Journal of the Society of Korea Industrial and Systems Engineering
Vol. 32, No. 1, pp.79－84, March 2009.

비가중 무제한 2차원 절단문제에 대한 최적-우선 분지한계 해법

윤기섭* · 윤희권** · 강맹규*†

*한양대학교 정보경영공학과
**CJ GLS(S) 인포텍

A Best-First Branch and Bound Algorithm for Unweighted Unconstrained Two-Dimensional Cutting Problems

Ki-Seop Yoon* · Hee-Kwon Yoon** · Maing-Kyu Kang*†

*Department of Information and Industrial Engineering, Hanyang University, Ansan Campus
**CJ GLS(S) Infotech Pte. Ltd.

In this paper, a best-first branch and bound algorithm based upon the bottom-up approach for the unweighted unconstrained two-dimensional cutting problem is proposed to find the optimal solution to the problem. The algorithm uses simple and effective methods to prevent constructing duplicated patterns and reduces the searching space by dividing the branched node set. It also uses a efficient bounding strategy to fathom the set of patterns. Computational results are compared with well-known exact algorithms and demonstrate the efficiency of the proposed algorithm.

Keywords : Two-Dimensional Cutting, Branch And Bound, Optimization

1. 서 론

2차원 절단문제는 하나의 큰 직사각형 자재(plate)로부터 가치(value)가 주어진 다양한 크기의 작은 직사각형 부품(piece)들을 절단할 때, 절단된 부품 가치의 합을 최대화하는 문제이다. 즉, 최대가치를 갖는 절단패턴(cutting pattern)을 구하는 문제이다. 절단패턴(cutting pattern)이란 하나의 자재로부터 절단되는 제품의 조합을 의미한다. 이 문제는 Gilmore와 Gomory[6]가 산업분야에 처음 적용하였으며, 유리, 철판, 섬유, 종이, 가구 등의 생산과 다중처리 시스템 및 다중프로그램 컴퓨터 시스템 등 매우 광범위하게 응용할 수 있다[7].

일반적으로 2차원 절단문제는 여러 기준에 따라 구분한다. 부품의 가치 결정 방법에 따라, 부품의 가치를 제품의 넓이로 정한 비가중(unweighted) 문제와 부품의 넓이와 관계없이 정한 가중(weighted) 문제로 구분한다. 절단되는 부품의 개수 제한 여부에 따라, 제한이 없는 무제한 2차원 절단(unconstrained two-dimensional cutting : UTDC)문제와 제한이 있는 제한(constrained) 2차원 절단문제로 구분한다. 또한, 자재의 절단방법에 따라, 절단이 일단 시작되면 도중에 멈추지 않고 끝까지 절단하는 길로틴(guillotine) 문제와 그렇지 않은 비길로틴(non-guillotine) 문제로 구분한다. 특별한 언급이 없으면 절단문제는 길로틴 제약을 고려한다. 본 연구에서는 비가중 UTDC 문제를 다룬다. 부품의 방향은 고정된 것으로 가정한다.

2차원 절단문제에 대해 Christofides and Whitlock[3]은 자재를 부품의 크기로 잘라나가는 하향식(top-down)

논문접수일 : 2008년 09월 19일　논문수정일 : 2009년 02월 01일　게재확정일 : 2009년 02월 05일
† 교신저자 ricelove@paran.com

