韓國經營科學會誌
第34권 第3號
2009年 9月

계단 구조를 이용한 팔레트적재문제의 새로운 해법

지영근* · 진고환**†

A New Exact Algorithm Using the Stair Structure for the Pallet Loading Problem

Yeonggeun Ji* · Gowhan Jin**

■ Abstract ■

The pallet loading problem(PLP) requires the best orthogonal layout that loads the maximum number of identical boxes(small rectangle) onto a pallet(large rectangle). Since the high pallet utilization saves the distribution and storage costs, many heuristic and exact algorithms have been developed so far. Martins and Dell have found the optimal layouts for the all PLPs less than or equal to 100 boxes except for only 5 problems in their recent research. This paper defines the 'stair structure' and proposes a new exact algorithm applying it. In order to show the priority of the proposed algorithm, computational results are compared to previous algorithms and the optimal layouts for the 5 unsolved problems are given.

Keyword : Pallet Loading Problem, Branch and Bound

1. 서 론

팔레트적재문제 (L, W, l, w)는 크기가 (L, W)로 주어진 팔레트 위에 크기가 (l, w)로 주어진 작은 직사각형의 상자를 가장 많이 실을 수 있는 최적의 적재 패턴을 찾는 문제이다. 실제 생산 현장 또는 물류 현장에서는 대량 생산된 제품을 보관 또는 운송할 때 팔레트에 상자를 쌓아 포장하고, 다시 이

논문접수일 : 2008년 04월 04일 논문게재확정일 : 2009년 07월 11일
논문수정일(1차 : 2009년 05월 05일)
* 포디엄시스템 기술연구소
** 우송대학교 IT경영정보학과
† 교신저자

팔레트들을 화물차, 철도 화차 또는 컨테이너 등의 적재함에 쌓아 운송한다. 대부분의 물류 현장에서 제품 상자는 운송 중 제품의 손상 및 변형을 방지하기 위해서 반드시 바로 세워야 하는(this side up) 경우가 많으며, 또한 한 팔레트 위에는 단일 제품을 적재한다[7, 13]. 따라서 팔레트 적재시 각 층의 높이는 제품 상자의 높이와 일치하며, 한 층의 적재 패턴을 찾기 위해서 오직 상자의 가로길이(l)와 세로길이(w)만을 고려하면 된다. 일반적으로 한 층의 적재 패턴에 포함된 상자의 최대 개수 n을 팔레트 적재문제의 크기(problem size)로 정의한다.

팔레트의 적재 효율은 제품의 운송 및 보관과 관계된 물류비용에 직접적인 영향을 미치기 때문에 이 문제를 위한 매우 많은 연구가 지금까지 진행되어 왔다. Martins and Dell[10]은 최대 적재 개수가 100개 이내의 모든 팔레트적재문제들에 대한 3,080,730개의 최소 크기의 동치문제(minimum size instances of equivalence class)를 구했는데, 이것은 크기가 100개 이내인 임의의 팔레트적재문제가 주어졌을 때, 이 문제를 이들 동치문제 중 하나로 변환할 수 있다는 것을 의미한다. 따라서 3,080,730개의 동치문제들 모두에 대해 최적해를 구하면, 크기가 100개 이내인 모든 팔레트적재문제에 대한 최적해를 구할 수 있다.

팔레트적재문제를 풀기 위한 최근의 연구들 중 가장 대표적인 발견적 해법으로는 G4 또는 G5-구조(G5-structure)를 기반으로 한 해법[14]과 5 블록을 초과하는 블록구조를 고려한 해법(HONG : Higher-Order Non-Guillotine)[12], 그리고 G Y.-G.와 M.-K. Kang[7]의 대용량 컨테이너 적재를 위한 발견적 해법들이 있다. 이들 해법들은 비교적 빠른 시간 안에 대부분의 팔레트적재문제들에 대한 최적의 적재패턴들을 구해주지만, 항상 최적해를 보장하지는 않는다. Martins and Dell[11]에 따르면, G5 해법을 사용했을 때 7006개의 동치문제를 제외한 모든 문제들에 대해 최적해를 구할 수 있음을 보였고, HONG 해법을 사용했을 경우 남은 7006개의 문제 중 오직 54개의 문제에 대해서만 추가로 최적해를 구할 수 있었다.

팔레트적재문제는 여러 연구에서 NP complete 문제로 인식되고 있으며, 이러한 점 때문에 아직 다항식 계산량을 갖는 최적해법이 발견되지 못하고 있다[14]. 지금까지 발표된 팔레트적재문제를 위한 최적 해법으로는 그래프 이론(graph theory)을 적용한 Dowsland [6]의 해법, MBFS(Maximal Breadth Filling Sequence) 기법의 Bhattacharya et al.[3]의 해법(이하 MBFS 해법), 그리고 HVZ 코딩(coding) 기법을 사용한 Martins and Dell[11]의 해법(이하 HVZ 해법)이 있다. 하지만 이들 해법은 분지한계법(branch and bound method)를 사용하고 있기 때문에, 최악의 경우 문제의 크기 n에 지수적으로 비례하는 계산량을 가진다. Martins and Dell[11]은 가장 최근 연구에서 발견적 해법으로 최적해를 구하지 못 한 6952개의 동치 문제들에 대해 HVZ 해법을 적용하여 다음 5문제 (74, 46, 7, 5), (86, 52, 9, 5), (95, 92, 11, 8), (172, 66, 19, 7), (178, 60, 16, 7)를 제외한 남은 모든 문제에 대해 최적해를 구하였다.

