

# 기존 적재 알고리즘을 적용한 후 사후 선택 시 20 FT 컨테이너에 다양한 형태의 화물 적재 효율 정량화

류기상<sup>†</sup>, 박준선<sup>††</sup>

## Quantification of Loading Efficiency of Various Type Loads in a 20 FT Container with Post Selecting Process after Applying Conventional Loading Algorithms

Kisang Ryu<sup>†</sup>, Junseon Park<sup>††</sup>

### ABSTRACT

Container load problems are conventional issues in history. Loading cylindrical or rectangle packages in restricted space is required to be efficient to minimize remaining area. Many people have depended on experience or known simple algorithms for a long time. Recently systematic algorithms are published over the world with developed computer technology. Maximal rectangles algorithm is well known to load arbitrary rectangles in a larger rectangle container with the organized manners. However, this algorithm has significant drawbacks, which are influenced by orientation of input load. We propose post-selective method to complement these orientation effects, which is found out to improve the loading efficiency.

**Key words:** Maximal Rectangles Algorithm, Post-selective Method, Container Loading Problems

### 1. 서 론

전 세계적으로 300m가 넘는 대형 컨테이너 선박들이 항해하고 있는 최근까지도 이슈가 되고 있다. 몇 십년 간 화물 적재 문제는 해운 회사들이 풀어야 할 고질적인 문제였고, 적재하는 알고리즘에 따라서는 한 번에 실을 수 있는 화물의 개수가 달라지고, 이에 따라 이익률에 차이가 있으니, 정해진 부피의 컨테이너 선에 적절한 배치를 통하여 최대 부피의 화물을 적재하는 문제가 많은 연구자들로부터 연구되고 있다. 이 문제는 단순한 정사각형이나 원을, 보다 큰 다각형에 넣는 문제부터 시작해서, 아주 복잡한 형태

의 3차원 화물들을 여러 컨테이너에 효율적으로 나누는 문제까지 여러 가지 범주로 나눌 수 있다.

Bortfeldt et al. 에 따르면, 정해진 화물량을 최소한의 컨테이너에 적재하는 문제 (input minimization type)와, 정해진 컨테이너 부피에 최대 화물량을 적재하는 문제(output maximzation type)로 크게 나눌 수 있으며, 각각의 문제 유형들은 아래와 같다: [1]

- Single Stock-Size Cutting Stock Problem (SSSCSP): 약간 다른 형태의 짐들을, 최소한의 같은 형태 컨테이너에 적재하는 문제
- Multiple Stock-Size Cutting Stock Problem

\* Corresponding Author : Kisang Ryu, Address: (607-063) Oncheon 3-dong, Dongnae-gu, Busan, Korea, TEL : +82-505-847-4000, FAX : +82-505-847-3000, E-mail : wondboy@naver.com

Receipt date : Nov. 8, 2017, Revision date : Feb. 23, 2018  
Approval date : Mar. 9, 2018

<sup>†</sup> Howonsoft

<sup>††</sup> Howonsoft (E-mail : wondboy@naver.com)

\* This research was supported by 2017 Nurimaru R&BD project funded by the Busan IT Industry Promotion Agency

(MSSCSP): 약간 다른 형태의 짐들을, 최소한의 약간 다른 형태 컨테이너에 적재하는 문제

- Residual Cutting Stock Problem (RCSP): 약간 다른 형태의 짐들을, 최소한의 완전히 다른 형태 컨테이너에 적재하는 문제

- Single Bin-Size Bin Packing Problem (SBSBPP): 완전히 다른 형태의 짐들을, 최소한의 같은 형태의 컨테이너에 적재하는 문제

- Multiple Bin-Size Bin Packing Problem (MBSBPP): 완전히 다른 형태의 짐들을, 최소한의 약간 다른 형태의 컨테이너에 적재하는 문제

- Residual Bin Packing Problem (RBPP): 완전히 다른 형태의 짐들을, 최소한의 완전히 다른 형태의 컨테이너에 적재하는 문제

- Open Dimension Problem (ODP): 정해진 짐들을 하나의 컨테이너에 모두 넣기 위해, 해당 컨테이너의 부피를 최소화하는 문제

위와는 다르게, 최대화하는 문제는 다음과 같다:

- Identical Item Packing Problem (IIPP): 하나의 컨테이너에 최대한의 동일 크기의 짐들을 적재하는 문제

- Single Large Object Placement Problem (SLOPP): 하나의 컨테이너에, 약간 다른 형태의 짐들을 최대한 적재하는 문제

- Multiple Identical Large Object Placement Problem (MILOPP): 약간 다른 형태의 짐을, 한 종류의 여러 컨테이너에 최대한 적재하는 문제

- Multiple Heterogeneous Large Object Placement Problem (MHLOPP): 약간 다른 형태의 짐을, 다른 종류의 여러 컨테이너에 최대한 적재하는 문제

- Single Knapsack Problem (SKP): 하나의 컨테이너에, 완전히 다른 종류의 짐들을 최대한 적재하는 문제

- Multiple Identical Knapsack Problem (MIKP): 한 종류의 컨테이너들에, 완전히 다른 종류의 짐들을 최대한 적재하는 문제

- Multiple Heterogeneous Knapsack Problem (MHKP): 다른 형태의 컨테이너들에, 완전히 다른 형태의 짐들을 최대한 적재하는 문제

가장 실용적인 문제이며, 가장 많이 연구 되어진

연구 유형은 SLOPP와 SKP로서 2011년까지 각각 37건, 57건의 논문이 발표되었으며, IIPP 또한 10건의 논문이 발표되었다 [2]. 즉, 약간 다른 형태의 짐들을 한 컨테이너에 최대한 적재하는 문제와, 완전히 다른 형태의 짐들을 한 컨테이너에 최대한 적재하는 문제를 해결할 수 있는 알고리즘이 가장 필요하다는 것으로 해석할 수 있다. 그 중 가장 간단한 IIPP의 대표 문제로서는 정사각형에 가장 많은 원을 적재하는 문제가 있으며, 현재 50개 이하의 원은 Fig. 1에서와 같이 잘 알려진 방법으로 적재하여 효율을 높일 수 있다[3].

다만, 해석적인 해가 존재하지 않고, 반복적 수치해 밖에 존재하지 않는 영역에서는 꾸준히 더 효율적인 방법들이 발표되고 있다.

현재까지 알려진 바로는 Fig. 2에서와 같이 적재 효율이 이론적 상한에 근접할 수 있도록 연구가 되고 있다. 이론적인 상한 점선은 세 개의 원의 중심이 정삼각형을 이루며 비활용면적을 최소화할 때 가능한 밀도이며, 이론적인 하한 점선은 네 개의 원의 중심이 정사각형을 이루며 정사각형의 컨테이너에 적재될 때 달성되는 밀도이다. 특히 20개 미만의 경우에는 정삼각형을 이루는 구조를 만들더라도 벽과의 거리가 멀게 되어 효율이 좋지 않지만, 개수가 많아질수록 이론적인 상한 점선에 근접하는 형태를 띠게 된다. 이 외에 컨테이너 형상이 정사각형이 아닌 경우에도 알고리즘이 개발이 되었는데, 다양한 컨테이너 형상에 가장 많은 원을 적재하는 문제가 있으며, Fig. 3과 같이 컨테이너 형상과 크기는 원, 삼각형, 정사각형, 직사각형 등으로, 비선형 방정식을 풀어서 적재하게 된다[4].

