한국철도학회논문집
제13권 제5호(2010년 10월) pp. 535-540

화물열차 작업선배정 및 열차조성을 위한 수리모형 및 해법

An Optimization Model for Assignment of Freight Trains to Transshipment Tracks and Allocation of Containers to Freight Trains

김경민[1] · 김동희[1] · 박범환[†]
Kyung min Kim · Dong-Hee Kim · Bum Hwan Park

Abstract We present an optimization model for how to assign the freight trains to transshipment tracks and allocate the containers to the freight trains in a rail container terminal. We formulate this problem as a multi-criteria integer programming to minimize the makespan of job schedule and simultaneously to maximize the loading throughput, i.e. the number of containers to be disposed per day. We also apply our model to the instance obtained from the real-world data of the Uiwang Inner Container Depot. From the experiments, we can see an improvement of approximately 6% in makespan, which means that our model can contribute to the improvement of the disposal capacity of containers without additional expansion of facilities.

Keywords : Assigning transshipment track, Container allocation, Mathematical model, Rail container terminal

초 록 본 연구는 철도물류터미널에서 작업선 배정 및 열차조성을 위한 최적화 모형을 제시한다. 본 연구에서는 하루 터미널 작업 종료시간을 최소화하며 처리 컨테이너수를 최대화하는 다기준 정수 계획 모형을 개발하였고, 의왕ICD로부터의 실제 데이터로부터 구성된 현실 문제에 대해 본 모형을 적용하여 실험하였다. 실험 결과 목적함수의 가중치에 따라 다양한 파레토 최적해를 산출할 수 있었으며, 기존의 해보다 전체 작업종료 시간을 약 6% 감소시켰으며 이러한 결과는 본 연구에서 개발한 최적화 모형을 적용할 경우 설비의 추가적인 확장 없이도, 철도 컨테이너 터미널의 처리효율 향상을 가져올 수 있음을 의미한다.

주요어 : 작업선배정, 열차조성, 수리모형, 철도컨테이너터미널

1. 서 론

1.1 연구의 배경 및 목적

최근 지속가능 국가교통물류 개선 시책에 따라 철도수송력 증대 및 철도물류거점화 등이 추진되고 있으며 철도운송 물동량 역시 연평균 10%내외로 증가할 것으로 예상된다. 특히, 컨테이너는 다른 품목에 비해 네트워크가 비교적 간단하고 고객요구에 맞추어 소형화물로의 운송이 용이하다는 특징 때문에 점차 철도운송에서 차지하고 있는 비중이 증대되고 있다[1].

이에 따라 철도 컨테이너 터미널의 처리능력 향상을 위하여 터미널 인프라의 확충 및 시설/장비(게이트, 야드, 인입선, 작업선, 장비, 동선)의 효율적 운영에 대한 요구가 커지고 있다[2]. 그러나 이러한 물리적인 설비(facilities)의 확장은 매우 긴 구축 기간과 큰 비용을 필요로 하다.

본 연구에서는 큰 비용과 오랜 구축기간을 필요로 하는 물리적인 설비의 확충이 아닌, 작업선 배정 및 열차조성을 위한 최적화 방법을 이용하여, 운영 방법의 개선만으로도 현재 철도 컨테이너 터미널의 용량 확대를 도모할 수 있는 방법론을 제시하고자 한다.

작업선배정이란 화물열차에 적하역을 위한 철도작업선을 배정하는 것으로 의왕ICD의 경우 하루 약 40여편의 도착/발송 화물열차가 11개의 작업선에 나누어 배정 되고 있는데, 이는 운영 계획 담당자의 경험적 방법에 의해 이루어지고 있다. 또한 열차조성계획은 목적지 및 입주사별 하루 배정된 컨테이너를 각 화물열차로 나누어 컨테이너별 발송 화물열차를 결정하는 것으로 항만 터미널과는 다른 철도터미널의 고유 특성이라고 볼 수 있다. Fig. 1은 열차별 작업선 배정 및 컨테이너의 열차 할당을 그림으로 표현한 것으로서, 작업선 혹은 적하역 장비의 증대가 아닌 운영 방법 개선을 통해 컨테이너 터미널의 처리 용량을 확대하고자 할 경우, 작업선 할당과 열차조성을 동시에 고려해야 함을 보여주고 있다.

