Korean J. Chem. Eng., 34(4), 952-960 (2017)
DOI: 10.1007/s11814-016-0328-7

pISSN: 0256-1115
eISSN: 1975-7220

Minimizing loss in LCD glass manufacturing by cutting pattern optimization based on integer programming

Jun-Hyung Ryu*,†, Kyung Tae Park, and In-Beum Lee****

*Department of Nuclear and Energy System Engineering, Dongguk University, Gyeongju 38066, Korea
**Department of Chemical Engineering, POSTECH, Pohang 37673, Korea

(*Receivd 7 August 2016 • accepted 20 November 2016*)

Abstract−Loss reduction plays a critical role in improving both process efficiency and overall profitability. As processes become more complex, loss reduction becomes more challenging and systematic decision-supporting methods are thus needed. This paper illustrates such a method in the context of liquid crystal display (LCD) glass production, which involves a series of chemical processes. Loss-minimizing integer programming models are proposed to compute a cutting pattern that allocates multiple demands to LCD mother glass. Numerical examples from actual LCD glass processes are presented to illustrate the applicability of the proposed method.

Keywords: LCD Glass, Integer Programming, Cutting Pattern, Two-stage Two-dimensional Strip Packing Problem, Loss Minimization

INTRODUCTION

Loss in process operations reduces efficiency and profitability. To minimize loss, efforts should be made to minimize the consumption of resources. Process systems engineering (PSE) principles can be used to develop rigorous systematic decision-supporting frameworks by improving process efficiency for reducing the loss.

Special high-quality glass is used as a substrate for LCDs, which are an essential device in TVs, monitors, and mobile phones. The glass is produced by a series of chemical processes. Two challenges in the LCD glass manufacturing industry are the large number of products and automated manufacturing processes. Various LCD glass products of different sizes must be manufactured. Newer products are usually much bigger, while smaller, older products must still be produced as well. High capital investment is needed to automate LCD glass manufacturing processes because the glass is too large to be processed manually. To increase operation efficiency and profitability, minimizing process loss is a primary issue.

Individual LCD glass sizes are not manufactured separately. Mother LCD glass is large, rectangular, and has a fixed width. Fig. 1 illustrates a typical LCD glass cutting problem. The initial strip is cut to make multiple sizes according to a cutting pattern, which has an important impact on the process utilization.

A bad cutting pattern over-consumes resources, generates more losses, delays delivery time, and decreases profitability. The impact of the cutting pattern on the chemical processes should not be underestimated. The entire series of chemical processes might be wasted due to a bad cutting pattern. It is also inefficient to try to reuse the leftover glass after the main cutting due to the large glass sizes and their numbers. A rigorous decision-making methodology is thus needed to compute a good cutting pattern. This problem can be regarded as a two-dimensional strip packing problem (2SP) [1]. The aim is to pack all items into mother glasses of large rectangle, which are assumed to have infinite length and finite width as shown in Fig. 2. Another objective is to minimize the length of the strip. The problem has been extensively researched based on various principles.

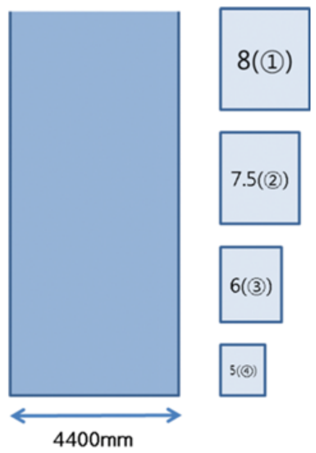

Fig. 1. Illustration of a typical LCD glass cutting problem with mother glass and individual demands.

There are other two-dimensional cutting and packing problems that are similar to 2SP, such as two-dimensional cutting stock problem (2CP) and two-dimensional bin-packing problem (2BP). 2BP is concerned with packing all small rectangles into *identical* rectangles of finite length and width while minimizing the number of identity rectangles, whereas the aim of 2CP is to cut all small rectangles from a rectangle of finite length and width to minimize the number of the rectangles.

In 2SP, a cut is made from an edge to the opposite edge, which is often called a *guillotine cut* and is shown in Fig. 3. 2SP is further

†To whom correspondence should be addressed.
E-mail: jhryu@dongguk.ac.kr

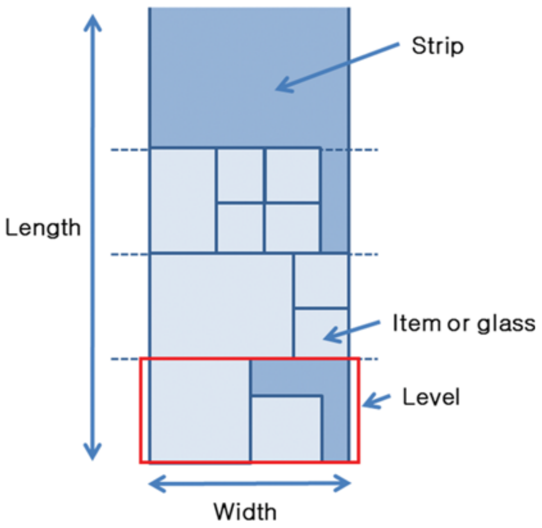

Fig. 2. Schematic of LCD glass manufacturing problem as a two-dimensional strip-packing problem.

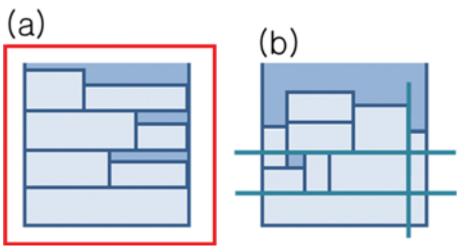

Fig. 3. Schematic of glass cutting using (a) guillotine cut, and (b) non-guillotine cut.