이러한 IIPP 문제는 해석적인 방정식을 사용하여 풀기 보다는, 기존의 해법을 많이 사용하게 된다 [5]. 발견적 해법이 최적임을 증명하기 위해서, 다른 방법보다 더 효율적인 해법을 제안하는 식으로 증명하고 있다. 이러한 방식으로 20개의 원까지는 최적해가 잘 알려져 있고, 이후 21개의 원부터 발견적 해법으로 78개까지 효율적인 방법이 제시되었으며, 이를 바탕으로 특정 패턴을 분석하여 이상적인 효율을 구할 때 사용하기도 한다[6]. 일례로 36개까지의 원은 정사각형 형태의 패턴으로 적재되는 것이 가장 활용면적을 늘릴 수 있지만, 49개 부터는 정사각형 형태가 아닌 무작위스러워 보이는 Fig. 4의 형태가 더 효

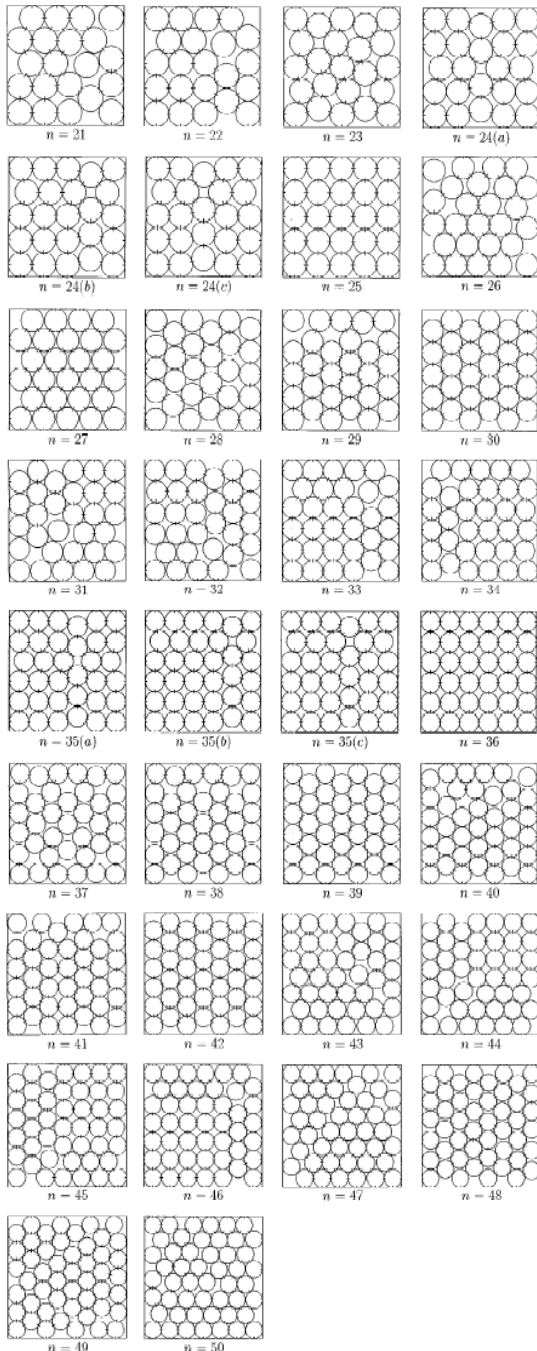


Fig. 1. Best known packings for  $21 \leq n \leq 50$  [3].

울적인 것으로 밝혀졌다. 따라서 본 연구에서는 상용 직육면체 컨테이너의 크기에 따라 최대로 들어갈 수 있는 원의 개수를 구하고, 이에 따른 면적 효율을 비교해 본다.

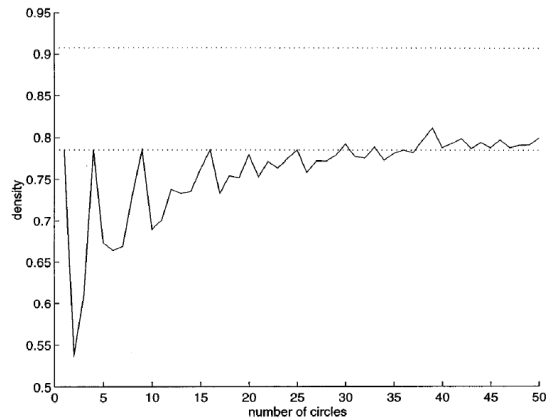


Fig. 2. Densities of best known packings [3].

직사각형 화물에서는 약간 다른 직사각형 화물을 정해진 직사각형 컨테이너에 최대한 적재하는 문제 (SLOPP)를 실용적인 측면에서 2차원적으로 접근한 Jylänky의 Maximal Rectangles 알고리즘[7]을 구현하고, 보다 더 높은 효율을 줄 수 있는 발견적 해법을 제시하고자 한다. Maximal Rectangles 알고리즘을 보완하기 위해 다양한 사전 선택[8]과 알고리즘이 개발되어 왔지만, 아직까지 사후 선택 알고리즘을 적용한 경우가 드물고, 모든 경우의 방향을 고려하여 최적의 세트를 추출하는 방식은 상당히 효율을 높여 줄 수 있다. 이렇게 개발된 사후 선택 알고리즘과 실시간으로 형상과 거리를 분석하는 알고리즘[9]과 융합되면 최단시간 적재 프로그램을 제작할 수 있을 것으로 기대된다.

## 2. 이 론

### 2.1 원 배열 알고리즘

앞 서 설명한 정사각형에 원을 적재하는 IPPP 문제의 연장선상에서 직사각형에 원을 넣는 알고리즘은 간단하면서도 주요한 문제이기 때문에 여러 방면에서 연구가 되었다. 화물 적재 문제에서 시작하여 토질의 공극 문제, 금속의 결정 문제 등을 모델링할 때도 사용되었다. Fig. 5에서와 같이 가장 간단하게는 원의 중심이 정사각형을 이루는 격자 구조와, 정삼각형을 이루어 7개의 원이 중심부를 포함해 육각 형태를 이루는 구조로 나뉘어 질 수 있는데, 일정 크기의 원을 정확하게 담을 수 있는 직사각형의 경우에는 최적화 과정이 간단하지만, 컨테이너의 너비와 길이

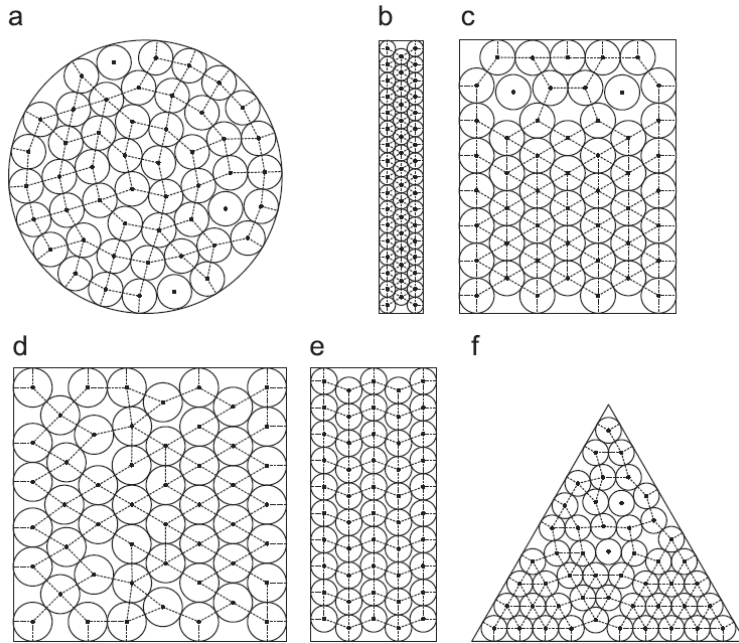


Fig. 3. Optimal layouts for 50 items in: (a) a circle; (b) a rectangle, minimizing its area; (c) a rectangle, minimizing its perimeter; (d) a square; (e) a strip, with fixed length  $L=9:5$ ; and (f) an equilateral triangle [4].

