

무제한 2차원 절단문제에 대해 개선된 최적-우선 분지한계 해법

윤기섭* · 방성규** · 강맹규*

An Improved Best-First Branch and Bound Algorithm for Unconstrained Two-Dimensional Cutting Problems

Ki-Seop Yoon* · Sung-Kyu Bang** · Maing-Kyu Kang*

■ Abstract ■

In this paper, we develop an improved branch and bound algorithm for the (un)weighted unconstrained two-dimensional cutting problem. In the proposed algorithm, we improve the branching strategies of the existing exact algorithm and reduce the size of problem by removing the dominated pieces from the problem. We apply the newly proposed definition of dominated cutting pattern and it can reduce the number of nodes that must be searched during the algorithm procedure. The efficiency of the proposed algorithm is presented through comparison with the exact algorithm known as the most efficient.

Keyword : Cutting Problem, Two-dimensional, Branch and Bound, Optimization

1. 서 론

2차원 절단문제는 하나의 큰 직사각형 자재(plate)로부터 크기와 가치가 각각 주어진 직사각형

부품(piece)들의 가치합이 최대가 되는 절단패턴(cutting pattern)을 찾는 문제이다. 부품은 자재로부터 잘라내야 하는 작은 조각(piece)이다. 부품은 길이, 너비, 가치를 가지며, 잘라낼 수 있는 개수의

논문접수일 : 2004년 7월 8일

논문게재확정일 : 2005년 10월 6일

* 한양대학교 산업공학과

** 현대자동차

제한이 있을 수 있다. 절단패턴은 하나의 자재로부터 잘린 부품들의 조합을 의미한다. 길이 l , 너비 w 인 직사각형의 크기를 (l, w) 라 하자. 자재의 크기가 (L, W) , 부품 i 의 크기, 가치, 개수제한이 각각 (l_i, w_i) , p_i , b_i 로 주어졌을 때, 2차원 절단문제는 다음과 같이 모형화할 수 있다.

$$\begin{aligned} & \text{최대화} && \sum_{i=1}^n p_i x_i \\ & \text{제약조건} && \text{가능절단패턴} \\ & && 0 \leq x_i \leq b_i, \quad i=1, 2, 3, \dots, n \end{aligned} \quad (1.1)$$

여기서 x_i 는 잘라낸 부품 i 의 개수, 즉 절단패턴에 포함된 부품 i 의 개수이다. 가능절단패턴 제약은 절단된 부품의 면적의 합이 자재의 면적을 넘을 수 없다는 것과 절단된 부품들은 서로 중첩되지 않아야 한다는 것을 포함한다. 이 두 가지 제약 이외에 다른 제약이 추가되면 절단패턴은 이러한 모든 제약을 만족해야 한다. 절단문제는 특별한 언급이 없는 경우, 길로틴(guillotine) 절단 제약을 고려한다. 길로틴 절단은 한번 절단을 시작하면 도중에 멈추지 않고 자재를 두 개의 작은 자재로 만드는 절단을 의미한다. 일반적으로 2차원 절단문제는 절단되는 부품 개수의 제한이 없는 무제한 2차원 절단(unconstrained two-dimensional cutting : UTDC) 문제와 제한이 있는 2차원 절단(constrained two-dimensional cutting : CTDC)문제로 나눌 수 있다. 또한 2차원 절단문제는 부품의 가치가 부품의 면적으로 주어지는 비가중(unweighted) 문제와 부품의 면적과 관계없이 주어지는 가중(weighted) 문제로 나눌 수 있는데, 본 연구에서는 비가중과 가중 특성을 가진 길로틴 UTDC문제를 다룬다.

길로틴 제약을 고려한 UTDC문제에 대해 Gilmore와 Gomory[6]는 비가중 문제와 가중 문제 모두에 적용할 수 있는 동적 계획법(dynamic programming)을 제안하였다. Herz[7]는 이 방법보다 더 효율적인 깊이-우선탐색(depth-first search)을 이용한 분지한계법을 제안하였으나 가중 문제에 대해서는 적용할 수 없다. Beasley[1]는 Gilmore와 Gomory

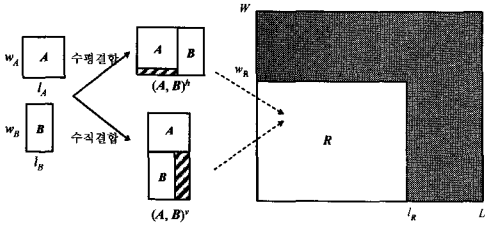
의 해법을 변형하여 큰 문제에도 적용할 수 있는 해법을 제안하였다. Morabito et al.[10]은 AND-OR 그래프로 문제를 표현하여 깊이-우선과 언덕등반(hill climbing) 탐색을 이용한 해법을 제안하였다. Hifi와 Zissimopoulos[9]는 효율적인 상한(upper bound)과 하한(lower bound)을 이용한 동적 계획법을 제안하였다. G et al.[4]은 최적-우선 분지한계법(best-first branch and bound)을 제안하였는데, 이 해법은 UTDC문제에 대해 지금까지 알려진 가장 우수한 최적 해법이다. 본 논문에서는 G et al.[4]의 최적-우선 분지한계법의 분지전략(branching strategy)을 개선한 해법을 제안하고 실험 결과를 통해 제안하는 해법의 우수성을 보인다.

