A SELF-BOUNDARY FALL FREE GENETIC ALGORITHM FOR 2D OPEN DIMENSION RECTANGLE PACKING PROBLEM OF SATELLITE LAYOUT DESIGN

Junjie Yang, Xiaoqian Chen, Ning Wang, Jie Qi

College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China

ABSTRACT

Layout design of spacecraft module belongs to scheme design problem, which has been proved to be NP hard. In practical engineering, the dimension of satellite configuration is usually unknown and needs to be optimized (generally minimized) as well, while the dimensions of satellite components are known. And the layout design of satellite module can be simplified to a 2D open dimension rectangle packing problem. This paper proposed a self-boundary fall free genetic algorithm (GA&SBFFA) for the open dimension rectangle packing problem, which is used to optimally arrange the rectangles densely and minimize the area of enveloping rectangle. Meanwhile, the shape of the enveloping rectangle is maintained as square as possible so as to satisfy the static equilibrium requirement in some complex system design, e.g. satellite. Two experiments are used to testify the proposed method and the efficacy is demonstrated.

***Index Terms*—** Packing, layout optimization, area minimization

1. INTRODUCTION

The 2D open dimension packing problem (ODP) is an NP-hard layout optimization problem with a high computational complexity，which is a core issue of VLSI design and layout design of satellite components. And one of the reduced forms of the 2D ODP is rectangle packing area minimizationproblem (RPAMP). It aims to place all the axis-aligned rectangular items of known sizes onto a plane without overlapping and meanwhile minimize the area of the enveloping rectangle. Because of the high computational complexity of RPAMP, the exact algorithm which can determine the optimal layout for solving RPAMP can handle 30 modules at most [1]. Therefore, the design of efficient algorithm to solve RPAMP becomes the main trend of the present research. One of the typical approaches is reduction method which transforms an instance of the RPAMP to a series of instances of the strip packing problem (SPP) or the rectangle packing problem (RPP) [2], and the method proposed in this paper is reduction method as well.

The main idea of the reduction method is to construct aset of candidate widths or heights of the enveloping rectangle to transform an RPAMP instance into a series of instances of the RPP or the SPP and then to design algorithms for the RPP and SPP. Because various reduction methods adopt similar approaches to constructing candidate dimensions of the enveloping rectangle, the design of the reduction method focuses on seeking efficient RPP and SPP algorithm. By combing an improved least flexibility first principle and greedy search, Wu and Chan [3] introduces an optimization algorithm for the RPP. Based on the conceptions of corner action and smooth degree, He, Huang and Jin [4] proposed a best fit algorithm (BFA) for the RPP. To solve the SPP, article [5] suggested a two-stage intelligent search algorithm that first constructed a solution greedily, and then improved the solution by a local search and a SA algorithm.In Table 1 an extensive ample of papers for solving the RPAMP with hard modules is listed [6]-[15].These methods mentioned above are time-consuming and the constraint of the methods is onefold.

Considering the computational expense and the filling rate (the ratio of the sum area of the items and the area of the enveloping rectangle) synthetically, a Self-Boundary Fall Free Algorithm (SBFFA) is proposed in this paper. Taking some practical application demands into account, e.g. the rectangular items should be axis-aligned and the layout shape should be as square as possible so that three-axis stabilization can be more easily realized in satellite design in this research it is assumed that the items cannot rotate in the coordinate system and the aspect ratio of enveloping rectangle should be less than 2 (the aspect ratio is set as 1.2 in this paper).Based on these assumption, this paper generates the random packing sequence number of the items by GA, constructs the packing layout dynamically by SBFFA and selects the best result whose area of the enveloping rectangles the minimum. From the results of two experiments, the approach proposed in this paper can satisfy the requirement of the layout shape while optimize the layout area and reduce the computational expense.

Table 1Solution methods for the RPAMP and FPP with fixed modules (sample).

