Journal of the Korean Institute of Industrial Engineers
Vol. 37, No. 3, pp. 198-208, September 2011.

타부서치 알고리즘을 이용한 적치장의 블록 반출계획

이상협[1] · 김지온[1] · 문일경[2†]

[1]현대중공업 산업기술연구소 자동화연구실 / [2]부산대학교 산업공학과

Deployment Planning of Blocks from Storage Yards Using a Tabu Search Algorithm

Sanghyup Lee[1] · Jion Kim[1] · Ilkyeong Moon[2]

[1]Automation and Control Research Department, Industrial Research Institute,
Hyundai Heavy Industries Co. LTD., Ulsan 682-792, Korea
[2]Department of Industrial Engineering, Pusan National University, Busan 609-735, Korea

At a shipyard, the efficient handling of blocks is one of the most important factors in the shipbuilding process. We consider the problem of deployment planning of blocks from storage yards. As some information of block arrangement should be considered to handle the problem, we adopt the block arrangement based on the coordinates and sizes of each block at a storage yard. Deployment planning for a block involves deciding upon its transportation route from the storage yard and searching for blocks that would obstruct its transportation along this route. A tabu search algorithm for deploying several blocks is developed to minimize the number of obstructive blocks deployed together from the storage yards at a shipyard. The results of computational experiments show that the developed algorithm is very useful in the deployment planning of multiple blocks from the storage yards.

Keywords: Deployment Planning, Block Arrangement, Storage Yard, Shipyard, Tabu Search

1. 서 론

선박 블록 건조공법은 블록 단위의 중간 제품을 제작하여 최종적으로 도크(Dock)에서 탑재하는 선박건조 방식이다. 선박건조 과정에서 제작되는 중간 제품인 재공품을 블록(Block)이라고 하며, 하나의 선박은 100여 개의 블록이 조립되어 완성된다. 일반적으로 선박건조에서 임의의 공정을 끝내고 블록을 다음 작업장으로 이동할 때, 다음 공정에서 블록을 수용할 공간이 부족한 경우가 빈번하게 발생하므로 블록을 임시로 보관할 공간이 필요하게 된다. 이런 이유로 블록을 필요한 시점에 필요한 곳으로 운반하여 공급하는 것은 중요 관리 항목 중의 하나이다. 더욱이 선박 건조량이 증가할 경우, 선박건조 과정 중에 블록 수량도 증가하여 블록의 관리의 중요성은 훨씬 강조된다. 하지만 블록 수량의 증가에 비해 블록운반 관리가 효율적이지 못하면, 적치장 공간 활용 효율이 낮아지고 불필요한 운반이 많아지므로 전체 선박건조의 생산성이 저하된다.

블록은 트랜스포터(Transporter)로 운반되는데, 이는 블록 및 선박용 엔진 등의 대형 구조물을 운반하기 위한 특수 운반 차량이다. 블록을 보관하는 장소를 적치장(Storage Yard)이라 하며, 이곳에 블록은 4개 이상의 지지대(Supporter) 위에 놓여져 있다. 조선 야드의 적치장에는 블록을 직접 들어 올릴 수 있는 크레인 등의 장비가 없으므로 트랜스포터가 블록을 받치고 있는 지지대 사이로 진입하여 자체 유압 장치에 의해 높이를 조정하여 블록을 싣고 내린다.

블록 운반의 주요 목표는 운반할 블록을 이동 목적지에 빠른 시간 내에 운반 하는 것인데, 이러한 목표에 맞는 블록 운반

본 논문은 2011년 교육과학기술부로부터 지원받아 수행된 연구임(지역거점연구단육성사업/차세대물류IT기술연구사업단).
† 연락저자 : 문일경 교수, 609-735 부산광역시 금정구 장전동 산30번지 부산대학교 공과대학 산업공학과, Tel : 051-510-2451, Fax : 051-512-7603, E-mail : ikmoon@pusan.ac.kr
2011년 5월 7일 접수; 2011년 7월 11일 게재 확정.

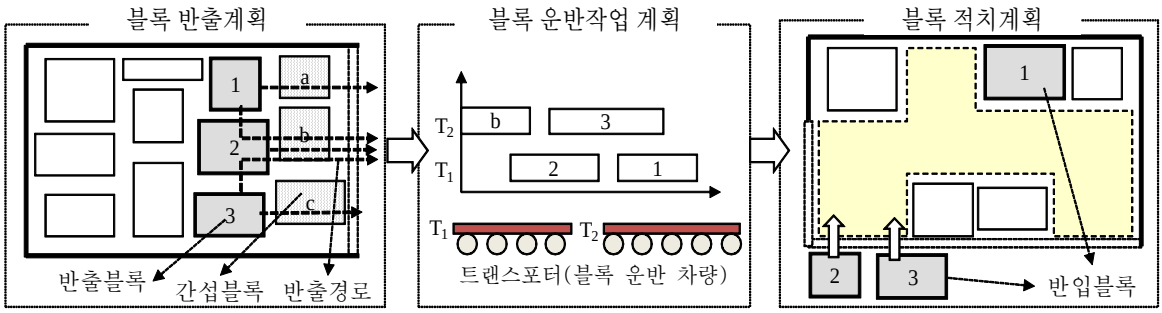

Figure 1. Block transportation planning problems in a shipyard

계획을 수립하는 것은 중요한 일이지만 매우 어려운 문제이다. <Figure 1>에 나타낸 바와 같이 조선 야드의 블록 운반계획에는 다음의 3가지 사항을 고려해야 한다.

먼저, 운반할 블록들을 적치장에서 반출하기 위해 우선적으로 고려해야 할 것은 이들이 반출되는 경로에 놓인 블록들도 함께 반출하여야 한다는 것이다. 따라서 운반할 블록들을 정해진 시간에 운반하기 위해서는 추가로 운반되는 블록 수를 최소화하는 반출방법이 요구된다.

다음으로 운반 블록들에 대한 운반장비(트랜스포터)의 운반작업 일정계획을 고려해야 한다. 즉, 운반될 각 블록을 어느 운반장비로 운반할 것 인지와 각 운반장비에 할당된 블록들에 대해 어떤 순서로 언제 운반할 것 인지를 계획해야 한다.

마지막으로 운반 블록이 반입되는 적치장에서 해당 블록을 어느 위치에 적치할 것 인지를 결정해야 한다. 적치장의 크기가 제한되어 있으므로 적치장의 공간을 효율적으로 사용할 수 있도록 이동될 블록의 적치위치를 결정하는 방법이 필요하다.

<Figure 1>에 나타낸 블록 운반계획에 대한 3가지 고려사항에서 가장 중요한 것은 운반될 블록의 수량을 결정하는 반출계획이라 할 수 있다. 즉, 반출계획에서 얼마큼의 추가 운반 블록을 계획하는지에 따라서 트랜스포터의 운반 작업계획과 블록의 적치계획이 달라진다. 따라서 운반 신청된 블록들을 정해진 시간 내에 운반하기 위해서는 추가로 운반되는 블록 수를 최소화하는 반출 방법이 필요하다.

블록과 같은 대상물의 공간 배치 및 저장과 관련된 문제는 조선 야드의 적치장과 항만 컨테이너 터미널 등의 운영에 대한 연구에서 많이 다루어졌다. 항만 컨테이너 터미널의 수출 및 수입되는 컨테이너를 효율적으로 취급하기 위해 적치 공간에 컨테이너를 보관하는 위치를 결정하는 문제를 다루었다 (Bazzazi *et al*., 2009; Kozan and Preston, 2006; Zhang *et al*., 2003). 그리고 Kim and Hong(2006)은 적치된 컨테이너를 꺼내는 방법에서 컨테이너를 취급하는 순서와 취급된 컨테이너의 위치를 결정하는 문제를 다루었다. 항만 컨테이너 터미널의 컨테이너 취급에 대한 문제와 조선 야드의 블록 운반 문제는 제한된 저장 공간을 효율적으로 활용한다는 측면에서는 유사성이 있다고 할 수 있다. 컨테이너 터미널에서는 컨테이너를 겹쳐 쌓는 것이 가능하므로 3차원으로 보관되지만, 컨테이너를 크레인으로 취급하기 때문에 많은 관련 연구에서는 컨테이너의 적치 상태를 2차원으로 다루고 있다. 이 또한 조선 야드에서 블록배치 상태를 2차원으로 다루고 있는 것과 유사하다.

