

선형 재료절단 문제의 다항시간 알고리즘

이 상 운*

A Polynomial-Time Algorithm for Linear Cutting Stock Problem

Sang-Un, Lee *

요 약

일반적으로 재료절단 문제는 재료를 절단할 수 있는 패턴을 찾고 선형계획법으로 최적의 패턴 수를 찾는다. 그러나 패턴 수는 일반적으로 지수적으로 증가하기 때문에 사전에 모든 패턴을 고려하는 것은 비현실적인 것으로 알려져 있다. 본 논문은 Suliman의 실현 가능 패턴을 구하는 방법을 적용하여 사전에 패턴을 구하는 방법을 적용하였다. 또한, 실현 가능 패턴들을 대상으로 선형계획법이나 근사 알고리즘을 적용하지 않고 정확한 해를 다항시간으로 얻는 방법을 제안하였다. 제안된 방법은 실현 가능 패턴들 중 모든 요구의 1st 발생 빈도가 손실량 0에 모두 분포하는 경우와 다양한 손실량에 분산되어 분포하는 경우로 구분하여 패턴 수를 분배하는 방법을 적용하였다. 제안된 알고리즘을 2개의 데이터에 적용한 결과 모든 데이터에서 정확한 해를 구하는데 성공하였다.

▶ Keywords : 재료절단 문제, 1차원 재료절단 문제, 최초 발생 빈도, 배정

Abstract

Commonly, one seeks a particular pattern suitable for stock cutting and the number of such patterns through linear programming. However, since the number of the patterns increases exponentially, it is nearly impossible to predetermine all the existing patterns beforehand. This paper thus proposes an algorithm whereby one could accurately predetermine the number of existing patterns by applying Suliman's feasible pattern method. Additionally, this paper suggests a methodology by which one may obtain exact polynomial-time solutions for feasible patterns without applying linear programming or approximate algorithm. The suggested methodology categorizes the feasible patterns by whether the frequency of first occurrence of all the demands is distributed in 0 loss or in various losses. When applied to 2 data sets, the proposed algorithm is

•제1저자 : 이상운

•투고일 : 2013. 4. 20, 심사일 : 2013. 5. 13, 게재확정일 : 2013. 5. 20.

* 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Eng., Gangneung-Wonju National University)

found to be successful in obtaining the optimal solutions.

▶ Keywords : Cutting Stock Problem, 1D-CSP, 1st Frequency, Allocation

I. 서 론

재료절단 문제 (cutting stock problem)는 선형계획법 또는 최적화 문제로 산업계에서 널리 응용되고 있다[1-4]. 제지공장 (paper mill)에서 고정된 폭의 제지 롤에 대해 다른 고객들이 다양한 폭의 다른 개수를 원하는 경우, 절단 손실 (trim loss 또는 waste)을 최소화하는 절단계획을 수립할 필요가 있다. 제지공장에서 1%의 손실량 차이는 연간 백만 달러 이상의 비용 절감 효과를 나타내는 만큼 재료절단 문제의 최적해를 얻는 문제가 대단히 중요하게 대두되고 있다.[1] 재료절단 문제는 1차원 (1D-CSP)과 2차원 (2D-CSP) 재료절단 문제로 구분될 수 있다[1].

m 개의 j , ($j = 1, 2, \dots, m$) 고객으로부터 길이 l_j 의 주문량 q_j 를 의뢰받았을 때, 고정된 제지 롤의 폭 L 을 주문 l_j 의 실현 가능한 (feasible) 조합으로 절단하는 것을 절단 패턴 (cutting pattern, p_i)이라 한다. 패턴 사용 횟수를 x_i , 주문 j 가 패턴 p_i 에 발생한 빈도를 a_{ij} , 패턴 i 의 절단 손실량을 w_i 라 하면 재료절단 문제는 식 (1)을 만족하는 롤 수 $\sum_{i=1}^n x_i$ 을 구하는 것이다.

$$\begin{aligned} & \underset{x_i}{\text{minimize}} \sum_{i=1}^n w_i x_i \\ & \text{s.t.} \sum_{i=1}^n a_{ij} x_i \geq q_j, \forall j = 1, 2, \dots, m, x_i \geq 0 \quad (1) \end{aligned}$$

재료절단 문제는 실현 가능한 패턴을 찾고, 이 패턴들을 대상으로 절단 손실량이 최소가 되면서 요구량을 모두 만족하도록 최적의 롤 수 $\sum_{i=1}^n x_i$ 을 구하는 2단계를 수행해야 한다.