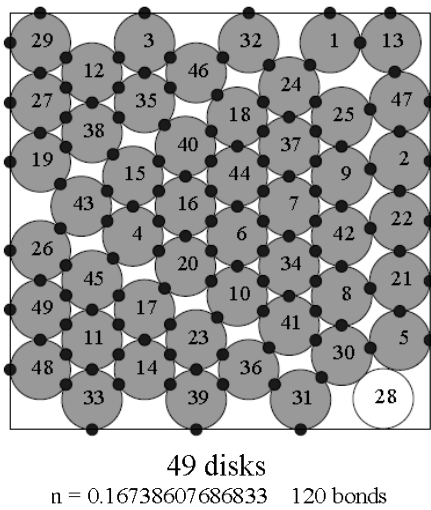


Fig. 4. The densest packings found of 49 disks [6].

가 원의 공배수가 아닐 때는 개수와 크기에 따라서 수치해석적인 비교가 필요하다.

### 2.2 Maximal Rectangles 알고리즘 [7]

컨테이너 부피를 2차원의 면적으로 간소화하고, 각 화물의 다양한 면적으로 최대화할 수 있는 알고리

즘으로 기요틴 (Guillotine) 알고리즘 [11]이 널리 쓰이고 있다. 하지만, 최초로 고안한 저자는 알 수 없으며, Watt et al.이 발표한 게임[12]에 쓰였으며 이후 다양하게 활용되었다. 기요틴 알고리즘은 Fig. 6과 같이 컨테이너 면적에 특정 형상의 화물이 차지하면, 나머지 면적을 분할하여 다시 새로운 면적으로 사용하는 방식을 사용한다.

Jylänky 가 고안한 최대 직사각형 (Maximal Rectangle) 알고리즘은 기본적인 형태는 기요틴 알고리즘과 유사하지만, Fig. 7에서처럼 하나의 직사각형이 투입되고, 나머지 영역의 공통 부분을 절단하는 것이 아니라, 공통 부분을 포함하여 최대 직사각형을 만들고, 이에 다른 화물들을 채워 넣은 다음, 가장 많이 들어가는 쪽으로 진행해 나가는 점이 다르다.

최대 직사각형 (Maximal Rectangle) 알고리즘은 다음과 같다.[7]

### 3. 제안한 방법

#### 3.1 원 육각 배열 계산 알고리즘

정해진 컨테이너 치수 안에서 어느 크기까지 정사각형 알고리즘이 유리한 지, 육각 배열 알고리즘이

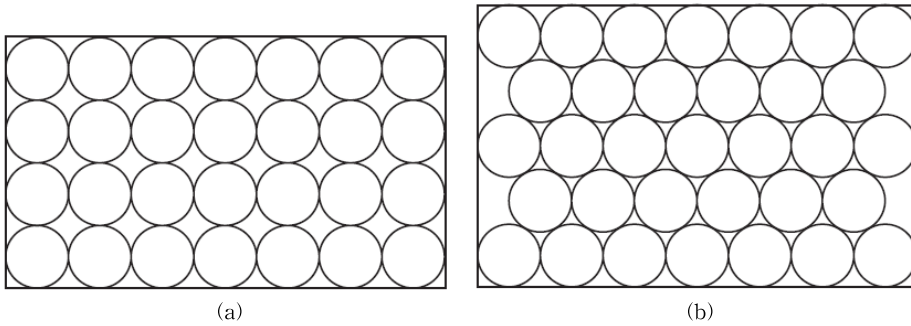


Fig. 5. Two lattice packings: (a) square arrangement; and (b) hexagonal arrangement [10].

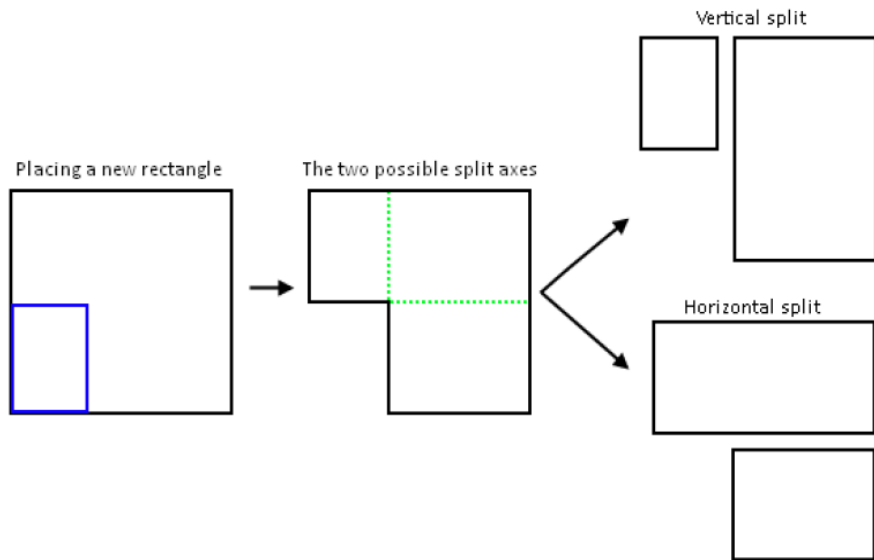


Fig. 6. The guillotine split placement process. After placing a rectangle, there are two ways to store the remaining free area [7].

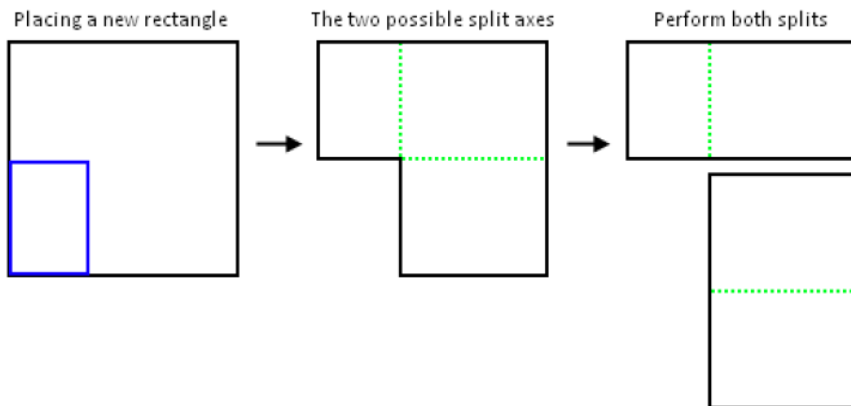


Fig. 7. The rectangle placement rule for the MAXRECTS data structure. Both the rectangles on the right are store in F [7].

