

Hatch를 고려한 컨테이너 터미널 적재순서 최적화 연구

이상현^{1*} · 김문규¹ · 안태호²

¹국방대학교 운영분석학과 / ²송실대학교 경상대학 경영학부

A Study on the Optimization of Stowage Planning for Container Terminal Considered by Hatch

Sang-Heon Lee¹ · Moon-Gyu Kim¹ · Tae-Ho Ahn²

¹Department of Operations Research, Korea National Defense University, Seoul, 122-875

²Department of Management, Soongsil University, Seoul, 156-743

The shipping plan for an efficient augmentation of terminal is NP-hard problem which is subordinate to a shipping quantity. The load location problem and stowage planning problem are two important tactical problems for the efficient operations of container terminals. This paper is concerned with reduction of feasible region in constraints for stowage planning. After designing model which minimizes the number of shifting and the travel distance of the transfer crane, the simulated annealing algorithm is employed to search optimal solution quickly and accurately. In order to apply more realistic approach, the partial restriction through adding hatch is complemented to stowage planning decision problem applied in cluster. A variety of numerical experiments demonstrated that solutions by simulated annealing algorithm are useful and applicable in practice.

Keyword: stowage planning, simulated annealing, container terminal, cluster, hatch

1. 서론

현재 국제시장은 단일화 및 개방화의 추세에 따른 국제무역의 양적인 증대와 기존의 운송방법에 컨테이너라는 규격화된 용기를 사용하는 복합화물운송(multi-modal transportation) 및 일괄수송체계의 보편화로 인해 컨테이너 수송량은 급속히 증대되고 있으며, 고객의 수요에 대응하고 비용감소를 통해 타 선사와의 경쟁력을 확보하고자 컨테이너선의 크기를 대형화하고 있는 추세이다. KMI 세계해운전망(2003)에서는 우리나라 해상물동량을 연도별로 판단하였는데 1999년에 697만 TEU, 2002년에는 1,180만 TEU로서, 2011년까지는 연평균 8.8%로 증가할 것으로 예상하고 있다. 또한 컨테이너 터미널에서 발생하는 비용이 컨테이너 총 수송원가의 상당 부분을 점유하고 있고 터미널에서의 작업 소요시간 증가와 컨테이너 운송을 위한 전용선

운항이 주를 이루면서, 각국의 주요 항만은 터미널 운영방식을 개선하는 등의 운영효율화에 노력의 박차를 가하고 있다.

늘어나는 컨테이너 처리량에 대해 효율적으로 대처하고 터미널의 경쟁력을 향상시키기 위해 비용 대 효과면에서 우수한 기존 운영방식 개선에 중점적으로 연구가 이루어지고 있으며 생산성을 결정하는 한 요소인 컨테이너 적·양하를 위한 터미널 하역작업계획이 이에 속한다. 이러한 계획수립의 기본 목적은 컨테이너의 적·양하작업에 소요되는 하역시간의 단축으로 각종 부대비용 절감 및 서비스 향상에 따른 다수의 선사유치 효과를 기대하는데 있다.

터미널 하역작업계획 중 적하계획은 선사에서 전달된 선적지시서의 요구사항에 따라 터미널에서 야드 및 선박의 장비운영계획과 컨테이너 목적지, 유형, 무게 등을 고려해 선박의 지정된 적하위치에 야드 상의 수출 컨테이너를 일정한 순서로

*연락처 : 이상현 교수, 122-875 서울시 은평구 수색동 205번지 국방대학교 운영분석학과, Fax : 02-309-6233, E-mail : leesangh@kndu.ac.kr
2006년 01월 접수, 1회 수정 후 2006년 09월 게재확정.

신도록 계획하는 것으로 상당히 복잡하며, 수립여부에 따라 실 작업시간 및 비용에 상당한 영향을 미친다. 또한, 적하계획은 적하량에 종속되어 수많은 경우의 수를 판단해야 되는 NP-hard 문제에 속하며, 그 속성상 크게 적재위치 결정문제와 적재순서 결정문제로 분류되어 연구되었다(Lee *et al.*, 2002).

적하계획을 수립하는 과정은 선사 및 터미널에서 발행되는 정보를 바탕으로 터미널의 플래너에 의해 여러 단계로 나누어져 작성된다. 선박은 기항지(Port of destination)를 순환하는 정기선의 형태로 도착하는 기항지의 순서에 따라 내리고 실어야 할 컨테이너를 결정해 각 컨테이너의 정보가 저장되어있는 선적지시서를 발행하며, 야드는 수출 및 수입 컨테이너를 임시적으로 보관하는 곳으로 컨테이너에 대한 상세한 정보를 야드 맵에 작성하여 보관한다. 플래너는 이러한 선적지시서와 야드 맵을 바탕으로 선적계획, 크레인 일정계획, 적하 및 양하계획 등을 수립한다(Jang *et al.*, 2002).

크레인 일정계획은 하나의 선박에 몇 대의 크레인을 투입할 것인지 결정하는 것으로, 작업물량에 따라 선박 한 대당 2~5대의 CC(Container Crane)가 투입되며 각 CC가 작업해야 할 선박의 bay 영역과 작업순서가 결정된다. 따라서 선박의 각 bay 단위로 적재순서를 결정할 수 있으면 크레인 일정계획을 고려하여 전체의 적재순서를 결정할 수 있다(Lee, 2004).

본 논문은 컨테이너 터미널의 효율성을 증가시키기 위한 연구로서 선창(hatch)이 고려된 선박 bay를 기준으로 수출 컨테이너의 적재순서를 결정하기 위해 클러스터 순서조합 최적화문제로 전환시켜 재처리 횟수 및 TC(Transfer Crane) 이동거리를 최소화하는 적재순서 결정모형을 제시한다. 또한, 이 모형에 대한 해법절차로 대규모의 순서조합 최적화 문제에서도 효율적이고 해수렴속도가 빠른 SA 알고리즘을 도입하여 해 유도과정으로 구성한다. 특히 SA는 크고 복잡한 순서조합 최적화문제에 대해서도 비교적 짧은 계산시간으로 근사 최적해를 제공할 수 있기 때문에 장비고장 등의 우발상황 발생시 빠르게 대처할 수 있어 터미널의 효율적인 운영에 크게 기여할 수 있을 것으로 판단된다.

본 논문의 구성은 다음과 같다. 2장에서는 적하계획 최적화방안에 관한 기존 논문을 고찰하고, 특히 클러스터를 이용한 모형과 해법에 대해 자세히 분석한다. 3장에서는 hatch를 고려한 적재순서 최적화 모형을 설계하고, 4장에서는 최적해를 효율적으로 탐색할 수 있는 SA를 이용한 해 유도과정을 제시한다. 5장에서는 실험을 통한 도출결과를 분석하고, 6장에서는 결론 및 향후 연구방향을 제시한다.

2. 기존연구 고찰

2.1 적하문제 접근방법

적하순서 결정문제는 터미널 및 장비의 효율적 운영에 중점을 둔 것으로 수학적 접근방법, 휴리스틱을 이용한 접근방법, 메타 휴리스틱을 이용한 접근방법 등 여러 방법에 의해 연구

가 진행되었다.

수학적 접근방법을 살펴보면, Cho(1982)는 정수계획법을 이용한 수학적 모델을 제시하였으며 Botter and Brinati(1992)는 선박의 안정성과 크레인의 움직임, 컨테이너 적하순서와 같이 적하계획과 관련된 다양한 특성을 고려한 정수계획법 모델을 제시하고 명시적 계산방법으로 해를 구하였다. 그러나 수학적 접근방법은 소규모 계획에서는 정확하였지만 실질적인 문제 크기에 따라 계산량이 기하급수적으로 늘어나 대규모의 계획에 적용이 불가능하였다.

휴리스틱을 이용한 접근방법은 Kang and Lee(1990)가 선박의 복원성과 야드 컨테이너의 기대 재처리 횟수를 최소화 하는 정수계획모형을 제시하고 분지한계법으로 초기해를 구하여 동적계획법으로 최적해를 도출하였다. Avriel *et al.*(1998)은 직사각형의 hatch가 없는 하나의 bay를 대상으로 오직 재처리 횟수를 줄일 수 있는 간단한 휴리스틱 기법으로 빠른 시간에 해를 구할 수 있는 방법을 제시하였다. Kang and Kim(2000)은 선박의 안정성을 유지하는 동시에 적하와 양하에 필요한 시간을 최소화하는 선박의 화물배열을 찾아내기 위해, hatch에 컨테이너 그룹을 할당하는 방법과 각 slot에 개별의 컨테이너를 할당하는 두 단계로 나누어 해를 찾았다. 그러나 휴리스틱을 이용한 접근방법은 대규모 계획시 절차가 복잡해지며 해의 정확도가 상대적으로 낮은 경향이 있다.

