

자동화 장치장의 재정돈 계획 최적화를 위한 협력적 공진화 알고리즘

박기역* · 박태진** · 류광렬†

*,**부산대학교 컴퓨터공학과, † 부산대학교 정보컴퓨터공학과 교수

A Cooperative Coevolutionary Algorithm for Optimizing Remarshaling Plan in an Automated Stacking Yard

Kiyeok Park* · Taejin Park** · Kwang Ryel Ryu†

*,** Department of Computer Engineering, Pusan National University, Pusan 609-785, Korea

† Professor, Department of Computer Science & Engineering, Pusan National University, Pusan 609-785, Korea

요약 : 본 논문은 재정돈 계획의 최적화를 위해 협력적 공진화 알고리즘을 이용하는 방법을 제안한다. 재정돈이란 컨테이너 터미널에서 적하 작업 시 발생하는 지연을 줄이기 위해 선박에 적하될 컨테이너의 위치를 사전에 변경하는 작업이다. 재정돈 계획 수립을 위해서는 적하 시 작업 효율이 최대가 되고 재정돈 시간이 최소가 되도록 컨테이너가 재정돈 후 배치될 장치형태와 재정돈 시 컨테이너를 옮길 순서를 결정해야 한다. 협력적 공진화 알고리즘은 주어진 문제가 세부 문제들로 분할 가능할 때 분할된 세부 문제들을 동시에 탐색하여 문제를 효율적으로 해결하는 방법이다. 이에 본 논문에서는 재정돈 계획 문제를 장치형태 결정 문제와 이동 우선순위 결정 문제로 분할하고 협력적 공진화 알고리즘을 적용하여 재정돈 계획을 최적화하였다. 실험결과 문제를 분할한 협력적 공진화 알고리즘이 문제를 분할하지 않은 접근 방법에 비해 더욱 효과적으로 재정돈하는 계획을 수립함을 확인할 수 있었다.

핵심용어 : 컨테이너 터미널, 자동화 장치장, 재정돈, 최적화, 협력적 공진화 알고리즘

Abstract : In this paper, we propose optimizing a remarshaling plan in an automated stacking yard using a cooperative coevolutionary algorithm (CCEA). Remarshaling is the preparation task of rearranging the containers in such a way that the delay are minimized at the time of loading. A plan for remarshaling can be obtained by the following steps: first determining the target slots to which the individual containers are to be moved and then determining the order of movement of those containers. Where a given problem can be decomposed into some subproblems, CCEA efficiently searches subproblems for a solution. In our CCEA, the remarshaling problem is decomposed into two subproblems: one is the subproblem of determining the target slots and the other is that of determining the movement priority. Simulation experiments show that our CCEA derives a plan which is better in the efficiency of both loading and remarshaling compared to other methods which are not based on the idea of problem decomposition.

Key words : Container Terminal, Automated Stacking Yard, Remarshaling, Optimization, Cooperative Coevolutionary Algorithms

1. 서 론

컨테이너 터미널의 전체 생산성은 컨테이너들을 배에 싣는 적하 작업의 효율에 의해 크게 좌우된다. 적하 작업의 효율은 터미널 내부에 임시로 컨테이너를 쌓아두는 공간인 장치장에 컨테이너가 쌓여 있는 형태에 크게 영향을 받는다. 컨테이너들이 적하될 때에는 선박의 안정성과 중간 기착항에서의 작업 용이성 등을 고려하여 정해진 순서를 따르게 된다. 만약 이번 적하할 컨테이너 상단에 다른 컨테이너들이 쌓여있다면 그것들을 다른 곳으로 치우는 추가 작업이 발생하여 적하 작업이 지연된다. 또한, 크레인도 멀리 이동해야 할 위치에 컨테이너가 쌓여 있을 경우에도 작업 시간이 길어질 수 있다.

수출용 컨테이너는 대개 약 1, 2주의 기간에 걸쳐 반입되어

장치장에 임시로 저장되어 있다가 적하된다. 이 기간 동안에는 여러 선박에 실릴 여러 종류의 컨테이너가 반입되고 장치장의 용량이 제한되어 있어 아무리 적하 작업의 효율을 미리 고려하여 쌓으려 하더라도 실제로 이상적인 형태로 쌓기가 어렵다. 뿐만 아니라, 반입 시에는 컨테이너의 무게도 정확히 모르고 정확한 적하 순서도 결정되어 있지 않다.

이러한 문제의 해결을 위한 한가지 방법으로 기존 연구들은 재정돈 작업을 제안하였다(강 등, 2006; 오 등, 2006; Kang et al., 2006; Hirashima et al., 2006; Kim와 Bae, 1998; Lee와 Hsu, 2007; Zhang et al., 2007). 재정돈이란 적하 작업의 효율을 최대화하기 위하여 적하 작업이 시작되기 전에 컨테이너들의 장치 위치를 미리 변경해 두는 작업을 말한다. 기존 연구들은 다양한 접근 방법을 통해 재정돈 문제를 해결했다. 특히 Kang et

* 정회원, nerissa79@pusan.ac.kr, 051) 510-3531

** 정회원, parktj@pusan.ac.kr, 051) 510-3531

† 교신저자 : 류광렬(정회원), kr Ryu@pusan.ac.kr 051) 510-2453

al.(2006)은 재정돈 계획 문제를 적하 작업 시 재취급이 발생하지 않으며 재정돈 시간 역시 단축되도록 하는 재정돈 후의 컨테이너 장치위치를 결정하는 문제와 재정돈 시간을 줄이는 컨테이너의 이적 순서를 결정하는 문제로 나누었다. 그 후, 장치위치를 잘 결정하는 것이 재정돈 작업 시간과 적하 작업 시간을 모두 줄이는 데 더욱 도움이 된다는 판단 하에 장치위치를 주로 탐색하며 후보 장치위치를 평가하기 위해 컨테이너의 이적 순서를 탐색하도록 하였다. 그러나 이 기존 연구는 비자동화 장치장의 재정돈 문제를 해결한 것으로 본 논문이 대상으로 하고 있는 무인 장비를 사용하는 자동화 장치장과 배치 방식이 다르고 그로 인해 재정돈 작업을 수행하기 위해 고려해야 하는 요소에도 차이가 있다.

