

Stack Bin Packing Algorithm for Containers Pre-Marshalling Problem

Sang-Un Lee *

Abstract

This paper deals with the pre-marshalling problem that the containers of container yard at container terminal are relocated in consensus sequence of loading schedule of container vessel. This problem is essential to improvement of competitive power of terminal. This problem has to relocate the all of containers in a bay with minimum number of movement. There are various algorithms such as metaheuristic as genetic algorithm and heuristic algorithm in order to find the solution of this problem. Nevertheless, there is no unique general algorithm that is suitable for various many data. And the main drawback of metaheuristic methods are not the solution finding rule but can be find the approximated solution with many random trials and by coincidence. This paper can be obtain the solution with $O(m)$ time complexity that this problem deals with bin packing problem for m stack bins with descending order of take out ranking. For various experimental data, the proposed algorithm can be obtain the optimal solutions for all of data. And to conclude, this algorithm can be show that most simple and general algorithm with simple optimal solution finding rule.

▶ Keyword : Pre-marshalling, Bay, Stack, Tier, Bin packing

I. Introduction

해상운송을 하는 컨테이너들은 컨테이너 터미널의 야적장 (storage yard)에 일시적으로 보관되었다가 선박에 선적된다. 전 세계적인 컨테이너 물동량 증가에 따라 무역항들의 경쟁도 날로 치열해지고 있다. 이러한 상황에서, 항구의 경쟁력을 갖추려면 입항하는 컨테이너 선박을 신속하게 출항시킬 수 있도록 하기 위해, 선박에 실려진 컨테이너를 양하 (unload)하거나 선박으로 적하 (load)하는 하역작업을 빠르게 수행하여 터미널의 생산성을 높이는 것이 관건이다[1].

부두의 컨테이너 야적장 (storage yard)은 효율적인 공간 활용을 위해 아래에서 위로 쌓아 보관하게 된다. 이 경우, 컨테이너를 반출하는 순서는 선박에 컨테이너를 싣는 적하 순서에 따라 결정되며, 이 컨테이너들은 야적장의 위에서 아래로 순서대로 반출될 수 있다. 그러나 아래쪽에 해당 컨테이너가 위치하고, 이후에 반출될 컨테이너들이 이 컨테이너 위에 놓여 있으면 이후에 반출될 컨테이너들을 치워야 하는 비생산적인 작업

이 추가로 필요하다. 이를 컨테이너 재취급 (rehandling)이라 하며, 야적장 운영 능력을 저하시키는 주요 요인이 된다.

컨테이너 야적장의 적하 순서와 반출 순서의 불일치로 인해 발생하는 재취급 작업은 선박의 출항 일정을 지연시키는 직접적인 요인이 된다. 따라서 입항할 컨테이너 선박을 신속하게 출항시킬 수 있도록 하기 위해서는 사전에 컨테이너들을 사전정렬 (pre-marshalling)해야 한다[1-3].

주어진 야적 컨테이너들의 문제에 대해, 반출 우선순위가 결정되었을 때 최소의 재배치 (relocation) 횟수를 갖도록 사전에 정렬하는 문제를 컨테이너 재배치 문제 (container relocation problem, CRP) 또는 컨테이너 사전정렬 문제 (container pre-marshalling problem, CPP)라 한다[1-3].

이 문제와 관련하여 Ha와 Kim[1]과 Zhu et al.[4]는 A* 알고리즘, Tierney et al.[5]는 A*와 IDA* 알고리즘을, Bortfeldt와 Forster[6,7]는 트리 탐색법 (tree search, TS)을, Jin et al.[8]은 향상된 탐욕 알고리즘 (improved greedy algorithm, IGA)을, Gheith et al.[9]은 정수계획법 (integer programming, IP)을, Lee와

*First Author: Sang-Un Lee, Corresponding Author: Sang-Un Lee

*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Engineering, Gangneung-Wonju National University

*Received: 2015. 07. 06, Revised: 2015. 07. 27, Accepted: 2015. 08. 18.