일반적으로, 가능한 패턴 수 p_i 는 주문 수 m 의 함수로 지수함수인 2^m 으로 증가한다. 결국, m 이 크면 가능한 절단 패턴들을 모두 나열하는 것은 현실적으로 불가능해진다. 따라서 사전에 모든 가능한 패턴을 구하는 사전 열 생성 (advance column generation) 대신 실시간 패턴 생성 (on-line pattern generation)으로 선형계획법을 적용해

최적해를 얻는 방법이 일반적으로 적용된다[2-5].

본 논문은 사전 열 생성 방법을 적용하여 최적의 롤 수를 구하는 간단한 방법을 제안한다. 2장에서는 재료절단 문제와 관련된 기존 연구와 문제점을 고찰한다. 3장에서는 최적해를 다항시간 복잡도로 쉽게 찾는 알고리즘을 제안한다. 4장에서는 2개 데이터에 제안된 알고리즘을 적용하여 최적해를 찾는 지 평가해 본다.

II. 관련 연구와 연구배경

재료절단 문제는 첫 번째로 실현 가능한 절단 패턴을 구해야만 한다. 일반적으로 가능한 패턴 수 p_i 는 주문 수 m 에 대해 $2^m - 1$ 개가 존재하여 m 이 크면 가능한 절단 패턴들을 모두 나열하여 최적해를 찾는 Brute-Force 방법은 현실적으로 불가능해진다. 두 번째로 가능한 절단 패턴들을 대상으로 최적해를 얻는 방법으로 일반적으로 선형계획법을 적용한다.

절단패턴은 사전에 모든 가능한 패턴을 구하는 사전 열 생성과 실시간 패턴 생성 방법 중 하나를 적용한다. 사전 열 생성 방법은 소형 또는 중형 규모의 주문에 사용되며, 실시간 패턴 생성 방법은 대형 규모의 주문에 대해 사전에 몇 개의 패턴으로 시작하여 필요시 추가적인 패턴을 생성하여 선형계획법 문제를 푸는데 사용된다.

사전 열 생성 방법은 Suliman[2]와 Goulimis[6]이 있다. Suliman[2]은 트리 개념을 적용하여 $\max l_j$ 부터 $\min l_j$ 까지 내림차순으로 레벨을 설정하여 가능한 모든 패턴을 찾는 방법을 제안하였다.

실시간 패턴 생성 방법은 Gilmore와 Gomory[7]의 지연 열 생성 (delayed column generation) 방법과 Haessler[8]의 순차적 휴리스틱 절차 (sequential heuristic procedure)가 있다. 지연 열 생성 방법은 몇 개의 패턴으로 시작하여 문제를 풀고, 필요시 추가적인 패턴을 생성하는 방법이며, 순차적 휴리스틱 절차는 모든 요구사항이 만족될 때까지 패턴을 순서대로 생성하는 방법이다.

재료절단 문제의 최적해를 얻는 방법은 선형계획법, 동적 계획법, 휴리스틱 방법, 유전자 알고리즘, 전문가 시스템들이 적용되고 있다.

현재까지 재료절단 문제와 관련된 벤치마킹 데이터는 주문

수 m 이 대용량인 데이터가 거의 없다. 따라서 사전 열 생성 방법으로 모든 실현 가능한 패턴을 구하고, 다항시간으로 정확한 해를 구하는 방법을 제시할 필요가 있다.

III. 선형 재료절단 알고리즘

제안된 알고리즘은 먼저 Suliman[2]이 제안한 재료절단 패턴 생성 (열 생성) 방법을 적용하여 가능한 모든 절단 패턴을 얻고, 절단 손실량을 최소로 하는 패턴들을 간단히 선택하는 과정을 다음과 같이 수행한다.

Step 1. 모든 실현 가능한 패턴 p_i 를 구하여 Trim Loss (Waste, w_i) 오름차순으로 정렬한다.

Step 2. 모든 요구 l_j 의 1st 발생이 존재하는 손실량 w_i 까지만 고려한다.

if 모든 요구 l_j 의 1st 발생이 손실량 $w_i = 0$ 에 미 존재 then

Step 3. 손실량 내림차순으로 해당 패턴 p_i 의 최소 요구량 $\min q_i$ 을 x_i 로 배정한다. 나머지 패턴들에 대해 여분의 요구량을 배정한다.

else if 모든 요구 l_j 의 1st 발생이 손실량 $w_i = 0$ 에 존재 then

Step 3. $|q_i|$ 내림차순으로 해당 패턴의 p_i 의 최소 요구량 $\min q_i$ 을 x_i 로 배정한다.