본 연구는 기존의 경험적 방법에 의존한 열차별 작업선 배정 및 열차별 컨테이너 할당 방법을 개선하여, 보다 빠른 시간 안에 보다 많은 컨테이너를 처리함으로써, 철도 컨테이너 터미널의 처리 효율을 증가시킬 수 있는 수리적 모형 및 실험결과를 제시한다.

[†]교신저자 : 한국철도대학 경영정보과
E-mail : beomi72@hanmail.net
[1]한국철도기술연구원 교통물류연구실

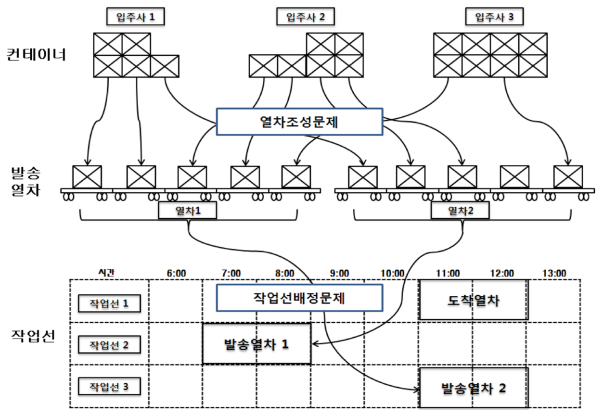

Fig. 1 Assignment to transshipment tracks and allocation of containers

1.2 관련연구

본 절에서는 작업선 배정과 열차별 컨테이너 할당에 관련된 연구를 구분하여 살펴본다. 먼저 철도 컨테이너 터미널에서의 작업선 배정과 관련한 연구는 전무하며, 철도 컨테이너 터미널에서의 작업선 배정과 유사한 문제라 볼 수 있는 항만 터미널에서의 선석 계획(Berth Allocation)과 관련한 연구는 최적화 분야에서 오래된 연구주제이다. [3]의 선석계획은 QC(Quay crane)의 작업 시간 및 용량을 고려하여 선박별로 선석 시간 및 위치를 결정하는 모형으로서, 본 연구는 철도 터미널에서의 작업선에 설치된 적하역 장비는 고정되어 있어, 작업선별 작업 속도는 일정하다. 따라서 본 연구의 작업선 배정은 선석계획 문제보다는 보다 단순한 형태가 된다. 이러한 열차의 작업선 배정 문제는 병렬 기계(Parallel Machines)에 비선점(nonpreemtion)으로 작업을 스케줄링 하는 문제와 유사하며, 전체작업시간의 종료시간(makespan)을 최소화하는 문제는 NP-hard로 알려져 있다[4]. 하지만, 본 연구는 열차별 작업선배정 뿐만 아니라, 배정된 열차에 대해 컨테이너를 할당하는 문제를 동시에 다룸으로써, 기존의 병렬 기계에서의 비선점 스케줄링 문제보다 훨씬 복잡하다.

열차별로 컨테이너를 어떻게 적재할 것이냐에 관한 연구와 관련해서는, 그것의 목적함수에 따라 다양한 모델이 개발되어 있다. 먼저 [5]은 공로-철도간 환적 작업 시, 컨테이너의 재작업(Double handling)에 관련되는 일련의 비용 최소화를 위한 컨테이너 위치 할당 문제를 제시하였고, [6]는 공기역학에 기인한 에너지 효율적 이단적재열차(Double Stack Train)의 열차조성패턴을 위한 정수계획(Integer Programming) 모형을 제시하였다. 이와 같은 연구는 일관수송 중 철도운송 거리가 길고 이단적재 및 화물장대 열차가 보편화된 미국에서 주로 연구되었다. [7]는 화물열차간 환적과 관련한 비용을 최소화하는 컨테이너별 화물열차별 위치를 계산하는 최적화 모형을 제시한 바 있다. 뿐만 아니라 컨테이너 화물의 수송 품질 개선을 위해, 컨테이너별 수송 납기 기한(Due date)가 주어져 있을 때 열차 스케줄 및 컨테이너의 열차별 할당 방법을 계산하는 최적화 모형으로는 [8,9]의 연구가 있다.

