Journal of Computational Information Systems 9: 15 (2013) 6171–6181
Available at http://www.Jofcis.com

# An Adaptive Chaotic Differential Evolution Algorithm for Layout Optimization with Equilibrium Constraints

Qingyun YANG*, Minghao WANG, Bing QIAO

*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*

**Abstract**

Circles, two dimensions with equilibrium and noninterference constraints, packing into a circular container is a NP-hard constrained layout optimization problem. That has broad application in engineering. Classical Differential Evolution (DE) for solving these problems is readily falling into local optima. An adaptive chaotic DE algorithm is proposed to improve the performance in this paper. The weighting parameters are changed dynamically with chaotic mutation in the searching procedure. The penalty factors of the cost function are modified during iteration. To keep the diversity of the population, we limit the population's concentration. To enhance the local search capability we adopt adaptive mutation of the global optimal individual. The improved algorithm can maintain the basic algorithm's structure as well as extend the search scope, and can hold the diversity of population as well as increase the search precision. Also, our improved algorithm can escape from premature and speedup the convergence. Experiments indicate the feasibility and efficiency of our algorithm.

*Keywords*: Differential Evolution; Constrained Optimal Layout; Chaos; NP-hard Problem

# 1 Introduction

Layout optimization problem has wide application, such as container packing, spacecraft interior layout and integrate circuit layout design, etc. It investigates how to distribute geometrical objects (graph units) with different shape and size in a certain space, at the same time those geometrical objects should satisfy some certain constraints. Layout problem belongs to combination optimization, and is a NP-hard problem. According to layout dimension, layout problem partitions to 2 dimensions and 3 dimensions. According to complexity, it partitions to performance-free constrained layout problem and performance-related constrained layout problem (we call it constrained layout problem in brief). Performance-free constrained layout problem, such as container packing, concentrates on maximize the space utility without interference among graph units or between graph units and the container. However, performance-related constrained layout problem needs consider the satisfaction of object inertia, equilibrium or stability besides the satisfaction of

*Corresponding author.
*Email address:* qingyunyang77@gmail.com (Qingyun YANG).

1553–9105 / 
DOI: 10.12733/jcisP0383
August 1, 2013

performance-free constrained layout problem. Circular constrained layout problem is a particular circumstance of 2 dimension performance-related constrained layout problem. It concerns placing circles with different or same size into a circular container meanwhile the circles can interfere each other and the same do the circles and the edge of the container. Furthermore, the center positions of circles need near the center positions of the container, at the same time, the inequilibrium of the whole system need reach the minimization.

Teng [1] proposed a mathematic model concerning dishes installed on rotating table, which was based on the background of rough layout optimization of returned satellite cabin. And a deterministic algorithm was proposed to solve these small scale problems. When the number of cell to be distributed is increased, it is difficult to search the optimized layout since the layout forms increase dramatically. In this situation, Teng and his cooperators proposed a series of evolution algorithms to solve these problems and gained good effects, such as improved genetic algorithm [2], human-machine interaction based genetic algorithm [3, 4], immune genetic algorithm [5, 6] and particle swarm optimization based cultural algorithm model [7]. Furthermore, other algorithms like improved particle swarm optimization [8-10], differential evolution [11] and ant colony optimization [12] were proposed by other literatures.

An adaptive chaotic differential evolution algorithm is proposed in this paper to overcome the plunging into local optima of traditional differential evolution algorithm. Dynamically permute the weight parameters by chaotic mutation, which is used to improve the algorithmic performance that affects by parameters. A method to keep the diversity of population is proposed to enhance the global search capacity. To improve the local search capacity, a strategy of mutation of the global best individual is adopted. To accelerate the convergence of our algorithm, an approach is designed that dynamical changes the penalty factor of cost function. Experimental results indicate the proposed algorithm can avoid precocious convergence, accelerate the convergence speed, and gain good results.

This paper is organized as follows. In section 2 we will review the definition and mathematic model of equilibrium constraint layout problem. We proposed the adaptive chaotic differential evolution algorithm in section 3, detailed description are discussed, followed by the experimental evaluation in section 4. The conclusion is given in section 5.