Hsu[10]는 이웃탐색법 (neighborhood search, NS)을, Ceserta와 Voß[11,12]는 통로법 (corridor method, CM)을, Zehendner와 Feillet[13]은 분기평가법 (branch-and-price, B&P)을, Sriprabu et al.[14]은 유전자 알고리즘 (genetic algorithm, GA)을, Jovanovic et al.[15]은 다중 휴리스틱 알고리즘을 제안하였다. 이와 같이 다항시간으로 해를 찾기 위한 휴리스틱 방법들과 더불어 다항시간으로 정확한 해를 구하는 규칙이 존재하지 않는다는 가정 하에 근사 해를 다항시간으로 구하기 위한 메타휴리스틱 방법들이 적용되고 있는 실정이다.

본 논문에서는 CPP에 대해 $O(m)$ 의 선형시간으로 해를 찾아가는 규칙을 제시한 휴리스틱 알고리즘을 제안한다. 2장에서는 CPP를 정의하여 본다. 3장에서는 CPP의 해를 선형시간으로 구할 수 있는 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 다양한 사례들에 적용하여 본다.

II. Problem Description

그림 1은 부두의 야적장 (yard)을 보여주고 있다[1-3].

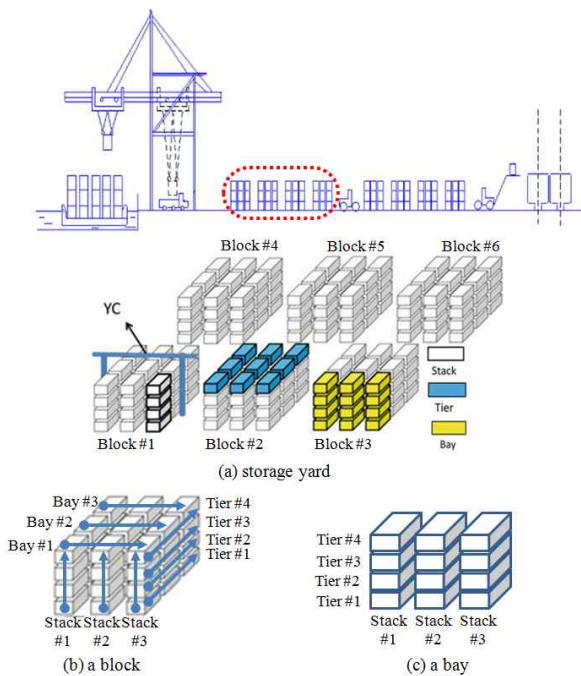


Fig. 1. Container terminal

야적장에는 컨테이너들이 (a)와 같이 블록 (block) 단위로 보관되고 있다. 하나의 블록은 다시 가로의 더미 (stacks)와 세로의 줄 (tiers)로 구성된 구역 (bays)들로 분할된다. 크레인 (yard crane, YC)은 앞과 뒤의 구역 단위 (Z)로 움직이며, 한 구역에 대해서는 좌우의 더미 단위 (X)로, 하나의 더미에 대해서는 맨 위쪽 줄 (Y)에 있는 컨테이너를 이동시킬 수 있다. 따라서 크레인은 특정 구역에서 정지한 채로 구역 내에 있는 모든 컨테이너를 취급할 수 있어 컨테이너 사전정렬은 하나의 구역을 대상으로 수행하는 것이 효율적이다. 여기서, 한 구역

내 사전 정렬 계획의 목적은 재배치 횟수를 최소화하는 재배치 순서를 도출하는 것으로 다시 말하면 다음에 도착하는 선박에 대해 가장 먼 거리에 위치한 항구에 도착하는 컨테이너는 가장 나중에 하역해야 하므로 이들 컨테이너를 먼저 선적하여 선박의 아래쪽에 배치하기 위해 야적장의 위쪽에 사전에 배치되어야 하며, 반대로 가장 가까운 부두에 도착하여 하역할 컨테이너들은 가장 나중에 선적하여 선박의 윗부분에 위치해야 하므로, 야적장 측면에서는 가장 아래쪽에 위치해야 한다.

