

3차원 컨테이너 적재 문제를 위한 발견적 해법

장창식 · 강맹규

한양대학교 산업공학과

A Heuristic for the Container Loading Problem

Chang-Sik Jang · Jaing-Kyu Kang

Dept. of Industrial Engineering, Hanyang University

A new heuristic algorithm for the heterogeneous single container loading problem is proposed in this paper. This algorithm fills empty spaces with the homogeneous load-blocks of identically oriented boxes and splits residual space into three sub spaces starting with an empty container. An initial loading pattern is built by applying this approach recursively until all boxes are exhausted or no empty spaces are left. In order to generate alternative loading patterns, the load-blocks of pattern determining spaces are replaced with the alternatives that were generated on determining the load-blocks. An improvement algorithm compares these alternatives with the initial pattern to find improved one. Numerical experiments with 715 test cases show the good performance of this new algorithm, above all for problems with strongly heterogeneous boxes.

Keywords : container Loading Problem, Packing, Heuristic

1. 서 론

본 연구는 크기가 다른 3차원 직육면체 상자를 한 대의 컨테이너에 적재할 때 컨테이너의 용적 사용을 최대화하는 3차원 혼합 적재 문제를 다룬다. 각 상자에 대해 적재할 개수가 주어지며 최대 6가지의 적재 방향이 허용된다. 또한 이 상자들의 가장자리는 모두 컨테이너의 가장자리와 평행하고 서로 겹침이 없이 적재되어야 한다.

Dyckhoff[9]에 따르면 3차원 컨테이너 적재 문제는 크게 두 종류로 분류된다. 첫번째인 단순 컨테이너 적재 문제(single container loading problem)는 컨테이너의 용적이 박스를 모두 적재할 수 없는 경우에 불용 공간을 최소화하여 가급적 많은 박스를 적재하는 것이다. 해법의 성능은 용적률(volume efficiency)에 의해 결정된다. 용적률이란 컨테이너의 용적에 대한 적재된 박스의 총 용적의 비율을 말한다. 두번째인 복수 컨테이너 적재 문제(multiple container loading problem)는 주어진 모든 박스를 적재하는 컨테이너 대수를 최소화하는 것이다.

상자의 종류에 따라 분류하면 한 종류의 상자를 적재하는 단일 적재 문제 (homogeneous loading problem)와 크기가 다른 여러 상자를 함께 적재하는 혼합 적재 문제(heterogeneous loading problem)로 분류된다[4]. 혼합 적재 문제는 다시 상자 종류가 많고 각 종류의 개수가 적은 강혼합(strongly heterogeneous) 적재 문제와 종류가 적고 각 종류의 개수가 많은 약혼합(weakly heterogeneous) 적재 문제로 분류된다.

본 연구는 단순 컨테이너의 혼합 적재 문제(heterogeneous single container loading problem)를 다룬다.

3차원 적재 문제는 실제 산업계에서 매우 광범위하게 사용된다. 제조, 유통 및 물류 업계에서 화물 운송을 위한 차량 적재 계획 수립, 공장에서 완성품을 적재하기 위한 포장 상자 및 팔레트 소요량 산정, 그리고 창고에서 대리점까지 배송할 때 최적의 물동량 산정 등을 예로 들 수 있다.

이러한 폭넓은 산업계의 요구에 따라 이 문제에 대한 다양한 연구가 진행되어 왔다. 그러나 이 문제는

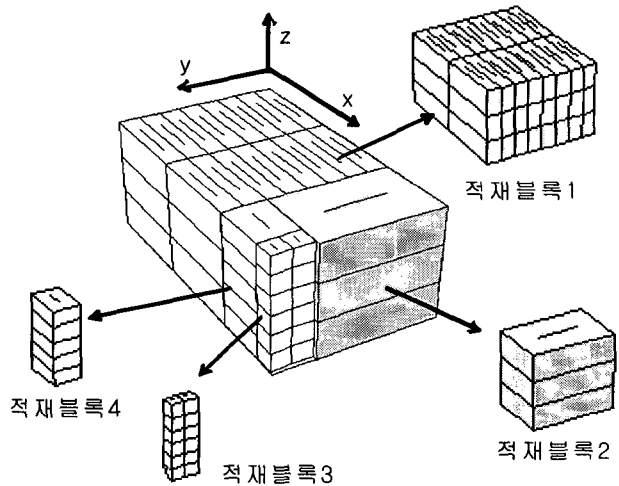
NP-Hard 문제로서 실제 필요한 최적의 해법을 찾는 것이 쉽지 않다. 그 동안 발표된 연구는 대부분 발견적 방법을 사용하는데 크게 4개의 접근 방법에 따라 분류된다.

- (1) 벽 쌓기(wall building) : 컨테이너 안쪽부터 벽을 쌓듯이 상자를 길이 방향으로 계속적으로 적재하는 방식이다. 이 방법은 불용 공간이 많이 발생하여 적재의 안정성을 보장하지 못한다[11,12].
- (2) 탑 구성(tower arrangement) : 컨테이너 바닥 면에 상자를 올리고 그 위로 더 작은 상자를 계속해서 적재하는 방식이다[4,6,10].
- (3) 수직 층(vertical layers) : 컨테이너를 수직으로 분할하는 층으로 나누고 다시 층 내부를 수평으로 분할하는 조각(strip)으로 채운다. 배낭 문제(knapsack)를 사용하여 조각에 상자를 채워 최종 적재를 완성한다[17,19].
- (4) 블록 조합(block arrangement) : 컨테이너의 내부 공간을 동일한 종류의 상자가 동일한 방향으로 배열된 직육면체 블록으로 적재하는 방식이다. 공간에 적재블록이 정해지면 남은 공간을 분할한다. 이 방식은 최근의 논문에서 자주 사용되었다. Bortfeldt와 Gehring[5]는 한 공간에 최대 2개의 블록을 적재하고 3개의 하위 공간으로 분할한다. 그리고 터부 검색으로 적재를 완성한다. Bortfeldt et al.[7]도 한 공간에 최대 2개의 블록을 적재하지만 최대 6개의 하위 공간을 생성한다. 병렬 터부 검색을 사용하여 적재를 완성한다. Eley[13]은 한 공간에 한 개의 블록을 적재하고 3개의 하위공간을 생성한다. 트리 검색으로 적재를 완성한다.

