Scientia Iranica E (2017) 24(6), 3287–3296

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

Developing column generation approach to solve the rectangular two-dimensional single knapsack problem

M.A. Hatefi*

Department of Energy Economics & Management, Petroleum University of Technology (PUT), Sattarkhan Ave., Khosrow Jonoubi St., Tehran, Iran.

Received 20 August 2014; received in revised form 4 July 2015; accepted 15 October 2016

KEYWORDS
Cutting;
Packing;
Two-dimensional knapsack problem;
Mathematical programming;
Column generation approach.

Abstract. The rectangular two-dimensional Single Knapsack Problem (SKP) consists of packing a fixed rectangular space (so-called pallet) with a subset of smaller rectangular shapes (so-called pieces) of different dimensions and without rotation. Pieces have different values. The objective is to maximize the sum of the values of the pieces packed. This paper proposes a new method for solving rectangular two-dimensional SKP based on the column generation approach. The mathematical formulation of the proposed model is the simplest of all present mathematical formulations in the the state-of-the art. The computational performance indicates that it is an effective method based on quality of solution.

1. Introduction

Cutting and Packing (C&P) problems are highly encountered in many fields, such as management science, engineering sciences, information sciences, production, and logistics. They arise in several real-world industries such as glass, steel, wood, paper, clothing, and leather. C&P problems are special cases of combinatorial optimization problems [1] and are the common problem of packing a large space with some smaller shapes. Based on problem dimension, there are three different types of C&P problems: one-dimensional, two-dimensional, or three-dimensional; for instance, steel bar cutting in one-dimensional [2], glass sheet cutting in two-dimensional [3], and packing boxes into a container in three-dimensional problems [4]. Concerning piece number limitation, there are two different types of C&P problems: constrained or unconstrained. The characteristic of "constrained" refers to those whose number of the ordered pieces is well known and limited. In other words, the packed pieces exceeding the ordered well-known demands are considered as wasted space or scrap. C&P problems are also classified in two types: weighted or un-weighted. In the weighted type, a piece has a predetermined value, but in the un-weighted one, the value of a piece is concerned with its dimensions.

The present paper is concerned with the problem of packing a number of rectangles (so-called pieces) of different dimensions into a larger stock sheet (so-called pallet), such that the total value of the packed pieces is maximized. The problem, based on Dychoff's coding scheme [1], is denoted as $2/B/O/F$. The "2" indicates that a two-dimensional pattern is considered. The aim is to pack a subset of the pieces "B" within one pallet "O", and we have relatively few "F" non-identical types of pieces. Besides, according to the typology of Wascher et al. [5], the problem falls into the rectangular two-dimensional Single Knapsack Problem (SKP). The paper introduces a new mathematical formulation

*. *Tel.: +98 21 44260128; Fax: +98 21 44214222*
E-mail address: Hatefi@put.ac.ir

doi: 10.24200/sci.2017.4401

based on the column generation approach [6,7]; so, this research belongs to the approach of mathematical programming formulation.

The rectangular two-dimensional SKP belongs to the class of NP-hard problems [8]. Several heuristic methods have been developed for this problem such as genetic algorithm [9], quasi-human based heuristic [10], tabu search [8], GRASP [11], sequence-pair representation [12], simulated annealing [13], particle swarm optimization [14], greedy [15], ant colony [16], etc. For this problem, a few mathematical formulation models are introduced in the literature. The traditional approach generates all of the possible packing patterns and formulates an individual model. In the 60s, the cyclical approach of Gilmore and Gomory [6,7], namely column generation, was introduced in which a 1-Dimensional Pattern (1DP), based on the dual prices, is generated within each round of the algorithm loop. Beasley [17] introduced a simple formulation in Euclidean coordinates, such that the bottom-left corner coordinate defines the piece position. He used four constraints for each piece to prevent pieces' overlapping. Chen et al. [18] relaxed Beasley formulation by using four binary variables for packing a composition of two pieces. Hadjiconstantinou and Christofides [19] introduced a zero-one integer programming formulations of the problem. Even for small-problem instances, they have to consider very large zero-one programs, because the number of variables depends on the size of the pallet that is to be packed. Formulation of Tsai et al. [20], furthermore, was the same as that of Chen et al. [18], except that it includes two binary variables instead of four. The table of the two variables includes four compositions as each of them refers to one of the basic orientations: left, right, top, and bottom. Research work of Tsai et al. [21] was a kind of strip formulation based on the linear equations. They divided the pallet length into various strips, i.e. strips lengths are equal to the pallet width and strips widths are equal to unit-scale. In the first step, the upper limit for wasted space is calculated; therefore, applying the calculated upper limit reduces the complexity. In the second step, the strips are generated, and in the final step, an individual model is formulated and solved. Caprara and Monaci [22] introduced some numeration algorithms to search for the best pattern. Fekete and Schepers [23] and Fekete et al. [24] developed different tree search algorithms for solving the problem. They combined the use of data structure for characterizing feasible packing with new classes of lower bounds and some heuristics.

The structure of this paper is as follows. The column generation approach [6,7] is briefly explained in Section 2, and then, a detailed description of the proposed model is given in Section 3. Analytical results are presented in Section 4 and finally, conclusions of this research study are given in Section 5.

