

# Swarm-based intelligent optimization approach for layout problem

Fengqiang Zhao<sup>1,2</sup> · Guangqiang Li<sup>1,3</sup> · Rubo Zhang<sup>2</sup> ·  
Jialu Du<sup>1</sup> · Chen Guo<sup>1</sup> · Yiran Zhou<sup>1</sup> · Zhihan Lv<sup>4</sup>

Received: 24 September 2015 / Revised: 17 October 2015 / Accepted: 15 December 2015 /  
Published online: 23 December 2015  
© Springer Science+Business Media New York 2015

**Abstract** Layout problem is a kind of NP-Complete problem. It is concerned more and more in recent years and arises in a variety of application fields such as the layout design of spacecraft modules, plant equipment, platforms of marine drilling well, shipping, vehicle and robots. The algorithms based on swarm intelligence are considered powerful tools for solving this kind of problems. While usually swarm intelligence algorithms also have several disadvantages, including premature and slow convergence. Aiming at solving engineering complex layout problems satisfactorily, a new improved swarm-based intelligent optimization algorithm is presented on the basis of parallel genetic algorithms. In proposed approach, chaos initialization and multi-subpopulation evolution strategy based on improved adaptive crossover and mutation are adopted. The proposed interpolating rank-based selection with pressure is adaptive with evolution process. That is to say, it can avoid early premature as well as benefit speeding up convergence of later period effectively. And more importantly, proposed PSO update operators based on different versions PSO are introduced into presented algorithm. It can take full advantage of the outstanding convergence characteristic of particle swarm optimization (PSO) and improve the global performance of the proposed algorithm. An example originated from layout of printed circuit boards (PCB) and plant equipment shows the feasibility and effectiveness of presented algorithm.

---

✉ Fengqiang Zhao  
fqzhao2002@163.com

✉ Guangqiang Li  
lgq@dlnu.edu.cn

✉ Zhihan Lv  
zh.lv@siat.ac.cn

<sup>1</sup> School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

<sup>2</sup> College of Mechanical and Electronic Engineering, Dalian Nationalities University, Dalian 116600, China

<sup>3</sup> Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854, USA

<sup>4</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Science, Shenzhen, China

**Keywords** Genetic algorithms · Particle swarm optimization · Hybrid methods · Layout · Swarm intelligence

## 1 Introduction

Layout problems [4, 12] study how to put objects into limited space reasonably under given constraints (for example, no interference, increasing space utilization ratio). There are also lots of complex problems in various engineering fields, such as layout design of spacecraft modules and plant equipment. As for complex layout problems, some additional behavioral constraints need to be taken into account, such as the requirements for equilibrium, connectivity and adjacent states. These layout problems are of great importance. But unfortunately, due to their NP-Complete complexity, it is hard to solve them satisfactorily.

Relevant references [1, 4, 34] summarized the common approaches to solving layout problems, including mathematical programming and criterion methods, heuristic algorithms, graph theory, expert systems and algorithms based on swarm intelligence and natural laws. Mathematical programming and criterion methods possess relatively well-developed theoretical systems. Since these methods usually have the property of local convergence, it is rather difficult for them to solve large-scale problems. As a rule, we can obtain satisfactory solutions by heuristic algorithms. But every heuristic algorithm has its own scope of application and it is only effective for a restricted kind of problems. Some spatial relationships such as “adjacent” and “distance” are used in graph theory. These relationships can be adopted to cut out some search branches and relax “combination explosion” in search space. But several new questions (e.g., incomplete solution space) appear. Besides, the descriptive method of the solution space by graph theory is quite complicated. The limitation on solving complex layout problems by expert systems lies in that it is not easy to acquire expert knowledge and create inference engines. In accordance with the algorithm trend and solution quality, the robust universal algorithms based on swarm intelligence have significant advantages. They are particularly fit to solve medium or large-scale complex layout problems, compared with other traditional methods [35, 46]. Meanwhile, these algorithms also have several defects with regard to themselves, including premature and slow convergence rate. In this paper, some measures are taken and a novel swarm-based intelligent optimization algorithm is presented on the basis of PGA [9, 20]. We hope our work can benefit solving complex layout problems satisfactorily.

## 2 Presented Swarm-Based Intelligent Optimization Algorithm

### 2.1 Chaos Initialization

The purpose of adopting chaos initialization is to improve the quality of initial individuals. Chaos is a nonlinear phenomenon, which extensively exists in nature [33, 36]. Chaos systems possess the characteristics, such as randomness, ergodicity and sensibility to initial conditions [24]. By means of these characteristics, we can initialize population superiorly. The basic idea of chaos initialization can be stated as follows. First of all, generate the same number of chaos variables as many as decision variables. Then introduce chaos into decision variables and map

the ergodic range of chaos variables onto the definition ranges of decision variables. Here we select the following Logistic mapping as chaos generator.

$$Z_{k+1} = f(\mu, Z_k) = \mu Z_k(1-Z_k) \quad k = 0, 1, 2, \dots \quad (1)$$

where  $\mu$  is a control parameter and the system is in chaotic state when  $\mu=4$ .

The concrete procedure of chaos initialization is as follows. Assume that the number of decision variables is  $n$ . Firstly assign  $n$  original values  $Z_{i0}$  ( $i=1, 2, \dots, n$ ) to  $Z_k$  in formula (1), which are all between 0 and 1. So it can generate  $n$  different sequences of chaos variables, i.e.,  $\{Z_{ik}, i=1, 2, \dots, n\}$ . Then introduce every chaos variable into its corresponding decision variable by (2).

$$x_{ik} = a_i + (b_i - a_i)Z_{ik} \quad (2)$$

where  $b_i$  and  $a_i$  are the upper and lower bounds of decision variable  $x_i$  respectively.

