# An Adaptive Genetic Algorithm for 2D Packing Problem

Baochun Wang

School of Mechanical Engineering, Tianjin University of Technology and Education

Tianjin 300222, China

E-mail: wangbaochun4221@163.com

# Abstract

Rectangle packing problems solved belong to NP-Hard Problems. It is complex combination optimization in nature. This, an adaptive hybrid algorithm is proposed, in order to deal with optimization Algorithm limitations. In the end, the high efficiency of the layout optimization strategy and reasoned conclusions is verified by simulation results.

Keywords: Rectangle-packing Problem, Simulated Annealing Algorithm, Genetic Algorithm, Combination optimization

# 1. Introduction

Rectangle packing problems means there are some small rectangles and a layout space that the size of the known. We must put these small rectangular into the layout of space, which need to satisfy the necessary constraints, and reached a certain optimal standards. As combinatorial optimization problems that have NP-hard, the packing problem exists widely in mechanical design and manufacture, transportation, large scale circuit design and Aerospace. It is crucial for the development of enterprises that a good layout design can improve product's performance and reduce the product's manufacturing and transportation costs. Because packing problems are NP-hard problem, belong to a complex combination optimization. In other words, algorithm for solving this problem needs a very long running time and a great storage space when the scale of the problem is larger. This is the so-called combinatorial explosion. But, a satisfactory solution of the problem can be solved in the existing computer, after a number of research scholars have continuous improvement on algorithm.

For solving combinatorial optimization problem, simulated annealing algorithm and genetic algorithm is an effective methods. However, they have their own advantages and disadvantages. For instance, simulated annealing algorithm (KANG,Li-shan,1995) needs a high enough temperature and premature convergence of genetic algorithms, which decided to separate the algorithm, can not solve effectively the rectangle layout problem. This article focuses on the work of Crossover probability, mutation rate and Boltzmann acceptance probability will be obtained by an adaptive method, in order to achieve the optimization of rectangular layout problem positioning function (Wang,Jinmin,2005) of the parameters, and improve the layout results. In consequence, this article focuses on optimization of positioning rules.

# 2. Problem description

We will adopt approach of "Dynamic attractive factors" (P.C.Gilmore, 1961) to determine the layout rectangular location in the layout space.

If there are n small rectangular that is waiting into layout space. Our aim is put small rectangular blocks into the layout space as much as possible. Requirement is that the utilization rate of the area of the layout space is as high as possible, and requires any two of the rectangular block into a container can not overlap each other.

Rectangular block i's positioning function as follow:

$$\min\{ g(x_i, y_i) = \sum_{t=1}^{p} \omega_t g_t(x_i, y_i) \} <2.1>$$

In here:  $g_t(x_i, y_i) = \alpha_t | x_i - x_{0t} | + \beta_t | y_i - y_{0t} | (t = 1 \dots p, i = 1 \dots n)$ 

 $g_t(x_i, y_i)$  is the positioning function of dynamic attractive factor t, and p is the number of attractor.  $(x_i, y_i)$  is the coordinates of point of rectangular block i.

 $(x_{0t}, y_{0t})$  is the coordinates of the layout of attractor.  $\alpha_t$ ,  $\beta_t$ —Weighting factor,  $\alpha_t + \beta_t = 1$ .  $\omega_t$ —Weighting

factor, 
$$\sum_{t=1}^{m} \omega_t = 1$$

Different parameters of the positioning function determine different locations of rectangular blocks, which will result in different results of layout. Therefore, the problem is transformed into the optimization problem of multiple parameters. In other words, this article focuses on the work of optimization parameters and to find the optimal combination of parameters about different combinations of rectangular.

### 3. An Adaptive Hybrid Algorithms

Usually, in Genetic Algorithm (Min, Lin,2008), the mutation probability and crossover rate is fixed as a constant in the entire optimization process, which can bring out the local optimum problem and prolong the searching time. In the simulated annealing algorithm, Boltzmann acceptance probability is also a constant, therefore, the algorithm efficiency is limited. Through analysis of the problem, in the paper, we will use the adaptive crossover and mutation probability, which will help improve the convergence speed of genetic algorithms.

### 3.1 Basic Operation

### (1) Selection operator

First, using elitist strategy in this article, that is, make the individual of the highest fitness directly into the next generation without any operation. Next, we are using the ratio of roulette selection to choose. The advantage of doing so is that the best individual in next generation must be better than the previous generation of the entire individual.

### (2) Crossover operator

In the early stages of the algorithm, we make individual random pairing.

We note that through cross-action will not be able to generate new individuals when the individual of current population is very similar. Then carrying out the operations will hamper the maintenance of population diversity, crossover Operator also become meaningless. Therefore, we will accord to "difference" to select the individuals to perform matching operation in the algorithm.

**Definition:** For the two individual  $G_K$ ,  $G_L$ ,  $X_{i(k)}$  is the *ith* component of the individual  $G_K$ ,  $X_{i(L)}$  is the *ith* component of the individual  $G_L$ . Then by the formula:

$$H = \sum_{i=1}^{n} \mid \mathbf{x}_{i(k)} - \mathbf{x}_{i(L)} \mid$$

We can Obtain the variable H by calculating, which is the "difference" between the individual  $G_K$  and the individual  $G_L$ , and n is the number of individual components.

When paired, calculate the "difference" between the each individual. We make the biggest "difference" between the two individuals to pairing operation.