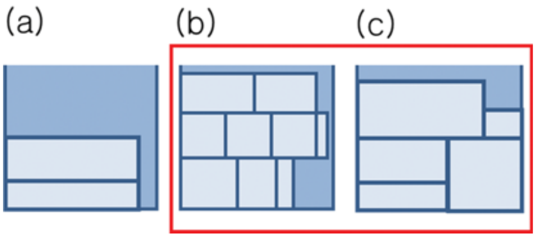

Fig. 4. Schematic of (a) one-stage, (b) two-stage, and (c) three-stage 2SP.

categorized according to the number of guillotine cuts into *two- or three-stage* 2SP, as shown in Fig. 4. An additional cut to remove waste is called non-exact cut.

In the open literature, there have been some works. Lodi et al. presented an overview of methods for handling 2SP and 2BP [2] and also discussed methodologies for computing lower bounds for these problems. Martello et al. proposed a heuristic algorithm for 2SP [3]. Lodi et al. presented a *level model* that packs small rectangles into a large rectangle by levels for two-stage 2SP and two-stage 2BP [4]. The bounds of the solution were computed. Puchinger and Raidl extended the level model to the case of three-stage 2BP [5].

Cintra et al. presented algorithms using dynamic programming for 2SP and 2BP [6]. Alvarez-Valdes et al. proposed a branch and bound algorithm for 2SP [7]. Gilmore and Gomory proposed an integer programming method using column generations for 2CP [8]. Correia et al. proposed a discrete formulation for the 2CP with variable size [9]. Macedo et al. presented an arc-flow model for 2CP [10]. Silva et al. proposed an integer programming model for two- and three-stage 2CP using a cutting concept that defines activities to generate items and new plates from old plates through guillotine cuts [11]. They extended the integer programming model for the one-dimensional cutting stock problem based on work by Dyckhoff et al. to 2CP [12].

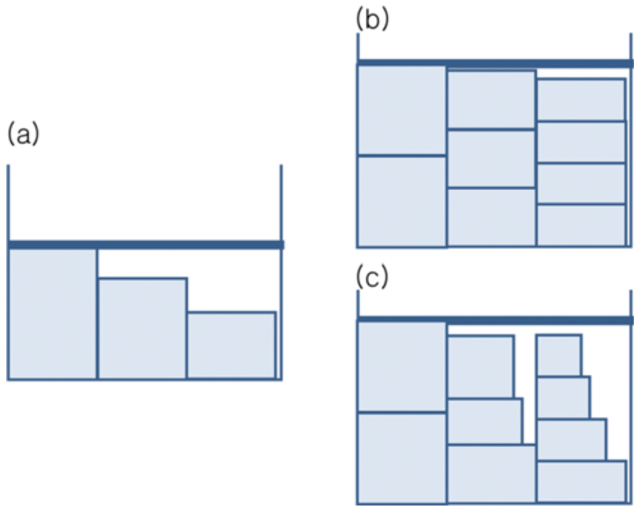

Fig. 5. Schematic of (a) non-exact two-stage cutting, (b) exact three-stage cutting, (c) non-exact three-stage cutting.

Based on the above categorization, it is recognized that the LCD glass cutting problem is posed as non-exact two-stage, exact three-stage, or non-exact three-stage problems, as shown in Fig. 5. Few works address LCD glass cutting in the PSE literature. Park et al. discussed the importance of the LCD glass-cutting problem for the first time [13]. They proposed a series of heuristics to compute a solution of the problem [13,14]. Later, the problem was formulated into a mixed integer nonlinear programming (MINLP) problem [15]. They computed the solution by transforming the problem into equivalent MILP problems. It is worthwhile to use a different modeling perspective. Further systematic efforts should be made to compute the cutting pattern with less dependence on heuristics for fully automated LCD glass manufacturing processes.

We transformed the problem into an integer programming problem. Specifically, the idea of a cut is used in the LCD mother glass cutting problem in the context of two- and three-stage 2SP in this paper. To compare the performance of the proposed model, similar level problems are considered [4]. Because the level model for three-stage 2SP was not studied before, the level model of the three-stage 2BP proposed by Puchinger and Raidl is addressed [5].

The rest of this paper is organized as follows. The operation of LCD mother glass cutting is transformed into two- and three-stage 2SP in Section 2. In the next section, the two-stage and three-stage 2SP are formulated as integer programming models. In Section 4, numerical case studies and actual industrial cases are presented, followed by some remarks.

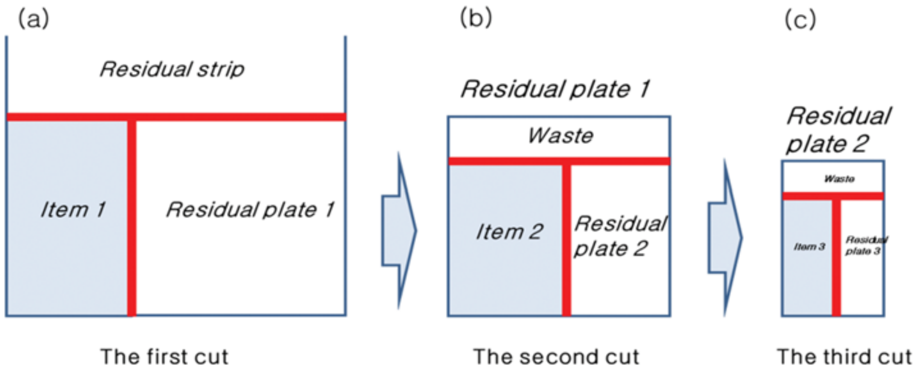

Fig. 6. Concept of a cut: (a) first cut, (b) second cut, (c) third cut.

INTEGER PROGRAMMING MODEL FOR TWO-STAGE 2SP

1. The Cut

Items can be separated by a guillotine cut [12]. As shown in Fig. 6, item type 1 in the first cut and residual plate 1 are obtained from the initial strip. The remaining area above item type 1 is called a residual strip and will be used in the next cut. Residual plate 1 is used as a new mother glass in the second cut, and item type 2 and residual plate 2 are obtained. The area above item type 2 is not used in the next cut and becomes waste. In the third cut, item type 3 and residual plate 3 are obtained from residual plate 2. The area above item type 2 also becomes waste.