본 논문의 구성은 다음과 같다. 2장과 3장에서 G et al.[4]이 사용한 상향식 접근법과 최적-우선 분지한계법, 분지전략에 대해 정리한다. 4장에서는 G et al.[4]이 정의한 열등패턴(dominated pattern)의 개념을 확장하여 재정의한다. 5장에서는 4장에서 정의한 열등패턴의 정의를 이용하여 열등부품(dominated piece)를 정의하고 이를 문제에서 원천적으로 제거하는 방법을 설명한다. 6장에서는 4장의 내용을 적용하여 G et al.[4]의 분지전략을 개선한다. 7장에서는 다양한 문제들을 실험하여 제시한 해법의 우수성을 보인다. 8장에서는 결론을 맺는다.

2. 최적-우선 분지한계법

G et al.[4]은 UTDC에 대해 상향식 접근법을 이용한 최적-우선 분지한계법을 제시하였다. 상향식 접근법은 두 개의 절단패턴 A, B 를 수평결합(horizontal build) 또는 수직결합(vertical build)하여 새로운 절단패턴을 생성하는 방법이다[11]. 절단패턴 A 와 B 의 수평결합과 수직결합을 각각 $(A, B)^h$ 와 $(A, B)^v$ 로 정의하자. 절단패턴 A 와 B 의 크기를 각각 (l_A, w_A) 와 (l_B, w_B) , 가치를 각각 $g(A)$ 와 $g(B)$ 라 하면 A 와 B 의 수평결합 $(A, B)^h$ 의 크기는 $(l_A + l_B, \max(w_A, w_B))$ 이고 가치는 $g(A) + g(B)$

가 된다([그림 2.1] 참조). 같은 방법으로 A 와 B 의 수직결합 $(A, B)^v$ 는 크기가 $(\max(l_A, l_B), w_A + w_B)$ 이고 가치가 $g(A) + g(B)$ 인 절단패턴을 생성한다. A 와 B 의 수평 또는 수직결합으로 새롭게 생성된 절단패턴 R 은 자재의 왼쪽 아래부분에 위치한다.



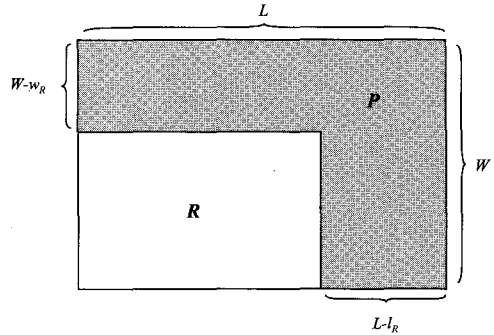
[그림 2.1] 수평 및 수직결합

최적-우선 분지한계법은 최적해가 될 가능성이 가장 높은 노드, 즉 상한이 가장 큰 노드를 우선적으로 분지한다. 최적-우선 분지한계법의 일반적인 절차는 다음과 같다.

- 1) 분지할 노드집합, 즉 잎새(leaf) 노드집합에서 상한이 가장 높은 노드를 선택한다.
- 2) 선택한 노드를 분지한 노드집합으로 옮긴다.
- 3) 선택한 노드를 분지하여 자식노드를 생성한다.
- 4) 생성된 자식노드중 상한이 현재까지의 최적값보다 좋은 노드만 분지할 노드집합에 추가한다.

최적-우선 분지한계법에 상향식 접근법을 이용할 경우 각 노드는 하나의 절단패턴을, 분지는 수평결합 또는 수직결합을 나타낸다. 또한 자식노드는 부모노드와 분지된 노드들 간에 수평결합 또는 수직결합으로 생성된 새로운 절단패턴을 나타낸다. 최적-우선 분지한계법으로 UTDC를 풀기 위해서는 노드에 해당하는 임의의 절단패턴 R 에 대한 상한을 구해야 한다. [그림 2.2]는 임의의 절단패턴 R 과 자재에서 R 을 제외한 영역, 즉 사용되지 않은 영역 P 를 나타낸다. $g(R)$ 을 절단패턴 R 의 가치, $h(R)$ 을 영역 P 의 최대가치라고 하면, $f(R) = g(R) + h(R)$ 은 R 을 포함한 절단패턴의 최대가치이다. $g(R)$ 은 절단패턴 R 에 포함된 부품의 가치합으로

계산할 수 있다. $h(R)$ 은 정확한 값을 계산하기가 힘들지만 $h(R)$ 의 상한인 $h'(R)$ 은 효율적으로 구할 수 있다. 따라서 $f(R) = g(R) + h'(R)$ 이 절단패턴 R 의 상한이 될 수 있다.



[그림 2.2] 절단패턴 R 과 영역 P

상향식 접근법을 이용한 최적-우선 분지한계법의 절차를 정리하면 다음과 같다.