컨테이너 취급과 블록 운반 문제를 비교하면, <Figure 2>에서와 같이 컨테이너의 크기는 정형화되어 있는 반면에 선박블록은 크기와 모양이 다양하다. 따라서 컨테이너의 적치는 적치공간을 단위 구간으로 나누어 단위 구간에 컨테이너가 배치되는 것으로 표현할 수 있으나, 블록의 배치는 블록의 크기와 형상에 따라 다양하고 유연한 배치 형태로 표현해야 한다. 왜냐하면 블록배치를 컨테이너 배치 형태로 표현하면 문제 해결에 대한 현실성이 결여될 수 있기 때문이다. 그리고 컨테이너 적치장에서 컨테이너의 반출이나 반입은 한쪽 측면에서만 발생하므로 컨테이너 움직임을 한 방향으로만 가정할 수 있다. 반면에 블록 적치장의 출입구는 여러 개가 있고 블록 이동을 위한 움직임 방향도 다양하게 고려할 수 있다. 따라서 컨테이너 취급에서 특정 컨테이너의 이동 시 간섭되는 컨테이너를 파악하는 것은 비교적 쉽지만, 블록 적치장에서 특정 블록의 이동에 따른 간섭되는 블록을 파악하는 것은 쉽지 않다. 이런 점들을 고려하면 조선 야드의 블록 운반 문제는 컨테이너 취급 문제보다 더 복잡해진다.

따라서 조선 야드의 블록 운반 문제는 컨테이너 터미널의 컨테이너 취급 문제와는 또 다른 측면에서 다루어질 필요가 있다. 즉, 컨테이너 취급과는 다르게 블록의 2차원 배치, 블록의 다양한 크기 및 이동방법 등을 고려하여 블록 운반 문제를 다루어야 한다.

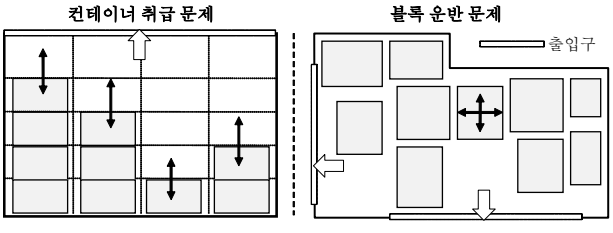

Figure 2. Comparison between container handling and block transportation problems

조선업체에서 블록 운반 외의 공간 활용 문제는 공장 내에서 주어진 공간에 작업 대상물인 블록을 배치하는 공간 일정계획 분야에서 주로 다루어졌다(Koh *et al*., 1999; Lee *et al*.,

1996; Lee *et al.*, 1997). 공간 일정계획은 블록 조립공장 내에서 블록배치 방법에 대한 것으로 주어진 평면 공간에 다각형 형상의 블록을 최대로 배치하는 것을 목적으로 한다. 그리고 공간 일정계획에서는 공장 내에서 블록을 크레인으로 운반하므로 블록 이동보다는 블록의 크기와 형상에 대한 제약을 주로 고려하여 문제를 다루었다.

공장 내에서와 달리 블록을 크레인으로 취급할 수 없는 조선 야드 적치장에서 블록의 취급 방법과 보관 위치에 대한 문제가 일부 연구에서 다루어졌다(Park and Seo, 2009a; Park and Seo, 2009b; Park and Seo, 2010). 이러한 기존 연구에서는 직사각형의 적치장을 m×n의 셀(Cell)로 나누고, 블록을 하나의 셀 내에 배치하는 것으로 가정하고 있다. 즉, 블록의 다양한 크기를 반영하기 보다는 컨테이너처럼 블록의 크기를 정형화하여 해당 문제를 다루었다. 반면에 본 연구에서는 블록의 다양한 크기와 적치장에서 평면 2차원 블록 배치를 고려한 블록 반출계획 문제를 다루고자 한다.

본 연구는 5장으로 구성되어 있으며 각 장의 내용은 다음과 같다. 제 2장은 반출블록에 대한 반출방법과 간섭블록 탐색 방법을 소개하고, 제 3장에서는 다수 블록 반출에 대한 타부서치 알고리즘의 적용 방법을 제안하며, 제 4장에서 본 연구에서 제안한 타부서치 알고리즘의 효율성을 수치 실험을 통해 검증한다. 마지막으로 제 5장에서 결론 및 향후 연구를 제시한다.

2. 블록 반출 방법 및 간섭블록 탐색

본 연구에서는 블록 운반신청에 의해 적치장에서 꺼내야 하는 블록을 '반출블록', 반출블록을 꺼내기 위해 더불어 반출하는 블록을 '간섭블록'이라고 정의한다. 그리고 적치장내에서 반출블록이 현 위치에서 적치장의 출구까지 이동하는 경로를 '반출경로'라고 정의한다. 그리고 반출블록과 간섭블록은 해당 적치장에서 반출되어 정해진 다른 적치장으로 옮겨지는 것을 가정한다. 그리고 현실적으로 임의의 적치장에서 블록의 반출과 반입작업을 동시에 실시하는 것은 흔한 경우가 아니므로 본 연구의 블록 반출계획에서는 블록의 반출과 반입이 동시에 발생하는 상황을 고려하지 않는다.

조선 야드에서 블록을 적치하는 적치장의 형상은 매우 다양하다. 하지만 적치장에서 블록을 배치할 때, 블록의 효율적인 취급을 위해 적치장의 형상에 맞춰 블록을 배치하기 보다는 주로 열이나 줄을 맞춰 블록을 배치한다. 따라서 본 연구에서는 적치장의 형상을 수직과 수평 선분으로 구성된 직각다각형으로 정의한다. 적치장 형상을 구성하는 직선 선분은 <Figure 3>의 오른쪽과 같이 블록의 반출이 가능한 '출구'와 불가능한 '벽'으로 구분한다. 그리고 블록을 반출하기 위해 이동하는 경우 블록의 가로 또는 세로 폭 크기의 통로가 필요하므로 블록의 형상을 <Figure 4>의 오른편과 같이 블록이 포함되는 최소의 직사각형으로 표현한다.

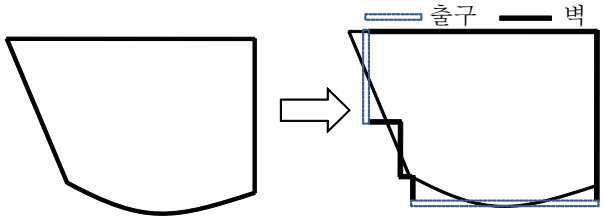

Figure 3. Representation of a planar storage yard

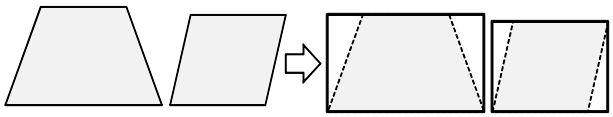

Figure 4. Representation of the shape of blocks

조선업체에서 기존의 블록관리는 적치장을 정해진 격자 단위로 나누고 지번을 부여하여 지번 기준으로 블록배치를 관리하였다. 지번기반의 블록배치에서는 임의의 블록은 하나의 격자 내에 배치되는 것을 기본 전제로 한다. 하지만 지번기반의 블록관리에서는 블록의 크기가 다양하고 블록의 크기와 지번의 크기가 일치하지 않기 때문에 하나의 블록이 두 개 이상의 지번을 차지 것과 같은 적치장의 블록배치 상황을 정확하게 반영하지 못한다. 따라서 지번기반의 블록배치에서는 현실적인 블록 반출계획의 수립이 매우 어렵다. <Figure 5>에서 보는 것처럼 지번기반 블록관리에서는 출구까지의 지번들이 모두 비어있을 경우에 블록 A의 반출이 가능하다고 판단하는 반면, 실제로는 블록 반출이 불가능할 수 있다. 그리고 블록 B의 경우, 실제 반출이 가능한 상황이지만 지번기반의 블록배치에서는 반출이 불가능한 것으로 판단한다.