본 논문에서는 적재 패턴의 계단 구조(stair structure)를 이용한 새로운 최적 우선 분지한계법(best-first branch and bound method)을 제안하고 남은 5개 문제의 최적해를 구한다. 또한 다양한 수리적 예제를 통해 기존 최적해법과의 비교하여, 새로운 해법의 우수성 및 최적해의 도출을 입증한다.

2. 계단 구조(The stair structure)

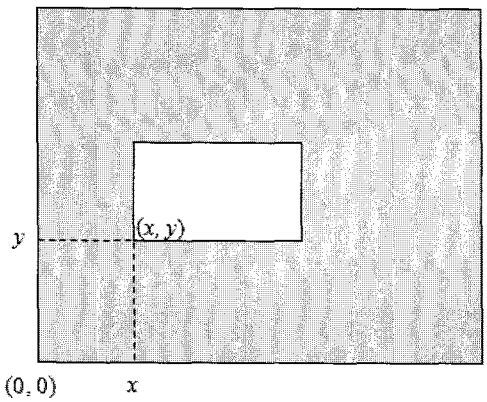
팔레트적재문제 (L, W, l, w)에 대해 문제의 일반성을 해치지 않고 다음 부등식 (1)을 항상 만족한다고 가정할 수 있다.

$$L \geq W \geq l \geq w \tag{1}$$

제품 상자를 팔레트 위에 오직 (l, w)와 (w, l)의 2가지 방향으로 놓을 수 있으므로, 가로로 놓은(가로길이가 세로길이보다 큰 적재 방향)(l, w)를 H-box, 세로로 놓은 (w, l)을 V-box라 하자. 본 논문에서는 [그림 1]과 같이 팔레트 좌측 하단 모서리를

(0, 0)으로하는 x-y 좌표계를 사용하여 상자의 좌측 하단 모서리의 위치 (x, y)를 상자의 위치로 정의한다.

한 패턴 안의 모든 상자가 더 이상 왼쪽 방향 또는 아래쪽 방향으로 이동할 수 없는 경우, 이 패턴을 정규화 패턴(normalized pattern)이라 한다. 임의의 최적 패턴은 정규화 될 수 있기 때문에 정규화 패턴만 고려하여도 최적 패턴을 찾을 수 있다.



[그림 1] 상자의 위치

계단 구조를 정의하기 위해 [그림 2]의 a와 같이 (0, 0)에 위치한 H-box를 고려하자. 여기에 새로운 V-box를 정규화 패턴으로 추가할 수 있는 후보지는 $(0, w)$와 $(l, 0)$이다. [그림 2]의 b와 같이 V-box가 $(0, w)$에 위치한다면 다시 새로운 상자가 추가될 수 있는 후보지는 각각 $(0, w+l)$, (w, w), $(l, 0)$이다. 이와 같이 한 적재 패턴 내의 m개의 후보지가 있고, 이를 가장 왼쪽에 위치한 후보지부터 각각 (x_1, y_1), (x_2, y_2), $\cdots$ (x_m, y_m)이라 하면, $x_1 \leqq x_2 \leqq \cdots \leqq x_m$이다. 만약 다음 조건 (2)를 만족한다면 이 적재패턴을 계단 구조(stair structure)라 정의한다.

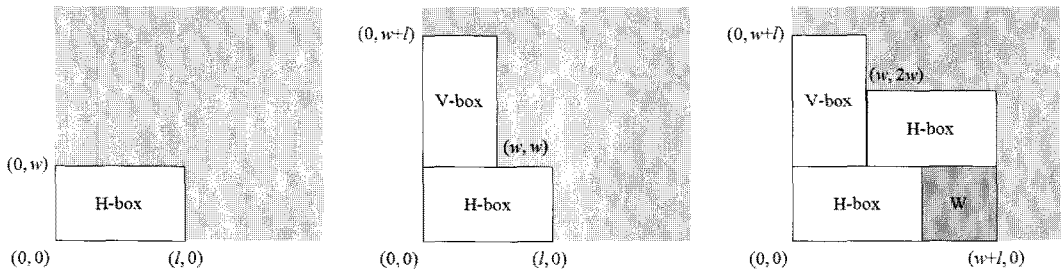
$$x_1 < x_2 < \cdots < x_m \quad \text{and} \qquad (2)$$
$$y_1 > y_2 > \cdots > y_m$$

[그림 2]의 a와 b 패턴은 모두 조건 (2)를 만족하므로 계단 구조이다. 여기서 [그림 2]의 b패턴에서 후보지 (w, w)에 H-box를 추가하는 경우, [그림 2]의 c패턴과 같이 후보지 $(l, 0)$을 제거하고 대신 $(w+l, 0)$을 추가하여 계단 구조를 유지할 수 있다. 즉, 영역 W는 빈 공간(wasted area)이 되는 것이다. 이를 일반화하기 위해 임의의 계단 구조 S의 후보지 집합 $C(S) = \{(x_i, y_i) \mid i=1, 2, \cdots, m\}$를 정의하고, k번째 후보지 (x_k, y_k)에 크기 (l, w)의 상자를 추가한다고 하자. 그러면 상자가 추가된 새로운 계단 구조 T의 후보지 집합 C(T)는 다음과 같이 정의할 수 있다.

