

3차원 물품 적재에 관한 마이크로 컴퓨터용 프로그램 개발*

Development of a Micro-Computer based Program for the Three Dimensional Packing Problem

황 학** · 이관재*** · 선지웅**

Hark Hwang** · Kwan Jae Lee*** · Ji Ung Sun**

Abstract

This paper presents a heuristic solution for the three dimensional container loading problem with boxes which are different in size. The container loading problem is formulated as a zero-one mixed integer programming model whose objective is to maximize the container utilization. Due to its problem complexity, we propose a heuristic based on layered loading technique. Also, presented is a way to achieve the weighting balance of the stacked boxes. A micro-computer based program is developed which generates container loading pattern provided by the proposed heuristic. The results are displayed by computer-graphic. To evaluate the performance of the proposed heuristic, computational experiments are performed. The results indicate that in most of the cases the proposed heuristic performs better than an existing heuristic in terms of the container utilization.

1. 서론

오늘날의 기업들은 대외개방의 가속화, 고객요구의 다양화 및 신속성의 추구, 기업간의 경쟁심화, 환경문제 등 매우 동적인 경영

환경에 처해 있다. 많은 기업들은 이러한 경영 환경의 변화에 대처하고자 기업 구조의 다각화, 고도화, 자동화 등을 추구하고 있다. 특히, 과거 생산과 판매를 연결하는 징검다리 역할로서만 인식되었던 물류에 대한 개념

* 본 연구는 산학협동재단(과제 번호:산학 제 95-54호)의 지원에 의해 수행된 연구과제임

** 韓國科學技術院 産業工學科

*** LG-EDS CIM 사업부

이 효율적인 관리를 통해 불필요한 비용을 크게 절감할 수 있는 영역으로 인식됨에 따라 물류 시스템에 대한 합리화의 중요성이 과거 어느 때보다 부각되고 있다. 또한 많은 기업들이 세계화 전략에 기초하여 국내로부터 KD(Knock Down) 부품을 공급받아 해외에서 완제품을 조립하는 현지 생산공장의 설립을 추진하고 있는 현실을 감안할 때 부품의 포장 및 수송과 관련된 물류 부문에 대한 지속적인 연구의 필요성이 요구되고 있다.

최근 상품 수송에 있어서 육상, 해상, 공중의 대부분 수송 경로에서 광범위하게 이용되고 있는 컨테이너에 의한 수송 방법을 컨테이너화(Containerization)라고 부르며 이러한 컨테이너의 도입은 보관, 하역, 수송 등과 관련된 물류 관리 분야에서 그 효과가 높게 평가되고 있다. 특히 수출상품의 해상수송과 항공수송에서는 완전한 컨테이너화 체계가 확립되어 육상수송과의 합동일관수송이 실현되고 있다. 컨테이너 단위로 화물을 운송함으로써 얻을 수 있는 장점으로는 화물이 파손 내지는 오손될 가능성의 감소, 하역작업에 있어서 기계화에 의한 일괄 하역의 실현, 수송 포장에 수반하는 노무비의 절감, 신속한 화물반입을 통한 하역비의 절감, 컨테이너 단위의 보관을 통한 보관비의 절감 등을 들 수 있다.

본 연구에서는 물류 분야 중 포장 및 수송과 관련하여 활용도가 높은 컨테이너(Container)의 효율적인 적재 방안을 다루고자 한다. 기존의 연구 결과를 토대로 컨테이너로 표시되는 3차원 공간에 다양한 규격을 가진 Box 형태의 제품을 적재하고자 할 때 적재 물품간 하중 분포의 균형을 취하면서 최대

용적률을 추구하는 적재 방안을 연구하고, 이 결과를 관련 산업체에서 쉽게 활용할 수 있도록 마이크로 컴퓨터용 소프트웨어를 개발하는 데 그 목적을 둔다.

팔레트 및 컨테이너에 물건을 적재하는 1-차원과 2-차원 문제에 대하여 그동안 여러 사람들이 많은 연구를 진행하였다. Golden[8]은 1-차원상의 막대자르기 문제(Cutting Stock Problem)를 다루었고, Gilmore[7], Christofides[3] 등은 잘라질 평면의 크기와 수에 제약을 가할 수 있는 2-차원상의 막대자르기 문제를 해결할 수 있는 알고리즘을 제시하였다. 또한 단일 물품을 일정한 크기의 팔레트에 효율적으로 적재시키기 위한 팔레트 적재패턴을 구하는 문제도 막대자르기 문제와 비슷한 맥락에서 연구되었다. Steudel[11], Smith[12], Hodgson[13] 등은 2-차원상의 팔레트 적재문제(Pallet Loading Problem)를 동적계획법(Dynamic Programming)과 정수계획법(Integer Programming) 등에 기초를 둔 발견적 기법(Heuristics)을 제시, 문제를 해결하고 있다. Tsai et. al[14]은 2-차원 팔레트 적재문제에 있어서, 여러가지 물품을 적재하는 문제를 다루는 선형계획법(Linear Programming) 모형을 제시하고 있다.

그러나 3-차원 팔레트 및 컨테이너 적재 문제에 관해서는 현재까지 활발한 연구가 진행되지 못한 실정이다. Tsai et al.[15]는 한 대의 팔레트에 다양한 Box 크기를 고려한 3-차원 팔레트 적재문제를 Mixed 0-1 정수 계획법 모델로 개발하였다. 이 모델의 해는 각 Box의 적재된 수와 팔레트 내부의 점으로 표시되는 Box의 적재 위치를 제공한다. 그러나 Box의 회전은 고려하지 못한 단점을 갖고 있

다. Chen *et al.*[2]은 다수의 컨테이너에 다양한 Box 크기를 고려한 3-차원 팔레트 적재 문제를 역시 Mixed 0-1 정수 계획법 모델로 개발하였다. 이 논문에서는 각 Box의 회전을 고려하였으며, 다른 특수한 상황(하중 분포의 균형, 컨테이너의 최적 깊이 결정)을 모델의 변형을 통해서 설명하고 있다.

위의 두 가지 수리적 모형은 문제의 최적해를 보장해주지만 Box의 갯수가 커짐에 따라 현실적인 시간안에 문제의 해를 제공하지 못한다. 따라서 빠른 시간안에 비교적 좋은 해를 제공하는 발견적 해법 개발의 필요성이 대두된다. Hodgson[13]은 Layered-Pallet-loading과 Stacked-pallet-Loading의 개념을 최초로 소개했다. 또한 IPLS(Interactive Pallet Loading System)라 불리는 시스템을 개발한 바 있다. Han *et al.*[10]은 크기가 모두 동일한 Box를 각 단계(iterations)마다 L-shaped Layer의 형태로 적재하는 발견적 기법을 제시했다. George and Robinson[5]은 각 형태의 Box 갯수와 치수가 알려진 3-차원 컨테이너 적재 문제에 대한 발견적 기법을 제시했다. Gehring *et al.*[6]

또한, George and Robinson과 비슷한 개념의 발견적 기법을 제시하였다. 그러나 그들은 각 Box 형태의 제약이 없고 Box의 회전을 고려한 적재 방안을 제시하였으며 하중 분포의 균형화 방안에 관해서도 언급을 하였다. 그러나 구체적인 균형화 방안을 소개하지는 않았다. Abdou *et al.*[1]은 팔레트의 적재 문제를 4가지로 분류하였으며 적재에 영향을 미치는 요소들에 관해 기술하였다. 또한 적재의 안정성을 고려한 Layer순서를 발견하였다. Haessler and Talbot[9]는 저밀도 물품을 적재하는 Load Diagram을 제시하였다. 특히 물품 적재의 방안으로 Stacked-Loading 기법을 사용하였다. 이상에서 제시된 문헌을 그 성격에 따라 분류하면 아래 표 1과 같다.

표 1에서 Type 1은 모든 Box의 크기가 동일한 경우를 말하고 Type 2는 모든 Box들의 높이는 동일하나 길이와 너비가 다른 경우를 말한다. 그리고 Type 3은 동일한 크기를 갖는 Box들이 여러 개 존재하여 그룹핑을 할 수 있는 경우이고 Type 4는 모든 Box들의 크기가 다른 경우를 말한다. 본 연구에서 다양

표 1. 3-차원 컨테이너 적재 문제의 분류

Math. Model	Permission of Box rotation		Chen <i>et al.</i> [2]
	Non-Permission of Box rotation		Tsai <i>et al.</i> [15]
Heuristic	Stacked-Loading+		Haessler & Talbot[9]
	Layered Loading+	TYPE 1	Han <i>et al.</i> [10]
		TYPE 2	Abdou <i>et al.</i> [1]
		TYPE 3	George & Robinson[5]
		TYPE 4	Gehring <i>et al.</i> [6]

주) + Hodgson, IIE Transaction (1982)

한 Box들의 크기를 고려하고 Box들의 회전을 고려한 Type 4의 문제를 대상으로 정형적인 모형을 개발하고, 빠른 시간안에 문제의 해를 도출할 수 있는 발견적 기법을 제시하고자 한다. 또한 컨테이너 적재 문제에 있어서 또 하나의 중요한 부분을 이루는 컨테이너의 하중 분포의 균형화 방안에 관해서도 구체적인 대안을 제시하고자 한다.