2. Column generation approach

Assume a lot of bars with identical lengths and also a set of pieces with different sizes and various demands. The question is: How many bars should be cut for each 1DP to meet the piece's demands and minimize the total cut bars? Figure 1 shows the original column generation approach to answer the above question. In this figure, Model (1)-(3) selects the best set of 1DPs, and Model (4)-(6) generates a new 1DP, in each round, based on the shadow prices derived from Model (1)-(3). So, in Model (1)-(3), a new column, indicating a new 1DP, is added to the revised simplex tablet, in each round, in order to optimize the cutting process:

$$\min \quad \sum_{k=1}^{K} T_k \times X_k + \sum_{p=1}^{P} l_p \times Y_p, \tag{1}$$

$$\text{St.} \quad \sum_{k=1}^{K} a_{pk} \times X_k - Y_p = H_p \quad p = 1, ..., P, \tag{2}$$

$$\forall\ X_k, \quad Y_p \geq 0, \tag{3}$$

where:

T_k	Scrap length in 1DP k $(k = 1, \cdots, K)$;
X_k	The numbers of bars to be cut as 1DP k $(k = 1, \cdots, K)$;
l_p	Length of piece p $(p = 1, \cdots, P)$;
Y_p	The additional number of cut piece p $(p = 1, \cdots, P)$;

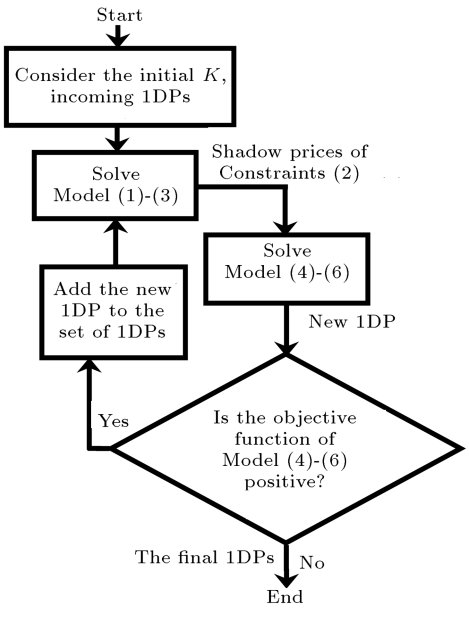

Figure 1. The original column generation approach.

a_{pk} The number of piece p included in 1DP k ($p = 1, \cdots, P$ and $k = 1, \cdots, K$);

H_p Demand of piece p ($p = 1, \cdots, P$),

and:

$$\text{Max} \quad \sum_{p=1}^{P} \pi_p \times t_p, \tag{4}$$

$$\text{St.} \quad \sum_{p=1}^{P} l_p \times t_p \leq L, \tag{5}$$

$$\forall\ t_p \in (0,1), \tag{6}$$

where:

L Length of the bars;

l_p Length of piece p ($p = 1, \cdots, P$);

π_p Shadow prices concerning Constraints (2) ($p = 1, \cdots, P$);

t_p Is a binary variable and is equal to 1 if 1DP contains piece p; otherwise, it is equal to 0 ($p = 1, \cdots, P$).

3. The proposed model

Consider a single rectangular pallet of dimensions (L, W) and set P including m ($i = 1, \cdots, m$) rectangular pieces of dimensions (l_i, w_i) with predetermined values v_i. The objective is to pack the pallet with a subset of pieces (i.e., the unconstrained problem) without rotation in such a way as to maximize the sum of the values of the packed pieces. Each piece to be packed has a fixed orientation (it cannot be rotated, i.e. length of the packed pieces should be parallel to length of the pallet). The considered 2-Dimentional Pattern (2DP) should be orthogonal (i.e., edges of pieces must be parallel to the edges of the pallet).

The paper proposes a model to generate a 2DP based on a new idea. The new definition of piece location in the pallet is illustrated in Figure 2. In this figure, the vertical dotty lines split 2DP into a set of inclusive 1DPs. The 1DPs have a length equal to the pallet width, but with different widths. Each piece is split into some parts in strips. For instance, in Figure 2, piece 1 is split into four parts in 1DPs 1, 2, 3, and 4. 1DP 4 also contains parts of pieces 1, 2, and 6. There are two kinds of scraps: the scrap in 1DPs, namely inner scrap (e.g., the hatched area in 1DP 2), and the scrap at the end side of the pallet, namely marginal scrap (e.g., the marginal scrap with width S).

Concerning the above definitions, Model (7)-(13) is the preliminary mathematical formulation. The objective function, Term (7), is the sum of values of the packed pieces. According to Terms (8), the sum of

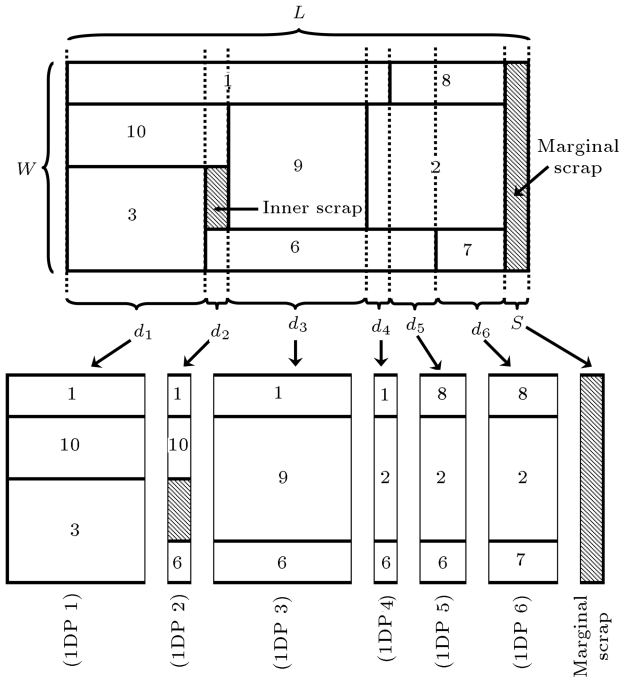

Figure 2. Splitting a 2DP into the inclusive 1DPs.