For a given  $k$ , decision vector  $X_k = (x_{1k}, x_{2k}, \dots, x_{nk})^T$  represents a solution (an individual) to the problem. Along with increase in the value of  $k$ , we can obtain a series of initial individuals. Finally, we calculate the fitness of every obtained individual, select superior individuals to form initial population and divide it into initial subpopulations.

## 2.2 Interpolating rank-based selection with pressure

In traditional rank-based selection operator of genetic algorithms, a probability assignment table should be preset. But there is no deterministic rule for design of the table. And it is difficult for traditional rank-based model to make the selection probabilities of individuals adaptively changed along with evolution process. So some research works have been devoted to the improvement of traditional rank-based selection for these years [3, 37]. In this paper, based on the mathematical concept of interpolation method, we introduce interpolating rank-based selection with pressure and its relevant formulas. It can overcome the above-stated shortcomings of traditional rank-based selection operator.

### 2.2.1 Parameter decision

There are three control parameters in this kind of selection. They are selection pressure, distribution of interpolation points corresponding to individuals and probabilistic interpolating function.

Selection pressure  $\alpha$  denotes the ratio of the maximal individual selection probability  $P_{\max}$  to the minimal individual selection probability  $P_{\min}$  within a generation, i.e.,  $P_{\max} = \alpha P_{\min}$ . This parameter indicates the priority that the better individuals are reproduced into the next generation during selection process and it is changeable along with the evolution process of the algorithm. Because the fitness values of individuals within a population are usually not much different from one another in the final stage of genetic algorithms and traditional proportional selection model can't assign higher selection probability values to superior individuals, it usually takes a long time to converge to final results for genetic algorithms. However the proposed concept of changeable selection pressure  $\alpha$  can overcome this difficulty effectively. In the early stage, lesser selection pressure  $\alpha$  can keep population diversity and avoid early premature; while in the late stage, greater selection pressure  $\alpha$  helps to speed up algorithm convergence.

Let  $\alpha=f(K)$ ,  $K$  and  $f$  denote the generation number and an increasing function respectively.  $f$  can be multiple forms and we adopt linear increasing function for the sake of simplicity. Let  $\alpha_{\max}$  and  $\alpha_{\min}$  denote the maximum and minimum of selection pressure respectively, then

$$\alpha = \frac{(K-1)(\alpha_{\max}-\alpha_{\min})}{K_{\max}-1} + \alpha_{\min} \tag{3}$$

where  $K_{\max}$  is the maximal generation number set in a algorithm. And our numerical experiments show that  $\alpha_{\max}$  and  $\alpha_{\min}$  may be chosen in the interval [38, 46] and [1.5, 5] respectively [21].

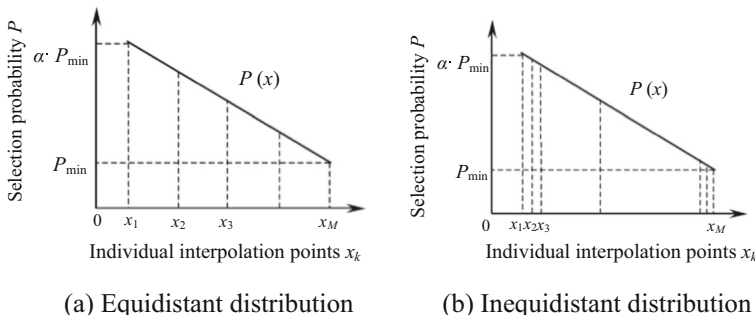
To calculate the selection probability of every individual, we should arrange all the individuals within a population in descending order based on their fitness values at first. And then determine every interpolation point corresponding to every individual. Interpolation points can be denoted by  $x_{k+1}=x_k+h_k$ ,  $k=1, 2, \dots, M-1$ , where  $M$  is the population size.  $x_k$  is the  $k$ th interpolation point corresponding to the  $k$ th individual within the descending order arrangement.  $h_k$  is the step size of interpolation. If  $h_k=c$  ( $k=1, 2, \dots, M-1$ ),  $c$  is a constant, then the distribution of individual interpolation points is equidistant. Otherwise, it is inequidistant. The concrete distribution types should be determined according to the requirement of actual computational condition. For example, under the circumstances of the same  $\alpha$  and  $P(x)$  (see the next paragraph), comparing the equidistant distribution shown in Fig. 1a to the inequidistant distribution with more compact ends shown in Fig. 1b, we know that the latter lays more emphasis on the function of the superior individuals with greater fitness values.

$P(x)$  is called probabilistic interpolating function and it is a decreasing function. The selection probability of the  $k$ th individual is  $P_k=P(x_k)$ . And there exist  $P_{\min}=P(x_M)$  and  $P_{\max}=P(x_1)$ .  $P_{\max}$  and  $P_{\min}$  are the maximal and minimal selection probability respectively.  $P(x)$  can be linear or nonlinear functions.

### 2.2.2 Realization process

Assume that the probabilistic interpolating function is linear and the distribution of individual interpolation points is equidistant. We present derived formulas of calculating selection probabilities of individuals in this case as follows. The relevant formulas in other cases can be derived similarly.

As it's shown in Fig. 1a, let  $\delta_k=P(x_k)-P(x_{k+1})$ ,  $k=1,2, \dots, M-1$ . And assume that the difference between  $P_{\max}$  and  $P_{\min}$  is  $\Delta=(\alpha-1)P_{\min}$ . Because  $P(x)$  is a linear function and  $h_k=x_{k+1}-x_k=c$  ( $k=1,$



**Fig. 1** Distribution types of individual interpolation points

2, ..., M-1),  $\delta_k(k=1,2,\dots, M-1)$  is a constant, denoted by  $\delta$ . And there exists  $\delta=\Delta / (M-1)=[(\alpha-1)P_{min}]/(M-1)$ . Therefore the selection probability of the  $k$ th individual is

$$P_k = \alpha P_{min} + [(1-\alpha)P_{min}(k-1)] / (M-1) \tag{4}$$

The sum of all the individual selection probability is 1, i.e.,

$$\sum_{k=1}^M \left[ \alpha \cdot P_{min} + \frac{(1-\alpha) \cdot P_{min} \cdot (k-1)}{M-1} \right] = 1 \tag{5}$$

