

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

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

## A Layout Pattern Based Particle Swarm Optimization for Constrained Packing Problems

<sup>1</sup>Shi Yan-Jun, <sup>2,3</sup>Wang Yi-Shou, <sup>1</sup>Wang Long and <sup>1</sup>Teng Hong-Fei

<sup>1</sup>School of Mechanical Engineering, Dalian University of Technology, Dalian 116024,  
People's Republic of China

<sup>2</sup>State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology,  
Dalian 116024, People's Republic of China

<sup>3</sup>Beijing Aeronautical Science and Technology Research Institute of COMAC,  
Beijing 102211, People's Republic of China

---

**Abstract:** A Layout Pattern Based Particle Swarm Optimization algorithm (LPPSO) is presented herein for solving the two-dimensional packing problem with constraints. In the optimizing process of LPPSO, some individuals are constructed according to non-isomorphic layout pattern and these individuals are added into the current population of Particle Swarm Optimization (PSO) algorithm to replace the bad individuals, the new population is created as a result. Moreover, a non-isomorphic layout pattern is constructed based on exact boundary line approach to avoid premature convergence and improve the computational efficiency. This study also discussed the basic idea, key problem and the process of the proposed LPPSO algorithm. Two examples of constrained packing problems showed that LPPSO was feasible and effective in the experiments.

**Key words:** Constrained-packing, layout optimization, isomorphic, non-isomorphic, PSO algorithm

---

### INTRODUCTION

Packing problem (Dowsland and Dowsland, 1992) is to reasonably place the equipments and devices called objects (or components) in the container to maximize the use of space while satisfying the following constraints: All the objects should be contained with no overlapping among the components. Packing problem belongs to combinatorial optimization problems and it is generally NP-Hard (Lodi *et al.*, 2002). Constrained packing problems (Teng *et al.*, 1994) are more difficult to solve than pure packing problems. To improve the ability of solving constrained packing problems by EA, the properties of the layout problem and the algorithm should be combined. One important property of layout problem is that its solution approach has a close relationship with layout pattern. Layout pattern has many different definitions. Feng *et al.* (1999) defined isomorphic and non-isomorphic layout pattern with graph theory and group theory. Li and Teng (2003) defined the layout pattern as the relationship of relative positions among objects and proposed a matrix expression approach to construct homeomorphic and non-homeomorphic layout pattern. Hongfei *et al.* (1995) discussed homeomorphic and non-homeomorphic layout pattern and gave the

hypothesis about relationship between layout pattern and combination explosion. Hongfei *et al.* (2006) presented homeomorphic and non-homeomorphic layout pattern based on complete incidence graph.

The study uses the Particle Swarm Optimization (PSO) algorithm to solve the 2D layout problem with equilibrium constraints. PSO proposed by Eberhart and Kennedy (1995) is a representative algorithm of SIAs and has been applied in wide fields, such as neural network training, engineering optimization and data mining. PSO is simple, effective and robust. However, the biggest weakness of PSO is that it is easy to get stuck in local optima and suffer from premature convergence when solving multi-peak function optimization problems and its exploitation ability at later search stage is poor.

Previous studies have successfully applied the PSO algorithm for other several problems. Chen (2011) proposed a hybrid algorithm used the PSO algorithm with Taguchi method for the dynamic optimal power flow problem. In 2010, as for the web service selection problems, Fan and Fang (2010) put forward an improvement PSO algorithm (called Niche Particle Swarm Optimization) with feasible and efficient performance in the solving process of web service selection problems. Fang *et al.* (2011) presented the Discrete Particle Swarm

---

**Corresponding Author:** Wang Yi-Shou, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024/Beijing Aeronautical Science and Technology Research Institute of COMAC, Beijing 102211, People's Republic of China

Optimization (DPSO) as a process mining method. With the characteristic of complexity and intractability, the discrete optimization problems were better solved by the simple quantum-inspired PSO algorithm in reference by Gao *et al.* (2011). Gao *et al.* (2009) enhanced the local search ability of the PSO algorithm based on the improved chaos searching strategy. Other researchers, such as Jing-Lei and Zhi-Jian (2011), Heng *et al.* (2006), Liu *et al.* (2007), Tu *et al.* (2011), Lu and Chen (2011), Ren and Zhong (2011), employed the PSO algorithms for different angles and issues with some modified strategies or methods.