본 논문은 자동화 장치장의 재정돈 작업 계획 최적화를 위해 협력적 공진화 알고리즘 (Husband와 Mill, 1991) (Cooperative CoEvolutionary Algorithms, CCEA)을 이용하는 방안을 제안한다. 주어진 문제가 간단한 세부 문제들로 분리 가능한 문제인 경우, 협력적 공진화 알고리즘은 간단한 세부 문제들로 나누어 줄어든 탐색 공간에서 각 세부 문제별로 해 집단을 구성하여 동시에 진화시킴으로써 탐색의 효율을 향상시키고자 하는 방법이다(Wiegand, 2003). 본 논문에서 제안하는 방안은 Park et al.(2009)의 연구와 같이 재정돈 계획 문제를 두 가지 세부 문제로 분할하여 CCEA를 적용하였다. 그리고 자동화 장치장의 특성을 고려하여 탐색으로 찾아진 해를 재정돈 계획으로 해석하는 휴리스틱 방안을 제안한다. 시뮬레이션을 통한 실험 결과, 문제를 분리하지 않고 일반적인 최적화 알고리즘을 사용한 경우보다 문제를 분리하여 CCEA를 사용한 경우가 적하 소요시간 면에서나 재정돈 작업 자체의 소요시간 면에서 모두 더 우수한 계획을 생성해 주는 것으로 확인되었다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 자동화 장치장의 재정돈 계획 문제를 보다 구체적으로 소개하고, 3장에서는 CCEA를 이용한 재정돈 계획 최적화 방안을 상세히 설명한다. 4장에서는 시뮬레이션을 통한 여러 가지 실험 결과를 분석하고, 5장에서는 결론을 맺는다.

2. 자동화 장치장의 재정돈 문제

Fig. 1은 본 논문에서 대상으로 하는 자동화 컨테이너 터미널

을 보이고 있다. 컨테이너 터미널은 선박 작업을 수행하는 안벽, 컨테이너를 임시로 적재해두는 장치장, 외부에서 들어온 차량이 다니는 배후지의 세 부분으로 구분할 수 있다. 안벽에서 수행되는 선박과의 작업은 컨테이너 크레인(CC)가 담당하고, 장치장에서의 작업은 자동화 장치장 크레인(ASC)이 담당한다. 장치장과 안벽 사이의 컨테이너 이송은 무인 이송 차량(AGV)이 담당한다. 자동화 장치장에는 블록들 수십 개가 안벽에 대해 수직방향으로 배치되어 있고, 각 블록마다 2 대의 ASC가 설치되어 하나는 해측 작업을 그리고 다른 하나는 육측 작업을 담당하게 한다. 하나의 블록은 수십 개의 베일로 구성되어 있고, 한 베이는 정해진 단 수만큼의 컨테이너를 수직으로 쌓은 여러 개의 스택으로 이루어져 있다. 본 논문에서 대상으로 하고 있는 터미널은 5단 10열 41베이의 구조로 이루어져 있다.

컨테이너 터미널에서는 여러 종류의 작업이 다수의 장비의 연계를 통해 수행된다. 터미널의 작업에는 크게 장치장에 쌓여 있는 컨테이너를 선박에 싣는 적하 작업, 선박에서 컨테이너를 내려 장치장에 쌓는 양하 작업, 외부로부터 들어온 컨테이너를 장치장에 쌓는 반입 작업, 장치장에 쌓여있는 컨테이너를 꺼내 외부로 보내는 반출 작업이 있다. 안벽 쪽에서 수행되는 양하/적하 작업은 CC, ASC, AGV가 연계되어 작업을 수행하고, 배후지 쪽에서 수행되는 반입/반출 작업은 ASC와 외부트럭이 연계되어 작업을 수행한다.

컨테이너 터미널의 생산성은 가장 중요한 장비인 CC에 의해 이루어지는 적하 작업의 효율에 의해 크게 좌우된다. 적하 작업은 ASC가 컨테이너를 블록에서 꺼내어 AGV에 전달하고, AGV가 다시 CC에게 전달하여 CC가 선박에 싣는 것으로 종료된다. 따라서 블록에서 컨테이너를 꺼내는 ASC 또는 CC에게 컨테이너를 전달하는 AGV의 작업에 지연이 발생하면 자연히 CC의 작업도 지연된다. 재정돈 작업은 이 중 ASC를 지연시킬 가능성이 있는 요소를 미리 제거하여 적하 작업의 효율을 향상시키고자 하는 것이다.

수직으로 배치된 장치장에서 적하 시 ASC 작업이 지연되는 경우는 크게 2가지가 있다. 첫 번째는 ASC가 원거리를 이동해야 하는 경우이다. 외부 트럭에 의해 반입되는 수출용 컨테이너는 주로 블록의 육측 가까이 장치된다. 수출 컨테이너들을 반입 시 바로 해측 가까이 장치하려고 할 경우 육측 ASC와 해측 ASC 사이에 간섭이 발생하여 해측 ASC의 작업을 방해할 수