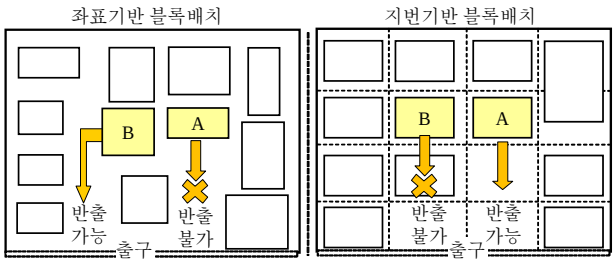

Figure 5. Differences between two methods for block arrangement

<Figure 5>의 오른쪽과 같이 블록배치를 지번기반으로 나타낼 경우에 블록 반출계획의 현실성이 떨어지므로 본 연구에서는 블록의 위치와 크기를 고려한 좌표기반의 블록배치에서 블록 반출계획 문제를 다루고자 한다. 좌표기반으로 블록과 적치장의 위치를 나타내기 위해 <Figure 6>과 같이 적치장의 좌표의 원점은 좌측 상단으로 정의하며, 오른쪽과 아래로 갈수록 좌표 값이 증가한다. 마찬가지로 직사각형으로 정의된 블록의 위치는 적치장의 좌표 원점을 기준으로 한 좌측 상단 점의 좌표로 표현한다. 본 연구에서 임의의 블록이나 개체 A의 위치와 크기에 대한 기호 정의는 다음과 같다.

• (x_A, y_A) : 2차원 평면에서 A의 위치
• h_A : A의 세로 폭
• w_A : A의 가로 폭

그리고 가로형 출입구(E1)의 세로 폭 (hE1)과 세로형 출입구(E2)의 가로 폭(w_{E2})은 0으로 정의한다.

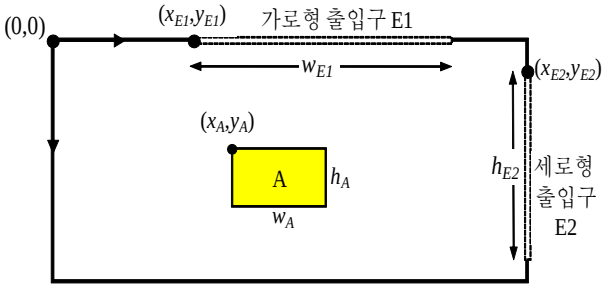

Figure 6. Representation of the coordinates of block and exits of a storage yard

2.1 반출경로 생성

본 절에서는 반출블록의 반출경로를 만들기 위해 반출점(deployment point)을 설정하는 방법과 정해진 반출점까지 블록의 이동방법을 제안한다. 먼저 반출경로를 생성하기 위해 적치장에서의 블록 이동 규칙에 대해 설명한다. 블록을 운반하는 트랜스포터는 대형 구조물을 운반하기 위한 특수 차량으로 상부에 자신보다 크기가 큰 블록을 싣고 운반한다. 따라서 적치장에서 블록의 반출경로를 고려할 때 트랜스포터의 크기보다는 블록의 크기를 반영한다. 트랜스포터는 방향 전환이 자유로워 다양한 이동 경로를 갖는다. 하지만 대형 구조물을 운반하는 트랜스포터는 잦은 방향 전환을 자제하고 필요할 경우에만 방향을 전환한다. 따라서 본 연구에서는 적치장에서의 블록 이동을 다음과 같이 가정한다.

• 가능한 블록 이동방법 :
  - 수평과 수직 이동
• 불가능한 블록 이동방법 :
  - 사선 이동(<Figure 7>-①)
  - 정해진 반출점에서 멀어졌다가 접근하는 우회 이동 (<Figure 7>-②)
  - 출구에 도달한 다음에 출구를 따라 움직이는 이동 (<Figure 7>-③)

앞서 블록의 이동방법을 가정한 바에 따라 <Figure 7>에서 블록 A를 정해진 반출점으로 운반하기 위한 반출경로는 수평과 수직 이동으로 조합된 경로만을 고려한다. 그리고 이러한 반출경로의 구성은 블록의 현 위치와 반출점을 기준으로 정해진 사각형 범위 내에서만 고려한다.

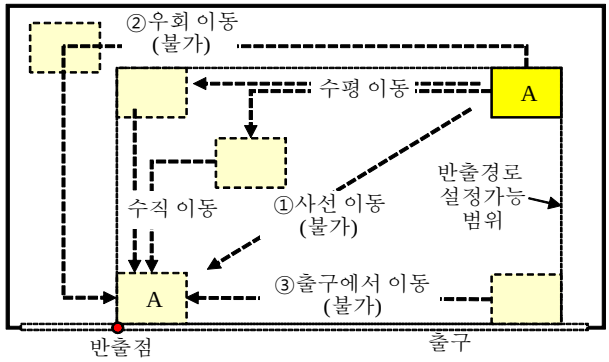

Figure 7. Methods of block movement in a storage yard

2.1.1 반출점 설정

반출점은 반출블록이 적치장을 벗어나는 지점으로 적치장의 출구에 설정한다. 본 연구에서는 반출블록의 반출점을 적치장에 배치된 각 블록에 대해 해당 블록을 지나치지 않고 접하는 지점과 적치장 출구 좌표를 기준으로 설정한다. 본 절에서는 반출블록의 현 위치와 적치장의 출구 타입에 따라 반출점을 설정하는 방법을 소개한다. 이를 위해 사용한 해당 적치장의 블록과 출구를 나타내는 기호의 정의는 다음과 같다.

• Bi : 적치장에 배치된 i번째 블록
• Ej : 적치장의 j번째 출구

먼저, 블록을 가로형 출구로 반출하는 경우 반출블록의 왼쪽과 오른쪽에 위치한 블록으로 구분하여 반출점을 설정한다. 반출블록을 좌측으로 이동하여 반출하는 경우에는 반출블록의 좌측에 위치한 블록의 우변을 기준으로 반출점을 설정하고, 우측으로 이동하여 반출하는 경우에는 우측에 위치한 블록의 좌변을 기준으로 설정한다. <Figure 8>에서 반출블록 A를 좌측으로 이동한 후 블록 B1를 접하면서 출구 E1와 E2로 반출하는 반출점은$(x_{B1}+w_{B1}, y_{E1})$와 $(x_{B1}+w_{B1}, y_{E2})$가 된다. 그리고 블록 A를 오른쪽으로 이동한 후 블록 B2를 접하면서 반출할 경우 반출점은(x_{B2}, y_{E1})와 (x_{B2}, y_{E2})가 된다.

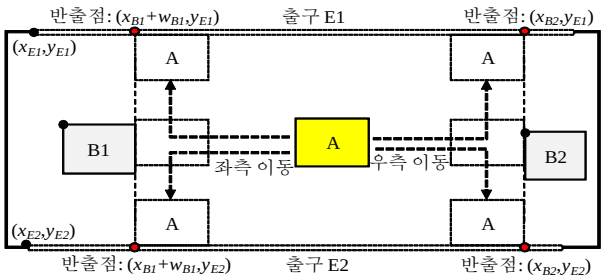

Figure 8. Setting deployment points on the horizontal type exits

임의의 반출블록 A에 대한 가로형 출구에서의 반출점은 자신의 왼쪽 또는 오른쪽에 위치한 블록에 따라 다음과 같이 설정한다.

• 반출점이 $(x_{Bi}+w_{Bi}, y_{Ej})$로 설정되는 대상 블록 범위 :
  1) $x_{Ej} \le x_{Bi}+w_{Bi} \le min\{x_{Ej}+w_{Ej}-w_A, x_A\}$
  2) $y_{Ej} < y_{Bi} < y_A+h_A$ 또는 $y_A < y_{Bi}+h_{Bi} < y_{Ej}$, $\forall i$, $\forall j$
• 반출점이(x_{Bi}, y_{Ej})로 설정되는 대상 블록 범위 :
  1) $max\{x_{Ej}, x_A\}+w_A < x_{Bi} < x_{Ej}+w_{Ej}$
  2) $y_{Ej} < y_{Bi} < y_A+h_A$ 또는 $y_A < y_{Bi}+h_{Bi} < y_{Ej}$, $\forall i$, $\forall j$
  그리고 다음 조건의 출구 끝 지점을 반출점으로 추가한다.
• 반출점(x_{Ej}, y_{Ej}) : $x_{Ej} \le x_A$, $w_{Ej}>0$, $\forall j$
• 반출점$(x_{Ej}+w_{Ej}, y_{Ej})$: $x_{Ej}+w_{Ej} \ge x_A+w_A$, $w_{Ej}>0$, $\forall j$

다음으로 세로형 출구에서는 반출블록의 위쪽과 아래쪽에 위치한 블록을 구분하여 반출점을 설정한다. 블록을 위로 이동하여 반출하는 경우에는 위쪽에 위치한 블록의 밑변을 기준으로 반출점을 설정하고, 아래로 이동하여 반출하는 경우에는 아래쪽에 위치한 블록의 윗변을 기준으로 설정한다. <Figure 9>에서 블록 A를 위로 이동한 후 블록 B1을 접하면서 출구 E1과 E2로 반출하는 반출점은 각각$(x_{E1}, y_{B1}+h_{B1})$과 $(x_{E2}, y_{B1}+h_{B1})$가 된다. 블록 A를 아래로 이동하여 블록 B2를 접하면서 출구 E1과 E2로 반출하는 반출점은 각각(x_{E1}, y_{B2})와 (x_{E2}, y_{B2})가 된다.