메타 휴리스틱을 이용한 접근방법은 Lee(2002)와 Ryu *et al.*(2001)이 선박과 야드의 클러스터 단위로 적하계획을 수립하는 방법을 제시하였으며, 개미 알고리즘과 빔 탐색법을 이용하여 해를 도출하였다. Wilson and Roach(1999)와 Wilson *et al.*(2001)은 일반화된 컨테이너들을 블록화된 장치장(Cargo-space)에 할당하고, hatch에 대응하는 slot들을 함께 그룹화하는 전략 계획과정과 특정의 컨테이너들이 결정된 block 안에 특정한 slot에 할당하는 전술계획과정으로 나누어, 분지한계법과 타부서치로 해를 구하였다. Kim *et al.*(1997)은 탐색공간의 수를 줄이기 위해 각 단계마다 불필요한 노드를 제거한 후 계속 탐색해 나가는 필터 빔 탐색법에 의해 최적해를 도출하였다. Lee(2004)는 재처리를 고려한 클러스터를 구성하여 관계를 결정하였으며, 클러스터간의 순위제약을 위배하지 않는 해의 범위 내에서 TC의 이동거리를 평가요소로 하는 유전자 알고리즘을 이용하여 해를 도출하였다. 그러나 기존의 메타 휴리스틱에 의한 접근방법도 계산시간 단축 및 국부해에 빠지지 않는 전략이 추가적으로 필요하게 되었으며 hatch를 미 고려한 상황설정으로 실 작업환경과 다소 차이가 발생되었다. Akio *et al.*(2006)은 선박의 안정성과 컨테이너 재처리횟수를 최소화하는 다목적 정수계획모형을 제시하고 유전자알고리즘을 이용하여 실용적인 해를 구하였다.

이상과 같이 기존 연구들을 고찰해 본 결과, 적재순서 결정시 클러스터 구성법과 메타 휴리스틱을 적용한 해 탐색방법이 효율적이고 많이 연구되고 있는 추세로 본 논문에서는 hatch를 직접 고려한 계층적 클러스터 단위 적재순서 결정모형을 설계

하고, 적하계획 해법절차로 한 번도 사용되지 않았던 SA 알고리즘을 적용하여 적재순서를 결정하였다.

2.2 클러스터를 이용한 기존모형 분석

2.2.1 클러스터 단위 적재순서모형

일반적으로 선적지시서나 야드맵에서는 목적지, 유형, 크기 등 동일한 종류의 컨테이너들이 인접한 위치에 있어 함께 처리되는데 이를 클러스터라 한다. 클러스터를 이용해 해의 범위를 축소시켜 최적해를 탐색하는 해법에 대한 연구가 적재순서 문제를 해결하기 위한 효율적인 한 방법으로 제시되었다.

Ryu *et al.*(2001)은 클러스터 단위 적재순서모형을 제시하였는데 2단계의 계층적 구조로 가정하여 1단계는 <Figure 1>과 같이 hatch를 미 고려한 선박과 야드에 동일한 특성을 가진 클러스터를 만들고 이들 간의 작업순서를 개미 알고리즘을 적용해 선박 클러스터에 야드 클러스터를 할당해 주었다. 2단계는 1단계의 결과를 바탕으로 적재순서에 따른 평가값을 이용하여 빔 탐색법과 타부서치로 컨테이너 개개의 작업순서를 결정하는 알고리즘을 개발하였다.

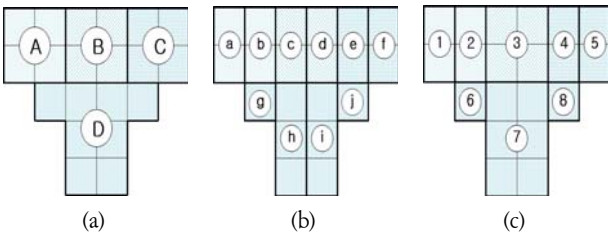


Figure 1. The method divided by cluster's unit

<Figure 1>은 선박 클러스터 분할의 예를 보여주며 야드 클러스터도 동일하게 수행한다. (a) 단계는 동일한 컨테이너를 병합한 것이며, (b) 단계는 작업순서 제약을 완화하고 작업유연성을 위하여 stack단위로 모든 클러스터를 구분한 것이다. 최종적인 (c) 단계에서는 분리된 클러스터 c와 d, h와 i를 동일한 특성을 가진 클러스터로 다시 병합하여 <Figure 2>와 같이 각 클러스터간 작업 선후관계행렬(ordering matrix)을 작성하여 이를 기준으로 작업순서를 어기지 않는 모형을 제시하였다. <Figure 2>에서 행렬의 각 행은 클러스터 번호를 의미하며 자신보다 먼저 처리되어야 하는 클러스터 번호의 열에 1을 표시한 것이다.

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Figure 2. An example of ordering-matrix

그러나, 이 모형은 선박의 하단부분인 hold와 상단부분인

deck를 구분하는 hatch를 미 고려하여 hatchcover를 기준으로 hold → deck순의 stack 단위로 이루어지는 실 작업순서를 제대로 반영하지 못하였으며 선박 클러스터의 작업순서를 결정 한 후 야드 클러스터를 할당해 주는 방식은 두 제약조건을 동시에 고려하지 못해 국부해로 도달하거나 최적해를 탐색시 시간이 오래 걸리는 결과를 초래한다. 또한 클러스터 재구성 단계가 없어 선박 클러스터의 컨테이너 수에 야드 클러스터의 컨테이너수를 맞추기가 어려우며 병합 후에 추가적으로 발생되는 클러스터간의 제약조건이 고려되지 못하였다.

2.2.2 클러스터 재구성을 통한 적재순서모형

기존의 클러스터 단위 적재순서모형을 기초로 Lee(2004)는 계층적 클러스터링 기법을 이용하여 최대한 재처리가 발생하지 않도록 미리 선박과 야드의 클러스터를 재구성하는 방법을 제시하였다.

클러스터 재구성을 통한 적재순서모형은 거리를 기준으로 한 bottom-up방식의 계층적 클러스터링 기법을 이용해 stack 단위까지만 클러스터를 구성하고 병합하지 않으며, 선박 클러스터는 관계를 맺은 야드 클러스터의 개수에 따라 하위 클러스터로 분할한다. Hatch를 미 고려한 선박 bay내 선박 클러스터와 야드 bay내 야드 클러스터를 계층적 클러스터링 기법을 이용하기 위해 클러스터의 위치를 정의하였으며, 정의된 위치 정보를 기준으로 클러스터간 거리가 1일 때만 병합하여 선박 클러스터와 야드 클러스터를 구성하고 작업가능 순위에 따라 다시 분류하였다.

Stack 단위의 클러스터가 구성되면 작업가능 순위를 바탕으로 한 다음과 같은 규칙을 기준으로 클러스터 재구성 과정을 거쳐 선박 및 야드 클러스터를 병합하고 각각의 관계를 정의하였다.

- (1) 순서가 동일하고 클러스터내 컨테이너수가 동일한 경우
- (2) 순서가 동일하고 클러스터내 컨테이너수가 틀린 경우
 - 컨테이너수가 많은 클러스터를 분할
- (3) 순서가 틀리고 클러스터내 컨테이너수가 동일한 경우
 - 순서가 높은 클러스터의 순서를 따름
- (4) 순서가 틀리고 클러스터내 컨테이너수가 틀린 경우
 - 순서가 높은 클러스터의 순서를 따르고 컨테이너수가 많은 클러스터를 분할

이러한 과정에서 TC의 재처리 발생여부를 알 수 있어 재처리가 발생하지 않는 범위내에서 TC 이동거리 최소화를 평가요소로 탐색기법인 유전자 알고리즘을 적용하여 최적해를 도출하였다. 그러나 이 방법도 기존 모형과 같이 hatch를 미 고려하여 실 작업순서와 차이가 발생되었고, 클러스터 재구성 과정에서는 컨테이너 위치에 따라 재처리가 발생될 수밖에 없는 상황이 발생하여 분류기준을 적용하지 못하는 경우가 발생되었다. 또한 작업 우선순위 결정시 선박과 야드 상의 컨테이너 배치에 따라 작업의 우선순위를 결정할 수 없는 상황이 발생

되고 이를 조정하기 위해 클러스터간 교환방법을 적용하였으나 이 방법으로 해결되지 않는 상황이 초래되었다.