Step 4. 요구량을 충족시키지 못한 재료들을 포함하는 패턴 p_i 가 존재하면 해당 패턴을 추출하여 패턴의 최소 요구량 $\min q_i$ 을 배정한다. 만약, 해당 패턴이 존재하지 않으면 남은 재료들로 새로운 패턴을 생성하여 x_i 를 배정한다.

1,000Cm의 제지 제품을 생산하는 제지공장에서 표 1과 같이 주문을 받았을 경우에 대해 AIMMS[9]는 표 2와 같이 37개의 패턴을 얻고, 선형계획법과 LILE 알고리즘으로 해를 얻은 결과를 제시하였다.

표 1. 초기 데이터
Table 1. Example data for Paper Mill

Order (j)	Width (l_j)	주문량 (q_j)
1	450	97
2	360	610
3	310	395
4	140	211
합	415.240Cm	

표 2. AIMMS의 초기해
Table 2. Paper Mill Solution of AIMMS
(a) 패턴

p_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
450	2	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1	1	1	1	
360		1	1											2	1	1	1	1	1	
310			1	1	1									1	1	1	1	1	1	
140						3	2	1		2	1			2	1			4	3	2
w_i	100	50	190	100	240	130	270	410	550	0	140	280	20	50	190	330	80	220	360	
p_i	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37		

(b) 배정

l_j	450	360	310	140	선형계획법		LILE 알고리즘	
q_j	97	610	395	211	x_i	w_i	x_i	w_i
p_i	1	2			48.50	4.850	48	4,800
	10		2		105.50	0	105	0
	12		2		100.75	10.075	100	28,000
	13		1	2	197.50	3.930	197	3,940
	2	1	1		-	-	1	50
	12		2		-	-	1	280
	30			1	-	-	1	690
계					452.25	18.875	453	37,760

본 문제의 손실량을 최소로 하는 해는 $\sum_{i=1}^4 l_i q_i / L = 415.24$

로 416개의 롤이 필요하다. 그러나 본 데이터는 주문량 q_j 가 일일양포를 하지 않아 LILE 알고리즘은 453개의 롤이 필요하고 총 손실량은 37,760Cm이다. 반면에, 분수 (fractional) 선형계획법은 x_i 를 최대한으로 배정하였으나 우리가 얻고자 하는 값은 정수값으로 실제로 적용할 수 없다.

제안 알고리즘은 AIMMS[9] 데이터에 적용한 결과 얻은 해는 표 3과 같다. 제안된 알고리즘은 12개의 실현 가능한 패턴 수를 얻었으며, LILE 알고리즘과 동일하게 453개의 롤이 필요하며, 총 손실량은 37,760Cm이다.

표 3. 제안된 알고리즘 적용 해
Table 3. The Solution of Proposed Algorithm
(a) 패턴

w_i	p_i	450	360	310	140	수행순서
0	5		2		2	3
20	6		1	2		2
	12				7	4
50	2	1	1		1	1
	7		1	1	2	5

(b) 1st 발생 손실량

p_i	1	2	3	4	5	6	7	8	9	10	11	12
450	2	1	1	1								
360		1			2	1	1	1				
310			1			2	1		3	2	1	
140			1	3	2	1	2	4		2	4	7
w_i	100	50	100	130	0	20	50	80	70	100	130	20

(c) 배정

w_i	p_i	450	360	310	140	수행순서	롤 수
0	5		2x57		2x57	3	57
20	6		1x197	2x197		2	197
	12				7x0	4	0
50	2	1x97	1x97		1x97	1	97
	7		1x0	1x0	2x0	5	0
미 배정량		0	202	1	0		

(d) 미 배정량 추가 배정

w_1	p_1	450	360	310	140	롤 수
280	-		1x1	1x1		1
640	-		2x101			101
미 배정량		0	0	0	0	

IV. 알고리즘 적용성 평가

본 장에서는 표 4와 같이 Wikipedia[1]의 초지기 (paper machine) 데이터와 Umetani et al.[10,11]의 일 본의 화학섬유 (chemical fiber) 회사의 실제 수행된 fiber06_9080 데이터의 데이터를 대상으로 제안된 알고리즘 을 평가해 본다. 초지기의 최소로 필요한 롤 수는 72.7로 73 이며, fiber06_9080은 18.44로 19이다.