2. 수리적 모델

여러 가지 다양한 크기의 Box가 주어지고 한 대의 컨테이너에 이 Box를 가능한 많이 적재하려 한다. 이때 어떤 방법으로 적재하여야 컨테이너의 용량을 충분히 활용하면서 부득이 사용되지 못하는 공간을 최소화할 수 있는가의 문제가 제기된다. 상기 문제에 대해 기존의 연구 결과를 토대로 다품목의 적재 패턴을 Mixed zero-one 정수 계획법으로 모형화할 수 있다. 여기서는 Box의 회전을 고려한 적재 패턴을 개발하고자 하는 것이며 컨테이너의 하중 분포는 별도로 고려하게 된다. 수리적 모형을 개발하기 위한 중요 결정 변수와 시스템 parameters를 정의하면 다음과 같다.

Parameters

m : 적재 대기 상태 Box의 총 갯수

M : 임의의 큰 수

L, W, H : 컨테이너의 길이(X축 방향), 너비(Y축 방향), 높이(Z축 방향)

(l_i, w_i, h_i) Box i 의 세 가지 치수 ($l_i \geq w_i \geq h_i$)

결정 변수

(x_i, y_i, z_i) : Box i 가 컨테이너에 적재되었을 때 전-좌-하(front-left- bottom) 모서리의 좌표

(l_{xi}, l_{yi}, l_{zi}) : 적재된 Box i 의 치수 l_i 가 X-, Y-, Z-축 중 어디에 평행한가를 나타내는 0-1변수. 이때 평행하면 1, 그렇지 않으면 0으로 각각 표시한다. 예: (0,1,0)

(W_{xi}, W_{yi}, W_{zi}) : 적재된 Box i 의 치수가 X-, Y-, Z-축 중 어디에 평행한가를 나타내는 0-1변수. 이때 평행하면 1, 그렇지 않으면 0으로 각각 표시한다.

(h_{xi}, h_{yi}, h_{zi}) : 적재된 Box i 의 치수가 X-, Y-, Z-축 중 어디에 평행한가를 나타내는 0-1변수. 이때 평행하면 1, 그렇지 않으면 0으로 각각 표시한다.

U_i : Box i 가 컨테이너에 적재가 되었는지를 나타내는 변수. 적재되었으면 1, 그렇지 않으면 0

L_{ik} : Box i 가 Box k 의 좌측에 있으면 1, 그렇지 않으면 0

R_{ik} : Box i 가 Box k 의 우측에 있으면 1, 그렇지 않으면 0

F_{ik} : Box i 가 Box k 의 앞측에 있으면 1, 그렇지 않으면 0

B_{ik} : Box i 가 Box k 의 뒤측에 있으면 1, 그렇지 않으면 0

A_{ik} : Box i 가 Box k 의 위측에 있으면 1, 그렇지 않으면 0

G_{ik} : Box i 가 Box k 의 밑측에 있으면 1, 그렇지 않으면 0

$$\text{Min } L*W*H - \sum_{i=1}^m (l_i*w_i*h_i)*U_i$$

$$\text{s.t } x_i + l_i*l_{xi} + w_i*w_{xi} + h_i*h_{xi} \leq x_k + (1-L_{ik})M$$

$$\text{for all } i, k \ i < k \quad (1)$$

$$x_k + l_k^* l_{xk} + w_k^* w_{xk} + h_k^* h_{xk} \leq x_i + (1 - R_{ik})M$$

$$\text{for all } i, k \ i < k \quad (2)$$

$$y_i + w_i^* w_{yi} + l_i^* l_{yi} + h_i^* h_{yi} \leq Y_k + (1 - F_{ik})M$$

$$\text{for all } i, k \ i < k \quad (3)$$

$$y_k + w_k^* w_{yk} + l_k^* l_{yk} + h_k^* h_{yk} \leq y_i + (1 - B_{ik})M$$

$$\text{for all } i, k \ i < k \quad (4)$$

$$z_i + h_i^* h_{zi} + w_i^* w_{zi} + l_i^* l_{zi} \leq z_k + (1 - G_{ik})M$$

$$\text{for all } i, k \ i < k \quad (5)$$

$$z_k + h_k^* h_{zk} + w_k^* w_{zk} + l_k^* l_{zk} \leq z_i + (1 - A_{ik})M$$

$$\text{for all } i, k \ i < k \quad (6)$$

$$L_{ik} + R_{ik} + F_{ik} + B_{ik} + G_{ik} + A_{ik} \geq 1$$

$$\text{for all } i, k \ i < k \quad (7)$$

$$x_i + l_i^* l_{xi} + w_i^* w_{xi} + h_i^* h_{xi} \leq L^* U_i$$

$$\text{for all } i, k \ i < k \quad (8)$$

$$y_i + w_i^* w_{yi} + l_i^* l_{yi} + h_i^* h_{yi} \leq W^* U_i$$

$$\text{for all } i \quad (9)$$

$$z_i + h_i^* h_{zi} + w_i^* w_{zi} + l_i^* l_{zi} \leq H^* U_i$$

$$\text{for all } i \quad (10)$$

$$\sum_{i=1}^m (l_i^* w_i^* h_i^*) U_i \leq L^* W^* H$$

$$\text{for all } i \quad (11)$$

$$x_i, y_i, z_i \geq 0 \quad \text{for all } i$$

$$l_{xi}, l_{yi}, l_{zi}, w_{xi}, w_{yi}, w_{zi}, h_{xi}, h_{yi}, h_{zi}, L_{ik}, R_{ik},$$

$$F_{ik}, B_{ik}, G_{ik}, A_{ik}, U_i + 0 \text{ or } 1$$

적재할 Box들의 면은 컨테이너면에 직각으로 놓여진다고 가정하고 컨테이너의 전-좌-하(front-left bottom) 모서리의 좌표를 원점으로 생각하여, 이미 기술된 컨테이너 적재 문제를 Mixed zero-one 수리 모델로 정형화하면 위와 같다.

위 모형의 해는 한 개의 컨테이너에 컨테이너 용적률을 최대한 활용할 수 있도록 적재된 Box종류 및 적재 위치를 의미한다. 제

약식 (1)에서 (6)까지는 Box들이 컨테이너 안에서 서로 겹치지 않아야 함을 나타낸다. Box 간의 겹침에 관한 제약은 대상 Box들이 컨테이너에 적재된다는 전제하에 성립하는데 이는 제약식 (7)에 나타나져 있다. 제약식 (8)에서 (10)까지는 Box가 컨테이너에 놓여진 상태에서 세가지 각 치수가 대응하는 컨테이너 치수보다 클 수가 없다는 것을 나타내고, 마지막으로 제약식 (11)은 적재되는 Box들의 부피의 총합은 컨테이너의 부피보다 클 수가 없음을 나타낸다. 위의 최적화 모형은 LINDO, MathPro/XPRESS-MP, MPS/MIP와 같은 상용 패키지(package)를 사용하여 풀 수 있다. 그러나 이 모델은 Box의 숫자가 커짐에 따라 해를 찾기가 거의 불가능하여지기 때문에 발견적 해법 개발의 필요성이 대두된다.

3. 3-차원 컨테이너 적재 문제의 발견적 기법

3.1 가정 및 기본 사항

3-차원상의 컨테이너 적재문제에 관한 기존의 연구 중 George[5]와 Gehring[6]은 하나의 컨테이너에 다양한 크기의 Box를 적재하는 발견적 기법을 제시한 바 있다. 본 연구에서는 위 두 문헌에서 제시된 방법론을 참고로 하여 보다 개선된 또한, 컨테이너 하중을 균형화시킬 수 있는 적재 방안을 제시한다. 본 연구에서 제시하는 적재방안은 그림 1에서 보듯이 기본적으로 컨테이너의 길이 방향으로 Layer를 형성해 가면서 Box를 적재하는 Layered-pallet-loading 방식을 따르고 있다. 컨테이너 적재 문제의 발견적 해법을 개발하기 위해 사용된 가정과 Notations은 아래와 같다.(그림 1 참고)

가 정

- ① 각 Box는 컨테이너 각 면에 수직 혹은 수평으로 적재되는 한 어떤 방향으로든 적재 가능하다.
- ② 어떤 Box이든지 서로 인접하여, 또한 상하로 적재될 수 있다.
- ③ 컨테이너 내부의 Box 적재 장소에 대한 제한 조건이 없다.

적재하는 것이 컨테이너 용적률을 높이는 데 유리하기 때문이다.