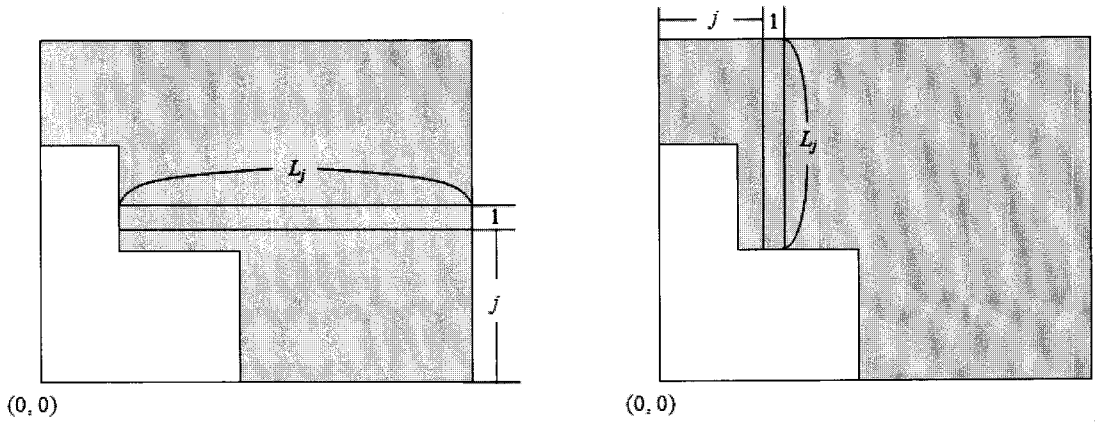
$\mathbf{A} = \{(x_i, y_i) \mid x_k \leqq x_i \leqq x_k + l \text{ or } y_k \leqq y_i \leqq y_k + w,\ i=1, 2, \cdots, m\}$이고,

$a = \min\{x \mid (x, y) \in \mathbf{A}\}$, $b = \min\{y \mid (x, y) \in \mathbf{A}\}$일 때,

$\mathbf{C}(T) = (\mathbf{C}(S) - \mathbf{A}) \cup \{(a, y_k + w), (x_k + l, b)\}$.



[그림 2] 계단 구조



[그림 3] 계단형 구조의 상한 구하기

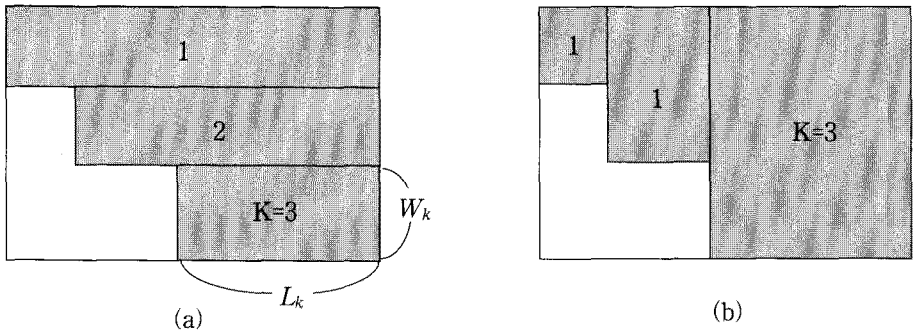
본 논문에서 제안하는 분지전략(branching strategy)을 설명하기 위해 임의의 계단 구조를 탐색 노드 S라 하자. 그러면 노드 S의 자식노드(child node)로 S의 첫 번째 후보지 (x_1, y_1)에 H-box를 추가한 경우와 V-box를 추가한 경우에 대한 2개의 노드를 구성한다. 그런데 새로 추가한 H-box의 우측모서리 x_1+l이 x_2보다 크다면 새로운 자식노드의 후보지 집합에서는 계단형 구조를 유지하기 위해 부모노드의 후보지 (x_2, y_2)가 제외되므로, (x_2, y_2)에 상자를 추가할 수 있는 기회를 상실한다. 따라서, 부모노드의 후보지 (x_1, y_1)에 상자를 싣지 않고 후보지 (x_2, y_2)에 H-box와 V-box를 각각 추가한 경우에 대한 2개 노드를 자식 노드로 추가함으로써 상실된 기회를 탐색에 포함시킬 수 있다. 이상의 과정을 일반화하기 위해 $x_i+l \leq x_{i+1}$를 만족할 때까지 다음 후보지들에도 반복 적용한다.

본 논문에서 제안하는 분지한계법에서는 팔레트에 아무 상자도 싣지 않은 상태의 후보지 집합 {(0, 0)}을 갖는 초기노드로부터, 앞에서 설명한 자식노드 생성 전략을 적용하여 탐색을 시도한다. 각 잎노드(leaf node)의 상한을 계산하고, 가장 높은 상한값을 갖는 노드를 먼저 탐색하는 최적 우선(best-first) 탐색 전략을 사용한다.