Therefore we obtain

$$P_{min} = \frac{2}{M(\alpha + 1)} \tag{6}$$

Substituting above formula into formula (4), it is easy to find that

$$P_k = \frac{2\alpha \cdot (M-k) + 2(k-1)}{M \cdot (\alpha + 1) \cdot (M-1)} \quad k = 1, 2, \dots, M \tag{7}$$

Proposed selection operation can be described as follows. We first reproduce the best individual of current generation and put its copy into the next generation directly based on elitist model. Then calculate selection probabilities of all the individuals according to formula (7). Finally generate the other M-1 individuals of the next generation by fitness proportional model. The advantage of proposed selection operation is that it can conveniently change the selection probabilities of individuals by changing selection pressure during the evolution process. Therefore, the selection operation can be more adaptive to the algorithm run.

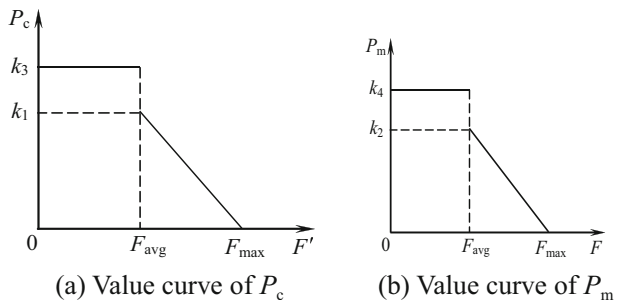
### 2.3 Improved adaptive crossover and mutation

To avoid early premature of genetic algorithms effectively and protect superior individuals from untimely destruction, the idea of adaptive crossover and mutation is proposed by Srinivas and Patnaik [38], see (8) and (9) and shown in Fig. 2. Here  $P_c$  and  $P_m$  denote crossover and mutation rate respectively.

$$P_c \begin{cases} k_1(F_{max}-F)/(F_{max}-F_{avg}), & F' \geq F_{avg} \\ k_3, & F' < F_{avg} \end{cases} \tag{8}$$

$$P_m \begin{cases} k_2(F_{max}-F)/(F_{max}-F_{avg}), & F' \geq F_{avg} \\ k_4, & F' < F_{avg} \end{cases} \tag{9}$$

**Fig. 2** Adaptive crossover & mutation operators by Srinivas and Patnaik [38]



In formula (8) and (9),  $F_{\max}$  and  $F_{\text{avg}}$  denote the maximal and average fitness of current population.  $F'$  denotes the greater fitness of the two individuals that participate in crossover operation.  $F$  denotes the fitness of the individual under mutation operation.  $k_1, k_2, k_3, k_4$  are constants. And there exist  $0 < k_1, k_2, k_3, k_4 \leq 1.0, k_1 < k_3, k_2 < k_4$ .

But according to these operators, crossover and mutation rate of the best individual among a population are both zero. It may lead to rather slow evolution in the early stage. To avoid its occurrence, it's better to let the individuals have due crossover and mutation rates, whose fitness values are equal or approximate to the maximal fitness. Therefore, improved adaptive crossover rate  $P_c$  and mutation rate  $P_m$  are presented as follows and shown in Fig. 3.

$$P_c \begin{cases} k_1 \exp \left[ \frac{(F_{\max} - F')}{F_{\max} - F_{\text{avg}}} (\ln k_3 - \ln k_1) \right], & F' \geq F_{\text{avg}} \\ k_3, & F' < F_{\text{avg}} \end{cases} \quad (10)$$

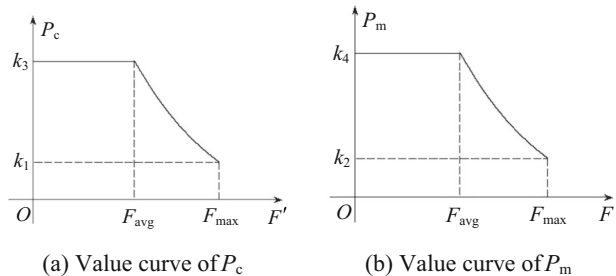
$$P_m \begin{cases} k_2 \exp \left[ \frac{(F_{\max} - F)}{F_{\max} - F_{\text{avg}}} (\ln k_4 - \ln k_2) \right], & F \geq F_{\text{avg}} \\ k_4, & F < F_{\text{avg}} \end{cases} \quad (11)$$

The basic idea of the improved adaptive operators can be described as follows. When the fitness value of an individual is less than the average fitness of the whole population, this individual is assigned greater crossover and mutation rates. It contributes to further exploration of solution space and prevention the algorithm from premature. While when the fitness value of an individual is greater than the average fitness of the whole population, the crossover and mutation rate of this individual decline exponentially with the increase of its fitness value. It can help the algorithm to enforce the exploitation ability and consolidate local search around superior individuals.

### 2.4 Multi-subpopulation evolution

It is well known that both exploration and exploitation are necessary for the optimization algorithms of swarm intelligence. Exploration denotes the ability to investigate the various unknown regions in the solution space; while exploitation refers to the ability to apply the knowledge of the previous good solutions to find better solutions. In order to achieve excellent performance, the two abilities of one swarm-based algorithm should be well balanced. Therefore, we classify subpopulations of proposed algorithm into two classes (named class A and B) according to their crossover and mutation rates ( $P_c$  and  $P_m$ ). Suppose that there is

**Fig. 3** Improved adaptive crossover & mutation operators



only one subpopulation within every class, named class A and B subpopulation respectively. Their parametric features are shown in Table 1.