본 연구에서는 블록 조합을 사용한다. 한 공간을 한 개의 직육면체 블록으로 채운 후 남은 공간을 3개의 하위 공간으로 분할한다. 이 하위 공간을 다시 블록으로 채우고 공간을 분할하는 과정을 연속적으로 적용하여 초기 적재패턴을 생성한다. 이 초기 적재패턴에 개선 해법을 사용하여 개선한다.

이 직육면체 블록을 적재블록이라고 하자. 적재블록은 동일한 종류의 상자가 동일한 방향으로 적재된 직육면체 묶음으로 3개의 좌표축(x, y, z)에 대해 1개 이상의 상자로 구성된다(<그림 1> 참조).

본 연구에서 적재블록을 사용하는 이유는 컨테이너가 적재블록으로 채워질 경우 현실적인 면에서 여러 이점을 제공하기 때문이다. 예를 들면, 동일한 종류의 상자를 인접하게 위치하여 적재와 하역의 작업 시간을 줄일 수 있으며, 동일한 형태의 상자들은 그렇지 않은 경우보다 무게 하중에 강하여 다른 상자들과 혼합 적재하더라도 파손되거나 변형되지 않으며, 적재블록 안의 상자들이 서로 지지하므로 상자들이 미끄러지지 않는 것이다[13].



<그림 1> 4개의 적재블록으로 구성된 적재패턴

본 논문은 2장에서 새로운 해법을 제안하고, 3장에서 수치 실험과 기존 논문과의 비교 결과를 보이고 4장에서 본 연구를 결론 짓는다.

2. 제안하는 해법

컨테이너의 길이, 너비, 높이를 각각 L, W, H 라하고 상자 종류 i 의 길이, 너비, 높이를 각각 l_i, w_i, h_i , 용적을 v_i , 적재 개수를 q_i 라 하면 컨테이너에 적재된 상자의 용적합 v 는 식 (1)과 같이 정의된다. 그리고 컨테이너 용적에 대한 용적합의 비율인 용적률 E 는 식 (2)와 같다.

$$v = \sum_{i=1}^m q_i v_i \dots\dots\dots (1)$$

$$E = \frac{v}{LWH} 100 (\%) \dots\dots\dots (2)$$

여기서 m 은 상자 종류 수이다. 단수 컨테이너 혼합 적재 문제의 목적은 컨테이너에 최대한 많은 상자를 적재하여 이 용적률을 최대화하는 것이다.

상자 i 가 좌표 (x, y, z) 와 $(x+l_i, y+w_i, z+h_i)$ 사이에 위치한다면 (x, y, z) 를 상자 i 의 기준좌표(reference point)라 한다. 만약 p 개의 상자가 서로 겹침이 없이 모두 컨테이너 안에 적재된다면 $(x_j, y_j, z_j, \pi_j), j=1, \dots, p$ 의 집합은 이 p 개의 적재를 유일하게 표현한다. 이 집합을 적재패턴이라 한다. 여기서 x_j, y_j, z_j 는 상자 $\pi_j (\pi_j \in \{1, \dots, m\})$ 의 기준좌표를 말한다. 만약 컨테이너 안에 다른 공간이 남아

있고 이 공간이 다른 상자를 적재할 만큼 충분히 크다면 한 개 이상의 새로운 적재패턴을 구할 수 있다. 따라서, 컨테이너의 전체 적재패턴을 구하는 것은 빈 컨테이너에 한 개의 상자를 적재하여 적재패턴을 구성하고 새로운 공간에 다시 다른 적재패턴을 구성하는 과정을 반복하는 것과 같다.

p 개 상자가 적재된 것을 단계(stage) p 의 상태(state)라 하면, 최대의 용적합을 가지는 최적의 적재패턴은 매우 많은 수의 상태에서 유한계를 검색하여 찾을 수 있다. 그러나 이 유한개의 수를 모두 검색하는 것은 많은 시간과 메모리를 요구하므로, 이를 줄이는 효과적인 전략이 필요하다[14,18]. 이에 본 연구는 다음과 같은 전략을 사용한다.

- (1) 적재패턴을 가급적 동일한 종류의 상자와 방향으로 이루어진 적재블록으로 구성한다(<그림1> 참조).
- (2) 다음의 단계로 진행하기 전에 가능한 모든 상태를 평가함수로 미리 계산하여 특정 범위를 벗어나는 것을 제외한다. 즉, 평가함수에 의해 가장 좋은 평가값을 가지는 k 개만 계산하여 다음 단계로 진행한다(평가함수는 2.1.3 장에서 설명함).
- (3) 만약 현재의 단계가 전체 패턴의 형성에 큰 기여를 하는 것으로 판단되면 차선의 평가값을 가진 상태들도 함께 기억한다.
- (4) 일단 탐색이 종료되어 초기 적재패턴 T 를 찾았으면 (3)의 차선 상태들을 다시 사용하여 더 좋은 적재패턴 T^* 을 찾는다.

본 해법은 (1), (2), (3)의 전략을 사용하여 초기 적재패턴을 생성하고 이를 (4)를 통해 개선하는 방법을 사용한다. 다음 두 절부터 각각에 대해 설명한다.

2.1 초기 적재패턴 생성

초기 적재패턴을 생성하는 절차는 다음과 같다.

- step 1. 상자 리스트를 용적이 큰 순으로 정렬한다.
- step 2. 컨테이너의 크기를 현재 공간으로 지정한다.
- step 3. 현재 공간에 가장 적합한 적재블록을 결정한다. 적재블록의 결정은 현재 공간에 가능한 모든 대안블록을 생성하여 이를 모두 평가한 후 최상의 대안을 선택하는 방법을 사용한다(2.1.2 장에서 설명함).
- step 4. 결정된 적재블록을 컨테이너에 위치시킨다. 한 공간은 항상 한 개의 적재블록을 가지는 데 적재블록이 실제로 컨테이너에 적재될

때는 해당 공간의 기준좌표에 위치한다. 기준좌표는 컨테이너의 안쪽 하단의 모서리를 원점으로 할 때 이와 가장 가까운 꼭지점을 말한다.

- step 5. 결정된 적재블록에 해당하는 상자의 개수를 상자 리스트에서 감한다.
- step 6. 적재블록을 제외한 나머지 공간을 3개의 하위 공간으로 분할한다(2.1.1 장에서 설명함).
- step 7. 각 하위 공간에 다시 step 3~6을 재귀적으로 적용한다.
- step 8. 만약 현재 공간에 상자를 적재할 수 없는 경우 다른 불용 공간과 병합하여 재시도한다.
- step 9. 모든 상자가 적재되거나 더 이상 사용할 공간이 없으면 해법을 종료한다.

