International Journal of Control and Automation
Vol. 9, No. 8 (2016), pp.369-380
http//dx.doi.org/10.14257/ijca.2016.9.8.35

A Novel Grouping Particle Swarm Optimization Approach for 2D Irregular Cutting Stock Problem

Yanxin Xu

Automation Department of Shanghai Jiaotong University
iamxuyanxin@sina.com

Abstract

The Cutting stock problem is an important problem that arises in a variety of industrial applications. The problem includes a grouping phase (the grouping problem). This research adopts a novel grouping particle swarm optimization approach for grouping problem. In this article, new particle position and velocity updating equations are developed according to the structure of grouping problem for cutting stock problem. The approach works with groups of items instead of items exclusively, compared with conventional method for cutting stock problem. The proposed approach is tested on a set of real instances from the ship building industry with very good results. Moreover, the efficacy and efficiency of the proposed approach are indicated on a set of randomly generated instances.

***Keywords**: grouping problem; PSO; GPSO; irregular cutting stock problem*

1. Introduction

The cutting and packing problems are widely found in industrial applications such as the cutting of standardized stock units in the wood, steel and glass industries, packing on shelves or truck beds in transportation and warehousing, and the layout of articles in newspapers [5, 15]. There are many classical cutting and packing problems such as knapsack problem, bin packing problem and cutting stock problem. A useful classification of cutting and packing problem is proposed in Wäscher *et al.* (2007) [32]. Their classification partitioned the problems by dimensionality, objective of the assignment, large objects and small items. This research focuses on the cutting stock problem with irregular shapes. Using the typology of Wäscher, this is a two-dimensional Single Stock-Size Cutting Stock Problem (2DCSP).

Cutting stock problem is proved to be a NP-hard combinatorial optimization problem whose basic idea and main goal consists of fitting a finite number of items into a minimum number of stocks, subject to a practical set of restrictions and requirements [16]. Many solution methods have been proposed for the problem, including optimization approaches (*e.g.* linear programming, column generation), heuristic and meta-heuristic approaches, and the emerging approaches combining these methods into a solution approach.

Linear programming (LP) and Column Generation (CG) was first used to solve multistage cutting stock problem by Gilmore and Gomory (1961) [17]. Branch-and-bound and dynamic programming (DP) has also been applied extensively in this problem setting [29]. Due to the NP-complete nature of the problem, published solution approaches mostly focus on heuristic and meta-heuristics methodologies. Heuristic placement strategy is proposed to supply a rule for items to be placed on sheet, such as next-fit, first-fit, best-fit, bottom-left (BL) and bottom-left-fill [6, 7, 21, 26]. They are fast in generating a solution packing plan and particularly useful for problems with a high complexity, since deterministic methods have restrictions in applications involving an astronomical number of variables. But the quality of the solution is highly dependent on the input sequence of

ISSN: 2005-4297 IJCA

items. Meta-heuristics have also been employed to solve CSP, including simulated annealing, tabu search, neural networks, Evolutionary Algorithm (EA) and swarm intelligence Algorithm [1, 8, 18, 19, 25, 33]. They are general frameworks in solving combinatorial optimization problems by hybridizing with a heuristic placement strategy. In this two-stage approach, meta-heuristics searches for optimal input sequence of items and heuristic placement routine determines how the input sequence of items is allocated into stock sheets.

All the methods introduced above focus on items solely. However, CSP can be recognized as a grouping problem. Therefore, the method for grouping problem can also be utilized to solve CSP. In 1994, Falkenauer [14] firstly applied Grouping Genetic Algorithm (GGA) for solving grouping problems. Thereafter, GGA is widely applied for a variety of grouping problems. Swarm intelligence algorithms, especially particle swarm optimization (PSO), have received more and more attention in recent years [2, 4, 20]. This research employs GPSO to replace GGA to solve grouping problems. In this article, the proposed GPSO incorporate EA concepts such as the use of mutation operator as a source of diversity (called GEPSO), in order to change the internal structure of the particle to further optimize the solution.

Moreover, this research focuses on CSP with irregular shapes. Thus, the tools to assimilate the geometry and to detect feasible regions to pack items with complex shape are necessary in this area. In recent years, some effective approaches are proposed and used to check/generate feasible regions to pack items of irregular shape, such as NFP (No-Fit Polygon), Direct Trigonometry, Phi Function, Rectangle Enclosure and Grid Approximation [9, 10, 11, 13, 22, 28, 30, 31]. A detailed tutorial on the geometry of nesting problems is given in Bennell and Oliveira (2008) [12]. In this article, we focus on grid approximation method.