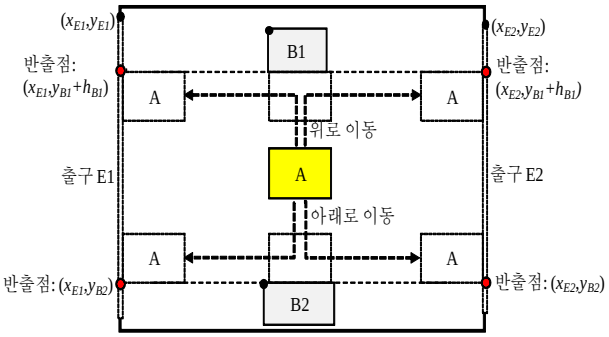

Figure 9. Setting deployment points on the vertical type exits

임의의 반출블록 A에 대한 세로형 출구에서의 반출점은 자신의 위쪽 또는 아래쪽에 위치한 블록에 따라 다음과 같이 설정한다.

• 반출점이$(x_{Ej}, y_{Bi}+h_{Bi})$로 설정되는 대상 블록 범위 :
  1) $y_{Ej} \le y_{Bi}+h_{Bi} \le min\{y_{Ej}+h_{Ej}-h_A, y_A\}$
  2) $x_{Ej} < x_{Bi} < x_A+w_A$ 또는 $x_A < x_{Bi}+w_{Bi} < x_{Ej}$, $\forall i$, $\forall j$
• 반출점이(x_{Ej}, y_{Bi})로 설정되는 대상 블록 범위:
  1) $max\{y_{Ej}, y_A\}+h_A < y_{Bi} < y_{Ej}+h_{Ej}$
  2) $x_{Ej} < x_{Bi} < y_A+w_A$ 또는 $x_A < x_{Bi}+w_{Bi} < x_{Ej}$, $\forall i$, $\forall j$
  그리고 다음 조건의 출구 끝 지점을 반출점으로 추가한다.
• 반출점(x_{E1}, y_{E1}) : $y_{Ej} \le y_A$, $h_{Ej}>0$, $\forall j$
• 반출점$(x_{Ej}, y_{Ej}+h_{Ej})$: $y_{Ej}+h_{Ej} \ge y_A+h_A$, $h_{Ej}>0$, $\forall j$

2.1.2 반출점과 경유점에 따른 블록의 반출경로 설정

반출블록이 현 위치에서 반출점까지 이동하는 방법은 수평과 수직 이동으로 제한하더라도 매우 다양하다. 하지만 현실적으로 트랜스포터가 블록을 싣고 이동할 때 많은 방향 전환을 하지 않으므로 본 연구에서는 3회 이상 방향 전환이 포함된 반출경로는 고려하지 않는다.

적치장의 출구가 가로형 또는 세로형에 있는지 따라 반출점을 설정하는 방법이 다르므로 반출경로도 출구 타입에 따라 생성하는 방법을 제안한다. 먼저, 반출블록의 반출점이 가로형 출구에 있는 경우의 반출경로를 생성하는 2가지 방법을 소개한다. 가로형 출구의 반출점까지 반출경로를 구성하는 첫 번째 방법은 <Figure 10>과 같이 반출블록을 정해진 반출점까지 먼저 수평 방향으로 반출점의 X좌표까지 이동하고, 방향을 전환하여 수직 이동으로 반출점의 Y좌표까지 이동하는 것이다.

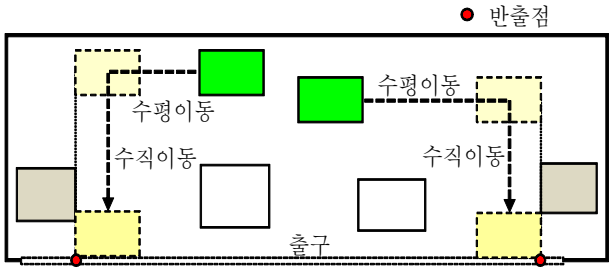

Figure 10. Moving route with horizontal → vertical movements for a deployment point on the horizontal type exits

가로형 출구의 반출점까지 경로를 구성하는 다른 방법은 반출블록과 반출점 사이에 배치되어 있는 임의의 블록을 우회하는 반출경로를 구성하는 것이다. 이를 위해 사전에 우회할 블록의 윗변이나 밑변을 기준으로 경유점(pass through point)을 설정하여 <Figure 11>과 같이 수직 → 수평 → 수직 이동으로 반출점에 도달하는 것이다. <Figure 11>에서 반출블록을 이동하는 방식은 같지만 반출블록과 반출점의 위치 관계에 따라 상·하 수직 이동과 좌 · 우 수평 이동의 다양한 조합으로 구성됨을 알 수 있다.

임의의 반출블록에 대해 가로형 출구에 있는 반출점까지 특정 블록을 우회하는 반출경로를 구성할 때, 반출블록과 반출점 사이에 배치되어 있는 블록들이 우회할 대상이 된다. 그리고 경유점으로 우회할 블록의 밑변 또는 윗변을 기준으로 네 꼭지점 중에서 하나를 선정한다.

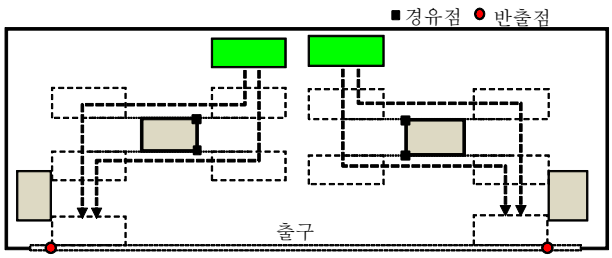

Figure 11. Moving route with vertical → horizontal → vertical movements for a deployment point on the horizontal type exits

다음으로, 적치장의 세로형 출구에 반출점이 있는 경우에 반출경로를 구성하는 2가지 방법을 소개한다. 세로형 출구의 반출점까지 반출경로를 구성하는 첫 번째 방법은 <Figure 12>와 같이 먼저 반출점의 Y좌표까지 수직으로 이동한 후, 이동 방향을 전환하여 반출점의 X좌표로 수평으로 이동하는 것이다.

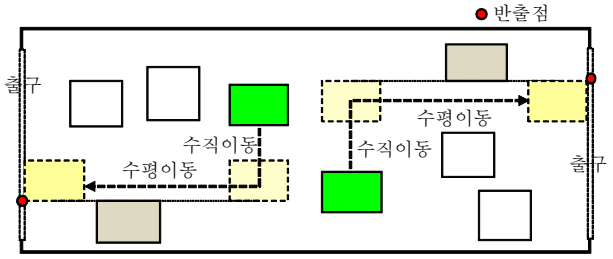

Figure 12. Moving route with vertical → horizontal movements for a move-out point on the vertical type exits

세로형 출구의 반출점까지 반출경로를 구성하는 두 번째 방법은 <Figure 13>과 같이 우회할 블록의 좌변 또는 우변을 따라 우회하기 위해 수평 → 수직 → 수평 이동의 경로를 구성하는 것이다. 가로형 출구에서와 마찬가지로 반출블록과 반출점 사이에 배치되어 있는 블록들이 우회할 대상이 된다. 경유점은 우회할 블록의 네 꼭지점 중에서 좌변 또는 우변을 기준으로 하나를 선정한다.