2.2.3 기존 해법 분석

일반적으로 클러스터를 이용한 적재순서 결정문제는 클러스터의 작업순서를 방문순서 결정문제인 TSP(Traveling Salesman Problem)와 유사한 문제로 전환시켜 접근하였다. TSP 문제는 대표적인 NP-hard 문제로 정수계획법에 의한 해법들로 정확한 해를 구할 수는 있지만 문제의 크기가 커질수록 경우의 수가 기하급수적으로 증가하여 계산시간이 지수적으로 증가하게 된다. 그리고 최근 컨테이너선의 크기가 커짐에 따라 모형 구조가 매우 복잡하고 문제의 규모 역시 대단히 방대하게 모델링되는 추세이기 때문에 오랜 계산시간동안 정확한 해를 찾기보다는 합리적인 시간내에 근사 최적해를 제공할 수 있는 메타 휴리스틱 접근방법이 효율적으로 판단된다.

클러스터를 이용한 적재순서 결정문제의 해를 구하기 위한 기존 해법으로는 개미시스템(ant system)과 빔 탐색법(beam search)(Kim *et al.*, 2000), 타부탐색(tabu search)(Wilson *et al.*, 2001), 유전자 알고리즘(Lee, 2004) 등 메타 휴리스틱에 의한 방법이 활용되어왔다. 이러한 연구들은 메타 휴리스틱 기법을 이용하여 기존의 그리디기법(greedy algorithm)에서 갖는 국부탐색(local search)의 단점을 극복하고 전역 최적해를 유도해냄으로써 이들 기법이 순서조합 최적화문제에 대한 효율적인 해법절차가 될 수 있음을 제시하였다.

그러나 개미시스템은 개미의 습성을 관찰하여 적용한 맹목적 탐색방법으로 효과적인 전역탐색을 수행하지만 대체적으로 탐색 소요시간이 길며, 타부탐색은 hill-climbing 탐색과 같이 지역적인 이웃해들을 탐색해 나가면서 tabu-list와 집중화, 다각화 전략을 이용하여 지역적 최적해를 넘어 전역적 최적해를 탐색할 수 있는 탐색기법으로 짧은 수행시간에 전역 최적해를 찾을 수 있지만 탐색공간이 매우 불규칙한 문제에서는 전역 최적해를 찾기에는 한계가 있다. 그리고 유전자 알고리즘은 자연 진화의 법칙인 적자생존과 자연도태의 원리를 바탕으로 자연계의 현상인 교차(crossover)와 돌연변이(mutation)과정에서 환경에 적합한 개체는 생존확률이 높으며 부적합한 개체는 도태되어 간다는 진화론을 일반 최적화 문제에 적용한 기법으로 그 개념과 이론이 단순하고 해의 탐색이 우수하나 좋은 해의 이용만을 강조하면 조기 수렴하여 부분 최적해로 빠지며 문제의 크기가 커짐에 따라 계산시간이 급격히 증가한다. 또한, 기존 연구들에 적용된 문제구성 자체가 hatch를 고려하지 않은 단순한 형태의 순서조합문제로 한정되었고 재처리 종류에 따른 패널티값 차등적용이나 계산시간을 효율적으로 단축해 본 사례는 매우 드물다.

클러스터를 이용한 적재순서 결정문제는 적하게 계획을 현실적으로 묘사하기 위해 복잡한 구조의 대규모 비선형문제로 확대되고 있다. 또한 터미널에서의 적재순서에 소요되는 시간은

서비스 및 비용과 직접적으로 연결되어 있어 빠른 시간에 계산할 수 있는 모형의 해법에 관한 연구들이 필수적이다. 그러나 기존 모형들은 hatch를 미 고려하여 현실성이 떨어지고 대규모 문제에서 계산시간이 기하급수적으로 많이 증가되어 실제 적용하기 곤란하다. 이를 위해서는 hatch가 고려된 현실적인 모형을 바탕으로 빠른 시간에 전역 최적해에 도달할 수 있는 보다 일반화된 탐색기법에 대한 연구를 확대시켜야 한다. 따라서 본 연구에서는 클러스터를 이용한 적재순서 결정문제를 TSP와 같은 순서조합 최적화문제로 변경하고 이를 해결하기 위해 흔히 사용하는 방법인 SA 알고리즘을 도입하여 대규모 적재순서 결정문제에 대한 해 유도과정을 구성해 그 효율성을 분석함으로써 메타휴리스틱 기법 중 SA가 대규모 적재순서 결정문제를 위한 해법이 될 수 있는가를 판단하고자 한다.

3. 적재순서 최적화모형

3.1 문제정의 및 가정사항

본 연구에서는 컨테이너 터미널의 효율성과 생산성을 향상시키기 위한 수출 컨테이너 적재순서를 결정하는 문제로 hatchcover를 가진 하나의 bay에 재처리수와 TC 총 이동거리를 최소로 하는 컨테이너 적재순서를 결정하는 것이다. 이러한 적재계획은 선박에 적재할 수 있는 컨테이너의 수에 종속된 크기의 문제로 선박의 크기가 점차 대형화 추세여서 문제의 크기 또한 점차 커지고 있다.

Hatch가 있는 bay 문제를 풀기 위해서는 한 stack에 대한 재처리는 물론이고, hatch에 의한 재처리도 고려하여야 하므로 기존의 hatch를 고려하지 않거나 강제로 할당하는 방법으로는 좋은 해를 구할 수가 없다. 기존논문들은 hatch 상황을 고려하기 위하여 미리 재처리를 고려해 hold와 deck의 컨테이너 그룹을 강제로 할당할 후에 각 그룹 내에서 최적의 해를 찾았다. 그러나 본 논문에서는 그룹화 과정을 통해 컨테이너 그룹을 강제로 할당하지 않고 클러스터 구성과정에서 hatch를 고려하여 구성하고 클러스터간 작업순위 판단시 hatch가 반영되어 컨테이너 터미널의 실 작업환경과 같은 상황을 묘사해 양질의 해를 구할 수 있다.

전체 컨테이너의 적재순서를 결정하기 위해서는 컨테이너를 CC 및 TC의 수직 작업 특성을 고려하여 hatch 및 stack 단위의 동일한 특성을 가진 클러스터로 구성하고 재처리 등 제약요소가 고려된 클러스터간 작업순서를 결정해야 한다. 클러스터 내의 개별 컨테이너 작업순서는 클러스터가 stack단위로 구성되어 있기 때문에 선박에서는 아래쪽이 우선이며 야드에서는 위쪽이 우선으로 작업순서를 부여하면 된다. 상기 적재순서 결정문제를 해결하기 위한 가정사항은 다음과 같다.

- (1) 모든 컨테이너는 20 ft 컨테이너로 크기가 같다.
- (2) 재처리 시간 및 비용은 모든 컨테이너가 동일하다.

- (3) Hatchcover의 개폐비용은 선박내의 모든 bay에서 동일하다.
- (4) Hold 내부에 빈 공간이 존재한 채로 deck에 컨테이너를 적재할 수 없다.
- (5) 선박의 안정성을 고려하기 위한 컨테이너 무게는 모두 같다.

3.2 재처리 정의

컨테이너선상의 컨테이너들은 선박의 hold와 deck의 아래부터 위순으로 차례로 적재되어진다. 하역시스템구조상 컨테이너에 대한 접근은 오직 상부에서만 할 수 있으며, 특히 적하계획 수립시 선박에서는 아래부터 위로 적재하고 야드는 반대로 위에서부터 아래로 작업을 해야 한다. <Figure 3>은 재처리의 예를 나타내고 있는데, 3번째 항구의 컨테이너를 적재하기 위해서는 그 위에 놓여있는 4번째 항구의 컨테이너 2개를 제거한 후 작업해야하며 이 작업을 재처리(shifting)작업이라 하고 일반적으로 항구 i 에서 적하되어야 할 컨테이너들 위에 i 항구보다 먼 j 항구의 컨테이너가 쌓여있을 경우와 hold에 적재되어야 할 컨테이너가 적재되기 전에 deck에 먼저 적재가 되어있는 경우에 일어난다. 여기서 i, j 는 항구 및 컨테이너 index이다.