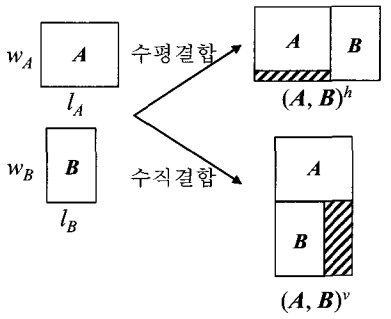
방법을 이용한 분지한계법을 제안하였고, Viswanathan and Bagchi[10]는 두 절단패턴을 수평 또는 수직으로 결합하여 나가는 상향식(bottom-up) 방법을 이용한 최적-우선 분지한계법(best-first branch and bound)을 제안하였다. Hifi and Zissimopoulos[9]는 상향식 방법을 바탕으로 하여 효율적인 상한(upper bound)과 하한(lower bound)을 이용한 동적 계획법(dynamic programming)을 제안하였다. Cung 등[4]는 Viswanathan and Bagchi[10]의 해법에 개선된 상한과 중복패턴(duplicated pattern) 제거 방법을 적용한 최적-우선 분지한계법을 제안하였다. G 등[5]는 띠(strip)을 이용한 상한과 개선된 분지전략(branching strategy)을 적용한 최적-우선 분지한계법을 제안하였다. 윤 등[2]는 G 등[5]의 해법에 열등패턴(dominated pattern)을 재정의하고 개선된 분지전략을 적용하였는데, 이 해법은 UTDC 문제에 대해 지금까지 알려진 가장 우수한 최적 해법이다.

본 논문은 먼저 상향식 접근법을 사용할 때 중복패턴을 효율적으로 방지하는 방법을 설명한다. 그리고 이를 이용하여 분지된 노드집합의 탐색범위를 축소시키는 방법을 제안하고 이 방법을 윤 등[2]의 해법에 적용한다. 또한 하나의 패턴이 아닌 패턴집합을 분지끝으로 판별할 수 있는 효율적인 한계전략(bounding strategy)을 적용하여 해법의 성능을 향상시킨다. 끝으로 실험을 통해 제안하는 해법이 기존의 해법보다 매우 빠른 시간 내에 최적해를 구할 수 있음을 보인다.

2. 최적-우선 분지한계법

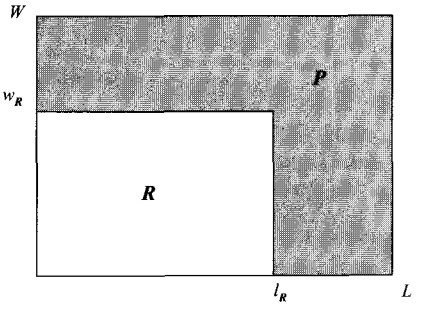
Viswanathan and Bagchi[10]는 UTDC에 대해 상향식 접근법을 이용한 최적-우선 분지한계법을 제시하였다. 상향식 접근법은 두 개의 절단패턴 A와 B를 수평결합(horizontal build)이나 수직결합(vertical build)을 이용하여 새로운 절단패턴을 생성하는 방법이다[11]. 최적-우선 분지한계법에 상향식 접근법을 이용할 경우, 각 노드는 하나의 절단패턴을, 분지는 수평결합이나 수직결합을 의미한다. 또한 자식노드는 부모노드와 분지된 노드들 간에 수평결합이나 수직결합으로 생성된 새로운 절단패턴을 의미한다. 상한이 가장 큰 노드를 우선적으로 분지하며 분지과정에서 더 이상 분지할 노드가 없으면 종료한다.

길이 x, 너비 y인 절단패턴의 크기를 (x, y)로, 절단패턴 A와 B의 수평결합과 수직결합을 각각 $(A, B)^h$와 $(A, B)^v$로 정의하자. 절단패턴 A와 B의 가치를 각각 $g(A)$와 $g(B)$, 길이와 너비를 각각 l_A와 w_A라 하면 A와 B의 수평결합$(A, B)^h$의 크기는$(l_A + l_B,\ \max(w_A, w_B))$이고 가치는 $g(A)+g(B)$가 된다. 같은 방법으로 A와 B의 수직결합 $(A, B)^v$는 크기가$(\max(l_A, l_B),\ w_A+w_B)$이고 가치가 $g(A)+g(B)$인 패턴을 생성한다(<그림 1> 참조).