Authors/source	Objectives	Additional constraints	Approach type	Additional characteristics of approach
Murata et al. (1996)	Area,wirelength		SA	Sequence pair (SP) represent
Guo et al. (1999)	Area, wire length		C/GH	O-tree represent, deterministic
Tang et al.(2001)	Area, wire length	Placement	SA	Sequence pair represent
Lin et al. (2002)	Area		SA	generalized Polish expression represent
Chan et al.(2004)	Area		B&B	Exact multi-level B&B, subtypes RF, RG considered
Drakidiset al. (2006)	Area	Symmetry	GA	Observed, sequence pair represent
Pisinger (2007)	Area	Placement	SA	Sequence pair represent, semi-normalized placements
Clautiaux et al. (2008)	Area		CP	Exact approach
Korf et al. (2010)	Area	Orientation	CS	Two exact approaches
Chen et al. (2011)	Area, wire length	Regularity	SA	Sequence pair represent

2. PROBLEM STATEMENT

Layout design of spacecraft module belongs to scheme design problem, which has been proved to be NP hard. This problem has not only computing complexity but also engineering complexity, and it is more difficult to tackle the challenge of practical application in engineering. In practical engineering, the dimension of satellite configuration is usually unknown and needs to be optimized (generally minimized) as well, while the dimensions of satellite components are known. Assumed that the material densities of all modules are same, the layout design of satellite module can be simplified to a 2D open dimension rectangle packing problem when the satellite configuration is cuboid.The modules in the satellite are all simplified to rectangles, and the satellite's frame is simplified to enveloping rectangle. The layout design of the spacecraft module can be stated as the following mathematic problem.

Given a set of n rectangular items $i(1 \le i \le n)$ with each item having the width w_i and the height h_i, the RPAMP requires determining a feasible arrangement of all the items on a larger rectangular plane [16].

The item i_{th} can be represented by the bottom-left vertex (x_{i1}, y_{i1}), the height h_i and the width w_i. The coordinates of other three vertexes are upper-left vertex (x_{i2}, y_{i2}), upper-right vertex (x_{i3}, y_{i3}) and bottom-right vertex (x_{i4}, y_{i4}) respectively. Denote $Q = \{q_1, q_2, q_3 ... q_n\}$ as the sequence of theitems and the size of the q_{ith} item can be defined by w_i and h_i. Denote S as the area of the enveloping rectangle.

As an optimization problem, the design variable is the sequence of the items to be packed, the objective function is the enveloping area of the envelope rectangle and the constraint is that the arrangement must be feasible. An arrangement is said to be feasible when no items overlap and all items are placed completely within the container and parallel to the container edges. The optimization problem can be formulated as follows:

$$design\ variable: Q = \{q_1, q_2, q_3 ... q_n\}$$
$$f = \min S(S = w_o(Q) * h_o(Q))$$
$$s.t.$$
$$(1) 0 \le x_{ik} \le w_o, 0 \le y_{ik} \le h_o, k \in \{1,3\}, i, j = (1,2,3...n), i \ne j$$
$$(2) \max\{x_{i1} - x_{j3}, x_{j1} - x_{i3}, y_{i1} - y_{j3}, y_{j1} - y_{i3}\} \ge 0, i, j = (1,2,3...n), i \ne j$$
$$(3) x_{i3} - x_{i1} = w_i, y_{i3} - y_{i1} = h_i, i, j = (1,2,3...n), i \ne j$$
$$(4) h_o / w_o \le 1.2, w_o / h_o \le 1.2$$

In the constraints (1)-(3), w_o and h_o are the width and the height of the enveloping rectangle respectively, which are the functions of the sequence Q and calculated by SBFFA which will be introduced in detail in next section. Constraint (1) implies that all items are placed in the enveloping rectangle. Constraint (2) implies that there is no overlap between any two items. Constraint (3) implies that each item should be placed orthogonally in the coordinate system. Constraint (4) implies that the aspect ratio of the enveloping rectangle should be less than 1.2.

3. SELF-BOUNDARY FALL FREE GENETIC ALGORITHM

This section presents a self-boundary fall free genetic algorithm to address the rectangle packing area minimization problem. For a specific order of the rectangles, the self-boundary fall free algorithm is used to layout the rectangles and calculate the area of the layout. Comparing with the best fit algorithm (BFA, the greedy construction method proposed in [17]), the combination of genetic algorithm and SBFFA includes not only the human experience but also the intelligent search.