기존 연구는 작업선이 이미 배정된 하나의 열차에, 컨테이너 재작업 비용과 관련하여 컨테이너들을 어떻게 적재할 것인지 혹은 컨테이너별 수송 납기 기한을 고려하여 컨테이너 별 어떤 열차에 할당할 것인지를 고려한 연구들로서, 단일 컨테이너 터미널의 운영 효율 개선을 도모하는 본 연구와는 차별성을 갖고 있다.

본 연구의 구성은 다음과 같다. 2장에서는 작업선배정 및 열차조성 문제를 정의한다. 3장에서는 정의된 문제의 수리모형 및 해법을 제시하며 4장에서는 현재 의왕ICD 데이터에 기반한 실험 시나리오에 개발된 해법을 적용한 결과를 보여 준다. 마지막 5장에서는 결론 및 향후 연구방향을 제시한다.

2. 문제정의

가용한 작업선의 수가 정해져 있고 하루에 터미널에서 처리되는 도착 및 발송열차가 정해져 있을 때 열차별 작업선 배정을 어떻게 하느냐는 전체 터미널 작업종료시간에 크게 영향을 미친다. 예를 들어, 작업선이 3개, 처리 열차수가 5개, 예정된 각 열차의 작업시간이 Fig. 2와 같을 때 좌측과 같이 작업선 배정을 한다면 전체작업종료시간은 18이 된다. 하지만, 우측과 같이 작업선을 배정하면 전체 작업종료시간을 13으로 줄일 수 있다. 이와 같이 전체 작업종료시간을 줄이는 것은 각 작업선의 작업시간을 평준화 시키게 되며 터미널의 처리 용량을 증가시킨다.

작업선 배정은 열차를 몇 번 작업선에 언제 작업을 시작하고 종료 할 것인지를 결정하는 것으로 이때, Fig. 3과 같

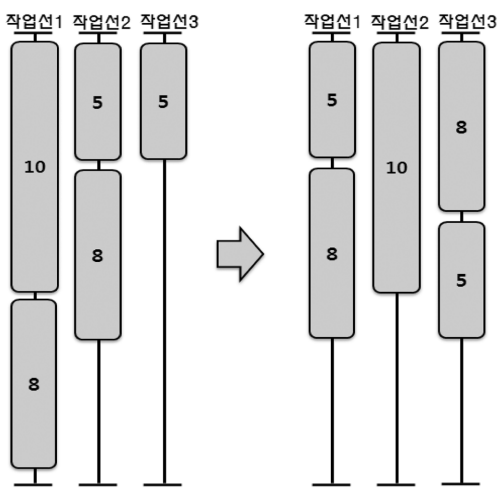

Fig. 2 Makespan time according to assignment transshipment tracks

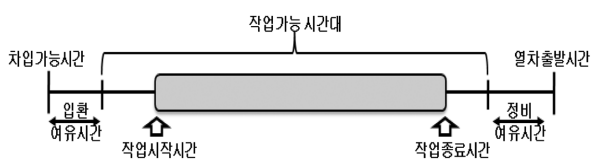

Fig. 3 Operation time on a train

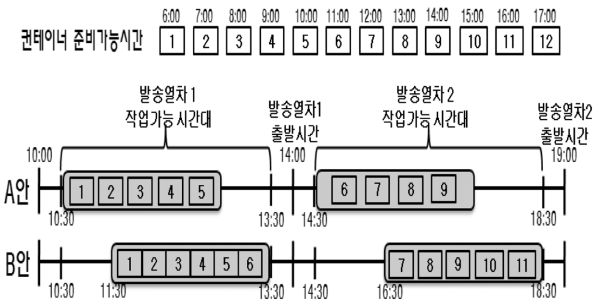

Fig. 4 Container not to be disposed

Table 1 Information for train

정보	설명
S_i	열차 i의 차입가능시간 (시간 slot의 인덱스값)
D_i	열차 i의 적하역소요시간 (시간 slot의 개수)
WA_i	발송열차 i의 최대 화차량수
$DPTTM_i$	발송열차 i의 출발계획시간 (시간 slot의 인덱스 값)
BT_i	발송열차 i가 BT열차가 아닌 경우 BT_i=1, BT열차인 경우 BT_i=2
$DEST_i$	발송열차 i의 목적지
$stTM$	작업선 입환여유시간 (시간 slot의 개수)
$spTM$	작업선 정비여유시간 (시간 slot의 개수)

Table 2 Information for container

정보	설명
$DEST_c$	컨테이너 c의 목적지 정보
R_c	컨테이너 c의 최소 준비가능시간 (시간 slot의 인덱스값)

이 작업시작시간은 차입시간에서 입환여유시간을 더한 이후가 되어야 하며, 작업종료시간은 열차출발시간에서 정비여유시간을 뺀 시간보다는 작아야 한다.