<그림 1> 수평결합과 수직결합

수평결합이나 수직결합으로 생성된 임의의 절단패턴 R의 크기와 가치를 각각 (l_R, w_R), $g(R)$이라 하자. R의 상한을 $U(R)$이라 하면 $U(R)$을 구하기 위해 <그림 2>에 있는 영역 P의 상한 $u(P)$를 구해야 한다. $u(P)$를 구하게 되면 절단패턴 R의 상한은 $U(R)=g(R)+u(P)$가 된다. 만약 $U(R)$이 현재의 최적값보다 작으면 R은 더 이상 분지되지 않는다.



<그림 2> 절단패턴 R과 영역 P

상향식 접근법을 이용한 최적-우선 분지한계법의 절차를 정리하면 <표 1>과 같다[10].

3. 중복패턴

Cung 등[4]는 세 가지 종류의 중복패턴을 정의했는데, 그들이 정의한 중복패턴은 크기가 같고 동일한 부품조합을 갖는 패턴과 크기는 같고 가치는 작은 패턴을 포함한다. G 등[5]는 크기는 크거나 같고 가치는 작거나 같은 패턴을 열등패턴(dominated)으로 정의했다. 본 연구에서는 크기와 부품 조합이 동일한 패턴만을 중복패턴으로 정의한다.

<표 1> 최적-우선 분지한계법의 절차

단계 1. [초기화]
- 각각의 부품 하나만으로 구성된 절단패턴을 분지할 노드 집합에 추가한다.

단계 2. [분지과정]
- 분지할 노드 중에서 상한이 가장 큰 노드를 선택하고 이 노드를 분지된 노드 집합으로 옮긴다.
- 만약 더 이상 분지할 노드가 없으면 단계 3으로 간다.
- 선택된 노드를 분지하여 자식 노드를 생성한다.
- 생성된 자식 노드들 중에서, 현재의 최적 값보다 더 큰 상한을 갖는 노드만 분지할 노드 집합에 추가한다.

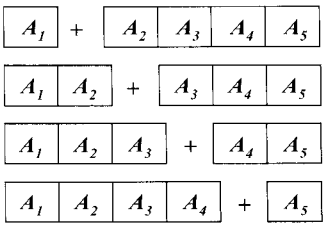
단계 3. [끝냄]
- 끝낸다.

상향식 접근법을 이용한 해법을 수행하는 도중 중복패턴이 많이 생성된다. 이러한 패턴들은 이후 분지과정이 모두 동일하다. 따라서 중복패턴들을 제거하거나 생성되지 않도록 하면 해법의 성능을 향상 시킬 수 있다. 그러나 이러한 중복패턴을 검사하는데 과도한 계산시간이 걸릴 경우 해법의 성능향상을 기대하기 어렵다. 본장에서는 중복패턴이 생성되는 새로운 두 가지 특수한 경우를 분류하고 중복패턴이 생성되지 않도록 하는 효율적인 방법을 설명한다. 이 방법은 제 4장에서 분지된 노드집합을 다중화할 때 적용된다.

• 경우 1 :

<그림 3>에 있는 네 개의 수평결합은 모두 동일한 패턴을 생성한다. 이런 경우 오른쪽의 패턴, 즉 결합하려는 패턴이 수평결합으로 생성된 패턴이 아닌 경우만 허용하면 처음 세 개의 중복패턴을 방지할 수 있다. 즉, R과 R'을 수평결합하여 $(R, R')^h$를 생성할 때, R'이 식 (1)을 만족하면 결합을 허용하지 않음으로써 중복을 방지한다.

$$R' = (A,\ B)^h \tag{1}$$



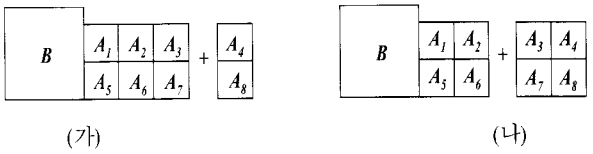
<그림 3> 중복패턴 생성(경우 1)

수직결합 할 때도 같은 방법을 적용하여 $(R, R')^v$를 생성할 때, R'이 식 (2)를 만족하면 결합을 허용하지 않는다.

$$R' = (A,\ B)^v \tag{2}$$

• 경우 2 :