3.1. SBFFA

This section presents the self-boundary fall free algorithm (SBFFA), which is the core of the entire algorithm. Self-boundary implies that the boundary of the enveloping rectangle is determined by the packing items rather than being predefined.The self-boundaryalgorithm is based on the followingdefinitions:

Definition 1: feasible corner (feasible point)

The first two feasible cornersare the bottom-right vertex and upper-left vertex of the first rectangle. And the meaning of the feasible corner is the place where the next rectangle can be arranged. Hence, the bottom-left vertex of the next rectangle should be placed on the feasible corner and the occupied feasible corner should be replaced by the new feasible corner, namely the bottom-right vertex and upper-left vertex of the next rectangle.The rectangle whose bottom-left vertex or the upper-right vertex is feasible corner is feasible rectangle.

Definition 2: feasible space

The feasible space is the space where the next rectangle can be arranged in, including the coordinates of the feasible points where the bottom left vertex of the next item can be placed and the widths of the feasible space. In general,the widths of the feasible space are the widths of the feasible rectangles. When the next rectangle is placed besides the feasible rectangle, the widths of the feasible width should be extended.

Definition 3: match

Match is a definition used to estimate whether the next rectangle could be placed in the feasible space. The feasible corner is sorted in ascending order of their ordinates and so does feasible space. If the width of the next rectangle is greater than the width of the first feasible space, it means that they do not match and the repeat the comparison successively until the next rectangle could be placed in the present feasible space.

Definition 4: self-boundary

In generic method, an ordered list of potential container widths for 2D-KPinstances is provided in advance by a dedicated heuristic. As mentioned above, Self-boundary implies that the boundary of the enveloping rectangle is determined by the packing items rather than being predefined. For example, the boundary of the enveloping rectangle in Fig 1(a) is w_o ,which is the abscissa of the bottom-right vertex of the second rectangle. It's the innovation of this paper and it makes the layout more compact.

The schematic graph of SBFFA is shown in Fig 1 and the main procedure is as follows:

Step 1: the first two items are placed side by side;

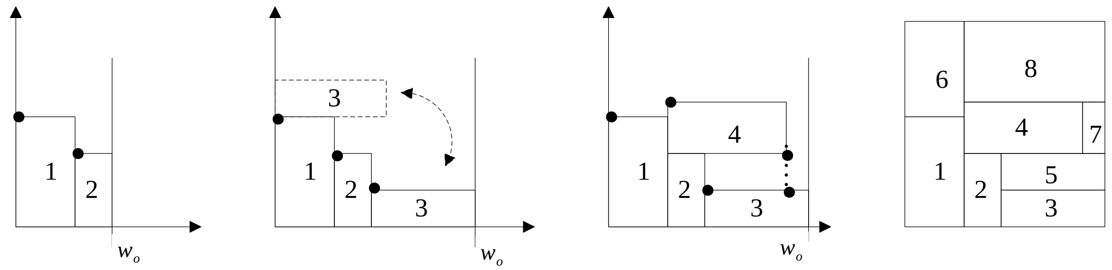
Step 2: the information of the feasible space where the next item can be placed is recorded, including the coordinates of the feasible points where the bottom left vertex of the next item can be placed and the widths of the feasible space. The feasible points include the upper left vertex and the bottom right vertex of each item which have already been packed as the dot symbols shown in the Fig 1(a). And if the next item overhangs the one under it, the point which is the intersection of the extension cord and the sideline of the item below is added into P as well, as shown in Fig 1(c). If one of the feasible points has been occupied by the next item, the point becomes useless and delete from the set P . Denote the feasible point set as $P=\{p_1, p_2 \dots p_k\}$, where the points are sorted in ascending order of the height. Denote the widths of the feasible space are $W=\{W_1, W_2 \dots W_k\}$.

Step 3: Place the next item in the feasible space. The lowest feasible point is preferred where the bottom left vertex of item should be attached. If the lowest point is unfeasible for the item, the point in $P=\{p_1, p_2 \dots p_k\}$ is selected in turn. If there is no feasible space for the item, the i_{th} item should be placed on the x-axis and the boundary of the enveloping rectangle w_o should be replaced by $w_o + w_i$, as shown in Fig 1(b). The preceding steps are repeated till all the items are packed, like Fig 1(d).