Avriel *et al.*(1998)과 Lee(2002)는 재처리가 되는 컨테이너를 재처리 원인에 따라 세 가지로 구분하였다.

- (1) Blocking container: <Figure 3>(a)와 같이 배가 항구 i 에 도착해서 컨테이너 i 를 적하할 때 컨테이너 i 보다 먼 항구 j 의 컨테이너가 컨테이너 i 의 적재위치 상부에 있을 경우에, 항구 j 의 컨테이너를 반드시 제거한 후에야 작업이 가능하다.
- (2) Hatch-blocking container: <Figure 3>(b)와 같이 항구 i 의 적하작업시 hatchcover를 기준으로 hold에 적재할 컨테이너를 작업하기 전에 deck에 적재한 컨테이너가 존재시 이를 제거한 후 hatchcover를 열고 hold에 컨테이너를 적재하는 컨테이너이다.
- (3) Voluntary container: <Figure 3>(c)와 같이 항구 i 의 적하작업이 완료된 후에 다음 항구의 재처리를 고려하여 미리 적재 형태를 조정하기위해 처리되는 컨테이너이다.

그러나, 본 논문에서 재처리 중 voluntary container는 선사에서 발행하는 선적지시서 작성시 고려되고 양하계획과 밀접한 연관이 되어있어 고려하지 않는다.

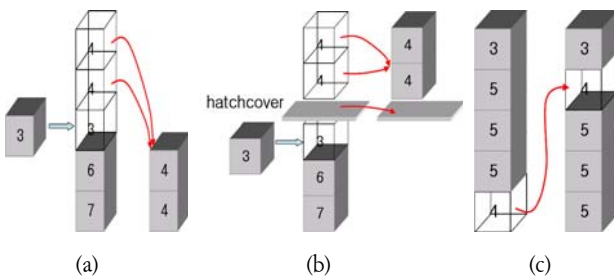


Figure 3. An example of shifting

3.3 Hatch를 고려한 적재순서 최적화모형

본 연구에서 제시하고자 하는 모형은 선박의 한 bay내에 3등분 되어있는 hatchcover가 동일한 stack들을 통합하여 hold와 deck순으로 작업순서를 결정하는 모형이다. 본 모형의 전반적인 절차는 <Figure 4>와 같이 선박과 야드 클러스터를 대상으로 계층적 클러스터링 기법을 적용해 클러스터 재구성을 통하여 클러스터간 관계를 설정하고 재처리수와 TC 총 이동거리를 최소화시키는 적재순서 최적해를 탐색하기 위해 SA 알고리즘을 적용하였다.

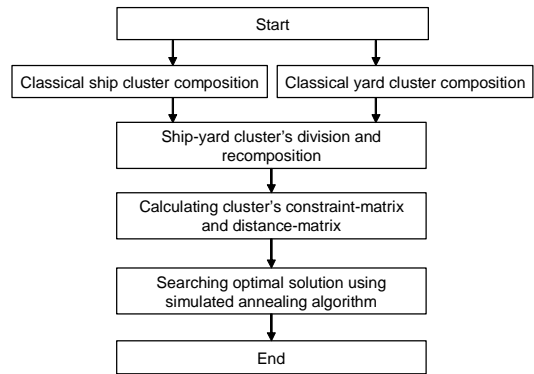


Figure 4. Basic structure of the optimization model

계층적 클러스터링 기법은 모든 개체를 하나의 클러스터로 간주한 후 가장 가까운 클러스터를 병합해 나가는 agglomerative 알고리즘의 단일연결 방법을 적용하여 컨테이너를 집단적으로 처리하기 위해 개개의 컨테이너를 클러스터로 간주하고 식 (1)을 이용하여 클러스터의 위치를 x 좌표값과 y 좌표값으로 정의한 후 위치정보를 이용해 클러스터간 거리가 1인 두 클러스터를 하나의 클러스터로 병합한다.

$$Z_{ij} = \begin{cases} 1, & \text{if } d_{ij} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

s.t. $cx_i = cs_i \times 10$
 $cy_i = ct_i + co_i + cw_i + ch_i$ (선박)
 $cy_i = ct_i + co_i + cw_i + (cbl_i \times 30) + (cba_i \times 8)$ (야드)
 $d_{ij} = \sqrt{(cx_i - cx_j)^2 + (cy_i - cy_j)^2}$
 $i \geq 0, j \geq 0$
 $i, j : \text{integer}$

위의 식 (1)에서 i 와 j 는 각각의 클러스터이며 목적함수인 Z_{ij} 는 기존 연구와 마찬가지로 두 클러스터의 거리값인 d_{ij} 가 1인 경우에만 병합하는 것을 의미한다. 이는 두 클러스터가 선박(야드)내 같은 stack에 위치해 목적지향과 크기, hatch 위치가 같은 클러스터가 상하 인접해 있는 경우에만 병합되도록 한다. 클러스터 i 의 x 좌표값인 cx_i 는 stack단위의 클러스터를 병합하기 위해 클러스터 i 의 선박(야드)내 stack값인 cs_i 에 일정 수

치인 10을 곱하여 정의한 것이다. 즉 클러스터 i 와 j 의 stack번호인 x 좌표값이 틀리면 두 클러스터의 거리값인 d_{ij} 가 1이 될 수 없어 병합될 수 없다.

클러스터 i 의 y 좌표값인 cy_i 는 선박과 야드 클러스터로 구분하였으며, 선박 클러스터에서는 클러스터 i 의 선박내 tier값인 ct_i 에 목적지값인 co_i 와 크기값인 cw_i , hatch 위치값인 ch_i 를 더하여 정의하였으며, 야드 클러스터에서는 클러스터 i 의 선박내 tier값인 ct_i 에 목적지값과 크기값, 클러스터 i 의 야드 위치에 따른 수치값인 cbl_i (block), cba_i (bay)를 더하여 정의하였다.

특히 야드 클러스터인 경우에 적용한 일정 수치 30과 8은 적재될 야드 클러스터의 block-bay번호를 기준으로 거리를 판단한 결과 TC의 작업특성상 block을 이동하는 거리가 같은 block내 bay간 이동하는 거리보다 멀기 때문에 이를 적절한 거리에 따른 수치로 판단한 것이며 block에 따른 거리차이를 30으로, bay에 따른 거리차이를 8로 적용하였다. 클러스터 i 와 j 의 거리값인 d_{ij} 는 기존 연구에서 적용하였던 것과 같이 x 좌표값과 y 좌표값을 바탕으로 거리를 구하는 식이며, 클러스터 i 와 j 의 변수범위는 0보다 큰 정수이다. 따라서 위 수식에 의거 y 좌표는 선박에서 hatch별 동일한 목적지와 크기를 가진 클러스터가 stack을 기준으로 상하 인접해 적재되어 있는 경우에 병합되고, 야드에서는 block-bay별 동일한 목적지와 크기를 가진 클러스터가 stack을 기준으로 상하 인접해 적재된 경우에만 병합된다.

Hatch를 고려한 stack단위의 클러스터 구성과정이 완료되면 선박 및 야드 클러스터의 재구성 과정을 거쳐 병합하게 되는데 재구성은 선박 클러스터와 야드 클러스터의 동질성과 작업 가능 순위에 따라 아래의 규칙에 의거 관계를 정의하고 클러스터내의 컨테이너 개수에 따라 클러스터가 분할된다.