표 4. 실험 데이터
Table 4. Experimental Data
(a) 초지기

Jumbo Roll Widths(L): 5,600mm			Optimal Σx_i : 73 Master Roll		
Order (j)	Width (l_j)	주문량 (q_j)	Order (j)	Width (l_j)	주문량 (q_j)
1	1,380mm	22	8	2,000mm	10
2	1,520mm	25	9	2,050mm	12
3	1,560mm	12	10	2,100mm	14
4	1,710mm	14	11	2,140mm	16
5	1,820mm	18	12	2,150mm	18
6	1,880mm	18	13	2,200mm	20
7	1,930mm	20			
합	407,160mm				

(b) fiber06_9080

Chemical Fiber Widths(L): 9,080		Optimal Σx_i : 19 Roll
Order (j)	Width (l_j)	주문량 (q_j)
1	520	91
2	1000	11
3	1066	18
4	1120	9
5	1150	64
6	1250	5
합	167,438	

Step 1은 롤의 폭을 최대 요구 재료의 폭 $\max l_j$ 으로 나누 정수값을 취하면 첫 번째 패턴을 얻을 수 있다. 최대 요구 재 료부터 최소 요구 재료까지 가능한 모든 경우수를 구한다. 이 과정을 최소 재료 폭까지 반복적으로 수행하면 모든 패턴을 구할 수 있다. 초지기의 Step 1은 표 5에 제시되어 있으며, 213개의 패턴을 얻었다. 213개 패턴들 중 손실량 오름차순 으로 모든 요구의 1st 발생 손실량을 구한 결과는 표 6에 제 시되어 있다. 1st 발생 손실량 내림차순으로 배정 순서를 결 정하였다. 해당 순서에서 재료들의 요구량의 최소값을 배정하 고, 나머지 패턴들에 남아 있는 요구량을 배정한 결과는 표 7 에 제시되어 있다. 요구량을 충족시키지 못한 재료들을 포함 하는 패턴을 추출하여 손실량 오름차순으로 해당 패턴의 최소