그러나 작업종료시간을 줄일 경우 미발송 컨테이너를 증가시킬 수 있다. 이는 열차별 적재 용량 때문에 발생하는 것이 아니라, 입주사가 확보하는 컨테이너 물량의 적재 준비시간 때문에 발생한다. 예를 들어, Fig. 4와 같이 발송 배정된 컨테이너가 6:00부터 17:00까지 매 시간 하나씩 총 12개이고 이를 계획된 발송열차 1과 2를 통해 발송해야할 때, 작업선 배정을 A안과 같이 발송열차1은 10:30에 적하작업을 시작, 발송열차2는 14:30에 적하작업을 시작하면 열차조성은 발송열차 1에 총 5개의 컨테이너를 발송하고 발송열차 2에 4개의 컨테이너를 발송하게 되며 미발송 컨테이너는 3개이다. 그러나, B안과 같이 발송열차1은 11:30에 적하작업을 시작, 발송열차2는 16:30에 적하작업을 시작하면 미발송 컨테이너는 1개 발생한다. 이와 같이 작업선 배정결과에 따라 열차조성이 변경되며 미발송 컨테이너의 수도 결정된다. 즉, 미발송 컨테이너를 최소화하기 위해서는 작업시작시간을 최대한 늦추어야 한다. 하지만, 이는 전체 작업종료시간을 최소화 시키는 것과 상충관계(trade-off)를 가진다. 본 연구에서는 이러한 상충관계를 고려하여 철송작업 종료 시간과 미발송 컨테이너 수를 동시에 고려할 수 있는 다기준(multi-criteria) 최적화 모형을 제시한다.

최적화 모형 작성을 위해 본 연구는 주어진 터미널 운영시간을 10분 단위의 시간 slot으로 세분화하여, 열차별 적하역 시간이나 준비시간 등과 같은 시간과 관련한 모든 데이터를 시간 slot으로 변환한다. 이러한 시간 slot을 이용하여, 작업선 배정 및 열차 조성에 필요한 열차 정보와 컨테이너 정보는 아래와 같다.

먼저 열차 정보는 Table 1과 같다. 차입시간(S_i)은 철도물류터미널의 작업선으로 입환가능한 최소시간을 의미하며, 적하역소요시간(D_i)은 실제 적하역에 소요되는 시간으로 도착열차의 경우 기조성된 화차량, 발송열차의 경우 최대 조성가능한 화차량에 비례하여 설정된다. 발송열차의 경우에는 열차출발계획시간($DPTTM_i$), 계약발송열차(Block Train: BT) 열차유무(BT_i), 목적지 정보($DEST_i$)를 가진다. 또한, 작업을 시작하기 전 입환여유시간($stTM$)과 작업종료후 정비여유시간($spTM$)을 가진다. 이때 입환여유시간과 정비여유시간은 열차별로 모두 동일한 값을 설정하였다.

열차조성을 위한 컨테이너 정보는 Table 2와 같다. 컨테이너는 목적지 정보를 가지며 계약발송열차의 경우 열차의 목적지와 컨테이너의 목적지가 일치하게 열차를 조성한다. 또한, 컨테이너의 최소 준비가능시간은 각 입주사에서 컨테이너를 발송열차에 적하 가능한 최소 시간을 의미한다. 본 연구에서는 열차의 작업시작시간 전에 준비가능한 컨테이너만 해당 열차에 적하 가능하다고 가정한다.

