x_{g3})$.

3. Crossover

The essence of crossover is to obtain an offspring by executing uniform crossover among the obtained individuals. A binomial cross method is used to execute crossover operation in order to generate a trial vector $u_i = (u_{i1}, u_{i2}, \cdots, u_{iD})$ according to the following equation.

$$u_{ij} = \begin{cases} v_{ij} & if\ rand[0,1]_j \le CR \vee j == j_{rand} \\ x_{ij} & otherwise \end{cases} \tag{3}$$

where $j = 1,2,3,\cdots,D,$ j is the variable of the j^{th}, D is variable dimension. And $rand[0,1]_j$ is a uniform random number between 0 and 1. And $j_{rand} \in [1, D]$ is randomly selected in order to guarantee that the trial vector u_i obtains at least one parameter from the mutated vector v_i.

4. Selection

In the selection, a greedy selection standard is used to compete the trial vector u_i and the target vector x_i according to their fitness values after the mutation operation and crossover operation. As far as a minimization problem is involved, the selected vector is given:

$$x_i^* = \begin{cases} u_i & f(u_i) < f(x_i) \\ x_i & otherwise \end{cases} \tag{4}$$

where $f(x)$ is the objective of a solution x. The x_i^* is a parent vector x_i that is used to replace the target vector x_i in the next generation. When the fitness value of trial vector u_i is better than the fitness value of target vector x_i, the trial vector u_i is selected as the offspring, otherwise, target vector x_i is selected as the offspring.

3. An Improved Differential Evolution (HMSDE) Algorithm

The DE algorithm is an efficient and powerful evolutionary algorithm and takes on some advantages of the simplicity, speed and robustness. But actual application in solving high-dimensional problems, it has some disadvantages of slow exploitation solution, difficult selecting parameters, the premature convergence and low computational efficiency and so on. For solving high-dimensional problems, the different strategies and parameter values are selected in different iterations. There have some different strategies which are proposed in recent ten years. But each strategy has a different function. Some strategies take on strong global search ability, the others take on strong local search ability, the faster convergence speed and precision convergence. Some strategies can balance the global search ability and local search capability among populations. At the same time, control parameters of DE algorithm have great influences to the optimal results.

In order to improve the optimization ability of DE algorithm and overcome the problem of easy local optima of heuristic algorithm, some new strategies of elite synergy strategy, multi-population strategy and dynamic adaptive strategy are introduced into DE algorithm in order to improve the performance of the algorithm. So an improved differential evolution (HMSDE) algorithm based on making full use of these new strategies is proposed in this paper. In the HMSDE algorithm, the population is dynamically divided into multi-populations according to the fitness values of the individuals in order to maintain the diversity of the population. The elite synergy strategy is used to enhance the local developing ability and convergence speed, and achieve information exchange among different sub-populations. The dynamic adaptive strategy is used to adaptively control the scaling factor and crossover factor in order to guarantee better robustness and accuracy and decrease the search time. The elite synergy strategy, multi-population strategy and dynamic adaptive strategy are used to improve the standard DE algorithm to obtain the HMSDE algorithm with remarkable optimized ability, faster convergence speed and higher search accuracy.

According to the idea of HMSDE algorithm, the HMSDE algorithm can effectively avoid the premature convergence and outperform several state-of-the-art performances. The flow of HMSDE algorithm is shown in Figure 1.