widths of 1DPs, including a given piece, must be equal to the given piece length. Note that the upper bound for the number of strips is the number of pieces ($n = m$). Terms (9) control the upper bound of the stacked sizes in 1DPs. Constraint (10) states that the pallet length is the upper bound for the sum of 1DPs widths:

$$\text{Max} \quad \sum_{i \in P} \alpha_i \times v_i, \tag{7}$$

$$\text{St.} \quad \sum_{j \in S} \beta_{ij} \times d_j - l_i \times \alpha = 0 \qquad i \in P, \tag{8}$$

$$\sum_{i \in P} w_i \times \beta_{ij} \leq W \qquad j \in S, \tag{9}$$

$$\sum_{j \in S} d_j \leq L, \tag{10}$$

$$\forall\ \alpha_i \in (0,1), \tag{11}$$

$$\forall\ \beta_{ij} \in (0,1), \tag{12}$$

$$\forall\ d_j \geq 0, \tag{13}$$

where:

$L\&W$ Length and width of the pallet;

P Set of pieces;

$l_i, w_i \& v_i$ Length, width, and value of piece i ($i = 1, \cdots, m$);

$\alpha_i =$ A binary variable and is equal to 1 if piece i is packed into the pallet; otherwise, it is equal to 0 ($i = 1, \cdots, m$);

S Set of strips;

β_{ij} A binary variable and is equal to 1 if 1DP j contains piece i, otherwise it is equal to 0 ($i = 1, \cdots, m$ and $j = 1, \cdots, n$);

d_j Width of 1DP j ($j = 1, \cdots, n$).

It should be noted that 2DP generated by Model (7)-(13) may contain interrupted pieces. This 2DP, under a relaxation named "contiguity", is a solution to the non-preemptive cumulative-resource schedule problem [25]. However, it is not a feasible point for SKP.

Some interruptions occur because all 1DPs, including a given piece, are not aligned side by side. For example, in Figure 3(a), piece 2 is interrupted, in which the interruption is solved by arranging the 1DPs as "1DP 2 - 1DP 1 - 1DP 3". Some interruptions also occur due to the inappropriate arrangement of parts in IDPs. For example, in Figure 3(b), piece 2 is interrupted, so that the interruption is solved by appropriately arranging the parts in 1DP 3.

But, some interruptions cannot be resolved. For example, in Figure 3(c), piece 2 is interrupted, but any arrangement of IDPs results in the interruption of one of the packed pieces. In this case, there is no pattern in which all 1DPs, including a given piece, can be aligned side by side. Figure 3(d) shows another interruption. In this example, all 1DPs, including a given piece, are aligned side by side, but the intrinsic combination of the pattern elements results in interruption; so, any arrangement of parts in IDPs results in the interruption of one of the packed pieces.

3.1. The model relaxation

Clearly, Constraints (8) of Model (7)-(13) includes nonlinear relations. Therefore, we try to convert the model into linear form. Constraints (9) are taken out of Model (7)-(13) and transferred into another model. Consequently, binary variables, β_{ij}, will be considered as the model parameters. After this relaxation, Model (7)-(13) is written as Model (14)-(18):

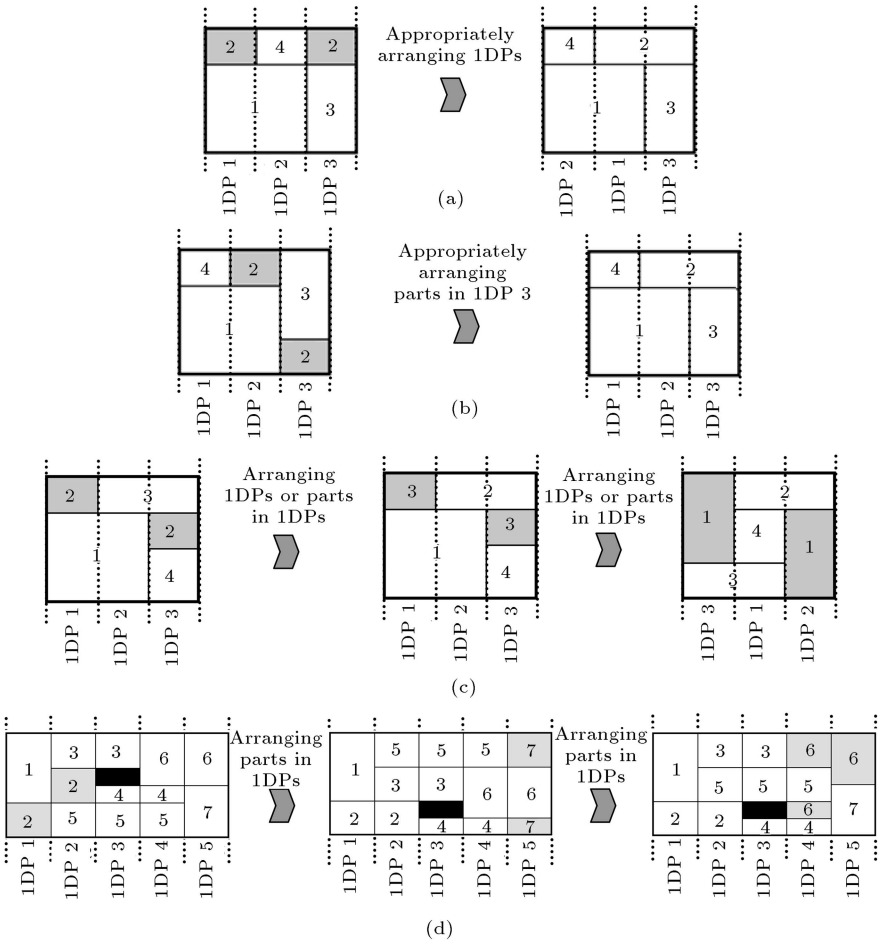

Figure 3. The piece interruption (arrow: interchange; grey: interrupted piece).