표 5. 초지기의 실현 가능 패턴
Table 5. Feasible Patterns for Paper Mill

p_i	2200	2150	2140	2100	2050	2000	1930	1880	1820	1710	1560	1520	1380	w_i
1	2	0	0	0	0	0	0	0	0	0	0	0	0	1200
2	1	1	0	0	0	0	0	0	0	0	0	0	0	1250
3	1	0	1	0	0	0	0	0	0	0	0	0	0	1260
4	1	0	0	1	0	0	0	0	0	0	0	0	0	1300
5	1	0	0	0	1	0	0	0	0	0	0	0	0	1350
6	1	0	0	0	0	1	0	0	0	0	0	0	1	20
7	1	0	0	0	0	0	1	0	0	0	0	0	0	90
8	1	0	0	0	0	0	0	1	0	0	0	0	1	0
9	1	0	0	0	0	0	0	1	0	0	0	0	0	140
10	1	0	0	0	0	0	0	0	1	0	1	0	0	20
11	1	0	0	0	0	0	0	0	1	0	0	1	0	60
12	1	0	0	0	0	0	0	0	1	0	0	0	0	200
13	1	0	0	0	0	0	0	0	0	1	1	0	0	130
14	1	0	0	0	0	0	0	0	0	1	0	1	0	170
15	1	0	0	0	0	0	0	0	0	0	1	0	0	310
16	1	0	0	0	0	0	0	0	0	0	2	0	0	280
17	1	0	0	0	0	0	0	0	0	0	0	1	1	320
18	1	0	0	0	0	0	0	0	0	0	0	1	0	460
19	1	0	0	0	0	0	0	0	0	0	0	2	0	360
20	1	0	0	0	0	0	0	0	0	0	0	1	1	500
21	1	0	0	0	0	0	0	0	0	0	0	0	1	640
22	0	2	0	0	0	0	0	0	0	0	0	0	0	1300
23	0	1	1	0	0	0	0	0	0	0	0	0	0	1310
24	0	1	0	1	0	0	0	0	0	0	0	0	0	1350
25	0	1	0	0	1	0	0	0	0	0	0	0	1	20
26	0	1	0	0	0	1	0	0	0	0	0	0	1	70
27	0	1	0	0	0	0	1	0	0	0	0	0	1	0
28	0	1	0	0	0	0	1	0	0	0	0	0	0	140
29	0	1	0	0	0	0	0	1	0	0	0	1	0	10
30	0	1	0	0	0	0	0	1	0	0	0	1	0	50
31	0	1	0	0	0	0	0	1	0	0	0	0	1	190
32	0	1	0	0	0	0	0	1	0	1	0	1	0	70
33	0	1	0	0	0	0	0	0	1	0	0	1	0	110
34	0	1	0	0	0	0	0	0	1	1	0	0	1	250
35	0	1	0	0	0	0	0	0	0	2	0	0	0	30
36	0	1	0	0	0	0	0	0	0	2	0	0	0	30
37	0	1	0	0	0	0	0	0	0	1	1	0	0	180
38	0	1	0	0	0	0	0	0	0	1	0	1	0	220
39	0	1	0	0	0	0	0	0	0	1	0	0	1	360
40	0	1	0	0	0	0	0	0	0	0	2	0	0	330
41	0	1	0	0	0	0	0	0	0	0	1	1	0	370
42	0	1	0	0	0	0	0	0	0	0	1	0	1	510
43	0	1	0	0	0	0	0	0	0	0	0	2	0	410
44	0	1	0	0	0	0	0	0	0	0	0	1	1	550
45	0	0	2	0	0	0	0	0	0	0	0	0	2	690
46	0	0	1	1	0	0	0	0	0	0	0	0	0	1320
47	0	0	1	0	1	0	0	0	0	0	0	0	0	1360
48	0	0	1	0	1	0	0	0	0	0	0	0	1	30
49	0	0	1	0	0	1	0	0	0	0	0	0	1	80
50	0	0	1	0	0	0	1	0	0	0	0	0	1	0
51	0	0	1	0	0	0	0	1	0	0	0	1	0	150
52	0	0	1	0	0	0	0	1	0	0	0	1	0	20
53	0	0	1	0	0	0	0	1	0	0	0	0	1	60
54	0	0	1	0	0	0	0	1	0	0	0	0	1	200
55	0	0	1	0	0	0	0	0	1	0	0	1	0	80
56	0	0	1	0	0	0	0	0	1	0	0	0	1	120
57	0	0	1	0	0	0	0	0	0	2	0	0	0	260
58	0	0	1	0	0	0	0	0	0	1	1	0	0	40
59	0	0	1	0	0	0	0	0	0	1	0	1	0	190
60	0	0	1	0	0	0	0	0	0	1	0	1	0	230
61	0	0	1	0	0	0	0	0	0	1	0	0	1	370
62	0	0	1	0	0	0	0	0	0	2	0	0	0	340
63	0	0	1	0	0	0	0	0	0	0	1	1	0	380
64	0	0	1	0	0	0	0	0	0	0	0	2	0	520
65	0	0	1	0	0	0	0	0	0	0	0	1	1	420
66	0	0	1	0	0	0	0	0	0	0	0	0	2	560
67	0	0	1	0	2	0	0	0	0	0	0	0	2	700
68	0	0	0	1	1	0	0	0	0	0	0	0	1	20
69	0	0	0	1	1	0	1	0	0	0	0	0	0	70
70	0	0	0	1	0	0	1	0	0	0	0	1	0	120
71	0	0	0	1	0	0	1	0	0	0	0	0	1	10
72	0	0	0	1	0	0	1	0	0	0	0	0	1	50
73	0	0	0	1	0	0	0	1	0	0	0	0	1	190
74	0	0	0	1	0	0	0	1	0	0	0	1	0	80
75	0	0	0	1	0	0	0	1	0	0	0	0	1	100
76	0	0	0	1	0	0	0	1	0	0	0	0	1	240
77	0	0	0	1	0	0	0	1	0	0	0	1	0	120
78	0	0	0	1	0	0	0	1	0	0	0	1	0	160
79	0	0	0	1	0	0	0	0	1	0	0	0	1	300
80	0	0	0	1	0	0	0	0	0	2	0	0	0	80
81	0	0	0	1	0	0	0	0	0	1	1	0	0	230
82	0	0	0	1	0	0	0	0	0	1	0	0	1	270
83	0	0	0	1	0	0	0	0	0	1	0	0	1	410
84	0	0	0	1	0	0	0	0	0	2	0	0	0	380
85	0	0	0	1	0	0	0	0	0	0	2	0	0	410
86	0	0	0	1	0	0	0	0	0	0	1	0	1	560
87	0	0	0	1	0	0	0	0	0	0	0	2	0	460
88	0	0	0	1	0	0	0	0	0	0	0	0	2	600
89	0	0	0	1	0	0	0	0	0	0	0	0	2	74

108	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	610
109	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2	0	510
110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	650
111	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	2	790
112	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	40
113	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	80
114	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	220
115	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	110
116	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	150
117	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	290
118	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	10
119	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	160
120	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	200
121	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	340
122	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	70
123	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	220
124	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	260
125	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	400
126	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	180
127	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	330
128	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	370
129	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	510
130	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	480
131	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	520
132	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	660
133	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	560
134	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	700
135	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	840
136	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	30
137	0	0	0	0	0	0	2	2	0	0	0	1	0	0	0	0	180
138	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	220
139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	360
140	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	80
141	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	230
142	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	270