3.2. Genetic Algorithm

Genetic algorithm is a mature intelligent algorithm. The key steps of the genetic algorithm are: 1) generate initial population, 2) calculate the fitness of individuals and select superior ones, 3) crossover and mutation, 4) new generation and repeat.

Fig 1 The schematic graph of SBFFA.



(a) (b) (c) (d)

The main key points of the genetic algorithm in this paper are listed below;

(1) The coding scheme is integer coding. And the chromosome is the serial number of the items.

(2) The crossover is explained by the Fig 2.The bold part is the part selected to cross, and the italic part is the part crossed.

Fig 2 The illustration of crossover.

Parent 1	1	**2**	**3**	**4**	5	6
Parent 2	3	**5**	**1**	**2**	6	4
Child 1	1	*3*	*2*	*4*	5	6
Child 2	3	*1*	*2*	*5*	6	4

(3) The probability of crossover is 0.9 and the probability of mutation is 0.1

In general, Based on the SBFFA to calculate the minimum enveloping rectangle space with given sequence of packing items. GA is used to solve the packing optimization problem by searching the optimal item sequence. In the GA&SBFFA algorithm, GA products the serial number of the items. Then the SBFFA places the items logically and outputs the area of the enveloping rectangle. The pseudo-code of GA-SBFFA is as follows:

Table 2Algorithm. GA&SBFFA.

Algorithm GA&SBFFA

input: number n , the widths and heights of rectangles $(h_i, w_i, i = 1,2,3...n)$

and the serial number of all rectangles $Q = \{q_1, q_2, q_3...q_n\}$.

output: the minimum area of the enveloping rectangle S .

//initialize layout

Generate initial serial number Q by function Initial-generation.

According to this sequence, place the first two items on the x-axis. And the bottom-left vertexes are (0,0),($w(q_1)$,0).

Then, record the current feasible points $P = \{p_1, p_2 \dots p_k\}$ (ascending order of their ordinates) and the widths of the feasible space $W = \{W_1, W_2 \dots W_k\}$.

The initial boundary $w_o = w(q_1) + w(q_2)$;

for i=3:n

for j=1:k

if $w(q_i) < W(j) \&\& w(q_i) < w_o \&\& Wk < 1.2 w_o$

place the i_{th} item on the j_{th} point; **end**

else place the i_{th} item on the x-axis and $w_o = w_o + w(q_i)$ **end**

output S

// genetic algorithm

S is the fitness function of the GA

Crossover and **Mutation**

Select the minimum S

4. EXPERIEMNTAL RESULTS

This paper has performed two computational experiments. All the experiments run on the computer with 2.40 GHz, CPU 2.0 memory and Window 7 operation system.

In the first experiment, the optimal layout is known and the filling rate is 100%. Five instance have been performed in experiment 1, and the results of are presented in Table 3.

Table 3 Experiment 1: The optimal layout is known and the filling rate is 100%.

n	fr/%	t/s
9	100	1.15
11	100	1.35
20	100	6.48
30	100	15.1
50	100	32.2

Note: n is the number of items; fr is the filling rater; t is the running time.

In the second experiment, the new RPAMP instances were generated at random by means of following characteristics and definitions proposed by Bortfeldt [16].

(1) Number of rectangular items is n .Those 5 instances were generated with 50 rectangles.

(2) Maximum aspect ratio of all rectangles of an instance $\rho = \max(\rho_i, i = 1,2...n)$. Here ρ_i is the aspect ratio of the i_{th} rectangle. Here $\rho = 2$. Hence ρ represents an upper bound of the proportion of the sides of each rectangle and it guarantees so to speak that rectangles do not degenerate into line segments.

(3) The heightsand widths of the rectangles are taken randomly from the interval [1, 20].

The results are shown in Table 4, and the results in the last row of the Table 4 are the average (ave) filling rate (fr) and average running time (t).