In the light of their properties of initial fitness as well as crossover and mutation rates, we can see that it is easier for class A subpopulation to explore new parts of solution space and guard against premature. Class B subpopulation is mainly to consolidate local search around superior solutions. Obviously, class A subpopulation is for exploration and Class B subpopulation is for exploitation. After chaos initialization, presented algorithm arranges all the generated individuals according to their fitness values. The initial individuals with greater fitness are allocated to class B subpopulation; the initial individuals with smaller fitness are allocated to class A subpopulation.

The individual migration strategy between subpopulations of presented algorithm is as follows. At intervals of given migration cycle, the algorithm copies the best individuals in class A and saves them into class B subpopulation, then update class B subpopulation (eliminate the inferior individuals from it) and keep the same subpopulation size. Meanwhile, it selects some individuals from class B subpopulation and makes them migrate to class A subpopulation respectively. The migration individuals will replace inferior individuals in above subpopulations respectively as well. This migration strategy can accelerate convergence. In addition, we set control parameter  $K_m$ . When generation number  $K$  is multiples of  $K_m$ , the algorithm merges all the subpopulations together and arrange all individuals according to their fitness. Then it reallocates individuals to two subpopulations respectively according to their fitness values.

### 2.5 PSO update operators

Particle Swarm Optimization (PSO) was originally developed by Kennedy and Eberhart [19]. In PSO, each particle as an individual in genetic algorithms represents a potential solution. There are mainly two forms of PSO at present, i.e., global version and local version.

With regard to global version of PSO, in the  $n$ -dimensional search space,  $M$  particles are assumed to consist of a population. The position and velocity vector of the  $i$ th particle are denoted by  $X_i=(x_{i1}, x_{i2}, \dots, x_{in})^T$  and  $V_i=(v_{i1}, v_{i2}, \dots, v_{in})^T$  respectively. Then its velocity and position are updated according to the following formulas.

$$v_{id}^{k+1} = w \cdot v_{id}^k + c_1 \cdot rand() \cdot (p_{id}^k - x_{id}^k) + c_2 \cdot rand() \cdot (p_{gd}^k - x_{id}^k) \tag{12}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{13}$$

where  $i=1,2,\dots, M$ ;  $d=1,2,\dots, n$ ;  $k$  and  $k+1$  are iterative numbers.  $p_i=(p_{i1}, p_{i2}, \dots, p_{in})^T$  is the best previous position that  $i$ th particle searched so far and  $p_g=(p_{g1}, p_{g2}, \dots, p_{gn})^T$  is the best previous position for whole particle swarm.  $rand()$  denotes a uniform random number between

**Table 1** Parametric features of two classes of subpopulations

Subpopulation	Class A	Class B
Crossover rate	$k_1 = 0.6; k_3 = 1.0$	$k_1 = 0.1; k_3 = 0.5$
Mutation rate	$k_2 = 0.2; k_4 = 0.4$	$k_2 = 0.05; k_4 = 0.2$
Initial fitness	Smaller	Greater

0 and 1. Acceleration coefficients  $c_1$  and  $c_2$  are positive constants (usually  $c_1=c_2=2.0$ ).  $w$  is inertia weight and it showed that  $w$  decreases gradually along with iteration can enhance entire algorithm performance effectively [32].

It is usually set limitation to a particle velocity. Without loss of generality, assume that relevant following intervals are symmetrical. There exists  $v_{id}^k \in [-v_{d,\max}, +v_{d,\max}]$ .  $v_{d,\max}$  ( $d=1,2,\dots,n$ ) determine the resolution with which regions between present position and target position are searched. If  $v_{d,\max}$  is too high, particles may fly past good solutions. While, if it is too small, the algorithm may be stuck to local optima. Suppose that the range of definition for the  $d$ th dimension of a position vector is  $[-x_{d,\max}, +x_{d,\max}]$ , i.e.,  $x_{id}^k \in [-x_{d,\max}, +x_{d,\max}]$ . Usually let  $\pm v_{d,\max} = \pm kx_{d,\max}$ ,  $0.1 \leq k \leq 1$ .

In local version of PSO, particle  $i$  keeps track of not only the best previous position of itself, but also the best position  $p_{li} = (p_{li,1}, p_{li,2}, \dots, p_{li,n})^T$  attained by its local neighbor particles rather than that of the whole particle swarm. Typically, the circle-topology neighborhood model is adopted [11]. Its velocity update formula is

$$v_{id}^{k+1} = w \cdot v_{id}^k + c_1 \cdot \text{rand}() \cdot (p_{id}^k - x_{id}^k) + c_2 \cdot \text{rand}() \cdot (p_{li,d}^k - x_{id}^k) \quad (14)$$

And its position update formula is same as that of the global version of PSO. Compared with global version of PSO, local version of PSO has a relatively slower convergence rate but it is not easy to be stuck to local optima.

In addition to global and local version of PSO, we propose an additional new version, named random version of PSO. In random version, the neighborhood  $N_i$  of particle  $i$  is composed of  $s$  particles. Apart from particle  $i$  itself, the other  $s-1$  particles are randomly selected from the whole population. That is to say, particle  $i$  keeps track of the best previous position of itself and the best position attained within its random neighborhood  $N_i$ . In the broad sense, random version PSO can be regarded as a special kind of local version PSO. Merely its topology structure of neighborhood is dynamic and stochastic. Therefore it helps to explore solution space thoroughly and prevent from premature. The position update formula of random version is also same as that of the global version of PSO.

PSO has been applied to many fields and results are satisfactory [18]. It is easy to be implemented and has quite fast convergence rate among evolutionary algorithms. We noticed that both genetic algorithms and PSO are based on swarm intelligence and can match each other fairly well. To make full use of outstanding convergence characteristic of PSO and global search ability of genetic algorithms, we propose this hybrid algorithm. Specifically, let velocity and position update formulas together serve as a new operator (PSO update). After conventional genetic operation, individuals go on with PSO update operation. It hopes to make hybrid algorithm possess more superior global performance.