- ② Box들은 한 Layer안에서만 적재가 가능하며, 이웃하는 Layer에 걸쳐서 적재하는 것은 허용하지 않는다.
- ③ 각 Layer의 깊이(Depth)는 Layer에 적재되는 첫번째 Box의 깊이에 의해서 결정된다. 이 Box를 LDB(Layer Determining

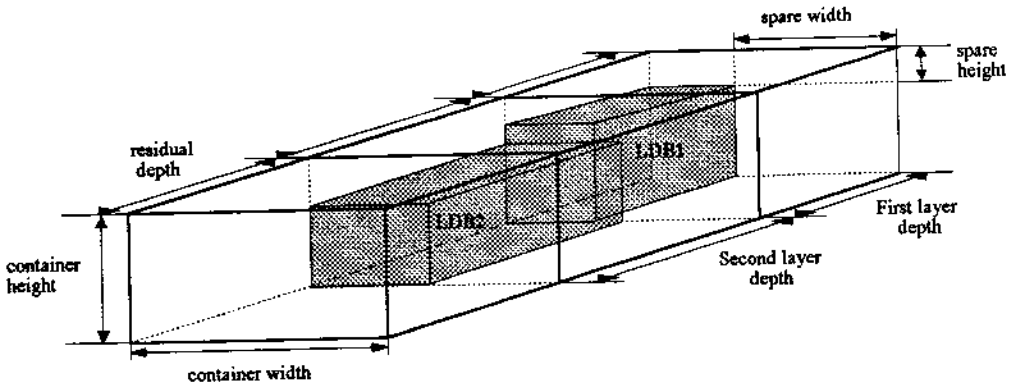


그림 1. Box를 적재한 컨테이너

Notations

- ① $Minh$: 조사 사점의 Box리스트에 속하는 Box의 세 가지 치수 중 제일 작은 치수 = $\min_{i \in \text{Box list}} h_i$
- ② LD : Layer 깊이(Depth)
- ③ LH : Layer 높이(Height)
- ④ LW : Layer 너비(Width)
- ⑤ LDB : Layer 깊이를 결정하는 Box (Layer Determining Box)

본 해법을 개발하기 위해 채택한 기본적인 사항을 제시하면 아래와 같다.

- ① Box들을 부피(Volume)가 큰 순서대로 정렬하여 Box리스트를 작성한다. Box리스트의 첫번째 요소부터 순서대로 컨테이너에 적재되는데 이는 부피가 큰 Box부터

Box라고 부른다. LDB의 회전을 고려하면 최대 3가지 다른 깊이를 가진 Layer를 만들 수 있다.

- ④ 빈 공간(Spare Space)에는 공간의 부피를 가장 잘 만족시키면서 적재 가능한 Box를 적재하게 된다. 빈 공간에 Box를 적제한 뒤에는 공간 절단을 통해서 Box의 옆, 앞, 위에 새로운 공간이 생성된다. (그림 5참고)
- ⑤ Layer의 LDB 회전을 고려하게 되면 Layer 내의 빈 공간의 모양이 달라져서 새로운 적재 패턴이 형성된다. 본 연구에서는 계산속도의 부담을 고려하여 편의상 첫번째~세번째 Layer에 대해서 각각의 LDB 회전을 고려해줌으로써 최대 6³가지의 적

재 상황을 고려하게 된다.

3.2 발견적 기법의 개괄적 흐름

발견적 기법의 개괄적 흐름을 살펴보면, 먼저 컨테이너와 Box들의 Data들을 입력받은 후 Box들의 부피를 계산하게 되고 이를 다시 부피(Volume)가 큰 순서대로 정렬한다. 입력된 컨테이너와 Box들의 Data를 사용하여 컨테이너에 Box를 적재하게 되는데 이때 Box 리스트의 첫번째 요소, 즉 부피가 가장 큰 Box로 컨테이너의 Layer의 깊이를 구성하게

된다. Layer를 구성한 다음, Layer에 Box를 적재하는 일정한 규칙(3.3.4)에 따라 적재를 하게 되며 모든 Layer들의 깊이가 결정되고, Layer내부를 적합한 Box들로 적재하게 되면 하나의 적재 패턴이 완성된다. 다른 적재 패턴을 형성하기 위해서 첫번째~세번째 Layer의 LDB회전을 고려하게 된다. Box의 깊이, 너비, 높이가 서로 상이할 수 있으므로 LDB의 회전에 의한 서로 다른 적재 패턴은 최대 6'가지가 나올 수 있다. 모든 적재 패턴을 형성한 후 그 중에서 용적률이 가장 높은 것을

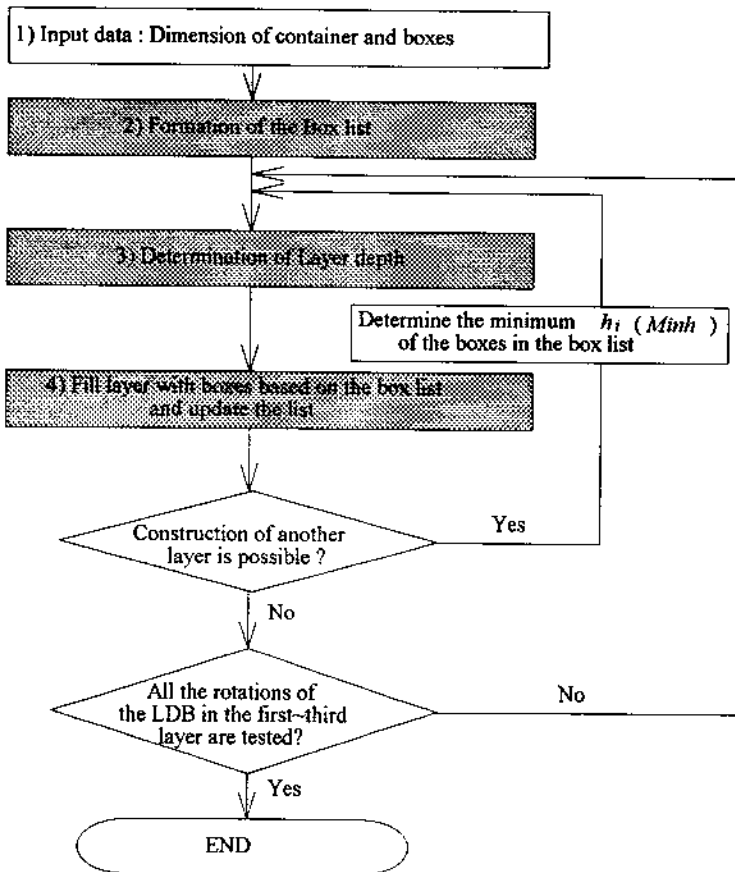


그림 2. 컨테이너 적재 방안의 개괄적 흐름도
(단계 2,3,4는 별도의 흐름도가 작성됨)

문제의 해로 선택한다.

발견적 기법의 개괄적 흐름을 흐름도(Flow Chart)로 나타내면 그림 2와 같다. 3.3에서는 그림 2에서 제시된 개괄적 흐름을 모듈(module)별로 자세히 살펴보도록 하겠다.

3.3 발견적 기법의 상세한 흐름

3.3.1 Data의 입력 (Input Data : Dimension of container and boxes)

입력 Data로 i)컨테이너의 길이, 너비, 높이
ii)Box의 경우에는 회전을 고려하게 되므로 최대, 중간, 최소 치수가 요구된다. ISO에서 규정한 표준 컨테이너의 규정을 보면 표 2와 같다.

표 2. 컨테이너 표준 규격

(단위 : m)

Container Data	길이 (Depth)	너비 (Width)	높이 (Height)
20 feet	6.09	9.14	12.19
30 feet	2.43	2.43	2.42
40 feet	2.43	2.43	2.43

3.3.2 Box번호의 indexing (Formation of the box list)

입력된 Data를 토대로 Box의 부피를 계산하여 부피가 큰 순서로 Box를 정렬하게 되며 부피가 동일할 때에는 임의로 순서를 정하게 된다. 부피 순으로 정렬된 Box 들은 가능한 한 순서대로 컨테이너에 적재되며, 적재가 된 Box는 Box 리스트에서 제거된다.

3.3.3 Layer깊이의 결정 방안(Determination of Layer depth)

이 모듈(Module)에서는 전체 컨테이너 깊

이를 얼마만큼의 깊이를 가진 Layer들로 구성할 것인가를 결정하게 된다. 컨테이너의 첫 Layer는 Box 리스트에서 첫번째 Box가 컨테이너에 적재 가능한가의 여부를 알아본 뒤 적재가 가능하다면, 그 Box의 중간 치수로 Layer의 깊이를 결정한다. Layer의 너비, 높이는 항상 컨테이너의 너비, 높이와 동일하다는 것을 그림 1에서 알 수 있다. 이렇게 결정된 첫번째 Layer에 3.3.4의 절차에 따라 Box들을 적재한 뒤 적재된 Box들을 제거하여 Box 리스트를 수정한다. 이와 동일한 방법으로, 수정된 Box 리스트에서 첫번째 Box의 중간 치수로 다음 Layer의 깊이를 결정하게 된다. Layer를 구성하는 시점에서 남아 있는 컨테이너의 깊이가 *Minh*보다 작은지의 여부를 항상 알아보게 되는데 컨테이너의 깊이가 *Minh*보다 크다면 현재의 Layer는 마지막 Layer가 아니다. 만약 *Minh*보다 작다면 현재의 Layer는 마지막 Layer이며, 남아 있는 공간은 Box 리스트의 Box들로 채울 수 없는 공간이 되므로 효율적으로 공간을 활용하기 위해서 컨테이너에 남아 있는 공간을 현재의 Layer에 통합시켜 통합된 공간의 깊이를 Layer의 깊이로 결정하게 된다. Layer깊이의 결정 방안을 흐름도 형식으로 나타내면 그림 3과 같다.