As for solving the layout optimization problem with constraints, how to use the PSO has also been studied. Li and Dong (2012) proposed a layout pattern-based GA for rectangle and circle packing problem with equilibrium constraints. The proposed approach uses the case retrieval to construct non-isomorphic layout pattern. Therefore, the corresponding case database of the problem is required. This study focuses on the packing problem from the view of the layout pattern. Solving approaches for packing problems are often classified into two categories: Heuristic approaches and stochastic approaches. Heuristic approaches are used to put objects in the container in sequence according to the given rules, without calculating the interference among objects or objects and container, such as BL Method; the other category is putting all the objects in the container (supposed that it is containable) in the random initialization phase, then optimization algorithm is used for optimizing and it requires the interference calculation. Evolutionary Algorithms (EAs) or Swarm Intelligent Algorithms (SIAs) are often used (Gao *et al.*, 2009).

### ISOMORPHIC LAYOUT PATTERN AND NON-ISOMORPHIC LAYOUT PATTERN

This section firstly describes isomorphic layout pattern and non-isomorphic layout pattern. For a given layout scheme, Layout Pattern (LP) is defined as a relative position relationship among objects (Hongfei *et al.*, 2006). According to the definition, it can be seen that LP is not related to the interferences among the objects.

It is assumed that there are two layout pattern  $S_1$ ,  $S_2$ . If the relative position relationship of layout objects' centroid in  $S_1$  is the same with that of  $S_2$  (namely  $LP(S_1) = LP(S_2)$ ), the layout pattern of the  $S_1$  and  $S_2$  is defined as the isomorphic, which is denoted by

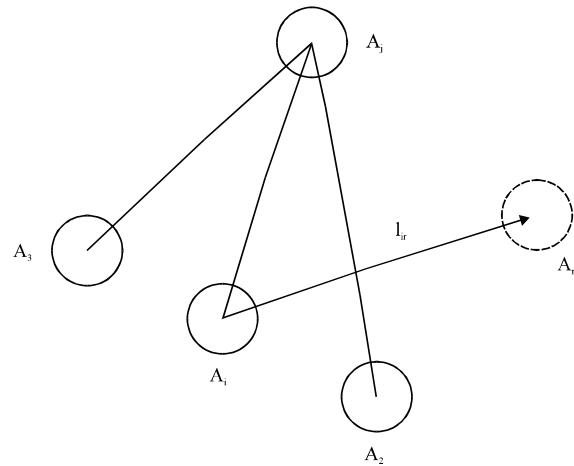


Fig. 1: Exact boundary line approach illustration,  $A_i$ : Circle layout objects,  $l_r$ : The distance between  $A_i$  and  $A_r$

$ILP(S_1, S_2)$ . Conversely, if  $LP(S_1) \neq LP(S_2)$ , the  $S_1$  and  $S_2$  are non-isomorphic, which is denoted by  $NLP(S_1, S_2)$ . According to the literature (Hongfei *et al.*, 1995), non-isomorphic layout pattern can be constructed by two approaches, i.e., the minimum polygon approach and the exact boundary line approach. In this study, the exact boundary line approach is adopted to construct ILP. The specific construct process shown in Fig. 1 can be described as follows: Firstly, two points  $A_i$  and  $A_j$  are selected randomly, then an exact boundary line  $A_i A_j$  is made in the control area of layout pattern about  $A_i$ .  $A_i$  is randomly moved to the other side of the exact boundary line  $A_i A_j$ , the new random point denoted by  $A_r$ . The distance between  $A_i$  and  $A_r$  is denoted by  $l_r$ . It should be assured that the value of  $l_r$  is less than the maximum value  $V_{max}$  of the particle velocity in PSO. A random point  $A_k$  is selected in the  $A_r$  region. Under the same constraints,  $A_k$  is moved to the other side of the exact boundary line  $A_i A_j$ . If the distance between  $A_k$  and  $A_j$  is too close or too far, it can be adjusted appropriately. Similarly,  $A_i$  can be moved to the other side of the exact boundary line  $A_i A_j$ .

### PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON LAYOUT PATTERN

When particle swarm optimization algorithm is used to solve the constrained packing problems, PSO algorithm is easy to get into a local optimum and get stagnant in the later stage (premature). To overcome above shortcomings of PSO, this study constructs non-isomorphic layout

schemes according to the current best individual obtained in the process; then the non-isomorphic schemes as individuals are added into the population to not only increase the population diversity, but also guide the exploration search in new regions with potential solutions. In this way, the PSO algorithm can jump out of the local optima and improve the computational efficiency. The above new approach is named as the Layout Pattern Based Particle Swarm Optimization algorithms (LPPSO). The basic idea of LPPSO is to construct non-isomorphic layout pattern solution using the exact boundary line approach on the basis of better individuals in an initial random layout process. Therefore, the PSO individuals would cover the solution space as large as possible, that is to say the algorithm would be started with good initial points. In the process of evolution with a certain interval (each generation or several generations), the non-isomorphic layout pattern is automatically constructed according to the best individual to replace the worse individual and help the algorithm escape from local optima easily.

### KEY STRATEGY OF LPPSO

- During initialization, the initial population is randomly generated ( $k = 0$ ), scale as  $M$ ,  $m$  ( $0 < m \ll M$ ) solutions were selected randomly from  $M$ , when  $M \leq 100$  and generally  $m = M \times 10\%$ . Calculating fitness of  $m$  random solutions and selecting the  $mL$  solutions  $P$   $mL$ , by using quasi-boundary line to selected better solutions from  $mL$  solutions and then creating  $mL$  non-isomorphic layout mode solutions (individuals) to replace randomly  $M-m$  individuals so that the population size remains as  $M$
- In the evolution of the iterative process, similarly, constructs the contemporary best individual non-isomorphic layout pattern solution to replace the generation of individual whose fitness is poorer, creating a new generation of population is better able to increase the diversity of groups and jump out of local optima and improve the layout of the problem solving efficiency and accuracy

### LPPSO STEPS OF THE ALGORITHM

This algorithm adopts the iterative formula of the GWPSO (Shi and Eberhart, 1998) and the pseudo codes are described in Fig. 2.

```

M = 50
Kmax = 500000
m = 5
step = 5
For each particle do
    Initialize particle
End for
Select m good particle
Create non-isomorphic particle based on these m particle and
replace the random particle
Do
    For each particle do
        Calculate particle velocity
        Update particle position
        Calculate fitness value
    End for
    If k%step==0 do
        Select the best and worst particle in the current
        population
        Create the non-isomorphic particle based on the
        best one to replace the worst one
    End if
    For each particle do
        If fitness < pBest do
            Save the new pBest
        If pBest < gBest do
            Save the new gBest
        End if
    End If
End for
k = k+1;
While k < Kmax

```

Fig. 2: Pseudo codes of LPPSO

### RESULTS AND DISCUSSION

**Experiment 1:** The example 1 was originated from satellite module layout optimization design and simplified as a kind of constrained packing problem with all the optimal solutions known. There were 4 big circles and 5 small ones that will be arranged in the circular container. Here the problem was to seek the optimal location of the circles such that the container has the smallest radius, while satisfying the following constraints: (1) There was no overlapping between any two circles, (2) Each circle should be include to D (3) The static non-equilibrium of the packing scheme should not exceed a permissible value, the smaller and the better. The data of all circles were described as follows: The radius of 4 big circles was all  $\sqrt{2} - 1$ , while the radius of 5 small ones was  $3 - 2\sqrt{2}$ . The mass value of each circle was equal to  $r_i^2$ . It could be confirmed that optimum solution is that the radius of enveloping circular container  $R$  is 1, static equilibrium amount  $T$  is 0. The detailed mathematical model was shown in literature (Che *et al.*, 2010).

As for the packing problems, previous studies have put forward these approach which were employed

with well performances, such as the GA (Mahfound, 1995), aiNet (De Castro and von Zuban, 2000), Clonalg (De Castro and von Zuban, 2001), DE (Storn and Price, 1997), TS (Glover, 1989), SA (Kirkpatrick *et al.*, 1983), wPSO (Clerc, 1999), xPSO (Shi and Eberhart, 2001) and GWPSO (Shi and Eberhart, 1998).