$$\text{Max} \quad \sum_{i \in P} \alpha_i \times v_i, \tag{14}$$

$$\text{St.} \quad \sum_{j \in S} \beta_{ij} \times d_j - l_i \times \alpha = 0 \qquad i \in P, \tag{15}$$

$$\sum_{j \in S} d_j \leq L, \tag{16}$$

$$\forall\ \alpha_i \in (0,1), \tag{17}$$

$$\forall\ d_j \geq 0, \tag{18}$$

where:

L	Length of the pallet;
P	Set of pieces;
$l_i \& v_i$	Length and value of piece i ($i = 1, \cdots, m$);
α_i	A binary variable and is equal to 1 if piece i is packed onto the pallet; otherwise, it is equal to 0 ($i = 1, \cdots, m$);
S	Set of strips;
β_{ij}	A binary parameter and is equal to 1 if 1DP j contains piece i; otherwise, it is equal to 0 ($i = 1, \cdots, m$ & $j = 1, \cdots, n$);
d_j	Width of 1DP j ($j = 1, \cdots, n$).

In order to solve the problem, the paper proposes an algorithm based on the column generation approach. In the proposed algorithm, in each round of the process loop, a 1DP is generated and added to Model (14)-(18). So, in each round, it should be written that: $n = n + 1$. Now, regarding Model (14)-(18), considers a given tablet of the revised simplex method with n initial 1DPs. In order to manage to obtain the optimal solution through the simplex tablet, the reduced cost of incoming d_j ($j = n + 1$ i.e. new 1DP) should be negative. In the simplex tablet, a vector $(\gamma_1, \gamma_2, \cdots, \gamma_m, \theta)$ represents the shadow prices. In this vector, γ_1 to γ_m represent the shadow prices concerning Constraints (15), and θ is the shadow price of Constraint (16). The technological coefficient of d_j is also written as $(\beta_{1j}, \beta_{2j}, \cdots, \beta_{mj}, 1)$. Thus, the reduced cost of d_j is written as in Term (19):

$$\begin{aligned} \text{Reduced cost}\ (d_j) &= (\gamma_1, \gamma_2, \cdots, \gamma_m, \theta) \\ &\quad \times (\beta_{1j}, \beta_{2j}, \cdots, \beta_{mj}, 1) \\ &= \theta + \sum_{i=1}^{m} \gamma_i \times \beta_{ij}. \end{aligned} \tag{19}$$

Note that the coefficient of d_j is zero for the objective function. In order to direct d_j as a basic variable, Term (19) should be negative, since the objective function of Model (14)-(18) is to be maximized. Finally, a model that generates a new 1DP could be considered as Model (20)-(22). It should be noted that Constraint (21) is the same as Constraint (9) that was taken out of Model (7)-(13) as the relaxation:

$$\text{Max} \quad -\theta - \sum_{i \in P} \gamma_i \times \beta_i, \tag{20}$$

$$\text{St.} \quad \sum_{i \in P} w_i \times \beta_i \leq W, \tag{21}$$

$$\forall\ \beta_i \in (0,1), \tag{22}$$

where:

W	Width of the pallet;
P	Set of pieces;
w_i	Width of the piece i ($i = 1, \cdots, m$);
γ_i	Shadow prices concerning Constraints (15) ($i = 1, \cdots, m$);
θ	Shadow price of Constraint (16);
β_i	A binary variable and is equal to 1 if 1DP contains piece i; otherwise, it is equal to 0 ($i = 1, \cdots, m$).

3.2. Dealing with the "piece interruption"

For the first action, we should try to obtain a pattern without piece interruption by appropriately arranging 1DPs and also parts in 1DPs. Otherwise, a pattern, including the intrinsic interruption, would be at reach. In such a pattern, a total value of the picked pieces is definitely equal to or better than the SKP optimal solution. Thus, the strategy to deal with these interruptions is to remove the interrupted pieces. Of course, the 1DPs, including the removed pieces, should also be eliminated. Note the fact that a combination of some 1DPs caused the current interruption; so, we have to prevent coming back to this combination. Therefore, prior to removing the interrupted piece(s), the combination of strips is put into set SC (Strips Combination). In addition, after solving Model (20)-(22), if the combination of the existing strips plus new generated strip is in set SC, the new 1DP should be rejected; besides, we should prevent its regeneration using the following constraint that is added to Model (20)-(22). In this constraint, K is the number of the parts in the strip:

$$\sum_{i \in \text{strip}} \beta_i \leq K - 1. \tag{23}$$

To remove interrupted pieces, we encounter the question "Which piece should be removed (e.g., piece 1,