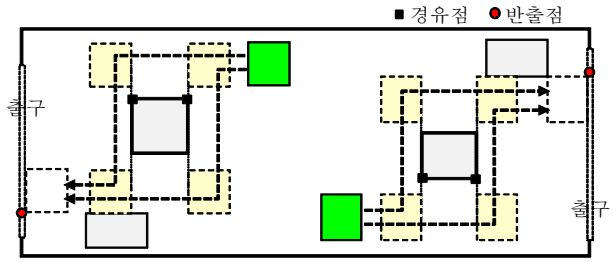

Figure 13. Moving route with horizontal → vertical → horizontal movements for a deployment point on the vertical type exits

2.2 간섭블록 탐색

반출블록이 제 2.1절에서 설정된 반출경로를 따라 이동할 때, 반출경로 상에 놓인 블록도 함께 반출해야 되는데 이러한 블록은 반출블록에 대한 '간섭블록'이 된다. 이러한 간섭블록은 반출블록에 앞서 반출해야 하므로 간섭블록의 크기를 블록 반출에 고려해야 한다. 즉, 주어진 반출경로를 따라서 경로상에 있는 블록을 탐색할 때 현재까지 반출해야 되는 블록들보다 더 큰 블록이 있으면, 그 이후는 해당 블록의 크기를 기준으로 블록 반출을 고려해야 한다.

본 연구에서는 반출블록의 주어진 반출경로에 어떤 블록이 있는지를 탐색하기 위해 가상으로 '탐색블록'을 생성하고, 탐색블록의 반출경로상에 있는 블록을 탐색하여 해당 블록의 크기를 블록 반출에 고려한다. 탐색블록이 수직 또는 수평으로 이동할 때, 이동의 기준이 되는 직선을 '이동기준선'이라고 한다. 탐색블록이 정해진 목표지점으로 수직 이동할 경우에 이동기준선은 탐색블록의 X좌표를 기준으로 한 수직선이고, 수평 이동할 때 이동기준선은 탐색블록의 Y좌표를 기준으로 한 수평선이 된다. 간섭블록을 탐색하기 위해 탐색블록은 이동기준선을 따라 이동하다가 반출경로에서 간섭되는 블록이 있으면, 해당 블록의 가로와 세로 폭을 자신의 각 폭과 비교하여 자신의 폭이 작으면 해당 블록의 폭만큼 늘린다. 이와 같은 탐색블록의 이동방법에 따라 본 절에서는 블록의 수직 또는 수평 이동에 따라서 간섭블록을 탐색하는 방안을 제안한다.

먼저, 반출블록이 수직 방향으로 이동할 때의 간섭블록 탐색 방법을 소개한다. 간섭블록을 탐색하기 위해 탐색블록이 현 지점에서 정해진 목표지점(반출점 또는 경유점)까지 수직으로 이동할 때, 목표지점이 탐색블록의 초기 위치에서 좌변 또는 우변에 맞춰 있는 경우로 나누어 간섭블록을 탐색하는 방법을 제시한다.

탐색블록의 초기 위치에서 탐색블록의 좌변이 목표지점과 일치하고 탐색블록이 수직으로 이동할 때, 간섭블록을 탐색하는 방법을 <Figure 14>의 왼쪽에서 보여주고 있다. 반출블록이 계획된 반출경로를 따라 이동할 때 발생하는 간섭블록을 탐색하기 위해 반출블록과 같은 크기의 탐색블록을 같은 위치에 생성하고, 탐색블록의 이동을 위한 이동기준선을 탐색블록의 X좌표를 기준으로 설정한다. 이동기준선에 따라 탐색블록을 이동하던 도중에 자신의 이동 경로에 포함되는 블록을 간섭블록으로 지정한다. 그리고 간섭블록의 가로와 세로 폭을 각각 탐색블록과 비교하여 탐색블록보다 크면, 탐색블록의 해당 폭을 간섭블록만큼 늘린다. 탐색블록의 가로 폭을 증가시킨 경우에 이동기준선으로부터 탐색블록이 가로 방향으로 적치장이나 출구를 벗어나면 현 지점에서 적치장의 최대 X좌표를 기준으로 이동기준선을 왼쪽으로 이동한다. 여기서 목표지점이 경유점이라면 경유점이 있는 우회 블록도 간섭블록으로 설정한다. 그리고 왼쪽으로 이동된 이동기준선의 X좌표 값이 현 지점의 최소 X좌표 값보다 작으면, 해당 반출경로를 통해 블록을 반출하는 것이 불가능한 것으로 판단한다.

탐색블록의 초기 위치에서 탐색블록의 우변이 목표지점과 일치하고 블록이 수직으로 이동할 때, 간섭블록을 탐색하는 방법을 <Figure 14>의 오른쪽에서 보여주고 있다. 목표지점이 좌변과 일치할 때와 마찬가지로 간섭블록 탐색을 위한 탐색블록을 생성하고, 이동기준선을 탐색블록의 X좌표를 기준으로 설정한다. 이동기준선에 따라 탐색블록을 이동하던 도중에 자신의 이동 경로에 포함되는 블록을 간섭블록으로 지정한다. 그리고 간섭블록의 가로와 세로 폭을 각각 비교하여 탐색블록보다 크면 탐색블록의 해당 폭을 간섭블록만큼 늘리고, 이동기준선을 목표지점의 X좌표에서 탐색블록의 가로 폭을 차감(왼쪽으로 이동)하여 설정한다. 이 때 설정된 이동기준선이 적

치장이나 출구의 범위 내에 포함되지 않으면 현 지점에서 적치장의 최소 *X*좌표 값으로 이동기준선을 다시 설정한다. 여기서 설정된 이동기준선에서 탐색블록의 가로 폭이 적치장이나 출구를 벗어나면 해당 반출경로를 통해 블록을 반출하는 것이 불가능한 것으로 판단한다.

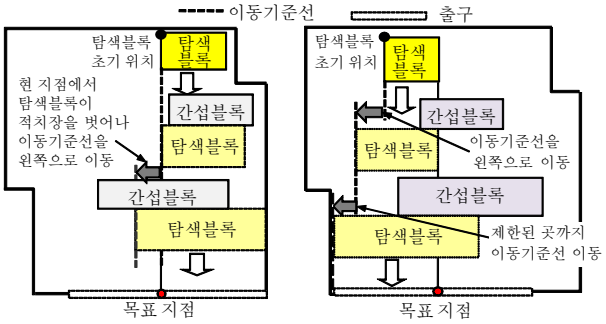

Figure 14. Methods of searching obstructive blocks for vertical movements

다음으로 반출블록을 수평 방향으로 이동할 때, 간섭블록을 탐색하는 방법을 소개한다. 간섭블록을 탐색하기 위해 탐색블록을 정해진 목표지점(반출점 또는 경유점)까지 수평으로 이동할 때, 목표지점이 탐색블록의 초기 위치에서 윗변 또는 밑변에 맞춰져 있는지에 따라 간섭블록을 탐색하는 방법을 제시한다. <Figure 15>는 탐색블록의 초기 위치에서 탐색블록의 윗변 또는 밑변이 목표지점과 일치하는 경우에 대한 간섭블록 탐색 방법을 보여주고 있다. 탐색블록 생성 후에 이동기준선을 *Y*좌표를 기준으로 설정하고 반출경로를 따라 이동하면서 간섭블록이 있으면 세로와 가로 폭을 각각 비교하여 탐색블록보다 크면 탐색블록의 해당 폭을 변경한다. 간섭블록의 세로 폭이 탐색블록의 세로 폭보다 더 크면 목표지점이 초기 탐색블록의 윗변과 일치하는 경우에는 이동기준선의 위치를 그대로 유지하는 반면에, 목표지점이 초기 탐색블록의 밑변과 일치하는 경우에는 이동기준선의 위치를 변경한다. 이동기준선의 위치와 탐색블록의 세로 폭을 고려하여 세로 방향으로 적치장이나 출구의 제한된 범위를 벗어나면 수직 이동의 경우와 같은 방안으로 이동기준선을 변경한다. 마찬가지로 수평 이동할 때의 간섭블록 탐색 절차도 수직으로 이동 할 때와 같은 방법으로 적용할 수 있다.

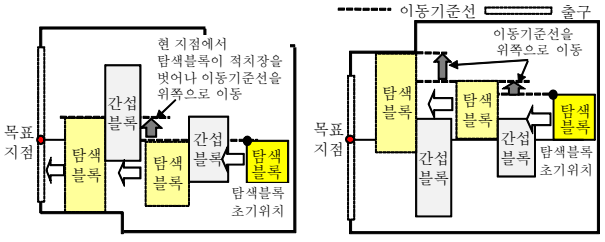

Figure 15. Methods of searching obstructive blocks for horizontal movements

3. 다수 블록 반출을 위한 타부서치 알고리즘

적치장에서 임의의 블록을 반출할 때, 간섭블록의 수를 최소로 하는 반출 방법은 제 2.1절에서 제안한 모든 반출경로를 고려하여 찾을 수 있다. 하지만 적치장에 다수의 반출블록이 있을 경우, <Figure 16>의 왼쪽과 같이 모든 반출블록에 대해 간섭블록 수가 최소인 각 반출경로를 통해 반출하는 것이 반드시 전체 간섭블록의 수를 최소화하지 못한다. 반면에 <Figure 16>의 오른쪽과 같이 공동 반출경로를 이용하는 것이 전체 간섭블록을 줄이게 된다. 이와 같이 적치장에 다수의 블록을 반출할 때는 각 반출블록에 대해 간섭블록 수가 최소인 반출경로를 설정하기 보다는 전체 간섭블록 수가 최소되는 반출경로를 찾는 것이 필요하다.

