

컨테이너 터미널에서의 트랜스퍼 크레인의 최적 운영 방안에 관한 연구*

김후곤** · 김철한***

A Study on the Optimal Routing Problem for a Transfer Crane*

Hu Gon Kim** · Chul Han Kim***

■ Abstract ■

To load a container in a yard block onto a ship, a Transfer Crane (TC) moves to a target yard bay, then its hoist picks up a selected container and loads it onto a waiting Yard Truck (YT). An optimal routing problem of Transfer Crane is a decision problem which determines a given TC's the visiting sequence of yard-bays and the number of containers to transfer from each yard-bay. The objective is to minimize the travel time of the TC between yard-bays and setup time for the TC in a visiting yard. In this paper, we shows that the problem is NP-complete, and suggests a new formulation for it. Using the new formulation for the problem, we investigate some characteristics of solutions, a lower and upper bounds for it. Moreover, our lower and upper bound is very efficient to applying some instances suggested in a previous work.

Keyword : NP-complete, Network Routing Problem, Column Generation Technique

논문접수일 : 2007년 06월 12일 논문게재확정일 : 2008년 02월 26일

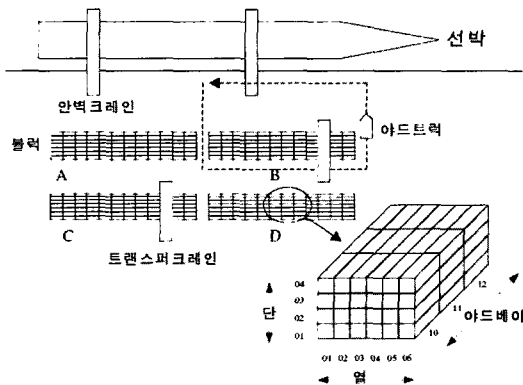
* 이 논문은 2004학년도 경성대학교 학술지원연구비에 의하여 연구되었음.

** 경성대학교 상경대학 e-비즈니스 전공

*** 경성대학교 상경대학 e-비즈니스 전공

1. 서 론

컨테이너 터미널은 [그림 1]과 같이 선박이 접안하는 안벽(Quay)과 대기중인 컨테이너들을 적재하여 두는 컨테이너 야드(Container Yard)로 구성되며, 컨테이너 터미널의 주요 장비로는 안벽에서 선박에 컨테이너를 싣고 내리는 안벽 크레인(Quay Crane, QC), 컨테이너 야드에서 컨테이너를 이송하기 위한 트랜스퍼 크레인(Transfer Crane, TC), 야드와 QC 간에 컨테이너를 운반하는 야드 트럭(Yard Truck, YT) 등이 있다. 컨테이너 야드는 보통 20에서 30개 사이의 야드베이(yard bay)로 구성된 다수의 블록(block)으로 구분되며, 하나의 야드베이에는 컨테이너들이 대개 4단 6열로 적재된다.



[그림 1] 컨테이너 터미널의 예

야드에 있는 수출용 컨테이너(export container)를 선박에 적하하는 경우를 살펴보면, TC는 지정된 야드베이로 이동하여 컨테이너를 들어 올린 후 블록의 측면으로 이동하여 대기중인 YT에 싣고, YT는 해당 컨테이너를 QC로 수송하게 된다. 그리고 QC는 YT에서 컨테이너를 들어 올려 선박에 적하한다.

적하계획(load planning)은 다음과 같이 수행된다. 우선 적하기획자(load planner)는 선박에 대한 적하안전성(loading stability)을 보장하기 위해, 선박회사가 제공한 적하 프로파일(load profile)과 QC의 이용가능성을 고려하여 QC의 작업일정(work schedule)

을 작성하게 된다.

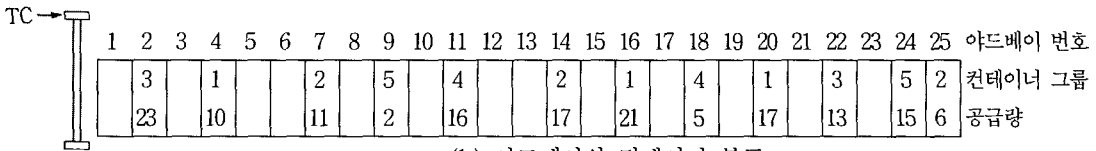
다음으로 작업일정이 결정된 후에, 선박 베이(ship bay)의 특정한 위치에 컨테이너를 실제로 할당하기 위해 적하순서를 결정하게 된다. 적하순서 문제(Load Sequencing Problem)는 매우 복잡하므로 TC 루팅 문제(Routing Problem for TC)와 픽업순서 문제(Pickup Sequencing Problem)로 나누어 생각해 볼 수 있다. TC 루팅 문제에서는 TC의 야드베이 방문순서와 야드베이를 방문할 때 픽업되는 컨테이너의 수를 결정하고, 픽업순서 문제(Pickup Sequencing Problem)에서는 TC와 QC의 취급활동의 편의성, 선박의 안정성, 특정한 선창에서의 최대 적재 높이, 특정한 갑판에서의 최대 중량 등과 같은 기타 제약을 고려하여 개별적인 컨테이너에 대한 적하순서를 결정한다(Kim과 Kim [6]).

[그림 2]의 (a)는 QC가 처리할 작업일정의 간단한 예를 보여 주고 있는데, 여기서 컨테이너 그룹은 동일한 선박에 적하되고 동일한 목적항에서 양하되는 동일한 길이의 컨테이너의 집합을 의미한다. [그림 2]의 (b)는 특정한 선박에 대한 적하를 위해 야드베이에 적재되어 있는 컨테이너들을 나타내고 있으며, TC는 지정된 야드베이로 이동하여 컨테이너를 들어 올려 YT에 싣게 된다. 그림에서 야드베이는 일련번호가 매겨져 있고, 컨테이너 그룹 및 공급될 컨테이너 수량이 표시되어 있다. 이때 TC는 야드베이간 이동시에 선형 궤도 운동을 하게 된다.

컨테이너 터미널에서의 운영 최적화와 관련된 기존의 연구를 살펴보면, Peterkofsky와 Daganzo[9] 및 Daganzo[4]는 QC의 작업일정에 대한 해법을 제시하고 있으며, Kim과 Kim[6]은 주어진 작업일정 및 컨테이너 분포 하에서 TC의 최적 운행 해법에 대한 연구를 수행하였고, Kim 등[5]은 적하순서결정을 위한 유전해법과 beam search 해법을 제안하였다. W.C. Ng와 K.L. Mak [8]은 이와 달리 서로 다른 준비시간을 가지는 양·적하 작업에서의 TC의 최적 알고리즘을 개발하였다. 그 외에 Kim 등[7]은 컨테이너 야드에서 적하시의 재배치 움직임의 수를 최소화하기 위한 저장위치를 결정하는 동적계획법 모형을

작업일정	1	2	3	4	5	6	7	8	9	10
컨테이너 그룹	2	1	3	1	4	2	1	3	5	4
컨테이너 수요	13	19	16	13	10	21	16	20	17	11

(a) 작업일정의 예



(b) 야드베이의 컨테이너 분포

[그림 2] 작업일정과 야드베이의 컨테이너 분포

개발하였다.