# 2 Problem Description and Mathematic Model

When placing graph units with different size into a rotating circular container, the centroids of these graph units are the unknown to be optimized on supposing that the center of mass coincides with the centroid of graph unit. Assuming there are $n$ different size circles with radius $(r_1, r_2, \cdots, r_n)$ and mass $(m_1, m_2, \cdots, m_n)$. The circular container has radius $R$, the centre of a circle is original coordinate $(0, 0)$, rotating at speed $\omega$. Now distributing these $n$ graph units into the container at the same time as near as possible to the center of the container. That is a $2n$ vector $X = (x_1, y_1, \cdots, x_n, y_n)^T \in R^{2n}$ determines a unique layout solution. Here $(x_i, y_i)$ represents the centroids of the $i^{th}$ graph unit. Also some constraints should be satisfied: (1) no interference exists among these graph units; (2) all graph units are contained in the container; (3) after distribution, the inequilibrium of the system is less than a fixed value $[\delta_J]$.

The general mathematic model is to solve $X$ so that it minimizes the maximal circumcircle radius, i.e.

$$\min F(X) = \max\left\{\sqrt{(x_i^2 + y_i^2)} + r_i\right\}, i \in I = \{1, 2, \cdots, n\}$$

s.t.

$$r_i + r_j \le \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, i \ne j;\ i, j \in I$$

$$\sqrt{(x_i^2 + y_i^2)} + r_i - R \le 0,\ i \in I$$

$$\sqrt{\left(\sum_{i=1}^{n} m_i x_i\right)^2 + \left(\sum_{i=1}^{n} m_i y_i\right)^2} - [\delta_J] \le 0,\ i \in I$$

# 3 Adaptive Chaotic Differential Evolution

Differential Evolution (DE) is a heuristic randomized search optimization algorithm in continuous space proposed by Rainer and Kenneth Price in 1995 [13]. As an excellent optimization algorithm, DE has outstanding representation in function optimization, and has broad application. The framework of DE is based on genetic algorithm (GA). Differential operation is designed for genetic individual's real number encoding, and is used to implement cross and mutation in genetic algorithm. According to different evolving strategy, DE can be presented as:

$$DE/x/y/z$$

Where $x$ is a random selected individual or the best individual in each iteration, $y$ is the number of differential vector, and $z$ is the cross mode. In this paper, we use $DE/rand - to - best/1/exp$

$$X_i = X_i + F_1 \times (X_{gbest} - X_i) + F_2 \times (X_{p1} - X_{p2})$$

Where $p1, p2 \in \{1, 2, \cdots, N\}$ are random numbers, and $N$ is the population size.

## 3.1 Adaptive chaotic mutation based parameters tuning

Main parameters in DE include population size $N$, mutation factor $F$ and cross factor $CR$. Different value choice of $F$ and $CR$ bring vast different performance in evolving procedure. Big $F$ parameter results in a random search for DE, thus it destroys the best solution and lower converge to global best solution. However, small $F$ reduces the diversity of population and induces premature of local optima. In contrast, when $CR$ is big, it promotes the local search capacity and accelerates the convergence while small $CR$ benefits the diversity of population and global search capacity as well.

Generally, $N$ is located in $(5 \times D, 10 \times D)$, where $D$ is the dimension. $F \in (0, 2)$, in most examples, $F = 0.5$ sounds be a nice choice, and usually $CR$ is restricted in $[0, 1]$, better choice is $CR = 0.3$ [14].

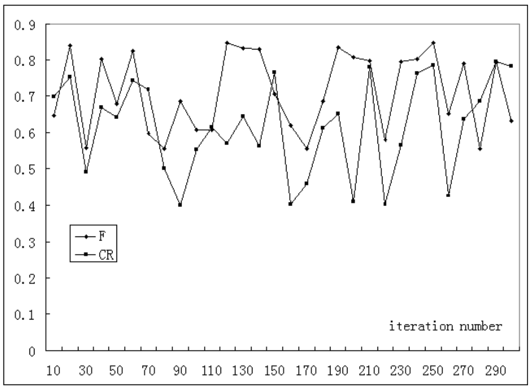


Fig. 1: The curve of $F$ and $CR$ when $F_1 = 0.7$, $F_2 = 0.3$, $CR_1 = 0.6$, $CR_2 = 0.4$

