

Meta-heuristic 기법을 이용한 2단계 컨테이너 적하계획 알고리즘 A Two-stage Meta-heuristic Algorithm for Container Load Sequencing*

김갑환¹ · 류광렬² · 박영만¹ · 강진수¹ · 이용환²

¹부산대학교 산업공학과 / ²부산대학교 컴퓨터공학과

Abstract

컨테이너 터미널에서 효율적인 적하작업 계획을 자동으로 생성하는 알고리즘을 연구하였다. 실제 터미널에서 계획자들이 적하작업 계획서에 고려하는 제약조건 및 효율적인 계획을 위한 고려사항을 조사하였다. 이를 바탕으로 1단계에서는 개미시스템(ant system)이라는 인공지능기법을 적용하여 제약조건을 만족시키면서 원활한 적하작업이 진행될 수 있도록 컨테이너 크레인과 트랜스퍼 크레인의 이동순서와 위치를 결정하고, 2단계에서는 1단계에서의 결과를 바탕으로 빔탐색법(beam search)을 사용하여 컨테이너 개개의 작업순서를 결정하는 알고리즘을 개발하였다. 또한 개발된 시스템의 성능을 검증하기 위하여 최근의 대형선박에 대한 실제 현장자료를 바탕으로 실험을 수행하였다.

1. 서론

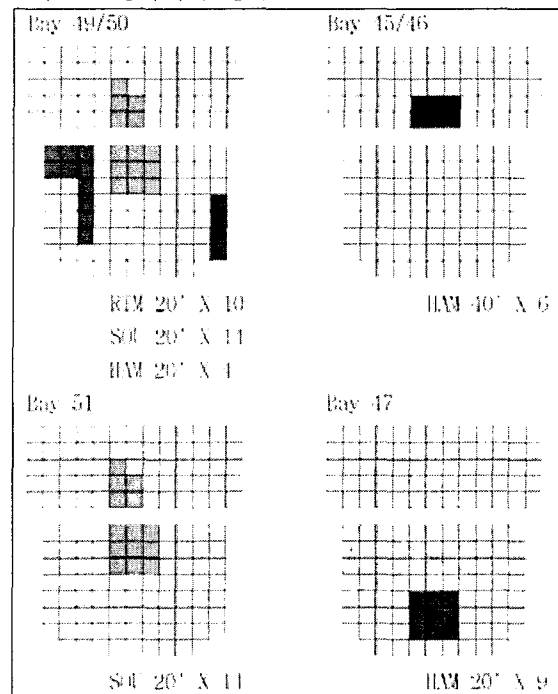
컨테이너선이 점차 대형화되어가고 선사의 빠른 서비스 요구에 대응해 컨테이너 터미널에서는 생산성을 증대하여 경쟁력을 기를 수 밖에 없으며, 컨테이너 터미널의 생산성은 양/적하작업의 효율성에 좌우된다. 양하작업은 컨테이너 터미널에서 선박에 실려져 있는 컨테이너를 하역하는 작업이며, 적하작업은 이와는 반대로 장치장의 컨테이너를 선박에 적재하는 작업이다.

양/적하작업 중에서 적하작업이 양하작업보다 중요시 되는데, 이는 적하작업이 양하작업보다 선박과 장치장(야드) 양측에 대해 고려사항과 예외사항이 다양하기 때문이며, 또한 전체작업시간 중에서 많은 부분을 차지하기 때문이다. 본 논문에서는 컨테이너 터미널의 생산성을 향상시키고 수동으로 작성되는 적하계획을 자동화하도록 하는 알고리즘의 개발에 대해 연구하였다.

일반적으로 컨테이너 터미널의 구조는 컨테이너선이 접안하는 선석과 선박에 컨테이너를 싣고 내리는 작업을 담당하는 컨테이너 크레인(Container Crane; CC), 야드에서 블록을 따라 이동하며 컨테이너를 처리하는 트랜스퍼 크레인(Transfer Crane; TC), 그리고 이 두 설비의 사이에서 컨테이너를 옮기는 역할을 담당하는 야드 트럭(Yard Truck; YT)이 있다. 선박이 선석에 접안하게 되면 먼저 CC는 양하 컨테이너를 내려 이를 YT에 전달하며, YT는 컨테이너를 지정된 장치위치로 옮기며, 해당위치에서는 TC가 이것을 받아서 장치한다. 반대로 적하작업시에는 야드에 장치된 컨테이너를 TC가 YT에 실어주면 YT는 CC로 옮기고, CC는 지정된 선박의 위치(ship-cell)에 적재하게 된다.