2. Proposed Model for Two-stage 2SP

To transform the two-stage 2SP into a mathematical model, decision variables associated with a cut for assigning items are employed. One decision variable corresponds to one cut, which links an item and a residual plate. The objective of the model is to minimize the length of the strip to satisfy demands.

The types of items are indexed by i (i=1, 2, ⋯, I). The types of mother plates are indexed by j (j=0, 1, ⋯, J), and the types of residual plates are indexed by k (k=1, 2, ⋯, K). The lengths of the item type i is denoted as h_i, and the demand is d_i. M indicates a large number. Parameter a_{ijk} is 1 when obtaining item type i and the residual plate k from the mother plate j; otherwise, a_{ijk} is 0. A decision variable n_{ij} is defined as the number of cuts to obtain item type i from mother plate j.

The proposed model is presented as follows:

At first, objective function (1) is used to minimize the length of the initial strip [1]:

$$\text{minimize} \sum_{i=1}^{I} h_i n_{i0} \tag{1}$$

This function is subject to the following constraints. Constraint (2) indicates that the demands of each item type should be met in the strip:

$$\sum_{j=0}^{J} n_{ij} \geq d_i,\ i=1, \ldots, I, \tag{2}$$

The number of mother glass used should be at least equal to the number of the used residual plates:

$$\sum_{j=0}^{J}\sum_{i=1}^{I} a_{ijk} n_{ij} \geq \sum_{i=1}^{I} n_{ik},\ k=1, \ldots, K, \tag{3}$$

In constraint (4), the cut occurs when a_{ijk} is equal to 1.

$$n_{ij} \leq \sum_{k=1}^{K} (M a_{ijk}),\ i=1, \ldots, I,\ j=0, \ldots, J \tag{4}$$

$$n_{ij} \geq 0 \text{ and integer},\ i=1, \ldots, I,\ j=0, \ldots, J$$

3. Solution Algorithm for Two-stage 2SP

The solution algorithm as summarized in Fig. 7 allows us to compute the item types and residual plate types from cuts in the above model. P denotes a set of residual plate types, and I is a set of item types. A residual plate type is selected as the mother plate type from P. Possible cuts are searched to generate residual plate types for all item types in I. Parameter $a_{ijk\ or\ l}$ becomes 1 when a cut is made. A new residual plate type is added to P. The algorithm stops when all residual plate types are assigned.

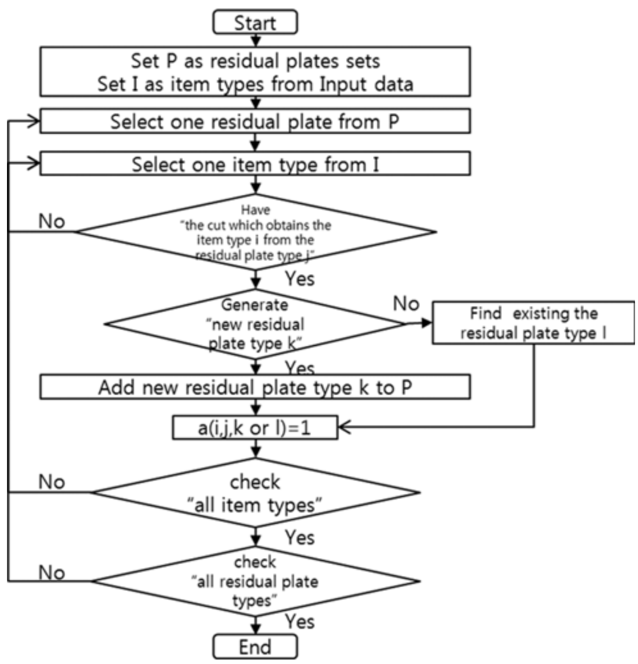

Fig. 7. Flow chart of the proposed enumeration algorithm for the two-stage 2SP.

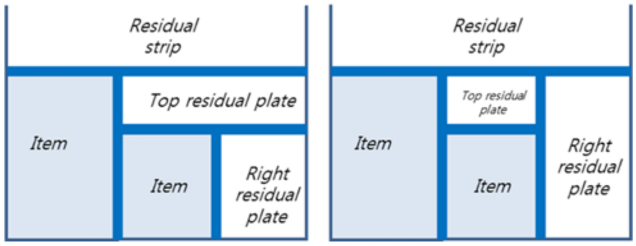

Fig. 8. Illustration of two types of residual plate in three stage 2SP.

THE INTEGER PROGRAMMING MODEL FOR THREE-STAGE 2SP

1. Two Challenging Issues in Three-stage 2SP

Two challenging issues should be addressed for three stage 2SP.

At first, two different types of residual plate are generated in the three-stage 2SP, as shown in Fig. 8. When a cut is made, one residual plate type is generated in the two-stage 2SP. On the other hand, a cut yields two residual plate types with one item in the three-stage 2SP. The two types of the residual plate are called the right residual plate and the top residual plate.

Secondly, stacks of items are another issue. When items are packed into the strip, a level appears, which represents the space between two adjacent horizontal guillotine cuts in the strip and consists of several items. If stacking items are not allowed in the level, then it becomes two-stage 2SP. Otherwise, the problem is three-stage 2SP. It is also possible to have infinite stacks of items in the level. In that case, it is difficult to compute the solution due to the presence of too many possible alternatives. To avoid the occurrence of such infinite item stacks, a maximum length of the level is imposed in the solution algorithm.