To improve the performance, variational parameters are used in evolving, such as adaptive parameters [15]. Chaotic signal has the properties of traversal and randomness. These properties can be utilized as an optimizing mechanism avoiding local optima to improve search capacity in a search procedure. Logistic equation [15] is a classic chaotic system:

$$z_{n+1} = \mu z_n (1 - z_n) \qquad n = 0, 1, 2, \cdots \qquad z_n \in [0, 1]$$

Where $\mu$ is control variable. Arbitrary $z_0 \in [0, 1]$, if $\mu$ is fixed, can iterate a determinate time series $z_1, z_2, z_3, \cdots$.

Let $rf$ and $rcr$ are chaotic variables, the parameters in DE can be rewriten as

$$F = F_1 + F_2 \times \left( rf - \frac{1}{2} \right)$$

$$CR = CR_1 + CR_2 \times \left( rcr - \frac{1}{2} \right)$$

When $F_2 = CR_2 = 0$, the above formula is degraded to parameter $F$ and $CR$ of standard DE. Empirical results indicate that $F_1 = 0.7$, $F_2 = 0.3$, $CR_1 = 0.6$, $CR_2 = 0.4$ sounds a good choice.

## 3.2 Improve the diversity adaptively

Since greedy search strategy is the dominant approach in DE, the convergence is fast meanwhile it increases the probability of premature. Eventually, all individuals trend to the global best individual in the evolving procedure. If the global best individual is not the global optimized solution, it is called premature. The essential cause of premature is that a "cluster" phenomenon occurs in searching procedure i.e. the diversity of population decreases. Good search strategy should maintain the diversity at the early stage to perform global search. However, at the late stage, we should strengthen the local search capacity to improve the accuracy.

Many methods are proposed to maintain the diversity, such as concentration [16], population activity criteria [17], fitness variance [18]. Concentration is also a criteria of "cluster". Higher concentration means more similar individuals or less diversity, and vice versa. It is naturally use concentration to enlarge the diversity of population in evolving procedure.

The concentration $c_v$ of individual v is defined as follows, and $c_v \in (0, 1]$

$$c_v = \left( \frac{1}{M} \sum_{w=1}^{M} A(v, w) \right)^{\left(1 - \frac{it}{\max It}\right) \cdot \gamma}$$

where $it$ and $maxit$ are the current iteration and maximum iteration respective. $\gamma$ is a known parameter, and $\gamma = 0.5$ , $A(v, w)$ is the affinity between individual $v$ and individual $w$. Since in DE, individual is encoded in vector, i.e. $v = (v_1, v_2, \cdots, v_n)$, $w = (w_1, w_2, \cdots, w_n)$, then

$$A(v, w) = 1 / \left( 1 + \sqrt{\sum_{i=1}^{N} (v_i - w_i)^2} \right)$$

$A(v, w)$ converges towards 1 when individual $v$ and individual $w$ becomes more similar, otherwise, $A(v, w)$ trends to 0. Comparing with traditional concentration definition in Immune Algorithm, the proposed equation has an extra power term, which guarantees $c_v$ converge to 1 and avoiding vibration in late evolving stage. Moreover, the proposed equation can calculate and implement easily.

After the calculation of concentration, a conventional selection operator is used.

$$p_v = \frac{c_v}{\sum\limits_{v=1}^{N} c_v}$$

The selection operator guarantees higher concentrating individual has higher selected probability to mutate, and vice versa. The used mutation operator is defined in the following subsection.

## 3.3 Mutation of best individuals

In the late period of search procedure, the diversity of population is declined. The best individual (global or local) will maintain unchanged in a considerable time. When individuals become identity by analyzing the formulation of DE, the mutation and cross operator can not bring new direction for population, which results in local optima. Here, a maximum consecutive unchanged counter, $gmax$, is used as criteria to perturb the best individual.

$$best'_k = \xi(best_k), k = 1, 2, \cdots, D$$

$$\xi(x) = \begin{cases} x & t < g\max \\ \zeta(x) & t \geq g\max \quad \& \quad P < P_0 \end{cases}$$

Where $t$ is an unchaged counter for best individuals, if it changes, $t$ is set to zero. $\zeta(x)$ is mutation function, it can be an unique random function or a Logistic equation, in this paper,

$\zeta(x)$ $N(x, \sigma)$ ($\sigma$ is the range of $x$). $P$ is the mutation probability and $P_0$ is the threshold of mutation probability. Empirical results indicate $P_0 \in \left[\frac{1}{D}, \frac{2}{D}\right]$ is a good choice.