3. 수리모형 및 해법

철송작업 종료시간 최소화 및 처리 컨테이너수를 최대화(미발송 컨테이너를 최소화)하는 작업선 배정 및 열차조성 문제는 정수계획법(Integer Programming)으로 모형화 된다. 아래 모형에서 I_D는, 적화 작업 대상 열차에 할당되지 않아 발송이 보류되는 컨테이너를 배정하기 위한 가상의 하나의 열차로 이루어진 열차 집합이다. 이 가상의 열차는 가상의 작업선(4절의 예제에서 12번째 작업선)에서 작업이 이루어지며 적화 작업 대상 열차에 적재하지 못하는 컨테이너는 언제든지 이 가상의 열차에 적재 가능하다고 가정한다.

집합

- J : 작업선 집합
- D : 목적지 집합
- T : 시간 slot 집합
- C : 컨테이너 집합
- I_U : 하화 작업 대상 열차집합
- I_L : 적화 작업 대상 열차집합

- I_D : 미발송컨테이너 배정 열차집합
- I : 열차 집합($I = I_U \cup I_L \cup I_D$)

파라미터

- S_i : 열차i의 작업선 차입가능시간(시간slot의 인덱스 값으로 표현)
- D_i : 열차 i의 모든 컨테이너가 적재 혹은 하화되는데 걸리는 시간(시간slot의 개수로 표현)
- WA_i : 발송열차($i \in I_L$) i의 최대 적재 컨테이너 수
- $DPTTM_i$: 발송열차($i \in I_L$) i의 출발시간(시간slot의 인덱스 값으로 표현)
- BT_i : 발송열차($i \in I_L$) i의 BT열차여부
- $DEST_i$: 발송열차($i \in I_L$) i의 목적지(목적지 인덱스 값으로 표현)
- R_c : c컨테이너의 준비가능시간(시간slot의 인덱스 값으로 표현)
- $DEST_c$: c컨테이너의 목적지(목적지 인덱스 값으로 표현)
- $stTM$: 작업선 입환여유시간(시간slot의 개수로 표현)
- $spTM$: 작업선 정비여유시간(시간slot의 개수로 표현)
- M : big-M

변수

- $q_{ijt} \in \{0,1\}$: i열차가 j작업선의 t번째 시간slot에 적하역을 시작하면 1 그렇지 않으면 0
- $x_{ic} \in \{0,1\}$: 컨테이너 c가 열차 i에 할당되면 1 그렇지 않으면 0
- $m \in T$: 전체 작업종료시간(makespan)
- $n_i \in T$: 열차 i의 작업 시작시간

수리모형

$$\min \ \alpha \cdot \sum_{i \in I_D} \sum_{c \in C} x_{ic} + (1-\alpha) \cdot m \tag{1}$$

subject to

$$\sum_{j \in J} \sum_{t \in T} q_{ijt} = 1, \forall i \in I \tag{2}$$

$$\sum_{p \in I: p \neq i} \sum_{s=t-stTM}^{t+D_i+spTM-1} q_{pjs} \leq M \cdot (1-q_{ijt}), \forall i \in I, \forall j \in J, \forall t \in T \tag{3}$$

$$\sum_{j \in J} \sum_{t \in T} t \cdot q_{ijt} = n_i, \forall i \in I \tag{4}$$

$$n_i + D_i \leq m, \forall i \in I_L \cup I_U \tag{5}$$

$$n_i - stTM \geq S_i, \forall i \in I \tag{6}$$

$$n_i + D_i + spTM - 1 \leq DPTTM_i, \forall i \in I_L \tag{7}$$

$$n_i \geq R_c \cdot x_{ic} - 1, \forall i \in I_L, \forall c \in C \tag{8}$$

$$\sum_{c \in C} x_{ic} \leq WA_i, \forall i \in I_L \tag{9}$$

$$\sum_{i \in I_L \cup I_D} x_{ic} = 1, \forall c \in C \tag{10}$$

$$\sum_{i \in I_L : BT_i = 2} \sum_{c \in C : DEST_c \neq DEST_i} x_{ic} = 0 \tag{11}$$