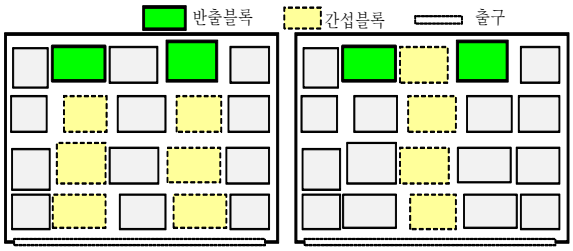

Figure 16. Comparison between individual and common moving routes in terms of the number of obstructive blocks

본 절에서는 적치장의 다수 반출블록에 대해 전체 간섭블록 수를 최소화하는 반출블록들의 반출경로 조합을 찾는 방법을 제안한다. 임의의 적치장에서 반출블록의 수가 많을수록 반출블록들의 반출경로의 조합도 급격하게 증가하므로 다수 반출블록에 대해 전체 간섭블록 수를 최소화하는 반출경로의 조합을 찾는 것은 복잡한 문제가 된다. Glover(1989)에 의해 소개된 타부서치(Tabu Search)는 일정계획 등과 같은 조합 최적화 분야에서 좋은 결과를 찾는 것으로 알려져 있다. 이에 본 연구에서는 다수 블록의 반출계획에 타부서치 알고리즘을 적용한다.

3.1 개체 표현(Individual representation)

제 2.1.1절에서 살펴본 바와 같이 반출블록의 반출점을 출구 타입에 따라 생성하므로 반출점의 속성에 출구 타입이 포함되어 있다. 그리고 주어진 반출점과 배치된 블록의 경유점을 기준으로 반출경로를 생성하고 이동 경로에 따라 간섭블록을 찾는 방법을 제 2.2절에 제시하였다. 따라서 임의의 반출블록에 대한 경유점과 반출점이 주어지면 반출블록에 대한 반출경로를 설정하고 이에 대한 간섭블록을 찾을 수 있다.

본 연구에서는 임의의 반출경로를 반출점과 경유점을 조합하여 하나의 반출번호로 나타낸다. 그리고 전체 반출블록의 반출경로에 대한 개체는 각 반출블록의 반출번호 값으로 구성된 스트링으로 나타낸다. <Figure 17>에서와 같이 *n*개의 반출블록이 있을 경우, 해당 적치장에서 *n*개의 반출블록에 대해 가능한 모든 반출점과 경유점을 찾은 다음 이들에 대한 일련의

리스트를 구성하고 반출번호를 설정한다. 그리고 각 반출블록에 대해 가능한 모든 반출번호는 해당 반출블록의 반출경로 후보가 될 수 있다.

반출블록 / 반출번호	1	2	3	…	*n*-1	*n*
	2	3	10	…	5	2
반출경로 후보 리스트	1	2	3	…	5	2
	2	3	4	…	12	4
	5	10	5	…	21	5
	10	11	10	…		7
	21		20	…		11
	23					

반출번호	반출점 (x, y)	경유점 (x, y)
1	(2.1, 4.5)	(12.1, 5.4)
2	(11.3, 2.3)	(4.6, 7.5)
3	(34.2, 21)	(23.3, 3.2)
:	:	:
m	(9.3, 2.4)	(3.4, 32.3)

Figure 17. Representation of an individual for the moving route of blocks

3.2 타부 속성과 타부 리스트 구조

반출블록에 대해 하나의 반출경로를 나타내는 반출번호는 반출점과 경유점으로 구성되어 있다. 적치장에 *n*개의 반출블록에 대한 모든 반출경로에 반출점과 경유점을 나열하면 서로 같은 반출점과 경유점이 있다. 적치장을 기준으로 각 반출점과 경유점을 정리하면 전체 반출경로 대안에는 *k*개의 반출점과 *h*개의 경유점이 가능하다. 임의의 반출블록의 반출경로를 나타내는 반출번호를 변경에 따라 반출점과 경유점이 변경될 수 있으므로 두 점에 대한 각각의 변경을 타부속성으로 고려한다. *n*개의 반출블록에 대한 반출점과 경유점의 변경을 반영한 타부 리스트의 구조는 <Figure 18>과 같다.

반출점 \ 블록	1	2	…	*n*
(x_{e1}, y_{e1})				
(x_{e2}, y_{e2})				
:				
(x_{ek}, y_{ek})				

경유점 \ 블록	1	2	…	*n*
(x_{v1}, y_{v1})				
(x_{v2}, y_{v2})				
:				
(x_{vh}, y_{vh})				

Figure 18. Tabu list for deployment and pass through points of blocks

3.3 이웃해 생성

반출블록들에 대한 각각의 반출경로를 나타내는 현재해로부터 이웃해는 <Figure 19>와 같이 임의의 반출블록에 대해 후보 리스트에서 현재 반출번호를 제외한 나머지 후보 중에서 하나를 선택하여 이웃해를 생성한다.

반출블록	1	2	3	4	…	*n*-1	*n*
현재해	2	3	10	21	…	5	2
이웃해	**5**	3	10	21	…	5	2
	2	**2**	10	21	…	5	2
	2	3	**12**	21	…	5	2
	:	:	:	:	:	:	:
	2	3	10	21	…	5	**4**

Figure 19. Method of creating neighborhoods

조합 최적화 문제에서 유전자 및 타부서치 알고리즘을 적용할 때, 문제의 특성에 맞게 이웃해를 생성하는 것이 매우 효과적임을 여러 연구(Aladag *et al*., 2009; Eksioglu *et al*., 2008; Gao *et al*., 2007; Xu *et al*., 2006)에서 입증하였다. 이런 취지로 본 연구에서도 블록 반출문제의 특성을 반영한 이웃해 생성 방법을 제안한다.

적치장에서 다수 블록을 반출할 때, 많은 블록을 공동 반출경로를 통해 반출하는 것은 간섭블록을 줄이는 중요 방안 중의 하나이다. 따라서 이웃해를 생성할 때, 여러 반출블록들이 공동의 반출경로를 갖도록 이웃해를 생성하는 방안을 제안한다. 이에 대한 첫 번째 방법으로 전체 반출점에 대해 각 반출점으로 반출 가능한 반출블록의 수를 비교하여 반출 가능한 블록이 많은 반출점을 이웃해로 선정될 확률을 높여 준다. 이를 위한 방법으로 반출후보에서 반출번호를 선택하기 위해 토너먼트를 실시한다. 토너먼트 방식은 미리 정한 개수만큼 반출번호 후보를 뽑고, 이 중에서 반출 가능한 블록이 가장 많은 반출점이 포함된 반출번호를 선택한다. 예를 들면, <Figure 20>과 같은 적치장에서 반출블록 A, B, C를 반출하고자 할 때, 각 반출점으로 반출 가능한 블록은 <Table 1>과 같다. 그리고 반출블록 A의 반출경로의 후보는 <Table 2>와 같다. 현재해에서 블록 A의 반출번호를 변경할 때 토너먼트의 대상으로 반출경로 '1', '6', '9'가 선택되었다면, 각 반출번호의 반출점에서 반출 가능 블록 수는 각각 3, 1, 2개이다. 따라서 반출 가능한 블록이 가장 많은 반출번호 1번을 블록 A의 반출경로로 선택한다.