This example run on PC with 1000M RAM and CPU was Intel Core Duo T2250 1.73 GHz. This example was solved by GA (Mahfound, 1995), GWPSO (Shi and Eberhart, 1998) and LPPSO that offered in this article. Then those results were compared. Population size of those three algorithms was all 50, Learning factors of GWPSO was 2.05 and initial inertia weight was 0.9. During initialization, LPPSO algorithm of this article selected 5 individuals from 50 individuals generating randomly to evaluate fitness and got three individuals of good fitness. Then solutions of non-isomorphic layout design were structured. At the last three individuals were replaced randomly from other 45 individuals so that initial population is still 50. During the optimization process, LPPSO selected an individual with better fitness from this generation after each 10 iterations. This algorithm constructed the solutions of non-isomorphic layout design by using Quasi-boundary straight-line method. Those solutions were used to replace the worse individuals from this generation. New population was generated. This algorithm was completed by repetitions iteration like this.

For those three algorithms, 50 calculations carried through, respectively. The maximal Evaluation times of fitness ( $K_{max}$ ) is 500000. In Table 1, the average 4 optimization results were shown in which the R was the radius of the envelope circle, T was the static equilibrium

amount and  $\Delta S$  was the intervening parameter. Table 2 showed the best calculated results of the three algorithms in which the  $x_i, y_i$  is the packing circles' coordinates. Judgment conditions of whether calculation was successful are evaluation times of fitness were less than  $K_{max}$ , intervening parameter ( $\Delta S$ ) was less than 0.01 and static equilibrium amount (T) was less than 0.001. Figure 3 showed the best layout scheme of those three algorithms. From Fig. 3, all the components had been

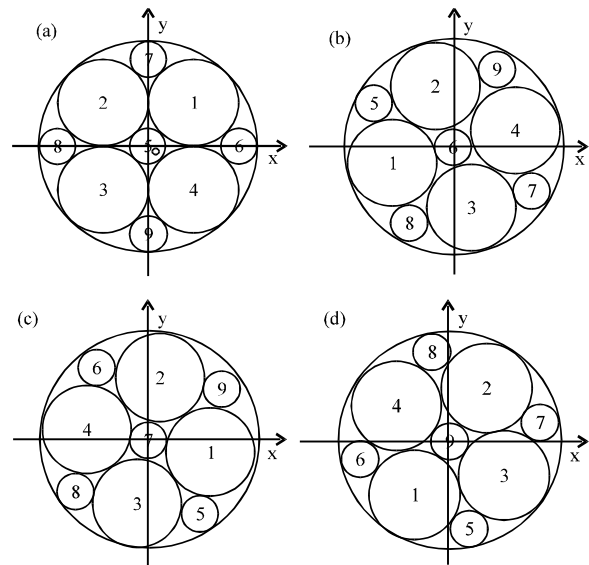


Fig. 3(a-d): Layout scheme of nine components in Experiment 1 of (a) Theoretical layout, (b) GA layout, (c) GWPSO layout and (d) LPPSO layout the number "1, 2,..., 9" stands for the serial number of the layout components

Table 1: Optimized result of experiment 1

Algorithm	R (mm)	T (g mm)	$\Delta S$ (mm)	t sec	Success rate (%)
Theoretic optimal	1.0000	0.0000	0.0000	-	-
GA	1.0844	0.0001	0.0015	40.9113	10
GWPSO	1.0452	0.0034	0.0023	14.9234	6
LPPSO	1.0068	0.0000	0.0001	14.2712	18

t: Calculating time, the success rate is percentage of the calculating successful solution and of all solutions, R: The calculating radius, T: The static equilibrium amount

Table 2: Optimized center coordinates of circle objects

No. of objects	GA (Mahfound, 1995)		GWPSO (Shi and Eberhart, 2001)		LPPSO (This paper)	
	$x_i$ (mm)	$y_i$ (mm)	$x_i$ (mm)	$y_i$ (mm)	$x_i$ (mm)	$y_i$ (mm)
1	-0.5803	-0.1547	0.5648	-0.1139	-0.3251	-0.4947
2	-0.1747	0.5747	0.1294	0.5853	0.3347	0.4853
3	0.1602	-0.5806	-0.1147	-0.6118	0.4932	-0.3248
4	0.5695	0.1563	-0.5782	0.0963	-0.4859	0.3254
5	-0.7484	0.4084	0.4758	-0.6964	0.1758	-0.8116
6	-0.0085	-0.0092	-0.4916	0.6789	-0.8169	-0.1684
7	0.7247	-0.4285	-0.0082	-0.0039	0.8195	0.1603
8	-0.4289	-0.7284	-0.6916	-0.4884	-0.1578	0.8136
9	0.4116	0.7356	0.6857	0.4691	0.0025	-0.0078