단계 1. [초기화]

분지될 노드집합 O 는 $O = \{R_1, R_2, \dots, R_n\}$ 이다. 여기서 R_i 는 부품 i 한 개로 구성된 절단패턴을 나타낸다. 분지된 노드 집합 C 는 \emptyset 이고, 현재의 최적해 $z = 0$ 이다.

단계 2. [분지과정]

(2.1) 집합 O 에서 가장 큰 상한, 즉 가장 큰 f 을 갖는 절단패턴 R 을 찾는다. 만약 O 가 \emptyset 이면 단계 3으로 간다.

(2.2) 선택된 절단패턴 R 을 분지한다. 즉 R 을 O 에서 C 로 옮기고, 수평과 수직결합으로 모든 가능절단패턴들의 집합 Q 를 생성한다. $q \in Q$ 은 R 과 C (R 을 포함하는)안의 모든 절단패턴들과의 수평과 수직결합으로 생성되며, q 의 크기 (l_q, w_q) 는 $(l_q, w_q) \leq (L, W)$ 를 만족한다. 즉, 절단패턴의 크기가 자재보다 큰 비가능(infeasible) 해는 제거한다.

(2.3) 절단패턴 $q \in Q$ 에 대해서 q 의 가치 $g(q)$ 가 $g(q) > z$ 이면 z 을 $g(q)$ 로 갱신하고, q 의 상한 $f'(q)$ 가 $f'(q) > z$ 이면 q 를 집합 O 에

추가한다. 그렇지 않으면 q 를 제거한다. 단계 2를 반복한다.

단계 3. [종료]

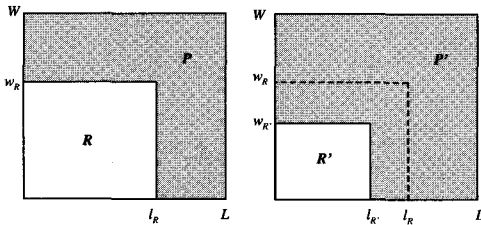
최적해 \bar{z} 을 리턴하고 종료한다.

3. 분지전략

G et al.[4]은 해법 수행중 불필요한 탐색을 줄이기 위해 다음 세 가지의 분지전략을 제안하였다.

[전략 1] 임의의 두 절단패턴 R 과 R' 에 대해 R 을 O 에서 C 로 옮길 때, C 에 식 (3.1)을 만족하는 절단패턴 R' 이 존재하면 R 을 제거한다. 이때, R 을 열등패턴(dominated pattern)이라고 한다.

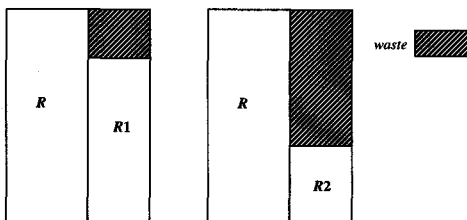
$$(l_R, w_R) \geq (l_{R'}, w_{R'}) \text{ and } g(R) \leq g(R') \quad (3.1)$$



[그림 3.1] 열등패턴 : R vs. R'

[전략 2] (1) 집합 O 에 포함되어 있는 절단패턴 R 에 대해 집합 C 에 속한 두 절단패턴 $R1$ 과 $R2$ 가 식 (3.2)를 만족하면 $H2 = (R, R2)^h$ 는 $H1 = (R, R1)^h$ 에 의해 제거된다. 수직결합에도 같은 방법을 적용한다.

$$w_R \geq w_{R1} \geq w_{R2} \text{ and } l_{R1} = l_{R2} \quad (3.2)$$

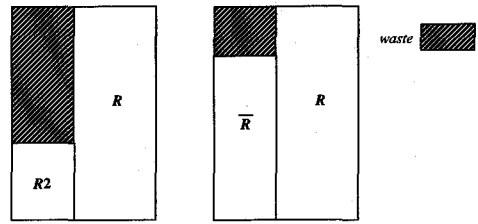


(a) $H1 = (R, R1)^h$ (b) $H2 = (R, R2)^h$

[그림 3.2] $(R, R1)^h$ vs. $(R, R2)^h$

(2) 집합 O 에 포함되어 있는 절단패턴 $R2$ 에 대해 식 (3.3)를 만족하는 절단패턴 \bar{R} 가 C 에 존재하면 $H2 = (R2, R)^h$ 는 $\bar{H} = (\bar{R}, R)^h$ 에 의해 제거된다. 수직결합에도 같은 방법을 적용한다.

$$w_{\bar{R}} = \min\{w_{R1} \mid l_{R1} = l_{R2}, w_{R1} > w_{R2}, R1 \in C\} \quad (3.3)$$



(a) $H2 = (R2, R)^h$ (b) $\bar{H} = (\bar{R}, R)^h$

[그림 3.3] $(R2, R)^h$ vs. $(\bar{R}, R)^h$

[전략 3] 임의의 절단패턴 q 를 집합 O 에 넣을 때, O 에 있는 R 에 대해 식 (3.4)가 만족되면 q 를 제거한다.