2.1.1 공간분할 전략

공간 S 의 기준좌표에 적재블록 B_S 가 위치하면 이 블록을 제외한 공간을 하위 공간으로 분할한다. 이 때 3가지의 분할 방법, *TopToFront*, *SideToTop*, *SideToFront* 중 하나를 사용한다. 공간마다 이 분할 방법이 서로 다른데 어떤 분할 방법을 사용하든 항상 3개의 하위 공간, S_{side} , S_{stop} , S_{front} 가 생성된다. S 의 공간분할 방법을 나타내는 변수를 t_s 라고 하자. t_s 는 해당 공간의 적재블록을 결정하는 시점에서 함께 결정된다(2.1.2장 참조).

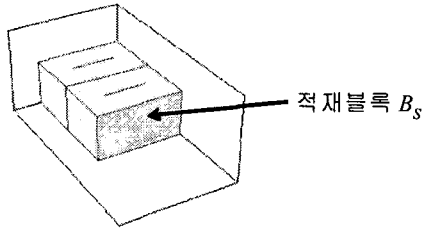
t_s 에 따라 하위 공간의 크기가 결정된다. 공간 S 의 길이, 너비, 높이를 각각 L_S, W_S, H_S , 적재블록 B_S 의 길이, 너비, 높이를 각각 L_{BS}, W_{BS}, H_{BS} 라 하자. 그리고 공간 x 의 크기를 D_x (x 의 길이, x 의 너비, x 의 높이)라 하자. 이 정의에 의해 각 하위 공간의 크기 $D_{S_{side}}, D_{S_{stop}}, D_{S_{front}}$ 는 t_s 가 *TopToFront*인 경우는 식 (3), *SideToTop*인 경우는 식 (4) 그리고 *SideToFront*인 경우는 식 (5)와 같다.

$$\begin{aligned} D_{S_{side}} &= (L_{BS}, W_S - W_{BS}, H_S) \\ D_{S_{stop}} &= (L_{BS}, W_{BS}, H_S - H_{BS}) \dots\dots\dots (3) \\ D_{S_{front}} &= (L_S - L_{BS}, W_S, H_S) \end{aligned}$$

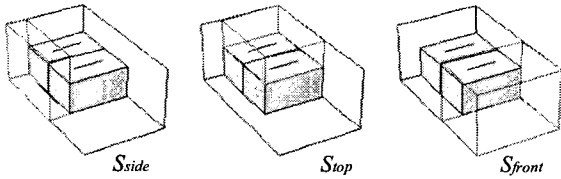
$$\begin{aligned} D_{S_{side}} &= (L_{BS}, W_S - W_{BS}, H_{BS}) \\ D_{S_{stop}} &= (L_S, W_S, H_S - H_{BS}) \dots\dots\dots (4) \\ D_{S_{front}} &= (L_S - L_{BS}, W_S, H_{BS}) \end{aligned}$$

$$\begin{aligned} D_{S_{side}} &= (L_{BS}, W_S - W_{BS}, H_{BS}) \\ D_{S_{stop}} &= (L_{BS}, W_S, H_S - H_{BS}) \dots\dots\dots (5) \\ D_{S_{front}} &= (L_S - L_{BS}, W_S, H_S) \end{aligned}$$

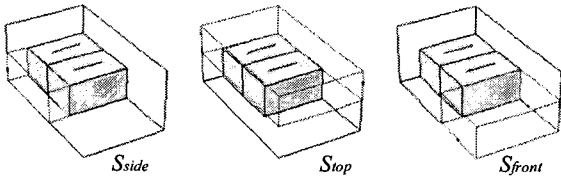
<그림 2>의 (a), (b), (c)에 각각 식 (3), (4), (5)에 따라 하위 공간의 크기가 결정되는 상황을 보인다.



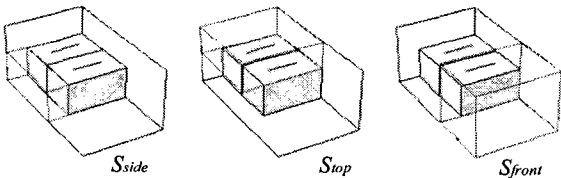
(a) $t_s = TopToFront$ 인 경우



(b) $t_s = SideToTop$ 인 경우



(c) $t_s = SideToFront$ 인 경우



<그림 2> t_s 에 따른 공간분할 방법

생성된 하위 공간의 사용 순서도 t_s 에 의해 결정된다. t_s 가 $TopToFront$ 인 경우는 $S_{stop} \rightarrow S_{side} \rightarrow S_{front}$, $SideToTop$ 인 경우는 $S_{side} \rightarrow S_{front} \rightarrow S_{stop}$ 그리고 $SideToFront$ 인 경우는 $S_{side} \rightarrow S_{stop} \rightarrow S_{front}$ 의 순서를 따른다.

만약 공간 S 에 적재를 시도하여 더 이상 적재할 수 없는 경우, 공간병합을 사용한다. 컨테이너에 존재하는 모든 불용 공간에 대해서 일정한 조건을 만족하면 S 에 병합한다. 공간 S' 이 S 에 병합되기 위한 조건은 다음과 같다.

- (1) 두 공간이 한 개의 인접면을 가질 것
- (2) S' 의 인접면의 크기가 S 보다 크거나 같을 것
- (3) S' 의 기준좌표가 S 의 것보다 작을 것($x_{S'} \leq x_S, y_{S'} \leq y_S, z_{S'} \leq z_S$)

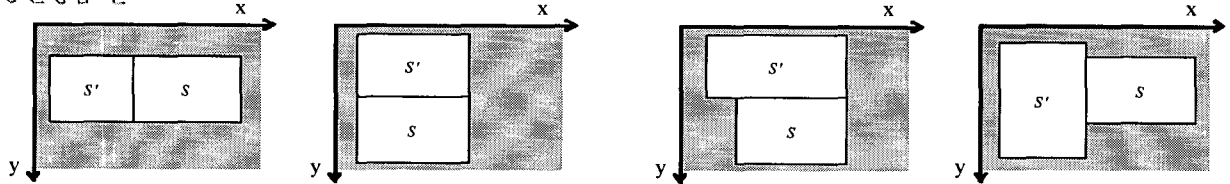
<그림 3>은 두 공간이 병합되는 상황을 단면도로 표시한다. 3-(a)는 두 공간의 인접면의 크기가 동일할 때 S' 이 S 에 병합되어 S' 이 제거되는 상황을, 3-(b)는 S' 의 인접면의 크기가 S 보다 클 때 S' 이 줄어드는 상황을 보인다. 병합에도 불구하고 상자를 적재할 수 없다면 S 를 불용 공간으로 표지하여 다음 번 공간병합에 사용한다.