Table 4 Experiment 2: Aspect ratio is $2 = \max(\rho_i, i = 1,2...n)$ and $(h_i, w_i) \in [1, 20]$

n	Bortfeldt[16]		Kun He [2]		GA&SBFFA*		GA&SBFFA	
	fr/%	t/s	fr/%	t/s	fr/%	t/s	fr/%	t/s
50	98.65	2501	99.19	1027	96.43	28	95.96	36
50	97.74	1553	98.50	1279	96.08	33	93.81	36
50	98.60	1697	98.55	1161	96.56	33	95.20	38
50	97.79	1556	97.12	1460	97.54	30	94.55	30
50	98.51	1785	97.73	1421	96.96	32	95.22	32
ave	98.26	1818	98.22	1296	96.66	32	94.95	34

The result of GA&SBFFA* is the result without the constraint of aspect ratio, while the aspect ratio of GA&SBFFA is set to less than 1.2. Comparing Fig3 (b) and Fig3(c), the layout shape has been optimized. Although the filling rates in this paper is a little less than the results in article [2] and [16], the computation time of the articles is about 35 times of the method proposed in this paper. In the actual situation,for example in the satellite design, considering the wiring, heat dissipation and electromagnetic compatibility, the filling rate is always less than 90%.Hence, the results in Table 3 and Table 4 satisfy the layout requirements of the actual situation. And the computational expense is important in the practical design as well. Considering layout shape and the actual demand of the filling rate and the computational expense synthetically, the algorithm proposed in this paper is more practical.

Fig3(a) The result of experiment 1(n=20).

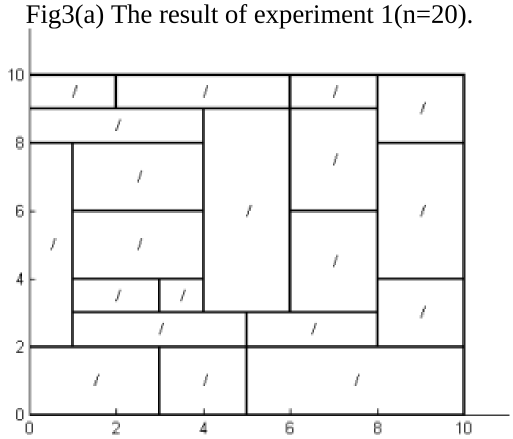


Fig 3(b) The result of GA&SBFFA* in experiment 2.

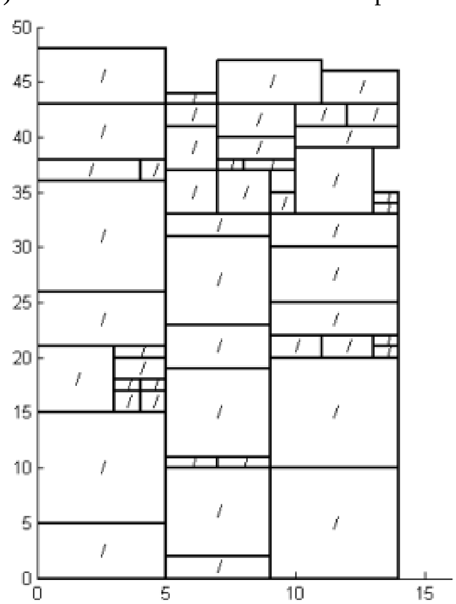
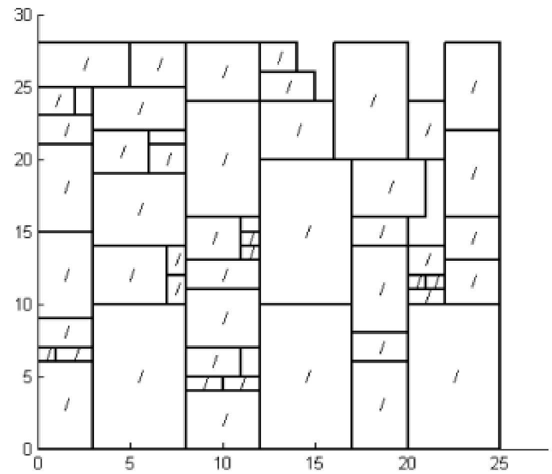


Fig 3(c) The result of GA&SBFFA in experiment 2.