그림 2는 Tierney et al.[5]에서 인용된 P_1 문제이다. 이 문제는 3×4 의 더미와 줄을 갖는 한 구역에 6개의 컨테이너가 보관되어 있으며, 컨테이너 식별부호:반출 우선순위가 표시되어 있다. 따라서 추후에 선박에 적재될 순서 (반출 우선순위)는 D, B, E, C, A, F 이며, 1 순위 D 를 가장 먼저 빼내기 위해서는 해당 더미의 가장 위쪽에 위치시켜야 하므로 2 순위인 B 는 다른 더미로 먼저 이동시켜야만 한다. 또한, 3 순위인 E 는 4 순위인 C 와 5 순위인 A 보다 위쪽에 있어야 하므로 C 와 A 를 다른 더미로 이동시켜야만 한다.

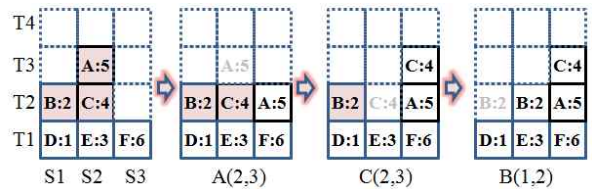


Fig. 2. P_1 problem

재배치되는 과정을 컨테이너 식별부호(출발 더미, 목적지 더미)로 표현하면 $A(2,3) \rightarrow C(2,3) \rightarrow B(1,2)$ 의 3회 이동으로 추후 선박에 원활하게 순서대로 적재하는데 문제가 없도록 사전정렬을 끝마쳤음을 알 수 있다.

부두 야적장의 컨테이너 사전정렬 문제를 다항시간으로 풀 수 있는 알고리즘은 제안되지 않고 있다. 따라서 아직까지는 메타휴리스틱 기법들을 단순히 적용하여 근사 해를 다항시간으로 얻고자 하였다. 그러나 메타휴리스틱 기법들의 문제점은 랜덤한 초기 해를 설정하고, 기율기 감소 기법을 적용해 해를 개선하여 더 이상 해가 개선되지 않으면 이 값을 해당 시행 (trial)의 결과 값으로 결정한다. 이 시행을 수 천, 수 만 회 시행으로 평균을 얻거나 최적 (최소) 값을 선택한다. 왜냐하면 기율기 감소법은 일단 지역 최적 점 (local minima)에 빠지면 이 보다 좋은 결과를 갖는 전역 최적 점 (global minima)인 최적 해에 도달하지 못하는 단점을 갖고 있다. 왜냐하면 지역 최적 점에서 탈출하여 전역 최적 점에 도달하기 위해 일단은 기율기가 증가되어야만 하나 알고리즘은 기율기가 증가하는 쪽으로는 가지 못하고 반드시 기율기가 감소하는 쪽으로만 가기 때문에 지역 최적 점에서 탈출하지 못한다. 또 한 가지 문제점은 메타휴리스틱 기법들은 해를 어떻게 찾아가는지 과정을 설명하지 못하며, 단지 해를 얻었다는 결과만을 제시할 따름이며, 랜덤한 초기치 설정으로 인해, 동일한 결과를 재현하지 못해 실제로 적용해도 동일한 결과를 얻는다는 보장이 없다.

3장에서는 본 문제에 대해 최적 해를 선형시간으로 한 번에 찾아갈 수 있는 규칙을 가진 다항시간 휴리스틱 알고리즘을 제안한다. 즉, 매 수행시마다 항상 동일한 결과를 얻을 수 있는 규칙을 가진 알고리즘이다.

III. Stack Bin Packing Algorithm

본 장에서는 컨테이너 터미널의 한 구역을 컴퓨터의 자료구조에 적용되고 있는 스택으로 연관시켜, m 개의 스택을 상자로 취급한 반출 우선순위 내림차순 상자 채우기 문제로 해를 찾는 규칙을 제시하고자 한다. 구역을 자료구조의 스택과 연관시키면 그림 3과 같다.