유리할 지 계산이 필요하며, 이를 수치 해석 방법을 활용하여 컨테이너의 단축과 똑같은 지름의 원의 배열부터 시작하여, 원의 지름을 1/2씩 줄이면서 원의 크기를 줄이고 각각의 원의 크기에서 정사각 배열과 육각 배열을 비교한다. 이 중에 주어진 원의 크기가 어느 범위에 있는지 먼저 결정하고, 더 큰 면적 효율을 갖는 배열 방식으로 배치하는 알고리즘을 설계하

였다.

이를 계산하는 계산식은 아래와 같다.

첫 번째, 정사각 배열의 경우 컨테이너의 장축이  $y$ , 단축이  $x$  라고 하면 ( $y \geq x$ ),  $mD \leq y$ 를 만족하는 최대 정수  $n$ 과,  $mD \leq x$ 를 만족하는 최대 정수  $m$ 을 구한다.

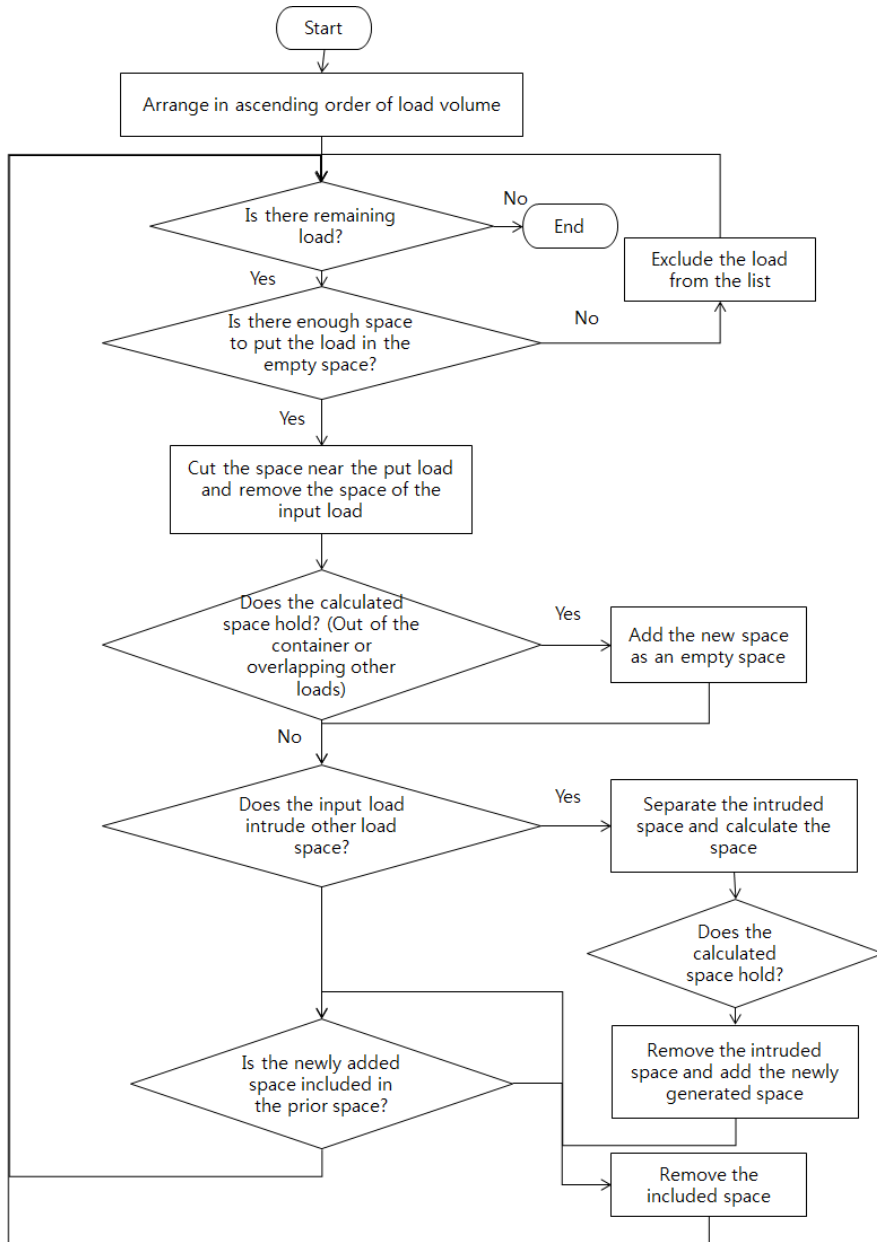


Fig. 8. Flowchart for Maximal Rectangles algorithm.

이 때 전체 컨테이너의 면적  $S$ 를 구하고, 원이 차지하는 면적  $S_D$ 를 구할 수 있다.

$$S = xy \quad (1)$$

$$S_D = nm \frac{\pi D^2}{4} \quad (2)$$

이 때의 공간 효율  $\eta$  는 다음과 같이 계산한다.

$$\eta = \frac{S_D}{S} \quad (3)$$

두 번째, 육각 배열의 경우 Fig. 5(b)와 같이 각각의 원이 인접한 열과 1개씩 차이가 날 수 있다. 최초 시작 열의 개수를 최대한으로 설정하여 정사각 배열의 경우와 마찬가지로 장축( $y$ ), 단축( $x$ )라고 할 때,  $nD \leq y$ 를 만족하는 최대 정수  $n$ 을 계산한다. 하지만 단축의 경우에는 원의 지름( $D$ )으로 결정되는 것이 아니라 원의 중심이 정삼각형을 이루기 때문에 처음과 끝을 제외한, 원끼리의 인접 길이는  $\frac{\sqrt{3}}{2}D$ 이 된다. 이때 최초 열이 최대치의 원이 들어가고, 다음 열도 같은 개수가 들어갈 때는 식 (2)과 같은 계산으로 면적을 구할 수 있다.

다만, 다음 열의 원의 개수가 1개 적게 되면  $m$ 의 값이 홀수인지 짝수인지에 따라 계산이 달라지게 된다.  $m$ 은 식 (4)를 만족하는 최대 정수가 된다.

$$D \left( 1 + \frac{m-1}{2} \sqrt{3} \right) \leq x \quad (4)$$

짝수인 경우에는

$$S_D = (2n-1) \frac{m}{2} \frac{\pi D^2}{4} = \left( nm - \frac{m}{2} \right) \frac{\pi D^2}{4} \quad (5)$$

홀수인 경우에는

$$S_D = \left( n \frac{m+1}{2} + (n-1) \frac{m-1}{2} \right) \frac{\pi D^2}{4} = \left( nm - \frac{m-1}{2} \right) \frac{\pi D^2}{4} \quad (6)$$

를 통해 공간 효율  $\eta$ 를 식 (3)을 통해 계산할 수 있다.