In the three-stage revised level model, the types of small rectangles are indexed by i' ($i'=1, 2, \cdots, n$), and the types of stacks which pile items are indexed by j' ($j'=1, 2, \cdots, n$). The types of levels which contain stacks and are packed into the large rectangle are indexed by k' ($k'=1, 2, \cdots, n$). The width of the large rectangle is denoted as W, and H is a large number. The length of the small rectangle types i' is denoted as $h_{i'}$, and the width of the small rectangle types i' is $w_{i'}$. The binary variable $\alpha_{j'i'}$ is 1 if item i' is packed into stack j' and $\alpha_{j'i'}$ is 0 otherwise. Similarly, $\beta_{k'j'}$ is 1 if stack j' is packed in level k', and $\delta_{k'i'}$ is 1 if item i' is packed into the first strip in level k'.

Finally, the level model revised for the three stage 2SP is constructed as follows. The objective function of the model minimizes the length of a strip.

$$\text{minimize} \sum_{k'=1}^{n}\sum_{i'=k'}^{n} h_{i'}\delta_{k'i'} \tag{5}$$

The function is subject to the following constraints. Each small rectangle is packed into stacks only once.

$$\sum_{j'=1}^{i'} \alpha_{j'i'}=1,\ \forall i'=1, \ldots, n, \tag{6}$$

Small rectangles are packed into the stack according to constraint (7).

$$\sum_{i'=j'+1}^{n} \alpha_{j'i'} \le (n-j')\alpha_{j'j'},\ \forall j'=1, \ldots, n-1, \tag{7}$$

Small rectangles that are packed into stacks have the same width. The sum of lengths of small rectangles in the same stack should not exceed H according to constraint (8).

$$\alpha_{j'i'}=0,\ \forall j'=1, \ldots, n-1,\ \forall i'>j' \,|\, w_{i'} \neq w_{j'} \vee h_{i'}+h_{j'}>H, \tag{8}$$

Constraint (9) indicates that each stack is packed into a level only once.

$$\sum_{k'=1}^{j'} \beta_{k'j'}=\alpha_{j'j'},\ \forall j'=1, \ldots, n, \tag{9}$$

The total length of all small rectangles in a stack should not exceed the length of the level, which is the total length of all small rectangles packed into the first stack in the level according to constraints (10) and (11).

$$\sum_{i'=j'}^{n} h_{i'}\alpha_{j'i'} \le \sum_{i'=k'}^{n} h_{i'}\alpha_{k'i'}+(H+1)(1-\beta_{k'j'}), \tag{10}$$
$$\forall k'=2, \ldots, n,\ \forall j'=1, \ldots, k'-1,$$

$$\sum_{i'=j'}^{n} h_{i'}\alpha_{j'i'} \le \sum_{i'=k'}^{n} h_{i'}\alpha_{k'i'}+H(1-\beta_{k'j'}), \tag{11}$$
$$\forall k'=1, \ldots, n-1,\ \forall j'=k'+1, \ldots, n,$$

Constraint (12) ensures that the width of the level is not bigger than the initial width of the strip.

$$\sum_{j'=k'}^{n} w_{j'}\beta_{k'j'} \le W\beta_{k'k'},\ \forall k'=1, \ldots, n, \tag{12}$$

Constraint (13) indicates that all small rectangles should be packed into the first stack in the level.

$$\alpha_{k'i'}+\beta_{k'k'}-1 \le \delta_{k'i'} \le (\alpha_{k'i'}+\beta_{k'k'})/2,\ \forall i'=k', \ldots, n,\ \forall k'=1, \ldots, n$$
$$\alpha_{j'i'},\ \beta_{k'j'} \text{ and } \delta_{k'i'}\ \ \forall i'=1, \ldots, n,\ \forall j'=1, \ldots, n,\ \forall k'=1, \ldots, n, \tag{13}$$

2. Modified Cut

To address the issue of multiple types of residual plate, the idea of a cut is modified. The modified cut allows us to obtain several items with a new residual plate from one cut through several guillotine cuts, as shown in Fig. 9. A group of possible stacks within the maximum length are introduced. An enumeration algorithm generates item groups for item types and obtains the parameter a_{gjk}

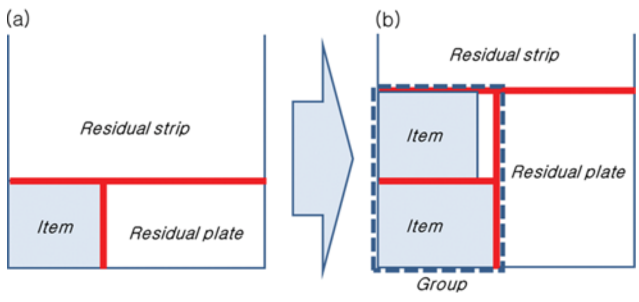

Fig. 9. (a) The concept of a cut: an item from one cut, (b) the concept of a modified cut: several items can be made from one cut.

as item groups and residual plates from cuts. The right residual plate is mainly considered in this paper because item groups consider all possible stacks of items.

3. Proposed Model for the Three-stage 2SP

The three-stage 2SP is mathematically formulated by incorporating the decision variables for the modified cut. One decision variable corresponds to one modified cut, which results in several items and one residual plate. An integer programming model for the three-stage 2SP can be formulated. The types of item groups are indexed by g (g=1, 2, ⋯, G). The length of the item groups g is denoted as h_g, and the types of item i (i=1, 2, ⋯, I) belonging to item group g are denoted as C_{gi}. Parameter $a_{gjk}=1$ when obtaining the item group g and the residual plate k (k=1, 2, ⋯, K) from the mother plate j (j=0, 1, ⋯, J), or $a_{gjk}=0$ otherwise. A decision variable n_{gj} is defined as the number of cuts to obtain the item group g from the mother plate j.