본 연구의 대상은 적하계획을 수립하는 과정으로서 적하계획은 선사에서 지정한 선박의 적하위치에 장치장의 어떠한 컨테이너를 싣게될지 결정하는 것으로 일반적으로 선사에서는 <그림 1>과 같은

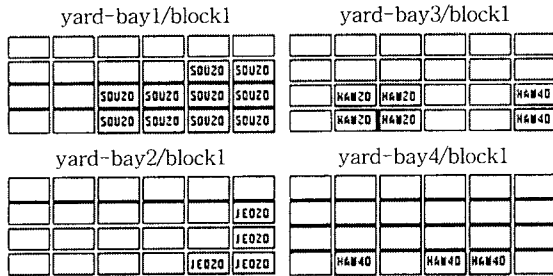
선적지시서(stowage instruction)라는 것을 작성하여 컨테이너 터미널로 전달하는데 여기에는 선박내에 어떠한 위치에 어떤 목적항과 종류, 크기를 가진 컨테이너를 실을 것인가하는 지시가 되어있다. 그러나 이것은 개략적인 지시만 되어있을뿐 개개의 컨테이너가 지정되어 있지는 않다. <그림 2>의 야드맵과 같이 실제 야드의 컨테이너는 동일한 목적항, 종류, 크기를 가진 것이 여러 곳에 여러 개가 존재하므로 현장작업을 위해서는 각각의 컨테이너의 선적위치를 지정하여야 하며, 또한 각각의 작업순서를 결정하여야 한다.



<그림 1> 선적지시서의 예

* 본 연구는 한국과학재단 지정 동아대학교 지능형통합항만관리연구센터의 지원에 의한 것입니다.

적하계획은 선적지시서와 <그림 3>과 같은 CC의 작업스케줄, 야드에서의 컨테이너위치와 정보, CC와 TC의 작업편의성등을 고려하여 작성하게 된다. 적하계획의 최종 결과는 선박의 적하위치를 야드에 장치되어 있는 특정 컨테이너와 1대1로 연결시키고 선박의 적하위치간의 적하순서를 결정하는 것이다. 참고로 그 최종 결과는 <그림 7>과 같다.



<그림 2> 야드맵의 예

작업 순서	위치	컨테이너 크기	컨테이너 개수	작업시간
1	bay 45/46 deck	40"	6	9:00
2	bay 47 hold	20"	9	
3	bay 49/50 hold	20"	28	
4	bay 51 hold	20"	9	
5	bay 51 deck	20"	5	
6	bay 53 deck	20"	7	
7	bay 55/56 deck	20"	6	11:50

<그림 3> CC 작업스케줄의 예

2 자동화 알고리즘

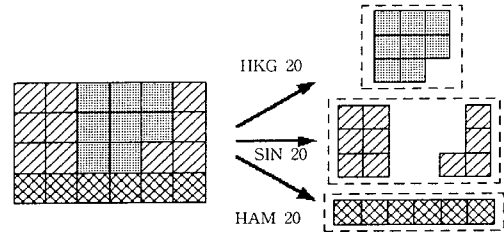
자동 적하 알고리즘을 크게 두 단계로 구분하여 개발하였다. 첫 번째 모듈은 선박과 야드의 컨테이너를 cluster(동일조건인 컨테이너 그룹)로 구분하고 개미시스템을 적용하여 cluster들을 연결하는 단계이고, 두 번째 단계는 첫 번째 단계의 결과를 사용하여 cluster내의 개개의 컨테이너들에 대한 작업순서를 결정하는 단계로 구성된다.