In presented algorithm, different versions of PSO update operator are introduced into different subpopulations. Global version PSO update operator is introduced into class B subpopulation in order to speed up the convergence of its individuals to global optima. While, random version PSO update operator is introduced into class A subpopulation. The reason lies in that it matches the function of exploring solution space of class A subpopulation and helps to prevent algorithm from premature.

We lay emphasis on two parameters in PSO update operator, i.e., inertia weight  $w$  and maximal velocity  $V_{\max}$ . Usually there exist  $w \in [0.3, 1.5]$ ,  $\pm v_{d,\max} = \pm kx_{d,\max}$  ( $0.1 \leq k \leq 1.0$ ). If they select greater values, the update operator is more likely to find out new parts of solution



**Table 2** Relevant settings of PSO update operators for two classes of subpopulations

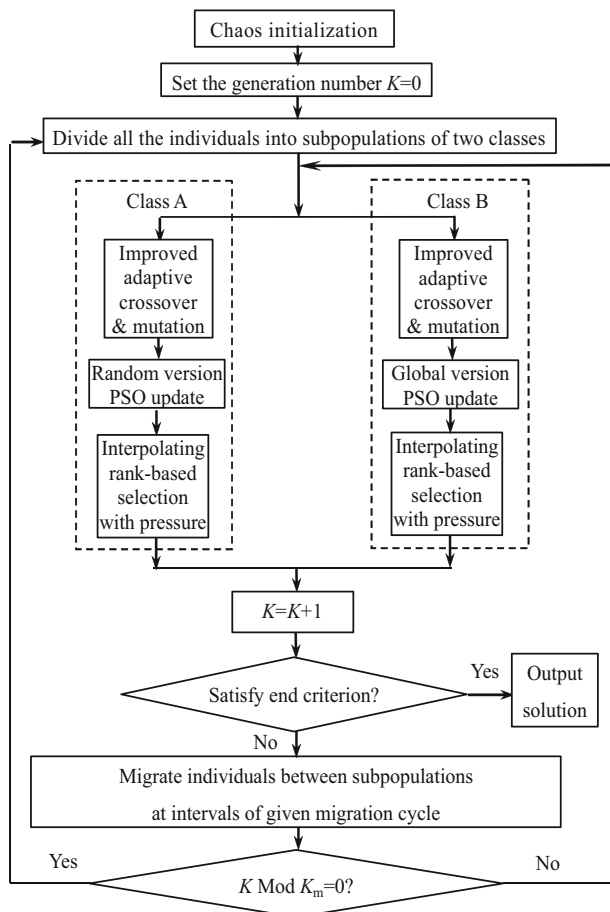
Subpopulation	Class A	Class B
Update operator	Random version	Global version
Inertia weight $w$	$w_{\max} = 1.5; w_{\min} = 0.9$	$w_{\max} = 0.9; w_{\min} = 0.3$
Coefficient $k$	$k_{\max} = 1.0; k_{\min} = 0.6$	$k_{\max} = 0.5; k_{\min} = 0.1$

space. Otherwise the update operator is good at local search. According to characteristics of different subpopulations, we set the range of  $w$  and  $V_{\max}$  of every mode of update operator in Table 2. Based on adaptation idea [32], we let  $w$  and  $k$  (coefficient of maximal velocity) decrease linearly along with evolution from their maximal values to the minimal values.

### 2.6 The procedure of presented algorithm

Flow chart of the presented swarm-based intelligent optimization algorithm is shown in Fig. 4.

**Fig. 4** Flow chart of the presented swarm-based intelligent optimization algorithm



**Table 3** The best layouts by two algorithms of the example

No.	The best layout by PGA		The best layout by presented algorithm	
	$x_i/mm$	$y_i/mm$	$x_i/mm$	$y_i/mm$
1	-24.80	-5.70	-8.03	6.85
2	22.85	-12.11	7.81	9.98
3	6.97	23.00	24.19	-20.64
4	-36.29	-15.28	-18.06	-4.36
5	-30.97	-26.00	-29.91	-2.44
6	13.69	2.08	-20.27	-19.87
7	6.76	-24.90	17.11	13.80
8	24.21	-23.85	31.24	18.62
9	-9.48	-1.18	-11.85	-8.68
10	-16.21	16.85	-28.92	18.57
11	14.12	-14.15	-11.04	24.54
12	35.70	-9.21	2.16	-19.88
13	31.22	9.35	30.54	1.49
14	-0.42	-10.45	4.48	23.72
15	-12.10	-25.00	11.81	-2.47

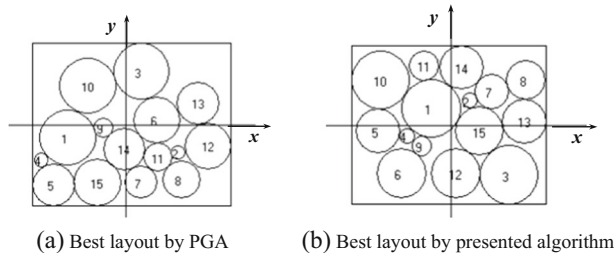
### 3 Numerical example

The engineering background of this example is the layout design of printed circuit boards (PCB) and plant equipment. Assume that there are  $n$  objects named  $A_1, A_2, \dots, A_n$  and the weight between  $A_i$  and  $A_j$  is  $w_{ij}, i, j=1, 2, \dots, n$ . Try to locate each object such that the value of expression  $S + \lambda_w C$  of a layout scheme is as small as possible and the constraints of no interference between any two objects are satisfied. Here  $S$  is the area of enveloping rectangle of a layout scheme.  $\lambda_w$  is a weight factor and  $C$  is the sum of the products of  $d_{ij}$  multiplied by  $w_{ij}$ , i.e.,

$$C = \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} w_{ij} \tag{15}$$

where  $d_{ij}$  is the distance between object  $A_i$  and  $A_j$ .  $w_{ij}$  may possess different meanings in different engineering problems. For example, in PCB layout design problems,  $w_{ij}$  denotes the connectivity between integrated devices. While in the layout design problems of plant equipment,  $w_{ij}$  denotes the adjacent requirement between equipment.