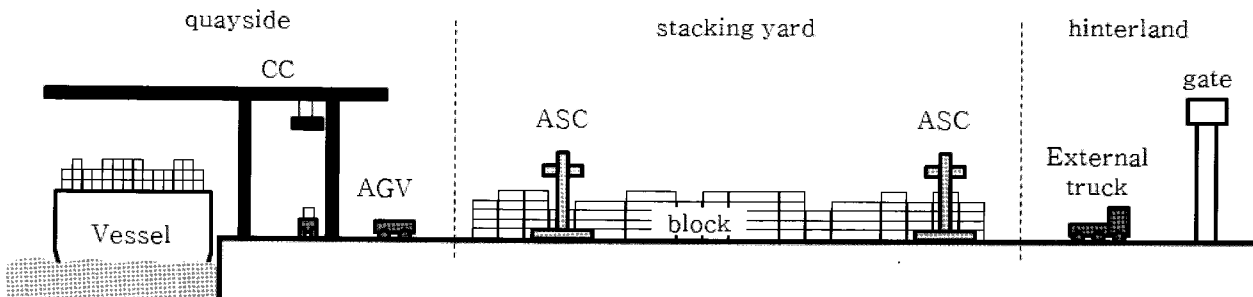


Fig. 1 The layout of the target container terminal

있다. 해측 ASC는 주로 선박 작업과 연계된 작업을 수행하므로 이를 방해할 경우 터미널의 전체 생산성을 저해할 수 있으므로 위험하다. 따라서 반입 시에는 해측 ASC가 작업을 수행 중이라면 흔히 안전하게 육측 가까이 컨테이너를 장치하게 되고 이럴 경우 적하 시 해측 ASC가 이동해야 하는 거리가 길어질 수밖에 없다. 두 번째는 적하 순서에 맞지 않게 컨테이너가 적체되어 컨테이너를 꺼내기 위해 다른 컨테이너를 치우는 재취급 작업이 발생하는 경우이다. 앞에서 언급하였듯이 반입 시 공간의 제약이나 정보의 부정확성, 작업상의 제약 등이 장치 형태를 처음부터 이상적으로 가져가지 못하게 하는 원인이 된다. 특히, 컨테이너의 정확한 무게는 적하 시점에 상당히 압박해야 알 수 있기 때문에 재취급 상황이 발생하는 경우가 많다. 재정돈 작업은 크레인의 유휴 시간을 이용하여 육측에 가까이 있는 컨테이너들을 해측 가까이 옮기고 적하 순서를 고려하여 컨테이너들의 적치 상태를 바꾸어 줌으로써 적하 작업 시 ASC에 의한 지연을 최대한 줄일 수 있게 한다.

3. 재정돈 계획 수립

재정돈 작업을 통해 달성하고자 하는 목표는 적하 작업 시 장치장의 크레인 작업 지연 단축이다. 앞서 설명한 바와 같이 적하 작업이 터미널의 생산성을 좌우하는 작업이고 크게 두 가지 원인으로 인해 적하 시 장치장 크레인의 작업이 지연될 수 있기 때문에 그를 미리 방지하고자 재정돈 작업을 수행한다. 그러나 재정돈 작업도 터미널의 자원을 소모하는 작업이므로 무한정 시간을 투자할 수 없다. 재정돈 계획은 적하 시 크레인이 최대의 성능으로 작업할 수 있게 하고 재정돈 작업도 최단 시간에 끝낼 수 있도록 수립되어야 한다.

본 논문에서는 재정돈 계획 수립을 위해 Park et al.(2009)의 연구와 같이 휴리스틱으로 재정돈 목적지 후보들을 선택한 후 협력적 공진화 알고리즘을 이용하여 컨테이너들의 구체적인 장치위치와 작업 순서를 결정한다. Park et al.(2009)의 연구에서 재정돈 목적지 후보를 선택하는 방법은 다음과 같다. 우선 임의의 순서로 나열된 CC의 리스트인 Q 에 속한 각 CC q 에 대해 CC q 가 적하해야 할 컨테이너의 수 n_q 를 구한다. n_q 는 컨테이너의 수이면서 재정돈 목적지 후보를 선택하며 확보해야 하는 컨테이너를 쌓을 수 있는 공간의 수이다. 기존 연구에서는 재정돈 시 재취급 작업의 발생을 피하기 위해 실제 컨테이너 수보다 10% 많은 공간을 선택하도록 하였다. 그 후 CC의 리스트 Q 에 나열된 순서대로 각 CC q 에 대해 재정돈 목적지 후보가 될 수 있는 스택을 하나씩 선택하여 CC q 의 재정돈 목적지 후보 집합인 T_q 에 추가한다. 서로 다른 CC의 컨테이너를 같은 스택에 쌓게 되면 적하 작업 시 재취급이 발생할 수 있으므로 이를 사전에 방지하기 위해 각 CC 별로 목적지 후보를 선택하게 하였다. 재정돈 목적지 후보로는 하나 이상의 빈 슬롯이 있거나 반출 컨테이너가 재취급 없이 꺼낼 수 있는 위치에 있는 스택을 선택한다. 이 때, T_q 에 포함된 스택들의 컨테이너를 쌓을 수 있는 슬롯들의 총 수가 n_q 이상이면 CC q 를 위해서는 더 이상

스택을 선택하지 않는다. Q 내의 마지막 CC의 재정돈 목적지 후보 집합에 스택이 추가되고 나면 다시 Q 의 첫 번째 CC로 되돌아가 이 과정을 반복한다. 각 CC의 T_q 의 사용 가능한 슬롯의 수가 n_q 이상이 될 때까지 과정을 반복한다.

컨테이너들의 구체적인 장치위치와 작업 순서 결정 시에는 적하 효율뿐만 아니라 재정돈 시 작업 효율까지 고려하여 가능한 짧은 시간에 재정돈을 완료할 수 있는 작업 계획을 수립하여야만 한다. 이 과정에서 CCEA는 장치위치와 작업 순서를 각각 서로 다른 집단을 이용해 진화시키며, 해의 평가 시 상대 집단의 정보를 이용하는 과정을 통해 두 집단이 상호작용을 하게 된다.

3.1 CCEA를 이용한 목표 위치 및 이동 우선순위 결정

협력적 공진화 알고리즘은 주어진 문제가 세부 문제들로 분할 가능할 때 주어진 문제를 세부 문제들로 분할하고 각 세부 문제의 해를 별도의 집단을 통해 진화시킨다. 만약 주어진 문제를 완전히 독립된 세부 문제들로 나눌 수 있다면 각 세부 문제들을 별도의 문제로 보고 따로 해결할 수 있을 것이다. 최종적인 해는 별도로 해결된 세부 문제들의 결과를 종합하여 얻을 수 있다. 그러나 실세계 문제들 중 완벽히 독립적인 세부 문제들로 분해되는 문제는 거의 없다. 주어진 문제를 세부 문제로 나누었다고 하나 그 문제들이 서로 연관되어 있어 따로 해결할 수 없는 것이다. 이처럼 주어진 문제를 세부 문제들로 분해할 수는 있으나 그 세부문제들 간에 연관 관계가 있고 특히 각 세부 문제가 하나의 목표 달성을 위해 협력해야 하는 경우에 협력적 공진화 알고리즘은 효율적인 탐색을 수행한다. 각 세부 문제의 해를 별도의 집단을 통해 진화시키지만 세부 문제들이 독립적이지 않으므로 세부 문제들의 해를 따로 평가할 수 없다. 이 경우 각 세부 문제들의 상호 정보 교환을 통해 서로의 해를 종합하여 평가한다.