5. CONCLUSION

This paper presents an effective algorithm for the open dimension packing problem. By dynamically changing the boundary of the enveloping rectangle, SBFFA reduces an RPAMP instance to a serial of RPP instance. This paper considered the actual demands of the layout shape, and the results of two experiments showed the high effectiveness of the proposed GA&SBFFA. When the optimal layout is known, all of the filling rates are 100%. When the widths and heights of the items are random, all of the filling rates are more than 94%. Most importantly, the computational expenses were reduced to about 30s when there are 50 items, which are much less than the reported methods.

The method proposed in this paper is valuable for the layout of the satellite modules. Based on the CAD model of satellite modules, the compact and constraint- satisfied layout is obtained by the SBFFA.

6. REFERENCES

[1] Korf RE, Moffitt MD, Pollack ME. Optimal rectangle packing. *Annuals of Operations Research*, 179(1): 261-295. 2010.

[2] Kun he, PengliJi, Chumun Li. Dynamic reduction heuristics for the rectangle packing area minimization problem. *European Journal of Operational Research* 241(2015): 674-685. 2015.

[3] Lin,J.M., Chang,Y.W.TCG:A transitive closure graph-based representation for general floor-plans. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*,13(2), 288–292. 2005.

[4] Wu,Y.L., Chan,C.K. On improved least flexibility first heuristics superior for packing and stock cutting problems. *Stochastic algorithms: Foundations and applications*. Lecture Notes in Computer Science, 3777, 70–81. 2005.

[5] Leung,S.C., Zhang,D., Sim,K.M. A two-stage intelligent search algorithm for the two-dimensional strip packing problem. *European Journal of Operational Research*, 215(1), 57–69. 2011.

[6] Murata, H., Fujiyoshi, K., Nakatake, S., Kajitani, Y. VLSI module placementbased on rectangle-packing by the sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 15, 1518 – 1524. 1996.

[7] Guo, P.N., Cheng, C.K., Yoshimura, T. An O-tree representation of non-slicing floorplans and its applications. In: Proc. of ACM/IEEE Design AutomationConference, pp. 268–273. 1999.

[8] Tang, X., Wong, D.F. FAST-SP: a fast algorithm for block placement basedsequence pair. In: Proc. of the Design Automation Conference, pp. 521 – 526. 2001.

[9] Lin, J.-M., Chang, Y.-W. TCG-S: Orthogonal coupling of P-admissible representations for general floorplans. In: Proc. of the Design AutomationConference, pp. 842 – 847. 2002.

[10] Chan, H.H., Markov, I.L. Practical slicing and non-slicing block-packingwithout simulated annealing. In: Proc. of the ACM Great Lakes Symposium onVLSI, pp. 282–287. 2004.

[11] Drakidis, A., Mack, R.J., Massara, R.E. Packing-based VLSI module placementusing genetic algorithm with sequence-pair representation. IEEE Proceedings:Circuits Devices and Systems 153, 545–551. 2006.

[12] Pisinger, D. Denser packings obtained in O (nloglogn) time. *INFORMS Journal on Computing*. 19, 395–405. 2007.

[13] Clautiaux, F., Jouglet, A., Carlier, J., Moukrim, A. A new constraintprogramming approach for the orthogonal packing problem. *Computers and Operations Research.* 35, 944–959. 2008.

[14] Korf, R.E., Moffitt, M.D., Pollack, M.E. Optimal rectangle packing. *Annals of Operations Research.* 179, 261–295. 2010.

[15] Chen, X., Hu, J., Xu, N. Regularity-constraint floorplanning for multi-coreprocessors. In: Proc. of the 2011 International Symposium on Physical design(ISPD ’11), ACM, New York, 2011. 2011.

[16] Bortfeldt,A. A reduction approach for solving the rectangle packing area minimization problem. *European Journal of Operational Research,* 224(3), 486–496. 2013.

[17] He,K., Huang,W., Jin,Y. An efficient deterministic heuristic for two dimensional rectangular packing. *Computers Operations Research*, 39(7), 1355–1363. 2012.