The proposed model is presented as follows:
The objective function (14) minimizes the length of the initial strip.

$$\text{minimize} \sum_{g=1}^{G} h_g n_{g0} \tag{14}$$

Constraints (15), (16), and (17) play the same roles as constraints (2), (3), and (4) in the two-stage 2SP, respectively.

$$\sum_{g=1}^{G}\sum_{j=0}^{J} C_{gi} n_{gj} \geq d_i,\ i=1, \ldots, I, \tag{15}$$

$$\sum_{j=0}^{J}\sum_{g=1}^{G} a_{gjk} n_{gj} \geq \sum_{g=1}^{G} n_{gk},\ k=1, \ldots, K, \tag{16}$$

$$n_{gj} \leq \sum_{k=1}^{K} (M a_{gjk}),\ g=1, \ldots, G,\ j=0, \ldots, J \tag{17}$$

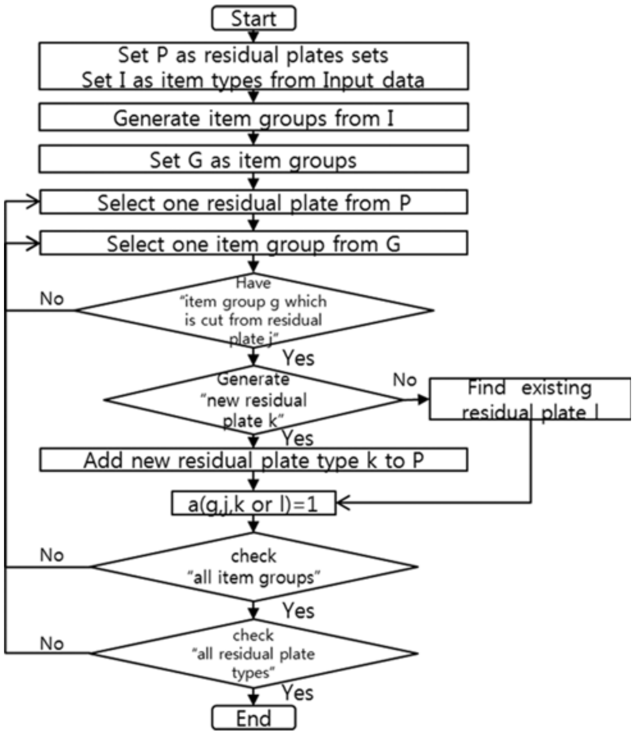

Fig. 10. Flow chart of the enumeration algorithm for three-stage 2SP.

$n_{gj} \geq 0$ and integer, g=1, …, G, j=0, …, J

4. Solution Algorithm for Three-stage 2SP

The solution algorithm for the three-stage 2SP allows us to generate item groups for item types and obtain the parameter a_{gjk} as item groups and residual plates from cuts. For non-exact three-stage 2SP, an item group is a combination of item types. For example, consider three item types (1, 2, 3). Without the maximum length constraint, the enumeration algorithm generates seven item groups: (1), (2), (3), (1, 2), (1, 3), (2,3), and (1, 2, 3). For the exact three-stage 2SP, item groups correspond to the combination of item types of the same width. By making a constraint limiting the maximum length, the number of item groups can be reduced. A flow chart of the enumeration algorithm is shown in Fig. 10.

In Fig. 10, set P denotes residual plate types, and I is the item type. Generated item groups from item type I are added, and G is set as an item group. One residual plate type is selected as the mother plate type from P. Possible cuts to generate residual plate types are found for all item groups G, and $a_{ijk\ or\ l}=1$. New residual plate types are added to P. The algorithm ends if all residual plates are checked.

COMPUTATIONAL RESULTS

This section illustrates the applicability of the proposed models and algorithms by numerical examples. We used Visual Basic for Applications (VBA) in Excel and Cplex12.2.0.2/Gams23.6.3 as computation tools. All computational results were obtained on an Intel Core (TM) i5 CPU 2.68 GHZ with 4 GB of memory. VBA calls the enumeration algorithm for obtaining item types (or item groups) and residual plate types from cuts and then calls the Cplex/Gams for solving the proposed model, as shown in Fig. 11. To reduce the occurrence of too many residual plate types, unused residual plate types are combined into one residual plate type.

1. Randomly Generated Instances

Numerical case studies with randomly generated instances are tested at first. Consider an LCD glass process. The width of the manufactured LCD glass strip was generated by a random uniform distribution from 3,000 to 6,000 mm with infinite strip length. The width and length of items were also generated by a random uniform distribution from 1,000 mm to 3,000 mm. Four kinds of item types were used. For each item type, demand is generated by a

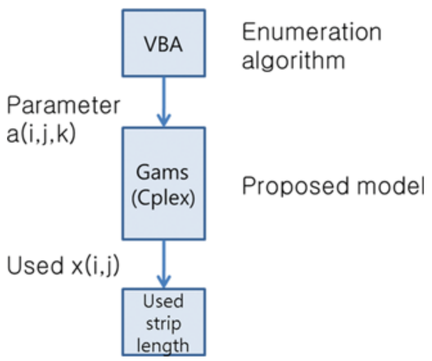

Fig. 11. Algorithm implementation using VBA and Cplex/Gams.

Table 1. Results for the non-exact two-stage 2SP

Item types	Demands	Two-stage non-exact cutting									
		C2stage					L2stage				
		Z	Solved	Time	LPR	Node	Z	Solved	Time	LPR	Node
5	1-5	13,213	10	0.0276	12,545	8	13,213	10	0.1151	10,832	414
	5-10	34,088	10	0.0169	33,452	12	34,088	7	98.237	28,195	194,177
	10-50	149,503	10	0.0245	149,088	62	149,522	4	180.83	125,710	82,272
10	1-5	26,994	10	0.04	26,561	24	26,994	10	3.0228	22,682	6,159
	5-10	74,929	10	0.0853	74,467	369	74,930	5	153.12	64,575	92,434
	10-50	281,973	10	0.1856	281,608	150	282,332	2	240.34946	238,568	30,405
15	1-5	39,126	10	0.3681	38,407	1,240	39,126	7	128.78091	35,031	174,940
	5-10	105,570	10	0.107	105,300	123	105,814	2	240.82312	94,219	106,530
	10-50	463,488	10	0.0919	463,046	144	465,863	2	241.75269	401,300	3,774
20	1-5	59,467	10	0.0995	59,186	24	59,467	10	36.479202	51,549	36,988
	5-10	131,399	10	0.378	131,020	470	131,543	1	275.1651	118,809	78,021
	10-50	628,187	10	0.1511	627,867	222	629,329	2	244.42769	536,069	6,074