### 3.2 MaxRects 사후 평가 알고리즘

앞 서 서론에서 논한 바와 같이, 사전 선택 알고리즘이 아닌, 다양한 방향의 경우의 수를 모두 계산한 후 가장 용적률이 높은 결과를 선택하는 것으로서, 화물 적재 효율을 높일 수 있다. 특히 화물들의 형태

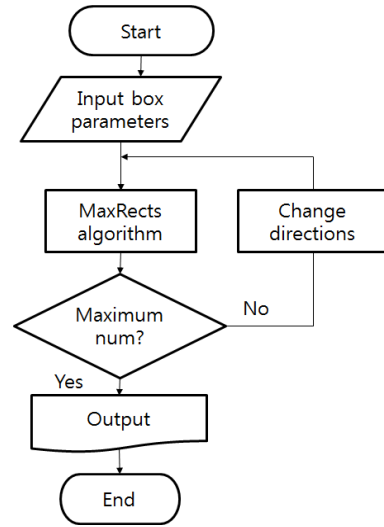


Fig. 9. Flowchart for Post Selecting Maximal Rectangles algorithm.

가 약간 다른 조건이나 완전히 다른 조건 모두에서 적용 가능하며, 형태의 종류에 따라 계산 시간이 늘게 되지만, 최근까지 발전된 컴퓨터의 계산 속도를 고려하면 무시할 수 있는 수준이다. Fig. 9에 제안하는 알고리즘의 전체적인 흐름도를 나타내었다.

## 4. 실험 결과 및 고찰

### 4.1 IIPP 문제 중 원의 경우

앞 서 언급한 바와 같이 국제 규격의 20" 컨테이너에 다양한 크기의 원통을 적재할 때, 정사각 배열과 육각 배열의 적재 방법에 따라 적재율에 차이를 보이게 된다. 하지만, 일반적인 규칙이 적용되는 것이 아니라, 컨테이너 크기와 원의 크기에 따라 비선형적으로 차이가 나기 때문에 Fig. 10(a)에서와 같이 지름이 줄어들 때, 두 가지 적재 방식에 따른 공간 효율을 표시하였다. 20" 컨테이너에 적재하였을 시 지름이 100 mm 이하일 경우 육각 배열(Hexagonal array)이 직각 배열(Square array) 보다 약 10% 이상 높지만, 100 mm 이상에서는 Fig. 10(b)에서와 같이 동일한 효율(차이:0%)이거나 33%부터 -7% 까지 다양한 효율 분포를 보인다.

원의 지름인 1205mm 일 때와 1145 mm인 적재 분포를 살펴보면 Fig. 11과 같다. 지름의 차이는 10% 도 나지 않지만, 적재 효율의 차이는 40% 이상 차이

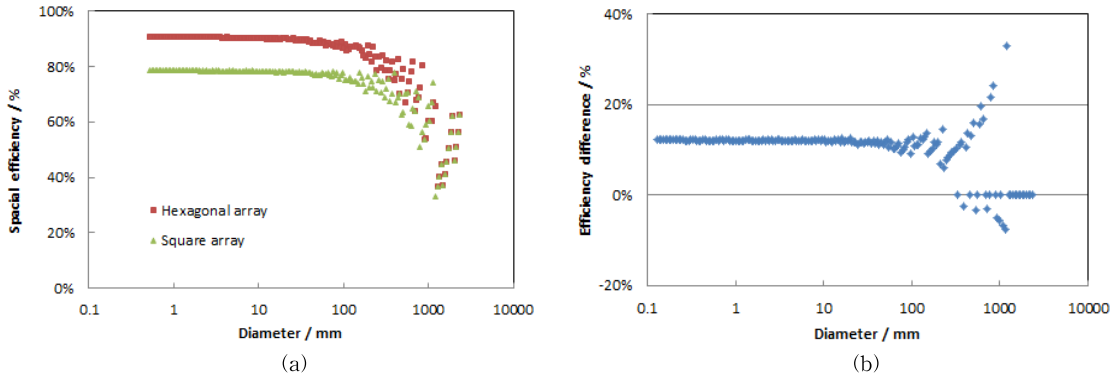


Fig. 10. (a) Spatial efficiencies of square array and hexagonal array as a function of various circle diameter in a 20FT container and (b) the difference of the spatial efficiency of square and hexagonal arrays.

가 나며, 각각의 적재 방법을 사후에 결정하는 것이 효과적이다.

일례로 Fig. 11(a)와 (b)는 같은 크기의 원일 때, 정사각 배열일 경우에는 4개의 원 밖에 들어가지 않지만, 육각 배열인 경우 8개 까지 들어가는 등 2배의 효율을 낼 수 있다. 반면에 Fig. 11(c)와 (d)의 경우 반대로 육각 배열인 경우가 1개의 원이 덜 들어가기 때문에 7% 정도의 효율 손실을 보게 된다. 따라서 각각의 경우에 2가지 조합을 모두 실시해 보고, 비교한 뒤 적재 알고리즘을 택하는 편이 효율적이게 된다.

4.2 IIPP 중 직사각형의 경우

국제 규격의 20" 컨테이너 (5896 mm×2348 mm×2372mm)에 길이 방향으로 4, 8, 16, 20개를 넣을 수

있는 면적을 가진 화물 4개를 각각 다른 방향으로 입력하여 Maximal Rectangles 알고리즘을 적용하여 보았다.

길이 방향이나 높이 방향 배열과 관계없이, Fig. 13(a)와 같이 치수가 입력된 순서에 따라서 길이 방향을 컨테이너의 왼쪽 위 모서리에 맞추고 시작하기 때문에, 같은 치수의 화물이라도 길이 방향과 높이 방향의 입력 순서에 따라서 용적률의 차이가 나기 시작한다. 반면에 Fig. 13(b)는 운 좋게도 컨테이너 높이 방향의 1/2부분부터 왼쪽 벽에 접해서 시작했기 때문에 100% 용적율을 달성할 수 있었다.

각각의 용적률은 Table 1에서와 같이 화물의 면적의 총합이 컨테이너 면적이 되도록 입력하였기 때문에, 컨테이너 면적 사용률과 화물의 적재율은 동일한

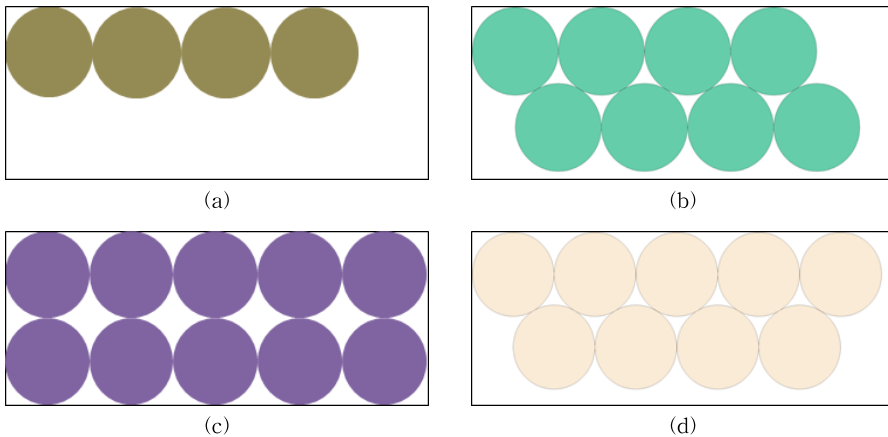


Fig. 11. Packing pattern (a) of 1205 mm diameter cylinders with rectangle array, (b) of 1205 mm diameter cylinders with hexagonal array, (c) of 1145 mm diameter cylinders with rectangle array and (d) of 1145 mm diameter cylinders with hexagonal array in a 20FT container.