2.1 선박/야드 cluster 생성

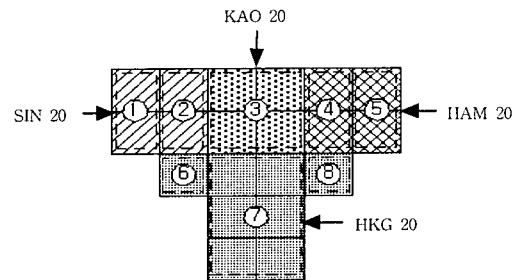
알고리즘을 사용하여 적하계획을 작성하기 이전에 알고리즘에 사용할 자료구조인 선박과 야드의 cluster를 생성한다. 먼저 야드 cluster는 다음과 같은 방법으로 생성한다. 야드상에서 동일 블럭, 배이에 존재하면서 목적지향, 컨테이너 크기, 컨테이너 유형이 동일한 컨테이너들을 하나의 cluster로 만드는 데, 예는 <그림 4>와 같다.

선박 cluster도 야드 cluster와 유사한 방법으로 생성한다. 선박에서의 작업 순서는 CC 작업스케줄에 의해서 그룹까지의 작업 순서는 결정되어진다. 따라서 선박에서는 같은 그룹내에 존재하고 인접하면서 목적지, 크기, 유형이 같은 컨테이너들을 하나의 cluster로 만든다. 그리고 각각의 cluster의 생성시에 CC 작업의 유연성을 위하여 cluster 사이의 작업의 선후 관계가 분명하도록 만든다. 예는 <그림 5>

과 같다. <그림 5>을 보면 cluster ①, ②는 모든 조건이 동일하므로 하나의 cluster로 만들어질 수 있다. 하지만 본 연구에서는 작업의 유연성을 위해 cluster를 구분하였다. 만약 ①, ②가 별개의 cluster로 만들어진다면 ①의 경우는 ⑥과는 아무런 관계 없이 작업이 처리되고 ②는 ⑥의 작업이 이루어진 후에 수행되어야 한다. 그러나 하나의 cluster로 만들어진다면 ⑥이 먼저 작업이 끝나기 전까지는 ①과 ② 모두 작업을 할 수 없다. 따라서 유연성을 높이기 위해서 같은 목적지향, 크기, 유형의 컨테이너가 서로 인접해 있더라도 다른 cluster로 만든다. 그리고 cluster가 만들어지면 그들 사이의 선후관계를 나타내는 선후관계 행렬(Ordering Matrix)을 만들어 cluster간의 작업순서 결정시 참고하도록 한다.



<그림 4> 야드cluster 생성



<그림 5> 선박cluster 생성

2.2 개미시스템

개미시스템[1]은 Colorni, Dorigo, Maniezzo 등에 의해 1991년 travelling salesman problem, quadratic assignment problems 등에 적용하기 위해 만들어진 방법이다[2]. 개미시스템의 기본적인 착상은 개개의 개미는 혼자서 먹이가 있는 곳까지의 최소 경로를 찾아가지 못한다. 그러나 개미는 자신이 지나가는 길에 페로몬을 분비함으로써 개미들의 군집은 페로몬에 의해 최소 경로를 찾아가는 것이다. 즉, 처음에는 모든 개미들은 임의의 경로로 가지지만 시간이 지남에 따라 짧은 경로에는 페로몬이 더 많이 쌓이게 되므로 서서히 짧은 경로로 가는 개미의 수가 많아지고 결국 모든 개미는 최소 경로를 찾아가게 되는 것이다.

마찬가지로 개미시스템은 현재 위치에서 다른 위치로 이동을 할 때 페로몬의 양에 의해 확률적으로 이동하게 된다. 그러나 개미시스템의 개미는 자연의 개미와는 달리 페로몬의 양에만 의존하는 것이 아니라, 두 지점 사이의 거리에 대한 정보도 같이 이용한다.

개미시스템에서 경로를 결정하기 위해 필요한 요소를 보면 다음과 같다.