본 연구에서는 [6]에서의 TC의 최적 운영 문제를 대상으로, 그 동안의 연구에서 알려지지 않았던 본 문제의 복잡성(complexity)이 NP-complete임을 보이고, 새로운 정식화를 통해 문제의 특성을 규명하고, 현실에서 적용가능한 효율적인 상한과 하한을 제시하도록 한다.

본 논문의 제 2장에서는 용어와 문제의 정의 및 새로운 정식화를 제안하고, 제 3장에서는 이 문제의 복잡성 분석, 최적해의 특성 및 상·하한과 관련된 이론적 결과들을 도출하도록 한다. 제 4장에서는 계산결과를 통해 제안된 하한 및 상한의 차이가 매우 적으면서, 짧은 수행 시간내에 구할 수 있음을 보이도록 하겠다.

2. 용어 및 문제의 정의

TC의 최적 루팅 문제(Transfer Crane Routing Problem, TCRP)는 컨테이너를 선박에 적하하는 작업을 대상으로 하고 있으며, [그림 2]와 같이 컨테이너 그룹 및 수요를 가지는 작업일정과 야드베이의 컨테이너 분포가 주어졌을 때, TC가 방문하는 야드베이들의 순서와 TC가 방문하는 각각의 야드베이에서 픽업하는 컨테이너의 수를 최적으로 결정함으로써 TC의 총 운행시간(total travel time)을 최소화하는 문제이다. 이때 목적함수는 TC의 야드베이간 이동 시간(travel time)과 야드베이에서의 작업준비 시간

(setup time)의 합을 최소화하는 것이 된다. 야드베이간 이동 시간은 야드베이간 거리에 비례하는 것으로 가정하고, 야드베이간 거리는 [그림 2]의 (b)에서 일렬로 열거된 야드베이 번호 간의 차이에 절대값을 취한 값이 된다(예를 들어 야드베이 9과 6사이의 거리는 3이 됨). 또한 작업준비 시간은 실제 픽업 작업을 위해 방문한 야드베이에서 TC가 정확한 위치를 잡은 후 컨테이너 픽업을 시작하기까지 소요되는 시간을 의미한다. 모든 야드베이에서의 작업준비 시간은 동일한 것으로 가정하며, 따라서 전체 야드베이에서의 작업준비 시간은 TC가 픽업을 위해 방문하는 야드베이의 수에 비례하게 된다. 그리고 TC가 하나의 컨테이너를 픽업하여 YT로 싣는데 소요되는 시간은 모두 동일한 것으로 가정하며, 이는 컨테이너 수요의 합에 비례하므로 목적함수에서 제외하기로 한다. 또한 YT는 TC의 작업을 위해 항상 이용가능하다고 가정한다.

TCRP를 위해 다음의 용어를 정의하자.

$G = \{1, 2, \dots, g\}$: 컨테이너 그룹의 집합

$I = \{1, 2, \dots, m\}$: 연속되는 작업일정(sequential work schedule)의 인덱스 집합

$J = \{1, 2, \dots, n\}$: 야드베이(yard bay)의 집합으로 $j \in J$ 는 j 번째 야드베이를 나타냄

a_i : $i \in I$ 번째 작업일정의 컨테이너 수요량

u_i ($i \in G$) : $i \in I$ 번째 작업일정이 속한 컨테이너 그룹

- b_j : $j \in J$ 번째 야드베이에 있는 컨테이너의 공급량
($b_j = 0$ 이면 야드베이에 컨테이너가 없는 경우임)
- $v_j (\in G)$: $j \in J$ 번째 야드베이가 속한 컨테이너 그룹
- $|s-t|$: 두 야드베이 $s, t \in J$ 간의 거리
- α : 야드베이에에서의 작업준비 시간(setup time)
- β : 인접 야드베이로 이동하는데 걸리는 시간(travel time)
- 0 : TC의 초기 출발 야드베이를 0으로 가정함

컨테이너 그룹 $k \in G$ 에 속한 작업일정들의 집합을 $I^k = \{i \in I : u_i = k\}$, 동일한 그룹 $k \in G$ 에 속한 공급 야드베이의 집합을 $J^k = \{j \in J : v_j = k\}$ 로 정의하자. 만약 $b_j = 0$ 이면 야드베이에 컨테이너가 없거나, 작업 일정에 속한 컨테이너 그룹을 포함하지 않는 컨테이너들만 적재되어 있는 경우를 나타낸다.

그룹 $k \in G$ 에 대한 컨테이너의 수요량과 공급량은

$$\sum_{i \in I^k} a_i = \sum_{j \in J^k} b_j \quad (1)$$

를 만족하여야 하며, 이를 그룹별 수요-공급 균형제약으로 부르기로 한다. 컨테이너 그룹 $k \in G$ 에 속한 $i \in I^k$ 의 수요 a_i 에 대해 야드베이 $j \in J^k$ 의 공급 b_j 에서 픽업하는 컨테이너의 수를 x_{ij} 라 하면

$$\sum_{j \in J^k} x_{ij} = a_i \quad \text{for } i \in I^k \quad (2)$$

$$\sum_{i \in I^k} x_{ij} = b_j \quad \text{for } j \in J^k \quad (3)$$

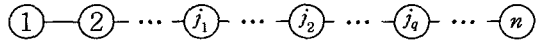
를 만족하여야 된다. 다음은 TC의 선행 궤도 운행 특성과 관련된 정리이다.

Property 1 : 수요 $a_i (i \in I^k)$ 를 위해 방문하는 야드베이의 집합을 $\{j_1, j_2, \dots, j_q\} \subseteq J^k$ 라 하고, $j_1 < j_2 < \dots < j_q$ 라 하자. 이때 $\{j_1, j_2, \dots, j_q\}$ 를 모두 방문하는 최단경로는

$$j_1 - j_2 - \dots - j_q \quad \text{또는} \quad j_q - j_{q-1} - \dots - j_1$$

이다.

(증명) TC는 야드베이간을 선행 궤도로만 운행할 수 있으므로, TC의 운행경로는 야드베이에 해당하는 n 개의 노드를 가지는 [그림 3]과 같은 interval graph로 나타낼 수 있다.