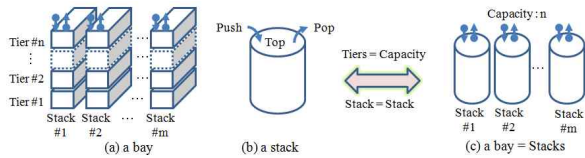


Fig. 3. Relation between a bay and stack data structure

자료구조의 스택은 LIFO (Last-In-First-Out) 구조로, Top에서 만 자료를 Push와 Pop이 발생하는 구조이다. 구역의 Stack을 자료구조의 Stack으로, Tier는 용량 (capacity)으로 대응시키면 한 구역은 용량 n 을 가진 m 개 스택으로 변환시킬 수 있다. 또한, 컨테이너의 스택에 대한 Push와 Pop은 자료구조의 스택과 동일한 LIFO 개념을 적용할 수 있다.

본 장에서는 용량이 n 인 m 개의 스택 상자들을 반출 우선순위 내림차순으로 채우는 방법을 제안한다. 제안된 방법을 스택 상자 채우기 알고리즘 (stack bin packing algorithm, ABPA)이라 하며, 다음과 같이 수행된다.

Step 1. 스택 분할

m 개 스택들을 right-ordered (R)와 wrong-ordered (W) 집합으로 분할

여기서 스택에 넣은 (쌓아 놓은) 컨테이너의 반출 우선순위가 내림차순인 경우를 right-ordered라 한다.

Step 2. W 집합 스택을 R 집합 스택으로 변환

Until $|W|=0$ do

W 의 top에 위치한 컨테이너 반출 우선순위 r_i 와 R 스택의 top에 위치한 컨테이너 반출 우선순위 r_j 에 대해,

if $r_i \leq r_j$ then 해당 컨테이너 x 를 S_i 에서 S_j 로 이동 : $x(S_i, S_j)$.

else 이동시킬 컨테이너의 반출 우선순위 최대값 $\max r_i$ 와 이동을 시키지 않아도 되는 컨테이너의 반출 순위 최대값 $\max r_j$ 에 대해

if $\exists r_j \in S_j, r_i \leq r_j$ then $r_j \in S_j$ 스택에 대해 r_j 위쪽 컨테이너들을 W 옴

else if $\nexists r_j \in S_j, r_i \leq r_j$ then $\min |S_i| \in W$ 스택을 비움

endif

$\max r_j$ 컨테이너부터 비워진 스택 S_i 를 채움
end
end

그림 2의 P_1 데이터에 제안된 ABPA를 적용한 결과도 그림 4와 같이 동일한 결과를 나타낸다. ABPA는 첫 번째로, $W = \{S_1, S_2\}, R = \{S_3\}$ 로 분할되었으며, 첫 번째로, $F:6 \in R \geq A:5 \in W$ 로 $A(S_2, S_3)$ 으로 이동된다. 다음으로, $A:5 \in R \geq C:4 \in W$ 로 $C(S_2, S_3)$ 으로 이동된다. 마지막으로 남은 $B \in W$ 와, $W = \{S_1\}, R = \{S_2, S_3\}$ 에 대해 $C:4 \in R \geq B:2 \in W$ 또는 $E:3 \in R \geq B:2 \in W$ 으로 $B(S_1, S_2)$ 또는 $B(S_1, S_3)$ 으로 이동된다.

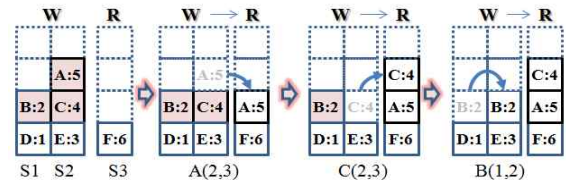


Fig. 4. Solution of ABPA for P_1 problem

IV. Applications and Evaluation

본 장에서는 그림 5의 7개 실험 데이터를 대상으로 제안된 ABPA의 적합성을 검증하여 본다.