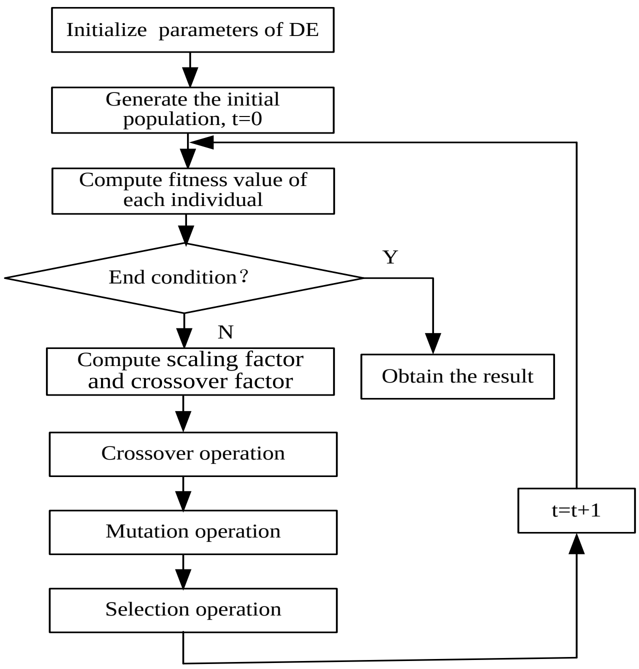

Figure 1. The Flow of HMSDE Algorithm

4. Numerical Experiment

In order to validate the effectiveness of the proposed HMSDE algorithm, eight benchmark test functions are selected to test the performance. These test cases include various types of objective functions with different number of decision variables. The functions $f_1 - f_4$ are unimodal functions and functions $f_5 - f_8$ are multimodal function where the number of local minima increases exponentially with the problem dimension. The operating environment is: i5-4200U Intel 64bit processors with a core frequency of 1.8GHz with 4GB memory on Windows 7 operating system, Matlab 7.8. The parameters are set: population size NP = 30, the function dimension is 30, the maximum evolution generation T_{max} =1000. For each test case, 20 independent runs are performed in Matlab. CR and F are randomly generated within[0.95,1.0] and [0.9,1.0] , respectively. The expression and variables range of eight benchmark functions are reported in table as shown in Table 2.

Table 2. Eight Benchmark Functions

Index	Function expression	Optimum	Variables range
f_1	$f_1(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$	0	$\lvert x_i \rvert \le 30$
f_2	$f_2(x) = \sum_{i=1}^{D} x_i^2$	0	$\lvert x_i \rvert \le 100$
f_3	$f_3(x) = \sum_{i=1}^{D} \lvert x_i \rvert^2 + \prod_{i=1}^{D} \lvert x_i \rvert$	0	$\lvert x_i \rvert \le 100$
f_4	$f_4(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_i)^2$	0	$\lvert x_i \rvert \le 30$
f_5	$f_5(x) = \frac{1}{40000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	0	$\lvert x_i \rvert \le 60$
f_6	$f_6(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	0	$\lvert x_i \rvert \le 5.12$
f_7	$f_7(x) = \sum_{i=1}^{D} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	0	$\lvert x_i \rvert \le 100$
f_8	$f_8(x) = -20\exp[\sum_{i=1}^{D} \frac{x_i^2}{n}] - \exp(\sum_{i=1}^{D} \cos(2\pi x_i)/n) + 20 + e$	0	$\lvert x_i \rvert \le 32$

The performance of HMSDE algorithm is compared with other published versions of standard DE algorithm and SaDE [17] algorithm. For each test case, all algorithms were run 20 times and the best value, worst value, mean value and standard deviation value of the results are reported in table as shown in Table 3. The best value and worst value are used to describe the solution quality. The mean value is used to describe the achievable accuracy of the HMSDE algorithm in the given times of function evaluation and reflect the convergence speed. The standard deviation value is used to reflect the stability and robustness of the HMSDE algorithm.

Table 3. The Experiment Results for Eight Benchmark Test Functions

Index	Algorithm	Worst	Best	Mean	Std
f_1	**DE**	3.021 15e-12	5.214 27e-14	6.346 25e-13	3.382 25e-14
	SaDE	5.348 23e-20	4.417 34e-22	6.326 37e-21	6.347 74e-21
	HMSDE	4.435 72e-28	0.000 00e+00	8.356 53e-25	4.932 46e-26
f_2	**DE**	3.643 27e-13	4.157 58e-14	4.142 68e-13	4.346 79e-12
	SaDE	6.106 49e-16	7.645 35e-18	5.638 32e-17	5.134 61e-17
	HMSDE	3.626 63e-28	6.665 83e-29	7.482 34e-29	6.856 17e-12