3. 상한(upper bound)과 하한(lower bound)

지금까지 팔레트적재문제의 상한(upper bound)에 대한 많은 연구가 있었다[2, 5, 9, 13]. 이들 연구에 대해서는 가장 최근의 Letchford and Amaral[9]의 연구와 Martins and Dell[11]의 연구에 잘 정리 되어 있다. 그러나 이들 논문에서 사용하는 상한들 중 남은 팔레트 면적을 상자의 면적으로 나누어 정수화하는 면적비상한(area ratio bound)만이 본 논문에서 고려하는 계단 구조의 상한으로 직접 사용될 수 있다. 선형계획법 기반(LP based)의 상한들 역시 약간의 변형을 통해 계단 구조의 상한으로 사용될 수 있으나, 이 들 상한은 계산시간이 매우 크다는 점 때문에 많은 회수에 걸쳐 상한을 구해야만 하는 분지한계법에 적용하기는 어렵다.

따라서, 본 논문에서는 빠른 계산 시간을 가지면서도 면적비상한보다 우수한 상한을 구해주는 G Y.-G.와 M.-K. Kang[8]의 상한을 사용한다. 구체적으로 적재공간(팔레트)를 높이(또는 길이) 1의 띠조각들로 나누고, 각 띠조각들의 길이에 대해 각 제품(상자)들의 가로와 세로 길이들을 최대로 적재하는 배낭문제(knapsack problem)를 풀고, 이 값들의 합을 상한으로 한다.



[그림 4] 계단 구조의 하한 구하기

G Y.-G.와 M.-K. Kang[8]의 상한을 본 논문의 상한에 적용하기 위해 [그림 3]의 a와 같이 임의의 계단 구조에 대해 남은 팔레트 영역을 높이 1의 띠조각들로 나눈다. 띠조각 j를 높이 j에 위치한 띠조각이라 하고 L_j를 띠조각 j의 길이라고 하자. 임의의 띠조각 j에 대한 최대 가능 적재량은 다음 식의 H_j와 같다.

$$H_j = \max\{ml+nw \mid ml+nw \leq L_j,\ m\text{과 } n\text{은 비음의 정수}\}$$

사실 H_j는 길이 L_j에 대한 완전분할(perfect partition)에 의한 적재량을 의미한다[4]. 이것은 임의의 적재패턴의 띠조각 j에 대한 적재량이 H_j를 초과할 수 없다는 것을 의미하므로, 다음 H는 계단 구조의 상한이다.

$$H = \left\lfloor \frac{\sum_{j=0}^{W-1} H_j}{lw} \right\rfloor$$

[그림 3]의 b와 같이 남은 팔레트 영역을 길이 1의 띠조각으로 나누는 경우에 대해서도 이와 유사한 방식으로 다음 상한 V를 구할 수 있다.

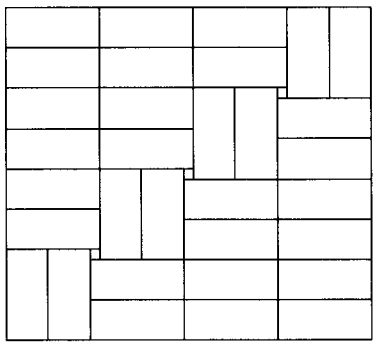
$$V = \left\lfloor \frac{\sum_{j=0}^{L-1} V_j}{lw} \right\rfloor$$

따라서 H와 V 중 최소값은 임의의 계단 구조에 대한 상한이다.

임의의 계단 구조의 하한을 구하기 위해 [그림 4]의 a 또는 b와 같이 남은 영역을 가로 또는 세로 방향으로 K(= 3)개의 직사각형으로 분할하는 경우를 생각해 보자. 각 직사각형에 대해 적은 계산량을 갖는 발견적 해법을 적용하여 효율적인 적재 패턴을 구하여 누적하면 효과적으로 하한을 구할 수 있다.

서론에서 이미 언급한 바와 같이 팔레트적재문제를 위한 많은 발견적 해법들이 개발되어 왔다. 그러나 분지한계법에서는 노드를 탐색할 때마다 해당 노드의 계단 구조에 대해 발견적 해법을 적용해야 하므로 빠르고 효율적인 해법을 사용해야 한다.

따라서 본 논문에서는 Martins and Dell[11]이 제안한 공동블록해법(hollow block heuristic)을 사용한다. 이 해법은 단지 두 쌍의 완전분할의 조합만을 탐색하여 [그림 5]와 같은 대각선블록(diagonal element block), 이를 제외한 메인블록(mail element block)의 적재 패턴을 적은 계산량으로 탐색하는 효율적인 해법이다. 즉, [그림 4]의 a에서 k 번째 직사각형을 고려할 때 HL_k는 길이 L_k에 대한 완전분할(perfect partition)에 의한 적재량이고, 완전분할의 모든 조합집합을 $P(HL_k, l, w)$라 하고, 이에 포함되는 원소를 (i, j)라 하면(i, j는 비음의 정수), $(i, j) \in P(HL_k, l, w)$이다. 또한, W_k에 대한 완전분할의 적재량을 HW_k, 완전분할의 모든 조합집합을 $P(HW_k, l, w)$, 포함되는 원소를 (f, g)라 하면(f, g는 비음의 정수),