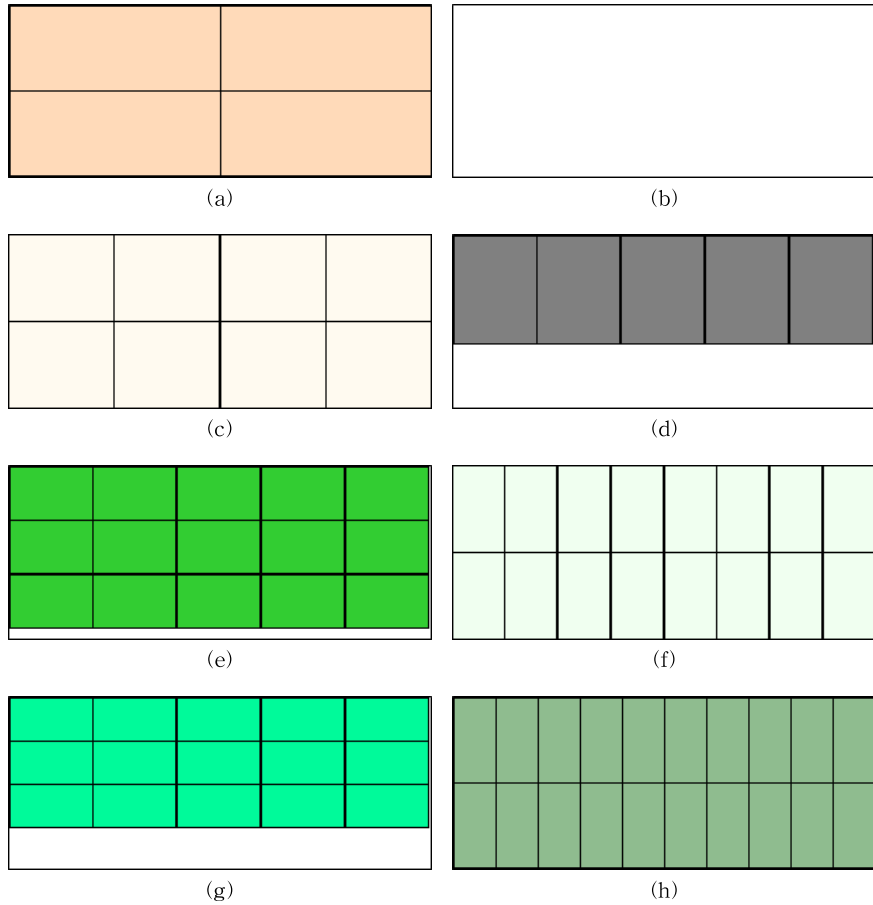


Fig. 12. Packing pattern in a 20FT container by MaxRects algorithm with (a) 4 longitudinal, (b) 4 lateral, (c) 8 longitudinal, (d) 8 lateral, (e) 16 longitudinal, (f) 16 lateral, (g) 20 longitudinal and (h) 20 lateral identical rectangles.

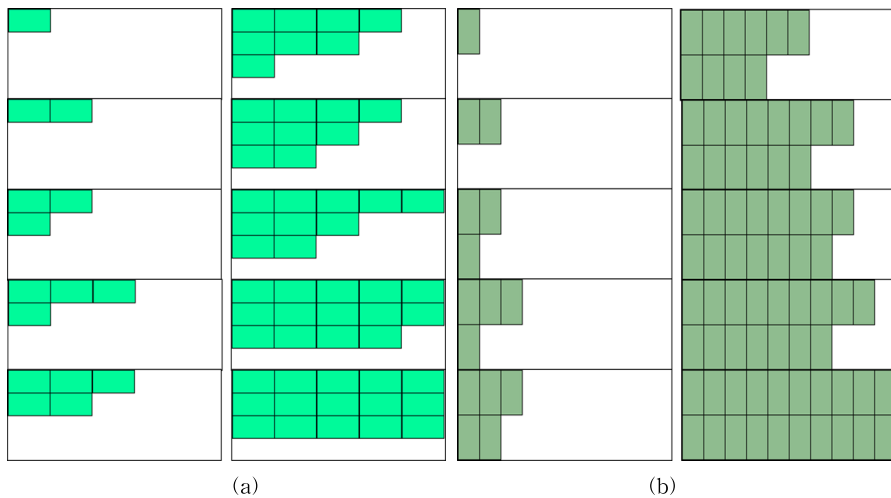


Fig. 13. Packing order in a 20FT container by MaxRects algorithm with (a) 20 longitudinal and (b) 20 lateral identical items.

Table 1. The efficiencies for conventional algorithms

Identical items	Longitudinal	Lateral
	Efficiency [%]	Efficiency [%]
4	100	0
8	100	62.5
16	93.75	100
20	75	100

효율이므로 하나의 효율만 표기하였다.

4.3 IIPP 중 직육면체의 경우

평면과 마찬가지로 MaxRects 알고리즘을 적용하였으나, 방향 전환은 사후 선택으로 결정하였다. Table 2에서는 컨테이너 크기의 각 길이별로 1/10의 크기인 직육면체를 각각의 방향에 따라 적재하였으며, 이에 따른 적재 효율을 기재하였다.

Table 2와 같이 각각의 방향에 따른 길이를 정해 주어 방향에 대한 효과를 관찰하였으며, 같은 크기의 직육면체이지만 적재 방향에 따라 100%에서 72%까지 약 30%가량 차이가 나는 것을 확인할 수 있었다.

4.4 SLOPP의 경우

하나의 형태를 가진 IIPP와 다르게, SLOPP의 경우 서로 다른 형태의 화물을 각각 다른 방향으로 입력하여 결과를 Fig. 15과 같이 확인 하였다.

결과적으로 IIPP와 유사하게 화물의 한 쪽 모서리가 컨테이너의 높이의 반만큼 되는 쪽으로 처음 입력이 되었을 때, 용적률이 100%가 됨을 Fig. 15(a)에서 확인하였으며, 그 외의 경우에는 모두 50%의 용적률 밖에 나타내지 못하였다.

3가지 다른 형상은 Maximal Rectangles 알고리즘을 사용하기 전에 면적이 큰 순서대로 순차적으로 배열이 된 후에 입력이 되므로, 조합에 영향을 끼치

지 못하지만, 방향의 경우 형상마다 2가지씩, 2의 3제곱이 되어 총 8개의 다른 결과를 Fig. 16과 같이 나타내게 된다. 그래프와 같이 100%의 정답이 있음에도 불구하고, 입력 방향에 따라 최대 50%까지 차이가 날 수 있다. 8개의 경우의 수를 모두 고려하면 항상 최상의 효율을 유지할 수 있으므로, 본 논문에서 제안한 방법을 통해 Maximal Rectangles 알고리즘의 방향성 문제를 발견적으로 해결할 수 있다.