- (1) 선박 클러스터가 hold이면서 목적지항과 컨테이너수가 동일한 경우
 - 작업순서가 높은 야드 클러스터와 병합
- (2) 선박 클러스터가 hold이면서 목적지항은 동일하나 컨테이너수가 틀린 경우
 - 작업순서가 높은 야드 클러스터 선정후 컨테이너 수가 많은 클러스터를 분할
- (3) 선박 클러스터가 deck이면서 목적지항과 컨테이너 수가 동일한 경우
 - 작업순서가 높은 야드 클러스터와 병합
- (4) 선박 클러스터가 deck이면서 목적지항은 동일하나 컨테이너수가 틀린 경우
 - 작업순서가 높은 야드 클러스터 선정후 컨테이너수가 많은 클러스터를 분할

위 규칙에 의거 클러스터 재구성과정은 먼저 모든 선박 클러스터를 대상으로 규칙 (1)과 같이 선박위치가 hold에 속하면서 야드 클러스터와 미 연결된 선박 클러스터를 선별하여 목적지항과 컨테이너 수가 동일한 야드 클러스터를 추출한다. 이때 추출된 야드 클러스터 중 작업가능 우선순위가 가장 빠

른 야드 클러스터를 결정하여 선박 클러스터와 병합하며 결합된 클러스터는 선박 및 야드 클러스터의 정보를 유지하도록 하였다. 위 과정이 종료되면 규칙 (2)에 의해 목적지항은 동일하나 컨테이너 수가 틀린 선박 및 야드 클러스터를 추출하여 작업가능 우선순위가 가장 빠른 야드 클러스터를 결정해 컨테이너 수가 많은 클러스터를 분할한 채로 병합하고 분할되어 남아있는 클러스터는 컨테이너 수가 조정되어 다시 규칙 (1)부터 적용한다.

선박위치가 hold인 선박 클러스터의 재구성과정이 완료되면 선박위치가 deck인 선박 클러스터를 대상으로 규칙 (3), (4)에 의해 앞에서 적용하였던 방법과 동일하게 진행한다. 모든 선박 및 야드 클러스터가 병합되어 재구성과정이 완료되면 최종 클러스터가 생성이 되며, 생성된 최종 클러스터를 대상으로 각각의 거리값과 작업순위에 따른 재처리 패널티값을 판단할 수 있다.

클러스터 재구성간 선박 클러스터가 hold인 것을 먼저 재구성하는 이유는 선박에 컨테이너를 적재시 hold부분의 컨테이너가 먼저 적재되어야 deck부분의 컨테이너를 적재할 수 있으므로 선박 클러스터가 hold인 것에 재구성의 우선권을 부여하여 야드 클러스터의 작업가능 순위가 높은 클러스터와 먼저 병합함으로써 재처리 발생빈도를 줄인 것이다. 또한, 클러스터간 컨테이너 개수가 틀린 경우 컨테이너 개수가 많은 클러스터를 분할하여 재구성함으로써 규칙의 유연성과 작업의 용이성을 향상시켜 다양한 형태의 클러스터가 구성되도록 하였다.

클러스터 재구성이 완료되면 병합된 클러스터간 제약행렬 h_{ij} 와 거리행렬 D_{ij} 를 산출한다. 클러스터간 제약행렬 h_{ij} 는 <Figure 5>와 같이 클러스터간 재처리 발생여부를 행렬로 나타낸 것이며, 선후관계행렬과 비슷하나 재처리 종류에 따라 구체적으로 구분하여 재처리가 미발생시 0으로 표현하고 재처리 중 blocking container가 발생시 1로 표현하며 재처리 중 hatch-blocking container가 발생시 2로 표현한다. 만약 blocking container와 hatch-blocking container가 동시 발생되면 hatch-blocking container의 패널티값이 크므로 2로 표현한다. 그러나 voluntary container는 재처리 정의시 고려하지 않기로 하였으며, 이는 선사에서 발행하는 선적지시서 작성시 이미 고려되고 적화 계획보다는 양하계획과 밀접한 연관이 되어있기 때문이다.

	1	2	3	4	5	...
1	0	1	0	0	0	
2	0	0	2	0	1	
3	0	0	0	1	0	...
4	2	0	0	0	0	
5	0	0	0	0	0	
⋮						

Figure 5. An example of cluster's constraints-matrix

클러스터간 거리행렬 D_{ij} 는 <Figure 6>과 같이 선박의 한 bay내 적재될 야드 클러스터의 block-bay번호를 기준으로 거리

를 판단하며 TC의 작업특성상 block을 이동하는 거리가 같은 block내 bay간 이동하는 거리보다 멀기 때문에 이를 적절한 거리에 따른 수치로 판단하여 block에 따른 거리차이는 30으로, bay에 따른 거리차이는 8로 적용하여 계산하였다.

$$\begin{matrix}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & \dots \end{matrix} \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \end{matrix} & \begin{bmatrix} 0 & 30 & 8 & 38 & 0 \\ 30 & 0 & 0 & 8 & 30 \\ 8 & 0 & 0 & 46 & 16 & \dots \\ 38 & 8 & 46 & 0 & 24 \\ 0 & 30 & 16 & 24 & 0 \\ \vdots & & \vdots & & \end{bmatrix}
 \end{matrix}$$

Figure 6. An example of cluster's distance-matrix

최종 클러스터간 제약행렬과 거리행렬이 결정되면 이를 바탕으로 재처리 횟수와 TC 총 이동거리를 최소화하는 적재순서 최적화 모형을 설계할 수 있으며 식 (2)와 같이 표현할 수 있다.

$$\begin{aligned}
 \text{Min } Z &= \sum_{i=1}^{n-1} (D_{x_i x_{i+1}} + P_{x_i x_{i+1}}) & (2) \\
 \text{s.t } P_{x_i x_{i+1}} &= \begin{cases} p_m, & \text{if } h_{x_i x_{i+1}} = 2 \\ p_n, & \text{if } h_{x_i x_{i+1}} = 1 \\ 0, & \text{otherwise} \end{cases} \\
 D_{x_i x_{i+1}} &\geq 0 \\
 x_i &\geq 0, \quad i = 1, 2, 3, \dots, n \\
 i &\geq 0, \quad j \geq 0 \\
 i, j &: \text{integer}
 \end{aligned}$$

위의 식 (2)에서 i 와 j 는 적재순서 index이며, x_i 는 적재순서 i 번째의 클러스터를 의미하고 적재순서는 $[x_1, \dots, x_n]$ 이다. 목적함수는 적재순서에 따른 총 패널티값과 거리값의 합을 최소화 하는 식으로서 재처리 횟수와 TC 총 이동거리를 관리한다. 제약식은 목적함수에 사용될 거리행렬과 제약행렬의 정의와 범위를 설정해 주는 것으로, $P_{x_i x_{i+1}}$ 는 적재순서에 따른 재처리 발생시 재처리 종류에 따른 패널티값으로 재처리 중 blocking container가 발생되면 패널티값 p_m 을, hatch- blocking container가 발생되면 패널티값 p_n 을 정의하는 식이다. 여기서 p_m 은 hatch-cover를 제거한 후 작업을 함으로써 많은 시간과 비용이 소요되는 패널티값으로 같은 stack내에서 위에 있는 컨테이너를 제거해 작업을 하는 패널티값 p_n 보다 커야하며 차등 적용해야 한다. 적재순서에 따른 클러스터간 거리값인 $D_{x_i x_{i+1}}$ 는 0보다 크며, 적재순서 index인 i 와 j 는 0보다 큰 정수값으로 각 변수들의 범위를 나타낸다.

본 연구의 적재순서 최적화모형은 모형내에 hatch 위치별로 구분하고 클러스터 분할 및 재구성을 통하여 구성되었기 때문에 다양한 상황변화에 따라 적절하게 문제를 재구성할 수 있다. 따라서 본 모형은 선박이 접안한 후 수출 컨테이너 적하에 따른 적재순서문제뿐만 아니라 장비고장이나 우발상황 등의

환경에서 신속한 의사결정을 필요로 하는 극단적인 적재순서 문제에도 그 활용도가 높을 것으로 판단된다. 그리고 이를 위해서는 현실적인 모형구성과 더불어 이 모형에 대한 효율적인 해 유도과정이 뒷받침되어야만 할 것이다.

다음 장에서는 본 연구에서 제안한 적재순서에 대한 비선형 수리모형의 결과를 신속하고 정확하게 계산해낼 수 있는 새로운 해법절차를 제시하고자 한다.

4. SA 알고리즘을 적용한 해 유도과정

4.1 SA(Simulated Annealing) 알고리즘

본 논문에서는 hatch를 고려한 적재순서 최적화모형에 대한 해법으로 그 구조가 단순하면서도 전역탐색능력이 우수한 SA 알고리즘을 적용하여 기본적인 해법알고리즘을 구성하고 해 공간 탐색시간이 오래 걸리는 단점을 보완하기 위하여 변형된 방법인 ASA 알고리즘 적용을 제안하였다.