3.3.4 Layer에 Box를 적재 (Fill Layer with boxes based on the Box list)

앞에서 언급한 바와 같이 Layer에 처음으로 적재되는 Box가 LDB가 된다. 첫번째~세번째 LDB의 경우 회전을 고려하게 되어 서로 다른 6°까지의 Layer 조합을 만들 수 있으므로 상이한 적재 패턴을 생성하게 된다. 네

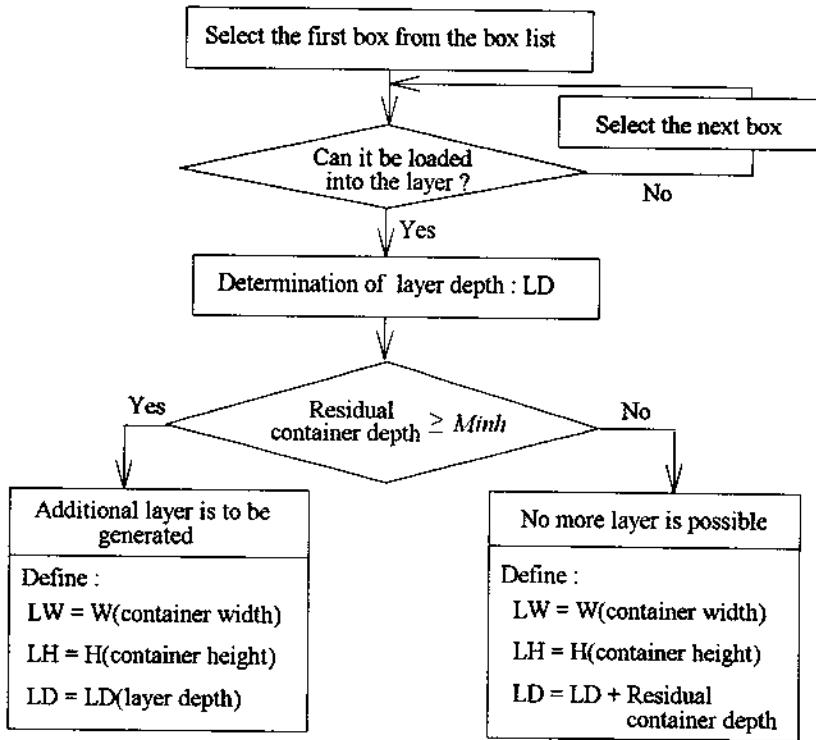


그림 3. Flowchart on 'Determination of layer depth (3)'

번째 Layer부터는 LDB가 고정된 위치로 적재가 된다. 즉 최대 치수를 높이방향으로, 중간 치수를 깊이방향으로, 최소 치수를 너비방향으로 Box를 회전시켜 Layer에 적재한다.

이와 같이 적재하는 이유는 LDB를 적재한 후 다음에 적재되는 Box들을 위해 최대한 많은 공간을 확보하기 위함이다. Layer에 LDB를 적재한 후 여분의 공간이 남게 되는데 이는 공간 절단을 통해서 각각 옆, 앞, 위의 새로운 공간으로 정의되며 이 공간들을 차례로 나열한 것을 Space리스트라고 칭한다.

Space리스트는 Layer가 결정되는 시점에서 Layer를 첫번째 요소로 삼으며 Box가 적재되면 적재된 현재의 공간(Layer)은 리스트에서

제거되고 Box적재 후 공간 생성을 통해 생성된 공간들은 리스트의 앞부분에 추가된다. 수정된 Space 리스트의 맨 앞에 있는 공간에 가장 적합한 Box를 적재하게 되면 마찬가지로 3개의 빈 공간이 형성이 되고 이 공간들은 Space 리스트의 앞부분에 추가된다. 그에 앞서 맨 앞에 있던 공간은 Box가 적재되었으므로 Space리스트에서 제거가 되고 그 이후에 있던 공간들은 새롭게 생성된 공간을 위해서 밀려나게 된다. 만약 빈 공간에 적합한 Box가 존재하지 않는다면 그 공간은 Space리스트에서 제거된다. 이와 같이 Space리스트는 후입선출(LIFO)의 개념으로 운용된다. 이러한 과정은 모든 Box들이 전부 적재되어

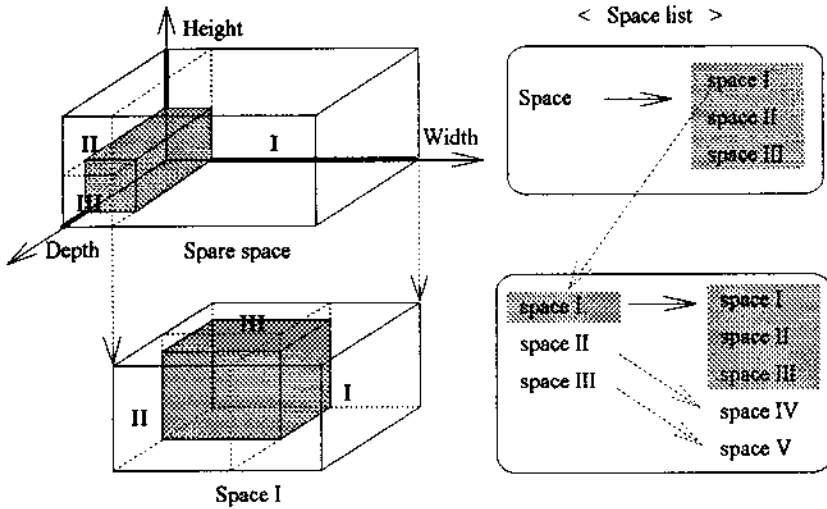


그림 4. Space리스트의 운용

서 더이상 적재할 Box가 없거나, Space 리스트에 남아 있는 Space가 없어서 현재의 Layer에 더이상의 Box적재가 불가능할 때까지 계속된다. 빈 공간에 Box가 적재되고 난 후 Space리스트의 운용 상황을 나타내면 그림 4와 같다.

또한, 빈 공간에 Box가 적재되고 난 후의 공간 생성과 우선 순위를 예제를 통해 나타내면 그림 5와 같다. 이때 공간의 가장 큰 치수와 Box의 가장 작은 치수, 공간의 가장 작은 치수와 Box의 가장 큰 치수가 대응되도록 Box를 회전시켜 적재한다. 이 또한 다음에 적재되는 Box들을 위해서 좀 더 많은 공간을 확보하고자 하는 효과적인 적재 방안이다. 공간을 확보하고자 하는 노력은 Box를 적재한 후 공간 절단을 통한 공간 생성에서도 나타나는데 그림 5에서 a,b,c 중 길이가 가장 긴 a부분의 공간을 공간 I으로 정의하고 공간 I을 최대한 살리기 위해서 깊이를 현재 공간 깊이와 동일한 10으로, 높이를 현재 공간

높이와 동일한 15로 공간 절단을 하고 두번째로 긴 b부분의 공간 II의 깊이를 10으로 절단한다. 생성된 공간 I, II, III는 순서대로 Space리스트의 앞부분에 추가되고 현재의 공간(깊이:10, 너비:20, 높이:15)은 Box가 적재되었으므로 제거된다.

빈 공간에 Box를 적재할 때 Box리스트에서 가장 부피(Volume)가 큰 Box부터 그 적재 가능성을 따져 나가게 되는데 만일 가장 큰 Box보다 부피는 작지만 두 Box의 부피 합이 가장 큰 Box의 부피보다 더 크고, 현재의 빈 공간에 적재가 가능하다면 두 Box를 현재의 빈 공간에 적재하도록 한다. 두 Box 중 부피가 큰 Box는 현재의 공간에 적재가 되고, 나머지 작은 Box는 부피가 큰 Box에 의해 생성되는 3개의 공간 중 하나에 미리 적재되는 것으로 간주한다. 그래서 그 Space(부피가 작은 Box가 적재된 공간)가 리스트의 맨 앞에 오게 되면 미리 적재된 것으로 간주된 Box(labeled Box)를 사용하게 되는 것이다.