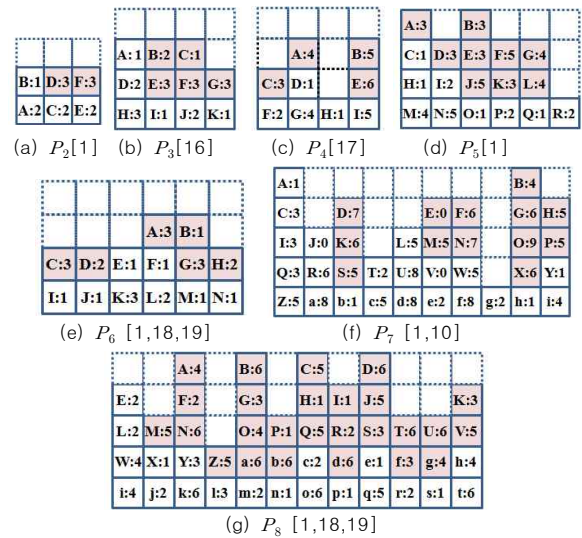


Fig. 5. Experimental data

그림 5의 실험 데이터에 제안된 알고리즘을 적용한 결과는 그림 6에 제시되어 있다. P_2 의 경우 $|S_2|=|S_3|=0$ 동일하며, $D:3=F:3 \leq x \notin S_1$ 으로 S_2 나 S_3 중 어느 하나의 스택을 비워야 한다. 여기서는 S_2 와 S_3 를 비우는 2가지 경우 모두 5회가 수행되었음을 알 수 있다.

P_3 의 경우 $C(S_3, S_1) \rightarrow G(S_4, S_3)$ 이 수행된 후 $|S_4|=1$ 로 $K(S_4, S_2)$ 로 S_4 를 비운 후 $\max r_j = 3$ 인 $G(S_3, S_4) \rightarrow F(S_3, S_4)$ 를 수행하면 $W = \{S_2\}, R = \{S_1, S_3, S_4\}$ 가 된다. 따라서 $K(S_2, S_4) \rightarrow B(S_2, S_3)$ 을 수행하면 $E: 3$ 이 S_3 과 S_4 에 갈 수 없으므로 부득이 $K(S_4, S_3)$ 을 수행하고 $E(S_2, S_4)$ 를 수행하였다. 따라서 9회가 수행된다.

P_4 의 경우 $R = \{S_3\}, W = \{S_1, S_2, S_4\}$ 이며, $H: 1 \in S_3$ 에 추가될 컨테이너가 W 에는 없음을 알 수 있다. 따라서 $H_1(S_3, S_1)$ 으로 이동시켜 S_3 을 비워둔 상태에서 $\max r_j = 6$ ($E: 6$)이므로 $B(S_4, S_3)$ 를 수행하지 않고 $B(S_4, S_2) \rightarrow E(S_4, S_3) \rightarrow B(S_2, S_4) \rightarrow A(S_2, S_3) \rightarrow H(S_1, S_2) \rightarrow C(S_1, S_3)$ 로 7회가 수행되었다. 동일한 방법으로 P_5 는 13회, P_6 은 9회, P_7 은 15회, P_8 은 35회가 수행되었다.

본 논문에서 거론된 데이터에 대해 알려진 해와 ABPA를 비교한 결과는 표 1과 같다.

Table 1. Compare with algorithm performances

문제	컨테이너 이동 횟수							
	[18]	[19]	[10]	[12]	[1]	[17]	[5]	ABPA
P_1	-	-	-	-	-	-	3	3
P_2	-	-	-	-	5	-	-	5
P_3	-	-	-	-	-	-	-	9
P_4	-	-	-	-	-	7	-	7
P_5	-	-	-	-	-	-	-	13
P_6	9	9	-	-	9	-	-	9
P_7	-	-	31	19	15	-	-	15
P_8	35	47	-	-	30	-	-	30