**Fig. 5** The obtained best layout patterns of the example by two algorithms



**Table 4** Comparison of obtained results of the best layouts by two algorithms of the example

Algorithms	$S/\text{mm}^2$	$C$	$t/s$
PGA	5996.46	89779.16	29.79
presented algorithm	5258.63	79082.28	27.53

Suppose that  $(x_i, y_i)$  is the coordinates of the center of the object  $A_i$ . The mathematical model for this problem is given by

$$\begin{aligned}
 &\text{Find } X = (x_i, y_i)^T, i \in \{1, 2, \dots, n\} \\
 &\min f(X) = S + \lambda_w C \\
 &\text{s.t. } \text{int}A_i \cap \text{int}A_j = \emptyset \quad i \neq j, i, j \in \{1, 2, \dots, n\}
 \end{aligned} \tag{16}$$

where  $\text{int}A_i$  presents the interior of object  $A_i$ .

Quoted from Ref. [22], 15 circular objects are contained in this example. Let  $\lambda_w=1$ . The radii of objects are  $r_1=r_3=r_{10}=12, r_2=r_4=3, r_5=r_{13}=r_{14}=9, r_6=r_{12}=r_{15}=10, r_7=7, r_8=8, r_9=4, r_{11}=6$  mm. The weight matrix is

$$W = \begin{pmatrix}
 0 & 0 & 0 & 98 & 98 & 0 & 81 & 0 & 92 & 93 & 45 & 61 & 99 & 84 & 27 \\
 0 & 0 & 34 & 0 & 0 & 0 & 93 & 44 & 0 & 0 & 33 & 60 & 0 & 0 & 56 \\
 0 & 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 85 & 0 & 65 & 39 & 0 & 50 \\
 98 & 0 & 0 & 0 & 91 & 50 & 5 & 24 & 73 & 0 & 4 & 0 & 0 & 31 & 23 \\
 98 & 0 & 0 & 91 & 0 & 37 & 0 & 16 & 78 & 95 & 0 & 0 & 73 & 32 & 0 \\
 0 & 0 & 0 & 50 & 37 & 0 & 0 & 35 & 0 & 31 & 0 & 0 & 0 & 48 & 0 \\
 81 & 93 & 0 & 5 & 0 & 0 & 0 & 94 & 33 & 34 & 26 & 61 & 0 & 87 & 87 \\
 0 & 44 & 0 & 24 & 16 & 35 & 94 & 0 & 91 & 0 & 0 & 0 & 59 & 39 & 0 \\
 92 & 0 & 0 & 73 & 78 & 0 & 33 & 91 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \\
 93 & 0 & 85 & 0 & 95 & 31 & 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 45 & 33 & 0 & 4 & 0 & 0 & 26 & 0 & 30 & 0 & 0 & 0 & 21 & 35 & 2 \\
 61 & 60 & 65 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 56 & 0 & 43 \\
 99 & 0 & 39 & 0 & 73 & 0 & 0 & 59 & 0 & 0 & 21 & 56 & 0 & 1 & 0 \\
 84 & 0 & 0 & 31 & 32 & 48 & 87 & 39 & 0 & 0 & 35 & 0 & 1 & 0 & 0 \\
 27 & 56 & 50 & 23 & 0 & 0 & 87 & 0 & 0 & 0 & 2 & 43 & 0 & 0 & 0
 \end{pmatrix} \tag{17}$$

To compare the performance of presented algorithm with that of traditional PGA objectively, we adopt presented algorithm and the PGA that possesses two subpopulations (same as presented algorithm) to solve this example respectively and the subpopulation sizes of both algorithms are identical. Moreover, any relevant contents of the two algorithms, such as encoding scheme, fitness function and migration cycle, that may be identical are selected as the same. All computation is performed on PC with CPU at 2.1GHz and RAM size of 2GB.

Both algorithms are calculated 20 times respectively. The best layouts among 20 optimal results by them are in Table 3 and the corresponding best geometric layout patterns are shown

**Table 5** Comparison of average values of optimal results by two algorithms of the example

Algorithms	$S/\text{mm}^2$	$C$	$K$
PGA	6153.83	95739.06	705
presented algorithm	5412.23	86962.57	512

in Fig. 5. The comparison of obtained results of the best layouts is given in Table 4. In Table 4,  $\Delta S$  and  $t$  denote the interference area and computation time respectively.

As the data presented in Table 4, for the best layout by PGA,  $S$ ,  $C$  and computation time  $t$  are  $5996.46 \text{ mm}^2$ ,  $89779.16$  and  $29.79 \text{ s}$ ; for the best layout by presented algorithm,  $S$ ,  $C$  and  $t$  are  $5258.63 \text{ mm}^2$ ,  $79082.28$  and  $27.53 \text{ s}$ . When obtained  $S \leq 5996.46 \text{ mm}^2$ ,  $C \leq 89779.16$  by presented algorithm, it takes  $22.91 \text{ s}$ . So in the sense of best results, to reach the same precision, presented algorithm reduces the cost of time by  $23.10 \%$  compared with PGA.

Table 5 lists relevant average values of obtained twenty optimal results of the example by two algorithms. In this table,  $K$  represents elapsed generation number for an optimal result.

Table 5 shows that compared with PGA, on an average, presented algorithm reduces the area of enveloping rectangle  $S$ , the parameter  $C$  and elapsed generation number  $K$  by  $12.05$ ,  $9.17$  and  $27.38 \%$ , i.e., from  $6153.83$  to  $5412.23 \text{ mm}^2$ , from  $95739.06$  to  $86962.57$  and from  $705$  to  $512$  respectively.