This paper is organized as follows. The proposed GEPSO algorithm and its various features are described in Section 2. In Section 3, a performance comparison with real data from industry and conventional method for CSP, GPSO is also provided to show the effectiveness and efficiency of GEPSO in solving CSP. The research is concluded and possible issues for future work are suggested in Section 4.

2. The Grouping Particle Swarm Optimization Algorithm for CSP

Since GPSO is to solve grouping problem, it must focus on group rather than items separately. The key issues are how to construct a special grouping encoding taking into account the structure of grouping problem. These are also the main differences between GPSO and classic PSO algorithm. The proposed GPSO is a PSO-based algorithm which incorporates EA concepts such as the use of mutation operator as a source of diversity. It is referred to D. Liu *et al*. (2008). For the success of GPSO, an appropriate particle representation, updating equations, mutation operator and heuristic placement method (Hybrid Method (HM), introduced in section 3.5) are adopted, which are introduced in detail in the following sections. The flowchart of GPSO is shown in Figure 1.

2.1. Solution Encoding

An effective encoding strategy is important for the success of GPSO. In general, the solution encoding of grouping problem consist of two parts, namely: an item part and a group part. The item part consists of an array of size n (n is the number of items). The group part consists of a permutation of D group labels. Each member in the item part can take any of D group tags, indicating that the corresponding item belongs to the group of the given tag [23]. In grouping problems, groups are the meaningful building blocks. It's meaningless to take one item solely during search process.

When optimizing a numerical function by PSO, each solution is generally represented with a vector of length D of real numbers (D is the search dimension), where each value is corresponded to one variable. In a similar way, when solving a grouping problem with PSO, one can represent a particle solution composed of D groups as a structure whose length is equal to the number of groups. Here, groups play the role of variables. In classical PSO algorithm, the location of particle in d-th dimension represents the value of d-th variable. It is similar to classical PSO algorithm to determine the items which belong to d-th group in GPSO algorithm. Note that the feasible solutions to grouping problems are not essentially all at the same length. This implies that GPSO has to handle solutions with variable length.

In the proposed GEPSO, an order-based variable length particle structure is adopted as the representation of a group for CSP. In other words, particle consists of a list of groups {g1, g2, ⋯ gd}， where the j-th group is a stock assignment representing the set of items assigned to the j-th stock sheet (1⩽j⩽d). Each particle encodes the packing solution including the sequence number of stocks used and the permutation of the items packed into this stock by the HM heuristic. Each stock must be allocated at least one item which must not be found in any other stocks. And the permutations of items in every stock sheet are unique. This encoding structure allows the algorithm to manipulate the permutation of items in each stock without affecting other stocks.

Specifically, each particle consists of several strings of distinct integers. Each string represents a stock and each integer represents an item. The pbest and gbest are encoded in the form of extra stocks stored in the particle memory. The pbest is defined as the particles' own best stock due to their own best values in the solution space. The gbest is defined as the best stock found by the population. We use the best stock instead of best solution as pbest and gbest in the proposed GEPSO in order to help the algorithm avoid falling into random search by transmitting good stock among generations. Furthermore, a state variable is attached to every index in the permutation so as to encode for orientation of item.

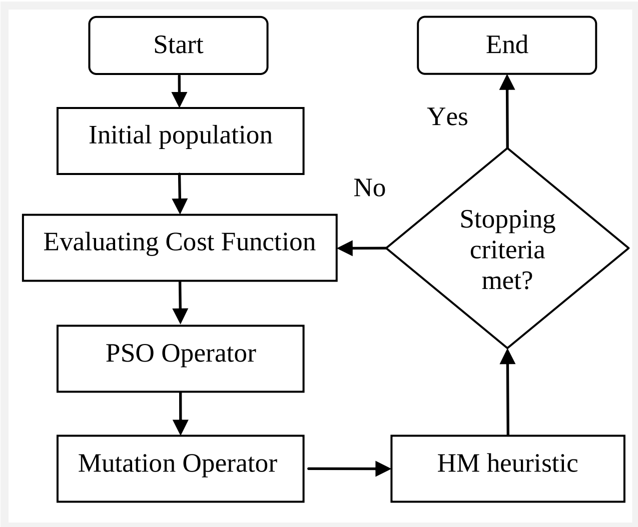


Figure 1. Flowchart of GPSO for CSP

2.2. Initialization

The first step of our approach is initialization. When evolutionary computation is applied to solve NP-hard problems, the best solution is obtained by getting initial population completely randomly. However, the computation time is expensive. 81 different rules for sequences in solving cutting stock problems are studied by Abdelhafiez (2008) [27]. It is found that the decreasing area and the decreasing length sequences give efficient results with all problem sets. Therefore, the initial sequence of the items is firstly determined in non-increasing order according to their area and their length.