[그림 3] TC의 운행경로를 위한 interval graph 표현

그런데 $j_1 < j_2 < \dots < j_q$ 을 만족하므로, interval graph에서 $\{j_1, j_2, \dots, j_p\}$ 를 모두 방문하는 경로의 길이는 최소한 $|j_q - j_1|$ 이 된다. 경로 $j_1 - j_2 - \dots - j_q$ 와 $j_q - j_{q-1} - \dots - j_1$ 의 이동 거리는

$$|j_1 - j_2| + |j_2 - j_3| + \dots + |j_{q-1} - j_q| = |j_q - j_1|$$

가 되고, 이는 최단경로를 만족한다. 이 외의 다른 경로의 길이는 모두 $|j_q - j_1|$ 를 초과하게 된다. ■

TCRP에서 작업일정 $i \in I^k$ 의 수요 a_i 를 위해 야드베이 $j \in J^k$ 에서 픽업되는 컨테이너 수량 x_{ij} 가 주어졌을 때, $p = \min\{j \in J^k : x_{ij} > 0\}$ 라 하고 $q = \max\{j \in J^k : x_{ij} > 0\}$ 라 하자. 작업 i 를 위해 실제 공급이 발생하는 경우는 $\{j \in J^k : x_{ij} > 0\}$ 로 나타낼 수 있는데, 이에 속한 야드베이 개수만큼의 작업준비 횟수가 발생하게 된다. 이때 TC의 운행경로는 Property 1로부터 p 에서 q 또는 q 에서 p 로 운행하여 작업 하는 두 가지 중에서 하나가 된다. 편의상 전자를 순방향 운행이라 하고, 후자를 역방향 운행이라 하자. 어떤 경우라 하더라도 TC의 운행경로의 길이는 Property 1로부터 $|p - q|$ 가 된다.

임의의 야드베이 쌍 $s, t \in J^k$ 에 대해 순방향 운행과 역방향 운행을 고려하여, 이 야드베이 쌍 사이에 포함되는 야드베이의 집합을

$$J^k(s, t) = \{j \in J^k : s \leq j \leq t\}, \quad \text{if } s \leq t$$

$$J^k(s, t) = \{j \in J^k : t \leq j \leq s\}, \quad \text{if } s > t$$

라 하자. $J^k(s, t)$ 는 노드 s 와 t 사이에 놓여 있으면서 그룹 $k \in G$ 에 속하는 야드베이를 포함하는 노드의 집합이다. 그리고 $i \in I^k$ 에 대해

$$L_i = \left\{ (s_i, t_i) : s_i, t_i \in J^k, \sum_{j \in J^k(s_i, t_i)} b_j \geq a_i \right\} \quad (4)$$

라 하자. $(s_i, t_i) \in L_i$ 는 작업일정 $i \in I^k$ 의 수요 a_i 를 잠재적으로 만족시킬 수 있는 야드베이의 집합 $J^k(s_i, t_i)$ 을 나타내기 위한 것이다. 작업일정 $i \in I^k$ 에 대해 정의되는 $(s_i, t_i) \in L_i$ 를 야드베이 쌍 후보(candidate yard bay pair)로 부르기로 하며, $j \in J^k(s_i, t_i)$ 에 대해 s_i 는 처음으로 방문하는 야드베이로 t_i 는 마지막으로 방문하는 야드베이로 간주한다. 전체 가능한 야드베이 쌍 후보 개수의 상한은 다음과 같다.

Property 2 : 전체 야드베이 쌍 후보의 수는 $\frac{n^2}{|G|}$ 이내이다.

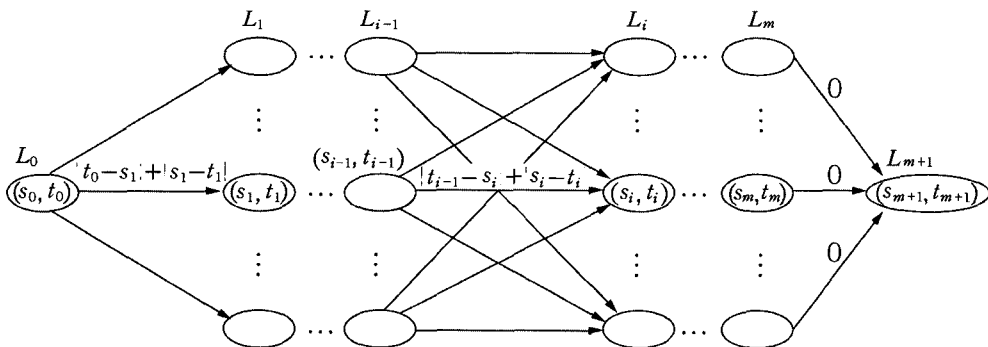
(증명) $n_k = |I^k|$ 라 하자. $k \in G$ 에 대한 야드베이 쌍 후보의 개수는 $s=t$ 인 경우와 $s \neq t$ 인 경우이므로 최대 $n_k + 2 \cdot \binom{n_k}{2} = n_k^2$ 가 된다. 그리고 $\sum_{k \in G} n_k = n$ 이므로,

$$\sum_{k \in G} n_k^2 \leq \sum_{k \in G} \left(\frac{n}{|G|} \right)^2 = \frac{n^2}{|G|}$$

가 성립함을 쉽게 보일 수 있다. ■

모든 작업일정 $i \in I$ 에 대한 L_i 를 계층으로 하는 계층 네트워크(layered network)를 $N=(L, E)$ 이라 하고, TC의 운행경로를 $N=(L, E)$ 상의 최단경로로 나타내기라 하자. $N=(L, E)$ 에서 노드의 집합 L 은 다음의 노드들로 구성된다. TC의 출발 야드베이는 0으로 가정 하였으므로, TC의 출발지 계층을 $L_0 = \{(s_0, t_0)\}$ 라 하고, 여기서 $s_0=0, t_0=0$ 라 하자. $1 \leq i \leq m$ 번째 계층의 노드는 $(s_i, t_i) \in L_i$ 에 속한 야드베이 쌍 후보로 구성된다. 마지막으로 TC의 목적지 계층 $L_{m+1} = \{(s_{m+1}, t_{m+1})\}$ 라 하고, 가상야드베이 $n+1$ 을 도입하여 $s_{m+1}=n+1, t_{m+1}=n+1$ 로 나타내기라 한다. $N=(L, E)$ 에서 아크의 집합 E 에 속한 아크는 인접 계층 간에만 정의된다. $1 \leq i \leq m$ 에 대한 노드 $(s_{i-1}, t_{i-1}) \in L_{i-1}$ 와 노드 $(s_i, t_i) \in L_i$ 간의 아크의 길이는 노드 (s_{i-1}, t_{i-1}) 에서 TC가 마지막으로 방문한 야드베이 t_{i-1} 로부터 노드 (s_i, t_i) 의 처음으로 방문하는 야드베이 s_i 로의 이동거리와 s_i 에서 t_i 까지의 이동거리의 합으로 나타내기로 한다. 이때 t_{i-1} 에서 s_i 로의 이동거리는 $|t_{i-1} - s_i|$ 이고, s_i 에서 t_i 로의 이동거리는 Property 1에 의해 $|s_i - t_i|$ 가 된다. 그리고 계층 m 에 속한 노드 $(s_m, t_m) \in L_m$ 에서 목적지 노드 $(s_{m+1}, t_{m+1}) \in L_{m+1}$ 로 가는 아크의 길이는 0으로 한다. 다음의 [그림 4]는 이렇게 구성된 계층 네트워크 $N=(L, E)$ 를 보여주고 있다.