$$(l_q, w_q) = (l_R, w_R) \text{ and } g(q) \leq g(R) \quad (3.4)$$

4. 열등패턴 재정의

열등패턴의 정의를 이용하면 분지과정중에 생기는 열등한 패턴을 제거하여 많은 불필요한 분지과정을 생략할 수 있다. G et al.[4]은 임의의 절단패턴 R 과 R' 이 식 (3.1)을 만족할 경우 R 을 열등패턴으로 정의하고, R' 에 의해 R 이 제거될 수 있다고 하였다. G et al.[4]은 식 (3.1)의 열등패턴을 정의할 때 패턴 자체의 크기를 고려하였으나, 본 연구에서는 자재에서 패턴을 제외한 부분([그림 4.1]에서 영역 P)의 크기를 고려한다. 다음의 [성질 1]과 [성질 2]을 이용하여 열등패턴을 새롭게 정의한다. $(a, b) \geq (c, d)$ 의 의미를 $a \geq c, b \geq d$ 라고 하자 (\geq 대신 $\leq, =$ 도 같은 식으로 정의함).

[성질 1] 크기가 각각 $(l_R, w_R), (l_{R'}, w_{R'})$ 인 임의의 두 절단패턴 R 과 R' 이 있다고 하자. 자재에

서 R 과 R' 을 제외한 영역을 각각 P 와 P' 이라고 할 때, 식 (4.1)이 성립하면 P 와 P' 의 가치 $g(P)$ 와 $g(P')$ 의 관계는 $g(P) \leq g(P')$ 이다.

$$(S_x(L - l_R), S_y(W - w_R)) \leq (S_x(L - l_{R'}), S_y(W - w_{R'})) \quad (4.1)$$

여기서 $S_x(X)$ 와 $S_y(Y)$ 는 각각 X 와 Y 를 넘지 않으면서 낭비되는 부분을 최소로 하는 부품 길이와 너비의 조합이다. n 종류의 부품에 대해 부품 i 의 크기가 (l_i, w_i) 로 주어졌을 때, $S_x(X)$ 와 $S_y(Y)$ 는 식 (4.2)를 이용하여 구할 수 있다.

$$S_x(X) = \max \{ \sum l_i \alpha_i : \sum l_i \alpha_i \leq X, \alpha_i \text{는 양의 정수, for } i=1, 2, \dots, n \}$$

$$S_y(Y) = \max \{ \sum w_j \beta_j : \sum w_j \beta_j \leq Y, \beta_j \text{는 양의 정수, for } j=1, 2, \dots, n \} \quad (4.2)$$

[증명] 영역 P 는 [그림 2.2]와 같다. 이때, $S_x(L - l_R)$ 과 $S_y(W - w_R)$ 은 각각 $L - l_R$ 과 $W - w_R$ 을 최대로 채울 수 있는 크기가 된다. 같은 방법으로 $S_x(L - l_{R'})$ 과 $S_y(W - w_{R'})$ 은 각각 $L - l_{R'}$ 과 $W - w_{R'}$ 을 최대로 채울 수 있는 크기이다. 따라서, 절단 패턴 R 과 R' 에 대해 식 (4.1)이 만족되면 P' 에는 P 보다 더 많은 패턴 조합을 사용할 수 있으므로 $g(P) \leq g(P')$ 이다.

[성질 2]는 [성질 1]을 이용하여 열등패턴을 새로 정의한다. 즉, 특정 패턴을 제외한 나머지 영역의 크기를 고려하여 기존 열등패턴의 개념을 확장한다.

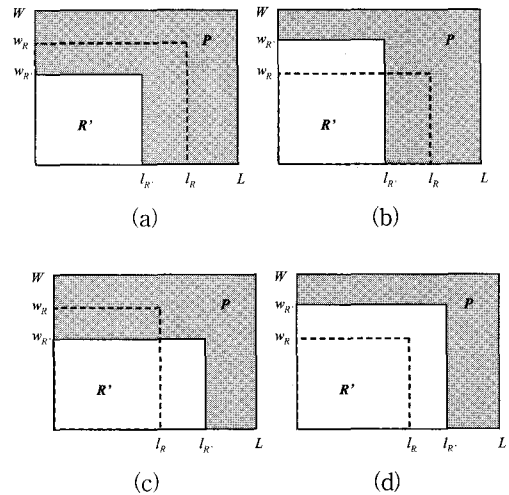
[성질 2] 크기가 각각 (l_R, w_R) , $(l_{R'}, w_{R'})$ 인 임의의 두 절단패턴 R 과 R' 이 식 (4.3)을 만족하면 절단패턴 R 은 R' 에 의해 제거된다.

$$(S_x(L - l_R), S_y(W - w_R)) \leq (S_x(L - l_{R'}), S_y(W - w_{R'})) \text{ and } g(R) \leq g(R') \quad (4.3)$$

[증명] 절단패턴 R 과 R' 이 식 (4.3)을 만족한다고

하자. 자재에서 R 과 R' 을 제외한 영역을 각각 P 와 P' 이라고 하면 성질1에 의해 P 와 P' 의 가치 $g(P)$ 와 $g(P')$ 은 $g(P) \leq g(P')$ 을 만족한다. 또한 R' 의 가치가 R 의 가치보다 더 크므로 R 은 R' 에 의해 제거될 수 있다.