In this research, each item is represented by a list of vertex coordinates $[(x_1,y_1),\ldots,(x_M, y_M)]$, where M is the number of vertices. During the allocation process, the degree of overlap among items on the stock sheet should be tested. The grid approximation, a digitized representation technique, is used to conquer these problems. This matrix representation method was proposed by Dagli. The detailed technique applied in the paper is referred to Dagli (2004) and Z.X. Guo (2009) [24].

The item with a two-dimensional matrix of size $A_W^{(k)} \times A_L^{(k)}$ is represented as follows:

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1A_L^{(k)}}^{(k)} \\ a_{21}^{(k)} & a_{22}^{(k)} & \cdots & a_{2A_L^{(k)}}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{A_W^{(k)}1}^{(k)} & a_{A_W^{(k)}1}^{(k)} & \cdots & a_{A_W^{(k)}A_L^{(k)}}^{(k)} \end{pmatrix}$$

Where, L and W are the length and the width of the stock sheet.

$$A_W^{(k)} = \frac{P_W^{(k)}}{R} \quad and \quad A_L^{(i)} = \frac{P_L^{(k)}}{R};$$

For each entry, $$a_{i,j}^{(k)} = \begin{cases} 1, & \text{if pixel } (i, j) \text{ is occupied} \\ 0, & \text{otherwise} \end{cases}.$$

Similar to the item representation, the stock sheet is able to be discretized into a finite number of equal-size pixels of size R^2. Hence, the stock sheet is characterized by a matrix U of size $U_W \times U_L$ as follows:

$$U = \left[u_{i,j}\right]_{U_W \times U_L},$$

Where, $$U_W = \frac{W}{R} \quad and \quad U_L = \frac{L}{R}$$

For each entry, $$u_{i,j} = \begin{cases} 1, & \text{if pixel } (i, j) \text{ is occupied} \\ 0, & \text{otherwise} \end{cases}.$$

By using this technique, each item is enclosed by an imaginary rectangle for the sake of obtaining the reference points during the nesting process. Then, this particular rectangular area is divided into a uniform grid called pixel. In the case, the value of a pixel is ‘1’ when the material of the sheet is occupied. Otherwise the value of the pixel is ‘0’. $P_L^{(k)}$ and $P_W^{(k)}$ denote the length and the width of an enclosing rectangle corresponding to the item p_k. R denotes the square side of a pixel (In this paper, R=0.085mm).

A value in the stock sheet matrix which is greater than one is an indication of an overlap. Although the overlap test can be performed easily and quickly, there is a need for a large memory for this representation scheme. Consequently, only required matrices are generated by the algorithm while the overlap test is performed.

Furthermore, the initial orientation for each packing item is confirmed by the MRE (Minimum Rectangular Enclosure (Jakobs 1996)) of each item. For each item, a rectangular enclosure, which can accommodate the whole irregular-shaped item, is used as an approximation of the area required to allocate the item on a stock sheet. An item with different orientations may produce different sizes of rectangular enclosures. The minimum enclosure found defines the lower bound of the area required to allocate the item. It is worth mentioning that the approach does not replace the item by their MREs. Rather, it uses MREs as additional information for orienting the items. Replacing irregular shapes with their MREs would generate less effective solutions and wasting significant quantities of material.

In order to find the minimum rectangular enclosure of an item, 90 different orientations (0-89) of the item are obtained. Then the angle produces MRE is selected as the initial orientation of the item by using list of vertex coordinates representation. The rotated item is obtained as follows:

$$\begin{bmatrix} x_{k\theta} \\ y_{k\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

Where (x_k, y_k) denotes the coordinates of the *k*th vertex of the original item and ($x_{k\Theta}, y_{k\Theta}$) denotes the coordinates of the *k*th vertex of the rotated item. Thus, the area of rectangular enclosure of the angle Θis computed as:

$$RC_\theta = \left[\max |_{k=1}^{M} \left\{ x_{k\theta} \right\} - \min |_{k=1}^{M} \left\{ x_{k\theta} \right\} \right] * \left[\max |_{k=1}^{M} \left\{ y_{k\theta} \right\} - \min |_{k=1}^{M} \left\{ y_{k\theta} \right\} \right]$$

Among the 90 different rectangular enclosures, the size of MRE is：

$$RC_{\min} = Min |_0^{89} RC_\theta .$$

Then the layout phase arrives, which is introduced as followed. If a displayed equation needs a number, place it flush with the right margin of the column (*e.g.,* see Eq. 1).