표 6. 1st 발생 손실량
Table 6. First Loss Quantity

w_i	p_i	2200 (20)	2150 (18)	2140 (16)	2100 (14)	2050 (12)	2000 (10)	1930 (20)	1880 (18)	1820 (18)	1710 (14)	1560 (12)	1520 (25)	1380 (22)	$ q_i $
0	8	1	0	0	0	0	0	0	1	0	0	0	1	0	3
	27	0	1	0	0	0	0	0	1	0	0	0	0	1	3
	29	0	1	0	0	0	0	0	1	0	0	0	1	0	3
	49	0	0	1	0	0	0	1	0	0	0	0	1	0	3
10	70	0	0	0	1	0	0	1	0	0	0	0	1	0	3
	118	0	0	0	0	0	0	1	0	1	0	1	0	0	3
	6	1	0	0	0	0	1	0	0	0	0	0	0	1	3
	10	1	0	0	0	0	0	0	0	1	0	0	0	0	3
20	25	0	1	0	0	1	0	0	0	0	0	0	0	0	3
	51	0	0	1	0	0	0	1	0	0	0	1	0	0	3
	67	0	0	0	2	0	0	0	0	0	0	0	0	1	3
	98	0	0	0	0	1	0	0	0	1	1	0	0	0	3
	159	0	0	0	0	0	0	0	2	1	0	0	0	0	3

표 7. 배정 순서 결정 및 배정
Table 7. Decision for Allocation Sequence and Assignment

w_i	p_i	2200 (20)	2150 (18)	2140 (16)	2100 (14)	2050 (12)	2000 (10)	1930 (20)	1880 (18)	1820 (18)	1710 (14)	1560 (12)	1520 (25)	1380 (22)	x_i
0	8	○							○	○				○	0
	27		○											○	0
	29			○						○			○		0
	49				16				16				16	16	0
10	70				0							0	0	0	0
	118						10		10		10			10	0
	6	○						○						○	0
	10	12								12		12		12	0
20	25		12			12								12	12
	51			○					○				○		0
	67				○,○					○	○			○	0
	98					○				○	○			○	0
	159								○,○					○	0
여분		8	6	0	14	0	0	4	8	6	4	0	9	10	

요구량을 배정한 결과는 표 8에 제시되어 있다.

fiber06_9080 데이터의 실현 가능 패턴수는 1,047개를 얻었으며 표 9에 일부만 제시하였다. 모든 요구의 1st 발생 손실량을 구한 결과는 표 10에 제시되어 있다. 1st 발생 손실량이 0인 7개 패턴에 모든 요구가 포함되어 각 패턴의 $|q_i|$ 내림차순으로 배정 순서를 결정하였다. 해당 순서에서 재료들의 요구량의 최소값을 배정하고, 나머지 패턴들에 남아 있는 요구량을 배정한 결과는 표 11에 제시되어 있다. 요구량을 충족시키지 못한 재료들을 포함하는 패턴을 추출하여 손실량 오름차순으로 해당 패턴의 최소 요구량을 배정한 결과는 표 12에 제시되어 있다.

표 8. 부족분 배정
Table 8. Assignment of Shortage

w_i	p_i	2200 (0)	2150 (5)	2140 (0)	2100 (0)	2050 (0)	2000 (0)	1930 (3)	1880 (0)	1820 (6)	1710 (4)	1560 (0)	1520 (0)	1380 (3)
30	35		○								○			
30	136							○				○		
30	144							○		○				
140	28		○											○
140	145							○		○				
140	179								○	○				
여분		0	5	0	0	0	0	3	0	6	4	0	0	3

w_i	p_i	2200 (0)	2150 (5)	2140 (0)	2100 (0)	2050 (0)	2000 (0)	1930 (3)	1880 (0)	1820 (6)	1710 (4)	1560 (0)	1520 (0)	1380 (3)	x_i
30	35		2							2.2					2
30	136							0.0							0
30	144							0		0.0					0
140	28		3					3						3	3
140	145							0		0					0
140	179								2.2	2					2
여분		0	0	0	0	0	0	0	0	0	0	0	0	0	0

표 9. fiber06_9080의 실현 가능 패턴
Table 9. Feasible Patterns for fiber06_9080

p_i	1250	1150	1120	1066	1000	520	w_i	p_i	1250	1150	1120	1066	1000	520	w_i
1	7						330	921			5	1		4	334
2	6						430	922			4	4		1	336
3	6	1					480	923			4	3	1		402
4	6		1				514	924			4	3		2	362
5	6			1		1	60	925			4	2	2		468
6	6				1	3	20	926			4	2	1	2	428
7	5	2				1	10	927			4	2	4	4	388
8	5	1	1			1	40	928			4	1	3	1	14
9	5	1		1		1	94			...					
10	5	1			1	1	160	1003			8			1	32
11	5		2			1	70	1004			7	1	1	1	98
12	5		1	1		1	124	1005			7			3	58
13	5		1		1	1	190	1006			6	2	1	1	164
14	5		1			3	150	1007			6	1	3	1	124