Table 2. Results for the exact and non-exact three-stage 2SP

Item types	Demands	Three-stage exact cutting										Three-stage non-exact cutting				
		EC3stage					L3stage					C3stage				
		Z	Solved	Time	LPR	Node	Z	Solved	Time	LPR	Node	Z	Solved	Time	LPR	Node
5	1-5	13,071	10	0.0326	12,321	27	13,071	10	0.6417	6,834	1,797	13,034	10	0.0321	12,317	16
	5-10	33,564	10	0.0323	33,079	14	33,564	5	167.28	17,862	210,657	33,538	10	0.0588	33,079	9
	10-50	147,604	10	0.0463	147,239	168	150,681	1	273.94	61,631	25,419	147,604	10	0.0541	147,239	225
10	1-5	26,614	10	0.0899	26,148	46	26,614	9	41.614	15,019	26,883	26,489	10	0.3466	26,096	96
	5-10	73,695	10	0.0852	73,189	141	73,799	0	300.15283	38,893	126,865	73,695	10	0.0996	73,188	182
	10-50	277,832	10	0.1025	277,534	178	386,918	0	304.64331	89,360	1,228	277,413	10	0.1713	277,111	225
15	1-5	38,596	10	0.1509	38,027	36	38,596	10	68.380905	22,828	60,822	38,429	10	0.5354	37,888	702
	5-10	103,448	10	0.4683	103,078	318	104,755	0	300.7558	51,697	20,981	102,819	10	0.6073	102,582	220
	10-50	455,541	10	0.195	455,084	379	a	a	a	a	a	455,161	10	0.3217	454,725	327
20	1-5	58,869	10	0.2061	58,489	150	58,869	5	154.7935	34,443	69,816	58,627	10	0.4012	58,333	149
	5-10	128,953	10	0.8386	128,401	917	130,926	0	301.06049	72,028	18,002	128,734	10	2.5027998	128,298	1,472
	10-50	617,398	10	0.4184	617,098	289	a	a	a	a	a	612,276	10	2.6480998	612,032	2,687

[a]The model is not constructed because of memory limits

random uniform distribution from 1 to 5, from 5 to 10, and from 10 to 50. The maximum length is 3,000 mm. Ten iterations were executed for all cases, and the average values are presented. The time limit for executing Cplex/Gams was 300 CPU seconds.

We considered three types of problems: the two-stage non-exact 2SP, the three-stage exact 2SP, and the three-stage non-exact 2SP. The results for the 2SP are summarized in Table 1 and Table 2. *C2stage* denotes the result of the proposed model for the two-stage 2SP, and *L2stage* is the level model from Lodi et al. [4]. The exact and non-exact three-stage 2SP results are represented by *EC3stage* and *C3stage*, respectively, and that for the revised level model for the exact three-stage 2SP is *L3stage*. *Z* denotes the objective value indicating the length of the strip, and *Solved* denotes the number of the optimal plate. *Time* denotes the computational time. *LPR* is the LP relaxation presented by Cplex/GAMS, and *node* is the number of nodes presented by Cplex/GAMS, which uses a branch and cut algorithm.

The numbers of the optimal state in 120 instances for *L2stage* and *L3stage* are 62 and 40, respectively. The level models can solve instances with a small number of items. The number of instances solved in the level models is reduced when the number of items increases. In contrast, the proposed models can solve all instances, regardless of the number of items within five minutes. Furthermore, when the number of item types (or item groups) increases, the computational time of the proposed models increases because the enumeration algorithm generates all possible residual plate types from item types (or item groups).

2. Industrial Case Studies

Actual industrial cases were also tested using the proposed method. It would be worthwhile to briefly describe the recent LCD in terms of size: The size of the LCD mother glass increases as the size of LCDs increases. The different sizes of LCDs are denoted as "generations". Companies are eager to make more large LCDs that are more expensive in the market. The 5th-8th generations have been used in major LCD production companies. The width of the strip in the 8th generation is generally 4,400 mm. Recently, the tenth generation of the LCD mother glass was also used and the width of the strip was 5,700 mm.

2-1. Case 1

Consider an LCD glass manufacturing process. The process should be operated to meet the demand summarized in Table 3. LCD mother glass is cut from a strip with a width of 4,400 mm, as shown Fig. 1. The demand consists of four LCD mother glass types, as shown in Table 3. The 8th, 7.5th, 6th, and 5th generations of the LCD mother glass are denoted as (1), (2), (3), and (4). The maximum length of the original glass is 3,000 mm.

Generating item types and residual plate types from cuts for the two-stage 2SP is presented in Fig. 12. The number of cuts used from the proposed model are shown in red on the left-hand side of the rectangle. Residual plate type 0 denotes the strip, and the objective value is the sum of the length of items in cuts used for this type. In the computational experiments, the unused residual plate types (5, 6, 7, 9, 10, 14, 15, and 16) are unified into residual plate type 17, as shown in Fig. 13. The number of residual plate types is reduced from 17 to 10.