[그림 5] 공동블록구조(hollow block structure)

$(f, g) \in P(HW_k, l, w)$이다. Martins and Dell[11]에 의하면 $i^*a \le j^*b$이고 $g^*b \le f^*a$이면 대각선 블록은 H-box와 메인블록은 V-box로 구성되며, 이때의 하한 R_k 은 다음과 같이 주어진다.

$$R_k = \max\{d^*(i^*g + j^*f/(d-1))\},$$
$$\text{for } d = 2, 3, \cdots, GCD(j, f)+1$$

여기서 $CDG(j, f)$는 j와 f의 최대 공배수이다. 이에 반해 $j^*b \le i^*a$이고 $f^*a \le g^*b$이면 대각선 블록과 메인블록은 각각 V-box와 H-box로 구성되고, 이때의 하한은 아래와 같다.

$$R_k = \max\{d^*(f^*j + i^*g/(d-1))\},$$
$$\text{for } d = 2, 3, \cdots, GCD(i, g)+1$$

K개의 직사각형에서 구한 하한들을 합하면 임의의 계단구조에 대한 하한이 된다.

4. 중복 탐색의 제거

본 연구에서는 탐색의 효율을 높이기 위해 두 가지 중복 탐색 제거 전략을 사용한다. 하나는 우위패턴(dominant pattern)에 의한 방법이고, 다른 하나는 적재패턴의 대칭성을 이용한 방법이다. 먼저 우위패턴에 의한 중복 탐색의 제거를 설명하기 위해 N(S), A(S)를 각각 계단 구조 S에 대한 적재된 상자의 개수와 적재를 위해 사용된 영역이라 정의하자. 그리고 서로 다른 계단 구조 S와 T가 있다고 가정하자. 만약 S와 T 사이에 다음 조건 (3)과 조건 (4)가 만족된다면 S를 T에 대한 우위패턴(dominant pattern)이라 한다.

$$N(\mathrm{S}) \ge N(\mathrm{T}), \tag{3}$$
$$\mathrm{A}(\mathrm{S}) \subset \mathrm{A}(\mathrm{T}). \tag{4}$$

조건 (3)은 계단 구조 S에 적재된 상자의 개수가 T에 적재된 상자의 개수보다 많거나 같다는 것을 의미하고, 조건 (4)는 S의 적재영역이 T의 적재영역의 부분집합임을 의미한다. 즉, T의 적재영역에 S의 적재영역이 완전히 포함됨을 의미한다. 이것은 T의 남은영역의 임의의 적재패턴이 S의 남은영역에 동일하게 적재될 수 있음을 의미하므로, T에 대해 분지(branch)를 진행하여도 S를 탐색하여 얻을 수 있는 최선의 적재패턴보다 더 나은 해를 얻을 수 없음을 쉽게 알 수 있다. 그러므로 이미 탐색된(분지된) 계단 구조 중에 현재 분지할 노드 T에 대한 우위패턴이 존재한다면, T에 대해 더 이상 분지를 진행하지 않아도 된다.

Martins and Dell[11]은 HVZ 해법에서 중복 탐색을 제거하기 위해 적재 패턴의 대칭성을 이용하였다. 이를 이해하기 위해 W(T)를 임의의 노드 T에 대한 계단 구조에서 비워진 공간(wasted area)의 면적이라 하고, N_{lb}를 노드 T의 탐색 시점까지 구해진 하한이라 하자. 노드 T에서 분지되어 얻을 수 있는 해는 최소한 현재까지 구한 하한인 N_{lb}보다 커야 하므로, 다음 조건 (5)와 조건 (6)을 만족한다면 더 이상 분지를 진행하지 않아도 된다.

$$N(\mathrm{T}) \le \left\lfloor \frac{N_{lb}+1}{2} \right\rfloor, \tag{5}$$

$$W(\mathrm{T}) > \frac{LW - lw(N_{lb}+1)}{2}. \tag{6}$$

즉, 노드 T에서 적재된 상자의 개수가 $N_{lb}+1$의 절반 보다 작거나 같을 때(조건 (5)), 계단 구조 내 빈공간의 면적이 전체 빈공간의 절반보다 클 경우(조건 (6)) 계단 구조 바깥 영역의 빈공간이 전체 빈

공간의 절반보다 작아야 $N_{lb}+1$개의 상자를 적재할 수 있다. 만약 이런 패턴이 존재한다면, 이 패턴을 좌우로 대칭 이동한 패턴은 조건 (6)을 만족하지 않는다. 따라서 다른 노드를 탐색하는 과정에서 동일한 해를 구할 수 있다. 좀 더 직관적으로 설명하면, 상자를 절반 정도 실었을 때 확정된 빈공간의 크기가 나머지 영역의 빈공간의 크기보다 작은 적재 패턴들만 탐색함으로써, 좌우 대칭의 중복 패턴 중 하나의 패턴만 탐색하는 것이다.