After modification, the update of each iteration is:

$$X_i = X_i + F_1 \times (\xi(X_{gbest}) - X_i) + F_2 \times (X_{p1} - X_{p2})$$

$gmax$ determines the mutation frequency of the best individual. Small $gmax$ means more frequent mutation, thus the best individual has more powerful global search capacity with lower convergence speed. While larger $gmax$ let DE trend towards basic operator, i.e. fast convergence speed with weak search capacity, which results in local optima. When $gmax$ is equal to maximum iteration, it degrades to standard DE operator.

### 3.4 Modification of cost function

According to the definition of constrained layout problem, the cost function and constraint function as follows:

$$\phi(X_i) = \lambda_0 F(X_i) + \sum_{k=1}^{3} \lambda_k f_k(X_i)$$

Where $f_k(X_i)$ is the $k^{th}$ constraint, $\lambda_i, i \in \{0, 1, 2, 3\}$ is penalty factor. Smaller cost function value means better solution.

Empirical results [9] indicated that ($\lambda_0 = 1, \lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 0.01$) was a good choice. This parameter setting can maintain the satisfaction of the constraints subjected to the cost function. But some researches figured out that algorithm can accelerate convergence in certain degree when the object function violates its constraints properly. In this paper, we use the following cost function:

$$\phi(X_i) = \begin{cases} \left[1 + \alpha\left(1 - \frac{it}{\max It}\right)\right] F(X_i) + \sum\limits_{k=1}^{3} \lambda_k f_k(X_i), \\ \qquad it \le \beta \cdot \max It \\ \lambda_0 F(X_i) + \sum\limits_{k=1}^{3} \lambda_k f_k(X_i), otherwise \end{cases}$$

where $\alpha, \beta$ are known parameters, in this paper, $\alpha = 2.5$, $\beta = 0.8$. $It$ is the iteration number, $maxIt$ is the maximal iteration number.

# 4 Experimental Results

4 examples are test to evaluate the performance of our algorithm. All these examples are taken from the literatures and the comparison between the best so far result and our proposed algorithm. The computing time is scaled to the same measure. All parameter settings are used in the above sections.

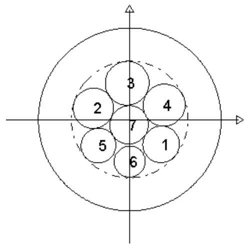


Fig. 2: Layout result in Lit. [9]

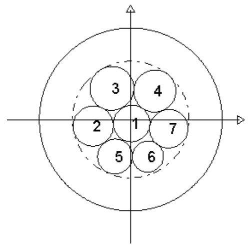


Fig. 3: Our result

Table 1: The radius and mass of each graph unit in experiment 1 and the comparison layout coordinates results of these graph units. r(mm) is radius in mm, m is the mass in g, the last two columns if the layout solution (center coordination of graph units) with our algorithm. The center two columns are results from literature [9]

| No. | r(mm) | m(g) | x(mm) | y(mm) | x(mm) | y(mm) |
|---|---|---|---|---|---|---|
| 1 | 10.0 | 100.0 | 17.15 | -13.59 | 0.47 | -2.18 |
| 2 | 11.0 | 121.0 | -19.89 | 6.14 | -20.51 | -3.57 |
| 3 | 12.0 | 144.0 | -1.40 | 19.84 | -10.30 | 17.00 |
| 4 | 11.5 | 132.0 | 18.81 | 7.84 | 13.16 | 15.56 |
| 5 | 9.5 | 90.25 | -17.12 | -14.26 | -8.63 | -20.24 |
| 6 | 8.5 | 72.25 | -0.03 | -23.02 | 9.37 | -20.29 |
| 7 | 10.5 | 110.25 | -0.38 | -2.95 | 20.77 | -5.08 |