f_3	**DE**	5.341 15e-08	7.279 46e-10	8.943 14e-08	3.143 24e-08
	SaDE	7.325 31e-11	5.052 16e-12	9.547 49e-11	2.318 66e-09
	HMSDE	4.621 52e-10	3.632 87e-11	7.429 18e-10	1.063 34e-09
f_4	**DE**	4.125 54e-06	9.134 48e-09	8.167 13e-07	3.041 24e-05
	SaDE	4.362 15e-10	2.472 13e-12	5.304 80e-10	1.143 39e-09
	HMSDE	1.816 17e-15	8.241 15e-18	1.030 27e-16	4.057 31e-13
f_5	**DE**	2.723 16e-10	5.146 29e-14	3.085 24e-11	6.335 12e-09
	SaDE	4.351 34e-13	2.631 38e-16	2.357 39e-14	2.937 34e-10
	HMSDE	1.462 79e-18	3.428 75e-20	8.452 11e-19	1.352 15e-17
f_6	**DE**	7.301 58e-07	4.432 12e-10	1.437 31e-08	4.312 17e-07
	SaDE	4.317 23e-12	7.532 17e-15	8.005 36e-13	4.064 43e-10
	HMSDE	2.643 29e-20	3.356 15e-23	1.432 35e-21	3.349 14e-16
f_7	**DE**	2.665 38e-10	6.346 19e-13	1.452 23e-11	2.453 95e-09
	SaDE	3.309 49e-16	5.879 74e-18	4.452 19e-17	4.542 18e-13
	HMSDE	4.643 16e-14	5.981 43e-17	7.436 88e-15	6.331 45e-12
f_8	**DE**	8.409 31e+00	2.660 96e-02	3.922 03e-02	8.385 08e-03
	SaDE	5.941 03e-09	3.524 24e-11	6.159 32e-10	3.903 21e-10
	HMSDE	1.341 28e-13	4.314 53e-15	7.006 14e-14	6.435 17e-12

It can be seen from Table 4 that with the same preset maximum number of iterations, the proposed HMSDE algorithm can obtain better “Best value” and “Mean value” on the 8 benchmark test functions at 30 dimensions except the function f_3 and f_7, and better “Standard deviation value” on the 8 benchmark test functions at 30 dimensions except the function f_2, f_3 and f_7. And the obtained solutions are very close to the global optima for benchmark test function f_2, f_4, f_5 and f_6. On the benchmark test function f_3 and f_7 at 30 dimensions, although the SaDE algorithm is better than the proposed HMSDE algorithm, the HMSDE algorithm outperforms the SaDE algorithm in standard deviation value. But the proposed HMSDE algorithm is superior to the standard DE algorithm in all benchmark test functions at 30 dimensions.

Summarizing the above statements, the proposed HMSDE algorithm significantly outperforms the SaDE algorithm and the standard DE algorithm on 6 out of 8 benchmark test functions at 30 dimensions, respectively. The SaDE algorithm wins only in two benchmark test functions (*i.e.* function f_3 and f_7). In a word, the proposed HMSDE algorithm obtained the better optimization performance than the DE algorithm and the SaDE algorithm for solving benchmark test functions. In a word, the proposed HMSDE algorithm can offer the higher accuracy than both SaDE and DE in most of the test functions.

5. Conclusion

Differential evolution algorithm is an efficient and powerful optimization algorithm, which widely applied in the industry, agriculture, defense, transportation, and management. For the problem of premature convergence and computational efficiency of traditional differential evolution algorithm in solving high-dimensional problems, many researchers used different trial vector generation strategies to control parameter settings of the DE algorithm in the literature. An improved differential evolution (HMSDE) algorithm based on combing elite synergy strategy, multi-population strategy and dynamic adaptive strategy is proposed in this paper, represented one of the first attempts along this direction. In the proposed HMSDE algorithm, the population is dynamically divided into multi-populations in order to keep the diversity of the population , elite synergy strategy is used to achieve information exchange among different sub-populations, and dynamic adaptive strategy is used to dynamically control the parameter values of scaling factor and crossover factor in order to improve the stability and robustness of the HMSDE algorithm. The experimental studies were carried out on eight benchmark test functions in this paper. The HMSDE algorithm was compared with the standard DE algorithm and SaDE algorithm. The experimental results that the proposed HMSDE algorithm takes on remarkable optimized ability, faster convergence speed and higher search accuracy. And the HMSDE algorithm can avoid the premature convergence and outperforms several state-of-the-art performances.