15	5			2		1	178	1008			6			5	84
16	5			1	1	1	244	1009			5	3	1	1	230
17	5			1		3	204	1010			5	2	3		190
18	5				2	1	310	1011			5	1	5	1	150
19	5				1	3	270	1012			4	4	1	1	296
20	5					5	230	1013			4	3	3	3	256
				...				1014			4	2	5	2	216
594	7				1		30	1015			4	1	7	1	176
595	7					1	510	1016			3	5	1		362
596	6			2			48	1017			3	4	3	3	322
597	6		1	1			114	1018			3	3	5	5	282
598	6		1			2	74	1019			3	2	7	2	242
599	6				2		180	1020			3	1	9	2	202
600	6			1	2	140	1021				3		11	1	162
601	6				4	100	1022				2	6	1	4	428
602	5	2	1			24	1023				2	5	3	3	388
603	5	2		1		90	1024				2	4	5	5	348
604	5	2			2	50	1025				2	3	7	3	308
605	5	1	2			78	1026				2	2	9	2	268
606	5	1	1	1		144	1027				2	1	11	1	228
607	5	1	1		2	104	1028				2		13	1	188
608	5	1		2		210	1029				1	8		14	14
609	5	1	1		2	170	1030				1	7	1	1	494
610	5	1		4	130	1031					1	6	3	3	454
611	5		3			132	1032				1	5	5	4	414
612	5		2	1		198	1033				1	4	7	3	374
613	5		2		2	158	1034				1	3	9	3	334
			...				1035				1	2	11	2	294
909	8					120	1036				1	1	13	1	254
910	7	1				174	1037				1		15	1	214
911	7		1			240	1038					9		8	80
912	7			2	200	1039						8	2	4	40
913	6	2			228	1040						7	4	0	0
914	6	1	1		294	1041						6	5	4	80
915	6	1		2	254	1042						5	7	4	400
916	5	3			282	1043						4	9	4	400
917	5	2	1		348	1044						3	11	3	360
918	5	2		2	308	1045						2	13	2	320
919	5	1	2		414	1046						1	15	2	280
920	5	1	1	2	374	1047							17	2	240

표 10. fiber06_9080의 1st 발생 손실량
Table 10. First Loss Quantity of fiber06_9080

w_i	p_i	1250 (5)	1150 (64)	1120 (9)	1066 (18)	1000 (11)	520 (91)
46		4				2	4
113		3			5		
154		2	2	2		1	2
479		1	1			2	9
620			4	4			
877			1		5		5
1039						7	4

표 11. 배정 순서 결정 및 배정
Table 11. Decision for Allocation Sequence and Assignment

w_i	p_i	1250 (5)	1150 (64)	1120 (9)	1066 (18)	1000 (11)	520 (91)	$ q_i $	배정순서
46		4				2	4	10	4
113		3			5			8	6
154		2	2	2		1	2	9	5
479		1	1			2	9	13	1
620			4	4				8	7
877			1		5		5	11	2
1039						7	4	11	3

w_i	p_i	1250 (5)	1150 (64)	1120 (9)	1066 (18)	1000 (11)	520 (91)	x_i
46		0				0	0	0
113		0			0			0
154		0	0	0		0	0	0
479		1x5	1x5			2x5	9x5	5
620			4x2	4x2				4
877			1x3		5x3		5x3	3
1039						0	0	0
여분		5-5 = 0	64-16 = 48	9-6 = 3	18-15 = 3	11-10 = 1	91-60 = 31	

표 12. 부족분 배정
Table 12. Assignment of Shortage

w_i	p_i	1250 (0)	1150 (48)	1120 (1)	1066 (3)	1000 (1)	520 (31)	x_i
100	600	0	6x7	0	0	0	4x7	7
100	여분	0	6	1	3	1	3	
144	605	0	5x1	1x1	1x1	1x1	0	1
144	여분	0	1	0	2	0	3	
4,238	-	0	1x1	0	2x1	0	3x1	1
4,238	여분	0	0	0	0	0	0	

$tloss = 4,482, 9082 \times 17 = 154,360 \Rightarrow$ 손실율: 2.9%

제안된 알고리즘과 기존의 알고리즘의 성능을 비교 평가한 결과는 표 13에 제시되어 있다. 제안 알고리즘의 총 손실량은 가장 좋은 결과를 얻었다. 또한, 마지막 패턴 4,238은 후후 1250+1150+1120+520=4,040으로 활용이 가능하다. 결국, 제안된 알고리즘의 총 손실량은 442 (0.29%)로 손실이 거의 발생하지 않음을 알 수 있다.