본 논문에서는 재정돈 작업 계획 문제를 효율적으로 해결하기 위해 대상 문제를 두 개의 세부 문제로 분해하여 각 세부 문제별로 별도의 해 집단을 구성하였다. 하나는 컨테이너가 재배치되어 쌓일 목표 위치를 결정하는 문제이고 하나는 재정돈 작업의 시간을 최소화 할 수 있는 컨테이너 이동 우선순위를 결정하는 문제이다.

먼저 목표위치 결정 문제와 이동 우선순위 결정 문제를 자세히 설명하기 전에 이후에 등장할 예시의 상황을 설명한다. 이후의 예시에 등장할 재정돈 대상 컨테이너는 Fig. 2(a)와 같은 형태로 육측 가까운 곳에 쌓여있다고 가정한다. 컨테이너 A1은 CC A의 적하 계획에 따라 CC A가 가장 먼저 적하할 컨테이너를 의미한다. 앞에서 설명한 재정돈 목적지 후보 선택 휴리스틱에 의해 Fig. 2(b)와 같은 스택이 선택되어 있다고 가정한다. 스택 A_1 , B_1 , C_1 은 각각 CC A가, CC B, CC C가 적하할 컨테이너를 재정돈 하여 쌓을 스택이다. 그 중 스택 C_1 은 반출 컨테이너 2개가 쌓여 있어 그 컨테이너를 비워내고 재정돈 컨테이너를 5개까지 쌓을 수 있다. 반출 작업은 블록의 육측 가장자리에서 수행되므로 반출될 컨테이너라면 육측 가까이에 있는 것이 좋

을 것이다. 그러므로 재정돈 목적지 후보 스택에 있는 반출 컨테이너를 재정돈을 통해 반출 컨테이너를 옮긴다.

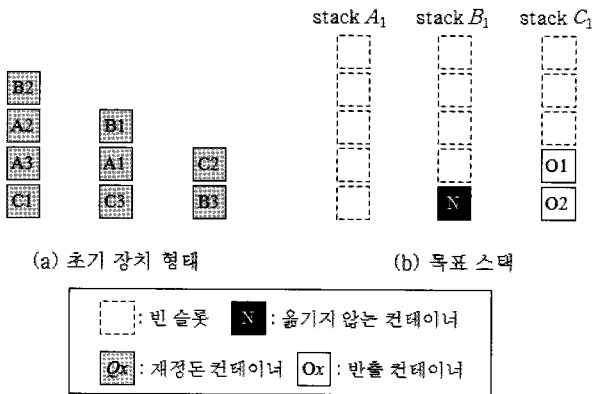


Fig. 2 The given situation for explaining the following example

목표 위치 결정 문제는 적하될 모든 컨테이너들이 앞 절에서 설명한 방법으로 결정된 재정돈 목적지 후보로 선택된 스택 내의 어느 슬롯으로 이동해야 하는지를 결정하는 문제이다. 이 문제는 적하 작업 수행 시 재취급이 적게 발생하며 재정돈 시간도 단축시키는 재정돈 후 컨테이너가 쌓일 위치를 결정하는 문제이다. 본 연구에서는 재정돈 컨테이너의 수보다 많은 재정돈 목적지와 이후에 설명할 해를 해석하는 과정에 의해 자칫 남은 공간을 탐색에 활용하지 못할 수가 있다. 남은 공간도 탐색에 활용할 수 있도록 그 공간의 수만큼 가상 컨테이너를 추가하되 각 CC 별로 재정돈 목적지 후보를 선택하였기 때문에 각 CC 별로 모자란 컨테이너의 수가 다르므로 가상 컨테이너도 각 CC 별로 모자란 컨테이너의 수만큼 추가한다. 이 문제를 위한 후보해(집단의 개체)는 Fig. 3(a)와 같이 재정돈 컨테이너와 가상 컨테이너를 임의의 순서로 나열하여 생성한다. Fig. 3(b)는 컨테이너 이동 우선순위를 결정 문제의 해 표현을 보이고 있다. 이 문제는 재정돈 작업을 최단 시간 내에 완료할 수 있도록 컨테이너들을 옮길 우선순위를 결정하는 문제로서 옮길 컨테이너를 임의의 순서로 나열함으로써 후보해를 생성한다. 옮길 컨테이너에는 재정돈 대상 적하 컨테이너뿐 아니라 재정돈 목적지 후보 내에 있는 치울 컨테이너까지 포함된다.

세부 문제의 후보해를 평가하려면 그 후보해가 나타내고 있는 정보가 좋은 재정돈 계획을 수립하는데 얼마나 도움이 되는 것인지를 판단할 수 있어야 한다. 그러나 협력적 공진화 알고리즘을 사용하기 위해 나눈 문제들은 독립이 아니므로 각 세부 문제들로부터 후보해들을 하나씩 모아서 그들의 정보를 종합해보아야 평가가 가능하다. 평가를 위해 모인 세부 문제의 후보해들을 흔히 조력자(collaborator)라 부른다. 사실 한 세부 문제의 한 후보해의 적합도는 어떤 조력자를 만나느냐에 따라 달라진다. 가장 적합한 조력자를 찾기 위해 모든 조력자 후보와 결합하여 평가를 할 경우 계산 부담이 지나치게 커지므로 흔히 CCEA는 유망해 보이는 조력자의 후보 집합을 별도로 유지하

여 그것들과만 조합을 하는 방식을 취하기도 한다(Wiegand와 Sarma, 2004; Panait et al. 2006). 본 논문에서는 상대 세부 문제의 후보해들 중 지금까지 가장 높은 평가값을 가진 것만 조력자로 선정함으로써 계산 부담을 최소화 하였다.