2.1.2 적재블록의 결정

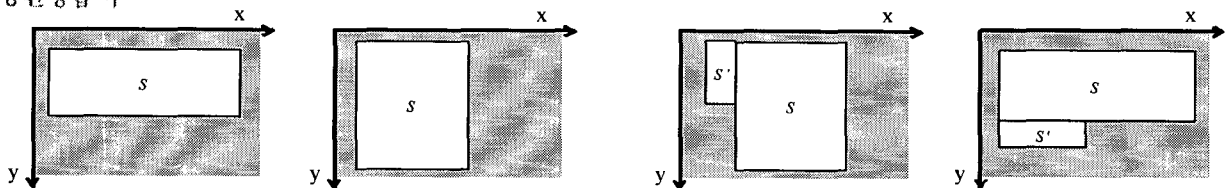
공간 S 의 적재블록 B_S 를 결정하기 위해 가능한 대안블록을 생성하여 이들을 모두 평가한 후 이 중에서 최상의 평가값을 가진 것을 선택한다.

대안블록이란 공간 S 에 적재 가능한 상자 종류, 적재 방향, 상자의 배열 및 공간분할 방법을 저장하는 구조체로 상자 i 의 k 번째 대안블록을 다음과 같이 표현할 수 있다.

공간병합 전



공간병합 후



(a) 두 공간의 인접면이 동일한 경우

(b) S' 인접면이 S 보다 큰 경우

<그림 3> 두 공간의 병합 (단면도)

$$B_{ik} = (i, j, \alpha_{ik}, \beta_{ik}, \gamma_{ik}, t_{ik})$$

$$(i=1, \dots, m'; k=1, \dots, u_j)$$

여기서 m' 은 현재까지 적재되지 않은 상자 종류 수, u_i 는 상자 i 에서 생성되는 대안블록의 수이다. $\alpha_{ik}, \beta_{ik}, \gamma_{ik}$ 는 상자 i 의 배열을 나타내는데 각각 x, y, z 축으로 위치하는 개수로서 총 상자 개수는 $\alpha_{ik}\beta_{ik}\gamma_{ik}$ 와 같다. t_{ik} 는 B_{ik} 를 평가할 때 사용할 공간분할 방법을 결정하는 변수로서 *TopToFront*, *SideToTop* 및 *SideToFront* 중 한 가지를 가진다.

$\alpha_{ik}, \beta_{ik}, \gamma_{ik}$ 와 u_i 를 계산하기 위해 기호를 다음과 같이 정의한다.

- B' : 적재되지 않은 상자의 집합
- q_i : 상자 i 의 적재되지 않은 개수 ($i \in B'$)
- d_i : 상자 i 에 허용된 적재 방향의 집합 ($d_i = \{1, \dots\}$)
- l_{ij} : 상자 i 가 j 방향일 때의 길이 ($j \in d_i$)
- w_{ij} : 상자 i 가 j 방향일 때의 너비
- h_{ij} : 상자 i 가 j 방향일 때의 높이
- a_{ij} : 상자 i 가 j 방향으로 S 의 길이(x 축)를 따라 적재될 수 있는 최대 개수로 다음과 같다. 여기서 $\lfloor x \rfloor$ 는 x 보다 크지 않은 정수이다.

$$a_{ij} = \lfloor L/l_{ij} \rfloor$$

b_{ij} : 상자 i 가 j 방향으로 S 의 너비(y 축)를 따라 적재될 수 있는 최대 개수로 다음과 같다.

$$b_{ij} = \lfloor W/w_{ij} \rfloor$$

c_{ij} : 상자 i 가 j 방향으로 S 의 높이(z 축)를 따라 적재될 수 있는 최대 개수로 다음과 같다.

$$c_{ij} = \lfloor H/h_{ij} \rfloor$$

$\alpha_{ik}, \beta_{ik}, \gamma_{ik}$ 는 아래의 범위에서 결정된다.

$$1 \leq \alpha_{ik} \leq a_{ij}$$

$$1 \leq \beta_{ik} \leq b_{ij}$$

$$1 \leq \gamma_{ik} \leq c_{ij}$$

$$1 \leq \alpha_{ik}\beta_{ik}\gamma_{ik} \leq q_{ij}$$

그리고 u_i 는 식 (6)과 같다. 여기서 3은 3개의 공간분할 방법을 모두 시도함을 의미한다. 또한 공간 S 의 전체

대안블록의 수 u_S 는 식 (7)과 같다.

$$u_i = 3 \sum_{j \in d_i} a_{ij} b_{ij} c_{ij} \tag{6}$$

$$u_S = \sum_{i=1, i \in E} u_i \tag{7}$$

이에 따라 적재블록 B_S 를 결정하는 절차는 다음과 같다.

- step 1. 상자 i 의 모든 B_{ik} 에 대해 평가함수를 사용하여 평가값 E_{ik} 를 계산한다. E_{ik} 는 평가함수에 의해 계산된 B_{ik} 의 평가값이다. 이 평가함수는 2.1.3장에서 다시 설명한다.
- step 2. B_i 를 구한다. B_i 는 상자 i 의 모든 대안블록 중에서 최대의 평가값을 갖는 대안블록으로서 $B_i = \max_{1 \leq k \leq u_i} B_{ik}$ 이다.
- step 3. B_i 를 대안블록 리스트 C_S 에 첨가한다. C_S 의 최대 크기는 m' 이다.
- step 4. step 1~3을 m' 개의 상자 종류에 적용하여 C_S 를 완성한다.
- step 5. C_S 를 평가값이 큰 순으로 정렬한다.
- step 6. C_S 의 첫 번째인 $C_S(1)$ 를 B_S 로 결정한다. 즉, B_S 는 $\max_{1 \leq i \leq m'} B_i$ 와 같다.