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in \Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} & \text{if } j \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

이때 $\eta_{ij} = \frac{1}{d_{ij}}$ 이며

- τ_{ij} :노드*i*와 노드*j* 사이의 페로몬의 양
- α : τ_{ij} 의 영향의 정도를 나타내는 매개변수
- η_{ij} :노드*i*에서의 노드*j*에 대한 가시도
- β : η_{ij} 의 영향의 정도를 나타내는 매개변수
- Ω :아직 방문 하지 않은 곳들의 집합
- d_{ij} :노드*i*에서 노드*j*까지의 거리

p_{ij} 는 *i*에서 *j*로 이동할 확률이다. 따라서 개미 시스템에서 노드를 방문할 때 노드를 방문할 확률은 페로몬의 양이 많을수록 가시도가 높을수록(거리가 가까울수록) 높아지게 된다.

위의 방법으로 개미의 군집이 모든 경로를 지나가게 되면 개미가 지나간 경로의 노드들 사이에 특정한 양의 페로몬이 분비되도록 한다. 그러나 페로몬의 양은 모든 개미가 지나간 경로에 같은 양이 분비되는 것이 아니라 각각의 개미가 지나간 경로의 총 거리에 반비례해서 분비되게 된다. 그리고 페로몬의 양은 시간의 흐름에 따라 일정 비율로 감소(회발)되게 된다. 따라서 매 cycle마다 변화된 페로몬 양은 식(2)와 같아진다.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

이때 $\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$ 이고

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ travels on edge } (i, j) \\ 0 & \text{otherwise} \end{cases}$$

그리고

- t :cycle 번호
- $\rho \in [0, 1]$:페로몬의 증발량을 나타내는 매개변수
- $\Delta\tau_{ij}$:edge(*i*, *j*)에서의 페로몬의 증가량
- $\Delta\tau_{ij}^k$:*k*번째 개미에 의한 edge(*i*, *j*)에서의 페로몬 증가량
- Q :개미가 지나간 개개의 경로당 뿌려지는 페로몬의 총량
- L_k :*k*번째 개미에 대한 경로의 총 이동거리

개미시스템은 매 cycle마다 식(1)에 따라 개개 개미들의 경로를 정하고, 그 경로에 의해서 식(2)에 의해 페로몬 양을 변화시킨다. 따라서 위의 과정을 반복적으로 수행함으로써 이동 거리가 짧은 경로로 페로몬의 분포가 수렴하게 된다.

위에서 만든 야드와 선박의 cluster를 서로 짝을 지어 주면서 작업 순서를 결정하기만 하면 첫 번째 모듈의 작업은 끝나게 된다. 본 연구에서는 이를 위하여 앞에서 소개된 기본적인 개미시스템에 약간의 변형을 주어 사용하였다. 본 시스템의 단계 I에서 다루는 문제는 기본적인 TSP 문제와 다음과 같은 몇가지 다른점을 갖고 있다.

- 기본적인 TSP 문제는 한번 방문한곳은 다시 갈 수 없다. 그러나 본 시스템은 방문한 곳이라도 아직 물량이 남아있으면 다시 방문할 수 있다.
- 기본적인 TSP 문제는 아직 방문하지 않은 모든 곳으로 갈 수 있다. 그러나 본 시스템은 선박에서 작업 가능한 cluster와 동일한 야드의 cluster들만 방문할 수 있다.
- 식(2)에서의 L_k 가 TC의 이동거리뿐만 아니라 setup time, 그리고 제약에 대한 penalty까지 포함한다.

개미시스템의 코드를 기술하면 아래와 같다.

For $t = 1$ to number of cycles do
 For $c = 1$ to number of CCs do
 For $k = 1$ to number of ants do

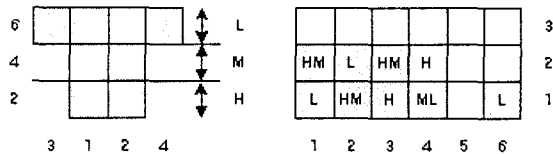
Repeat until ant *k* has treated all clusters of the *c*th CC
 Select the yard cluster *j* to be visited next with probability p_{ij}
 For $c = 1$ to number of CCs do
 Calculate the length L_k of the path generated by ant *k*
 Update the quantity of pheromone τ_{ij} on path of the ant *k*
 End