지금까지 설명한 계단 구조를 이용한 분지 방법과 상한, 하한 그리고 중복탐색을 제거하는 두 가지 기법을 적용한 최적 우선 분지한계법(best-first branch and bound method)을 나타내는 의사코드(pseudo code)는 다음과 같다. 여기서 L은 잎 노드(leaf node)의 집합, E는 분지 노드의 집합을 나타내며, l(S)와 u(S)는 각각 노드 S에 대한 하한과 상한을 나타낸다.

```
Procedure
  C(S) = {(0, 0)}
  L = {S}, E = {}
  N_lb = 0
  While N_lb < max{u(S) | S∈L}
    u(S*) = max{u(S) | S∈L}
    L = L−{S*}
    If l(S*) > N_lb then
      N_lb = l(S*)
    End If
    If there exists no node that dominates S*
      in E
      and the inequality (5) and (6) are not
      satisfied for S* then
      Create all child nodes by the branching
      strategy and insert them into L.
      L = L−{S | u(S) ≤ N_lb, S∈L}
    End If
  End While
End Procedure
```

5. 실험 결과

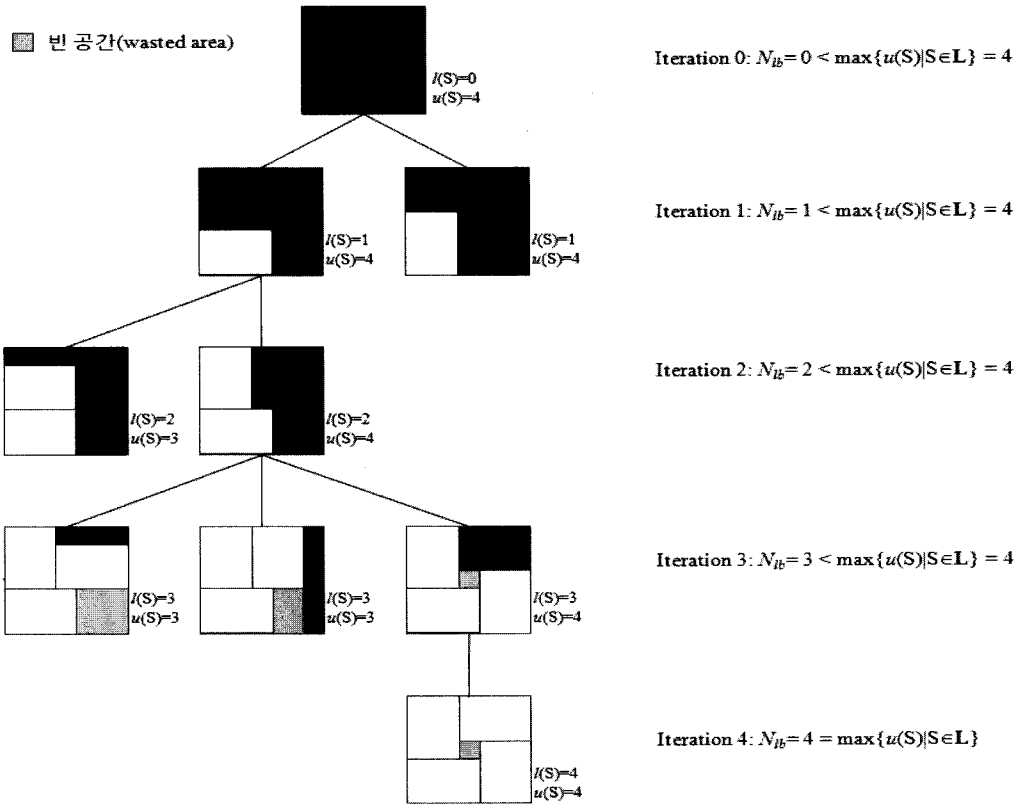
본 연구에서는 제안한 해법을 C++를 이용하여 구현하였으며, 인텔 코어 2듀오 2.13GHz의 중앙처리장치(CPU)와 2GB의 메인메모리(RAM)을 장착한 컴퓨터를 이용하여 실험하였다. 먼저 본 논문에서 제안하는 최적우선분지한계법(best-first branch and bound method)의 이해를 돕기 위해 해를 구하는 주요 과정을 (L, W, l, w)가 (5, 5, 3, 2)인 경우에 대해 간략하게 도식하면 [그림 6]과 같다.

<표 1>은 Martins and Dell[11]의 연구에서 사용한 예제에 대한 실험 결과를 비교한 것이다. 그들이 실험한 컴퓨터(펜티엄 3 600MHz)의 성능과의 차이(약 1/7)를 감안하더라도 제안한 해법이 전반적으로 빠른 시간 내에 최적해를 구할 수 있음을 알 수 있다. 특히 문제 N2와 ST5의 경우 HVZ 해법은 다른 문제들에 비해 매우 많은 계산 시간을 필요로 하지만, 제안한 해법은 큰 차이가 없음을 알 수 있다.

서론에서 밝힌 바와 같이 Martins and Dell[11]의 HVZ 해법은 적재 개수 100개 이하의 모든 동치 문제들 중에 오직 <표 2>에 포함된 5개의 문제에 대한 최적해를 구하지 못했다.