[1]은 A*, [5]는 A*와 IDA*, [10]은 NS, [12]는 CM, [17]은 LPF (lowest priority first), [18]은 2가지 종류의 휴리스틱 알고리즘 (heuristic algorithm, HA)을, [19]는 다중상품 망 흐름 문제를 가진 정수계획법 (integer programming with multi-commodity network flow problem, IP-MCNFP)을 적용하였다. 표에서 알 수 있듯이 8개의 문제에 대해 기존의 알고리즘은 특정 데이터에만 적용하였으며, 다양한 많은 데이터에 적용하여 일반화된 모델로서 검증을 하지 못하였다. 반면에, 제안된 ABPA는 8개 데이터 모두에 대해 최적 해를 얻어 기존 알고리즘들에 비해 보다 일반화된 알고리즘이라 할 수 있다. 또한, 제안된 알고리즘은 m 개의 스택에 대해 수행되므로 단순히 $O(m)$ 의 선형 복잡도를 가진 다항시간 알고리즘으로도 유전자 알고리즘과 같은 메타휴리스틱 방법들과 동일한 해를 얻을 수 있는 장점을 갖고 있다.

V. Conclusions

본 논문은 항구의 야적장에 보관된 컨테이너들을 추후에 선박에 적재될 우선순위로 반출이 가능하도록 사전에 정렬하는 문제를 다루었다.

본 문제는 항구의 경쟁력 우위 확보를 위해 해결해야 할 반드시 필요한 문제들 중 하나이다. 즉, 항구에 입출항하는 선박

의 선적이나 하역 작업 시간을 최소화시켜 선박 회전율 시간을 최소화해야 한다. 이러한 컨테이너 사전정렬 문제의 해는 주어진 구역에서 최소한의 이동횟수로 반출순위를 맞추어야만 한다.

지금까지는 컨테이너 사전정렬 문제의 해를 구하기 위해 유전자 알고리즘을 비롯한 메타휴리스틱 뿐만 아니라 다양한 휴리스틱 방법들이 제안되었다. 그러나 다양한 많은 데이터에 모두 적합한 보다 일반화된 유일한 모델이 없는 실정이며, 메타휴리스틱 기법들은 매 시행마다 다른 결과를 얻으며, 요행히도 최적 해를 얻었다 해도 이를 찾아가는 규칙을 설명하지 못해 동일한 결과를 재현이 불가하여 유사한 문제에 적용할 수 없었다.

본 논문은 $O(m)$ 의 선형시간 복잡도로 컨테이너 사전정렬 문제의 해를 찾아가는 규칙을 제시하였다. 제시된 규칙은 부두 야적장의 한 구역에 대해 m 개의 스택들로 구성된 상자 (용량이 n 줄)에 대해 반출 순서 내림차순으로 저장되도록 하는 상자 채우기 문제로 취급하였다.

다양한 문제들에 적용한 결과, 개별적으로 적용된 기존의 메타휴리스틱 기법들의 결과와 비교시 제안된 알고리즘은 모든 데이터에 대해 최소의 컨테이너 이동 횟수를 보여 가장 단순하면서도 선형시간으로 빠르게 해를 구할 수 있는 알고리즘임을 알 수 있다. 따라서 컨테이너 터미널의 야적장 관리를 효율적으로 하기 위해서는 제안된 알고리즘이 큰 도움이 될 것이다.

REFERENCES

- [1] B. H. Ha and S. S. Kim, "A* Algorithm for Optimal Intra-Bay Container Pre-Marshalling Plan," Journal of the Korean Institute of Industrial Engineers, Vol. 38, No. 2, pp. 157-172, Jun. 2012.
- [2] M. S. Gheith, A. B. Eltawil, and N. A. Harraz, "A Proposed Heuristic for Solving the Container Pre-Marshalling Problem," The 19th International Conference on Industrial Engineering and Engineering Management, pp. 955-964, Apr. 2013.
- [3] M. S. Gheith, A. B. Eltawil, and N. A. Harraz, "Container Pre-Marshalling Problem: Review and Solution Framework," The IEEE International Conference on Industrial Engineering and Engineering Management, At Hong Kong, pp. 283-287, Oct. 2012.
- [4] W. Zhu, H. Qin, A. Lim, and H. Zhang, "Iterative Deepening A* Algorithms for the Container Relocation Problem," IEEE Trans. on Automation Science and Engineering, Vol. 9, No. 4, pp. 710-722, Oct. 2012.
- [5] K. Tierney, D. Pacino, and S. Voß, "Solving the Pre-Marshalling Problem to Optimality with A* and IDA*," Conference of the International Federation of