2.3 빔탐색법

빔탐색법(beam search)은 1970년대 후반 음성인식과 이미지처리 부분에 적용되어 사용되기 시작하여 스케줄링 문제에 사용되어 짧은 시간에 우수한 해를 제공하는 것으로 보고되었다[3]. 빔탐색법은 search tree에서 해공간을 줄이기 위해 단계마다 가능성이 있는 일정수의 가지만 남기고 나머지 가지를 없앤 후, 남아있는 가지 대해서 탐색을 수행하는 과정을 모든 단계가 탐색될 때까지 반복한다. 가능성이 있는 가지(또는 beam node)는 평가함수에 의해 결정되며, 어떠한 평가함수를 사용하는가에 따라 해의 질에 영향을 끼친다고 알려져 있다.

본 연구에서는 빔탐색법을 개미시스템이 적용된 후의 결과인 각 CC의 작업순서와 야드의 이동순서를 나타내는 야드cluster와 선박cluster의 쌍에 대해 개별 컨테이너의 작업순서와 적하위치를 결정하기 위해 사용하였다. 야드cluster와 선박cluster내의 컨테이너와 cell간에는 다대다 대응관계가 되며 이들을 연결시키기 위해 빔탐색법을 적용하였다.

<그림 6>에서 좌측은 선박의 적재가 가능한 cell을 나타낸다. cell의 오른쪽에 지시된 H, M, L은 지정된 단(tier)에 각각 적재되어야 하는 컨테이너의 무게그룹을 지정한 것이다. 우측은 장치장의 컨테이너 적재상황을 나타낸 것으로 격자안의 문자는 컨테이너의 무게를 기초로 하여 컨테이너가 속한 무게그룹을 나타내는 것으로, HM은 H그룹과 M그룹에 모두 포함 가능하다(ML도 동일). 선박의 대상 cluster내 적재가 되지않은 cell중에서 각 열의 가장 하단 cell이 적재가 가능하다(하단 cell을 비워두고 적재할 수는 없다).



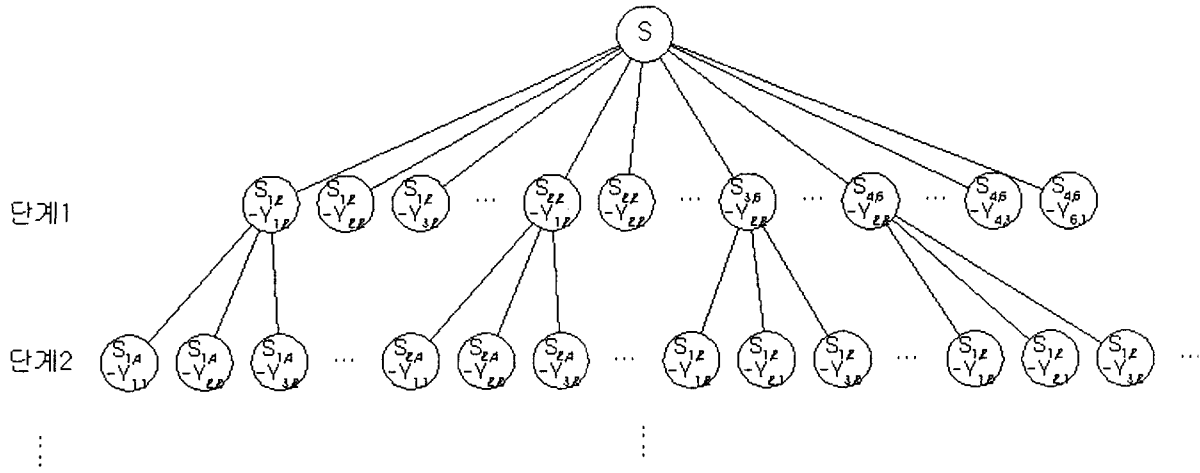
<그림 6> 빔탐색 대상

이들을 대상으로 야드의 컨테이너 중에서 재취급 횟수의 제한을 어기지 않는 컨테이너를 대응시켜 평가값을 계산한다. 평가값의 계산식은 아래와 같다.