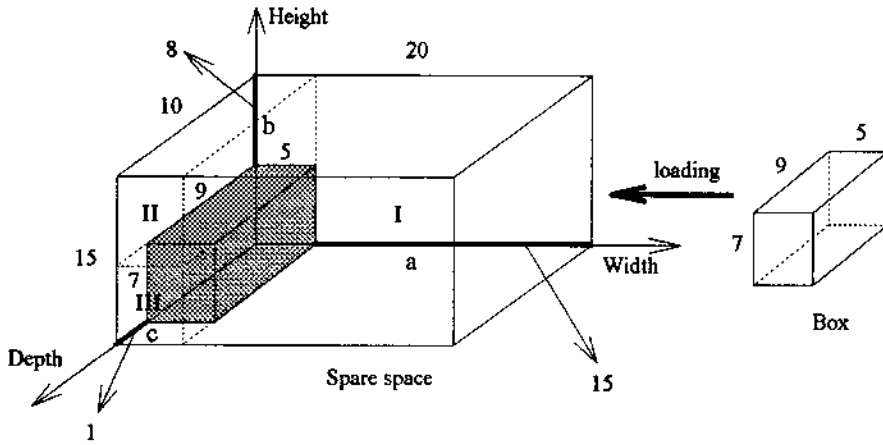


그림5. Box 적재 후 공간 생성

이는 공간의 용적률을 높이기 위해 해당 공간에 적합한 Box를 미리 내정하는 형식이다. 두 개의 Box를 빈 공간에 어떠한 방식으로 적재할 것인가는 다음과 같은 우선 순위로 한다. 두 개의 Box 중 부피가 큰 Box를 빈 공간에 적재한다. Box가 적재되면 공간 생성을 통해 생성된 공간들 중 공간 I에 부피가 작은 Box의 적재 여부를 알아본다. 만약 적재가 가능하지 않을 경우 공간 II, III에 대해서 순서대로 적재 여부를 알아본다. 그림 6에서는 부피가 큰 Box K가 적재되고 부피가 작은 Box L이 공간 II에 미리 적재된 것으로 간주되어 공간 II가 Space리스트의 맨 앞에 위치할 때 Box L이 사용되는 것이다.

Layer에 Box를 적재하는 과정을 흐름도(Flow Chart)로 나타내면 그림 7과 같다.

위의 흐름도(Flow Chart) 중 마지막 단계인 다음 빈 공간을 찾는 과정(Choose the next spare space)을 좀 더 자세히 나타내면 그림 8과 같이 보일 수 있다.

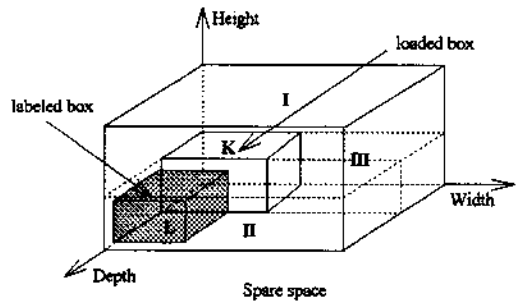


그림 6. 두 개의 Box가 적재되는 상황

3.4 하중 분포의 균형화

컨테이너의 물품 적재시 고려해야 할 중요한 요소 중 하나가 하중 분포의 균형화를 유지하면서 물품을 적재하는 것이다. 본 연구에서는 발견적 기법에서 제시된 문제의 해를 이용하여 하중 분포의 균형화를 이루는 방안을 제시하도록 한다. 발견적 기법을 통하여 컨테이너의 Layer가 구성되고 Layer i 에 적재된 Box의 하중의 합을 W_i 라 할 때의 상황을 나타내면 그림 9와 같다.

위에서 사용된 변수들은

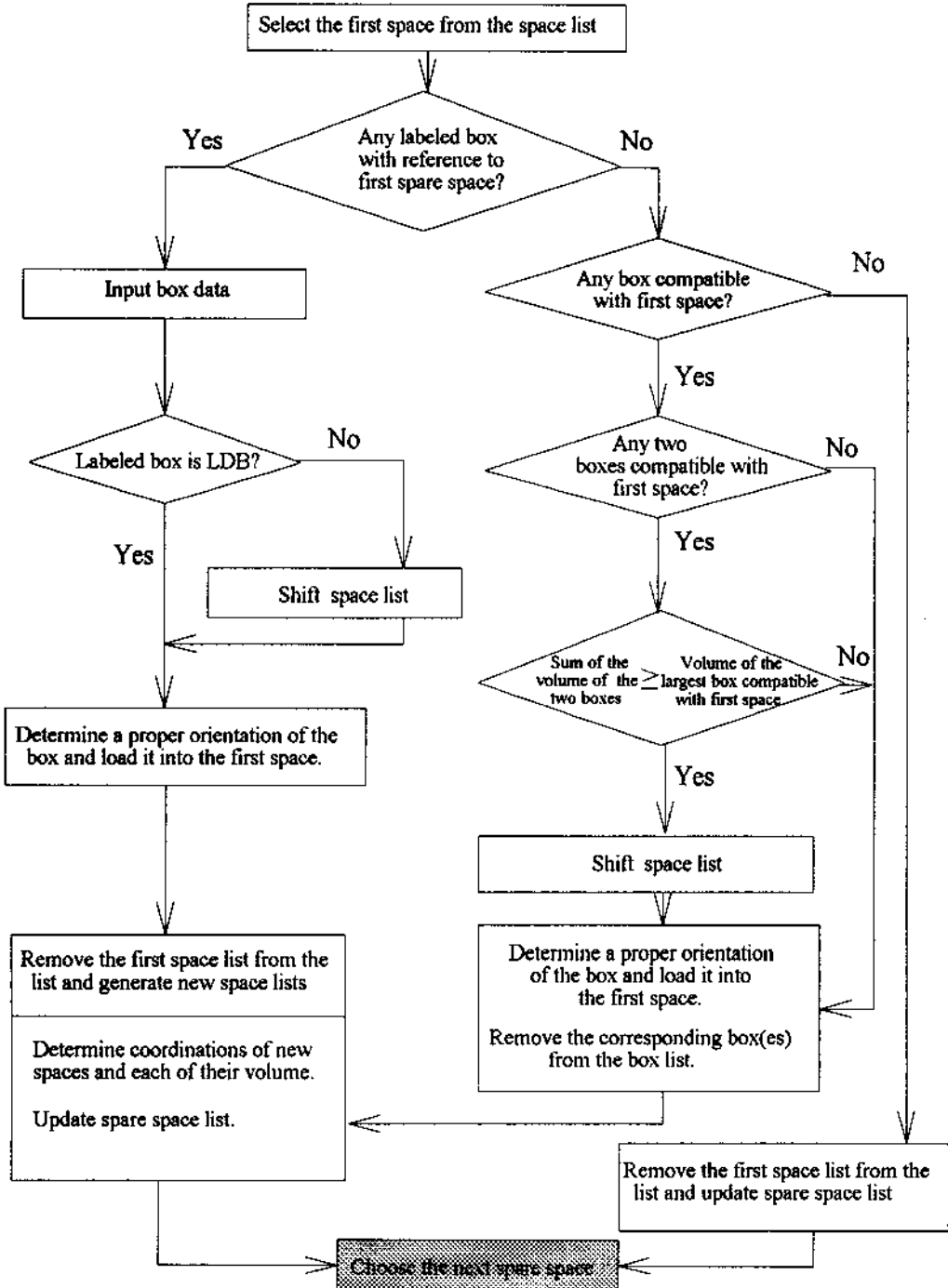


그림 7. Flowchart on 'Fill layer with boxes based on the box list(4)'

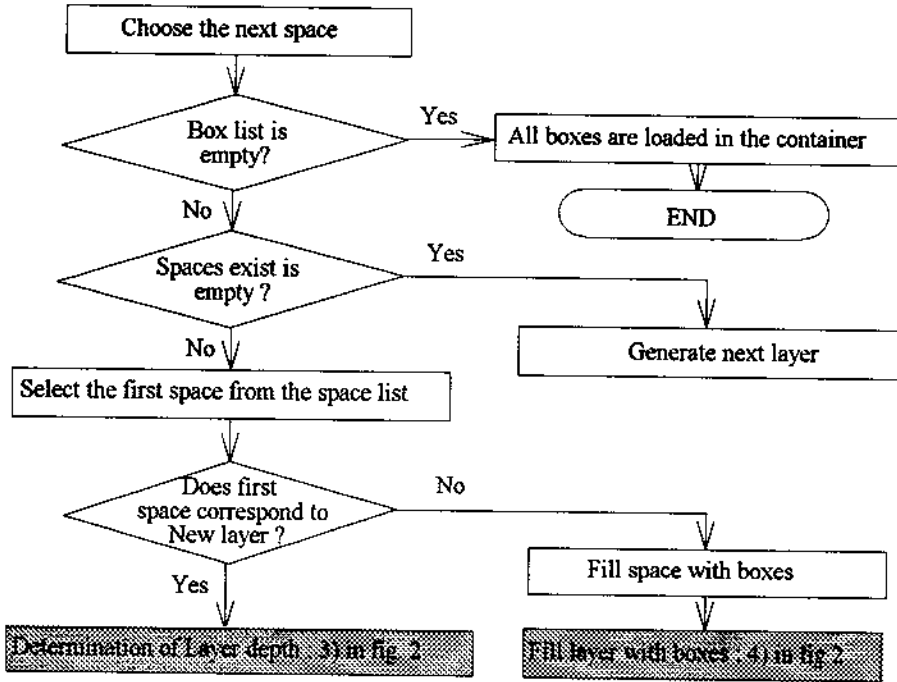


그림 8. Flowchart on 'Choose the next spare space'