새로 정의한 열등패턴과 기존 열등패턴의 차이는 기존의 열등패턴은 [그림 4.1]의 (a)와 같이 R' 의 크기가 R 보다 작아야만 적용이 가능하나, 새로 정의한 열등패턴은 [그림 4.1]의 (b), (c), (d)와 같이 R' 의 크기가 R 보다 작지 않더라도 R 을 제외한 영역에 사용할 수 있는 길이와 너비 조합이 같으면 적용이 가능하다는 점이다.



[그림 4.1] 열등패턴의 예

5. 열등부품 제거

성질 2를 이용해 문제에서 주어지는 부품중 열등한 부품을 제거할 수 있다. 이는 해법 수행전 최적해에 포함되지 않거나 포함되더라도 최적값에 영향을 주지 않는 부품을 문제에서 원천적으로 제거하는 것이다. 이러한 부품을 열등부품(dominated piece)이라 하자. 이 방법은 UTDC의 모든 해법에 적용될 수 있고, 문제 크기를 축소시키는 것이므로 해법의 성능을 크게 향상시킨다. 또한 제안하는 방

법을 이용하면 부품의 가치가 부품의 면적으로 주어지는 비가중 문제에서도 열등부품을 제거할 수 있다. 다음의 [성질 3]을 이용하여 열등부품을 정의한다. [성질 3]의 증명은 [성질 2]의 증명과 동일하다.

[성질 3] 크기가 각각 (l_i, w_i) 와 (l_j, w_j) 인 임의의 부품 i 와 j 가 식 (5.1)을 만족할 경우 부품 i 는 문제에서 원천적으로 제거된다. 이 부품을 열등부품이라 한다.

$$\begin{aligned} (S_x(L-l_i), S_y(W-w_i)) &\leq (S_x(L-a(l_j)), \\ S_y(W-\beta(w_j))) \text{ and } g(i) &\leq a\beta g(j), \quad (5.1) \\ a \in H, H = \{a: 1 \leq a \leq L/l_R, a \text{는 양의 정수}\} \\ \beta \in V, V = \{b: 1 \leq b \leq W/w_R, b \text{는 양의 정수}\} \end{aligned}$$

6. 개선된 분지전략

G et al.[4]은 해법 수행중 불필요한 탐색을 줄이기 위해 세 가지의 분지전략을 제안하였다. 다음의 [전략 1]은 G et al.[4]의 분지전략에 열등패턴을 새로 정의한 [성질 2]를 추가로 적용하여 G et al.[4]의 분지전략을 개선한 것이다. 여기서 O 는 분지될 절단패턴의 집합을, C 는 분지된 절단패턴의 집합을 의미하고, 절단패턴 R 과 R' 의 크기는 각각 (l_R, w_R) , $(l_{R'}, w_{R'})$ 라고 가정한다.

[전략 1] 임의의 두 절단패턴 R 과 R' 에 대해 R 을 O 에서 C 로 옮길 때, C 에 식 (4.3)을 만족하는 절단패턴 R' 이 존재하면 R 을 제거한다.

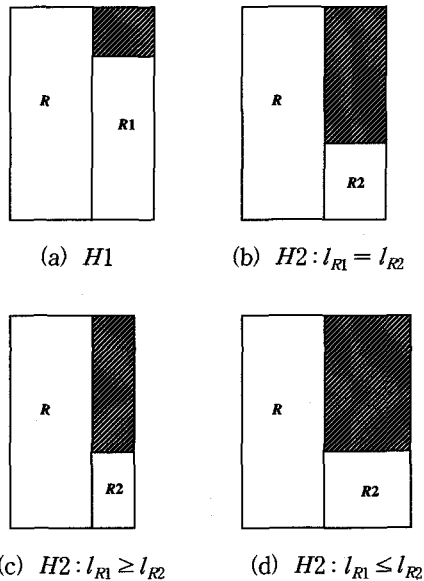
G et al.[4]의 [전략 1]은 R' 이 R 보다 크기가 작은 경우에만 적용되지만 개선된 분지전략 1은 R' 이 R 보다 작지 않은 경우에도 적용이 가능하다.

[전략 2] (1) 집합 O 에 포함되어 있는 절단패턴 R 에 대해 집합 C 에 속한 두 절단패턴 R_1 과 R_2 이 식 (6.1)을 만족하면 $H_2 = (R, R_2)^h$ 은 $H_1 = (R, R_1)^h$ 에 의해 제거된다. 수직결합에도 같은 방

법을 적용한다.

$$\begin{aligned} w_R \geq w_{R_1} \geq w_{R_2} \text{ and} \\ S_x(L-(l_R+l_{R_1})) = S_x(L-(l_R+l_{R_2})) \quad (6.1) \end{aligned}$$

개선된 [전략 2]의 (1)은 G et al.[4]의 것에서 $l_{R_1} = l_{R_2}$ 를 $S_x(L-(l_R+l_{R_1})) = S_x(L-(l_R+l_{R_2}))$ 로 수정한 것이다. G et al.[4]의 전략은 [그림 6.1] (b)와 같이 $l_{R_1} = l_{R_2}$ 인 경우에만 적용이 가능하지만, 개선된 전략은 [그림 6.1]의 (c)나 (d)와 같이 $l_{R_1} \neq l_{R_2}$ 이더라도 $S_x(L-(l_R+l_{R_1})) = S_x(L-(l_R+l_{R_2}))$ 을 만족하면 적용이 가능하다. 즉, $L-(l_R+l_{R_1})$ 과 $L-(l_R+l_{R_2})$ 을 채울 수 있는 최대 길이 조합이 같을 경우 적용이 가능하다.