<그림 4>의 (가)와 (나)의 수평결합 역시 동일한 패턴을 생성한다. 그러나 (나)의 경우 오른쪽의 패턴들이 수평결합이 아닌 수직결합으로 생성될 수 있다. 따라서 만약 (나)의 오른쪽 패턴이 $((A_3, A_7)^v, (A_4, A_8)^v)^h$이 아니라 $((A_3, A_4)^h, (A_7, A_8)^h)^v$라면 식 (1)을 이용하여 중복을 방지할 수 없다. 이러한 중복을 제거하려면 오른쪽에 오는 패턴이 단순하게 수평결합으로 생성된 패턴인지를 검사하는 대신에 수평으로 더 이상 나누어 질 수 없는 패턴인지를 검사해야 한다. 빠른 시간 내에 이를 검사하기 위해 결합하려는 패턴이 하나의 부품으로 이루어진 경우만 고려한다.



(가) (나)

<그림 4> 중복패턴 생성(경우 2)

임의의 부품 i의 크기와 가치를 각각 (l_i, w_i), $g(i)$라 하자. $(R, R')^h$를 생성할 때 R'이 식 (3)을 만족하면 결합을 허용하지 않는다. 식 (3)을 만족하는 패턴은 수직결합으로 생성되었더라도 수평으로 나누어질 수 있는 패턴이다. 식 (3)에서 α가 1인 경우 R'은 수직결합으로만 생성될 수 있으므로 α가 1인 경우는 제외한다. 식 (3)에서 $g(R') = \alpha\beta g(i)$는 R'이 i부품만으로 구성되도록 하기 위해 추가한 조건이다. 만약 R'이 i부품만으로 구성되지 않더라도 가치값이 $\alpha\beta g(i)$이면 i부품 조합으로 생성될 수 있으므로 $(R, R')^h$ 역시 중복패턴이 된다.

$$\begin{aligned} &R' = (\alpha l_i, \beta w_i) \text{ and } g(R') = \alpha\beta g(i) \\ &\alpha \in \{a : 1 < a \leqq L/l_i,\ a\text{는 양의 정수}\} \\ &\beta \in \{b : 1 \leqq b \leqq W/w_i,\ b\text{는 양의 정수}\} \end{aligned} \tag{3}$$

수직결합할 때도 같은 방법을 적용하여 $(R, R')^v$를 생성할 때, R'이 식 (4)를 만족하면 결합을 허용하지 않는다.

$$\begin{aligned} &R' = (\alpha l_i, \beta w_i) \text{ and } g(R') = \alpha\beta g(i) \\ &\alpha \in \{a : 1 \leqq a \leqq L/l_i,\ a\text{는 양의 정수}\} \\ &\beta \in \{b : 1 < b \leqq W/w_i,\ b\text{는 양의 정수}\} \end{aligned} \tag{4}$$

식 (3)과 식 (4)로부터 만약 R'이 식 (5)를 만족하면 R

과 R'의 수평결합과 수직결합 모두 허용하지 않는다.

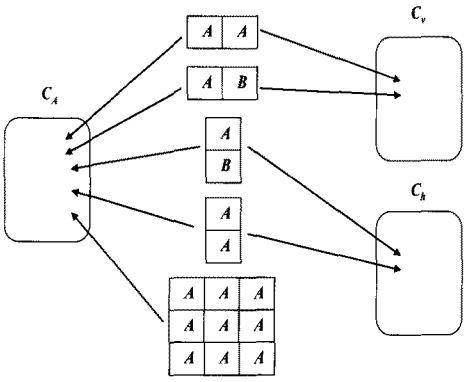
$$R' = (\alpha l_i,\ \beta w_i) \text{ and } g(R') = \alpha\beta g(i) \quad (5)$$
$$\alpha \in \{a : 1 < a \leq L/l_i,\ a\text{는 양의 정수}\}$$
$$\beta \in \{b : 1 < b \leq W/w_i,\ b\text{는 양의 정수}\}$$