$x_i$ : Center's X coordinates of circle objects,  $y_i$ : The center's Y coordinates of circle objects

Table 3: Object's radius and optimized mass-center coordinates

Object no.	Radius (r mm <sup>-1</sup> )	Mass (m g <sup>-1</sup> )	GA		GWPSO		LPPSO	
			x <sub>i</sub> (mm <sup>-1</sup> )	y <sub>i</sub> (mm <sup>-1</sup> )	x <sub>i</sub> (mm <sup>-1</sup> )	y <sub>i</sub> (mm <sup>-1</sup> )	x <sub>i</sub> (mm <sup>-1</sup> )	y <sub>i</sub> (mm <sup>-1</sup> )
1	12	144	-20.83	16.98	-13.35	38.00	-0.13	16.93
2	3	9	-7.14	5.72	8.81	-2.59	25.82	28.86
3	12	144	34.92	-16.07	38.00	38.00	-23.44	22.47
4	3	9	-6.52	21.69	-14.45	7.98	13.99	2.26
5	9	81	13.80	19.52	-20.63	18.26	3.61	-3.75
6	10	100	20.06	37.48	-38.00	3.95	38.00	1.42
7	7	49	-7.33	-4.30	-0.99	-0.64	19.27	21.32
8	8	64	3.87	5.72	-15.82	-2.93	34.08	18.99
9	4	16	-3.24	15.47	-24.11	5.74	14.03	9.26
10	12	144	10.58	-17.65	0.37	18.31	17.17	-0.70
11	6	36	0.80	26.98	-38.00	38.00	24.03	9.22
12	10	100	-11.56	-20.88	38.00	16.00	-7.87	38.00
13	9	81	-23.48	-3.90	7.66	38.00	10.88	34.94
14	9	81	-11.43	35.65	-38.00	23.00	21.22	-7.49
15	10	100	31.39	1.57	18.59	5.98	36.51	37.14

The number 1, 2, ..., 9 are the serial number of the layout components, x<sub>i</sub>: The center's X coordinates of circle objects, y<sub>i</sub>: The center's Y coordinates of circle objects

placed in the big circle as the container, while the interferences in different algorithms should be calculated to determine which performed better.

From the data in Table 1, it was known that compared with GA algorithm and GWPSO algorithm, the operating results of LPPSO algorithm, respectively reduced the radius of envelope circle to 7.16-3.67% and the results of static balance of LPPSO amounted for zero (less than 0.0001 is considered as 0) the results of the interference of GWPSO were bigger and LPPSO was zero; Compared the success rate, LPPSO were increased higher than GA and GWPSO, respectively by 8-12%. In calculating the time-consuming, GA consumed the longest time, while the time of LPPSO consumed was a little smaller than GWPSO, which may be due to the LPPSO employed the isomorphism layout mode which can be faster than PSO evolution.

**Experiment 2:** The example was originated from integrated circuit layout design and was simplified as a weighted packing problem (Li *et al.*, 2003). This kind packing problem was additionally constrained by circles relationship that each circles in a rectangle should satisfy a certain adjacent relation. The problem can be stated as follows: 15 circles were placed into a two-dimension rectangular space and the adjacent relation weights among them should satisfy a weight matrix  $W_{ij}$  (Stand for intimate degree of objects in the position of layout space), the objects were request to meet the following requirements: (1) Objects not to interfere (2) The objects possessed larger weights  $W_{ij}$  two objects  $i, j$  should be mutually together, which minimized the numerical value of weighted distance  $C$ , as in Eq. 1. The rectangular area  $S$  of the object was the minimum. The mathematical model was Eq. 2.

The weighted distance  $C$  calculating formula was as follows:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} W_{ij} \quad (1)$$

Where,  $d_{ij}$  is the distance between two objects.