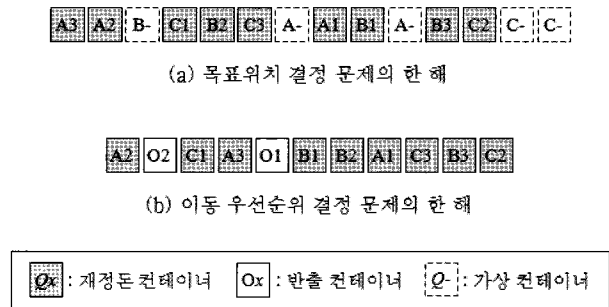


Fig. 3 Examples of candidate solutions(individuals) for the subproblems

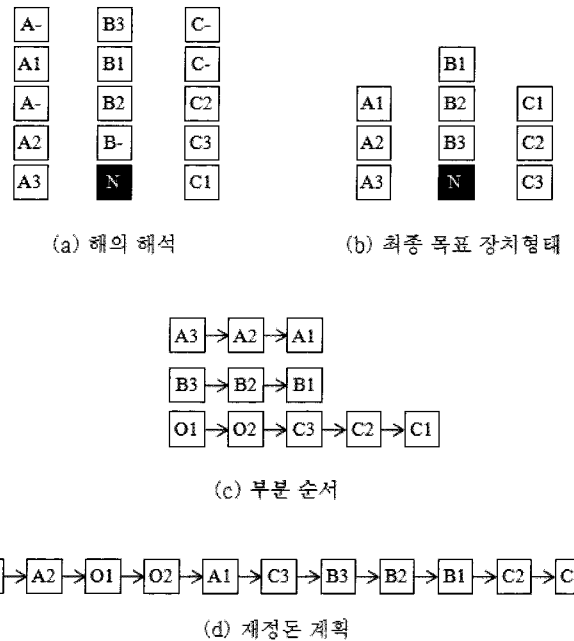


Fig. 4 An example of deriving a schedule from candidate solutions of subproblems

본 연구에서 후보해의 평가는 목표위치 결정 문제의 해를 해석하는 것에서 시작한다. 적하 작업 시 재취급 발생을 피하기 위해 먼저 Fig. 3(a)와 같은 평가 대상 해로부터 Fig. 4(b)와 같이 적하 작업 시 재취급이 발생하지 않는 목표 장치형태를 생성해야 한다. 이를 위해서는 Fig. 3(a)의 모든 컨테이너를 순차적으로 꺼내어 해당 컨테이너의 길이와 해당 컨테이너를 적하할 CC에 할당된 목표 스택에 하나씩 할당하면 Fig. 4(a)와 같이 해를 해석할 수 있다. 그 후, 각 스택에 할당된 컨테이너를 적하 계획에 따라 적하 순서의 오름차순으로 정렬하여 Fig. 4(b)와 같은 최종 목표 장치형태를 생성한다. 다음으로, 이 장치형태를 달성하기 위해 재정돈 작업을 수행하는 과정에서 재취급 발생

이 최소화 되도록 하는 부분 순서를 생성한다. 이는 각 스택별로 치위야 할 컨테이너부터 먼저 옮긴 다음 아래쪽에서 위쪽의 순서로 컨테이너를 옮기도록 함으로써 쉽게 생성될 수 있다. Fig. 4(c)의 부분순서의 예에서는 stack C_1 에 원래 반출 컨테이너가 있었기 때문에 그것들을 치우는 작업이 우선적으로 수행되어야 함을 보이고 있다. 이상과 같이 생성된 부분 순서로는 동일 스택 내에 장치될 컨테이너간의 순서만 정해질 뿐 서로 다른 스택에 장치될 컨테이너 간의 순서는 정해지지 않는다. 서로 다른 스택에 장치될 컨테이너들 간의 순서를 결정하기 위해서는 이 부분순서가 컨테이너 이동 우선순위 결정 문제의 한 해와 결합되어야 한다. 예를 들어 Fig. 3(b)의 해가 조력자로 선택되었다면 그 우선순위에 따라 Fig. 4(c)에 보인 A3, B3, O1 중에서 가장 앞선 A3이 먼저 옮겨져야 하고, 그 다음에는 다시 A2, B3, O1 중 가장 앞선 A2가 다음 순서가 된다.

컨테이너를 옮기는 순서가 결정되면 시뮬레이션을 통해 재정돈에 소요되는 총 작업시간을 추정할 수 있다. 재정돈 계획의 최종적인 평가는 그에 덧붙여 이 후 적하 시 예상되는 작업 시간까지 추정함으로써 완성된다. 본 논문에서 적하 시 예상되는 작업 시간은 ASC가 쉬지 않고 연속적으로 작업했을 경우의 총 작업 시간이다. 컨테이너가 안벽에 가까이 장치될수록 적하 작업의 지연을 줄일 수 있다는 가정 하에 적하 시 장치장 크레인의 예상 이동시간을 계산하여 평가에 반영하였다. 그러나 재정돈 시간을 줄이는 것과 적하 작업 시간을 줄이는 것 사이의 중요도는 재정돈 시간에 투자할 수 있는 시간에 따라 달라질 수 있다. 본 논문에서는 재정돈 시간과 적하 시 예상 크레인 이동 시간의 가중 합으로 계획을 평가하였다.

4. 실험

4.1 사용된 CCEA 알고리즘

CCEA는 각 집단의 진화 시기에 따라 병렬형(parallel model)과 순차형(sequential model)로 구분된다(Wiegand, 2003). 매 세대에 한 집단씩 교대로 평가하고 진화시키는 형태가 순차형이고 매 세대에 모든 세부 문제의 해 집단을 평가하고 모두 동시에 진화시키는 형태가 병렬형이다. 본 논문에서는 두 집단을 동시에 진화시키는 방법 중 세대형(generational) CCEA를 구현하여 실험하였다.