<표 2>는 이들 5개의 문제에 대한 제안한 해법의 실험 결과이며, [그림 7]은 해당 문제들에 대한 최적 적재 패턴을 나타낸 것이다. 이들 5개의 문제에 대한 실험 결과가 모든 팔레트 적재 문제에 대한 해법의 우수성을 보장하지는 않지만, 적재 개수 100개 이내의 모든 문제들에 대한 최적해를 구해냈다는 점에서 중요한 의미를 갖는다. 또한 <표 1>과 <표 2>의 실험결과를 종합해 볼 때 제안한 해법은 고려된 모든 경우에 대해 최적해를 구한 것은 물론이고, HVZ 해법보다 평균적으로 더 적은 탐색 시간으로 최적해를 찾아낼 수 있다고 볼 수 있다.

팔레트적재문제를 위한 또 다른 최적해법(exact algorithm)으로 MBFS(Maximal Breadth Filling Sequence)를 사용한 Bhattacharya et al.[3]의 해법이 있다. 이 해법은 이 후 다른 연구들에 의해 자주 언급되지는 않았지만, 해법의 분지 전략을 분석해



[그림 6] 최적우선분지한계법의 단계별 적용 과정((5, 5, 3, 2)인 경우)

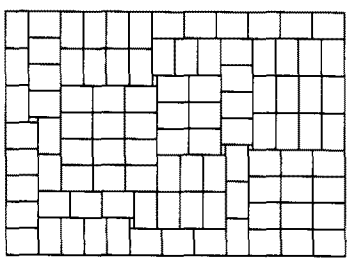
〈표 1〉 제안한 해법과 HVZ 해법과의 비교 실험

문제	(L, W, l, w)	n	실행 시간(초)	
			HVZ	계단 구조
D1	(22, 16, 5, 3)	23	0.00	0.00
D2	(23, 22, 4, 3)	42	0.00	0.00
N1	(43, 26, 7, 3)	53	0.17	0.15
N2	(87, 47, 7, 6)	97	4092.61	2.96
N3	(109, 71, 17, 5)	90	0.05	0.00
N4	(42, 39, 9, 4)	45	0.55	0.13
N5	(64, 41, 11, 5)	47	0.93	0.00
ST1	(40, 25, 7, 3)	47	0.16	0.00
ST2	(52, 33, 9, 4)	47	0.44	0.00
ST3	(57, 44, 12, 5)	41	0.06	0.00
ST4	(56, 52, 12, 5)	48	2.42	0.04
ST5	(127, 85, 9, 8)	149	2247.83	0.00

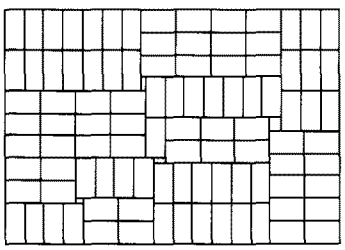
보면 매우 우수한 탐색 구조를 갖고 있음을 알 수 있다. 이 연구를 살펴보면 크게 2개 문제 군(group)에 대해 해법의 성능을 실험하였다. 첫 번째 문제 군은 팔레트 크기를 (1200, 1000)으로 고정하고 다음 두 개의 조건을 만족하는 무작위로 생성된 1000개의 문제로 구성된다.

〈표 2〉 나머지 5 문제에 대한 실험 결과

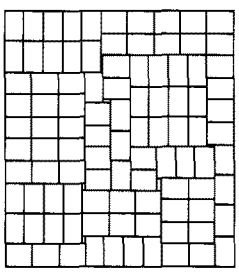
(L, W, l, w)	n	분지된 노드 수	실행 시간(초)
(74, 46, 7, 5)	97	189943	102.02
(86, 52, 9, 5)	99	374639	470.10
(95, 92, 11, 8)	99	390670	573.28
(172, 66, 19, 7)	85	87557	17.93
(178, 60, 16, 7)	95	25709	1.00



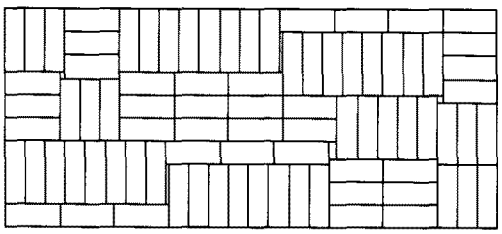
(74, 46, 7, 5)



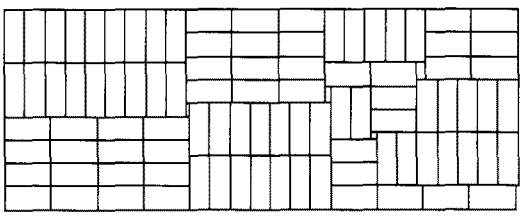
(86, 52, 9, 5)



(95, 92, 11, 8)



(172, 66, 19, 7)



(178, 60, 16, 7)

[그림 7] 남은 5개 문제에 대한 최적 적재 패턴

〈표 3〉 제안한 해법과 MBFS 해법과의 비교

팔레트 크기	해 법	평균실행시간 (초)	분지된 노드 수	
			평균	표준편차
(1200, 1000)	MBFS	0.99	650.89	1436.91
	계단 구조	0.01	112.20	1230.00
(1100, 1100)	MBFS	1.44	779.80	1854.75
	계단 구조	0.00	1.70	26.00