Table 2: The comparison results of performance with literature [9]. r(mm) is Circumcircle radius in mm, u(g.mm) is Unequilibrium in g.mm, i(mm) is Interference in mm, time means computing time in second

| Method | r(mm) | u(g.mm) | i(mm) | time (s) |
|---|---|---|---|---|
| Lit. [9] | 31.885 | 0.000002 | 0 | 982 |
| Our Alg. | 31.841 | 0 | 0 | 12 |

Experiment 1 [1]. A circular container with radius $R = 50mm$, which has 7 circular objects (graph unit), supposing static unequilibrium $J$ has $\delta_J = 3.4g.mm$. Other data of graph units and results are listed in Table 1, Table 2 and Figure 1 respectively.

In table 1 and table 2, we can see that circumcircle radius, unequilibrium and computing time gained by our algorithm are all superior to literature [9]. The circumcircle radius reduces 0.019% while computing time decreases 92.77%, and the unequilibrium reaches to zero in our algorithm.

Experiment 2 [2]. Circular container with radius $R = 880mm$, containing 40 graph units. Supposing static unequilibrium $J$ has $\delta_J = 20g.mm$. Other data of graph units and results are listed in table 3, table 4 and figure 2.

From the results of example 2, we can see that radius of circumcircle reduces 2.27% and computing time of our algorithm declines 25.83% in our algorithm comparing with literature [11],

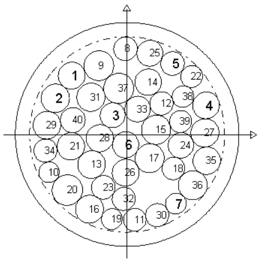

Fig. 4: Lit. [11]

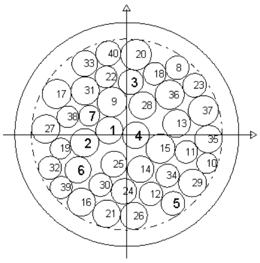

Fig. 5: Our result

Table 3: The comparison results of performance with literature [11]

| Method | r(mm) | u(g.mm) | i(mm) | time (s) |
|---|---|---|---|---|
| Lit. [11] | 768 | 0.0007 | 0 | 573 |
| Our Alg. | 750.541 | 0.004 | 0 | 425 |

while static enequilibrium increases 471.43%.

Example 3 [4]. A known optimized solution layout problem. It concerns placing 9 graph units into a $75mm$ radius circular container. Supposing the center coordinate, radius and mass are $(x_i, y_i)$, $r_i$, $m_i$ respectively, where $r_i = 30mm$, $i = 2, 3, 4, 5$; $r_i = 30\left(\sqrt{2}-1\right)mm$, $i = 1, 6, 7, 8, 9$. In this example, assume $m_i = r_i$. Detailed data are listed in table 5.

From Table 5 and Table 6, we can see that the radius of circumcircle reduces 0.48% and computing time reduces 28.35% in our algorithm. Meanwhile the static unequilibrium and interference are reach zero in our algorithm.

Example 4 [9]. Another known optimized solution layout problem. Assuming distributing 5 graph units into a circular container with radius $R = 125mm$. Detailed data are listed in table 7.

From the example 4, we can see that the consistent results of circumcircle in literature[9] and our algorithm. static unequilibrium is zero in our algorithm while it is near zero in literature [9]. The interference are zero in both algorithms. However, the computing time costs in our algorithm reduced 77.95%.