세대형 CCEA는 세대가 바뀌면 직전 세대의 개체가 집단 내에서 사라진다. 따라서 이 형태의 경우 세부 문제별로 직전 세대의 최적 개체를 별도로 유지하여, 현재 세대의 상대 개체군의 개체를 평가할 때 조력자가 되도록 하였다. Fig. 5는 본 논문에서 사용한 세대형 CCEA의 알고리즘을 보여준다. 먼저 3장에서 언급한 것과 같은 방법으로 슬롯 결정 문제와 우선순위 결정 문제용 초기 해집단 P_s 와 P_p 를 각각 생성한다. 다음으로 초기 해 집단의 평가를 위해 P_s 의 직전 세대 최적 개체 역할을 하는 c_s 와 P_p 의 직전 세대 최적 개체 역할을 하는 c_p 를 무작위로 생성하여 각각 P_p 와 P_s 의 초기 조력자가 되도록 한다. 그 다음에

는 조력자를 이용하여 각 해집단을 평가하고 그 중 최적 개체를 다음 세대의 조력자로 사용하기 위해 보관한다. 그리고는 유전연산을 적용하여 각 해 집단을 갱신하고 다시 평가 단계로 되돌아가 진화하는 과정을 반복한다.

```

algorithm pgCCEAforRemarshaling()
variables
   $P_s$  : the population for the target slot determination
   $P_p$  : the population for the movement priority determination
   $c_s$  : the collaborator for evaluating individuals of  $P_p$ 
   $c_p$  : the collaborator for evaluating individuals of  $P_s$ 
   $Fit[i]$  : fitness value of individual  $i$ 
   $sch$  : schedule for remarshaling
output the best-so-far schedule
begin
  Initialize-Population(  $P_p$  ) Initialize-Population(  $P_p$  )
   $c_p \leftarrow$  Select-Initial-Collaborator(  $P_p$  )
   $c_s \leftarrow$  Select-Initial-Collaborator(  $P_s$  )
  Until the termination criterion is met
    for each individual  $i_p$  in  $P_p$ 
       $sch \leftarrow$  Make-Schedule(  $i_p$  ,  $c_s$  )
       $Fit[i_p] \leftarrow$  Simulate-Schedule(  $sch$  )
    for each individual  $i_s$  in  $P_s$ 
       $sch \leftarrow$  Make-Schedule(  $c_p$  ,  $i_s$  )
       $Fit[i_s] \leftarrow$  Simulate-Schedule(  $sch$  )
   $c_p \leftarrow$  Select-Best-Individual(  $P_p$  )
   $c_s \leftarrow$  Select-Best-Individual(  $P_s$  )
   $P_p \leftarrow$  Apply-GeneticOperators(  $P_p$  )
   $P_s \leftarrow$  Apply-GeneticOperators(  $P_s$  )
end

```

Fig. 5 Algorithm for parallel generational CCEA

4.2 실험 환경

실험 대상 컨테이너 터미널의 환경은 다음과 같이 가정하였다. 터미널에는 한 척의 선박이 접안해 있고, 그 선박에 적하될 컨테이너는 7개의 블록에 흩어져 장치되어 있다. 한 블록은 41개의 베이로 이루어져있으며, 한 베이는 높이가 5인 스택 10개로 구성되어 있다. 장치장의 각 블록에는 육측과 해측에 각각 한 대씩 두 대의 크레인이 있지만 재정돈은 해측 크레인에 의해서만 수행되는 것으로 가정하였다. 적하할 컨테이너 1000개가 서로 다른 상태로 적재되어 있는 10가지 데이터 셋에 대해 실험하여 그 결과를 평균하였고, 데이터 셋들의 컨테이너 데이터는 기존 터미널에 대한 통계 자료를 기반으로하여 생성되었다. Fig. 6은 1000개의 적하 컨테이너가 장치되어 있는 10가지 데이터 셋의 평균적인 컨테이너 분포를 보여준다. 재정돈 계획의 시뮬레이션을 위해서 부산 신항만 터미널의 2-3단계 부두에 사용된 장비의 동작을 모사하여 작업하는 시뮬레이터를 제작하여 실험을 수행하였다.

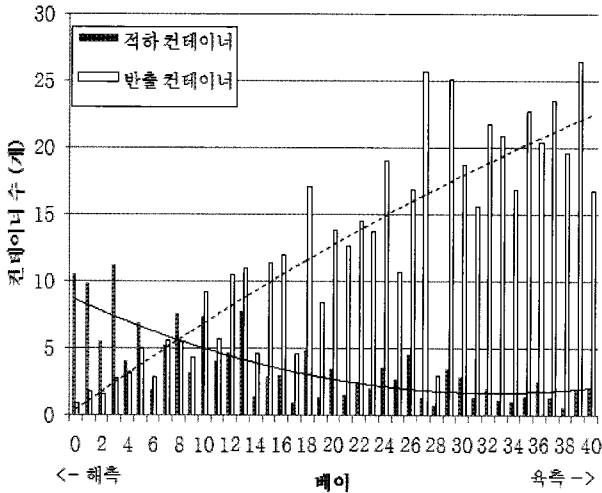


Fig. 6 The average distribution of the 1,000 containers in 10 different datasets used for testing

실험에서는 위에서 설명한 두 가지 협력적 공진화 알고리즘 외에도 이들과의 비교 대상으로 문제 분해의 개념이 없이 탐색하는 시뮬레이티드 어닐링 알고리즘(Simulated Annealing, SA)도 구현하여 시험하였다. 각 탐색 알고리즘의 성능을 공정하게 비교하기 위해 평가 횟수를 1,000,000번으로 제한하였다. 부모 선택을 위해서는 이진 토너먼트 선택(binary tournament selection)을 수행하였고 교배 연산자는 순서 교배 방법(order crossover)(Davis, 1985)를 이용하였다. 그리고 한 해에서 두 컨테이너의 위치를 교환하여 돌연변이가 발생하도록 하였다.

파라미터 값의 설정을 위해서는 미리 각 알고리즘에 대해 10, 50, 100의 집단크기와 돌연변이를 일으키는 해의 수가 집단크기의 0%, 10%, 50%인 경우에 대해 실험하였다. 실험결과, 세대형 CCEA는 대체로 집단 크기 100, 10%의 해를 변이시킬 때가 가장 좋았다.