SA 알고리즘은 조합 최적화문제의 한 해법으로서 Kirkpatrick *et al.*(1983)이 제안하였으며, 기존의 반복적 개선에 근거한 휴리스틱기법들이 국부최적해에 빠져버리는 단점을 보완한 범용의 최적화 기법으로 많은 분야에서 응용되고 있다. 또한, SA의 변형된 방법으로 ASA 알고리즘은 SA의 기본형태를 유지하여 수렴성에 관한 이론적인 조건을 되도록 위반하지 않으면서, 최적에 가까운 해로의 수렴시간을 단축해 주고 사용자가 미리 설정해 주어야 하는 파라미터의 수를 되도록 줄여주어 알고리즘의 간편성을 유지한 방법이다(Kim *et al.*, 2000).

4.2 해 표현 및 초기해 생성 알고리즘

본 연구에서 해는 <Figure 7>과 같이 클러스터 적재순서 리스트로 표현하며, 이는 SA 알고리즘을 적용하기 위해 수출 컨테이너의 적하계획 문제의 특성을 고려하여 해로 표현한 것이다. 클러스터 재구성이 완료되어 최종 클러스터가 결정되면 이들 간의 순서를 결정해야 하는데 해 표현인 $X[C]$ 는 결정된 클러스터의 적재순서를 차례로 배치한 것으로 표현의 용이성과 적재순서에 따른 패널티값을 적용할 수 있도록 구성한 것이다.

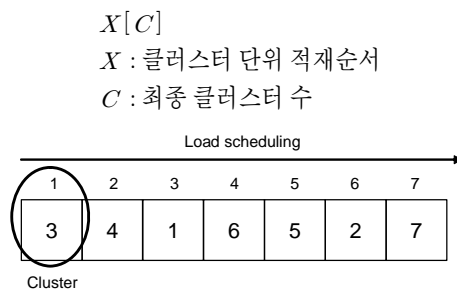


Figure 7. Solution expression

해에서 클러스터 단위로 나열된 순서는 최종적으로 TC 및 CC의 작업순서인 동시에 선박에 컨테이너가 장치되는 순서이다. 여기서 최종 클러스터는 적하게 계획 특성상 생성된 선박 및 야드 클러스터를 바탕으로 최적해에 가깝게 사전 재구성 과정을 통하여 서로 짝이 지어진 클러스터이다. 또한, 이러한 해 표현은 각각의 클러스터가 장비에 의해 작업이 한 번만 이루어지므로 중복방문이 없는 TSP문제와 유사 형태로 볼 수 있다.

SA 알고리즘을 이용하여 전역최적해로 빨리 수렴하기 위해서는 좋은 초기해를 갖는 것이 중요하다. 특히 다양한 초기해를 가지고 있어야 해의 범위를 넓게 조사할 수 있다. 본 논문에서는 초기해를 생성하는데 있어서 재처리 중 hatch-blocking container가 발생되지 않는 초기해를 생성한다. 이를 위하여 초기해 생성시 hold에 위치한 모든 클러스터를 적재한 후 deck에 있는 클러스터를 적재하도록 순서를 부여하였다.

4.3 이웃해 생성 알고리즘

계산시간이 빠른 ASA는 내부루프 L 을 최소한으로 줄여줌으로써 수렴시간을 상당히 단축시키고 수렴성능 또한 향상시킬 수 있다. 하지만 순서조합문제에 대한 전체 계산시간은 알고리즘의 수렴성능뿐만 아니라 이웃해 생성과정($Y = PERTURB(X)$)에 의해서도 많은 영향을 받는다.

본 논문에서는 이웃해를 생성하기 위해 Lin and Kernighan (1973)이 제시한 TSP문제에서의 2-전이(transition)방법을 발전시켜 다양한 가능해를 얻기 위해 3-전이방법을 사용하였다. 3-전이방법은 <Figure 8>과 같이 임의의 3개 적재순서 위치를 랜덤하게 선택하여 선택된 클러스터를 서로 바꿔주어 다양한 가능해를 얻는다. 근방의 크기는 $n(n-1)(n-2)/2$ 가 되고, 해의 변화되는 값은 <Figure 8>(b)를 예로하면 식 (3)과 같이 간단하게 계산될 수 있다.

$$\begin{aligned} \Delta f = & -D_{\pi^{-1}(p)p} - P_{\pi^{-1}(p)p} - D_{p\pi(p)} - P_{p\pi(p)} \\ & - D_{\pi^{-1}(q)q} - P_{\pi^{-1}(q)q} - D_{q\pi(q)} - P_{q\pi(q)} \\ & - D_{\pi^{-1}(r)r} - P_{\pi^{-1}(r)r} - D_{r\pi(r)} - P_{r\pi(r)} \\ & + D_{\pi^{-1}(p)q} + P_{\pi^{-1}(p)q} + D_{q\pi(p)} + P_{q\pi(p)} \\ & + D_{\pi^{-1}(q)r} + P_{\pi^{-1}(q)r} + D_{r\pi(q)} + P_{r\pi(q)} \\ & + D_{\pi^{-1}(r)p} + P_{\pi^{-1}(r)p} + D_{p\pi(r)} + P_{p\pi(r)} \end{aligned} \quad (3)$$

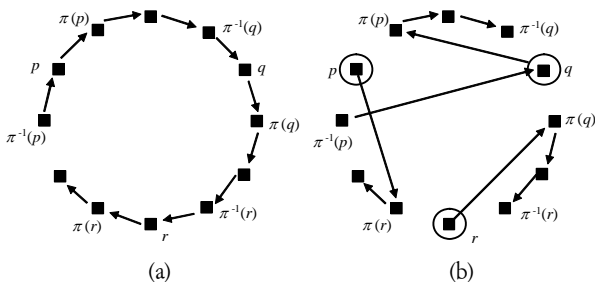


Figure 8. 3-transition method

5. 실험 및 결과분석

5.1 실험계획

본 연구에서는 최적해를 구하기 위하여 VC++로 구현을 하고 컴퓨터는 Pentium-IV CPU 1.80 GHz, RAM : 256M을 사용하였다.

실험 데이터는 컨테이너 개수에 따라 축소실험은 Lee(2004), 확대실험은 Kim et al.(1997)에서 사용한 데이터를 사용하였으나 전부 hatch를 미 고려하고 선박에서의 bay단위 결정이나 야드 위치에 따른 거리판단을 적용하지 않아 본 논문과 객관적인 비교가 될 수 없어 축소실험은 상황에 맞게 수정하여 결과를 비교하였으며 확대실험은 참고로 결과만을 제시하였다. 또한 이 문제가 수학적으로 최적해를 구할 수 없는 NP-hard 문제로 실험결과 및 분석은 적재순서에 따른 총 패널티값과 계산 시간만을 표시하였다.

컨테이너 개수는 축소실험에서 18개를, 확대실험에서 30, 50, 70, 100, 128, 300, 500, 1,000개 까지 검사를 실시하며 초기 온도, 냉각률, 현재해 변경비율, 반복수를 변화시켜 최적의 값을 도출한다. 실험에 사용한 상수값은 각 작업에 대한 시간을 고려하여 재처리 패널티값인 p_m 은 200, p_n 은 50을 부여한다. 초기온도는 전역 최적해의 수렴에 큰 영향을 주므로 적절한 초기온도를 구하기 위하여 50°, 70°, 100°, 130° 로 다르게 적용하고 다른 조건을 동일하게 적용해 실험한 결과, <Table 1>의 결과값과 같이 100°일 때 가장 좋은 근사 최적해를 얻어 초기 온도를 100°로 설정하였다.

Table 1. Parameter table(Initial temperature)

Num	50°		70°		100°		130°	
	P	Time	P	Time	P	Time	P	Time
18	88	00:06	88	00:10	87	00:13	87	00:16
30	143	00:16	141	00:19	141	00:24	141	00:28
50	380	00:54	374	00:58	372	01:03	372	01:11
100	552	02:02	548	02:09	547	02:21	547	02:32
128	641	02:14	640	02:32	635	02:48	635	03:03

Table 2. Parameter table(Changing rate)

Num	2%		5%		8%		10%	
	P	Time	P	Time	P	Time	P	Time
18	87	00:13	87	00:13	87	00:15	87	00:15
30	142	00:24	141	00:24	141	00:25	141	00:25
50	372	01:03	372	01:03	372	01:04	372	01:05
100	548	02:21	547	02:21	547	02:24	547	02:24
128	637	02:48	635	02:48	635	02:50	635	02:51

적당한 변화로 이웃해를 생성하기 위한 현재해의 변경비율은 2%, 5%, 8%, 10%로 주고 초기온도를 100°로 설정하여 실험

한 결과, <Table 2>의 결과값과 같이 근사 최적해의 수렴값에는 별로 영향을 끼치지 않았고 해 수렴 시간면에서 다소 좋은 결과를 보인 5%를 현재해의 변경비율로 설정하였다.