5. 결 론

본 논문에서는 기존의 잘 알려진 컨테이너 적재 알고리즘인 원의 육각 배열 알고리즘과 직사각형 상자를 최대한 많이 적재하는 Maximal Rectangles 알고리즘의 방향성에 대한 문제를 발견하고, 이를 개선하기 위한 사후 처리 방법을 제안하고자 하였다.

원의 지름이 컨테이너의 크기보다 상대적으로 작을 때는 육각 배열 알고리즘이 정사각 배열 알고리즘에 비해 항상 높은 적재 효율을 보인다. 원의 크기가 컨테이너의 단축의 절반부터 1/10 정도라면, 정사각 배열과 육각 배열을 모두 고려해야 한다는 사실을 알게 되었다. 이에 따라 효율이 크게는 30% 이상 차이가 나는 것을 확인하였으며, 실제적으로도 활용할 수 있게 되었다.

Maximal Rectangles 알고리즘은 모든 방향에서의 용적률을 고려한 뒤 최고의 용적률을 나타내는 패턴으로 화물을 적재하게 되면, 항상 최고의 용적률을 보여줄 수 있다. 이는 단일 형상의 화물이나 3가지 이상의 복합 형태의 화물 또한 유사한 결과를 보여주었다. 2차원 뿐만 아니라 3차원에서도 방향에 따른 사후 선택이 20~30% 가까이 효율 차이가 나는 것을 보여 주었다. 향후 이러한 알고리즘들을 복합적으로 사용할 경우 사후 선택에 의한 결정이 컴퓨터 수치 계산 시간은 늘어나지만, 화물 적재 효율을 크게 늘려

Table 2. The combination of 3 different directions of rectangular cuboid

Pattern	x dimension (mm)	y dimension (mm)	z dimension (mm)	Efficiency (%)
(a)	589	234	237	100
(b)	589	237	234	90
(c)	234	589	237	75
(d)	234	237	589	90
(e)	237	589	234	72
(f)	237	234	589	96

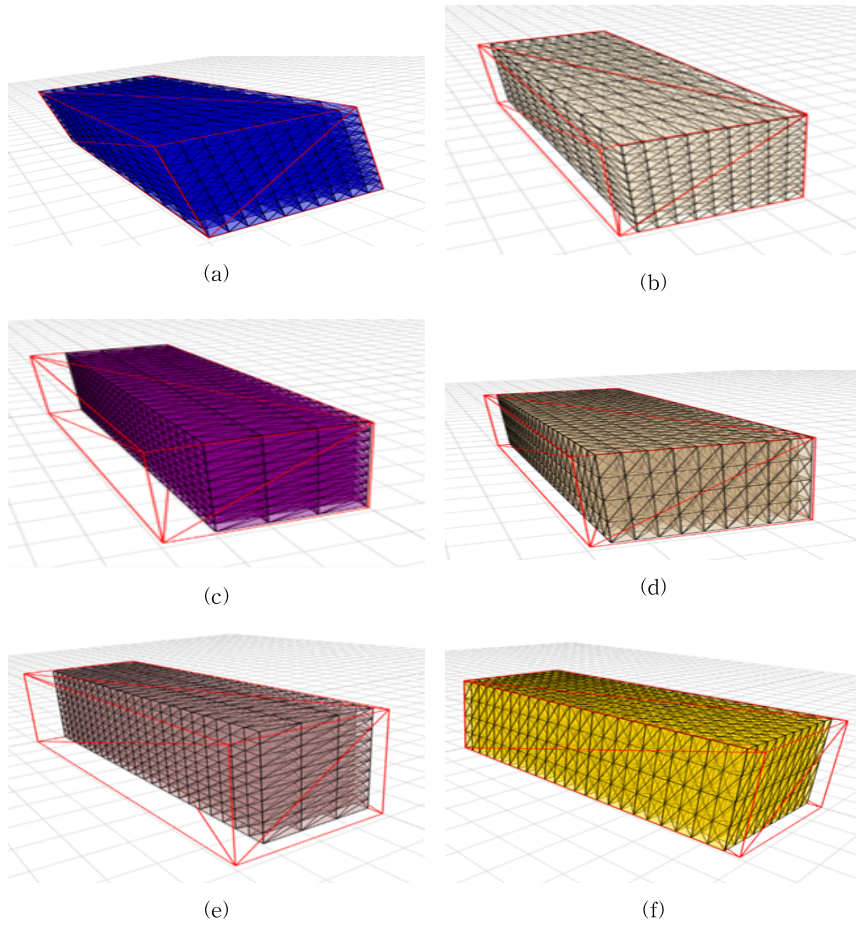


Fig. 14. Packing pattern in a 20FT container by MaxRects algorithm according to the Table 2.

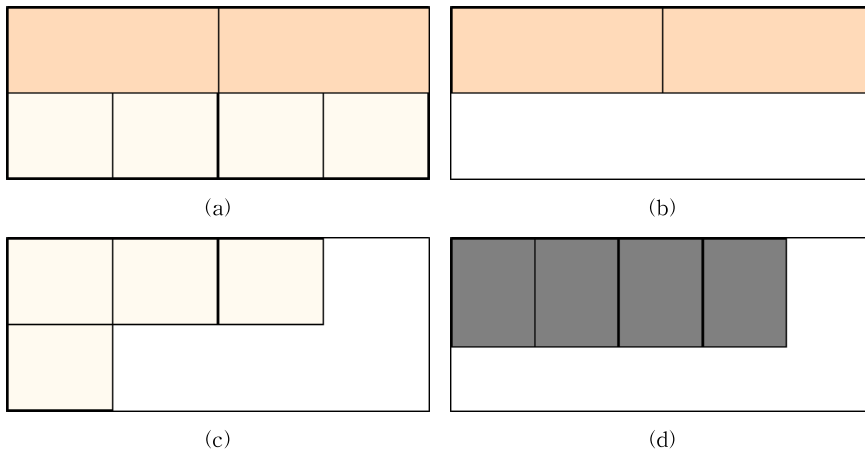


Fig. 15. Packing pattern in a 20FT container by MaxRects algorithm with (a) 2 longitudinal 1/2 size + 4 longitudinal 1/4 size, (b) 2 longitudinal 1/2 size + 4 lateral 1/4 size, (c) 2 lateral 1/2 size + 4 longitudinal 1/4 size, (d) 2 lateral 1/2 size + 4 lateral 1/4 size rectangles.