## 4 Conclusions

In order to solve complex layout problems more effectively, on the basis of PGA, we take several measures and propose a novel swarm-based intelligent optimization algorithm. These measures involve introducing chaos initialization, interpolating rank-based selection with pressure as well as multi-subpopulation evolution based on improved adaptive crossover and mutation into proposed algorithm. And more importantly, the idea of particle swarm optimization is introduced and PSO update operators can improve the global performance of the proposed algorithm. A numerical example shows that presented algorithm is feasible and effective for this kind of problems. It is really superior to PGA in accuracy and convergence rate. Our work is expected to provide inspiration and reference for solving layout problems satisfactorily. In addition, because presented algorithm is a universal algorithm, it also can be applied to solve other complex engineering optimization problems. The proposed optimization approach is able to be applied in some related research fields, such as network [13–17, 31], image processing [2, 7, 26–28, 39, 47], computer graphics [30], grid [5, 6, 23], cloud computing [40–43], multimedia [10, 25, 29, 45], optimization algorithms [8, 44, 48, 49].

**Acknowledgments** Our research work is financially supported by the National Natural Science Foundation of China (No. 61374114 and No. 51579024), the Fundamental Research Funds for the Central Universities of China (No. 3132014321, No. DC120101014, No. DC110320), the Applied Basic Research Program of Ministry of Transport of China (No. 2011-329-225-390, No. 2012-329-225-070), the China Scholarship council (No. 201306575010), the Higher Education Research Fund of Education Department of Liaoning Province of China (No. LT2010013), and the Doctor Startup Foundation of Liaoning Province (No. 20131006).

## References

1. Albert EFM, Manuel I, Silvano M, Marcos J, Negreiros G (2013) Optimal design of fair layouts. *Flex Serv Manuf J* 25(3):443–461
2. Boneh D, Lynn B, Shacham H (2004) Short signatures from the Weil pairing. *J Cryptol* 17(4):297–319

3. Boudissa E, Bounekhla M (2012) Genetic algorithm with dynamic selection based on quadratic ranking applied to induction machine parameters estimation. *Electr Power Compon Syst* 40(10):1089–1104
4. Cagan J, Shimada K, Yin S (2002) A survey of computational approaches to three-dimensional layout problems. *CAD Comput Aided Des* 34(8):597–611
5. Che L, Shahidehpour M, Alabdulwahab A, Al-Turki Y (2015) Hierarchical coordination of a community microgrid with AC and DC microgrids. *IEEE Trans Smart Grid*
6. Che L, Zhang X, Shahidehpour M, Alabdulwahab A, Abusorrah A (2015) Optimal interconnection planning of community microgrids with renewable energy sources. *IEEE Trans Smart Grid*
7. Chen Z, Huang W, Lv Z (2016) Towards a face recognition method based on uncorrelated discriminant sparse preserving projection. *Multimed Tools Appl*
8. Dang S, Kakimzhanov R, Zhang M, et al (2014) Smart grid-oriented graphical user interface design and data processing algorithm proposal based on LabVIEW. *Environment and Electrical Engineering (EEEIC), 2014 14th International Conference on*. IEEE 323–327
9. De La Calle FJ, Bulnes FG, García DF, Usamentiaga R, Molleda JA (2015) Parallel genetic algorithm for configuring defect detection methods. *IEEE Lat Am Trans* 13(5):1462–1468
10. Gu W, Lv Z, Hao M (2016) Change detection method for remote sensing images based on an improved Markov random field. *Multimed Tools Appl*
11. Jame K, Rui M (2002) Population structure and particle swarm performance. *Proceedings of the 2002 Congress on Evolutionary Computation*. Honolulu, HI, USA 2:1671–1676
12. Jankovits I, Luo C, Anjos MF, Vannelli A (2011) A convex optimization framework for the unequal-areas facility layout problem. *Eur J Oper Res* 214(2):199–215
13. Jiang D, Hu G (2009) GARCH model-based large-scale IP traffic matrix estimation. *IEEE Commun Lett* 13(1):52–54
14. Jiang D, Xu Z, Chen Z et al (2011) Joint time–frequency sparse estimation of large-scale network traffic. *Comput Netw* 55(15):3533–3547
15. Jiang D, Xu Z, Li W, Yao C, Lv Z, Li T (2015) An energy-efficient multicast algorithm with maximum network throughput in multi-hop wireless networks. *J Commun Netw*
16. Jiang D, Xu Z, Xu H et al (2011) An approximation method of origin–destination flow traffic from link load counts. *Comput Electr Eng* 37(6):1106–1121
17. Jiang D, Ying X, Han Y, et al (2015) Collaborative multi-hop routing in cognitive wireless networks. *Wirel Pers Commun* 1–23
18. Kameyama K (2009) Particle swarm optimization: a survey. *IEICE Trans Inf Syst* 92(7):1354–1361
19. Kennedy J, Eberhart R (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia 1942–1948*
20. Knysch DS, Kureichik VM (2010) Parallel genetic algorithms: a survey and problem state of the art. *Int J Comput Syst Sci* 49(4):579–589
21. Li GQ (2003) Research on theory and methods of layout design and their applications, Ph.D. dissertation. Dalian University of technology, Dalian, China
22. Li GQ (2005) Evolutionary algorithms and their application to engineering layout design, Postdoctoral Research Report, Tongji University, Shanghai, China
23. Li X, Lv Z, Hu J, et al (2015) Traffic management and forecasting system based on 3D GIS. *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. IEEE
24. Li C, Zhou J, Kou P, Xiao J (2012) A novel chaotic particle swarm optimization based fuzzy clustering algorithm. *Neurocomputing* 83:98–109
25. Lin Y, Yang J, Lv Z et al (2015) A self-assessment stereo capture model applicable to the internet of things. *Sensors* 15(8):20925–20944
26. S Liu, W Fu, L He, et al (2015) Distribution of primary additional errors in fractal encoding method [J]. *Multimed Tools Appl*
27. S Liu, Z Zhang, L Qi, et al (2015) A fractal image encoding method based on statistical loss used in agricultural image compression [J]. *Multimed Tools Appl*
28. Lv Z, Halawani A, Fen S, et al (2015) Touch-less interactive augmented reality game on vision based wearable device. *Pers Ubiquit Comput*
29. Lv Z, Halawani A, Feng S et al (2014) Multimodal hand and foot gesture interaction for handheld devices. *ACM Trans Multimed Comput Commun Appl (TOMM)* 11(1s):10
30. Lv Z, Tek A, Da Silva F et al (2013) Game on, science-how video game technology may help biologists tackle visualization challenges. *PLoS One* 8(3):57990
31. Lv Z, Yin T, Han Y, Chen Y et al (2011) WebVR—web virtual reality engine based on P2P network. *J Netw* 6(7):990–998
32. Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl Soft Comput J* 11(4):3658–3670