여기서 B_S 를 결정할 때 C_S 를 함께 생성하는 이유는 일단 컨테이너의 초기 적재패턴이 결정된 후에 C_S 의 다음 대안인 $C_S(a)$ ($2 \leq a \leq m'$)을 다시 적재블록으로 사용하여 다른 적재패턴을 생성하기 위해서다. 이는 2.2장에서 자세히 설명한다.

마지막으로 공간 S 의 하위 공간분할에 사용할 t_s 는 적재블록 B_S 의 공간분할 변수 t_{B_S} 로 결정한다.

2.1.3 대안블록의 평가함수

대안블록의 평가함수는 공간 S 와 대안블록 B_{ik} 가 주어졌을 때 평가값 E_{ik} 를 계산한다.

이를 위해 B_{ik} 의 크기를 제외한 공간을 3개의 하위 공간 S_{top} , S_{side} 와 S_{front} 로 분할한다. 그리고 분할된 각 하위 공간에 B_{ik} 의 상자를 제외한 나머지 상자들을 가상적으로 채운다.

여기서 V_B 를 B_{ik} 의 용적, G_x 를 공간 x 에 적재된 상자의 용적 그리고 V_S 를 공간 S 의 용적이라고 할 때, B_{ik} 의 평가값 E_{ik} 는 식 (8)과 같다. 즉, B_{ik} 는 공간 S 에 B 가 적재블록으로 결정될 때 예상되는 용적률을 탐욕적(greedy)으

로 계산한 결과이다.

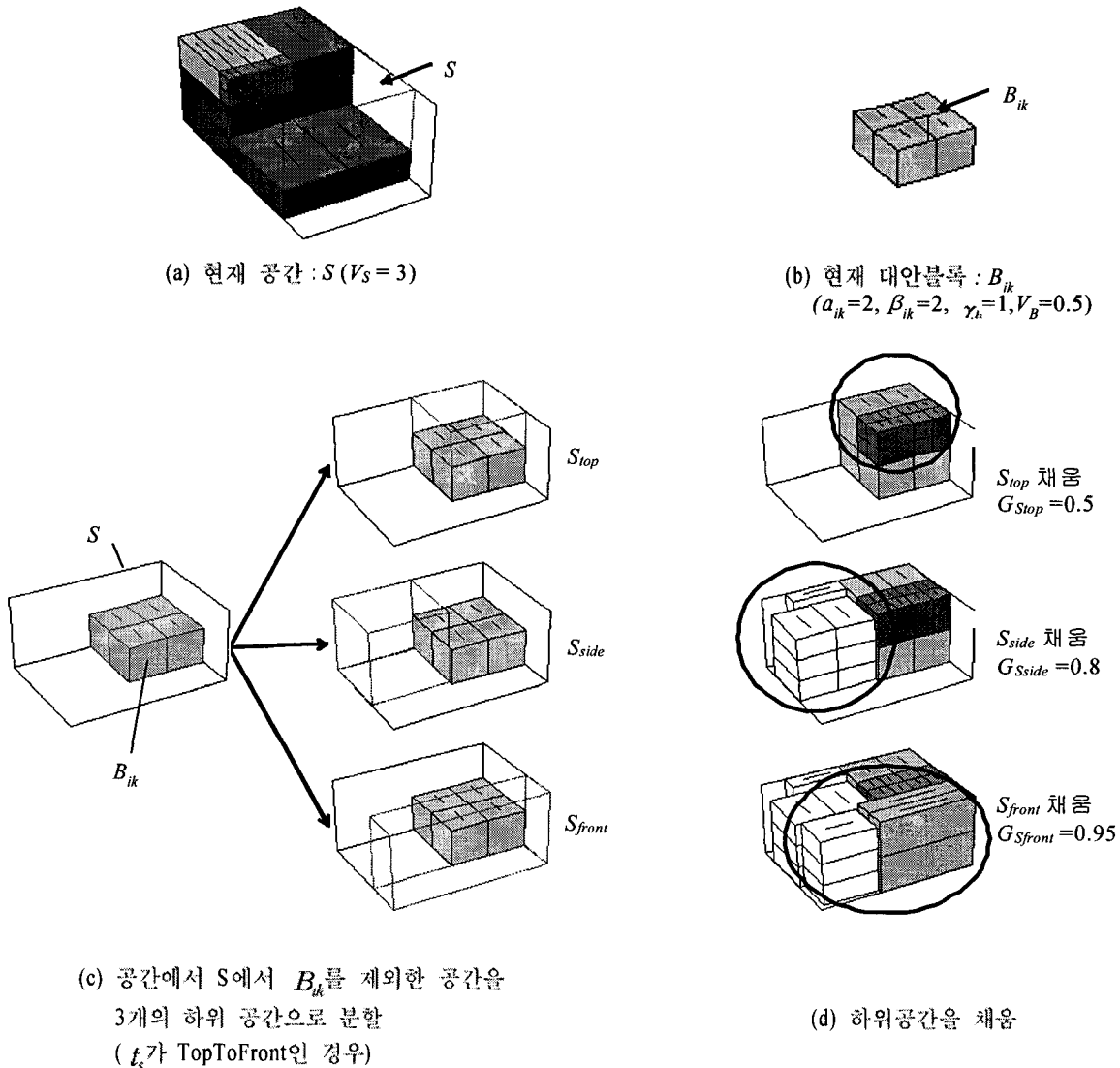
$$E_{ik} = \frac{VB + G_{Stop} + G_{Side} + G_{Sfront}}{V_S} 100(\%) \dots\dots\dots (8)$$

<그림 4>에 대안블록 B_{ik} 에 대한 E_{ik} 를 계산하는 예를 보인다. 4-(a)는 현재 공간 S , 4-(b)는 $\alpha_{ik}=2, \beta_{ik}=2, \gamma_{ik}=1$ 인 대안블록 B_{ik} 가 주어진 상황을 의미한다. 4-(c)는 t_{ik} 가 TopToFront인 경우에 B_{ik} 를 제외한 공간을 3개의 하위 공간으로 분할한 상황, 4-(d)는 각 하위 공간을 채운 상황이다. 식 (8)에 의해 E_{ik} 는 91.6이 된다.