2.3. Updating Equations in GEPSO

In PSO, each individual in the population will try to emulate the gbest and pbest solutions through updating by PSO equations [3]. How to emulate the pbest and gbest is also the key question in 2D-CSP. As described in section 2.1, packing quality information is kept in the stocks of each solution. As a result, our strategy is to set best stock as pbest and gbest, and let other stocks being repacked. In this way, best stock information is kept. The updating equations of the proposed GEPSO are shown as followed：

$$V_{id} = \alpha * P_{id} + (1-\alpha) * P_{gd}, \ \alpha = 0,1, \qquad (2.3.1)$$

$$X_{id} = X_{id} + V_{id}. \qquad (2.3.2)$$

GPSO is characterized by the fact that particle movement is directed by either personal best or global best only in each instance. This is contrary to existing works where particle movement is influenced by both personal and global best at the same time. The velocity is governed by either pbest or gbest as shown in Eq. (2.3.1). Which one is the governor is determined by the value of α. Then the movement of solution particle is confirmed by inserting the stock into the solution particle as the first stock and any duplicate items in other stocks will be deleted.

2.4. Generation of a New Particle Solution

The process of the proposed GEPSO is described as followed. In the initialization process, items with sorted sequences and initial orientations are packed using Hybrid Method (HM) into the form of particles (groups). Once initialization finished, the particles are evaluated against the fitness function. After selecting either their personal best stock or global best stock as the velocity vector, particles are updated by the following operations: inserting the stock represented by the velocity vector as the first stock, and deleting duplicate items in other stocks. At their new positions, the particles then undergo specialized mutation operations. If any constraint is violated during mutation, the stocks violating constrains and the two least filled stocks will be selected for repacking using HM.

The purpose to adopt mutation operators in GPSO is to change the internal structure of the particle so as to further optimize the solution. The proposed mutation operators consist of two modes: partial swap and merge stocks mechanism. In the first mode, sequences of items in two stocks can be randomly cut and exchanged. The operation is used to search for more closely packed items. The items in the two least filled stocks will be merged into one stock in the second mode. The operation may help reduce the number of stocks used. Firstly, which mode is applied is decided by a random-generated number. Then the other two modes of mutation are employed. The first one randomly rotates items in the stock, and the other one shuffle the items of the stock in order to improve the packing configuration and the stability of the stocks. Then a feasibility check is carried out to ensure the satisfaction of all the constraints. Whenever a constraint is violated, the particle eliminates two emptiest stocks, in which the items miss from the solution and be inserted back in a random order By HM heuristic. This is helpful to confirm that all particles are valid solutions of 2D-CSP. The mutation mode for a single particle is showed in Figure 2.