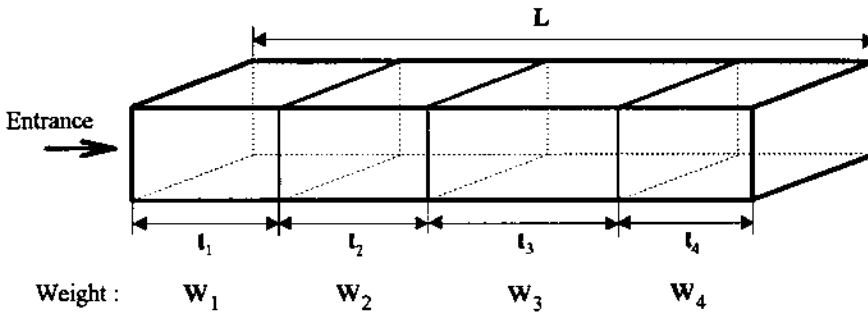


그림 9. 각 Layer 및 하층의 도시에

d_i : Box j 의 밀도(= 무게/부피)

l_i : Layer i 의 깊이

W_i : Layer i 에 적재된 Box들의 무게의 총

$$\text{합 } (W_i = \sum_{j \in \{\text{boxes in } i\text{th layer}\}} (l_i * w_j * h_j) * d_j)$$

L : 컨테이너의 길이 ($L = \sum_{i=1}^n l_i$)

n : 발견적 기법에서 제시된 Layer의 갯수

(그림 9에서는 $n=4$)

여기서 제시될 내용은 각각의 Layer 하층을 고려하여 발견적 기법에서 제시된 Layer

의 순서를 적절히 바꿔 봄으로써 전체적인 균형을 이루고자 하는 것인데 컨테이너의 전체를 통해서 하중을 최대한 균일하게 주는 Layer순서를 발견하는 Heuristic절차를 제시하면 다음과 같다.

STEP 1. 컨테이너의 완전 균형 하중 $S = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n l_i}$ 를 결정.

$W_c = L_c = 0$ 로 초기화. (W_c : Box의 누적 무게, L_c : Box의 누적 깊이)

STEP 2. $\min_{i \in \{Layer\}} |(W_c + w_i) - (L_c + l_i) * S|$ 인 Layer i 를 선택.
 $\{Layer\} = \{Layer\} - i$

STEP 3. $\{Layer\} = \phi$ 이면 STOP. Goto STEP 4. $\{Layer\} = \phi$ 이면 $W_c = W_c + w_i$, $L_c = L_c + l_i$ 로 수정. Goto STEP 2.
 STEP 4. STEP 2에서 결정된 Layer i 의 sequential index를 찾는다.

예) 위의 절차에 따라 다음의 Data를 가지고 Layer의 순서를 구해보면 표 3와 같다.

표 3. 각 Layer의 깊이(l_i)와 하중(w_i)

l_1	l_2	l_3	l_4	l_5	$\sum l_i$
10	15	8	12	5	50
w_1	w_2	w_3	w_4	w_5	$\sum w_i$
30	60	20	30	10	150

STEP 1. $S=3, W_c=L_c=0$.

STEP 2. $\min_{i \in \{Layer\}} |(W_c + w_i) - (L_c + l_i) * S|$ 가 되는 Layer는 1.
 $\{Layer\} = \{Layer\} - 1$

STEP 3. $W_c = w_1 = 30, L_c = l_1 = 10$. Goto STEP 2. 이상과 같은 절차를 반복.
 $\{Layer\} = \phi$ 이면 Stop. Goto STEP 4

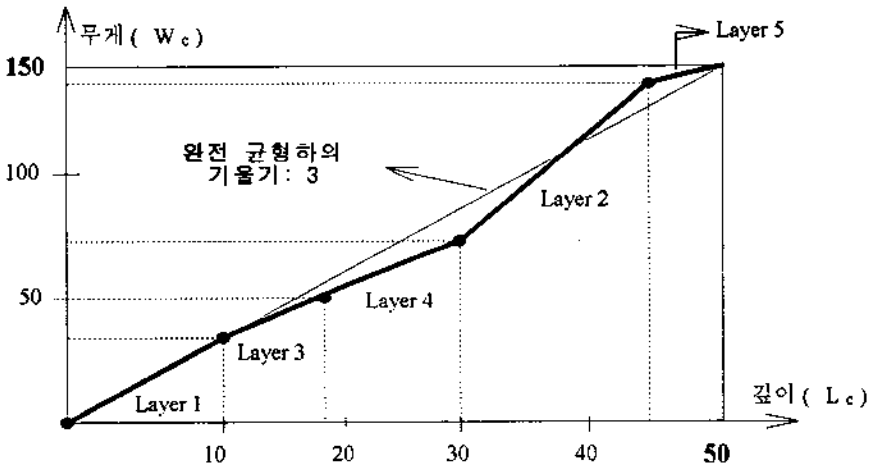


그림 10. 하중 분포의 균형을 고려한 Layer순서

STEP 4. Layer순서의 sequential index는
1-3-4-2-5.

STEP 1~3의 절차를 반복하여 구한 Layer 순서를 그림으로 나타내어 보면 그림 10과 같다.

4. 3-차원 컨테이너 적재 문제의 그래픽 표현

4.1 시스템 개요

4절에서는 3절에서 제시된 발견적 기법을 이용하여 실제 컨테이너 적재시 사용할 수 있는 컴퓨터용 프로그램을 개발하고, 이를 그래픽하게 표현하여, 사용자가 컨테이너 적재 알고리즘에 관해 지식이 없더라도 쉽게 적재 형태를 알 수 있도록 하였다. 적재 형태를 3-

화면을 제공함으로써 사용자가 큰 어려움 없이 적재 정보를 얻을 수 있도록 처리하였다. 또한 적재 과정이 Layer단위로 이루어지므로 출력되는 화면을 Layer단위로 처리하였다. 세 가지 그래픽 화면을 나타낸 것이 그림 11이다.

사용자는 입력 Data로 i)컨테이너의 깊이, 높이, 너비와 ii)적재할 Box의 최대, 중간, 최소 치수를 입력하게 된다. 입력된 Data를 이용한 프로그램의 출력은 각 Layer단위로 이루어진다. 각 Layer에 Box가 적재될 때마다 Box의 치수(깊이, 높이, 너비)가 화면에 출력되고, 적재 위치가 세 화면에 동시에 표시된다. 또한 현재 적재가 진행되고 있는 Layer에 Box의 적재가 끝나면 Layer의 적재 효율 (Volume Utilization)이 표시되고, 모든 Layer

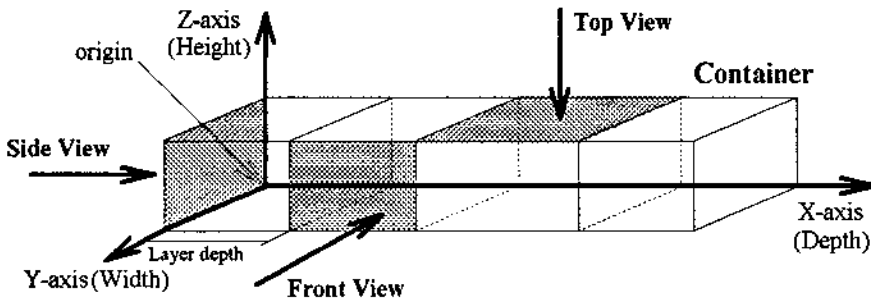


그림 11. 세 가지(Top View, Front View, Side View)그래픽 화면

차원으로 표현하는 것은 시각적인 혼란을 야기하고, 정확한 적재 정보를 얻기 어려우므로 이를 2-차원의 세 화면(Top View, Front View, Side View)으로 구성하였다. 2-차원으로 적재 형태를 표현하더라도 Box간의 겹침을 완전히 해소할 수 없어 한꺼번에 모든 Box를 적재한 화면을 보여주지 않고, Box하나가 컨테이너에 적재될 때마다 적재 위치에 대한

의 적재가 끝나면 전체 컨테이너의 적재 효율(Volume Utilization)이 표시된다. 하나의 대안이 완성되면 LDB 회전에 기인한 또 다른 적재 대안을 사용자가 선택해서 알아볼 수 있도록 처리하였다. 초기 그래픽 화면을 제시하면 그림 12와 같다.