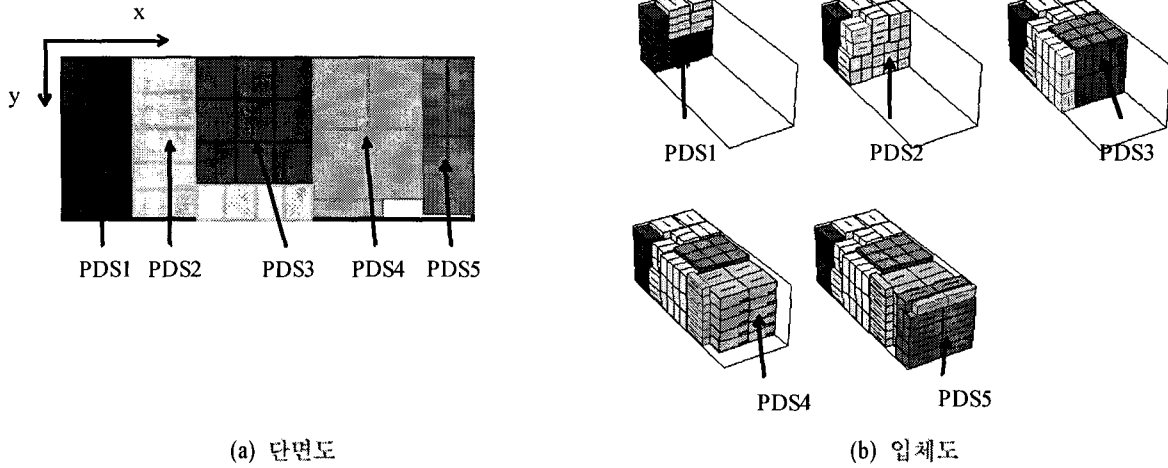
2.2 적재패턴의 개선

2.1에서는 초기 적재패턴을 생성하는 방법을 설명하였다. 이 절에서는 적재블록을 결정할 때 함께 생성한 대안을 이용하여 초기 적재패턴을 개선하는 방법을 설명한다. 이를 위해 다음과 같은 전략을 사용한다.

- (1) 전체 공간이 아닌 패턴 결정 공간(PDS : item determining space)만 대상으로 한다. 패턴 결정 공간이란 전체 적재패턴의 형성에 큰 기여를 하는 것으로 해당 기준좌표 (x, y, z)이 y=0, z=0인 공간을 말한다(<그림 5> 참조). 즉, 이 공간에 다른 적



<그림 4> 공간 S 에 대안블록 B_{ik} 를 평가하여 평가값 E_{ik} 를 계산하는 예



<그림 5> 5개의 패턴결정 공간으로 구성된 적재패턴

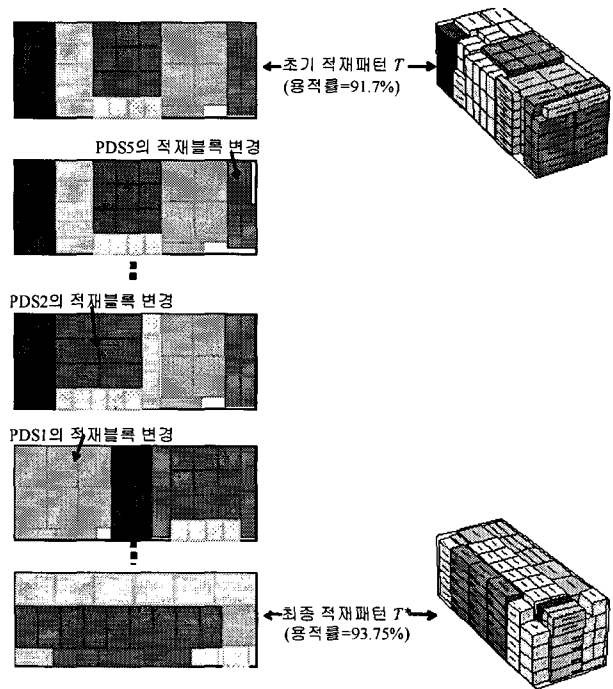
재블록이 위치하면 인접한 다른 공간의 크기에 영향을 주게 된다. 이로 인해 전체 적재패턴의 모양도 크게 바뀌어 개선의 기회를 크게 할 수 있기 때문이다.

- (2) 패턴 결정 공간의 적재블록 B_S 가 다른 형태로 변하면 이 공간의 기준좌표 (x, y, z) 보다 큰 모든 공간과 해당 적재블록을 모두 제거한다.

이에 따라 초기 적재패턴을 개선하는 절차는 다음과 같다.

- step 1. 구하려는 최종 적재패턴 T^* 를 초기 적재패턴 T 로 초기화한다.
- step 2. 패턴 T 의 패턴 결정 공간 S 의 적재블록 B_S 를 a 번째 대안블록 $C_S(a)$ 로 변경한다. 즉 $B_S=C_S(a)$ 가 된다($2 \leq a \leq m$).
- step 3. B_S 가 변경되므로 S 의 하위 공간 S_{stop} , S_{side} , S_{front} 와 각 공간에 위치한 적재블록 B_{Sstop} , B_{Sside} , B_{Sfront} 를 모두 제거한다.
- step 4. S 에 대해 '초기 적재패턴 생성' 절차의 step 4부터 적용하여 새로운 적재패턴 T' 을 생성한다. T' 의 용적률이 T^* 보다 좋으면 T^* 를 T' 으로 갱신한다.
- step 5. step 2~4를 C_S 의 모든 대안블록에 적용한다.
- step 6. step 5를 모든 패턴 결정 공간에 적용한다.
- step 7. T^* 을 최종해로 결정한다.

<그림 6>에 용적률이 91.7%인 초기 적재패턴 T 가 이 개선 절차에 의해 93.75%로 개선되는 것을 보인다.



<그림 6> 초기 적재패턴을 개선하는 예

3. 수치 실험

본 해법의 결과를 다음의 기존 논문과 비교한다(아래에서 H는 발견적 방법, GA는 유전 해법, TS는 터부 검색, HGA는 hybrid 유전 해법, PTS는 병렬 터부검색을 의미함).