Acknowledgements

This research was supported by Zhejiang Provincial Natural Science Foundation of China (LQ12E01003) and Zhejiang Provincial Technical Plan Project (2012C21099).

References

[1] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, vol. 214, no.12, (**2009**), pp.108-132.
[2] J. Jing, J. C. Zeng and C. Z. Han, "An extended mind evolutionary computation model", Applied Mathematics and Computation, vol.185, no. 2, (**2007**), pp. 1038-1049.
[3] D.E. Goldberg, "Genetic algorithms in search, optimization & machine learning", Addison-Wesley, (**1989**)
[4] D. Bunnag and M. Sun, "Genetic algorithm for constrained global optimization in continuous variables", Applied Mathematics and Computation, vol. 171, no.1, (**2005**), pp. 604-636.
[5] M. Ji, Z. Jin and H. Tang, "An improved simulated annealing for solving the linear constrained optimization problems", Applied Mathematics and Computation, vol. 183, no. 1, (**2006**), pp. 251-259.
[6] M. Clerc and J. Kennedy, "The particle swarm—explosion, stability, and convergence in a multidimensional complex space", IEEE Transactions on Evolutionary Computation, vol. 6, no.1, (**2002**), pp. 58-73.
[7] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol.11, no.4, (**1997**), pp. 341-359.
[8] S. Das and P. Suganthan, "Differential evolution: a survey of the state-of-the-art", IEEE Transactions on Evolutionary Computation, vol. 15, no.1, (**2009**), pp. 4-31.
[9] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 15, no.1, (**1997**), pp. 53-66.
[10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm", Journal of Global Optimization, vol. 39, no. 3, (**2007**), pp. 459-471.
[11] H. Y. Fan and J. Lampinen, "A trigonometric mutation operator to differential evolution," Journal of Global Optimization, vol. 27, no. 1, (**2003**), pp. 105-129.
[12] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," in Proc. of the 18th Int Parallel and Distributed Processing Symposium. Santa Fe, (**2004**), pp. 165-170.
[13] B. V. Babu and R. Angira, "Modified differential evolution (MDE) for optimization of non-linear chemical processes," Computers & Chemical Engineering, vol. 30, no. 6-7, **(2006)**, pp. 989-1002.
[14] J. Ronkkonen, S. Kukkonen and K. V. Price, "Real parameter optimization with differential evolution," in Proc. IEEE Congress on Evolutionary Computation, (CEC), vol. 1, (**2005**) , Piscataway, NJ: IEEE Press, pp. 506-513.
[15] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in Proc. IEEE Congress on Genetic Evolutionary Computation. (GECCO), (**2005**), pp. 991-998.
[16] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," Soft Computing, vol. 10, no. 8, (**2006**), pp. 637-686.
[17] A. K. Qin, V.L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," IEEE Transactions on Evolutionary Computation, vol. 13, no.2, (**2009**), pp. 398-417.

Authors

Bin Gao, Associate Researcher, he received the Doctor degree in mechanical engineering from Zhejiang University in 2006, Hangzhou, China. The main research directions: optimization design, CAE, physical vapor deposition.

Jing-Hua Zhu, Lecturer, he received the Engineering Master degree in computer software from University of Electronic Science and Technology of China in 2011, Chengdu, China. The main research directions: CAD&CG.

Wen-chang Lang, Assistant Researcher, he received the Doctor degree in materials processing engineering from Chinese Academy of Sciences in 2010, Shenyang, China. The main research directions: physical vapor deposition