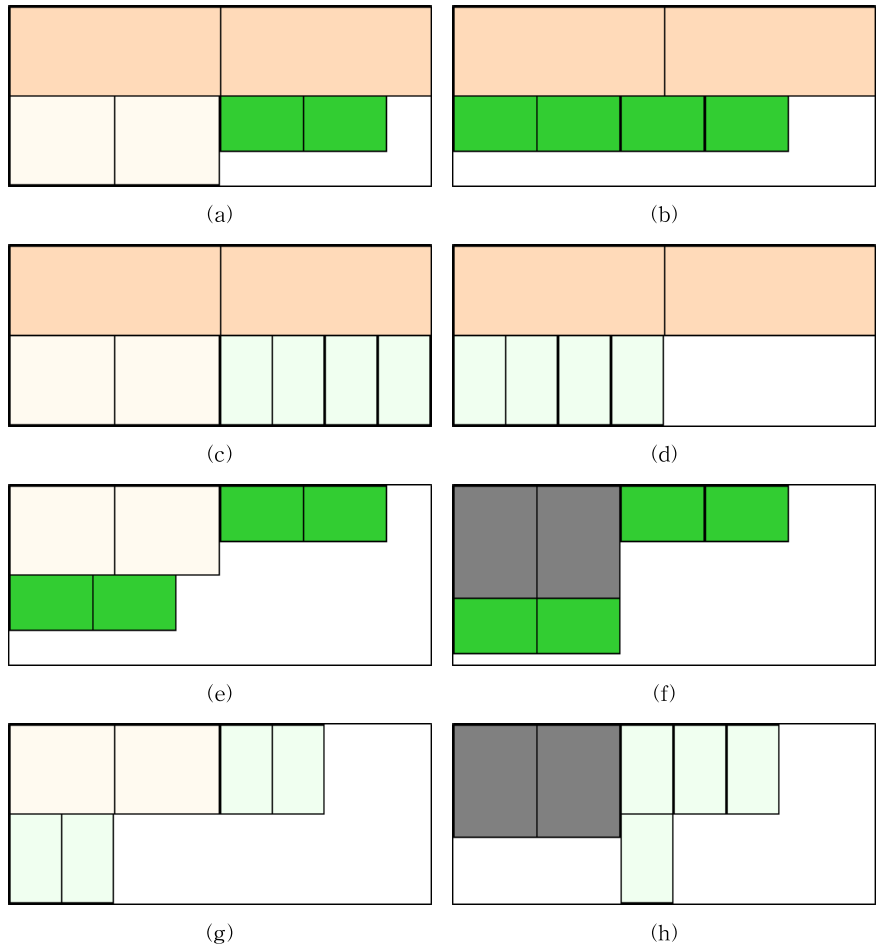


Fig. 16. Packing pattern in a 20FT container by MaxRects algorithm with 3 different dimensions and 2 different directions of rectangles as Table 2.

Table 3. The combination of 3 different dimensions and 2 different directions of rectangles

Pattern	1/2 size load (2948mm×1174mm)	1/4 size load (1474mm×1174mm)	1/8 size load (1174mm×737mm)
(a)	longitudinal	longitudinal	longitudinal
(b)	longitudinal	lateral	longitudinal
(c)	longitudinal	longitudinal	lateral
(d)	longitudinal	lateral	lateral
(e)	lateral	longitudinal	longitudinal
(f)	lateral	lateral	longitudinal
(g)	lateral	longitudinal	lateral
(h)	lateral	lateral	lateral

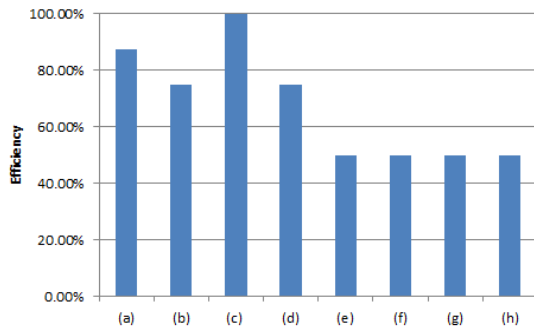


Fig. 17. Packing efficiency in a 20FT container by Max-Rects algorithm with 3 different dimensions and 2 different directions of rectangles as Table 2.

줄 수 있을 것으로 예상된다. 이는 변형된 Maximal Rectangles 알고리즘들의 약점 또한 보완해 줄 수 있을 것으로 기대된다.

## REFERENCE

- [ 1 ] G. Wäscher, H. Haußner, and H. Schumann, "An Improved Typology of Cutting and Packing Problems," *European Journal of Operational Research*, Vol. 183, No. 3, pp. 1109-1130, 2007.
- [ 2 ] A. Bortfeldt and G. Wäscher, "Constraints in Container Loading-A State-of-the-Art Review," *European Journal of Operational Research*, Vol. 229, No. 1, pp. 1-20, 2013.
- [ 3 ] K.J. Nurmela and P.R. Östergård, "Packing up to 50 Equal Circles in a Square," *Discrete and Computational Geometry*, Vol. 18, No. 1, pp. 111-120, 1997.
- [ 4 ] E.G. Birgin and J.M. Gentil, "New and Improved Results for Packing Identical Unitary Radius Circles within Triangles, Rectangles and Strips," *Computers and Operations Research*, Vol. 37, pp. 1318-1327, 2010.
- [ 5 ] D.M. Wiirtz, M. Monagan, and R. Peikert, "The History of Packing Circles in a Square," *Maple Technical Newsletter*, pp. 35-42, 1994.
- [ 6 ] R.L. Graham and B.D. Lubachevsky, "Repeated Patterns of Dense Packings of Equal Disks in a Square," *The Electronic Journal of Combinatorics*, Vol. 3, No. R16, pp. 1-17, 1996.
- [ 7 ] J. Jylänki, "A Thousand Ways to Pack the Bin-a practical Approach to Two-dimensional Rectangle Bin Packing," <http://clb.demon.fi/files/RectangleBinPack.pdf> (accessed Nov., 3, 2010).
- [ 8 ] B. Chazelle, "The Bottomn-left Bin-packing Heuristic: An Efficient Implementation," *IEEE Transactions on Computers*, Vol. C-328, Issue 8, pp. 697-707, 1983.
- [ 9 ] L. Wei and E.J. Lee, "Real-Time Container Shape and Range Recognition for Implementation of Container Auto-Landing System," *Journal of Korea Multimedia Society*, Vol. 12, No. 6, pp. 794-803, 2009.
- [ 10 ] Y.T. Feng, K. Han and D.R.J. Owen, "Filling Domains with Disks: An Advancing Front Approach," *International Journal for Numerical Methods in Engineering*, Vol. 56, pp. 699-713, 2003.
- [ 11 ] S. Polyakovsky and M. Rym, "An Agent-based Approach to the Two-dimensional Guillotine Bin Packing Problem," *European Journal of Operational Research*, Vol. 192, No. 3, pp. 767-781, 2009.
- [ 12 ] A. Watt and P.F. Policarpo, *3D Games, Vol. 2: Animation and Advanced Real-Time Rendering*, Addison-Wesley Longman Publishing Co., Inc., 2003. (21415 Network Place Chicago, IL 60673 United States)



류 기 상

밀양대학교 컴퓨터 공학 석사  
2015년~현재 ㈜호원소프트 대표  
이사  
2004년~2014년 ㈜라이온플러스  
개발총괄 부장 근무  
2002년~2004년 ㈜엔와이드 근무

관심분야: 소프트웨어 공학, U-IT 기술을 활용한 항만  
물류시스템 개발, IOT 기반 사물인터넷 시스  
템 개발 , 해양수산연구정보시스템 개발 등



박 준 선

양산대학 멀티미디어학 석사  
2015년~현재 ㈜호원소프트 개발  
팀 근무  
2007년~2014년 ㈜라이온플러스  
근무  
관심분야: 소프트웨어 공학, IOT  
모바일 개발 등