4. 분지된 노드 집합 다중화

분지된 노드집합은 해법 수행 중 생성되는 최적 부분 패턴들을 저장한다. 이 패턴들은 패턴을 생성할 때 결합할 패턴으로 이용되고, 생성된 패턴이 열등패턴인지 검사할 때도 이용된다. 해법 수행 중 많은 패턴이 생성되므로 결합할 패턴을 선택하기 위해, 열등패턴인지 검사하기 위해 분지된 노드 집합을 많이 탐색해야 한다. 따라서 분지된 노드집합의 크기를 줄여 탐색범위를 축소시키면 성능을 향상시킬 수 있다. 본 장에서는 제 3장에서 설명한 중복패턴을 방지하는 방법을 적용하여 분지된 노드집합을 세 개로 다중화 하여 탐색범위를 축소시키는 방법을 설명한다.

분지할 패턴과 분지된 노드집합의 패턴을 결합할 때 제 3장에서 제안한 방법을 이용하면 중복패턴을 효율적으로 방지할 수 있다. 하지만, 분지된 노드집합의 크기를 줄이기 위해 이 방법을 결합할 때가 아니라 분지할 패턴을 분지된 노드집합으로 옮길 때 이용한다. 제 3장의 내용에 따르면 특정패턴을 수평결합에 이용될 수 없는 패턴, 수직결합에 이용될 수 없는 패턴, 두 가지 결합에 모두 이용될 수 없는 패턴으로 구분할 수 있다. 분지할 패턴을 분지된 노드집합으로 옮길 때 세 가지 중 하나로 분류한다. 그리고 분지된 노드집합을 수평결합용 집합(C_h), 수직결합용 집합(C_v), 전체집합(C_A) 세 개로 분리하고, 생성된 패턴을 해당하는 집합에 삽입한다. 만약 생성된 패턴이 수평결합에 이용될 수 없으면, 즉 패턴이 식 (1), 식 (3), 식 (5)중 하나를 만족하면 수직결합용 집합과 전체집합에 삽입한다. 생성된 패턴이 수직결합에 이용될 수 없으면, 즉 패턴이 식 (2), 식 (4), 식 (5)중 하나를 만족하면 수평결합용 집합과 전체집합에 삽입한다. 수평/수직 결합 모두에 이용될 수 없으면, 즉 식 (5)만 만족하면 전체집합에만 삽입한다. 문제에서 주어지는 부품의 경우 수평/수직결합을 모두 해야 하므로 세 개의 집합에 모두 삽입한다. <그림 5>는 이러한 내용을 나타내고 있다.

분지된 노드집합을 이와 같이 분리하면 새로운 패턴을 생성할 때 결합할 패턴을 탐색하기 위해 분지된 노드집합 전체를 탐색할 필요가 없다. 즉, 수평결합 시에는 C_h에서, 수직결합 시에는 C_v에서 결합할 패턴을 탐색하여 새로운 패턴을 생성한다. 이 방법을 이용하면 메모리 사용량이 증가하는 단점이 있지만 패턴을 생성할 때 분지된 노드 집합의 탐색 범위를 축소하여 탐색시간을 단축시킬 수 있다.



<그림 5> 분지된 노드집합 다중화

5. 한계전략

분지끝(fathoming)이란 어떤 특정 노드의 자식노드에서 최적해가 나올 가능성이 없다는 결론에 도달하여 더 이상 분지할 필요가 없음을 뜻한다. 한계(bounding)란 가질 수 있는 가장 좋은 값이란 의미이고, 한계를 가지고 분지끝을 판별하기 위한 전략을 한계전략(bounding strategy)이라 한다[1]. 본 장에서는 패턴의 낭비값을 이용하여 하나의 패턴이 아닌 패턴 집합에 대해 패턴 집합에 속하는 패턴들을 생성하기 이전에 분지끝을 판별하는 방법을 설명한다.