그림 12의 Top_View는 현재 적재할 Layer의 깊이×너비 단면을 위에서 내려다 본 상

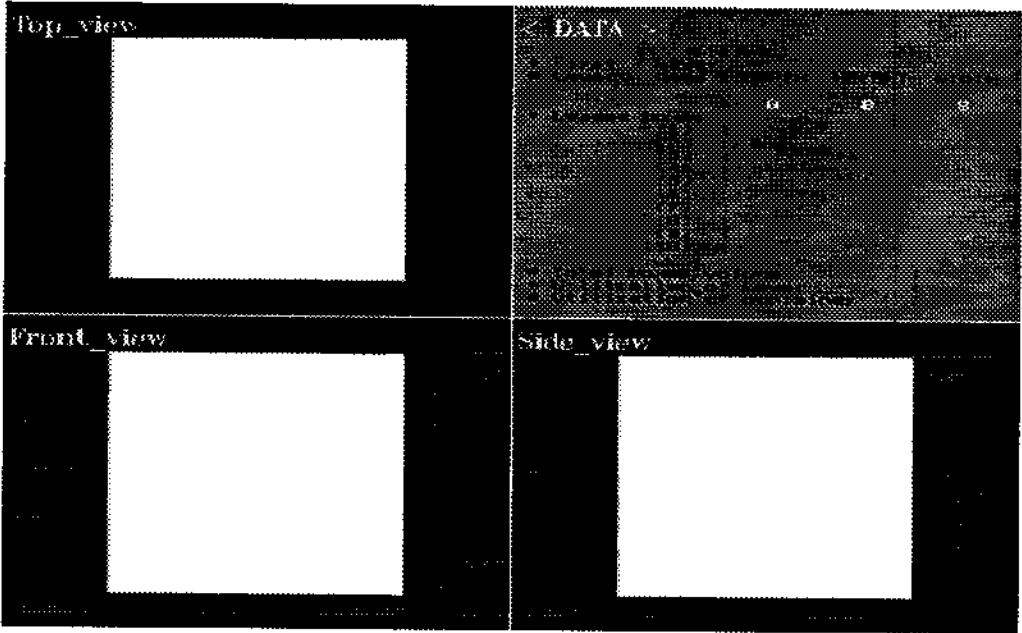


그림 13. 초기 화면

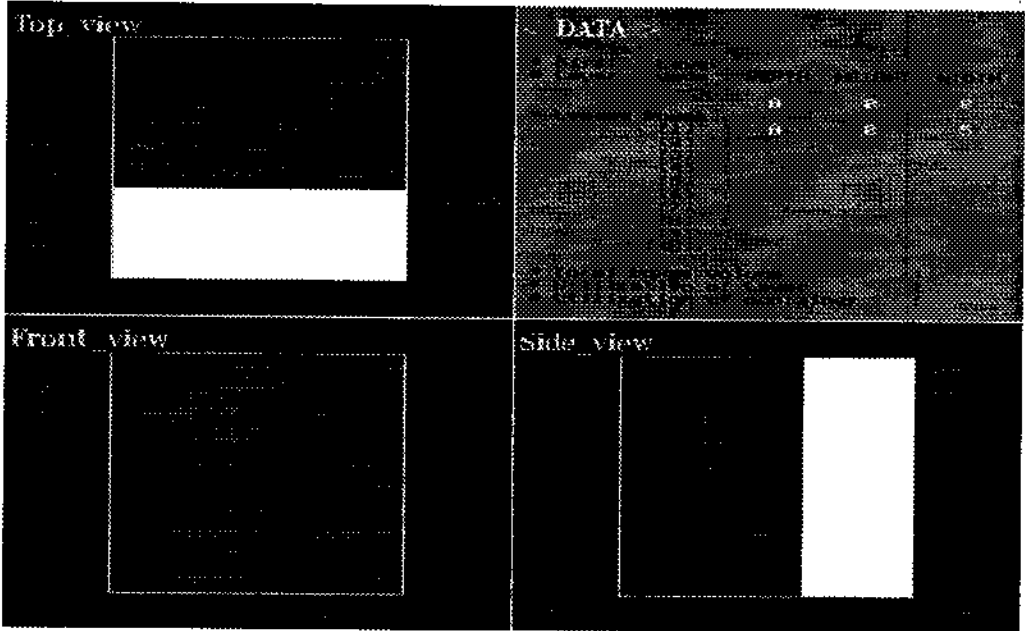


그림 14. 첫번째 Box의 적재 화면

회전하여 적재되었다. Box가 적재된 상황을 오른쪽 상단 창을 제외한 세 창에 나타냈으며, 오른쪽 상단에 Box의 Data를 표시하였다.

4.2.3 첫번째 Layer의 적재 완료

첫번째 Layer의 적재가 완료된 그림 15에서는 총 5개의 Box가 적재되었음을 알 수 있다. 현재 그림에서는 5개의 Box가 한꺼번에 적재된 것으로 묘사돼 정확한 적재 정보를 파악하기 힘들다, 실제 프로그램은 하나의 Box가 적재될 때마다 그 상황을 나타내 주기 때문에 이해하는데 큰 어려움은 없을 것이다. 첫번째 Layer에 적재된 Box들의 부피의 총합은 512이고, 그 결과 Layer 적재 효율은 100%를 나타내고 있다.

4.2.4 마지막 Layer의 적재 완료

마지막 Layer의 적재가 완료되어 하나의 적재 패턴이 완성되었다. 마지막 Layer에는 2개의 Box가 적재되어 84%의 Layer 적재 효율을 나타내었다. 마지막 Layer의 적재가 완료되면 전체 컨테이너의 적재 효율이 나타나게 되는데, 현재의 적재 패턴은 90%의 적재 효율을 나타내었다. LDB회전에 의한 다른 적재 대안도 같은 방식으로 출력될 것이다.