 $X_{i(k)}$  and  $X_{i(k+1)}$  are *ith* components of the individuals of implementing crossover,  $X'_{i(k)}$  and  $X'_{i(k+1)}$  are new components, then

$$\begin{aligned} X'_{i(k)} &= L \times X_{i(k)} + (1 - L) \times X_{i(k+1)} \\ X'_{i(k+1)} &= L \times X_{i(k+1)} + (1 - L) \times X_{i(k)} \\ L &= 0.5 + \alpha \end{aligned}$$

Here:  $\alpha$  is a random number between 0 and 1.

(3) Adaptive mutation probability

In this essay, the formula of solving the mutation probability  $P_m$  as following

$$P_{m} = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f - f_{avg})}{f_{max} - f_{avg}}, & f \ge f_{avg} \\ P_{m1}, & f < f_{avg} \end{cases}$$

Where,  $P_{m1} = 0.1$ ,  $P_{m2} = 0.001$ ,  $f_{max}$  is the biggest fitness in population,  $f_{avg}$  is the average fitness, f is the fitness of individual whose do mutation.

3.2 Adaptive Hybrid Algorithms flow chart

First of all, set the maxgen a fixed value. The process of Adaptive Hybrid Algorithms based on the Genetic Algorithm and Simulated Annealing algorithm is given by:

STEP1: Randomly generated initial population, initialize T=0;

STEP2: Calculate the fitness of all individuals in population;

STEP3: if T>maxgen, return the optimum result, end of program; otherwise continue;

STEP4: selection, crossover and mutation;

STEP 5: compute  $\Delta f_i$ , which is the increment of the ith individual fitness in the population.

If  $\Delta f_i > 0$ ,  $X_i = X_{new}$ , else, calculate the probability of receiving poor-quality solution q, set C=rand (rand is a random number between 0 and 1), if C>q,  $X_i = X_{new}$ . Else,  $X_i = X_i$ ;

STEP6: T=T+1; return STEP2;

### 4. Application examples

We implement the Adaptive Hybrid Algorithms procedure by  $c^{++}$  programming language. The 6 rectangle-packing instances proposed by Hopper et al.(Hopper.E, 2001) are used to test our algorithm. For each instance, the width of the layout space is known, the optimal height for each instance is known, and we use the optimal height as the height of the layout space. Our experiments were run on a computer with Celeron(R) CPU 2.93GHz, 512MB memory. The results of the algorithm are shown as table 1.

From table1 we can see that layout filling rate is above 97%. Hence, we can get good layout of the results using this algorithm.

In addition, we are using ref(Wang, P Y, 1983) data to compare our algorithm with the algorithms in ref(Wang, Aihu, 1996 and Liu, Tianliang). Table 2 shows the results of the comparison. The size of the layout space is  $2.5 \times 2$ . From table 2 we can see that this algorithm's result is better than others.

# 5. Conclusions

As to Positioning Rules of two-dimensional rectangular packing problem, the context of Dynamic attractive factors, make genetic algorithm and simulated annealing algorithm to combine, an adaptive hybrid algorithms is proposed in the paper. In the end, we prove the efficiency of the algorithm using some examples. The algorithm also need further study, in order to get a better layout of the results.

# Acknowledgements

This paper is part of the project supported by National Natural Science Foundation of China (Grant 60975046). The authors wish to express their appreciation to the supporter.

# References

Hopper E and Turton B. (2001). An empirical investigation of meta-heuristic and heuristic algorithm for a 2D packing problem. *European Journal of Operational Research* 2001; 128(1):34-57.

KANG, Li-shan and Xie, Yun. (1995). Non-numerical parallel computation-Simulated Annealing Algorithm. Beijing: Science Press, 1995.

Liu, Tianliang, Yuan, li and Hou, Yunzhang. (2003). A heuristic algorithm for solving rectangle packing problem. *Journal of Qingdao University*, 2003, 4(16):88-92(in Chinese).

Min,Lin and Li,Bai. (2008). An Adaptive Annealing Genetic Algorithm for Job-shop Scheduling. *Industrial electronics and application* .2008, 18-23.

P.C.Gilmore and R.E.Gomory. (1961). A linear programming approach to the cutting stock problem. *Operational Research*, 1961.9:848-859.

Wang, Aihu, E, Mingcheng and Cha, Jianzhong. (1996). A heuristic for two-dimensional packing problem based on space decomposition. *Journal of Tianjin University*, 1996, 29(6): 840-846(in Chinese).

Wang, P Y. (1983). Two algorithms for constrained two-dimensional cutting stock Problem. *Operationa Research*, 1983, 31: 573-586.

Wang, Jinmin and Yang, Weijia. (2005). Dynamic attractive factors applied in packing problems. *Journal of Computer-Aided Design & Computer Graphics*. 2005.17(8):1725-1730.

Instance	Number of rectangles	Layout space	Filling rate
1	16	20×20	100%
2	25	40×15	100%
3	28	60×30	97.20%
4	49	60×60	99.42%
5	73	60×90	98.74%
6	97	80×120	99.04%
7	197	160×240	99.23%

Table 2. Results of the comparison

Name	SA	ref(Wang, Aihu)	ref(Liu, Tianliang)	GA	this algorithm
Filling rate	80.75%	91.512%	94.804%	95.472%	97.77%