R과 R'이 결합하여 생성된 패턴을 q라 하자. Viswanathan and Baghi[10]는 q의 상한 $U(q)$와 현재까지의 최적해 $\hat{z}$가 식 (6)을 만족할 경우 q를 분지끝으로 판별하였다.

$$U(q) \leq \hat{z} \quad (6)$$

본 연구에서는 q의 분지끝을 판별하기 위해 q의 상한을 구하지 않고 R의 상한 $U(R)$과 패턴 R'의 낭비값 $W(R')$를 이용한 식 (7)을 이용한다.

$$U(R) - W(R') \leq \hat{z} \quad (7)$$

$U(R)$은 R이 포함된 상태에서의 예측값이다. 따라서 R' 자체의 낭비값 $W(R')$이 $U(R)$-$\hat{z}$보다 크면 q를 더 이상 분지할 필요가 없다. 식 (7)의 좌변은 R과 R'의 결합상

태를 고려하지 않고 R'의 낭비된 부분만을 추가로 고려하므로 식 (6)을 이용한 한계전략보다 더 많은 분지끝을 판별할 수는 없다. 그러나 이를 이용하면 하나의 패턴이 아닌 패턴집합에 대해 분지끝을 판별할 수 있다.

길이가 x인 분지된 노드 집합을 $C_{l=x}$라 하자. 만약 $\forall R' \in C_{l=x}$에 대해 식 (7)이 성립되면 집합 $\{(R, R')^h \mid R \in O, R' \in C_{l=x}\}$에 속한 모든 패턴을 분지끝으로 판별할 수 있다. 이 집합에 속하는 패턴들을 생성하기 이전에 이 집합에 대해 분지끝을 판별하기 위해 $\min\{W(R') \mid R' \in C_{l=x}\}$을 이용한다. 그리고 만약 R이 식 (8)을 만족하면 집합 $\{(R, R')^h \mid R \in O, R' \in C_{l=x}\}$에 속한 모든 패턴을 분지끝으로 판별한다.

$$U(R) - \min\{W(R') \mid R' \in C_{l=x}\} \leq \hat{z} \tag{8}$$

식 (8)을 이용하여 추가 연산 없이 패턴 집합을 분지끝으로 판별하기 위해 새로운 패턴을 생성할 때 매번 모든 x에 대해 $\min\{W(R') \mid R' \in C_{l=x}\}$를 갱신하여 저장한다. 수직결합인 경우도 같은 방법을 적용하여 식 (9)를 이용해 패턴집합 $\{(R, R')^v \mid R \in O, R' \in C_{w=y}\}$의 분지끝을 판별한다.

$$U(R) - \min\{W(R') \mid R' \in C_{w=y}\} \leq \hat{z} \tag{9}$$

6. 실험결과

본 장에서는 실험을 통해 제안한 해법의 우수함을 보인다. 제안하는 해법과 기존의 알려진 가장 우수한 해법인 윤 등[2]의 최적 해법으로 다양한 UTDC 문제의 해를 구하여 비교하였다. 실험은 인텔 코어 2 듀오 3GHz 중앙처리장치(CPU)와 2GB의 주기억장치(RAM)를 가진 개인용 컴퓨터에서 이루어졌다.

실험에 사용한 문제는 세 가지의 문제군으로 분류된다. 문제군 1은 Cung 등[4]가 소개한 11개의 비가중 문제이다. 이 문제들은 Hifi[8]에서 구할 수 있다. 문제군 2와 문제군 3은 임의로 생성한 비가중 문제인데, 문제군 2의 실험조건은 G 등[5]와 윤 등[2]의 랜덤 실험조건과 동일하며, 문제군 3은 문제군 2와 비교해 부품의 크기만 다르다. 문제군 2와 문제군 3은 문제를 자재의 크기에 따라 11가지 유형으로 분류하고, 유형별로 10개의 문제를 포함한다. 유형별 자재의 크기는 <표 2>의 문제군 1과 같다. 모든 유형에서 부품의 개수는 30개이다. 문제군 2에서 부품의 길이, 너비는 각각 [200, L/2], [200, W/2]에서 임의로 선택하고, 문제군 3에서는 [100, L/4], [100, W/4]에서 임의로 선택한다.