Property 2로부터 $N=(L, E)$ 의 노드의 개수는 출발지 및 목적지를 포함하여 최대 $n^2/|G|+2$ 이므로 $O(n^2)$ 가 된다. 또한 $|G|=2$ 인 경우에 두 그룹간 아크



[그림 4] 계층네트워크 $N=(L, E)$

의 개수는 $\frac{n}{2} \times \frac{n}{2}$ 가 되며, 이는 $N=(L, E)$ 에서 아크의 개수가 최대가 되는 네트워크가 됨을 쉽게 보일 수 있다. 따라서 $N=(L, E)$ 에서 아크의 개수는 $O(n^2)$ 가 된다.

이제 TC의 운행경로를 $N=(L, E)$ 에서 (s_0, t_0) 을 출발지로 하여 목적지 (s_{m+1}, t_{m+1}) 로 나타내기 위해, $(s_{i-1}, t_{i-1}) \in L_{i-1}$ 에서 $(s_i, t_i) \in L_i$ 로 가는 아크의 흐름변수를 $f_{(s_{i-1}, t_{i-1}), (s_i, t_i)}$ 라 하자. TC의 운행경로는 $N=(L, E)$ 에서 하나의 경로에 해당되므로, 다음의 흐름보존제약(flow conservation constraints)을 만족하여야 한다.

(TC 운행경로의 흐름 보존제약)

$$\sum_{(s_r, t_r) \in L_i} f_{(s_r, t_r), (s_i, t_i)} = 1 \quad (5)$$

$$\sum_{(s_{i-1}, t_{i-1}) \in L_{i-1}} f_{(s_{i-1}, t_{i-1}), (s_i, t_i)} \quad (6)$$

$$- \sum_{(s_{i+1}, t_{i+1}) \in L_{i+1}} f_{(s_i, t_i), (s_{i+1}, t_{i+1})} = 0, \text{ for}$$

$$(s_i, t_i) \in L_i, 1 \leq i \leq m$$

$$\sum_{(s_m, t_m) \in L_m} f_{(s_m, t_m), (s_{m+1}, t_{m+1})} = 1 \quad (7)$$

$k \in G$ 에 대해

$$L_i(\{j\}) = \{(s_i, t_i) \in L_i : j \in J^k(s_i, t_i)\}$$

$$\text{for } i \in I^k, j \in J^k$$

라 하자. 즉 $i \in I^k$ 에 대한 $L_i(\{j\}) \subseteq L_i$ 는 야드베이 $j \in J^k$ 를 포함하는 야드베이 쌍 후보 $(s_i, t_i) \in L_i$ 의 집합이다. 그러면 $i \in I^k$ 의 수요 a_i 중에서 $j \in J^k$ 에 할당되는 컨테이너의 수 x_{ij} 와 $L_i(\{j\})$ 간에는

$$x_{ij} \leq a_i \sum_{(s_r, t_r) \in L_i(\{j\})} \sum_{(s_{i-1}, t_{i-1}) \in L_{i-1}} f_{(s_{i-1}, t_{i-1}), (s_i, t_i)}, \quad (8)$$

$$\text{for } i \in I^k, j \in J^k$$

가 성립하게 된다. 위의 논의로부터 TCRP를 정식화 하여, $(P(f, x, y))$ 라 하자.

$$(P(f, x, y))$$

$$\min z = \alpha \sum_{i=1}^m \sum_{(s_r, t_r) \in L_i} \sum_{(s_{i-1}, t_{i-1}) \in L_{i-1}} (|t_{i-1} - s_i| + |s_i - t_i|) \quad (C0)$$

$$+ \beta \sum_{k \in G} \sum_{i \in I^k} \sum_{j \in J^k} y_{ij} \quad (C0)$$

$$\text{s.t.} \quad \sum_{(s_r, t_r) \in L_i} f_{(s_r, t_r), (s_i, t_i)} = 1 \quad (C1)$$

$$\sum_{(s_{i-1}, t_{i-1}) \in L_{i-1}} f_{(s_{i-1}, t_{i-1}), (s_i, t_i)} - \sum_{(s_{i+1}, t_{i+1}) \in L_{i+1}} f_{(s_i, t_i), (s_{i+1}, t_{i+1})} = 0,$$

$$\text{for } (s_i, t_i) \in L_i, 1 \leq i \leq m \quad (C2)$$

$$\sum_{(s_m, t_m) \in L_m} f_{(s_m, t_m), (s_{m+1}, t_{m+1})} = 1 \quad (C3)$$

$$x_{ij} \leq a_i \sum_{(s_r, t_r) \in L_i(\{j\})}$$

$$\sum_{(s_{i-1}, t_{i-1}) \in L_{i-1}} f_{(s_{i-1}, t_{i-1}), (s_i, t_i)} \text{ for } i \in I^k,$$