[그림 6.1] 개선된 [전략 2]의 (1)

(2) 집합 O 에 포함되어 있는 절단패턴 R_2 에 대해 식 (6.2)를 만족하는 절단패턴 \bar{R} 가 C 에 존재하면 $H_2 = (R_2, R)^h$ 는 $\bar{H} = (\bar{R}, R)^h$ 에 의해 제거된다. 수직결합에도 같은 방법을 적용한다.

$$\begin{aligned} w_{\bar{R}} = \min\{w_{R_1} \mid S_x(L-(l_{R_1}+l_R)) \\ = S_x(L-(l_{R_2}+l_R)), w_{R_1} > w_{R_2}, R_1 \in C\} \quad (6.2) \end{aligned}$$

개선된 [전략 2]의 (2)는 G et al.[4]의 것에서 $l_{R1} = l_{R2}$ 를 $S_x(L - (l_{R1} + l_R)) = S_x(L - (l_{R2} + l_R))$ 로 수정한 것이다. G et al.[4]의 전략은 $l_{R1} = l_{R2}$ 인 경우에만 적용이 가능하나, 개선된 전략은 $l_{R1} \neq l_{R2}$ 이더라도 $S_x(L - (l_{R1} + l_R)) = S_x(L - (l_{R2} + l_R))$ 를 만족한다면 적용이 가능하다.

[전략 3] 임의의 절단패턴 q 을 집합 O 에 넣을 때, 식 (6.3)을 만족하는 절단패턴 R 이 O 에 존재한다면 q 를 제거한다.

$$(S_x(L - l_q), S_y(W - w_q)) \quad (6.3)$$

$$= (S_x(L - l_R), S_y(W - w_R)) \text{ and } g(q) \leq g(R)$$

개선된 [전략 3]은 G et al.[4]의 것에서 $(l_q, w_q) = (l_R, w_R)$ 를 $(S_x(L - l_q), S_y(W - w_q)) = (S_x(L - l_R), S_y(W - w_R))$ 로 수정한 것이다.

지금까지 설명한 개선된 분지전략과 열등부품을 제거하는 절차를 적용하여 해법을 정리하면 다음과 같다.

단계 1. [열등부품 제거]

크기가 각각 (l_i, w_i) 와 (l_j, w_j) 인 임의의 부품 i 와 j 가 식 (4.1)을 만족할 경우 부품 i 를 문제에서 제거한다.

단계 2. [초기화]

분지될 노드집합 O 는 $O = \{R_1, R_2, \dots, R_n\}$ 이다. 여기서 R_i 는 부품 i 한 개로 구성된 절단패턴을 나타낸다. 분지될 노드 집합 C 는 \emptyset 이고, 현재의 최적해 $z = 0$ 이다.

단계 3. [분지과정]

- (3.1) 집합 O 에서 가장 큰 상한을 갖는 절단패턴 R 을 찾는다. 만약 O 가 \emptyset 이면 단계 4로 간다.
- (3.2) 만약 [전략 1]에 의해 R 을 제거하면 단계 3을 반복한다.
- (3.3) 선택된 절단패턴 R 을 분지한다. 즉 R 을

O 에서 C 로 옮기고, 수평과 수직결합으로 모든 가능절단패턴들의 집합 Q 를 생성한다. $q \in Q$ 는 R 과 C (R 을 포함하는)안의 모든 절단패턴들과의 수평과 수직결합으로 생성되며, 생성된 패턴중 [전략 2]에 의해 열등한 패턴을 제거한다. q 의 크기 (l_q, w_q) 는 $(l_q, w_q) \leq (L, W)$ 를 만족한다. 즉, 절단패턴의 크기가 자재보다 큰 비가능(infeasible) 해는 제거한다.

- (3.4) [전략 3]에 의해 제거되지 않은 각 절단패턴 $q \in Q$ 에 대해서 q 의 가치 $g(q)$ 가 $g(q) > z$ 이면 z 를 $g(q)$ 로 갱신하고, q 의 상한 $f(q)$ 가 $f(q) > z$ 이면 q 를 집합 O 에 추가한다. 그렇지 않으면 q 를 제거한다. 단계 3을 반복한다.

단계 4. [종료]

최적해 z 을 리턴하고 종료한다.

7. 실험결과

본 장에서는 실험을 통해 제안한 해법의 우수함을 보인다. 제안하는 해법과 기존의 알려진 가장 우수한 해법인 G et al.[4]의 최적 해법으로 다양한 UTDC문제의 해를 구하여 비교하였다. 실험은 펜티엄IV 2GHz급 중앙처리장치(CPU)와 256MB의 주 기억장치(RAM)를 가진 개인용 컴퓨터에서 이루어졌다.