From these experiments, no matter circumcircle radius, static unequilibrium, interference and computing performance, we can see that the results produced by our algorithm are encouraging

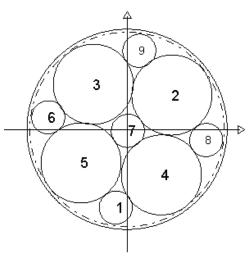

Fig. 6: Lit. [7]

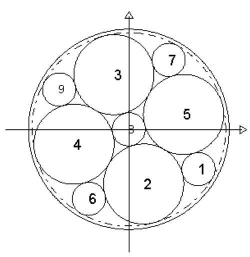

Fig. 7: Our result

Table 4: The radius and mass of each graph unit in experiment 3 and the comparison layout coordinates results of these graph units. The center two columns are results from literature [7]

| No. | r(mm) | m(g) | x(mm) | y(mm) | x(mm) | y(mm) |
|---|---|---|---|---|---|---|
| 1 | 12.43 | 12.43 | -8.40 | -58.63 | 50.05 | -29.84 |
| 2 | 30.00 | 30.00 | 33.56 | 25.69 | 11.11 | -40.95 |
| 3 | 30.00 | 30.00 | -25.34 | 33.75 | -11.11 | 40.95 |
| 4 | 30.00 | 30.00 | 25.76 | -34.15 | -40.95 | -11.11 |
| 5 | 30.00 | 30.00 | -34.48 | -25.14 | 40.95 | 11.11 |
| 6 | 12.43 | 12.43 | -59.04 | 9.06 | -29.84 | -52.05 |
| 7 | 12.43 | 12.43 | 0.59 | -1.12 | 29.84 | 52.05 |
| 8 | 12.43 | 12.43 | 59.26 | -8.01 | 0.00 | 0.00 |
| 9 | 12.43 | 12.43 | 9.14 | 58.90 | -52.05 | 29.84 |

Table 5: The comparison results of performance with literature [7]

| Method | r(mm) | u(g.mm) | i(mm) | time (s) |
|---|---|---|---|---|
| Lit. [7] | 72.7764 | 8.1855 | 4.4751 | 83.74 |
| Our Alg. | 72.4264 | 0 | 0 | 60 |

Table 6: The radius and mass of each graph unit in experiment 4 and the comparison layout coordinates results of these graph units. The center two columns are results from literature [9]

| No. | r(mm) | m(g) | x(mm) | y(mm) | x(mm) | y(mm) |
|---|---|---|---|---|---|---|
| 1 | 20.71 | 20.71 | 0.00 | 0.00 | -0.00 | -0.00 |
| 2 | 50.00 | 50.00 | -60.00 | -37.42 | -43.68 | -55.61 |
| 3 | 50.00 | 50.00 | 60.00 | 37.42 | 55.61 | -43.68 |
| 4 | 50.00 | 50.00 | -37.42 | 60.00 | 43.68 | 55.61 |
| 5 | 50.00 | 50.00 | 37.42 | -60.00 | -55.61 | 43.68 |

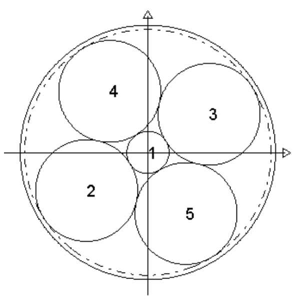


Fig. 8: Lit. [9]

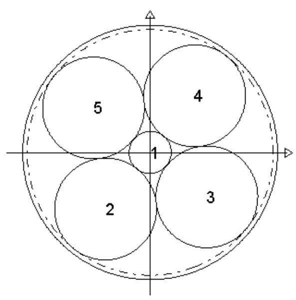


Fig. 9: Our result

Table 7: The comparison results of performance with literature [9]

| Method | r(mm) | u(g.mm) | i(mm) | time (s) |
|---|---|---|---|---|
| Lit. [9] | 120.7107 | 0.000703 | 0 | 263 |
| Our Alg. | 120.7107 | 0 | 0 | 5.54 |

and it is superior to most existed algorithms. For relatively simple layout problem, all algorithms have similar results, such as experiment 3 and experiment 4 who have known optimized solution, all existing algorithms can get nearly optimized solutions while our algorithm has better results. For more complex layout problem, our algorithm is more superior to other existing algorithms. These experiments indicate that our newly proposed algorithm is an effective approach to solve complex engineering layout problems.