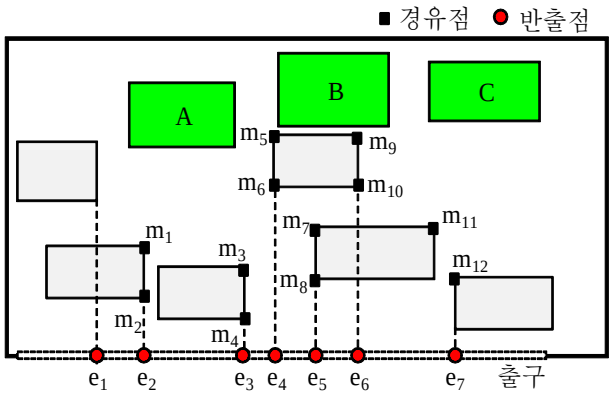

Figure 20. Example of an arrangement of blocks to explain the creation of neighborhood for using a common moving route

Table 1. Number of blocks for each deployment point in <Figure 20>

반출점	e_1	e_2	e_3	e_4	e_5	e_6	e_7
반출가능 블록	A, B, C	B, C	B, C	A	A	C	A, B

Table 2. Alternatives of moving route for Block A in <Figure 20>

반출번호	1	2	3	4	5	6	7	8	9	10	11	12	13
반출점	e_1	e_4	e_4	e_4	e_5	e_5	e_5	e_5	e_7	e_7	e_7	e_7	e_7
경유점	-	-	m_1	m_3	-	m_1	m_3	m_6	-	m_1	m_3	m_6	m_8

3.4 탐색 다양화

일반적으로 진화적 기법의 탐색 알고리즘을 적용할 때, 탐색 세대 수가 증가함에 따라 해의 개선 가능성은 점점 낮아진다. 본 절에서는 이러한 점을 보완하기 위한 방안으로 탐색 다양화 방법을 소개한다. 구체적으로 타부서치 알고리즘을 통해 반출경로의 조합을 찾는 과정에서 다양한 탐색을 위해 일정 세대 수가 지나도 해의 개선이 없으면 반출블록들의 반출경로 후보를 재구성하는 방법을 적용한다.

타부서치 알고리즘으로 반출블록들의 반출경로를 찾을 때, 각 반출블록에 대해 초기에는 정해진 몇 개의 반출점만 고려하여 반출경로 후보를 구성한다. 그런 다음에 탐색을 하면서 각 반출블록에 대해 현재 후보에 포함되지 않은 반출점도 반출경로 후보에 포함될 수 있도록 한다. 그리고 반출블록의 반출경로 후보를 변경하는 방법은 타부서치 알고리즘을 통해 반출경로의 조합을 탐색하면서 현재까지 찾은 가장 좋은 각 반출블록의 반출번호에 포함되어 있는 반출점으로 반출블록들의 반출경로 후보를 재구성하는 것이다.

예를 들어, <Figure 20>의 적치장에서 반출블록 A, B, C에 대해 2개만의 반출점(A : 'e_1, e_4', B : 'e_3, e_7', C : 'e_3, e_6')으로 반출경로 후보를 구성할 경우 블록별 반출경로 후보를 <Table 3>과 같이 구성하였다. 반출블록들의 반출경로 조합을 탐색하던 중에 현재 블록 A, B, C의 각 반출점이 'e_1', 'e_7' 'e_3'이라고 하면, 이때 반출블록들의 반출경로 후보를 재구성하는 방법은 각 반출블록에 대해 반출점 'e_1', 'e_7' 'e_3'가 포함되어 반출번호로 반출경로 후보를 교체하는 것이다. 즉, 반출블록 A, B, C 각각에 대해 'e_1, e_7', 'e_3, e_7', 'e_1, e_3'을 포함하고 있는 반출번호로 반출경로 후보를 재구성한다. 각 블록에 대한 반출경로 후보를 재구성하고 나면, 블록 A와 C의 반출경로 후보는 이전과 달리 변경된다. 이와 같은 방법으로 반출블록에 대한 반출경로 후보를 재구성하면 다양한 반출점으로 블록 반출을 고려할 수 있다.

본 연구에서는 적치장에서 다수 블록을 반출할 때, 간섭블록 수를 최소화하는 반출경로 조합을 찾기 위해 초기에는 반출블록의 반출경로 후보를 거리가 가까운 몇 개의 반출점으로 구성한다. 그리고 미리 정해진 탐색 세대 이내에 해의 개선이 없으면, <Figure 21>과 같이 그 때까지의 가장 좋은 해를 구성하는 반출점이 포함된 반출번호로 반출블록의 반출경로 후보를 재구성하여 반출경로에 대한 탐색을 시도한다. 반출블록들의 반출경로 조합에 대한 탐색에서 이러한 탐색 다양화 방법을 통해 보다 개선된 해를 찾을 수 있는 가능성을 높일 수 있다.

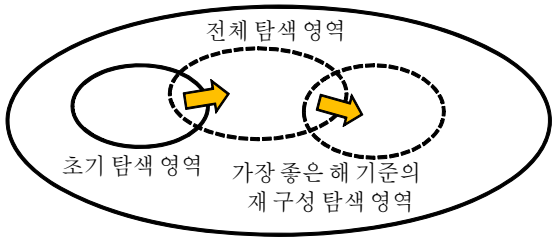

Figure 21. Diversification strategy : changing a search space from the best-so-far solution

3.5 타부서치 구조

다수 반출블록에 대해 간섭블록 수를 최소화하는 반출계획을 수립하기 위한 타부서치 알고리즘의 적용 절차는 <Figure 22>와 같다. 먼저, 이웃해에 대한 적응도는 간섭블록 수, 간섭블록들의 무게 합 또는 간섭블록 수와 간섭블록들 무게의 가중치 평균으로 평가한다. 임의의 이웃해가 타부 기간 중이지만 현재까지 가장 좋은 해보다 개선되었다면 이를 현재해로 선택하는 열망조건(Aspiration criterion)을 적용한다. 미리 정해진 세대 수가 지나도록 해의 개선이 없으면 현재까지 가장 좋은 해를 기준으로 반출블록들의 반출경로 후보를 재구성하고 변경된 반출경로 후보를 대상으로 반출경로의 조합을 찾는다. 반출블록에 대한 반출경로를 결정하는 반출점과 경유점 각각에 대한 변경사항을 타부 속성으로 정하고 이를 타부 리스트에 기록한다.

Table 3. All of the deployment and pass through points for each block planned to transfer in <Figure 20>

블록 A		블록 B		블록 C	
반출점	경유점	반출점	경유점	반출점	경유점
e_1		e_1	m_2, m_4, m_5, m_7	e_1	m_2, m_4, m_9, m_{10}, m_{11}, m_{12}
e_4	m_1, m_3	e_2	m_4, m_5, m_7	e_2	m_4, m_9, m_{10}, m_{11}, m_{12}
e_5	m_1, m_3, m_6	e_3		e_3	m_{10}, m_{11}, m_{12}
e_7	m_1, m_3, m_6, m_8	e_7		e_6	

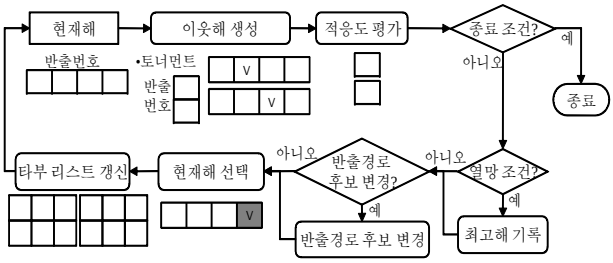

Figure 22. Structure of the tabu search algorithm for deployment planning of blocks

4. 수치 실험

본 장에서는 임의의 적치장에서 다수 반출블록에 대해 간섭블록 수를 최소화하는 반출계획을 수립하기 위해 본 연구에서 제안한 타부서치 알고리즘에 대한 수치실험을 소개한다. 실험은 <Figure 23>과 같이 적치장의 크기, 형상 및 출구가 다르고 배치된 블록 수가 각각 50, 53개인 두 적치장을 대상으로 실시하였다. 실험을 위해 각 적치장에 배치된 블록 중에서 임의로 20~25%를 반출블록으로 설정하여 50개의 반출 문제를 만들었다.

Figure 23. Two storage yards for the experiment of the deployment planning of blocks

먼저, 타부서치 알고리즘에서 반출점과 경유점의 타부 기간을 설정하기 위한 사전 실험으로 반출점과 경유점의 타부 기간을 (10, 10), (10, 15), (15, 10), (15, 15)로 설정하고, 50개의 문제에 대해 각 문제를 100세대를 반복하여 간섭블록 수가 가장 적었던 문제 횟수를 타부 기간별 정리하여 <Table 4>와 같은 결과를 얻었다. 그 결과 반출점과 경유점의 타부 기간이 (10, 10)일 때 결과가 가장 좋았다.