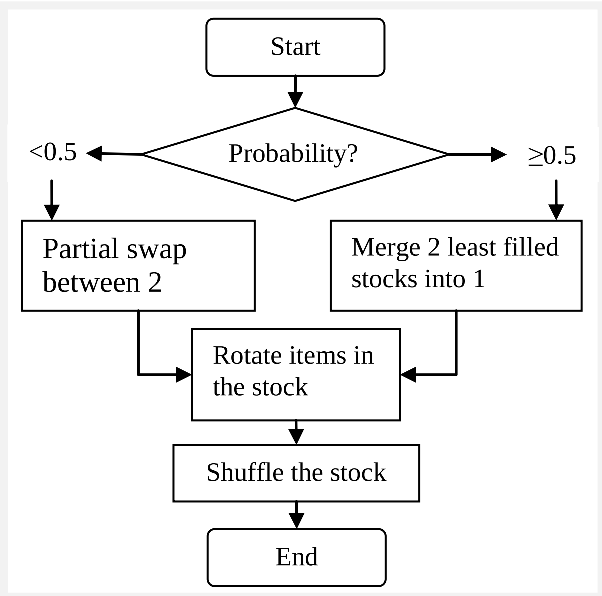


Figure 2. Mutation Mode for a Single Particle

2.5. Hybrid Method with RBL and Two-Stage Placement Strategy

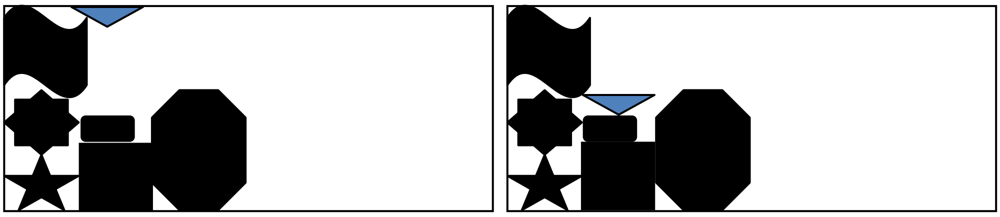
This research adopts a heuristic placement hybrid method (HM), which combines a Revised Bottom-Left (RBL) heuristic and a two-stage placement strategy. The RBL is used to allocate the current items on stock sheet without affecting any other allocated items. When the stock sheet is empty， RBL places the first packing item at the lower left hand corner. The following items are allocated along the Y axis until there is no enough space that can be used to place any item, then, a new row of items along X axis is formed.

The purpose of RBL is to minimize the total required area, namely the objective value. It can be stated as:

$$\text{Objective value of RBL} = x * W + y \quad (2.5.1)$$

Where x is the coordinate of the item along the X axis of the stock sheet's matrix representation; y is the coordinate of the item along the Y axis. W is the width of the stock sheet.

Since different orientation of item may create different result, several orientations are generated by rotating it by 0°, 90° and 180°. Based on the objective value, the candidate item needs to move to the left most possible on the stock sheet, which is the minimum value along X axis. However, we also need to consider another objective, since item is possible to be allocated in other position along Y axis, where requires less area. As a result, item is moved downward toward origin until the following conditions occur: new move produces more overlap area, or the resulting pattern cannot be fit within the current stock sheet. These movements are performed for all orientations, but the final selection is the one with minimum objective value. An example of RBL is shown in Figure 3. As shown, the blue triangle is allocated in the final position in Figure (b) instead of Figure (a) by considering the objective in formulation 2.5.1.



(a) (b)

Figure 3. An example of RBL

The RBL is referred to Dagli (2004), but we combine it with two-stage packing strategy in this article. In the two-stage packing case, the enclosing rectangles of the packing items are first examined, and then the packing items are compacted directly. The differences are that the compaction routine is done when each enclosing rectangle is placed rather than implement it in a single step after all the enclosing rectangles are allocated. This compaction routine is able to obtain a tighter packing pattern and provide more space for the unpacked items. The hybrid method improves the packing pattern quality without more computational effort.

3. Performance Evaluation

In this section, we test the performance of the proposed GEPSO against other algorithms. Few work for packing problems with items of irregular shape are done in the literature, and especially we can hardly find any related work for the 2DCSP and 2DBPP when pieces have irregular shape. Therefore, it is not so easy to find appropriate algorithm to compare performance. In addition, PSO has never been applied to solve CSP. As a result, in order to illustrate the applicability of the proposed GEPSO algorithm, we first test our algorithms on ten real instances provided by our industrial partner in ship building industry. Furthermore, we also design a group of randomly generated instances and compare the solutions obtained by GEPSO and GPSO algorithm. And they are also compared with the conventional method for CSP (CM), which adopts the same initialization process (section 3.2) and HM heuristic (section 3.5) without considering the structure of grouping problem and PSO algorithm. It is worth mentioning that the stock sheet is very expensive in ship building industry, therefore it's more important for saving material than time. Based on this situation, this article does not show the computational time.

All algorithms are implemented in Visual C++ and the tests are performed on a computer with processor Intel Core i5 3.0 GHz, 4 GB of RAM, and Windows XP operating system. The resolution ratio of the items and stock sheets are 300 PPI. First of all, according to the real data from our industrial partner, the results of the proposed GEPSO algorithm are compared with those derived by industrial practice in order to demonstrate the effectiveness of the proposed methodology. Figure 5 shows the shapes of the irregular items used in real instances. Then ten randomly generated instances according to the shapes showed in Figure 5 are designed. In these instances, the results of the proposed GEPSO algorithm are compared with those obtained by GPSO and CM.

In all experiments, the parameters of GEPSO and PSO are introduced as followed: the most appropriate range of the population size is from 100 to 500. Too small population results in less diversified patterns and too large population leads to longer computation time. And maximum number of generations = 1000.