절단문제는 자재크기, 부품 종류수, 부품의 크기에 따라 고려해야할 조합의 크기가 달라진다. 문제군 3은 문제군 2와 모든 조건이 동일하고 부품 크기만 더 작은 범위에서 선택하기 때문에 문제군 2보다 더 많은 조합을 고려해야 한다. 따라서 문제군 3은 문제군 2보다 문제의 크기가 크다고 할 수 있다. 문제군 3의 부품 크기를 문제군 2에 비해 작게 하여 문제의 크기를 크게 한 이유는 문제의 크기가 작을 경우 해법의 수행시간이 너무 짧아 성능평가가 어렵기 때문이다.

<표 2>는 문제군 1에 대한 결과이다. 문제군 1에서 UU11을 제외한 나머지 문제에서는 두 해법에서 모두 수행시간이 0.1초 미만으로 매우 짧고, 감소효과를 관찰할 수 없었다. UU11을 제외한 문제는 문제의 크기가 크지 않다. UU1을 예로 들면, UU1은 자재의 크기가 (500, 500)인데 비해 부품의 크기는 [105, 319]의 범위내에 있고 면적이 가장 작은 부품의 크기는 (131, 168)로 고려해야 할 조합이 많지 않다. 그에 반해 UU11은 자재 크기 (3500, 3765)에 비해 매우 작은 (37, 710)의 크기인 부품이 포함된 문제로서 그만큼 고려해야 하는 조합이 많기 때문에 문제의 크기가 다른 문제들에 비해 크다. 제안한 해법은 이 문제에서 31.25초가 걸려 80.83초가 걸린 윤 등[2]의 해법보다 수행시간을 61.3% 감소시켜 큰 효과를 나타냈다.

<표 3>과 <표 4>는 문제군 2와 문제군 3에 대한 결과이다. 문제군 2에서 제안한 해법은 윤 등[2]의 최적해법보다 평균적으로 수행시간을 약 17%, 최대 약 45%를 감소시켰다. 문제군 3에서 제안한 해법은 윤 등[2]의 최적해법보다 평균적으로 수행시간을 약 59% 감소시켰다.

<표 2> 문제군 1에 대한 실험결과

No.	ID	(L, W)	부품 수	최적해	수행시간(초)		개선 효과 (%)
					윤등[2]의 해법	제안 해법	
1	UU1	(500,500)	25	242919	0.000	0.000	-
2	UU2	(750,800)	30	595288	0.000	0.000	-
3	UU3	(1100,1000)	25	1072764	0.000	0.000	-
4	UU4	(1000,1200)	38	1179050	0.000	0.000	-
5	UU5	(1450,1300)	50	1868999	0.016	0.016	0.0
6	UU6	(2050,1457)	38	2950760	0.016	0.016	0.0
7	UU7	(1465,2024)	50	2930654	0.031	0.031	0.0
8	UU8	(2000,2000)	55	3959352	0.016	0.016	0.0
9	UU9	(2500,2460)	60	6100692	0.031	0.031	0.0
10	UU10	(3500,3450)	55	11955852	0.062	0.062	0.0
11	UU11	(3500,3765)	25	13157811	80.828	31.250	61.3

제안하는 해법은 문제의 크기가 큰 문제군 3에서 문제군 2보다 더 효과가 잘 나타났다.

<표 3> 문제군 2에 대한 실험결과

No.	(L, W)	수행시간(초)		개선 효과 (%)
		윤등[2]의 해법	제안해법	
1	(500,500)	0.000	0.000	-
2	(750,800)	0.003	0.003	0.0
3	(1100,1000)	0.017	0.017	0.0
4	(1000,1200)	0.016	0.014	12.5
5	(1450,1300)	0.045	0.042	6.6
6	(2050,1457)	0.131	0.103	21.3
7	(1465,2024)	0.104	0.087	16.3
8	(2000,2000)	0.237	0.181	23.6
9	(2500,2460)	0.876	0.607	30.7
10	(3500,3450)	3.778	2.090	44.6
11	(3500,3765)	3.222	1.889	41.3