Table 3. LCD mother glass types and sizes for case 1

LCD mother glass types	Width (mm)	Length (mm)	Area (mm^2)	Demand
8 (①)	2,200	2,500	5,500,000	8
7.5 (②)	1,950	2,250	4,387,500	10
6 (③)	1,500	1,850	2,775,000	9
5 (④)	1,100	1,250	1,375,000	15

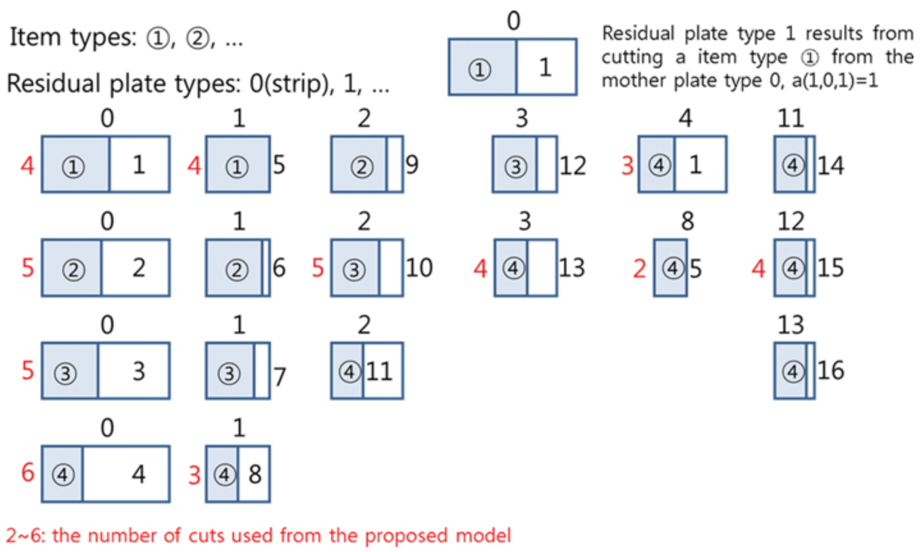

Fig. 12. Generation of item types and residual plate types by cuts in the two-stage 2SP.

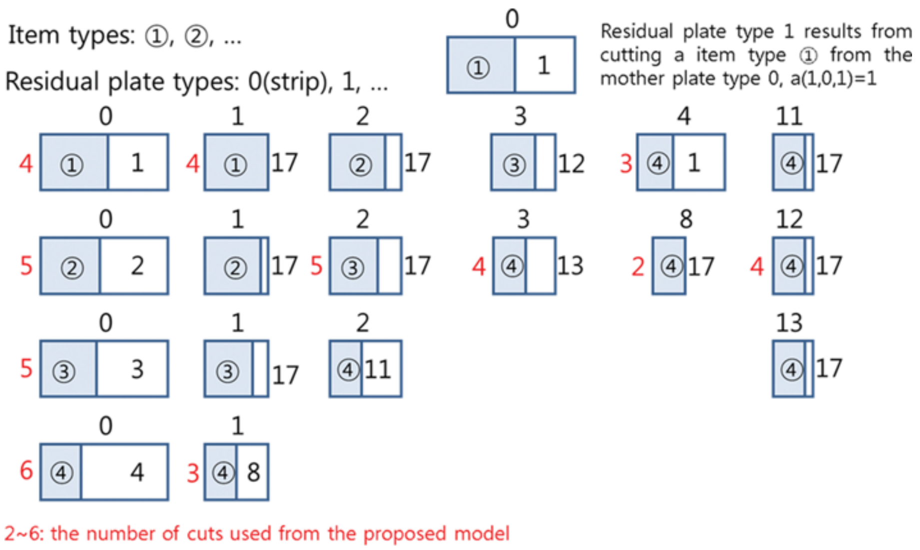

Fig. 13. Generation of item types and residual plate types reduced by cuts in the two-stage 2SP.

Table 4. Results of case 1

Model types	Z	Solved	Time	LPR	Node
L2stage	34,250	0	300	30,335	581,235
C2stage	34,250	1	0	32,856	45
L3stage	33,650	0	300	11,725	57,005
EC3stage	33,650	1	0	32,856	0
C3stage	33,650	1	0	32,856	0

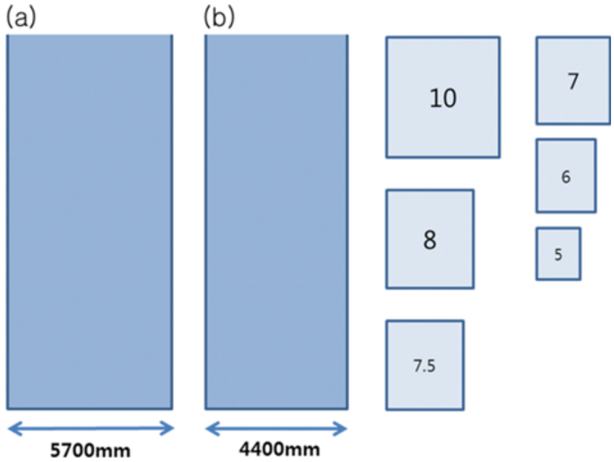

Fig. 14. The strip and LCD mother glasses for case study 2.

Table 5. LCD mother glass types and sizes for case 2

LCE mother glass types	Width (mm)	Length (mm)	Area (mm^2)	Demand
10	2,850	3,050	8,692,500	20
8	2,200	2,500	5,500,000	20
8	1,950	2,250	4,387,500	30
7	1,870	2,200	4,114,000	20
6	1,500	1,850	2,775,000	10
5	1,100	1,250	1,375,000	60

The results of the proposed models are better than those of the level models, as shown in Table 4. Although *L2stage* and *L3stage* reach the time limit of 300 seconds, the models for the two-stage 2SP have the same objective value, and the models for the three-stage 2SP also have the same objective value. The number of nodes is 0 in *EC3stage* and *C3stage* because the binary variables are deleted through a preprocessing step in the branch and cut algorithm.

2-2. Case 2

The tenth generation of LCD mother glass has widths of 5,700 mm and 4,400 mm, as shown in Fig. 14. Generally, a newer factory uses 5,700 mm. Table 5 summarizes the demand consisting of the six LCD mother glass types. The maximum length is changed from 3,000 to 5,000 mm because the length of the tenth generation exceeds 3,000 mm.