4.3 결과

Table 1은 1000개의 적하 컨테이너가 분포되어 있는 장치장에서 후보해의 평가값인 적하 시 예상 크레인 이동시간과 재정돈 시간의 가중치를 바꾸어가며 구해 본 각 알고리즘 별 최적해를 얻을 수 있는 재정돈 목적지 선택 휴리스틱의 파라미터인 추가 공간 선택 비율과 그 때의 평가값을 보인 것이다. 여기 보인 평가값은 적하 시 크레인의 이동시간과 재정돈 과정의 크레인 이동시간을 가중합한 것이다. 표에서 볼 수 있듯이 병렬 세대형 CCEA의 성능이 가장 좋은 것을 알 수 있다.

Fig. 7은 병렬 세대형 CCEA를 사용하며 가중치를 1:50으로 설정하고 목표 스택 선정 단계의 추가 공간 선택 비율을 변화시키며 얻어진 재정돈 시간과 적하 시 크레인의 최대 생산성을 보여준다. 본 논문의 적하 시 크레인의 최대 생산성은 크레인이 차량과의 교류 없이 연속적으로 작업했을 경우 얻을 수 있는 생산성을 나타낸 것이다. 그림에서 볼 수 있듯이 공간 비율이 늘어남에 따라 적하 시 ASC 생산성이 감소하는 추세를 있음을

확인할 수 있다. 이것은 추가 공간의 비율이 증가하면 해측에서 멀리 떨어져 적하 작업에 불리한 위치에 있는 스택이 목표 스택으로 선정되고 재정돈 후 컨테이너가 그 스택에 놓일 수 있기 때문이다. 이렇게 해측에서 멀리 떨어진 곳에 컨테이너가 쌓인다면 재정돈 전의 위치와 가까우므로 재정돈 시간은 감소해야만 한다. 그러나 20%의 추가 공간을 선택하면서부터 재정돈 시간 역시 증가하는 것을 확인할 수 있다. 원래 추가 공간이란 것은 재정돈 시에 발생하는 재취급을 줄이기 위한 목적으로 선택된다. 그러나 Fig. 6과 같이 재정돈 시에 비워내는 반출 컨테이너가 상대적으로 육측에 가까이 있는 스택에 많이 쌓여있어 많은 추가 공간을 선택하면 재정돈 시에 더 많은 반출 컨테이너를 치우게 될 수 있다. 또 해 표현 방식에 의해 추가 공간의 선택 비율이 높아질수록 탐색 공간의 크기가 증가하므로 같은 탐색 시간 안에서 좋은 해를 찾을 확률이 낮아지는 것이다.

Table 1 Evaluation values of the best solutions found by different algorithms with different parameters

Weight	Additional Spaces	Algorithms	Best evaluation values
1 : 1	0%	CCEA	24514.2
		SA	27789.3
	10%	CCEA	24608.7
		SA	28217.8
	20%	CCEA	24760.3
		SA	28876.2
30%	CCEA	24680.3	
	SA	29406.5	
1 : 20	0%	CCEA	22674.8
		SA	23189.4
	10%	CCEA	22698.7
		SA	23426.7
	20%	CCEA	22714.3
		SA	23736.9
	30%	CCEA	22771.6
		SA	23995.3

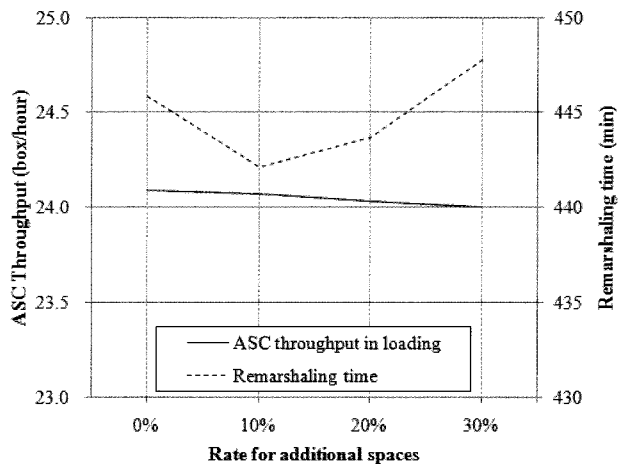


Fig. 7 The ASC throughput and the remarshaling time using parallel generational CCEA with various additional spaces

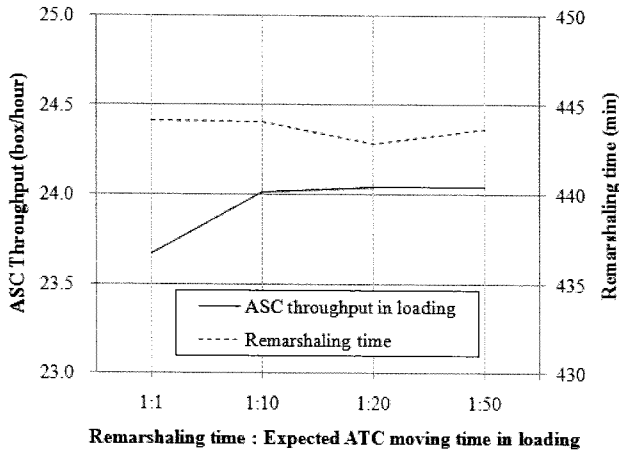


Fig. 8 The ASC throughput and the remarshaling time using parallel generational CCEA with various weights

Fig. 8은 병렬 세대형 CCEA를 사용하며 20%의 추가공간을 선택했을 경우 가중치 조정에 따라 재정돈 시간과 적하 시 크레인의 최대 생산성이 달라지는 모습을 보여준다. 적하 시 예상 이동 시간을 줄이는 쪽에 가중치를 높일수록 적하 시 ASC의 생산성은 증가하는 추세를 확인할 수 있다. 직관적으로 적하 시 ASC 생산성이 증가하면 앞의 실험결과와 반대로 재정돈 시간은 늘어나야 할 것이다. 그러나 반대로 재정돈 시간은 일정한 추세가 없는 것을 볼 수 있다. 이것은 적하 시 크레인 예상 이동 시간에 대한 가중치가 증가할수록 재정돈 시간의 변화가 전체 평가값에 미치는 영향이 줄어들기 때문에 재정돈 시간을 줄이는 데에 집중하지 않은 것으로 볼 수 있다.