$$j \in J^k, k \in G \quad (C4)$$

$$x_{ij} \leq \min\{a_i, b_j\} y_{ij}, \text{ for } i \in I^k,$$

$$j \in J^k, k \in G \quad (C5)$$

$$\sum_{j \in J^k} x_{ij} = a_i, \text{ for } i \in I^k, k \in G \quad (C6)$$

$$\sum_{i \in I^k} x_{ij} = b_j, \text{ for } j \in J^k, k \in G \quad (C7)$$

$$f_{(s_{i-1}, t_{i-1}), (s_i, t_i)} \in \{0, 1\}, \text{ for}$$

$$(s_{i-1}, t_{i-1}) \in L_{i-1}, (s_i, t_i) \in L_i,$$

$$1 \leq i \leq m+1 \quad (C8)$$

$$x_{ij} \geq 0, \text{ for } i \in I^k, j \in J^k, k \in G \quad (C9)$$

$$y_{ij} \in \{0, 1\}, \text{ for } i \in I^k, j \in J^k,$$

$$k \in G \quad (C10)$$

$(P(f, x, y))$ 에서 f, x, y 는 제약식 (C8), 제약식 (C9), 제약식 (C10)의 의사결정 변수를 벡터로 나타낸 것이다. $y_{ij} \in \{0, 1\}$ 는 $i \in I^k$ 의 수요 a_i 를 위해 야드베이 $j \in J^k$ 에서 컨테이너의 픽업이 발생하면 1, 아니면 0을 나타내는 이진변수로서 제약식 (C5)에 의해 결정된다. 목적함수 (C0)에서 첫 번째 항은 TC 운행경로의 길이 $|t_{i-1} - s_i| + |s_i - t_i|$ 에 비례하여 α 만큼의 시간이 소요되는 운행경로 시간의 합을 나타내고, 두 번째 항은 픽업을 위해 방문하는 야드베이의 수에 비례하

여 β 만큼 소요되는 작업준비시간의 합을 나타내고 있다. 편의상 $(P(f, x, y))$ 에서 TC 운행경로의 길이를 $T(f, x, y)$ 라 하고, 픽업을 위해 방문하는 야드베이의 수를 $S(f, x, y)$ 라 하면

$$T(f, x, y) = \sum_{i=1}^m \sum_{(s_i, t_i) \in L} \sum_{L(s_i, t_i) \in L} (|t_{i-1} - s_i| + |s_i - t_i|) f_{(s_i, t_i)(s_i, t_i)}$$

이고

$$S(f, x, y) = \sum_{k \in G} \sum_{i \in I} \sum_{j \in J} y_{ij}$$

가 된다. 그러면 $(P(f, x, y))$ 의 목적함수는 $z = \alpha \cdot T(f, x, y) + \beta \cdot S(f, x, y)$ 가 된다. $(P(f, x, y))$ 에서 의사결정 변수 y 를 제외시킨 문제를 $(P(f, x))$ 라 하면,

$$\begin{aligned} (P(f, x)) \quad \min \quad T(f, x) &= \sum_{i=1}^m \sum_{(s_i, t_i) \in L} \sum_{L(s_i, t_i) \in L} (|t_{i-1} - s_i| + |s_i - t_i|) f_{(s_i, t_i)(s_i, t_i)} \\ \text{s.t.} \quad (C1), (C2), (C3), (C4), \\ &(C6), (C7), (C8), (C9) \end{aligned}$$

이다. $(P(f, x))$ 는 TC가 픽업을 위해 방문하는 야드베이의 수를 고려하지 않고, 그룹별 수요-공급 균형제약을 만족시키는 TC의 최단 운행경로를 찾는 문제가 된다. 그리고 $(P(f, x, y))$ 에서 의사결정 변수 f 를 제외시킨 문제를 $(P(x, y))$ 라 하면,

$$\begin{aligned} (P(x, y)) \quad \min \quad S(x, y) &= \sum_{k \in G} \sum_{i \in I} \sum_{j \in J} y_{ij} \\ \text{s.t.} \quad (C5), (C6), (C7), (C9), (C10) \end{aligned}$$

이다. $(P(x, y))$ 는 TC의 운행경로에 관계없이, 그룹별 수요-공급 균형제약을 만족시키면서 TC가 픽업을 위해 방문하는 야드베이의 수를 최소화하는 문제가 된다. $(P(x, y))$ 는 $k \in G$ 에 대해서 분할되므로, k 에

대해 분할된 문제를 $(P_k(x, y))$ 라 하고,

$$S_k(x, y) = \sum_{i \in I} \sum_{j \in J} y_{ij}$$

라 하자. $(P(f, x, y))$ 의 최적해를 f^*, x^*, y^* 라 하고, $(P(f, x))$ 의 최적해를 f', x' 라 하고, $(P(x, y))$ 의 최적해를 x'', y'' 라 하면, 다음의 정리가 성립한다.

Property 3 : $T(f', x') \leq T(f^*, x^*, y^*)$

Property 4 : $S(x'', y'') \leq S(f^*, x^*, y^*)$

Property 3과 4로부터 다음과 같이 $(P(f, x, y))$ 에 대한 하한을 정의할 수 있다.

Lemma 1 : $\alpha \cdot T(f', x') + \beta \cdot S(x'', y'') \leq \alpha \cdot T(f^*, x^*, y^*) + \beta \cdot S(f^*, x^*, y^*)$

$(P(f, x))$ 의 최적해 f', x' 가 주어졌을 때, y' 을

$$y'_{ij} = \begin{cases} 1, & \text{if } x'_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

라 하면, x' 과 y' 은 제약식 (C5)와 (C10)을 만족하게 된다. 따라서 f', x', y' 은 $(P(f, x, y))$ 의 실행가능해가 되며,

$$S(f', x', y') = \sum_{k \in G} \sum_{i \in I} \sum_{j \in J} y'_{ij}$$

라 하면, 다음이 성립하게 된다.

Lemma 2 : $\alpha \cdot T(f^*, x^*, y^*) + \beta \cdot S(f^*, x^*, y^*) \leq \alpha \cdot T(f', x') + \beta \cdot S(f', x', y')$.

Lemma 1과 2로부터 $(P(f, x, y))$ 의 하한과 상한을 구하려면, $(P(f, x))$ 의 최적해 f', x' 와 $(P(x, y))$ 의 최적해 x'', y'' 를 구하는 2가지의 문제로 분할됨을 알

수 있다. 본 연구에서는 이 하한과 상한이 매우 효과적임을 보이고, 복잡한 $(P(f, x, y))$ 를 직접 다루지 않고 $(P(f, x))$ 와 $(P(x, y))$ 를 이용한 하한과 상한을 제시하도록 하겠다.

3. 문제의 특성 분석

3.1 TCRP의 복잡성 분석

TCRP에 대한 NP-complete 여부는 현재까지 알려져 있지 않다. 본 연구에서는 완전분할문제(the partitioning problem)를 이용하여 $(P(f, x, y))$ 와 $(P_k(x, y))$ 가 NP-complete임을 보이도록 하겠다.

Property 5 : $(P(f, x, y))$ 는 NP-complete이다.

(증명) 완전분할문제는 정수의 집합 $\{c_1, c_2, \dots, c_r\}$ 과 인덱스 집합 $N = \{1, 2, \dots, r\}$ 이 주어질 때, $\sum_{j \in X} c_j = \sum_{j \in M, X} c_j$ 를 만족하는 $X \subset N$ 가 존재하는지를 알아보는 문제이다. [그림 5]처럼 4개의 컨테이너 그룹, 5개의 작업일정, $r+3$ 개의 공급 야드베이가 있는 문제를 고려해 보자.