- N : goi et al.[16]의 발견적 해법
- B : ischoff et al.[1] 발견적 해법
- BR : ischoff와 Ratcliff[2]의 발견적 해법
- A_GB : ehrling과 Bortfeldt[10]의 유전 해법
- I_BG : ortfeldt와 Gehring[5]의 터부 검색
- T : erno et al.[19]의 발견적 해법
- GA_GB : ortfeldt와 Gehring[6]의 hybrid 유전 해법
- M : ley[13]의 발견적 해법
- S_BG : ortfeldt et al.[7]의 병렬 터부 검색

사용한다. 다음 그룹으로 갈수록 상자의 종류가 늘어나서 BR7은 20종을 사용한다. 국제 표준으로 사용되는 20FT 컨테이너(5870 x 2330 x 2200mm)를 사용하며 모든 상자의 치수는 [300,120], [25,100] 그리고 [20,80]mm 사이에 분포한다. 각 상자는 허용된 적재 방향이 주어진다. 두번째는 Loh와 Nee[15]에 의해 생성된 15개의 문제이다. 각 문제는 모두 다른 컨테이너 치수를 사용하는데 15개중 11개 문제의 컨테이너는 해당 상자들을 모두 적재할 수 있는 용적을 가지고 있다. 상자의 종류는 6에서 10개이고 상자의 개수는 100에서 250사이에 분포한다.

본 해법을 마이크로소프트 C++ 6.0으로 프로그램을 작성하고 이를 Windows XP Service Pack2가 설치된 펜티엄 PC 598MHz의 768MB RAM에서 실험하였다.

기존 논문에서 사용하는 두 종류의 데이터를 대상으로 실험하였다. 첫번째인 Bischoff과 Ratcliff[2]의 데이터는 총 7종의 그룹 BR1-BR7로 구성되며 각 그룹은 100개의 문제를 포함하고 있다. 이중 BR1은 3종의 상자만

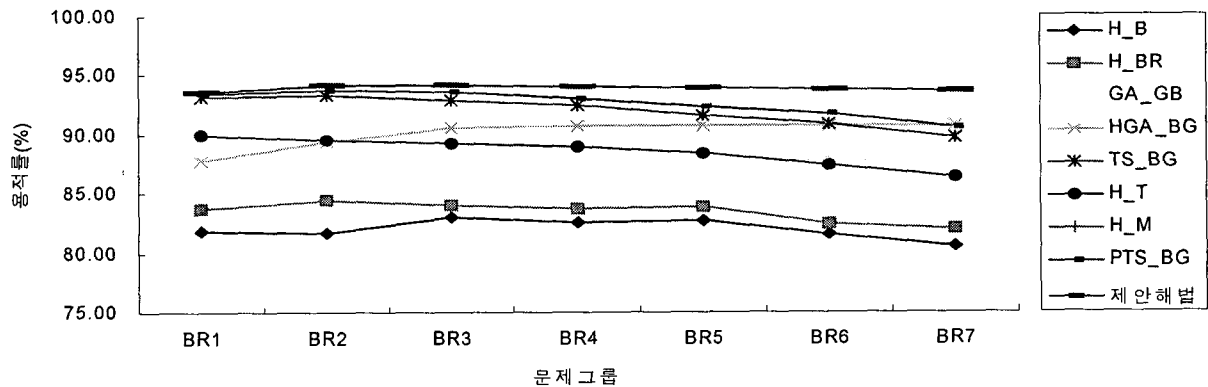
3.1 Bischoff과 Ratcliff 데이터의 실험 결과

Bischoff과 Ratcliff의 데이터를 본 해법으로 실험하고 각 그룹의 평균 용적률을 기존 논문과 비교하여 <표 1>에 보인다. 각 셀의 숫자는 해당 그룹에 속한 100개의 문제의 평균 용적률(%)을 의미한다. 그리고 문제 번호의

<표 1> Bischoff과 Ratcliff 데이터의 컨테이너 용적을 비교

문제번호	H_B	H_BR	GA_GB	TS_BG	H_T	HGA_BG	H_M	PTS_BG	제안해법
BR1(3,50.1)*	81.76	83.79	85.80	93.23	89.90	87.81	-	93.52	93.65
BR2(5,27.3)	81.70	84.44	87.26	93.27	89.60	89.40	-	93.77	94.14
BR3(8,16.8)	82.98	83.94	88.10	92.86	89.20	90.48	-	93.58	94.21
BR4(10,13.3)	82.60	83.71	88.04	92.40	88.90	90.63	-	93.05	94.10
BR5(12,11.1)	82.76	83.80	87.86	91.61	88.30	90.73	-	92.34	93.89
BR6(15,8.8)	81.50	82.44	87.85	90.86	87.40	90.72	-	91.72	93.82
BR7(20,6.5)	80.51	82.01	87.68	89.65	86.30	90.65	-	90.55	93.65
평균	81.97	83.45	87.51	91.98	88.51	90.60	88.75	92.70	93.92

-: 기존 논문에서 구할 수 없음
 *: 상자 종류 수, 상자 당 적재할 평균 개수를 의미함



<그림 7> Bischoff과 Ratcliff 데이터의 컨테이너 용적을 비교 그래프

괄호 안 2개의 숫자는 각각 상자 종류 수와 상자 종류 당 평균 개수를 의미한다. 아래쪽 문제로 갈수록 상자 종류는 많아지고 평균 개수는 적어져서 강혼합종류의 성격을 보인다.

이 결과를 보면 본 해법의 결과가 모든 다른 연구의 것보다 좋은 결과를 보인다. 그리고 700문제의 전체 평균 용적률(93.92)도 다른 결과에 비해 월등히 높은 것을 알 수 있다.

<그림 7>은 평균 용적률의 비교를 그래프로 표시한 것이다. 이는 각 그룹의 용적률 변화를 알기 위한 것으로, 다른 해법의 결과는 대체적으로 BR7로 갈수록 용적률이 낮아지는 경향을 보인다. 이는 뒤쪽의 문제가 강혼합종류를 포함하고 있어서 이러한 경우에 대해 효율적이지 못하다는 것을 의미한다. 이에 반해 본 해법은 이러한 경우에 대해서도 좋은 결과를 보이는 것을 알 수 있다.

3.2 Loh와 Nee 데이터의 실험 결과

Loh와 Nee의 데이터를 본 해법으로 실험하고 가 문제

의 용적률을 기존 논문과 비교하여 <표 2>에 보인다. 각 셀의 숫자는 해당 문제의 컨테이너 용적률을 의미한다. 이 문제에 대해서도 본 해법의 결과가 모두 좋았으며 특히 LN02의 결과가 월등히 뛰어난 것을 알 수 있다.

4. 결 론

본 연구는 크기가 다른 3차원 직육면체 상자를 고정된 크기를 가진 한 대의 컨테이너에 적재하는 3차원 컨테이너 적재 문제를 풀기 위한 새로운 발견적 해법을 제시하였다.