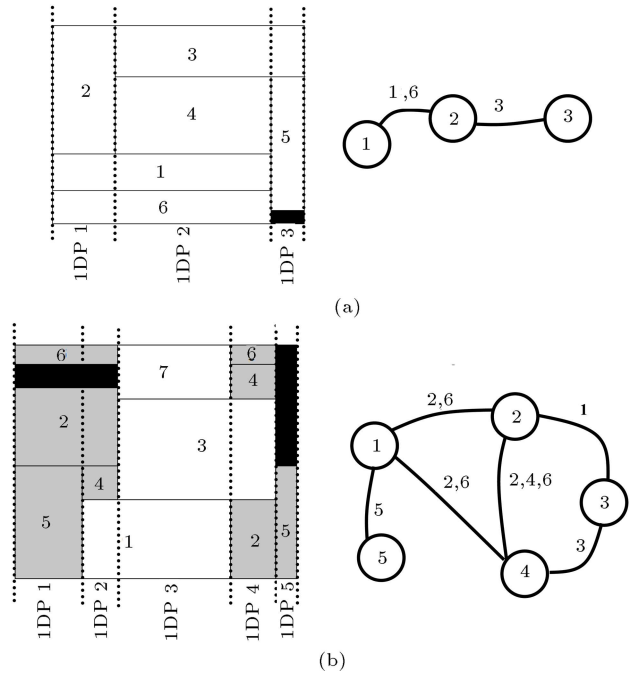

Figure 4. Modeling 2DPs as a graph (grey: interrupted piece; black: scrap).

piece 2, or piece 3 in Figure 3(c))?" To answer this question, in the interruption as in Figure 3(c), we model 2DP as a graph in which each vertex stands for a strip and the edges indicate adjacency of vertexes concern to pieces. By this definition, a pattern without the interruption is a tree in the form of serial of vertexes. For example, Figure 4(a) contains a pattern without the interruption; in the graph of this pattern, label "1, 6" on edge between vertexes 1 and 2 states that strips 1DP 1 and IDP 2 should be adjacent because of pieces 1 and 6. Figure 4(b) contains a pattern with interruption. The graph of this pattern is not a tree; so, to have a tree by the serial form of vertexes, some edges of this graph should be cut. The best cuts are the ones that minimize total values of cut edges. Note that value of an edge is the sum of the values of the concerned pieces. To this issue, the model proposes a Piece Interruption Resolving (PIR) algorithm that is derived from the Prim's algorithm [26] to solve Minimum Spanning Tree (MST) problem. A MST for a weighted graph is a spanning tree for which the sum of the weights of the edges is as small as possible. For our problem, some changes in Prim's algorithm are considered. First, a maximum spanning tree should be obtained; so, we keep choosing the edge with the biggest values. Second, to obtain a tree by the serial form of vertexes, in each of iterations, we select a new edge connected to tail-end vertexes of the serial tree. Third, in addition to the "value" criteria, the PIR algorithm considers the "label" criteria of the graph. The pseudo-code of the PIR algorithm is shown in Figure 5.

3.3. The final procedure

Figure 6 presents the final procedure of the proposed model.

Elements of the procedure are described as follows:

- At the start, some initial 1DPs are generated;
- Set S includes 1DPs for Model (14)-(18);
- Model (14)-(18) is to select the best 1DPs among a given set of 1DPs (Set S);
- Model (20)-(22) is to generate a new 1DP, based on the shadow prices of Constraints (15) and (16) derived from Model (14)-(18);

Inputs: Vertexes, List of values, List of labels on edges
Output: Removing some pieces and eliminating some strips in Model (14)-(18)
Method:
{
  Put the entire pieces included in the labels into set E
  Put the vertexes connected to edges with maximum value into set T
  Subtract the pieces included in the label of edge between the elements of T from E
  Put all the vertexes into set Q
  Let $Q = Q - T$
  Repeat until set Q is empty
  {
    Extract a vertex u from Q such that label from u to elements of T has maximum intersection to label between two elements of T; on tie, select the vertex that value from u to T is maximal; designate the success vertex from T as v (i.e. connected to u)
    Subtract the pieces included in the label of edge between u and v from E
    Subtract v from T
    Add u to T
    Subtract u from Q
  }
Remove the pieces of set E from Model (14)-(18)
Eliminate the 1DPs including the removed pieces from Model (14)-(18)
}

Figure 5. The PIR algorithm.