작업 일정	1	2	3	4	5
컨테이너 그룹	2	1	3	1	4
컨테이너 수요	a	$\sum_{i \in N} c_i/2$	a	$\sum_{i \in N} c_i/2$	a

(a) 작업일정 및 수요

야드베이 번호	j_1	j_2	j_3	j_4	j_{r+2}	j_{r+3}
컨테이너 그룹	2	4	1	1	...	1
공급량	a	a	c_1	c_2	...	c_r

(b) 공급 야드베이의 컨테이너 분포

[그림 5] TCRP의 예

[그림 5]에서 5개의 작업일정은 수요 $a_1 = a, a_2 = \sum_{i \in N} c_i/2, a_3 = a, a_4 = \sum_{i \in N} c_i/2, a_5 = a$ 를 가지고, 이들이 속하는 그룹은 $I^1 = \{2, 4\}, I^2 = \{1\}, I^3 = \{3\}, I^4 = \{5\}$

이다. 공급이 있는 $r+3$ 개의 야드베이는 $b_{j_1} = b_{j_2} = a$ 와 $1 \leq p \leq r$ 에 대해 $b_{j_{r+2}} = c_p, b_{j_{r+3}} = a$ 를 공급할 수 있고, 이들이 속한 그룹은 $J^1 = \{j_3, j_4, \dots, j_r, j_{r+1}, j_{r+2}\}, J^2 = \{j_2\}, J^3 = \{j_{r+3}\}, J^4 = \{j_2\}$ 이다.

그룹 2, 3, 4는 작업일정과 공급 야드베이가 각각 하나씩이므로, 작업일정 $a_1 = a$ 를 위한 j_1 의 방문, $a_2 = \sum_{i \in N} c_i/2$ 의 수요를 만족시키는 $Y \subset J^1$ 에 속한 야드베이의 방문, $a_3 = a$ 를 위한 j_3 의 방문, $a_4 = \sum_{i \in N} c_i/2$ 의 수요를 만족시키는 $J^1 \setminus Y$ 에 속한 야드베이의 방문, $a_5 = a$ 를 위한 j_4 로의 방문 순서를 지킬 때 TC의 운행 경로의 길이는 $|j_1 - j_{r+3}| + |j_{r+3} - j_2|$ 가 되어 최소가 된다. 만약 완전분할에서 $\sum_{j \in X} c_j = \sum_{j \in M, X} c_j$ 를 만족하는 $X \subset N$ 가 존재할 때, $Y = \{j_{i+2} : i \in X\}$ 라 하자. 그러면 픽업을 위해 방문하는 야드베이의 수는 $r+3$ 으로 주어진 야드베이의 수와 동일하게 되어 최소가 된다. 이때 목적함수는 $\alpha(|j_1 - j_{m+3}| + |j_{m+3} - j_2|) + \beta(m+3)$ 으로 최소가 된다. 반대로 TCRP의 목적함수가 $\alpha(|j_1 - j_{m+3}| + |j_{m+3} - j_2|) + \beta(m+3)$ 이면, 완전분할을 만족하는 $X = \{p : j_{p+2} \in Y\}$ 가 존재함을 보일 수 있다. 반면에 완전분할이 존재하지 않는 경우와 TCRP의 작업준비 횟수는 $m+4$ 가 되어 목적함수는 $\alpha(|j_1 - j_{m+3}| + |j_{m+3} - j_2|) + \beta(m+4)$ 가 되는 경우가 동일함을 쉽게 보일 수 있다. 따라서 TCRP는 완전분할문제로부터 NP-complete임을 알 수 있다. ■

Property 6 : $(P_k(x, y))$ 는 NP-complete이다.

(증명) 그룹 $k \in G$ 에 대한 작업일정의 집합은 $I^k = \{1, 2\}$ 이고, $a_1 = a_2 = \sum_{j \in J^k} b_j/2$ 라 하자. 이때 완전분할이 존재하는 경우와 작업준비의 횟수가 $|J^k|$ 인 경우가 동일함을 보일 수 있고, 완전분할이 존재하지 않는 경우와 작업준비의 횟수가 $|J^k|+1$ 인 경우가 동일함을 보일 수 있다. ■

$(P_k(x, y))$ 는 그룹별 수요-공급 균형 제약을 가지므로, 수송문제와 밀접한 관련을 가지고 있음을 알 수 있다. 수송문제의 기저가능해(Basic Feasible Solution,

BFS)와 관련하여 알려진 성질들로부터 다음을 유도할 수 있다[2].

Property 7 : $\max\{|I^k|, |J^k|\} \leq S_k(x, y) \leq |I^k| + |J^k| - 1$

Property 8 : 퇴화기저해(degenerate BFS)가 존재하지 않는 경우, $S_k(x'', y'')$ 는 $|I^k| + |J^k| - 1$ 이다.

퇴화기저해는 $I_0^k \subset I^k, J_0^k \subset J^k$ 에 대해

$$\sum_{i \in I_0^k} a_i = \sum_{j \in J_0^k} b_j$$

인 I_0^k 와 J_0^k 가 존재하는 경우에 발생한다[2]. Property 6으로부터 $(P_k(x, y))$ 에서 퇴화기저해의 존재여부를 검증하는 것 자체가 NP-complete가 됨을 알 수 있다. 가능한 퇴화기저해 중에서 $S_k(x, y)$ 를 최소화하는 퇴화기저해를 찾으려면, 모든 열과 행들의 조합을 구하여야 되므로 $O(|I^k| \cdot |J^k|)$ 만큼의 계산시간을 요구하게 된다. 비록 $(P_k(x, y))$ 가 NP-complete이지만, $(P_k(x, y))$ 는 그룹 $k \in G$ 별로 분할된 문제이므로 이의 최적해는 IP 패키지를 이용하면 쉽게 구할 수 있다. 물론 보다 체계적인 해법의 개발이 가능할 것으로 보이지만, 본 연구에서는 $(P_k(x, y))$ 의 최적해를 IP 패키지를 이용하여 구하기로 한다.

3.2 $(P(f, x))$ 의 해법 개발

Lemma 1과 Lemma 2에 의해 $(P(f, x, y))$ 의 하한과 상한은 $(P(f, x))$ 를 이용하여 쉽게 구할 수 있음을 알 수 있었다. 본 절에서는 열생성기법(column generation method)을 이용하여 $(P(f, x))$ 에 대한 최적해를 구하는 방안을 제시하도록 한다.