5. 실험 결과 및 분석

5.1 입력 Data 및 실험 결과

본 장에서는 발견적 기법을 이용하여 표 2에서 제시된 표준 컨테이너(20 feet, 30 feet,

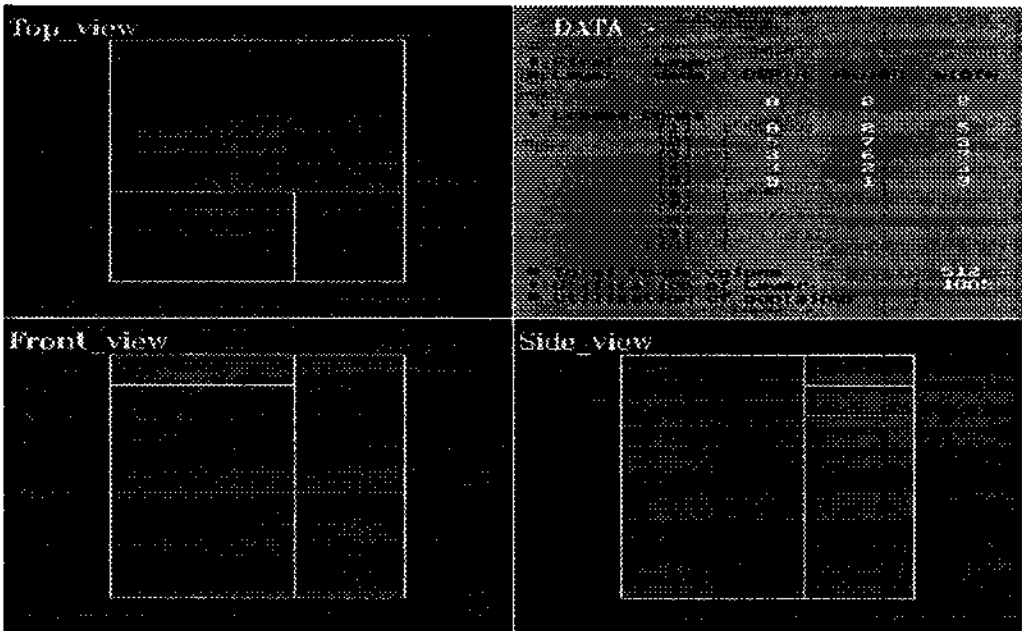


그림 15. 첫번째 Layer의 적재 완료

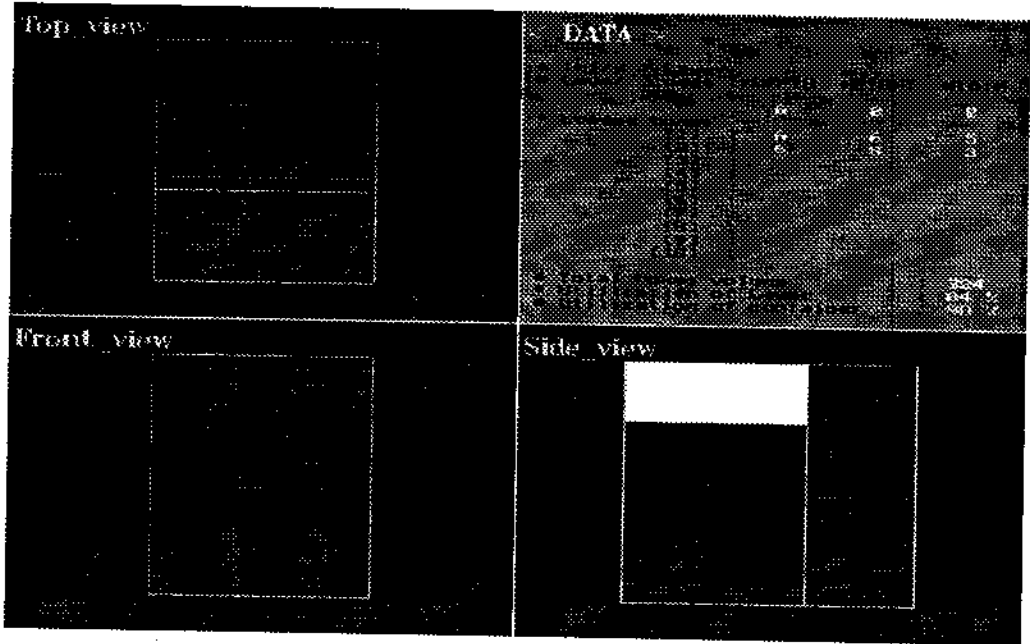


그림 16. 마지막 Layer의 적재 완료

표 5. 20 feet 컨테이너의 적재 결과 비교

Set	Number of Boxes	Proposed Algorithm		Results of Gehring <i>et al.</i> [6]	
		Number of Boxes left	Max.Volume Utilization(%)	Number of Boxes left	Max.Volume Utilization(%)
1	20	11	90.3*	8	90.1
2	20	10	85.7	8	89.4*
3	20	12	90.8	9	96.1*
4	20	11	91.6	13	98.6*
5	20	15	92.2*	14	92.1
6	20	5	88.5*	3	87.8
7	20	5	93.5	4	94.0*
8	20	5	90.9	4	94.3*
9	20	5	97.2	6	98.0*
10	20	4	94.1*	3	92.8

* : Max.Volume Utilization(%)이 높은 Algorithm

40 feet)에 Box를 적재한 결과를 살펴본다.

입력 Data (단위 : feet)

① 20 feet 컨테이너 : 깊이 20, 높이 8, 너

비 8

② 30 feet 컨테이너 : 깊이 30, 높이 8, 너

비 8

표 6. 30 feet 컨테이너의 적재 결과 비교

Set	Number of Boxes	Proposed Algorithm		Results of Gehring et al.[6]	
		Number of Boxes left	Max.Volume Utilization(%)	Number of Boxes left	Max.Volume Utilization(%)
1	30	24	97.4*	18	91.0
2	30	18	96.9*	18	91.6
3	30	16	96.4*	13	80.4
4	30	14	93.8*	16	87.3
5	30	18	98.7*	16	78.9
6	30	21	97.0*	20	95.9
7	30	16	97.4*	12	94.4
8	30	17	91.3*	16	89.3
9	30	18	96.3*	18	83.2
10	30	18	92.6	17	95.8*

* : Max.Volume Utilization(%)이 높은 Algorithm

표 7. 40 feet 컨테이너의 적재 결과 비교

Set	Number of Boxes	Proposed Algorithm		Results of Gehring et al.[6]	
		Number of Boxes left	Max.Volume Utilization(%)	Number of Boxes left	Max.Volume Utilization(%)
1	40	18	93.3*	17	81.7
2	40	18	92.1*	17	81.0
3	40	15	92.2*	15	90.7
4	40	22	95.8*	18	91.2
5	40	21	91.9*	18	90.5
6	40	23	93.4*	13	90.0
7	40	22	97.8*	23	97.1
8	40	23	95.9*	10	92.6
9	40	23	98.2*	16	88.0
10	40	20	96.5*	17	91.6

* : Max.Volume Utilization(%)이 높은 Algorithm

③ 40 feet 컨테이너 : 깊이 40, 높이 8, 너비 8

④ Box : [1~10] 의 난수를 발생시켜 최대, 중간, 최소 치수 선정.

20 feet 컨테이너에는 20개의 Box 입력.

30 feet 컨테이너에는 30개의 Box 입력.

40 feet 컨테이너에는 40개의 Box 입력.

실험은 각 컨테이너 당 10번씩 행해졌으며,

동일한 가정과 실험 환경을 사용한 Gehring *et al.*[6]의 논문과 그 결과를 비교하였다. 비교의 척도는 컨테이너의 적재 효율(Volume Utilization)을 사용하였으며, 각 실험에서 최대 적재 효율을 비교 대안으로 제시하였다. 또한 적재되지 않고 남은 Box의 수도 함께 제시하였다. 표 5에서 표 7까지 3가지 표준 컨테이너의 적재 결과를 Gehring *et al.*[6]의 결과와 함께 비교하여 나타내었다.

5.2 결과 분석

3가지 표준 컨테이너에 Box를 적재한 결과, 제안된 발견적 기법은 대부분 90%를 상회하는 높은 최대 적재 효율을 나타내었다. 또한, Gehring *et al.*[6]의 결과와 비교해 볼 때 20 feet 컨테이너의 경우에는 비슷한 결과를 나타내었지만 30 feet, 40 feet 컨테이너의 경우에는 높은 적재 효율을 나타내었다. 적재 효율면에서 제안된 발견적 기법은 Gehring *et al.*[6]의 알고리즘보다 우수한 것으로 평가된다.

6. 결론 및 추후 과제

본 연구에서는 컨테이너에 직육면체의 성격을 가진 물품을 적재시에 컨테이너의 공간 이용을 최대화시키면서 하중 분포의 균형을 이루는 3-차원 컨테이너 적재 문제를 다루고 있다.

각 물품의 컨테이너 내의 최적 위치를 결정하는 수리적 모델을 개발한 후 기존의 연구를 개선한 발견적 기법을 제시하였다. 특히, 물품 적재에서 중요한 요소인 하중 분포의 균형화 방안에 관한 구체적인 대안을 제

시함으로써 실제 현장에서 사용하는데 도움이 될 수 있을 것으로 생각한다.

또한, 발견적 기법에 의해 도출된 적재시의 위치 정보를 그래픽하게 제시하는 컴퓨터 프로그램을 개발하여, 컨테이너 적재 문제에 관한 지식이 없어도 사용자가 쉽게 적재 대안을 생성하고 확인할 수 있도록 하였다. 제안된 발견적 기법이 적재 효율면에서 기존의 알고리즘보다 효율적임을 예제를 통해 보여 주었다.

그러나 현실적으로, 물품을 적재할 때 적재 효율만 고려할 것이 아니라 적재의 용이성, 컨테이너의 적재 무게 제한 및 무게 중심, 안정성 등을 고려하는 알고리즘의 개발을 통해 보다 실용적인 기능을 가지는 프로그램의 개발에 관한 연구가 필요하다고 보여진다.

참 고 문 헌

- [1] Abdou.G, Yang.M, "A Systematic Approach for the Three-dimensional Palletization Problem", *International Journal of Production Research*, Vol. 32, No. 10, 1994
- [2] Chen.C.S, Lee.S.M, Shen.Q.S, "An analytical model for the container loading problem", *European Journal of Operational Research*, Vol. 80, 1995
- [3] Christofides, N. and Whitlock, C., "An Algorithm for Two-dimensional Cutting Problem", *Operations Research*, Vol. 13, 1977
- [4] Dowsland.W.B, "Three dimensional pack-

- ing - Solution approaches and heuristic development", *International Journal of Production Research*, Vol. 29, No. 8, 1991
- [5] George, J. A. and D. F. Robinson, "A Heuristic for Packing Boxes into a Container", *Computer and Operations Research*, Vol. 7, No. 3, 1980
- [6] Gehring.H, Menschner.K, Meyer.M, "A computer-based heuristic for packing pooled shipment containers", *European Journal of Operational Research*, Vol. 44, 1990
- [7] Gilmore, P. C. and R. E. Gomory, "Multistage Cutting Stock Problem for Two and More Dimensions", *Operations Research*, Vol. 13, No. 1, 1965
- [8] Golden Bruce L., "Approaches to the Cutting Stock Problem", *AIIE Transactions*, 1976
- [9] Haessler.R.W, Talbot.F.B, "Load planning for shipments of low density products", *European Journal of Operational Research*, Vol. 44, 1990
- [10] Han.C.P, Knott.K, Egbelu.P.J, "A heuristic approach to the three-dimensional cargo-loading problem", *International Journal of Production Research*, Vol. 27, No. 5, 1989
- [11] Harold J. Steudel, "Generating Pallet Loading Patterns: A Special Case of the Two-Dimensional Cutting Stock Problem", *Management Science*, Vol. 25, No. 10, 1979
- [12] Smith.A, De Cani.P, "An Algorithm to Optimize the Layout of Boxes in Pallets", *J. of Operations Research Society*, Vol. 31, 1980
- [13] Thom J. Hodgson, "A Combined Approach to the Pallet Loading Problem", *IIE Transactions*, Vol. 14, No. 3, 1982
- [14] Tsai.R.D, Malstrom.E.M, Meeks.H.D, "A Two-Dimensional Palletizing Procedure for Warehouse Loading Operations", *IIE Transaction*, Vol. 20, No. 4, 1988
- [15] Tsai.R.D, Malstrom.E.M, Kuo.W, "Three Dimensional Palletization of Mixed Box Sizes", *IIE Transactions*, Vol. 25, No. 4, 1993