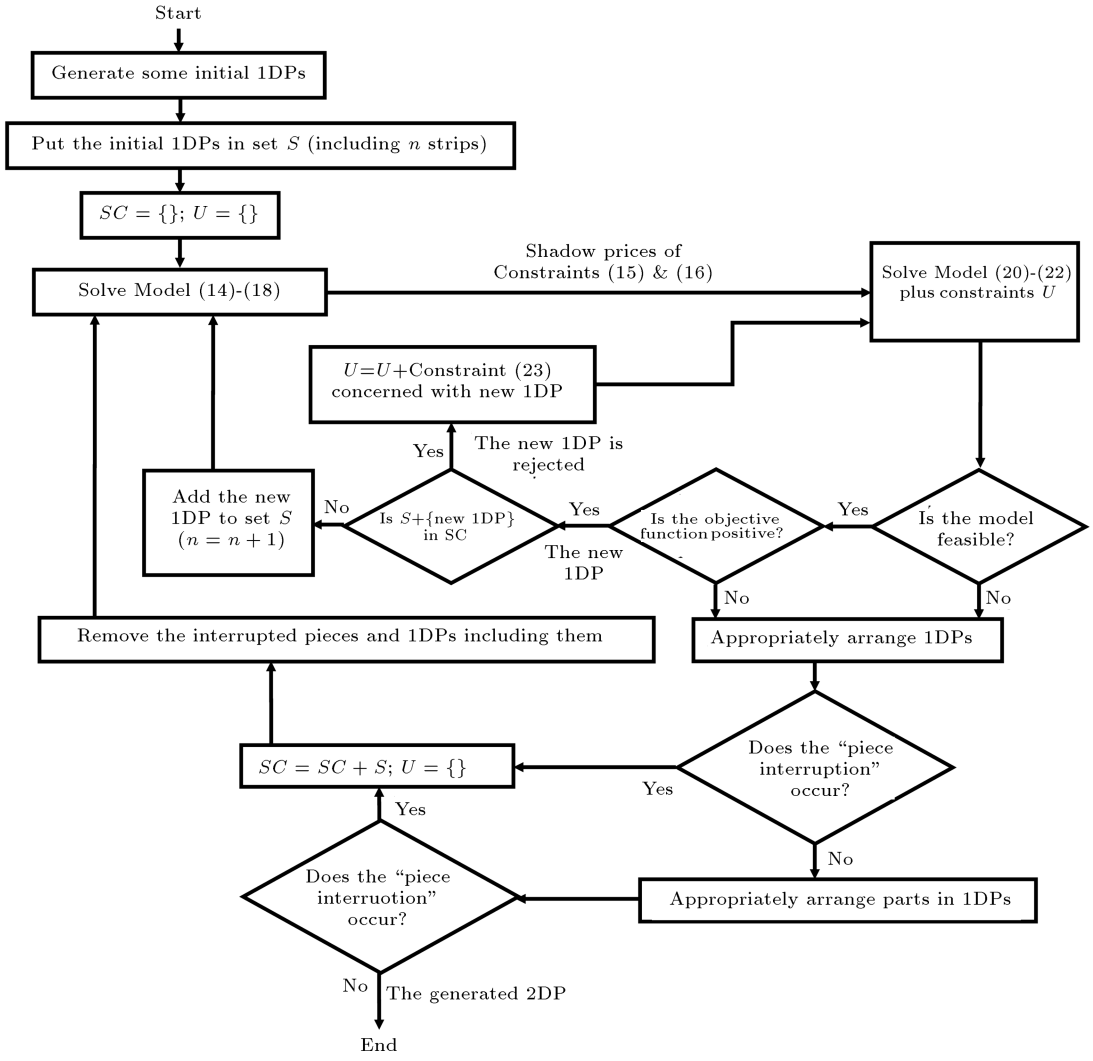

Figure 6. The final procedure of the proposed model.

- If objective function of Model (20)-(22) is not positive (a diamond in Figure 6), the new 1DP has the potential to improve the objective function of Model (14)-(18); so, it is added to set S; otherwise, this directs us to construct a 2DP for SKP by appropriately arranging 1DPs and also parts in 1DPs;

- If there is no arrangement of 1DPs in which all 1DPs, including a given piece, are aligned side by side (a diamond in Figure 6), the combination of the current 1DPs is put into set SC; then, the interrupted pieces and IDPs, including those, are removed. For this purpose, a PIR algorithm has been previously recommended;

- If there is an appropriate arrangement of 1DPs in which all 1DPs, including a given piece, are aligned side by side, but there is no arrangement of the parts in 1DPs including non-interrupted pieces (a diamond in Figure 6), the combination of the current 1DPs is put into set SC; then, the interrupted pieces and IDPs, including those, are eliminated;

- Set SC includes the combination of 1DPs containing piece interruption. A diamond is considered to prevent the occurrence of the SC elements. Thus, if the set of the current 1DPs (set S) plus new proposed 1DP is in SC, Constraint (23) is added to Model (20)-(22). Set U includes the entire inequalities of type Constraint (23) concerning the forbidden strips;

- Adding several limitations, Constraint (23) could bring about infeasibility of Model (20)-(22). A diamond is considered to control this situation.

4. Analytical results

4.1. Formulation analysis

In this section, the proposed model is compared with other mathematical programming formulations in the state of the art. So far, only a few mathematical pro-

Table 1. Structural components of the models formulations.

Model	T88	C91	T93	The proposed model
Growing the number of variables for increasing the number of the pieces	Exponential	Exponential	Linear	Linear
Growing the number of constraints for increasing the number of the pieces	Exponential	Exponential	Linear	Linear
Type of variables	Mixed (continuous and binary)	Mixed (continuous and binary)	Full integer	Continuous
Dimension measurement	Continuous input sizes	Continuous input sizes	Integer input sizes	Continuous input sizes
Dealing with the "piece interruption"	N.A.	N.A.	The interrupted pieces are removed	The issue is resolved

gramming formulations have been proposed. Table 1 shows the comparison of structural components of the proposed model and studies of Tsai et al. [20] (in brief: T88), Chen et al. [18] (C91), and Tsai et al. [21] (T93). By comparison, it is clear that our new formulation constitutes a significant progress. One indication is the fact that the relation between formula size and the number of pieces is linear, while the others are exponential. Besides, the proposed model includes a full continuous linear programming formulation, i.e. it could be solved by more efficient algorithms. The comparison is less conclusive: both have pretty formulations, except that the "piece interruption" is not resolved in the T93.