표 13. 알고리즘 성능 평가
Table 13. Compare of Performance for Algorithms

알고리즘	p_i	Σx_i	Σw_i
SHP[1]	5	19	3.04%
	5	20	2.95%
ILS[10,11]	5	20	8.46%
제안 알고리즘	6	19	2.90%

V. 결론 및 향후 연구과제

본 논문은 선형 재료절단 문제의 정확한 해를 다항시간 복잡도로 구하는 알고리즘을 제안하였다. 제안된 알고리즘은 먼저 실현 가능한 패턴을 간단히 구하는 방법을 적용하였으며, 모든 요구에 대한 손실량이 최소가 되는 패턴들을 추출하여 패턴의 수를 결정하였다. 만약, 모든 요구의 1st 발생이 손실량 0에 미 존재시 손실량이 최대인 요구부터 최소 손실량 요구로 내림차순으로 배정 순서를 결정하였으며, 모든 요구의 1st 발생이 손실량 0에 존재하는 경우 각 패턴의 q_i 를 계산하여 내림차순으로 패턴 수 x_i 의 배정순서를 결정하였다. 다음으로 충족시키지 못한 요구량에 대해서는 해당 요구를 포함하는 패턴을 찾아 요구량을 배정하는 방법을 적용하였다. 2개의 실험 데이터에 제안된 알고리즘을 적용한 결과 정확한 해를 구하는데 성공하였다.

본 논문에서는 제안된 알고리즘을 1D-CSP에만 적용되었으며, 추후 보다 어려운 2D-CSP 문제에 적용하여 적합성을 검증할 예정이다.

참고문헌

[1] Wikipedia, "Cutting Stock Problem," http://en.wikipedia.org/wiki/Cutting_stock_problem, Wikimedia Foundation Inc., 2013.

[2] S. M. A. Suliman, "Pattern Generating Procedure for the Cutting Stock Problem," *International Journal of Production Economics*, Vol. 74, No. 1-3, pp. 293-301, Dec 2001.

[3] S. Umetani, M. Yagiura, and T. Ibaraki, "A Local Search Approach for One Dimensional Cutting Stock Problem," MIC 2001 - 4th Metaheuristics International Conference, Portugal, Jul 2001.

[4] J. Nazemi, "Kiln Planning, A Cutting Stock Approach," Industrial Engineering Department, AZAD University, Tehran, IRAN, 2008.

[5] G. Belov and G. Scheithauer, "The Number of Setups (Different Patterns) in One-Dimensional Stock Cutting," *Institute for Numerical Mathematics*, Dresden University, 2003.

[6] C. Goulimis, "Optimal Solutions for the Cutting

Stock Problem," *European Journal of Operational Research*, Vol. 44, No. 2, pp. 197-208, Jan 1990.

[7] P. C. Gilmore and R. E. Gomory, "A Linear Programming Approach to the Cutting Stock Problem," *Operations Research*, Vol. 9, No. 6, pp. 849-859, Nov 1961.

[8] R. W. Haessler, "Controlling Cutting Pattern Changes in One-Dimensional Trim Problems," *Operations Research*, Vol. 23, No. 3, pp. 483-493, May 1975.

[9] J. Bisschop and M. Roelofs, "AIMMS Optimization Modeling, AIMMS 3.8," *Paragon Decision Technology*, 2007.

[10] S. Umetani, "Combinatorial Optimization and Algorithms: Benchmark Problem Instances" Department. of Advanced Science and Technology, Graduate School of Engineering, Toyota Technological Institute, Nagoya city, Japan, 2003.

[11] S. Umetani, M. Yagiura, and T. Ibaraki, "One Dimensional Cutting Stock Problem to Minimize the Number of Different Patterns," *European Journal of Operational Research*, Vol. 146, No. 2, pp. 388-402, Apr 2003.

저 자 소개



이 상 운(Sang-Un, Lee)

1983년 ~ 1987년 :

한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 :

경상대학교 컴퓨터과학과 (석사)

1998년 ~ 2001년 :

경상대학교 컴퓨터과학과 (박사)

2003.3 ~ 현재 :

강릉원주대학교 멀티미디어공학과 부교수

관심분야 : 소프트웨어 프로젝트 관리,

소프트웨어 개발 방법론,

소프트웨어 신뢰성,

신경망, 뉴로-퍼지,

그래프 알고리즘

e-mail : sulee@gwnu.ac.kr