5. 결론

본 논문은 협력적 공진화 알고리즘을 이용하여 자동화 장치장에서 컨테이너의 재정돈 작업 계획을 수립하는 방법을 제안하였다. 재정돈에 관한 기존의 연구들은 수평 배치 장치장을 대상으로 하였기 때문에 장치장이 수직으로 배치된 자동화 컨테이너 터미널에 적용될 수 없다는 문제가 있었다. 또, 기존 논문 및 본 논문의 목표 스택 선정 시 추가적인 공간을 선택하는 것이 합당한 이유를 실험 결과를 통해 제시하였다.

본 논문과 Park et al.(2009)의 연구에 사용된 목표 스택 선정 휴리스틱은 탐색에 앞서서 적하에 유리하도록 컨테이너들을 옮겨 둘 충분한 공간을 확보해 준다. 비교 대상 탐색 알고리즘들은 모두 이렇게 확보된 공간 내에 최대한 해측 가까이 그리고 재취급이 발생하지 않는 형태로 컨테이너들을 성공적으로 옮겨 주고 있다. 따라서 추가공간을 선택하는 비율이 같다면 재정돈 후 적하에 소요되는 시간은 거의 유사하기 마련이다. 다만, 어떤 알고리즘을 이용하여 해를 얻었느냐에 따라 그러한 목표 위치로 컨테이너들을 이동시키는 과정에 소요되는 시간이 달라서 성능의 차이를 보이게 되는 것이다.

본 논문은 재정돈 시간을 줄이며 적하 시간 역시 줄이는 재정돈 계획을 수립할 수 있음을 보였다. 적하 작업 시간을 계산하기 위해 크레인이 연속적으로 작업하는 상황을 가정했다. 그러나 실제 터미널에서의 적하 작업은 크레인과 내부 차량의 상호 컨테이너 전달 과정이 포함된다. 그러나 본 논문은 재정돈 작업 계획 수립만을 목표로 하기 때문에 내부 차량과의 교류는 고려하지 않았고, 그 결과 서로 다른 알고리즘으로 수립한 재정돈 계획에 따라 컨테이너를 정돈하여도 적하 작업 시간이 비슷하게 계산되었다. 만약 이러한 내부 차량과의 상호 컨테이너 전달 과정 역시 평가에 포함할 수 있는 방법을 찾는다면 더욱 정교한 재정돈 계획을 수립할 수 있을 것이다. 그리고 본 논문의 실험결과는 장치장에 있는 컨테이너의 분포에 의해 달라질 수 있다. 본 논문에서는 평균적으로 이미 잘 분포되어 있는 장치장을 대상으로 실험하여 4.3결과 같은 결과를 얻을 수 있었다. 그러나 본 논문이 제안하는 방법이 터미널에서 있을 수 있는 컨테이너의 다른 분포 상황에서도 완벽히 수행된다고 보장할 수는 없다. 따라서 본 논문에서 이미 실험한 컨테이너의 분포와 다른 컨테이너 분포 상황 하에서 실험을 수행해보아야 할 것이다.

후 기

본 연구는 국토해양부 “지능형 항만물류시스템 기술개발” 사업의 연구비 지원에 의해 수행되었습니다.

참고 문헌

- [1] 강재호, 오명섭, 류광렬, 김갑환 (2005), “컨테이너 터미널 장치장에서 블록 내 이적을 위한 컨테이너 이동 순서 계획,” 한국항해항만학회지, 제 29권 1호, pp. 83-90.
- [2] 오명섭, 강재호, 류광렬, 김갑환 (2005). “복수 크레인을 활용한 블록 내 컨테이너 이적 계획,” 한국 항해 항만 학회지 제 29권 5호, pp. 447-455.
- [3] Davis, L. (1985), “Applying adaptive algorithms to epistatic domains,” Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 162-164, 1985.
- [4] Hirashima, Y., Ishikawa, N., and Takeda, K. (2006), “A new reinforcement learning for group-based marshaling plan considering desired layout of containers in port terminals,” Proc. IEEE Conf. Networking, Sensing and Control, pp. 670-675.
- [5] Husbans, P. and Mill, F. (1991), “Simulated coevolution as the mechanism for emergent planning and scheduling,” Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 264-270.
- [6] Kang, J., Oh, M. S., Ahn, E. Y., Ryu, K. R., and Kim, K. H. (2006), “Planning for intrablock remarshaling in a container terminal,” IEA/AIE, LNAI 4031, pp. 1211-1220.

- [7] Kim, K. H., and Bae, J. W. (1998), "Remarshaling export containers in port container terminals," *Computers and Industrial Engineering*, vol. 35, no. 3-4, pp. 655-658.
- [8] Lee, Y., and Hsu, N. Y. (2007), "An optimization model for the container premarshalling problem," *Computer and Operations Research*, vol. 34, pp. 3295-3313.
- [9] Panait, L., Luke, S., and Harrison, J. F. (2006), "Archive based cooperative coevolutionary algorithms," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 345-352.
- [10] Park, K., Park, T., and Ryu, K. R. (2009), "Planning for remarshaling in an automated container terminal using cooperative coevolutionary algorithms," *ACM Symposium on Applied Computing*.
- [11] Wiegand, R. P. (2003), *Analysis of cooperative coevolutionary algorithms*, PhD thesis, Department of Computer Science, George Mason University.
- [12] Wiegand, R. P. and Sarma, J. (2004), "Spatial embedding and loss of gradient in cooperative coevolutionary algorithms," *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature*, pp. 912-922.
- [13] Zhang, Y., Mi, W., Chang, D., and Yan, W. (2007), "An optimization model for intrabay relocation of outbound container on container yards," *International Conference on Automation and Logistics*, pp. 776-781.

원고접수일 : 2009년 6월 24일

심사완료일 : 2009년 8월 20일

원고채택일 : 2009년 8월 25일