4.2. Computational analysis

The study is conducted on three groups of instances. The first is 12 small-test instances SKP1 through SKP12. In this group, the number of pieces ranges from 5 to 10. The second group is 12 instances Beasley1 through Beasley12 taken from Beasley's OR library [27]. They can be found at http://mscmga.ms.ic.ac.uk/jeb/orlib/ngcutinfo.html. The third group includes instances OKP1 through OKP5 that are considerably larger than the previous groups. They were taken from [23] and were originally designed for broader tests. Table 2 shows the characteristics of the selected instances and computational results. The first column lists the instance names; the second shows lengths and widths of pallets followed by the number of pieces. For example, in instance SKP1, the pieces are {(70, 60), (50, 40), (50, 40), (30, 30), (30, 30)}.

The computational comparison is included for five exact algorithms ([17] (in brief: B85); [19] (HC95); [23] (FS97); [22] (CM04); [24] (FSV07)) and the proposed model. In order to evaluate the performance of the proposed model, it is implemented in Delphi programming language and is tested on a PC with a Pentium IV processor (2.33 GHz) with 2.0 GB memory. In Table 2, "-" indicates that no solution is reported in the literature. Note that column "CM04" is the best running times among four algorithms A_0, A_1, A_2, A_3 as reported in [22]. In addition, the running times for instances OKP1 through OKP5 in column "B85" are reported in [11]. The same table shows the proposed model which has found an optimal solution for all instances, but in considerably high running times. Despite the running time in comparison, it is clear that the method is a really exact way to get the optimal solution. Figure 7 presents the 2DPs generated by the proposed model for SK4, SKP5, and SKP7 test instances.

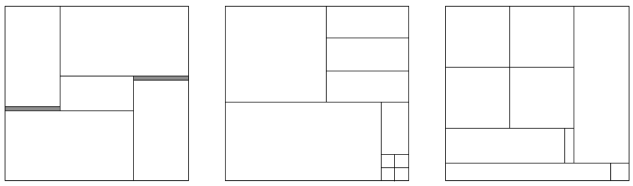

Figure 7. The 2DPs generated by the proposed model for SK4, SKP5, and SKP7.

5. Conclusion

This paper proposed a new exact model for solving the rectangular two-dimensional SKP (Single Knapsack Problem). The main contribution of this research was in offering an idea for developing the column

Table 2. Computational results.

Instance name	Pallet size; piece number	Running time (seconds); computational results											
		B85		HC95		FS97		CM04		FSV07		The model	
SKP1	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	9000
SKP2	(100,100); 6	—	—	—	—	—	—	—	—	—	—	0.01	125800
SKP3	(100,100); 7	—	—	—	—	—	—	—	—	—	—	< 0.01	9750
SKP4	(100,100); 9	—	—	—	—	—	—	—	—	—	—	0.02	24650
SKP5	(100,100); 10	—	—	—	—	—	—	—	—	—	—	0.02	13040
SKP6	(100,100); 7	—	—	—	—	—	—	—	—	—	—	< 0.01	11600
SKP7	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	9500
SKP8	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	8900
SKP9	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	12500
SKP10	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	5500
SKP11	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	7700
SKP12	(100,100); 5	—	—	—	—	—	—	—	—	—	—	< 0.01	9500
Beasley1	(10,10); 10	0.9	164	—	—	0.03	164	—	—	< 0.01	164	0.05	164
Beasley2	(10,10); 17	4	230	—	—	0.04	230	—	—	< 0.01	230	0.05	230
Beasley3	(10,10); 21	10.5	247	—	—	0.09	247	—	—	< 0.01	247	9.8	247
Beasley4	(15,10); 7	0.1	268	0.04	268	0.01	268	—	—	< 0.01	268	0.05	268
Beasley5	(15,10); 14	0.4	358	—	—	0.01	358	—	—	< 0.01	358	0.02	358
Beasley6	(15,10); 15	55.2	289	45.2	289	0.13	289	—	—	< 0.01	289	25.02	289
Beasley7	(20,20); 8	0.5	430	0.04	430	0.01	430	—	—	< 0.01	430	0.07	430
Beasley8	(20,20); 13	218.6	834	—	—	0.09	834	—	—	< 0.01	834	42.25	834
Beasley9	(20,20); 18	18.3	924	5.2	924	0.03	924	—	—	0.02	924	11.25	924
Beasley10	(30,30); 13	0.9	1452	—	—	0.02	1452	—	—	< 0.01	1452	0.85	1452
Beasley11	(30,30); 15	79.1	1688	—	—	0.13	1688	—	—	< 0.01	1688	88.32	1688
Beasley12	(30,30); 22	229	1801	> 800	1851	0.26	1865	—	—	< 0.01	1865	121.25	1865
Okp1	(100,100); 50	19.71	27486	—		11.6	27718	24.06		10.85	27718	77.53	27718
Okp2	(100,100); 30	13.19	21976	—		116.24	22502	1535.95		20.25	22502	351.25	22502
Okp3	(100,100); 30	11.46	23743	—		73.03	24019	1.91		5.98	24019	128.25	24019
Okp4	(100,100); 61	32.08	31269	—		50.09	32893	0.85		2.87	32893	20.21	32893
Okp5	(100,100); 97	83.44	26332	—		40.14	27923	488.27		11.78	27923	65.00	27923

generation approach to generate two-dimensional packing patterns. Comparison of the proposed model with the current mathematical formulation models in the literature confirms the progress in the structural aspects. Besides, the proposed model gets the optimal solution to all of the instance tests. Despite this fact, the running times of the solved instances do not show the efficiency of the view of solving speed. However, it is expected that this research leads to progress for other problem variants. In addition, it is proposed that the C&P researchers develop the column generation approach to generate three-dimensional packing patterns.

References

1. Dychoff, H. "A typology of cutting and packing problems", *European Journal of Operational Research*, **44**(2), pp. 145-159 (1990).