<표 7.1>은 Cung et al.[3]에 의해 소개된 UW1~UW11(가중 문제)과 UU1~UU11(비가중 문제)에 대한 결과이다([ftp://panoramix.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting](http://panoramix.univ-paris1.fr/pub/CERMSEM/hifi/2Dcutting)). 제안한 해법은 가중 문제에서 G et al.[4]의 최적 해법보다 수행시간을 평균 38% 줄였으며, 분지된 노드 수는 평균 28% 줄였다. 비가중 문제에서는 수행시간을 평균 18%, 분지된 노드 수를 평균 16% 줄였다. 특히 제안한 해법은 UU11을 제외한 모든 문제에서는 최적해를 0.3초 이내의 빠른 시간내에 구했다. 그러나

〈표 7.1〉 Cung et al.[3]이 소개한 문제에 대한 실험결과

No.	ID	(L, W)	최적해	수행시간(초)			분지된 노드 수(개)		
				G et al. [4]의 해법	제안하는 해법	개선효과 (%)	G et al. [4]의 해법	제안하는 해법	개선효과 (%)
1	UW1	(500,500)	6036	0.005	0.000	100.0	62	34	45.2
2	UW11	(555,632)	15747	0.015	0.010	33.3	136	66	51.5
3	UW2	(560,750)	8468	0.010	0.000	100.0	46	41	10.9
4	UW3	(700, 650)	5964	0.010	0.000	100.0	36	36	0.0
5	UW4	(1245,1015)	8326	0.015	0.015	0.0	35	25	28.6
6	UW5	(1100,1450)	7780	0.018	0.015	16.7	110	76	30.9
7	UW6	(1750,1542)	6615	0.039	0.031	20.5	133	68	48.9
8	UW7	(2250,1875)	10464	0.062	0.046	25.8	45	43	4.4
9	UW8	(2645,2763)	7692	0.093	0.085	8.6	26	25	3.8
10	UW9	(3000,3250)	7038	0.125	0.109	12.8	114	62	45.6
11	UW10	(3500,3650)	7507	0.171	0.156	8.8	311	194	37.6
12	UU1	(500,500)	242919	0.010	0.010	0.0	178	128	28.1
13	UU2	(750,800)	595288	0.015	0.015	0.0	135	134	0.7
14	UU3	(1100,1000)	1072764	0.030	0.030	0.0	290	232	20.0
15	UU4	(1000,1200)	1179050	0.058	0.039	32.8	696	513	26.3
16	UU5	(1450,1300)	1868999	0.046	0.039	15.2	424	397	6.4
17	UU6	(2050,1457)	2950760	0.048	0.041	14.6	183	150	18.0
18	UU7	(1465,2024)	2930654	0.125	0.094	24.8	1141	876	23.2
19	UU8	(2000,2000)	3959352	0.093	0.078	16.1	548	473	13.7
20	UU9	(2500,2460)	6100692	0.125	0.093	25.6	353	336	4.8
21	UU10	(3500,3450)	11955852	0.312	0.290	29.8	1162	838	27.9
22	UU11	(3500,3765)	13157811	435.562	315.625	27.5	158587	147644	6.9

〈표 7.2〉 가중 랜덤 문제에 대한 실험결과

NO.	(L, W)	수행시간(초)			분지된 노드 수(개)		
		G et al.[4]의 해법	제안하는 해법	개선 효과 (%)	G et al.[4]의 해법	제안하는 해법	개선 효과 (%)
1	(500,500)	0.005	0.003	32.6	8.1	6.1	24.0
2	(555,632)	0.006	0.005	12.9	16.5	9.7	41.3
3	(560,750)	0.009	0.006	33.3	21.9	11.2	48.8
4	(700,650)	0.009	0.006	32.3	19.5	13.6	30.1
5	(1245,1015)	0.053	0.038	29.4	98.5	63.6	35.5
6	(1100,1450)	0.038	0.031	16.8	128.0	80.6	37.0
7	(1750,1542)	0.292	0.164	43.9	126.8	96.5	23.9
8	(2250,1875)	0.109	0.120	7.1	201.9	168.1	16.7
9	(2645,2763)	0.181	0.172	5.2	343.6	250.5	27.1
10	(3000,3250)	0.192	0.188	2.4	444.0	345.1	22.3
11	(3500,3650)	0.467	0.442	5.4	686.3	531.5	22.6

〈표 7.3〉 비가중 랜덤 문제에 대한 실험결과

NO.	(L, W)	평균 수행시간(초)			평균 분지된 노드 수(개)		
		G et al.[4]의 해법	제안하는 해법	개선 효과 (%)	G et al.[4]의 해법	제안하는 해법	개선 효과 (%)
1	(500,500)	0.012	0.008	33.3	15.9	9.8	38.5
2	(750,800)	0.085	0.051	40.1	93.5	66.4	29.0
3	(1100,1000)	0.738	0.444	39.8	241.4	190.8	21.0
4	(1000,1200)	0.741	0.462	37.7	282.5	227.7	19.4
5	(1450,1300)	2.288	1.507	34.1	711.9	530.2	25.5
6	(2050,1457)	7.591	5.045	33.5	1292.7	1066.6	17.5
7	(1465,2024)	5.883	3.856	34.5	1634.0	1179.7	27.8
8	(2000,2000)	35.325	20.854	41.0	3365.8	2553.0	24.1
9	(2500,2460)	29.250	21.045	27.9	8605.4	6215.7	27.8
10	(3500,3450)	59.178	49.406	16.5	15549.2	13077.3	15.9
11	(3500,3765)	127.258	81.148	36.2	34301	27844.1	18.8

UU11의 경우에는 크기가 (37, 710)인 작은 부품과 자재에 비해 상대적으로 작은 크기의 부품들이 분지과정에 많이 포함되어 다른 문제에 비해 수행시간이 많이 걸렸다.