For:

$$X = (x_i, y_i)^T \quad i = 1, 2, \dots, n$$

and satisfy:

$$\begin{aligned} \min F(X) &= S + \lambda C \\ \text{s.t. int } A_i \cap A_j &= \Phi \quad i, j = 1, 2, \dots, n \quad i \neq j \end{aligned} \quad (2)$$

where,  $S$  standard for the area of the envelope of rectangular,  $\lambda$  was  $C$ 's weights that relative to  $S$ ,  $A_i$  and  $A_j$ , respectively stand for the area of objects  $i$  and  $j$ , the radius of objects showed in Table 3. In Table 3, the number "1, 2, ..., 9" stood for the serial number of the layout objects, the mass of each object was presented and  $x_i$  stood for the center's X coordinates of circle objects while  $y_i$  was the center's Y coordinates of circle objects which were optimized respectively by Genetic Algorithm (GA), global vision of PSO with inertia weight (GWPSO) and LPPSO (this study presented). Besides, the weight matrix can be seen in literature (Che *et al.*, 2010).

The weighting factor can be set  $\lambda = 0.8$ , the containers was rectangular, the layout container size was unlimited and the objects were 15 circles. This was a two-dimension layout problem.

The three algorithms such as Genetic Algorithm (GA), global vision of PSO with inertia weight (GWPSO) and LPPSO (this study presented), were, respectively applied that the article mentioned to solve this example and compare with each other. The initial parameters set of three algorithm was the same as experiment 1. Three algorithms were calculated 20 times and the maximum

Table 4: Optimized performance indexes of experiment 2

Algorithm	S (mm <sup>2</sup> )	C (mm)	$\Delta$ (mm <sup>2</sup> )
GA	6249.3	76891.0	0.1054
GWPSO	5971.5	79374.6	0.0000
LPPSO	5381.6	69325.8	0.0000

C: Numerical value of weighted distance, S: Rectangular area of the object,  $\Delta$ : Amount of interference

Table 5: Optimized average performance indexes of experiment 2

Algorithm	S (mm <sup>2</sup> )	C (mm)	$\Delta$ (mm <sup>2</sup> )	t sec	Successful rate (%)
GA	6154.1	81874.3	0.61	33.46	75
GWPSO	7477.6	87886.3	0.0	26.07	95
LPPSO	6091.3	80289.7	0.0	26.84	95

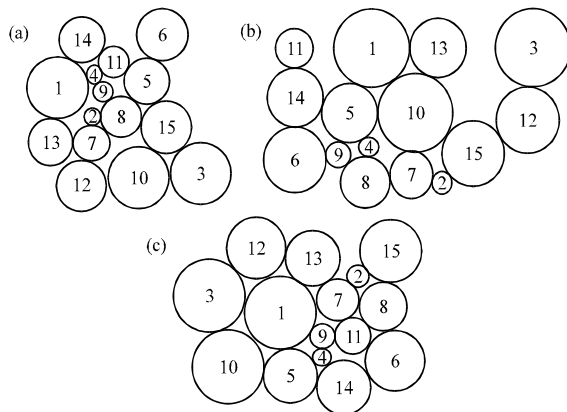


Fig. 4 (a-c): Layout scheme of experiment 2, (a) GA layout, (b) GWPSO layout, (c) LPPSO layout

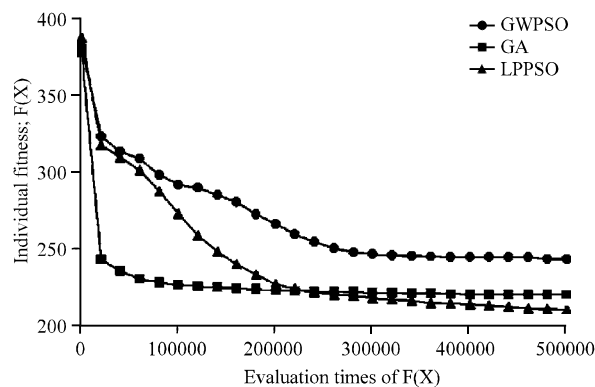


Fig. 5: Convergence graph of experiment 2

number of fitness value evaluation  $K_{max} = 500\ 000$ . The obtained best set of the layout result and data were shown in Table 4. The judgment condition of successful calculation must meet the formula (2) and the interference was less than 0.35. Fig. 4 was the result for the above three algorithms to calculate the optimal layout.