33. Pluhacek M, Senkerik R, Zelinka I (2014) Chaos driven particle swarm optimization with basic particle performance evaluation—an initial study. *Lect Notes Comput Sci* 8838:445–454
34. Qian ZQ, Teng HF (2002) Algorithms of complex layout design problems. *China Mech Eng* 13(8):696–699
35. Rocca P, Mailloux RJ, Toso G (2015) GA-based optimization of irregular subarray layouts for wideband phased arrays design. *IEEE Antennas Wirel Propag Lett* 14:131–134
36. Silva CP (1996) Survey of chaos and its applications. *Proceedings of the 1996 I.E. MTT-S International Microwave Symposium Digest, San Francisco, CA* 1871–1874
37. Sokolov A, Whitley D, Salles Barreto ADM (2007) A note on the variance of rank-based selection strategies for genetic algorithms and genetic programming. *Genet Program Evolvable Mach* 8(3):221–237
38. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans Syst Man Cybern* 24(4):656–667
39. Su T, Wang W, Lv Z et al (2016) Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve. *Comput Graph* 54:65–74
40. Wang Y, Su Y, Agrawal G (2015) A novel approach for approximate aggregations over arrays. *Proceedings of the 27th International Conference on Scientific and Statistical Database Management. ACM* 4
41. Wang K, et al (2015) Load-balanced and locality-aware scheduling for data-intensive workloads at extreme scales. *Concurrency Comput Pract Exp*
42. Wang K, et al (2015) Overcoming Hadoop scaling limitations through distributed task execution. *Proc IEEE Int Conf Clust Comput*
43. Xu C, He X, Abraha-Weldemariam D (2012) Cryptanalysis of Wang’s auditing protocol for data storage security in cloud computing. In *Proc. ICICA’12, Springer-Verlag* 422–28
44. Yang J, Chen B, Zhou J et al (2015) A low-power and portable biomedical device for respiratory monitoring with a stable power source. *Sensors* 15(8):19618–19632
45. Yang J, He S, Lin Y, Lv Z (2016) Multimedia cloud transmission and storage system based on internet of things. *Multimed Tools Appl*
46. Yang J, Yang J (2011) Intelligence optimization algorithms: a survey. *Int J Adv Comput Technol* 3(4):144–152
47. Zhang S, Jing H (2014) Fast log-gabor-based nonlocal means image denoising methods. *IEEE Int Conf Image Proc (ICIP) 2014:2724–2728*
48. Zhang X, Xu Z, Henriquez C, et al (2013) Spike-based indirect training of a spiking neural network-controlled virtual insect. *2013 I.E. 52nd Annual Conference on Decision and Control (CDC). IEEE* 6798–6805
49. Zhang S, Zhang X, Ou X (2014) After we knew it: empirical study and modeling of cost-effectiveness of exploiting prevalent known vulnerabilities across iaas cloud. *Proceedings of the 9th ACM symposium on Information, computer and communications security. ACM* 317–328



**Fengqiang Zhao** received his Ph.D. degree in Mechanical and Electronic Engineering from Dalian University of Technology, Dalian, China, in 2006. He is a Lecturer in the College of Electromechanical & Information Engineering, Dalian Nationalities University, China. Currently he is also a Postdoctoral Fellow in the College of Information Science and Technology, Dalian Maritime University, China. His research interests include intelligent systems and optimization, fault diagnosis and noise control.



**Guangqiang Li** is an Associate Professor in the College of Information Science and Technology, Dalian Maritime University, China. He received his Ph.D. degree from Dalian University of Technology in 2003. During 2004 and 2005, he was a Postdoctoral Fellow in Tongji University, Shanghai, China. His current research focuses on intelligent systems and optimization, evolutionary algorithms, ship motion control and complex layout design.



**Rubo Zhang** is Professor of College of Electromechanical and Information Engineering at Dalian Nationalities University. He received his master degrees in Navigation, Guidance and Control from Harbin Shipbuilding Institute in 1987 and received his doctor degrees in Control Theory and Control Engineering from Harbin Engineering University in 1999. His research interests include intelligent control, intelligent robotics and machine learning.



**Jialu Du** is a Professor in the College of Information Science and Technology, Dalian Maritime University, China. She received her Ph.D. degree from Dalian Maritime University in 2005. Her research interests include nonlinear control, intelligent control and ship motion control.



**Chen Guo** is a Professor in the College of Information Science and Technology, Dalian Maritime University, China. He received Ph.D. degree from Dalian Maritime University in 1991. His research interests include ship automation and intelligent control.





**Yiran Zhou** is an Associate Professor in the College of Information Science and Technology, Dalian Maritime University, China. He received his Ph.D. degree from Zhejiang University in 2008. His current research focuses on intelligent systems and optimization and measurement and control technology.



**Zhihan Lv** is a native Chinese. He is an engineer and researcher of virtual/augmented reality and multimedia major in mathematics and computer science, having plenty of work experience on virtual reality and augmented reality projects, engage in application of computer visualization and computer vision. His research application fields widely range from everyday life to traditional research fields (i.e., geography, biology, medicine). During the past years, he has completed several projects successfully on PC, Website, Smartphone and Smartglasses.