$(P(f, x))$ 에서 (C8)을 선형완화한 문제를 $(\bar{P}(f, x))$ 라 하고, $(\bar{P}(f, x))$ 에서 x 와 관련된 제약식 (C4), (C6), (C7), (C9)을 제외한 문제를 $(\bar{P}(f))$ 라 하자. 즉 $(\bar{P}(f))$ 는 TC의 운행경로만을 고려하고, 수요-공급 제약은

제외시킨 문제이다.

$(\bar{P}(f))$ 에서, 편의상 $i=1, 2, \dots, m+1$ 번째 계층의 노드 $(s_i, t_i) \in L_i$ 로 들어오는 흐름의 합을

$$F_i(s_i, t_i) = \sum_{(s_r, t_r) \in L_{i-1}} f_{(s_r, t_r), (s_i, t_i)} \quad (10)$$

라 하면, $0 \leq F_i(s_i, t_i) \leq 1$ 와 $\sum_{(s_r, t_r) \in L_i} F_i(s_i, t_i) = 1$ 를 만족한다. 이때 $k \in G$ 에 대해, 야드베이 $j \in J^k$ 를 포함하는 야드베이 쌍 후보 중에서 적어도 하나 이상은 선택되어야 하므로, $L_i(\{j\})$ 의 정의에 의해

$$\sum_{i \in I^k} \sum_{(s_r, t_r) \in L_i(\{j\})} F_i(s_i, t_i) \geq 1, \text{ for } j \in J^k \quad (V1)$$

가 성립하여야 된다.

(C4)에 의해

$$x_{ij} \leq a_i \sum_{(s_r, t_r) \in L_i(\{j\})} F_i(s_i, t_i), \text{ for } i \in I^k, \\ j \in J^k, k \in G$$

이므로 그룹 $k \in G$ 의 작업일정 $i \in I^k$ 에 대해 $F_i(s_i, t_i)$ 가 주어지면 수요 a_i 중에서 $a_i \cdot F_i(s_i, t_i)$ 는 $j \in J^k(s_i, t_i) \subseteq J^k$ 에 속한 야드베이에서만 처리될 수 있음을 나타낸다. 따라서 그룹 $k \in G$ 의 작업일정 $i \in I^k$ 에 대해 $(\bar{P}(f))$ 에서 $F_i(s_i, t_i)$ 가 주어졌을 때, (C4), (C6), (C7), (C9)를 만족하는 x 가 존재하는지를 알아보는 문제는 다음의 Separation_Problem(k, f)로 정의할 수 있다.

Separation_Problem(k, f)

$$\sum_{j \in J^k(s_i, t_i)} x_{ij} = a_i F_i(s_i, t_i), \text{ for } (s_i, t_i) \in L_i, \\ i \in I^k$$

$$\sum_{i \in I^k} x_{ij} = b_j, \text{ for } j \in J^k$$

$$x_{ij} \geq 0, \text{ for } i \in I^k, j \in J^k$$

where $F_i(s_i, t_i) = \sum_{(s_r, t_r) \in L_{i-1}} f_{(s_r, t_r), (s_i, t_i)}$

Separation_Problem(k, f)는 주어진 $k \in G$ 와 f 에

대한 실행가능해 x 를 찾는 문제로서, 이는 bipartite graph에서의 최대흐름문제(Bipartite Flow Feasibility Problem, BFFP)라고도 한다[3]. 만약 Separation_Problem(k, f)를 만족하는 x 가 존재하면 최대흐름은 $\sum_{j \in J'} b_j$ 가 되고, 그렇지 않으면 최대흐름은 $\sum_{j \in J'} b_j$ 보다 작은 값을 가진다. BFFP에 대해서는 Cezik과 Gunluk [3]에 의해 많은 특성들이 연구되었는데, 특히 그들은 Separation_Problem(k, f)와 동일한 문제에 대해 x 가 존재하기 위한 필요충분조건을 Lemma 3과 같이 보였다.

$i \in I^k$ 와 $J' \subset J^k$ 에 대해 $L_i(J') = \{(s_i, t_i) \in L_i : J^k(s_i, t_i) \subseteq J', F_i(s_i, t_i) > 0\}$ 라 하자.

Lemma 3 : (Cezik과 Gunluk[3]) Separation_Problem(k, f)에서 x 가 존재하기 위한 필요충분조건은

$$\sum_{i \in I^k} \sum_{(s_i, t_i) \in L_i(J')} a_i F_i(s_i, t_i) \leq \sum_{j \in J'} b_j, \quad J' \subset J^k \quad (11)$$

이다.

Lemma 3은 Separation_Problem(k, f)를 위해 bipartite graph에서 최대흐름을 직접 구할 필요가 없이 식 (11)을 만족하지 않는 $J' \subset J^k$ 를 찾으면 된다는 것이다. [3]에서는 Separation_Problem(k, f)에서의 실행가능해 x 가 존재할 때의 최대흐름 값을 찾는 Max_Flow와 x 가 존재하지 않을 때 식 (11)을 위반하는 $J' \subset J^k$ 를 찾는 Min_Cut을 제안하고 있는데, 이들 해법의 수행시간은 $O(p|J^k|)$ (여기서 $p = \sum_{i \in I^k} |\{(s_i, t_i) \in L_i : F_i(s_i, t_i) > 0\}|$)으로 bipartite graph에서의 최대흐름해법보다 매우 효과적이다.

Separation_Problem(k, f)에서의 실행가능해 x 가 존재하지 않는 경우 Lemma 3에 의해 (V2)를 만족하지 않는 어떤 $J' \subset J^k$ 이 존재하게 되고, 이를 $J_0^k \subset J^k$ 라 하자. 그리고 $I(J_0^k) = \{i \in I^k : L_i(J_0^k) \neq \emptyset\}$ 라 하면, Lemma 3에 의해

$$\sum_{i \in I(J_0^k)} \sum_{(s_i, t_i) \in L_i(J_0^k)} a_i \cdot F_i(s_i, t_i) > \sum_{j \in J_0^k} b_j \quad (12)$$

가 성립하게 된다. 즉 (12)에 의해

$$\sum_{i \in I(J_0^k)} \sum_{(s_i, t_i) \in L_i(J_0^k)} a_i \cdot F_i(s_i, t_i) \leq \sum_{j \in J_0^k} b_j \quad (V2)$$

는 $(P(f, x))$ 에 대한 유효부등식이 된다.