$1 \leq l/w \leq 4,$

$1 \leq LW/lw \leq 50.$

두 번째 문제 군은 팔레트 크기를 (1100, 1100)으로 고정하고 위와 동일한 조건을 만족하는 무작위로 생성된 1000개의 문제로 구성된다. 이 두 개의 문제 군들에 대한 제안한 해법의 실험 결과는 <표 3>과 같다. Bhattacharya et al.[3]의 연구에서 실험을 위해 사용한 하드웨어 및 소프트웨어 환경이 본 연구에 사용된 것과 다르므로, 계산 시간을 비교하

는 것 보다 탐색한(분지된) 노드의 개수를 비교하는 것이 더욱 타당하다. <표 3>의 결과를 살펴보면 첫 번째 문제 군에 대해서 제안한 해법이 기존 해법에 비해 분지된 노드의 개수가 1/6 수준으로 감소된 것을 볼 수 있다. 그러나 표준편차는 평균에 비해 상대적으로 더 큰 것을 볼 수 있는데, 이것은 제안한 해법에서는 탐색과정 중 매 반복주기마다 구하는 하한에 의해 해법 초기에 최적해를 구하는 경우가 많이 발생했기 때문이다. 이에 반해 MBFS 해법의 경우 구조적으로 하한을 구하기가 어렵고, 실제로 해법 내에서 하한을 구하지 않는다. 이러한 현상은 두 번째 문제 군에서 더욱 극적으로 드러나는데, 분지된 노드 수의 평균이나 표준편차에 있어서 제안한 해법이 월등히 우수한 성능을 보여준다.

6. 결 론

본 연구에서는 계단 구조를 유지하는 분지전략을 적용한 팔레트적재문제를 위한 최적 우선 분지한계법(best-first branch and bound method)을 제안하였다. 각 노드에 대응하는 계단 구조의 특성으로 인해 각 노드에 대한 더욱 효율적인 상한과 하한을 구할 수 있으며, 이로 인해 탐색의 효율을 향상시킬 수 있다.

해법의 우수성을 보이기 위해 최근의 연구들 중 최적해법으로 잘 알려진 Martins and Dell[11]의 HVZ 해법과 Bhattacharya et al.[3]의 MBFS 해법과 비교하여 실험하였다. 실험 결과, 제안한 해법이 이들 두 해법에 비해 월등히 적은 계산량으로 최적해를 구해낼 수 있음을 알 수 있었다. 특히, 가장 최근 연구인 Martins and Dell[11]의 연구에서는 100개 이내의 적재 개수를 갖는 모든 팔레트적재문제들에 모든 최소 크기의 동치문제들 중 5개의 문제에 대한 최적해를 구하는데 실패했는데, 제안한 해법은 이들 5개의 문제에 대한 최적해를 구하여 제시하였다. 앞으로 계단 구조의 기하학적 특성을 이용하여 더욱 효율적인 상한과 하한을 구한다면, 제안한 해법의 성능을 더욱 향상 시킬 수 있을 것이다.

참 고 문 헌

[1] Arenales M. and R. Morabito, "An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems," *European Journal of Operational Research*, Vol.84(1995), pp.599-617.

[2] Barnes F., "Packing the maximum number of $m \times n$ tiles in a large $p \times q$ rectangle," *Discrete Mathmatics*, Vol.26(1979), pp.93-100.

[3] Bhattacharya S., R. Roy, and S. Bhattacharya, "An exact depth-first algorithm for the pallet loading problem," *European Journal of Operational Research*, Vol.110(1998), pp.610-625.

[4] Dowsland K., "The Three-dimensional pallet chart : An analysis of the factors affecting the set of feasible layouts for a class of two-dimensional packing problems," *Journal of the Operational Research Society*, Vol.35(1984), pp.895-905.

[5] Dowsland K., "Determining an upper bound for a class of rectangular packing problems," *Computers and Operations Research*, Vol.12 (1985), pp.201-205.

[6] Dowsland K., "An exact algorithm for the pallet loading problem," *European Journal of Operational Research*, Vol.31(1987), pp.78-84.

[7] G Y.-G. and M.-K. Kang, "A fast algorithm for two-dimensional pallet loading problems of large size," *European Journal of Operational Research*, Vol.134(2001), pp.193-202.

[8] G Y.-G. and M.-K. Kang, "A new upper bound for unconstrained two-dimensional cutting and packing," *Journal of the Operational Research Society*, Vol.53(2002), pp.587-591.

[9] Letchford A. and A. Amaral, "Anlysis of upper bounds for the pallet loading problem,"

European Journal of Operational Research, Vol.132(2001), pp.582-593.

[10] Martins G. and R. Dell, "The minimum size instance of a pallet loading problem equivalence class," *European Journal of Operational Research*, Vol.179(2007), pp.17-26.

[11] Martins G. and R. Dell, "Solving the pallet loading problem," *European Journal of Operational Research*, Vol.184(2008), pp.429-440.

[12] Morabito R. and S. Morales, "A simple and effective recursive procedure for the manufacturer's pallet loading problem," *Journal of the Operational Research Society*, Vol.49(1998), pp.819-828.

[13] Nelißen J., "How to use structural constraints to compute an upper bound for the pallet loading problem," *European Journal of Operational Research*, Vol.84(1995), pp.662-680.

[14] Scheithauer G. and J. Terno, "The G4-heuristic for the pallet loading problem," *European Journal of Operational Research*, Vol.108(1996), pp.511-522.