2. Belov, G., Kartak, V., Rohling, H. and Scheithauer, G. "One-dimensional relaxations and LP bounds for or-

thogonal", *International Transactions in Operational Research*, **16**, pp. 745-766 (2009).

3. Vanderbeck, F. "A nested decomposition approach to a three-stage two-dimensional cutting-stock problem", *Management Science*, **47**(6), pp. 864-879 (2001).
4. Pisinger, D. "Heuristics for the container-loading problem", *European Journal of Operational Research*, **141**(2), pp. 382-392 (2002).
5. Wascher, G., Haubner, H. and Schumann, H. "An improved typology of cutting and packing problems", *European Journal of Operational Research*, **183**(3), pp. 1109-1130 (2007).
6. Gilmore, P.C. and Gomory, R.E. "A linear programming approach to the cutting stock problem / part 2", *Operations Research*, **11**(6), pp. 863-888 (1963).
7. Desaulniers, G., Desrosiers, J. and Solomon, M.M., *Column Generation*, Springer Science & Business Media (2006).
8. Shigehiro, Y., Koshiyama, S. and Masuda, T. "New approach to rectangle packing problem based on stochastic tabu search", *Transactions of the Society of Instrument and Control Engineers*, **40**(7), pp. 747-754 (2004).
9. Changdar, C., Mahapatra, G.S. and Pal, R.K. "An improved genetic algorithm based approach to solve constrained knapsack problem in fuzzy environment", *Expert Systems with Applications*, **42**(4), pp. 2276-2286 (2015).
10. Wu, Y.L., Huang, W., Lau, S.C., Wong, C.K. and Young, G.H. "An effective quasi-human based heuristic for solving the rectangle-packing problem", *European Journal of Operational Research*, **141**(2), pp. 341-358 (2002).
11. Alvarez Valdes, R., Parreno, F. and Tamarit, J.M. "A grasp algorithm for constrained two-dimensional non-guillotine cutting problems", *Journal of Operational Research Society*, **56**, pp. 414-25 (2005).
12. Egeblad, J. and Pisinger, D. "Heuristic approaches for the two- and three-dimensional knapsack packing problem", *Computers and Operations Research*, **36**(4), pp. 1026-49 (2009).
13. Leung, S.C.H., Zhang, D., Zhou, C. and Wu, T. "A hybrid simulated annealing meta-heuristic algorithm for the two-dimensional knapsack packing problem", *Computers and Operations Research*, **39**, pp. 64-73 (2012).
14. Chand Bansal, J. and Deep, K. "A modified binary particle swarm optimization for knapsack problems", *Applied Mathematics and Computation*, **218**(22), pp. 11042-11061(2012).
15. Gorski, J., Paquete, L. and Pedrosa, F. "Greedy algorithms for a class of knapsack problems with binary weights", *Computers & Operations Research*, **39**(3), pp. 498-511 (2012).
16. Changdar, C., Mahapatra, G.S. and Pal, R.K. "An ant colony optimization approach for binary knapsack problem under fuzziness", *Applied Mathematics and Computation*, **223**, pp. 243-253 (2013).
17. Beasley, J.E. "An exact two-dimensional non-guillotine cutting tree search procedure", *Operations Research*, **33**(1), pp. 49-64 (1985).
18. Chen, C.S., Sarin, S. and Ram, B. "The pallet packing for un-uniform box sizes", *International Journal of Production Research*, **29**(10), pp. 1963-1968 (1991).
19. Hadjiconstantinou, E. and Christofides, N. "An exact algorithm for general, orthogonal, two-dimensional knapsack problems", *European Journal of Operational Research*, **83**, pp. 39-56 (1995).
20. Tsai, R.D., Maelstrom, E.M. and Meeks, H.D. "A two-dimensional palletizing procedure for warehouse loading operations", *IIE Transactions*, **20**(4), pp. 418-425 (1988).
21. Tsai, R.D., Maelstrom, E.M. and Kuo, W. "Three-dimensional palletization of mixed box sizes", *IIE Transactions*, **25**(4), pp. 64-75 (1993).
22. Caprara, A. and Monaci, M. "On the two-dimensional knapsack problem", *Operations Research Letters*, **32**(1), pp. 5-14 (2004).
23. Fekete, S.P. and Schepers, J. "On more-dimensional packing III: exact algorithms", *ZPR Technical Report 97-290*, http://www.zpr.uni-koeln.de/~paper.
24. Fekete, S.P., Schepers J. and Van Der Veen, J.C. "An exact algorithm for higher-dimensional orthogonal packing", *Operations Research*, **3**(55), pp. 569-87 (2007).
25. Belov, G. and Scheithauer, G. "Setup and open stacks minimization in one-dimensional stock cutting", *Informs Journal on Computing*, **19**(1), pp. 27-35 (2007).
26. O'Rourke, J., *Computational Geometry in C*, 2nd Ed., Cambridge University Press, Cambridge, UK (1994).
27. Beasley, J.E. "OR-library: distributing test problems by electronic mail", *Journal of Operational Research Society*, **41**, pp. 1069-1072 (1990).

Biography

Mohammad Ali Hatefi is an Assistant Professor at the Department of Energy Economics & Management at Petroleum University of Technology (PUT). He graduated from Iran University of Science and Technology (IUST) with honor. His areas of interest are operations research, decision analysis, project management and risk management. He is the author of several scientific publications in the area of operations research and risk management. He is currently serving as the Chief of Tehran Faculty of Petroleum, a branch of PUT.