<표 4> 문제군 3에 대한 실험결과

No.	(L, W)	수행시간(초)		개선 효과 (%)
		윤등[2]의 해법	제안해법	
1	(500,500)	0.399	0.030	92.6
2	(750,800)	0.469	0.258	44.9
3	(1100,1000)	1.433	0.750	47.6
4	(1000,1200)	1.464	0.772	47.2
5	(1450,1300)	6.689	3.108	53.5
6	(2050,1457)	24.392	10.614	56.4
7	(1465,2024)	13.574	5.951	56.1
8	(2000,2000)	46.499	17.936	61.4
9	(2500,2460)	75.277	30.619	59.3
10	(3500,3450)	298.642	100.186	66.4
11	(3500,3765)	314.280	110.417	64.8

7. 결 론

본 연구에서는 무제한 2차원 절단문제에 대해 효율적인 최적해법을 제안하였다. 제안한 해법은 중복패턴을 방지하는 효율적인 방법을 제시하고, 이를 이용하여 분지된 노드 집합을 다중화하여 탐색 범위를 축소시켰다. 또한 하나의 패턴이 아닌 패턴집합을 분지끝으로 판별할 수 있는 한계전략을 제안하여 해법의 성능을 높였다.

실험을 통해 윤 등[2]의 해법과 비교한 결과, 제안하는 해법은 Cung 등[4]의 문제 중 문제 크기가 매우 큰 UU11문제에서 약 61% 정도 수행시간을 감소시켰다. 임의로 생성한 두 개의 문제군에 대해 제안하는 해법은 문제군 2에서는 약 17%, 문제군 3에서는 약 59%의 수행시간을 감소시켰다. 문제군 3이 문제군 2에 비해 문제의 크기가 크다는 것을 고려하면 세 개의 문제군에 대한 실험 결과 제안하는 해법은 문제의 크기가 클수록 더 효과가 잘 나타났다.

참고문헌

[1] 김종수; 유비쿼터스 경영과학, 박영사, 145-146, 2005.

[2] 윤기섭, 방성규, 강맹규; "무제한 2차원 절단문제에 대해 개선된 최적-우선 분지한계 해법", 한국경영과학회지, 30(4) : 61-70, 2005.

[3] Christofides, N. and Whitlock, C.; "An algorithm for two-dimensional cutting problems," *Operations Research*, 26 : 30-44, 1977.

[4] Cung, V. D., Hifi, M., and Cun, B. L.; "Constrained two-dimensional cutting stock problems a best-first branch-and-bound algorithm," *International Transactions in Operational Research*, 7 : 185-210, 2000.

[5] G., Y. G., Seong, Y. J., and Kang, M. K.; "A best-first branch and bound algorithm for unconstrained two-dimensional cutting problems," *Journal of the Operational Research Society*, 53 : 587-591, 2002.

[6] Gilmore, P. C. and Gomory, R. E.; "A linear programming approach to the cutting-stock problem," *Operations Research*, 9 : 849-859, 1961.

[7] Hifi, M.; "The DH/KD algorithm : a hybrid approach for unconstrained two-dimensional cutting problems," *European Journal of Operational Research*, 97 : 41-52, 1997.

[8] Hifi, M.; Retrieved July 30, 2008, http://www.laria.u-picardie.fr/hifi/OR-Benchmark/2Dcutting/.

[9] Hifi, M. and Zissimopoulos, V.; "A recursive exact algorithm for weighted two-dimensional cutting," *European Journal of Operational Research*, 91 : 553-564, 1996.

[10] Viswanathan, K. V. and Bagchi, A.; "Best-first search methods for constrained two-dimensional cutting stock problems," *Operations Research*, 41 : 768-776, 1993.

[11] Wang, P. Y.; "Two algorithms for constrained two-dimensional cutting stock problems," *Operations Research*, 31 : 573-586, 1983.