Table 4 showed that LPPSO which in this article was better than GWPSO and GA in solving the layout problem. The envelope rectangular area which was

derived from LPPSO optimal results decreased by 9.88-13.88%, respectively compared with GWPSO and GA and weighted distance C relative reduced by 12.66- 9.84%. From Table 5, the results of LPPSO in this article were that average envelope rectangular area minimum, slightly better than GA and GWPSO and GWPSO was the worst, which stood that the layout of the results of LPPSO in this article was a higher space utilization rate. Weighted distance of LPPSO was still slightly better than GA and GWPSO the worst. In the aspect of interference, the average amount interference of LPPSO and GWPSO are both 0, while the average amount interference of GA proceeds was 0.61. LPPSO was higher than GWPSO by 2.95-19.78% less than the GA in average computational time, there had little difference among algorithms, as they belonged to the same order of magnitude. In summary, it can be seen that the LPPSO obtained the best performance in solving the type of layout problems, followed by GA and GWPSO performed worst.

Figure 5 is the calculation convergence map of GA, GWPSO and LPPSO. Early in the calculation, GA performances faster convergence, but after the 20,000 times of functional evaluations, LPPSO lead in GA which may be due to the crossover and mutation-based search strategy of GA that is more conducive the potential solution of layout topology model early in the search, due to weak local search ability of GA late in the search, a lot of better potential solution did not ultimately optimize the feasible solution which satisfy the constraints. In the Fig. 5, LPPSO based on the speed of iterative formula of GWPSO and add non-isomorphic transformation. The Fig. 5 shows plus an unusual structure transform the convergence of the iterative steps (approximately after 1/5 of the number of iteration times  $K_{max}$ ), LPPSO convergence significantly faster than GWPSO. This shows that LPPSO good performance and convergence in solving the layout problem. The reason was that the LPPSO not only inherited the particle swarm algorithm has a better computational performance and robustness and LPPSO a combination of the layout pattern filling the unique knowledge of the layout problem and helped to improve the quality of the calculation and efficiency in the computation of the layout problem. When LPPSO random population initialization, it constructed a number of non-isomorphic layout mode solution to increase the diversity of the population; construct better solutions of non-isomorphic layout and algorithm optimization process solutions to substitute the poor solutions and this makes the algorithm easier to jump out of local optima.

## CONCLUSION

In this study, a so-called Layout Pattern-based Particle Swarm Optimization algorithm (LPPSO) is proposed for 2D packing problems. By analyzing results

of LPPSO in this study which solve the constrained 2D layout problems, the following conclusions can be obtained: (1) Constructing a reasonable initial layout mode will be useful to reduce combinatorial explosion of solution space and (2) Non-isomorphic layout pattern introduced into the iterative process will help the algorithm escape from local optima and explore search space near the optimum solution to accelerate the optimization process of the algorithm.

It should be noted that the layout pattern is constructed completely according to exact boundary line approach. Non-isomorphic layout individuals are added into the population, which is equal to add the potential good individuals that would guide the algorithm to move into a new region with better solution. How to construct non-isomorphic pattern in a simple and effective way is worthy of further study.

#### ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of People's Republic of China (Grants No. 51005034, 50975039) and the Fundamental Research Funds for the Central Universities (NCET-11-0055, DUT12LK33).