Table 4. Results of a preliminary experiment for tabu tenure of deployment and pass through points

적치장 \ 타부기간(반출점, 경유점)	(10, 10)	(10, 15)	(15, 10)	(15, 15)
①	28	19	19	17
②	23	20	13	15
합계	51	39	32	32

블록 반출계획에 대한 수치 실험에서는 반출블록들의 공동 반출경로 활용을 높이기 위한 이웃해 생성과 반출블록들의 반출경로 후보 구성 변경에 대한 효과를 실험하였다. 이를 위해 각 반출블록의 반출경로 후보를 자신의 위치에서 가까운 반출점 3, 5, 7, 9, 11, 13개로부터 구성하였을 때, 전체 간섭블록의 수를 분석하였다. 정해진 반출점 수로 반출경로 후보를 구성할 때, 블록별 평균 반출경로 후보 수는 <Table 5>와 같다.

Table 5. Average number of moving route alternatives from limited t deployment points

적치장 \ 반출점 수	3	5	7	9	11	13
①	7.8	18.3	33.9	56.3	73.3	104.3
②	8.5	19.4	34.7	57.8	69.7	110.6

실험은 타부서치 알고리즘에서 다음의 두 가지 방법을 각각 적용하여 실시하였다.

1) 일반 : 이웃해 생성을 각 반출블록에 대해 반출경로 후보 중에서 임의로 선택하여 변경(<Figure 19>)
2) 제안 : 본 연구에서 제안된 이웃해 생성 방법과 반출경로 후보 재구성 적용

실험은 각 문제에서 반출점과 경유점의 타부기간을 (10, 10)으로 설정하여 700세대까지 실시하였다. 블록별 평균 반출경로 후보 수에 따른 적치장별 50개 문제의 전체 간섭블록 수의 합은 <Figure 24>와 같다.

<Figure 24>에서와 같이 타부서치 알고리즘 적용시 일반적인 탐색 방법보다 본 연구에서 제안된 이웃해 생성과 반출경로 후보 구성을 변경할 때 간섭블록 수가 더 적은 반출방법을

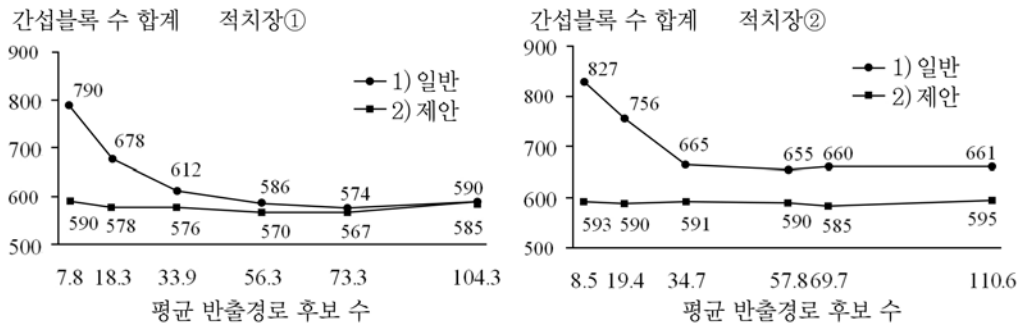

Figure 24. Results of the experiment for the proposed tabu search algorithm

찾을 수 있다는 것을 알 수 있다. 그리고 본 연구에서 제안된 타부서치 알고리즘은 반출경로 후보 수의 변화에 민감하지 않음을 보여 준다. 이것은 각 반출블록의 초기 반출경로 후보 수는 적었지만 탐색 중에 반출블록의 반출경로 후보를 재구성하여 다양한 탐색을 수행했기 때문이라고 판단된다. 그리고 <Figure 24>에서 알 수 있듯이 간섭블록 수를 최소화하는 반출방법을 찾는 실험에서 각 블록의 반출경로 후보 수를 약 70개 정도(10개의 반출점 사용)로 구성하여 본 연구에서 제안한 타부서치 알고리즘을 적용하는 경우의 결과가 가장 좋았다.

5. 결론 및 향후 연구

본 연구에서는 조선 야드 적치장에서 블록배치를 좌표기반으로 고려한 블록 반출계획 문제를 다루었다. 적치장에서 블록을 반출하기 위해 블록배치 상태를 고려하여 출구에서의 반출점과 경유점을 기준으로 블록을 이동하는 방법과 이에 따라 반출경로를 설정하는 방법을 제안하였다. 그리고 반출블록이 주어진 반출경로를 통해 이동할 때 발생하는 간섭블록을 설정하는 방법을 소개하였다. 적치장에서 다수 블록을 반출할 때, 간섭블록을 최소화하는 반출계획을 수립하기 위한 타부서치 알고리즘을 적용하는 방법을 제안하였다. 타부서치 알고리즘에는 반출블록들이 공동 반출경로를 갖도록 유도하는 이웃해 생성 방법과 반출경로 탐색 중에 반출블록의 반출경로의 후보를 변경하는 방법을 제안하였고, 제안된 방법의 성능은 수치 실험을 통해 검증하였다.

향후 연구로는 적치장에서 직선 이동 이외에 다양한 블록 이동 방법을 고려하여 블록 반출에 대한 현실성을 높일 필요가 있다. 아울러 다수 블록 반출계획에서의 타부서치 알고리즘 외의 다른 메타 휴리스틱의 추가 적용으로 블록 반출을 위한 보다 효율적인 방법에 대한 연구가 필요하다. 임의의 적치장에서 반출된 블록은 정해진 목적지 적치장으로 옮겨진다. 이 때, 해당 적치장으로 반입되는 블록을 어떻게 배치하는지에 따라 해당 적치장의 미래 반출계획에 영향을 미친다. 따라서 적치장으로 반입되는 블록의 적치위치를 결정할 때 추후 반출될 블록들을 고려하여 간섭이 되지 않는 곳에 적치할 필요가 있다. 즉, 적치장 출입구의 위치와 추후 반출될 블록들의 반출경로 등을 고려한 반입블록의 적치 위치 결정에 대한 연구가 필요하다.

참고문헌

Aladag, C., Hocaoglu, G., and Basaran, M. (2009), The effect of neighborhood structure on tabu search algorithm in solving course timetabling problem, *Expert Systems with Applications*, **36**, 12349-12356.

Bazzazi, M., Safaei, N., and Javadian, N. (2009), A genetic algorithm to solve the storage space allocation problem in a container terminal, *Computers and Industrial Engineering*, **53**(1), 44-52.

Eksioglu, B., Eksioglu, S., and Jain, P. (2008), A tabu search algorithm for the flowshop scheduling problem with changing neighborhoods, *Computers and Industrial Engineering*, **54**, 1-11.

Gao, M., Gen, M., Sun, L., and Zhao, X. (2007), A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems, *Computers and Industrial Engineering*, **53**, 149-162.

Glover, F. (1989), Tabu search-Part I. *ORSA Journal on Computing*, **1**, 190-206.

Kim, K. and Hong, G. (2006), A heuristic rule for relocating blocks, *Computers and Operations Research*, **33**, 940-954.

Koh, S., Park, J., Choi, Y., and Joo, M. (1999), Development of a Block Assembly Scheduling System for Shipbuilding Company, *IE interfaces*, **12**(4), 586-594.

Kozan, E. and Preston, P. (2006), Mathematical modeling of container transfers and storage locations at seaport terminals, *OR Spectrum*, **28**, 519-537.

Lee, K., Lee, J., and Choi, S. (1996), A spatial scheduling system and its application to shipbuilding : DAS-CURVE, *Expert Systems with Applications*, **10**, 311-324.

Lee, K., Lee, J., Park, H., Hong, J. and Lee, S. (1997), Developing scheduling systems for Daewoo Shipbuilding : DAS project, *European Journal of Operational Research*, **97**, 380-395.

Park, C. and Seo, J. (2009a), Genetic Algorithm of Planar Storage Location Assignment Problem, *Journal of the Korean Institute of Industrial Engineers*, **35**(2), 129-140.

Park, C. and Seo, J. (2009b), Mathematical modeling and solving procedure of the planar storage location assignment problem, *Computers and Industrial Engineering*, **57**, 1062-1071.

Park, C. and Seo, J. (2010), Comparing heuristic algorithm of the planar storage location assignment problem, *Transportation Research Part E*, **46**, 171-185.

Xu, J., Sohoni, M., McCleery, M., and Bailey, T. (2006), A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems, *European Journal of Operational Research*, **169**, 978-993.

Zhang, C., Liu, J., Wan, Y., Murth, K., and Linn, R. (2003), Storage space allocation in container terminals, *Transportation Research Part B*, **37**, 883-903.