$$p_{ij} = R_j + W_{ij} + CC_i + TC_j$$

이때

- i :선박의 cell index
- j :야드의 컨테이너 index
- p_{ij} :*j*컨테이너를 *i* cell에 적재할 때의 평가값
- R_j :*j* 컨테이너의 재취급 평가값
- W_{ij} :*i* cell과 *j* 컨테이너와의 무게편차 평가값
- CC_i :CC의 이전 작업위치와 *i* cell과의 작업편의성 평가값
- TC_j :*j* 컨테이너의 TC 작업용이성 평가값



<그림 7> 빔탐색의 예

작업순서	작업CC	컨테이너번호	야드위치	선박위치	목적항	크기	무게
1	108	HDMU2016775	3C-1-4-1	22-2-88	SIN	20	13
2	108	HDMU2112465	3C-1-5-2	22-0-88	SIN	20	11
3	108	HDMU2166944	3C-1-6-3	22-1-88	SIN	20	9
4	108	HDMU2235650	3C-1-6-1	22-2-90	SIN	20	12
5	108	HDMU2311169	3C-1-6-2	22-0-90	SIN	20	12
6	108	HDMU2915615	3C-1-4-2	22-1-99	SIN	20	10
7	108	CAXU7098428	2C-5-1-2	22-3-88	HKG	20	12
8	108	MOGU0048435	2C-5-2-2	22-3-90	HKG	20	13
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

<그림 8> 2단계 알고리즘의 적용결과 예

평가값이 우수한 일정수(빔넓이; beam width)의 조합을 단계 1의 빔노드로 선택한다. 단계 2로 진행하기 위해서는 단계 1에서 선택된 노드(circle)들을 대상으로 컨테이너가 1개 적하된 것으로 처리하고 나머지 컨테이너와 ship cell에 대해 같은 과정을 통해 진행해 나간다. 이때 빔넓이 만큼의 노드 선택은 상위 단계의 빔노드에서 가지를 치던지 관계 없이 동일 단계에서의 모든 대안을 평가 후보로 놓고 선택한다.

<그림 6>에 대한 빔탐색법의 적용 결과는 <그림 7>과 같다. 여기에서 재취급이 1미만인 야드의 컨테이너만을 대상으로 하였으며, 노드안의 기호 S_{ij} 는 선박의 i -열, j -단의 cell을 의미하며, Y_{kl} 은 야드의 컨테위치가 k -열, l -단임을 의미한다.

실제 현장의 적하 자료를 대상으로 2단계 알고리즘을 적용한 최종적인 적하순서의 예는 <그림 8>과 같다.

3. 결론

본 연구에서는 컨테이너 터미널의 적하 작업 계획시에 계획시간을 단축하고 선사의 요구사항과 터미널의 생산성을 향상시킬 수 있도록 하는 알고리즘을 개발하였다. 본 연구에서 제시된 알고리즘은 유연성과 터미널의 요구사항을 잘 만족하면서 작업효율을 높일 수 있도록 개발하였다. 그러나 일부 cluster가 너무 작게 만들어져 개미시스템의 경우 문제의 크기가 커지게 되면 수렴하는 시간도 길어지고, 해의 편차도 커지는 문제점이 있다. 따라서

향후 연구과제로서 합리적이고 빠른 처리시간의 휴리스틱 알고리즘을 개발하는 것이 필요하며, 실제 현장에 적용할 수 있는 시스템의 개발이 요구되어진다.

참고문헌

[1] A. Colomi, M. Dorigo and V. Maniezzo, "Distributed Optimization by Ant Colonies", Proceeding of ECAL-91 - European Conference on Artificial Life, Paris, France, pp.134-142, 1991.
 [2] A. Colomi, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini and M. Trubian, "Heuristics from Nature for Hard Combinatorial Problems", International Transactions in Operational Research, 3(1), pp.1-21, 1996.
 [3] S. De and A. Lee, "Flexible Manufacturing System (FMS) Scheduling using Filtered Beam Search", Journal of Intelligent Manufacturing, 1, No.3, pp.165-183, 1990.