The performance was measured according to the number of occupied stock sheets and packing density (PD). In order to evaluate the resulting solutions, we consider the ratio of the total area of all the packed shapes to the area of the occupied stocks as a performance measure. This measure is called packing density.

$$PD = \frac{\text{Area of Allocated Items}}{\text{Area of Occupied Stocks}}.$$

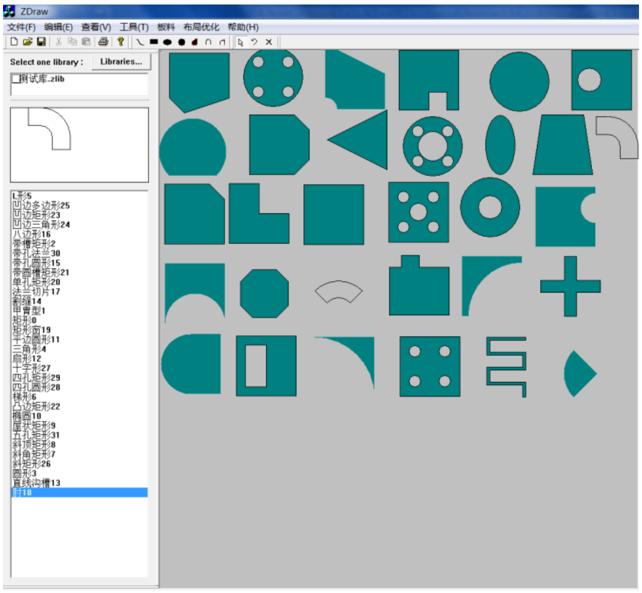


Figure 4. Shapes of Items from Real Instances

Table 1 presents some information about the real instances used in this work. A total of 10 instances were considered for computational tests. The first four columns of this Table have the following information: instance name (Ins.); the number of items (Num); length of the stock sheet (L); width of the stock sheet (W). According to the data from industrial instances, GEPSO is applied to solve these instances. The number of the occupied stock sheets (Num of SS) and PD of all the occupied stock sheets are used to indicate the performance of the packing pattern. Table 1 summarizes the results of the industrial instances solved by GEPSO, and the results marked in bold and italics show its better performance than the real data. The performance of the proposed GEPSO, the real data, and the improvement between them are respectively shown in the last three columns of Table 1. The results indicate that the proposed GEPSO improves the PD of the packing pattern and decreases the number of the occupied stock sheets. Such results demonstrate that our algorithm is able to effectively deal with irregular cutting stock problem of real world in ship building industry.

In order to illustrate the efficacy and efficiency of the proposed GEPSO, ten randomly generated instances are employed. In these instances, the stock sheets are chosen as the same as the real instances in Table 1 and all items are chosen from the shapes in Figure 5.

And it is randomly generated that which shape and how many items of this shape are chosen. Each instance is run ten times by the proposed GEPSO, GPSO algorithm and CM. Table 2 summarizes the average results of these instances, and the results marked in bold and italics show its better performance of GEPSO than the others.

The first four columns of Table 2 present the same information with Table 1. The number of items in these instances is less than the ones in Table 1, in order to reduce the amount of calculation without affecting the comparisons. We still indicate the performance of the packing pattern by the number of the occupied stock sheet and PD. The performance of the packing pattern for the proposed GEPSO, GPSO and CM are respectively shown in the last three columns of Table 2. For this set of instances, both of GEPSO and GPSO get better performance than CM on 8 instances. This shows that exploitation of the structure of grouping problems through using an appropriate particle representation can effectively solve CSP. Although CM can't represent all the method for CSP without grouping thinking, the results still indicate that GPSO has the ability to conquer CSP as the other approaches. Furthermore, the proposed GEPSO outperforms GPSO in PD aspect on 7 instances, although the number of occupied stock sheets cannot be decreased. The results demonstrate the advantages to use mutation operator as a source of diversity to change the internal structure of the particle to further optimize the solution.

Table 1. Results Obtained For the 2CS Based On Real Instances

Ins.	Num	L(mm)	W(mm)	GEPSO		Real Instances		Improvement of PD
				Num of SS	PD (%)	Num of SS	PD (%)	
1	400	4520	2170	45	**87.51**	45	86.24	1.27
2	420	5238	2170	42	**87.68**	42	86.63	1.05
3	300	4340	2170	***44***	**86.58**	45	84.33	2.25
4	480	4488	2170	***44***	**86.33**	45	83.45	2.88
5	500	5236	2170	***47***	**87.09**	48	85.12	1.97
6	900	11880	2170	***57***	**84.20**	58	80.96	3.24
7	1000	11880	2340	***67***	**83.59**	68	79.42	4.17
8	1000	11880	2430	***64***	**83.82**	65	79.86	3.96
9	880	11880	2470	***58***	**85.14**	59	81.78	3.36
10	850	11880	2170	***53***	**84.77**	54	81.75	3.02