#### REFERENCES

- Che, C., H.F. Teng and Y.S. Wang, 2010. Constrained circles packing test problems: All the optimal solutions known. *Int. J. Comput. Math.*, 87: 2887-2902.
- Chen, G., 2011. Dynamic optimal power flow in FSWGs integrated power system. *Inform. Technol. J.*, 10: 385-393.
- Clerc, M., 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. *Proceeding of the Congress on Evolutionary Computation*, July 6-9, 1999, Washington, DC, USA., pp: 1951-1957.
- De Castro L.N. and F.J. Von Zuben, 2000. An evolutionary immune network for data clustering. *Proceeding of the 6th Brazilian Symposium on Neural Networks*, Nov. 22-25, Rio de Janeiro, RJ, Brazil, pp: 84-89.
- De Castro, L.N. and F.J. Von Zuben, 2001. Immune and neural network models: Theoretical and empirical comparisons. *Int. J. Comput. Intell. Appl.*, 1: 239-257.
- Dowsland, K.A. and W.B. Dowsland, 1992. Packing problems. *Europ. J. Operat. Res.*, 56: 2-14.
- Eberhart, R.C. and J. Kennedy, 1995. A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, October 4-6, 1995, Nagoya, Japan, pp: 39-43.
- Fan, X. and X. Fang, 2010. On optimal decision for QoS-aware composite service selection. *Inform. Technol. J.*, 9: 1207-1211.
- Fang, X., X. Gao, Z. Yin and Q. Zhao, 2011. An efficient process mining method based on discrete particle swarm optimization. *Inform. Technol. J.*, 10: 1240-1245.
- Feng, E., X. Wang, X. Wang and H. Teng, 1999. An algorithm of global optimization for solving layout problems. *Europ. J. Operat. Res.*, 114: 430-436.
- Gao, H., J. Cao and M. Diao, 2011. A simple quantum-inspired particle swarm optimization and its application. *Inform. Technol. J.*, 10: 2315-2321.
- Gao, X.Y., L.Q. Sun and D.S. Sun, 2009. An enhanced particle swarm optimization algorithm. *Inform. Technol. J.*, 8: 1263-1268.
- Glover, F., 1989. Tabu search: Part I. *ORSA J. Comput.*, 1: 190-206.
- Heng, X.C., Z. Qin, X.H. Wang and L.P. Shao, 2006. Research on learning Bayesian networks by particle swarm optimization. *Inform. Technol. J.*, 5: 540-545.
- Hongfei, T., Y. Zhicheng and G. Xuan, 1995. The engineering application of the manual of Chinese Go in the packing problems. *Proceeding of the 18th International Conference on Computer and Industrial Engineering*, October 25-27, 1995, Shanghai, China, pp: 1494-1499.
- Hongfei, T., L. Ziqiang, S. Yanjun and W. Yishou, 2006. A brief report on constructing isomorphic, non-isomorphic layout pattern. *Chinese J. Comput.*, 129: 985-991.
- Jing-Lei, G. and W. Zhi-Jian, 2011. Parameter identifications of elliptic differential equation by hybrid particle swarm optimization. *Inf. Technol. J.*, 10: 152-157.
- Kirkpatrick, S., C.D. Gelatt Jr. and M.P. Vecchi, 1983. Optimization by simulated annealing. *Science*, 220: 671-680.
- Li, G.Q. and H.F. Teng, 2003. Isomorphic and non-isomorphic layout patterns of packing problems. *Chinese J. Comput.*, 26: 1248-1254.
- Li, G.Q., H.F. Teng and J.Z. Huo, 2003. Parallel hybrid immune algorithm and its application to layout design. *Chin. J. Mech. Eng.*, 39: 79-85.
- Li, Z. and M.J. Dong, 2012. Case-based reasoning genetic algorithm for rectangle and circle packing problem with equilibrium constraints. *Adv. Intellig. Soft Comput.*, 122: 267-273.
- Liu, H., B. Feng and J. Wei, 2007. The method of the least reduction in oil reservoir based on rough set particle swarm. *Inform. Technol. J.*, 6: 865-871.
- Lodi, A., S. Martello and M. Monaci, 2002. Two-dimensional packing problems: A survey. *Europ. J. Operat. Res.*, 141: 241-252.



- Lu, H. and X. Chen, 2011. A new particle swarm optimization with a dynamic inertia weight for solving constrained optimization problems. *Inform. Technol. J.*, 10: 1536-1544.
- Mahfound, S.W., 1995. Niching methods for genetic algorithm. University of Illinois at Urbana-Champaign, Urbana, IL., USA.
- Ren, B. and W. Zhong, 2011. Multi-objective optimization using chaos based PSO. *Inform. Technol. J.*, 10: 1908-1916.
- Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceedings of the 1998 IEEE World Congress on Computational Intelligence and IEEE International Conference on Evolutionary Computation, May 4-9, 1998, Piscataway, New Jersey, pp: 69-73.
- Shi, Y. and R.C. Eberhart, 2001. Fuzzy adaptive particle swarm optimization. *Evol. Comput.*, 1: 101-106.
- Storn, R. and K. Price, 1997. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11: 341-359.
- Teng, H.F., S.L. Sun and W.H. Ge, 1994. Layout optimization for the objects installed on a rotating table the packing problem with equilibrium behavioral constraints. *Sci. China (Series A)*, 37: 1272-1280.
- Tu, J., Y. Zhan and F. Han, 2011. An improved PSO algorithm coupling with prior information for function approximation. *Inform. Technol. J.*, 10: 2226-2231.