이 해법은 컨테이너의 공간을 일정한 공간분할 전략에 의해 분할하고 여기에 동일한 종류의 상자와 동일한 방향으로 구성된 적재블록으로 채운다. 이 방법을 연속적으로 적용하여 초기 적재패턴을 생성한 후 적재블록을 결정할 때 함께 생성된 대안블록을 초기 적재패턴의 패턴 결정 공간에 다시 적재하여 초기 적재패턴을 개선하였다.

본 해법을 715개의 데이터에 대해 지금까지 연구된 9

<표 2> Loh와 Nee 데이터의 컨테이너 용적률 비교

문제번호	H_N	H_B	H_BR	GA_GB	TS_BG	H_T	HGA_BG	H_M	PTS_BG	제안 해법
LN01	62.50	62.50	62.50	62.50	62.50	62.50	62.50	62.50	-	62.50
LN02	80.70	89.70	90.00	89.50	96.60	93.90	89.80	90.80	-	97.78
LN03	53.40	53.40	53.40	53.40	53.40	53.40	53.40	53.40	-	53.40
LN04	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00	-	55.00
LN05	77.20	77.20	77.20	77.20	77.20	77.20	77.20	77.20	-	77.20
LN06	88.70	89.50	83.10	91.10	96.20	91.60	92.40	87.90	-	96.55
LN07	81.80	83.90	78.70	83.30	84.70	84.70	84.70	84.70	-	84.70
LN08	59.40	59.40	59.40	59.40	59.40	59.40	59.40	59.40	-	59.40
LN09	61.90	61.90	61.90	61.90	61.90	61.90	61.90	61.90	-	61.90
LN10	67.30	67.30	67.30	67.30	67.30	67.30	67.30	67.30	-	67.30
LN11	62.20	62.20	62.20	62.20	62.20	62.20	62.20	62.20	-	62.20
LN12	78.50	78.50	78.50	78.50	78.50	78.50	78.50	78.50	-	78.50
LN13	84.10	82.30	78.10	85.60	85.60	85.10	85.60	85.60	-	85.60
LN14	62.80	62.80	62.80	62.80	62.80	62.80	62.80	62.80	-	62.80
LN15	59.60	59.60	59.60	59.60	59.50	59.60	59.50	59.60	-	59.60
평균	69.00	69.50	68.60	69.90	70.90	70.34	70.10	69.90	70.90	70.96
최적해 개수	11	10	11	12	13	13	13	13	13	13

-: 기존 논문에서 구할 수 없음

개의 기존 해법과 비교한 결과 모든 데이터에서 기존 연구보다 우수한 결과를 보였다. 특히 종류가 많고 각 종류의 개수가 적은 상자를 혼합 적재하는 강혼합종류 적재 문제에서 월등한 우의를 보여 본 해법의 우수성을 증명하였다.

참고문헌

- [1] Bischoff, E.E., Janetz, F., and Ratcliff, M.S.W., "Loading Pallets with Non-identical Items", *European Journal of Operational Research*, 84(3), 681-692, 1995.
- [2] Bischoff, E.E. and Ratcliff, M.S.W., "Issues in the Development of Approaches to Container Loading", *International Journal of Management Science*, 23(4), 377-390, 1995.
- [3] Bischoff, E.E. and Ratcliff, M.S.W., "Loading Multiple Pallets", *Journal of the Operational Research Society*, 46(11), 1322-1336, 1995.
- [4] Bortfeldt, A., "A Genetic Algorithm for the Container Loading Problem", *Proceedings of the Conference on Adaptive Computing and Information Processing London*, 2, 25-32, 1994.
- [5] Bortfeldt, A. and Gehring, H., "Applying Tabu Search to Container Loading Problems", *Operations Research Proceedings 1997*, Springer, Berlin, 533-538, 1998.
- [6] Bortfeldt, A. and Gehring, H., "A Hybrid Genetic Algorithm for the Container Loading Problem", *European Journal of Operational Research*, 131, 143-161, 2001.
- [7] Bortfeldt, A., Gehring, H., and Mack, D., "A Parallel Tabu Search Algorithm for Solving the Container Loading Problem", *Parallel Computing*, 29, 641-662, 2003.
- [8] Davies, A. P. and Bischoff, E. E., "Weight Distribution Considerations in Container Loading", *European Journal of Operational Research*, 114(3), 509-527, 1999.
- [9] Dyckhoff, H., "A Typology of Cutting and Packing Problems", *European Journal of Operational Research*, 44, 145-159, 1990.
- [10] Gehring, H. and Bortfeldt, A., "A Genetic Algorithm for Solving the Container Loading Problem", *International Transactions in Operational Research*, 4(5-6), 401-418, 1997.
- [11] Gehring, H., Menschner K., and Meyer, M., "A Computer Based Heuristic for Packing Pooled Shipment Containers", *European Journal of Operational Research* 44(2), 277-288, 1990.
- [12] George, J.A. and Robinson, D.F., "A Heuristic for Packing Boxes into a Container", *Computers & Operations Research*, 7, 147-156, 1980.
- [13] Eley, M., "Solving Container Loading Problems by Block Arrangement", *European Journal of Operational Research*, 141, 393-409, 2002.
- [14] Martello, S., Pisinger, D., and Vigo, D., "The Three-Dimensional Bin Packing Problem", *Operations Research*, 48, 256-267, 2000.
- [15] Loh, H.T. and Nee, A.Y.C., "A Packing Algorithm for Hexahedral Boxes", *Proceedings of the Industrial Automation 92 Conference*, Singapore, 115-126, 1992.
- [16] Ngoi, B.K.A., Tay, M.L., and Chua, E.S., "Applying Spatial Representation Techniques to the Container Loading Problem", *International Journal of Production Research*, 32(1), 111-123, 1994.
- [17] Pisinger, D., "Heuristics for the Container Loading Problem", *European Journal of Operational Research*, 141, 382-392, 2002.
- [18] Scheithauer, G., "Algorithms for the Container Loading Problem", *Operational Research Proceedings*, 1991, 445-452, 1992.
- [19] Terno, J., Scheithauer, G., Sommerweiß, U., and Riehme, J., "An Efficient Approach for the Multi-pallet Loading Problem", *European Journal of Operational Research*, 123(2), 372-381, 2000.