〈표 7.2〉는 랜덤하게 생성한 가중 문제에 대한 결과이다. 실험문제를 자재의 크기에 따라 11가지 유형으로 분류하였고, 유형별로 15개의 문제를 생성하였다. 유형별 자재의 크기는 UW1~UW11과 같다. 모든 유형에서 부품의 개수는 30개이며, 부품의 길이, 너비, 가치는 각각 $[200, L/2]$, $[200, W/2]$, $[200, 500]$ 에서 랜덤하게 선택되었다. 제안한 해법은 G et al.[4]의 최적해법보다 평균적으로 수행시간을 20%, 분지된 노드 수를 30% 줄였다.

〈표 7.3〉은 랜덤하게 생성한 비가중 문제에 대한 결과이다. 유형별 자재의 크기는 UU1~UU11와 같으며, 다른 조건은 랜덤 가중 문제의 실험 조건과 동일하다. 제안한 해법은 평균적으로 수행시간을 34%, 분지된 노드 수를 24% 줄였다.

8. 결론

본 연구에서는 무제한 2차원 절단(UTDC)문제에 대해 효율적인 최적해법을 제안하였다. 제안하는

해법은 기존의 가장 우수한 해법으로 알려진 G et al.[4]의 최적-우선 분지한계법에 개선된 분지전략을 적용하였다. 제안하는 해법에서는 먼저 패턴의 크기와 자재의 크기를 동시에 고려하여 기존 열등패턴의 개념을 확장하였고, 새로운 열등패턴의 정의를 이용하여 열등부품을 정의하였다. 그리고 열등부품을 문제에서 원칙적으로 제거하고, 열등패턴의 정의를 분지전략에 적용하여 분지과정에서 탐색되는 노드 수를 크게 줄였다.

실험을 통해 G et al.[4]의 해법과 비교한 결과, 제안하는 해법은 평균적으로 가중 문제에 대해 29%, 비가중 문제에 대해 26%의 수행시간 감소를 보였다. 본 연구에서 제안한 열등패턴의 개념과 분지전략은 배낭문제, 적재문제 등 여러 관련 문제에 적용이 가능하고 3차원 문제로의 확장도 가능할 것이다.

참고 문헌

- [1] Beasley, J.E., "Algorithms for Unconstrained Two-dimensional Guillotine Cutting," *Journal of the Operational Research Society*, Vol.36(1985), pp.297-306.

- [2] Christofides, N. and C. Whitlock, "An Algorithm for Two-dimensional Cutting Problems," *Operations Research*, Vol.26(1977), pp.30-44.
- [3] Cung, V.D., M. Hifi, and B.L. Cun, "Constrained Two-dimensional Cutting Stock Problems a Best-first Branch-and-bound Algorithm," *International Transactions in Operational Research*, Vol.7(2000), pp.185-210.
- [4] G, Y.G., Y.J. Seong, and M.K. Kang, "A Best-first Branch and Bound Algorithm for Unconstrained Two-dimensional Cutting Problems," *Operations Research Letters*, Vol.31(2003), pp.301-307.
- [5] Gilmore, P.C. and R.E. Gomory, "A Linear Programming approach to the Cutting-stock Problem," *Operations Research*, Vol.9(1961), pp.849-859.
- [6] Gilmore, P.C. and R.E. Gomory, "The Theory and Computation of Knapsack Functions," *Operations Research*, Vol.14(1969), pp.1045-1074.
- [7] Herz, J.C., "A Recursive Computational Procedure for Two-dimensional Stock Cutting," *IBM Journal of Research and Development*, Vol.16(1972), pp.462-469.
- [8] Hifi, M., "The DH/KD Algorithm : A Hybrid approach for Unconstrained Two-dimensional Cutting Problems," *European Journal of Operational Research*, Vol.97(1997), pp. 41-52.
- [9] Hifi, M. and V. Zissimopoulos, "A Recursive Exact Algorithm for Weighted Two-dimensional Cutting," *European Journal of Operational Research*, Vol.91(1996), pp.553-564.
- [10] Morabito, R.N., M.N. Arenales, and V.F. Arcaro, "An And-or-graph Approach for Two-dimensional Cutting Problems," *European Journal of Operational Research*, Vol.58(1992), pp.263-271.
- [11] Wang, P.Y., "Two Algorithms for Constrained Two-dimensional Cutting Stock Problems," *Operations Research*, Vol.31 (1983), pp.573-586.