$$q_{ijt}, x_{ic} \in \{0,1\} \tag{12}$$

$$m, n_i \geq 0, \text{ 정수} \tag{13}$$

본 연구에서는 컨테이너 처리 용량과 작업종료 시간을 동시에 고려하기 위해, 식(1)과 같이 두 목적의 가중합, 즉 $\alpha \cdot \Sigma\Sigma x_{ic} + (1-\alpha) \cdot m$을 최소화한다. 가중치 α값 조정을 통해 다양한 파래토 해를 구함으로써, 의사결정자에게 보다 다양한 대안을 제공할 수 있다. 식(2)에 의해, 각 열차에 배정되는 작업선 및 작업 시작 시간은 유일하게 결정된다. 식(3)은 i열차가 j작업선을 t번째 시간slot부터 D_i 개의 시간slot을 점유하면 다른 열차는 해당 작업선을 해당시간 동안 점유할 수 없음을 의미한다. 식(4)는 열차i의 작업시작시간인 n_i을 계산한다. n_i을 이용하여, 식(5)와 같이 전체작업종료 시간을 계산할 수 있다. 식(6)은 열차의 차입시각에서 입환여유시간을 더한 이후에 작업을 시작하게 하는 제약이며, 식(7)은 작업시작시간으로부터 작업시간, 정비여유시간을 고려하여, 발송열차들의 열차출발시간을 준수하기 위한 제약식이다. 또한 식(8)에 의해, 발송열차의 작업시작시간 전에 준비 가능한 컨테이너만을 화차에 적재된다. 식(9)는 열차당 발송가능한 최대 컨테이너 수를 제약하고, 식(10)은 컨테이너 c는 반드시 발송열차 혹은 더미열차 중 하나의 열차에는 적재되어야 하는 제약이며 식(11)은 계약발송열차의 경우 열차의 목적지와 컨테이너의 목적지를 일치시키는 제약이다.

본 연구에서는 위 수리모형을 IBM ILOG OPL 6.3을 사용하여 모델링한 후 상용최적화 프로그램인 CPLEX를 활용하여 해를 도출한다. 다음 실험결과에서도 확인할 수 있지만, CPLEX에 내장되어 있는 분지한계법[10]만으로도 충분히 빠른 시간안에 해를 구할 수 있음을 확인할 수 있다.

4. 실험결과

앞 절에서 제시한 작업선 배정 및 열차조성 최적화 수리모형의 효과를 분석하기 위하여 다음과 같은 의왕ICD 운영상황을 바탕으로 시나리오를 구성하였다. 총 11개의 작업선과, 42개의 열차(도착열차 20대, 발송열차 22대)가 있으며, 수리모형을 위해 더미열차 1대, 가상의 작업선 1개를 구성한다. 이 때 더미열차의 차입시간은 시작시간(시간slot:1)이다. 발송열차는 출발계획 시간에서 약 4시간 전에 터미널로 차입가능하다고 가정하였고, 계약발송열차는 최대화차량을 28대, 적하역소요시간을 1시간 30분으로, 그 외의 열차는 30대, 2시간으로 각각 설정하였다.(Table 3) 입환여유시간 및 정비여유시간은 10분으로 설정하였다. 하루 발송배정 된 컨테이너는 총 700개이며 컨테이너 및 열차의 목적지는 8가지이고 컨테이너의 목적지와 준비가능시간 정보는 현재 의왕ICD 실적을 반영하여 생성하였다. 시간범위는 8:00~23:00이며 10분 간격으로 총 90개의 시간slot을 구성하였다.