Table 2. Results for the Ten Illustrative Instances

Ins.	Num	L(mm)	W(mm)	GEPSO		GPSO		CM	
				Num of SS	PD (%)	Num of SS	PD (%)	Num of SS	PD (%)
1	400	5238	2170	***38***	***85.58***	***38***	84.86	39	83.65
2	400	5238	2170	***40***	80.26	***40***	***80.32***	41	78.11
3	400	5238	2170	***40***	82.56	***40***	***82.62***	41	80.41
4	400	5238	2170	52	77.82	52	76.77	52	***77.86***
5	400	5238	2170	***39***	***83.78***	***39***	81.83	40	80.62
6	800	11880	2470	***50***	***86.96***	***50***	85.33	51	84.12
7	800	11880	2470	64	79.58	64	79.07	64	***79.95***
8	800	11880	2470	***54***	***83.04***	***54***	82.55	55	81.34
9	800	11880	2470	**60**	79.76	**60**	***79.78***	61	77.94
10	800	11880	2470	***55***	***82.66***	***55***	81.53	56	80.32

4. Conclusion

Since CSP can be recognized as a grouping problem, a novel grouping particle swarm optimization algorithm is presented for grouping problem to solve two dimensional irregular cutting stock problems. This presented approach has mainly three features as follows: firstly, the approach focuses on groups instead of items solely. A special solution encoding is proposed to suit the structure of grouping problem. Secondly, PSO is utilized as an evolutionary search engine. Furthermore, a specialized mutation operator is added to PSO as elements of evolutionary algorithm. Thirdly, grid approximation method provides an easier way to detect whether overlap occurs, compared with other geometrical tools. The two-stage placement strategy improves the packing pattern quality without compromising the computational effort. The proposed GEPSO is assessed on 10 real instances from ship building industry and 10 randomly generated instances. The results demonstrate that GEPSO can deal with not only CSP but practical problems with complex shapes and even outperform than conventional method.

Acknowledgements

The work was supported by National Natural Science Foundation of China, No. 61290323, No. 61304214, No. 61290322 and No. 61203178.

References

[1] R. Alvarez-Valdes, F. Parreño and J.M. Tamarit, "A tabu search algorithm for two-dimensional non-guillotine cutting problems", European Journal of Operational Research, vol. 183, no. 3, **(2007)**, pp. 1167–1182.
[2] H.M. Abdelsalam and A.M. Mohamed, "Optimal sequencing of design projects' activities using discrete particle swarm optimization", International Journal of Bio-inspired Computation, vol. 4, no. 2, **(2012)**, pp. 100-110.
[3] E. Abdelhafiez, "Efficient Items-Ordering Rules in Cutting Stock Problem", 9th International Conference on Mechanical Design and Production Engineering, MDP9, Cairo -Egypt, **(2008)**.
[4] I. Budinská, T. Kasanický and J. Zelenka, "Production planning and scheduling by means of artificial immune systems and particle swarm optimisation algorithms", International Journal of Bio-inspired Computation, vol. 4, no. 4, **(2012)**, pp. 237-248,.
[5] J.A. Bennell and J. F. Oliveira, "A tutorial in irregular shape packing problems", Journal of the Operational Research Society, vol. 60, **(2009)**, pp. 93–105.
[6] E. Burke, R. Hellier, G. Kendall and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem", Operations Research, vol. 54, no.3, **(2006)**, pp. 587–601.
[7] E.K. Burke, R.S.R. Hellier, G. Kendall and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem", Operations Research, vol. 54, no. 3, **(2006)**, pp. 587–601.
[8] E.K. Burke, G. Kendall and G. Whitwell, "A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem", INFORMS Journal on Computing, vol. 21, no. 3, **(2009)**, pp. 505–516.
[9] E.K. Burke, R.S. R. Hellier, G. Kendall and G. Whitwell, "Irregular Packing Using the Line and Arc No-Fit Polygon", Oper. Res, vol. 58, no. 4, **(2010)**, pp. 1-23.
[10] J.A. Bennell and X. Song, "A comprehensive and robust procedure for obtaining the no-fit polygon using Minkowski sums", Computers and Operational Research, vol. 35, no. 1, **(2008)**, pp. 267–281.
[11] J. Bennell, G. Scheithauer, Y. Stoyan and T. Romanova, "Tools of mathematical modelling of arbitrary object packing problems", Annals of Operations Research, vol. 179, **(2010)**, pp. 343–368.
[12] J. A. Bennell and J. F. Oliveira, "The geometry of nesting problems: A tutorial", Eur. J. Oper. Res, vol. 184, **(2008)**, pp. 397–415.
[13] N. Chernov, Y.G. Stoyan and T. Romanova, "Mathematical model and efficient algorithms for object packing problem", Computational Geometry: Theory and Applications, vol. 43, no. 5, **(2010)**, pp. 535-553.
[14] E. Falkenauer, "New representation and operators for GAs applied to grouping problems", Evolutionary Computation, vol. 2, **(1994)**, pp. 123–144.
[15] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems", European Journal of Operational Research, vol. 141, **(2002)**, pp. 359–370.
[16] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman and Company, New York, **(1979)**.
[17] P.C. Gilmore and R.E. Gomory, "A linear programming approach to the cutting stock problem", Operations Research, vol. 9, no. 6, **(1961)**, pp. 849-859.
[18] A.M. Gomes and J.F. Oliveira, "Solving irregular strip packing problems by hybridizing simulated annealing and linear programming", European Journal of Operational Research, vol. 171, no. 3, **(2006)**, pp. 811–829.
[19] Z. X. Guo, W. K. Wong, S. Y. S. Leung, J. T. Fan and S. F. Chan, "A genetic-algorithm-based optimization model for scheduling flexible assembly lines", International Journal of Advanced Manufacturing Technology, vol. 36, no. 1–2, **(2008a)**, pp. 156–168.
[20] E. García-Gonzalo and J. L. Fernández-Martínez, "A brief Historical Review of Particle Swarm Optimization (PSO)", Journal of Bioinformatics and Intelligent Control, vol. 1, no. 1, **(2012)**, pp. 3-16.
[21] E. Hopper and B. Turton, "An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem", European Journal of Operational Research, vol. 128, **(2001)**, pp. 34–57.