기타 요소로 냉각률은 0.95, 평형상태에 도달하기 위한 반복수는 10,000개로 설정하였으며, 종료조건으로 SA 알고리즘은 온도가 충분히 낮을때 종료되도록 하고 ASA 알고리즘은 200번의 외부루프를 반복하는 동안 목적함수의 변화가 없거나 내부루프 안에서 최소점을 갱신하는 간격이 50번을 넘으면 종료하도록 하였다.

5.2 축소실험 및 결과분석

축소실험 데이터는 <Figure 9>와 같이 Lee(2004)에서 이용한 데이터를 바탕으로 18개의 컨테이너를 사용하여 선박에서는 hatch를 고려하기 위해 hatchcover를 추가로 설정하고 야드에서는 야드 위치에 따른 거리를 판단하기 위해 block-bay-stack 단위로 구분하였다.

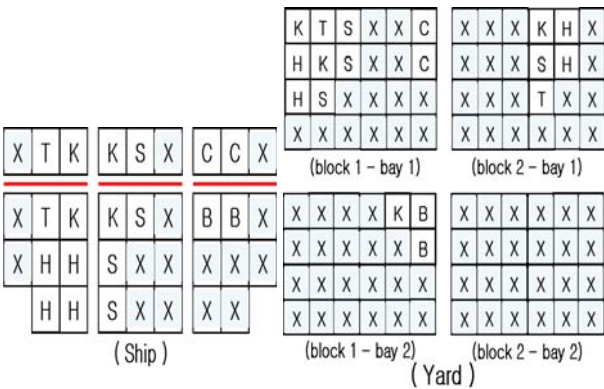


Figure 9. Arrangement situation of the small-scale test

본 논문에서 제안한 모형을 바탕으로 실험한 결과 선박과 야드의 클러스터를 도식하면 <Figure 10>과 같고, <Table 3>은 선박 클러스터로 구성된 작업순서 결과이며 <Table 4>는 야드 클러스터로 구성된 작업순서 결과이다.

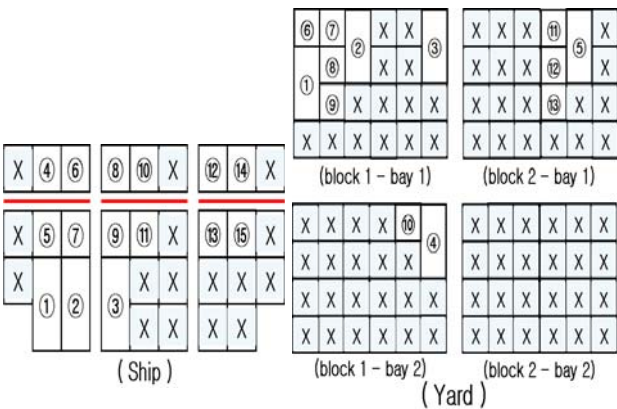


Figure 10. Ship-yard cluster's composition

Table 3. Load sequence's table of ship cluster

Num	POD	Container	Hatch	Stack	Rank
1	H	2	hold	2	1
2	H	2	hold	3	1
3	S	2	hold	4	1
4	T	1	deck	2	3
5	T	1	hold	2	2
6	K	1	deck	3	3
7	K	1	hold	3	2
8	K	1	deck	4	3
9	K	1	hold	4	2
10	S	1	deck	5	2
11	S	1	hold	5	1
12	C	1	deck	7	2
13	B	1	hold	7	1
14	C	1	deck	8	2
15	B	1	hold	8	1

Table 4. Load sequence's table of yard cluster

Num	POD	Container	Block	Bay	Stack	Rank
1	H	2	1	1	1	2
2	S	2	1	1	3	1
3	C	2	1	1	6	1
4	B	2	1	2	6	1
5	H	2	2	1	5	1
5	K	1	1	1	1	1
6	T	1	1	1	2	1
7	K	1	1	1	2	2
8	S	1	1	1	2	3
10	K	1	1	2	5	1
11	K	1	2	1	4	1
12	S	1	2	1	4	2
13	T	1	2	1	4	3

<Table 3>과 <Table 4>에서 나온 선박 클러스터 15개와 야드 클러스터 13개를 바탕으로 클러스터 재구성을 실시한 결과는 <Table 5>와 같으며 총 15개의 최종 클러스터로 구성된다.

<Table 5>의 최종 클러스터를 바탕으로 TC 총 이동거리와 재처리수를 최소로 하는 적재순서를 결정하기 위해 적재순서 최적화모형을 바탕으로 SA 및 ASA 알고리즘을 적용한 모델 결과값과 GA를 적용한 결과값을 비교분석한 결과는 <Table 6>과 같다.

본 논문에서 적용한 SA, ASA 알고리즘과 GA 알고리즘의 결과값을 비교하면 <Table 6>과 같이 목적함수값은 차이가 없으나, 계산시간면에서 ASA 알고리즘이 SA나 GA 알고리즘보다 빠른 속도로 문제를 해결할 수 있어 우수한 성능을 보였다.

축소실험 결과로 최적 적재순서는 5 → 2 → 1 → 3 → 4 → 11 → 7 → 8 → 10 → 6 → 9 → 15 → 12 → 14 → 13의 클러스터 순이며, 각 클러스터 내의 컨테이너 순서는 선박에서는 아래에 있는 컨테이너가 우선이며, 야드에서는 위에 있는 컨테이너가 우선적으로 작업을 실시하여 차례로 진행된다. 최적 적재순서에 따른 TC 이동거리와 재처리 패널티값의 합은 87로 나왔으며, 이는 GA와 SA 알고리즘에서 공통적으로 나타난 결과이다.

Table 5. Load sequence's table of final cluster

Num	POD	Container	Ship	Hatch	Yard	Rank
1	H	2	1	hold	5	1
2	H	2	2	hold	1	1
3	S	2	3	hold	2	1
4	T	1	5	hold	7	2
5	K	1	7	hold	6	2
6	K	1	9	hold	10	2
7	S	1	11	hold	12	1
8	B	1	13	hold	4	1
9	B	1	15	hold	4	1
10	T	1	4	deck	13	3
11	K	1	6	deck	11	3
12	K	1	8	deck	8	3
13	S	1	10	deck	9	2
14	C	1	12	deck	3	2
15	C	1	14	deck	3	2

Table 6. Comparison of the results(small-scale test)

Num	GA(Lee)		SA		ASA	
	Penalty	Time	Penalty	Time	Penalty	Time
18	87	01:35	87	00:17	87	00:13

다음절에서는 본 연구에서 적용한 해법이 기존 해법들에 비해 얼마나 향상될 수 있는가를 측정하기 위해 다양한 확대실험을 통해 그 결과를 비교분석하였다.

5.3 확대실험 및 결과분석

확대실험에서는 SA기법을 적용한 해 유도과정이 대규모의 적재순서 결정문제에서도 우수한 해법임을 입증하기 위하여 본 연구에서 개발한 적재순서 최적화모형을 이용하여 현실적인 선박 규모의 문제를 구성하고 이에 대한 확대실험을 실시한다. 비교 대상은 축소실험과 달리 Kim *et al.*(1997)의 모델에서 나온 결과값을 이용하였으나 hatch를 미 고려하는 등 적용한 상황이 틀려 패널티값은 무시한 채 컨테이너 수에 따른 계산 시간만을 SA와 ASA의 결과들과 비교분석하였다.