먼저 본 연구에서는 가중치 α의 변화에 따른 해의 변화를 살펴보기 위해, α를 0에서부터 1까지 0.1씩 증가시켜 가

Table 3 Sample trains

Train index	도착(1)/발송(2)/더미(3)	차입시간 S_i	작업시간 D_i	최대Z 발송량수 WA_i	출발계획시간 $DPTTM_i$	BT 열차여부 BT_i	목적지 $DEST_i$
1	1	57	10	-	-	-	-
2	1	59	11	-	-	-	-
3	1	62	12	-	-	-	-
4	1	70	12	-	-	-	-
5	1	72	12	-	-	-	-
6	1	1	9	-	-	-	-
7	1	1	12	-	-	-	-
8	1	1	10	-	-	-	-
9	1	1	11	-	-	-	-
10	1	1	12	-	-	-	-
11	1	1	7	-	-	-	-
12	1	1	13	-	-	-	-
13	1	1	9	-	-	-	-
14	1	1	10	-	-	-	-
15	1	1	10	-	-	-	-
16	1	1	7	-	-	-	-
17	1	5	12	-	-	-	-
18	1	13	12	-	-	-	-
19	1	19	13	-	-	-	-
20	1	20	14	-	-	-	-
21	2	1	12	30	17	1	4
22	2	1	12	30	19	1	3
23	2	8	12	30	32	1	4
24	2	10	12	30	34	1	4
25	2	14	12	30	38	1	5
26	2	26	12	30	50	1	4
27	2	31	12	30	55	1	3
28	2	40	12	30	64	1	5
29	2	42	12	30	66	1	4
30	2	46	12	30	70	1	4
31	2	47	9	28	71	2	4
32	2	48	12	30	72	1	3
33	2	52	12	30	76	1	5
34	2	54	12	30	78	1	3
35	2	57	9	28	81	2	4
36	2	61	9	28	85	2	4
37	2	61	12	30	85	1	3
38	2	61	12	30	85	1	4
39	2	61	12	30	85	1	5
40	2	61	12	30	85	1	3
41	2	61	12	30	85	1	5
42	2	61	12	30	85	1	4
43	3	1	111	0	-	-	-

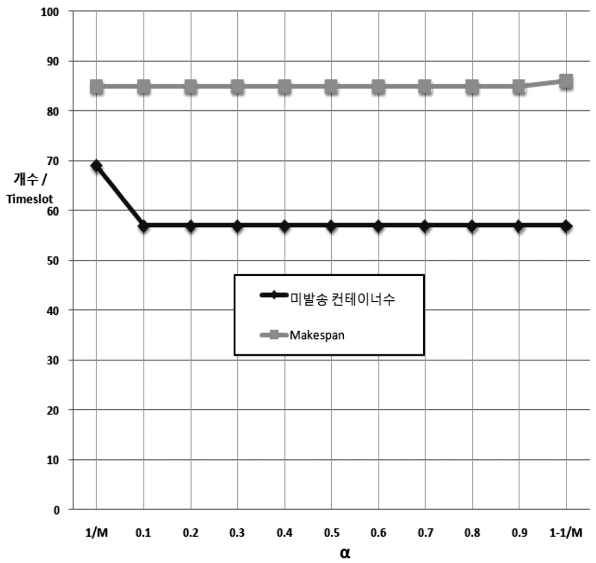

Fig. 5 Effect of weight parameters

중치에 따른 해의 변화를 살펴보았다. Fig. 5에서 보는 것처럼 α=0.1 이후에는 해의 변화가 거의 없음을 알 수 있다. 가중치의 민감도는 발송해야 할 컨테이너 수, 그리고 작업선 등 문제 크기에 따라 달라진다. 여기서 중요한 점은 α를 적정하게 설정하면, 기존의 경험에 의한 방법론보다 훨씬 좋은 해를 도출할 수 있다는 점이다.

이를 위해 본 연구에서는 두 가지 경험적 방법론과 비교해보고자 한다. 경험적 방법으로는 FIFO(First-In First-Out)와 LILO(Last-In Last-Out) 방식을 고려해 볼 수 있다. FIFO 방식은 차입시간 순서에 따라서 입환여유시간 이후 곧바로 작업선을 배정하고 배정된 열차에 컨테이너의 준비시간 순서에 따라 컨테이너를 조성시키는 방법이다. 이는 작업종료시간을 최소화하기 위한 가장 간단한 방법 중 하나이다. LILO방식은 반대로 열차를 작업선에 최대한 늦게 배정하여 컨테이너를 가능한 한 많이 수송하기 위한 방식이다.

Table 4는 Fig. 5로부터 도출된 값을 이용하여 앞의 두 가지 경험적 방법과 최적화 모형의 해를 비교한 것이다. 작업종료시간을 가장 앞당기는 FIFO방식일 때 작업종료시간은 22 : 00(시간slot : 84)이고 총 컨테이너 중 137개를 미발송 하였다. 반대로 발송컨테이너를 최대로 하는 LILO방식일 때 작업종료시간은 23 : 10(시간slot : 91)이고 총 컨테이너 중 57개를 미발송 하였다. 최적화 모형의 경우 가중치에 따라서 변하지만 작업종료시간은 FIFO와 LILO의 종료시간 사이인 22 : 00(시간slot : 84)~23 : 10(시간slot : 91)에서 값을 가지며 미발송컨테이너수 역시 57~137개 사이의 값을 가졌다. 예를 들어, α가 0.5인 경우 작업종료시간은 22 : 10(시간slot : 85), 미발송 컨테이너는 57개였다. 즉, 동일한 처리 컨테이너수를 유지하면서 LILO대비 작업종료시간은 6% 감소했으며 유사한 작업종료시간 안에 FIFO대비 처리 컨테이너의 수는 약 14% 증가한다.