[22] S. Jakobs, "On genetic algorithms for the packing of polygons", European Journal of Operational Research, vol. 88, no. 1, **(1996)**, pp. 165–181.
[23] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, **(1995)**.
[24] A.H. Kasha, M.H. Kashan and S. Karimiyan, "A particle swarm optimizer for grouping problems", Information Sciences, vol. 252, **(2012)**, pp. 81-95.
[25] D. Liu, K. Tan, S. Huang, C. Goh and W. Ho, "On solving multi-objective bin packing problems using evolutionary particle swarm optimization", European Journal of Operational Research, vol. 190, **(2008)**, pp. 357–382.
[26] J.F. Oliveira, A.M. Gomes and J.S. Ferreira, "TOPOS – a new constructive algorithm for nesting problems", OR Spektrum, vol. 22, no. 2, **(2000)**, pp. 263–284.
[27] R. Poli, J. Kennedy and T. Blackwell, "Particle swarm optimization: an overview", Swarm Intelligence Journal, vol. 1, no. 1, **(2007)**, pp. 33-57.
[28] P. Poshyanonda and C.H. Dagli, Genetic neuro-nester, Journal of Intelligent Manufacturing, 15(2): 201–218 (2004).
[29] K. Richter, "Solving sequential interval cutting problems via dynamic programming", European Journal of Operational Research, vol. 57, **(1992)**, pp. 332-338.
[30] W.K. Wong, X.H. Zeng and W.M.R. Au, "A decision support tool for apparel coordination through integrating the knowledge-based attribute evaluation expert system and the T-S fuzzy neural network", Expert Systems with Applications, vol. 36, no. 2, **(2009)**, pp. 2377–2390.
[31] W.K. Wong, X.X. Wang, P.Y. Mok, S.Y.S. Leung and C.K. Kwong, "Solving the two-dimensional irregular objects allocation problems by using a two-stage packing approach", Expert Systems with Applications, vol. 36, **(2009)**, pp. 3489–3496.
[32] G. Wäscher, H. Haußner and H. Schumann, "An improved typology of cutting and packing problems", European Journal of Operational Research, vol. 183, no. 3, **(2007)**, pp. 1109–1130.
[33] C.W.M. Yuen, W.K. Wong, S.Q. Qian, L.K. Chan and E.H.K. Fung, "A hybrid model using genetic algorithm and neural network for classifying garment defects", Expert Systems with Applications, vol. 36, no. 2, **(2009a)**, pp. 2037–2047.

Author

Yanxin Xu, He received the M.S. degree from Haerbin University of Science and Technology, Haerbin, China. He is currently pursuing the Ph.D. degree in the Department of Automation, Shanghai Jiaotong University, China.

His research interests include production scheduling and optimization.