The results of the proposed models are better than those of the level models, as shown in Table 6. Although the *L2stage* reaches the time limit of 300 seconds, the models for the two-stage 2SP have the same objective value. But the objective value of *L3stage* is different from the objective of EC3stage because *L3stage* does not find a good solution within 300 seconds. The number of nodes for *L3stage* is 0, because this model executes the preprocessing steps in the branch and cut algorithm within 300 seconds.

Table 6. Results of case 2

(a)

Model types	Z	Solved	Time	LPR	Node
L2stage	123,600	0	300	116,126	87,354
C2stage	123,600	1	0	121,958	1,438
L3stage	331,250	0	301	0	0
EC3stage	121,050	1	1	120,292	13,295
C3stage	120,350	1	1	119,500	1,055

(b)

Model types	Z	Solved	Time	LPR	Node
L2stage	171,450	0	300	150,436	140,540
C2stage	171,450	1	0	170,469	25
L3stage	331,250	0	301	0	0
EC3stage	169,050	1	0	168,688	512
C3stage	164,550	1	0	164,268	502

The computational time of *EC3stage* (or C3stage) for case 2 (a) and case 2 (b) is different because of the different numbers of residual plate types. The upper bounds on residual plate types are relative to the width of the strip. If the width of the strip increases, the upper bounds on the residual plate types increase. Otherwise, the upper bounds decrease.

CONCLUDING REMARKS

There are several issues worthy of comment.

At first, we transformed the LCD glass cutting problem into a two- and three-stage two-dimensional strip packing problem. Our major contribution is that new integer programming models were proposed for computing LCD glass cutting pattern. In the literature, integer programming models were only formulated for other problem type, which is two-dimensional cutting stock problems. Another contribution is that the proposed models are based on the concept of cut, while previous similar works were based on the concept of level.

Secondly, in terms of constructing the model, a heuristic approach does not always need rigorous modeling. Constraints can be formulated in a straightforward manner by way of MINLP modeling. To obtain solutions of the resulting mixed integer nonlinear programming (MINLP) problem, nonlinear constraints should be transformed; for example, relaxed or objective functions should be also linearized. In the MINLP formulation, often a two-step approach should be made. The current approach is different from the previous works such as heuristic and MINLP approach. On the other hand, constraints should be formulated by considering all variables as integer in the integer programming model. The constraints in IP are relatively in a simpler format. But more rigorous consideration should be made in constructing the constraints. As mentioned in Silva et al. (2010), integer programming formulation allows us to use the existing advanced solution solver for computing the resulting model. This is the first approach to employ the integer programming modeling approach in the PSE community. Regarding the modeling in terms of integer programming or heuristics and MINLP formulation, it would be more appropriate to mention that they are different methodologies instead of asserting which one is better in the performance of all cutting pattern modeling and computation cases. It would be a good future research topic to compare their performances.

Thirdly, a computation methodology was proposed in this paper. When we used existing generic integer programming algorithms such as branch and bound or cutting plane methods, the computational times were very long, and sometimes no good solution was obtained within reasonable computation time. As an alternative, enumeration algorithms were developed in this paper. As was shown in numerical case studies, the proposed allowed us to compute the solution of the proposed integer programming problem efficiently.

LCD glass manufacturing companies are struggling to survive. Since most of their technologies are commercially available, one overcomes others by reducing the potential loss in the manufacturing processes. Efforts on computing the cutting pattern minimizing the loss thus may increase the competitiveness of the company. The proposed approach may be applied to other industrial cases.

ACKNOWLEDGEMENTS

The authors appreciate the help of Dr. Ho-kyung Lee in LG Chemistry for this paper. This paper was supported by the Korea Research Foundation Grant funded by the Korean Government (Basic Research Promotion Fund, NRF-2013R1A1A2A10060847) and the Dongguk University Research Fund of 2014.

REFERENCES

1. G. Wäscher, H. Hauβner and H. Schumann, *Eur. J. Oper. Res.*, **183**, 1109 (2007).
2. A. Lodi, S. Martello and M. Monaci, *Eur. J. Oper. Res.*, **141**, 241 (2002).
3. S. Martello, M. Monaci and D. Vigo, *Informs J. Comput.*, **15**, 310 (2003).
4. A. Lodi, S. Martello and D. Vigo, *J. Comb. Optim.*, **8**, 363 (2004).
5. J. Puchinger and G. R. Raidl, *Eur. J. Oper. Res.*, **183**, 1304 (2007).
6. G. F. Cintra, F. K. Miyazawa, Y. Wakabayashi and E. C. Xavier, *Eur. J. Oper. Res.*, **191**, 59 (2008).
7. R. Alvarez-Valdes, F. Parreño and J. M. Tamarit, *OR Spectrum*, **31**(2), 431 (2009).
8. P. C. Gilmore and R. E. Gomory, *Oper. Res.*, **13**, 94 (1965).
9. I. Correia, L. Gouveia and F. Saldanha-da-Gama, *Comput. Oper. Res.*, **35**, 2103 (2008).
10. R. Macedo, C. Alves and J. M. Valério de Carvalho, *Comput. Oper. Res.*, **37**, 991 (2010).
11. E. Silva, F. Alvelos and J. M. Valério de Carvalho, *Eur. J. Oper. Res.*, **205**, 699 (2010).
12. H. Dyckhoff, *Oper. Res.*, **29**, 1092 (1981).
13. K. T. Park, J. H. Ryu, H. K. Lee and I.-B. Lee, *J. Chem Eng. Jpn.*, **45**, 219 (2012).
14. K. T. Park, J. H. Ryu, H. K. Lee and I.-B. Lee, *Korean J. Chem. Eng.*, **30**, 278 (2013).
15. K. T. Park, H. Kim, S. Lee, H.-K. Lee, J. H. Ryu and I.-B. Lee, *Comput. Chem. Eng.*, **48**, 312 (2013).