실험환경으로 Core2 Quad CPU 2.50GHz, 3.50GB RAM, IBM ILOG OPL 6.3을 사용 하였을 때 계산시간은 30초에

Table 4 Summary of the experiment results

	FIFO 작업	최적화모형	LILO 작업
		α = 0.1	
작업종료시간 (시간 slot)	22:00(84)	22:10(85) (LILO대비▼6%)	23:10(91)
미발송 컨테이너수	137개	57개	57개
처리 컨테이너수	563개	643개 (FIFO대비▲14%)	643개

서 179초까지 걸렸으며 평균 124초인 것으로 나타나 본 모형이 사전계획의 변동이 자주 발생하는 철도물류터미널의 특성을 고려하여 짧은 시간 안에 합리적 해를 제공 가능한 것으로 확인되었다.

5. 결 론

본 연구에서는 철도수송의 시종착이 되는 대규모 철도컨테이너 터미널에서 발송열차의 출발시간을 준수하고 전체 터미널 작업종료시간을 최소화하면서 처리 컨테이너를 최대화하는 작업선 배정 및 열차조성에 대한 수리모형 및 해법을 제시하였다. 제시한 해법을 철도물류터미널 상황을 반영한 시나리오에 적용할 경우, 철도물류터미널의 처리용량 증대를 위한 별도의 설비 확장 없이도, 작업종료시간과 처리 컨테이너수를 획기적으로 개선할 수 있음을 확인하였다.

또한 본 연구에서 제시한 수리모형상의 열차별 작업시간, 준비시간 등은, 현재 존재하는 적하역 장비, 작업선, 터미널 레이아웃에 의해 결정되는 파라미터들이다. 따라서 향후 철도물류터미널의 설비의 변화 및 추가적인 확충이 발생 할 때, 수리 모형상의 파라미터 조정만으로도, 철도 물류 터미널의 처리효율 증대 효과를 사전에 시뮬레이션해 볼 수 있을 것이다.

참고문헌

[1] K.M. Kim, B.H. Park, D.H. Kim, S.H. Hong (2010) A Study on the Assigning Track of Freight Trains in a Rail Container Terminal, *Korean Society for Railway Spring Conference*.

[2] J. Woo, J. An, H. Kim (2009) A Study on the Evaluations and Issues of Rail Container Transport in Korea Railway, *Journal of the Korean Society for Railway*, Vol. 12, No. 5, pp. 751-760.

[3] K.H. Kim (2007) Decision-making Problems for the Operation of Container Terminals, *Journal of the Korean Institute of Industrial Engineers*, Vol. 33, No. 3, pp. 290-302.

[4] Pinedo (1995) Scheduling - Theory, Algorithms and Systems, Prentice Hall.

[5] P. Corry, E. Kozan (2006) An assignment model for dynamic load planning of intermodal trains, *Computers & Operations Research*.

[6] Y. Lai, P.L. Barkan, H. Onal (2008) Optimizing the aerodynamic efficiency of intermodal freight trains, *Transportations Research Part E 44*.

[7] N. Bostel, P. Dejax (1998) Model and Algorithms for Container Allocation Problems on Trains in a Rapid Transshipment Shunting Yard, *Transportation Science*.

[8] C.A. Yano, A.M. Newman (2001) Scheduling Trains and Contrainers with Due Dates and Dynamic Arrivals, *Transportation Science 35*.

[9] Nakorn Indra-Payoong, Raymond S K Kwan, Les Proll (2003) Constraint-Based Local Search for Rail Container Service Planning, *Technical report.*

[10] L.A. Wolsey (1999), Integer Programming, *John Wiley and Sons, INC*.

접수일(2010년 8월 30일), 수정일(2010년 10월 9일),
게재확정일(2010년 10월 19일)