- Operational Research Societies., Barcelona, Spain, Mar. 2014.
- [6] A. Bortfeldt and F. Forster, "A Tree Search Procedure for the Container Pre-Marshalling Problem," *European Journal of Operational Research*, Vol. 217, No. 3, pp. 531-540, Mar. 2012.
- [7] F. Forster and A. Bortfeldt, "A Tree Search Procedure for the Container Relocation Problem," *Computers & Operations Research*, Vol. 39, No. 2, pp. 299-309, Feb. 2012.
- [8] B. Jin, W. Zhu, and A. Lim, "Solving the Container Relocation Problem by an Improved Greedy Look-Ahead Heuristic," *European Journal of Operational Research*, Vol. 240, No. 3, pp. 837-847, Feb. 2015.
- [9] M. S. Gheith, A. B. Eltawil, N. A. Harraz, and S. Mizuno, "An Integer Programming Formulation and Solution for the Container Pre-Marshalling Problem," *Proceedings of 44th International Conference on Computers and Industrial Engineering & 9th International Symposium on Intelligent Manufacturing and Service Systems*, Istanbul, Turkey, pp. 2047-2056, Oct. 2014.
- [10] Y. S. Lee and N. Y. Hsu, "A Neighborhood Search Heuristic for Pre-Marshalling Export Containers," *European Journal of Operational Research*, Vol. 196, No. 2, pp. 468-475, Jul. 2009.
- [11] M. Caserta, S. Voß, and M. Sniedovich, "Applying the Corridor Method to a Blocks Relocation Problem," *OR Spectrum*, Vol. 33, No. 4, pp. 915-929, Oct. 2011.
- [12] M. Caserta and S. Voß, "A Corridor Method- Based Algorithm for the Pre-Marshalling Problem," *Proceedings of Applications of Evolutionary Computing, Lecture Notes in Computer Science*, Vol. 5484, pp. 788-797, Apr. 2009.
- [13] E. Zehendner and D. Feillet, "Branch and Price Approach for the Container Relocation Problem," *International Journal of Production Research*, Vol. 52, No. 24, pp. 7159-7176, Sep. 2014
- [14] P. Sriprabu, K. Sethanan, and B. Aronkijpanich, "A Solution of the Container Stacking Problem by Genetic Algorithm," *IACSIT International Journal of Engineering and Technology*, Vol. 5, No. 1, pp. 45-49, Feb. 2013.
- [15] R. Jovanovic, M. Tuba, and S. Voß, "A Multi-Heuristic Approach for Solving the Pre-Marshalling Problem," *Cornell University Library*, arXiv:1411.0967, pp. 1-19, Nov. 2014.
- [16] A. B. Eltawil, "A Systematic Approach to Container Terminal Planning and Operational Decision Making," *The International Maritime Transport & Logistics Conference*, pp. 1-14, Mar. 2013.
- [17] E. I. Christopher, M. B. Belén, and M. V. Marcos, "Pre-Marshalling Problem: Heuristic Solution Method and Instance Generator," *Expert Systems with Applications*, Vol. 39, No. 9, pp. 8337-8349, Jul. 2012.
- [18] S. H. Huang and T. H. Lin, "Heuristic Algorithms for Container Pre-Marshalling Problems," *Computers & Industrial Engineering*, Vol. 62, No. 1, pp. 13-20, Feb. 2012.
- [19] Y. S. Lee and N. Y. Hsu, "An Optimization Model for the Container Pre-Marshalling Problem," *Journal of Computers and Operations Research*, Vol. 34, No. 11, pp. 3295-3313, Nov. 2007.

Authors



Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively. He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.