각 실험의 규모는 컨테이너 수를 30, 50, 70, 100, 128, 300,

500, 1,000개로 증가시켜 구성하였고 객관적인 비교분석이 될 수 있도록 동일한 문제를 같은 조건에서 10번씩 수행하였으며 같은 방식으로 점차 문제의 규모를 확대하였다. 각 문제별로 초기온도, 냉각스케줄, 내부루프는 축소실험에 적용한 값과 동일하게 적용하였으며 이에 따른 결과값은 크게 변동이 없었다. 특히, 초기온도는 축소실험 결과에서 얻은 목적함수값에 근거하여 SA의 수렴성능이 보장될 수 있도록 적당한 T 값인 100° 를 지정해주었다.

Table 7. Comparison of the results(large-scale test)

Num	GA(Kim <i>et al.</i>)		SA		ASA	
	Penalty	Time	Penalty	Time	Penalty	Time
30	85	03:22	141	00:30	141	00:24
50	-	-	377	01:09	372	01:03
70	-	-	490	01:23	490	01:20
100	-	-	549	02:32	547	02:21
128	1515	25:15	637	02:58	635	02:48
300	-	-	1416	06:54	1410	06:21
500	-	-	2320	15:11	2316	13:58
1,000	-	-	8284	38:23	8272	30:42

<Table 7>에서 보는 바와 같이, 목적함수값은 컨테이너 개수가 30개 이하일 때는 SA와 ASA 알고리즘이 차이가 없었으나 컨테이너 개수가 30개 이상으로 증가함에 따라 ASA 알고리즘이 보다 좋은 해를 탐색하였다. 계산시간면에서는 SA 및 ASA 알고리즘이 3분 이내의 빠른 속도로 모든 문제를 해결할 수 있었으나 기존 연구에서 제시되었던 GA 알고리즘은 대체적으로 시간이 오래 걸리며 100개 이상부터는 20분 이상이 소요되었다. 특히, ASA 알고리즘이 컨테이너 수가 증가함에 따라 목적함수값과 계산시간의 모든 면에서 우월하다는 것을 알 수 있었으며 컨테이너 개수가 증가할수록 이러한 현상이 뚜렷하게 나타남을 알 수 있었다. 이러한 ASA 알고리즘의 특성은 터미널에서 컨테이너 처리량이 점점 증가하고 있는 추세에 비추어 작업처리 속도를 향상시킬 수 있으며 처리비용을 감소하는데 큰 효용성이 있다고 할 수 있다.

6. 결론 및 향후 연구방향

본 연구는 컨테이너 터미널에서의 최적 적재순서 결정문제를 해결하기 위해 기존 연구에서 고려되지 않았던 hatch를 고려하여 hold와 deck를 적재순서 결정시 중요한 요소로 포함하였으며 이에 따라 stack에 따른 계층적 클러스터링 기법을 통하여 선박 클러스터와 야드 클러스터를 결정한 후 클러스터 재구성 과정을 통하여 최종 클러스터를 구성하였다. 최종 클러스터를 바탕으로 각 클러스터간 상호관계를 정의하고 대규모 적재순서 결정문제를 효과적으로 계산해내기 위해 범용의 SA 알고리

즘을 도입하여 해 유도과정으로 구성하였다.

특히, 대부분의 기존 연구에서 제외되었던 hatch를 3등분하여 고려한 것은 터미널의 현실적인 환경에 더욱 근접한 방법이며, 해 탐색법으로 적하계획에 한 번도 사용되지 않았던 SA 알고리즘을 도입하여 적용가능성을 확인하였고 전역 최적해에 수렴하는 시간이 너무 길어지는 SA 알고리즘의 단점을 보완하기 위해 변형방법인 ASA 알고리즘을 적용하여 최적에 가까운 해로의 수렴시간을 단축하였다.

하지만, 본 연구에서는 적재순서 결정문제에 SA 알고리즘의 적용가능성까지만 확인하였고 문제규모에 따른 향상도 및 상관관계는 유도하지 않아 이 분야에 대한 세밀한 연구가 필요하며, SA를 적용함에 있어 개별적으로 지정해 주어야 하는 파라미터들에 대해서도 보다 체계적인 연구 분석을 통해 문제 규모별 알고리즘 운용방안 역시 적립해 볼 필요가 있다.

그리고, SA 알고리즘의 효율성을 향상시키기 위한 방안인 다른 기법들과의 결합으로 하이브리드(hybrid) 탐색방법 적용도 고려할 수 있다. 다음의 연구로 적재우치를 결정하는 시스템과 통합한 모형을 설계하고, 나아가 전체 계획을 효율적으로 수립할 수 있는 시스템의 개발로 진행되어야 할 것이다.

참고문헌

- Akio, Imai, Kazuya, Sasaki, Etsuko, Nishimura, and Stratos, Papadimitriou (2006), Multi-objective Simultaneous Stowage and Load Planning for a Container Ship with Container Rehandle in Yard Stacks, *European Journal of Operational Research*, 171, 373-389.
- Avriel, M., Penn, M., Shpirer, N., and Witteboon, S.(1998), Stowage Planning for Container Ships to Reduce the Number of Shifts, *Annals of Operations Research*, 76, 55-71.
- Botter, R. C. and Brinati, M. A.(1992), Stowage Container Planning: A Model for Getting An Optimal Solution, *IFIP Trans. B(App. in Tech.)*, B-5, 217-229.
- Cho, D. W.(1982), Development of a Methodology for Containership Load Planning, *Master Thesis, Oregon State University*.
- Jang, Y. J., Jang, S. Y., Yang, C. H., and Park, J. W. (2002), Study on the Resource Allocation Planning of Container Terminal, *Journal of the Korean Institute of Industrial Engineers*, 28(1), 14-24.
- Kang, J. G. and Kim, Y. D.(2000), Stowage Planning in Maritime Container Transportation, *Master Thesis, KAIST*.
- Kang, K. J. and Lee, C. Y.(1990), A Study on the Stowage Planning of Container Ship, *International Journal of Navigation and Port Research*, 14(4), 1-15.
- Kim, K. H., Kim, K. Y., and Ko, C. S.(1997), Load Scheduling Using a Genetic Algorithm in Port Container Terminals, *Journal of the Korean Institute of Industrial Engineers*, 23(4), 645-660.
- Kim, K. H., Ryu, K. R., Park, Y. M., Kang, J. S., and Lee, Y. H.(2000), A Two-stage Meta-heuristic Algorithm for Container Load Sequencing, *Proceeding of Korean Institute of Industrial Engineers / Korean Operations Research and Management Science Society 2000*, 9-12.
- Kim, Y. G., Yeun, B. S., and Lee, S. B.(2000), *Meta-heuristic*, Yeongji Moonhwasa, Seoul, Korea.
- Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M. P.(1983), Optimization by Simulated Annealing, *Science*, 220, 671-680.
- Lee, J. Y.(2004), A Study on the Optimization of Load Planning Using Genetic Algorithm, *Master Thesis, Kyungbee University*.
- Lin, S. and Kernighan, B. W.(1973), An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Oper. Res.*, 21, 498-516.
- Lee, S. W., Choi, H. R., Park, N. K., Kim, H. S., Park, B. J., and Noh, J. H.(2002), Container Stowage. Planning Using Genetic Algorithm, *The Korea Association for Creation Research*, 35(2).
- Ryu, K. R., Kim, K. H., Lee, Y. H., and Park, Y. M. (2001), Load Sequencing Algorithms for Container Ships by Using Metaheuristics, *ICPR-16*.
- Wilson, I. D. and Roach, P. A.(1999), Principles of Combinatorial Optimization Applied to Container-Ship Stowage Planning, *Journal of Heuristics*, 14, 403-418.
- Wilson, I. D., Roach, P. A., and Ware, J. A.(2001), Container Stowage Pre-planning: Using Search to Generate Solutions, *Knowledge-Based Systems*, 14, 187-195.



이상현

육군사관학교 전자공학과 학사
Naval Postgraduate School OR 석사
Georgia Institute of Technology 산업공학 박사
현재: 국방대학교 운영분석학과 교수
관심분야: 네트워크, 메타 휴리스틱, 로지스틱스, SCM, 시뮬레이션



김문규

육군사관학교 토목공학과 학사
국방대학교 OR 석사
현재: 육군항공학교 전술학처 교관
관심분야: 메타 휴리스틱, 시뮬레이션



안태호

고려대학교 독문학 학사
Penn. State Univ. MBA 석사
University of Florida 경영과학(OR) 박사
현재: 숭실대학교 경영학부 교수
관심분야: Scheduling, 시뮬레이션, 리스크 분석