# 5 Conclusion

According to the characteristic of constrained layout problem with unequal circle object, we analyzed the shortcoming of traditional differential evolution (DE) algorithm, and proposed an adaptive chaotic mutation DE algorithm. Although DE has simple model and few control parameters, the choice of parameters has great influence on algorithm performance. Traditional DE is difficult to avoid the local optima using fixed parameter setting. We proposed a chaotic mutation approach for parameter setting of DE to improve the algorithm performance. In the evolving procedure, the diversity of the population becomes smaller. We analyzed the gathering degree of the population and proposed a method to improve the diversity. This can keep certain diversity in evolution to enhance the global search capacity. When the algorithm sink into local optima, the global best individual will be fixed and result in slower convergence of the algorithm. In this paper, we proposed an approach, which can strengthen the local search capacity, to mutate the global best individual. Also an improved cost function was proposed to accelerate the convergence. Our approach is effective in solving constrained layout problem, which has fine solution, fast running time and could be helpful for complex engineering layout problems.

# References

[1] H. F. Teng, S. L. Sun, W.H Ge, et al, Layout optimization for the dishes installed on rotating table, Science in China (Series A) 37 (10). 1994, 1272-1280.

[2] F. Tang, H. F. Teng, A modified genetic algorithm and its application to layout optimization, Journal of Software 10 (10), 1999, 1096-1102.

[3] Z. Q. Qian, H. F. Teng, Z. G. Sun, Human-computer interactive genetic algorithm and its application to constrained layout optimization, Chinese Journal of Computers 24 (5), 2001, 553-559.

[4] Z. Q. Qian, H. F. Teng, D. L. Xiong, et al, Human-Computer cooperation genetic algorithm and its application to layout design, In: Proc of the 4th Asia-Pacific Conference on Simulated Evolution and Learning'02., 2002, pp. 299-302.

[5] G. Q. Li, H. F. Teng, J. Z. Huo, Parallel hybrid immune algorithm and its application to layout design, Chinese Journal of Mechanical engineering 39 (6), 2003, 79-85.

[6] G. Q. Li, H. L. Zhao, F. Q. Zhao, et al, Human-computer cooperative immune algorithm and its application to layout design, Computer Engineering 31 (21), 2005, 4-6, 27.

[7] J. B. Ai, H. F. Teng, Particle Swarm Optimization Based Cultural Algorithm Model and its Application to Layout Design.

[8] N. Li, F. Liu, D. B. Sun, A study on the particle swarm optimization with mutation operator constrained layout optimization, Chinese Journal of Computers 27 (7), 2004, 897-903.

[9] K. Y. Lei, Y. H. Qiu, A study of Constrained Layout optimization Using Adaptive Particle Swarm Optimizer, Journal of Computer Research and Development 43 (10), 2006, 1724-1731.

[10] C. Zhou, L. Gao, H. B. Gao, Particle swarm optimization based algorithm for constrained layout optimization, Control and Decision 20 (1), 2005, 36-40.

[11] J. Liu W. Q. Huang, A Modified Differential Evolution Algorithm for Solving Circles Packing Problem with Constraints of Equilibrium, Information and Control 35 (1), 2006, 103-107.

[12] Y. C. Xu, R. B. Xiao, Ant colony algorithm for layout optimization with equilibrium constrains, Control and Decision 23 (1), 2008, 25-29.

[13] R. Storn, K. Price, Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report International Computer Science Institute, Berkley, 1995.

[14] R. Storn, K. Price, Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4), 1997, 341-359.

[15] M. G. H. Omran, A. Salman and A. P. Engelbrecht, Self-adaptive Differential Evolution, in: Proc. CIS 2005, 2005, pp. 192-199.

[16] X. B. Cao, K. S. Liu, X. F. Wang, Desing Multilayer Feed-forward Networks Based on Immune Evolutionary Programming. Journal of Software 10 (11), 1999, 1180-1184.

[17] C. S. Zhang, J. G. Sun, Q. Y. Yang, et al, A hybrid algorithm for flow-shop scheduling problem, Acta Automatica Sinica 35 (3), 2009, 332-336.

[18] L. H. Wu, Y. N. Wang, X. F. Yuan, S. W. Zhou, Differential Evolution Algorithm with Adaptive Second Mutation, Control and Decision 21 (8), 2006, 898-902.

[19] K. Y. Yang, X. F. Wang, Research and implement of adaptive multimodel immune evolution algorithm, Control and Decision 20 (6), 2005, 717-720.