또한 식 (12)에 의해 $i \in I(J_0^k)$ 에 속한 작업일정 중 적어도 하나는 J_0^k 의 공급을 이용할 수 없으므로

$$\sum_{i \in I(J_0^k)} \sum_{(s_i, t_i) \in L_i(J_0^k)} F_i(s_i, t_i) \leq |I(J_0^k)| - 1, \quad J_0^k \subset J^k \quad (V3)$$

가 성립하게 된다. 이때 식 (V3)의 경우,

$$\sum_{i \in I(J_0^k)} \sum_{(s_i, t_i) \in L_i(J_0^k)} F_i(s_i, t_i) > |I(J_0^k)| - 1 \quad (13)$$

를 만족하지 않으면, 식 (V3)는 주어진 f 에 대해 redundant한 제약식이 됨에 주의하자.

다음은 $(\bar{P}(f, x))$ 의 최적해를 위에서 정의한 유효부등식을 이용하여 열생성기법으로 구하는 절차이다.

Procedure Column_Generation_of_ $(\bar{P}(f, x))$

Step 0 $(\bar{P}(f))$ 에 (V1)를 추가한 문제를 $(\bar{P}(f))$ 로 설정

Step 1 : $(\bar{P}(f))$ 의 최적해 \bar{f} 를 LP 패키지로 구함

Step 2 : (i) Separation_Problem(k, \bar{f})의 실행가능해 \bar{x} 가 존재하지 않는 모든 $k \in G$ 에 대해 식 (13)을 만족하면 $(\bar{P}(f))$ 에 식 (V3)를 추가하고, 그렇지 않으면 $(\bar{P}(f))$ 에 식 (V2)를 추가

(ii) 모든 $k \in G$ 에 대한 Separation_Problem(k, \bar{f})의 실행가능해 \bar{x} 가 존재하면 Stop. 그렇지 않으면 Step 1으로 이동

Column_Generation_of_ $(\bar{P}(f, x))$ 는 식 (12)를 만

족하는 $J_0 \subset J^*$ 가 존재하면 Step 2에서 $J_0 \subset J^*$ 에 대해 유효부등식 식 (V2) 또는 식 (V3)을 추가하게 되면, 다음 반복에서는 $J_0 \subset J^*$ 는 식 (11)을 만족하게 된다. 이러한 과정을 계속 반복하면 모든 가능한 $J' \subset J^*$ 에 대해 식 (11)을 만족하게 되고, 이 경우 Lemma 3에 의해 실행가능해 x 가 반드시 존재하게 된다. 따라서 Column_Generation_of_ $(\bar{P}(f, x))$ 는 유한한 반복 내에 $(\bar{P}(f, x))$ 의 최적해를 구할 수 있게 된다.

만약 Column_Generation_of_ $(\bar{P}(f, x))$ 의 최종 실행 후의 \bar{f} 가 식 (C8)의 정수제약조건을 만족하면 이는 $(P(f, x))$ 의 최적해가 된다. 그렇지 않고 \bar{f} 가 정수제약조건을 만족하지 못하면 $(\bar{P}(f))$ 에 식 (C8)을 추가한 문제 $(P(f))$ 라 하자. 다음은 $(P(f))$ 를 이용하여 $(P(f, x))$ 의 최적해를 열생성기법으로 구하는 절차이다.

Procedure Column_Generation_of_ $(P(f, x))$

- Step 1 : $(P(f))$ 의 최적해 f' 를 IP 패키지로 구함
- Step 2 : (i) Separation_Problem(k, f')의 실행가능해 x' 가 존재하지 않는 모든 $k \in G$ 에 대해 (V3)를 추가
- (ii) 모든 $k \in G$ 에 대한 Separation_Problem(k, f')의 실행가능해 x' 가 존재하면 Stop. 그렇지 않으면 Step 1으로 이동

$(P(f))$ 는 f 에 대한 정수제약조건 식 (C8)이 추가되어 있으므로, $i \in I(J_0^k)$ 에 대해 $L_i(J_0^k)$ 는 단 하나의 원소만으로 구성된다. 따라서 $J_0^k \subset J^*$ 에 대해

$$\sum_{i \in I(J_0^k)} \sum_{(s_i, t_i) \in L_i(J_0^k)} F_i(s_i, t_i) = |I(J_0^k)|, \quad J_0^k \subset J^*$$

가 되어, 식 (13)을 항상 만족하게 된다. 따라서 $J_0 \subset J^*$ 에 대해 식 (V3)는 항상 tight한 제약식이 되므로, Column_Generation_of_ $(P(f, x))$ 의 Step 2의 (i)은 식 (V2)를 제외하고 식 (V3)만을 고려하면 되고, 유한한 반복 내에 $(P(f, x))$ 의 최적해를 구할 수 있게 된다.

Column_Generation_of_ $(P(f, x))$ 에 의해 $(P(f, x))$ 의 최적해 f' 와 x' 가 주어지면, Lemma 1을 위한 $T(f', x')$ 을 구할 수 있다. 또한 f' 와 x' 로부터 식 (9)의 절차를 통해 $(P(f, x, y))$ 의 실행가능해 f', x', y' 와 $S(f', x', y')$ 를 구할 수 있다. Property 6에 의해 $(P_k(x, y))$ 은 NP-complete이지만, $k \in G$ 로 분할된 $(P_k(x, y))$ 는 매우 작은 크기의 문제이므로 IP 패키지를 이용하면 최적해를 쉽게 구할 수 있고, 이로부터 $S(x'', y'')$ 를 구할 수 있다. 다음은 Lemma 1과 2에서 정의된 하한과 상한이 매우 효과적임을 보여주고 있다.

$$\text{Lemma 4 : } \frac{\alpha \cdot T(f', x') + \beta \cdot S(f', x', y')}{\alpha \cdot T(f', x') + \beta \cdot S(x'', y'')} < 2$$

(증명) Property 6으로부터 $\max\{|I^k|, |J^k|\} \leq S_k(x, y) \leq |I^k| + |J^k| - 1$ 이고, 마찬가지로 $S(f', x', y')$ 는 수송 문제의 실행가능해이므로 $\max\{|I^k|, |J^k|\} \leq S_k(f, x, y) \leq |I^k| + |J^k| - 1$ 이 성립한다.

그리고 $\max\{m, n\} \leq \sum_{k \in G} \max\{|I^k|, |J^k|\}$ 이고, $\sum_{k \in G} (|I^k| + |J^k| - 1) = m + n - |G|$ 이므로

$$\frac{\alpha \cdot T(f', x') + \beta \cdot S(f', x', y')}{\alpha \cdot T(f', x') + \beta \cdot S(x'', y'')} \leq \frac{\alpha \cdot T(f', x') + \beta \cdot (m + n - |G|)}{\alpha \cdot T(f', x') + \beta \cdot \max\{m, n\}} \quad (14)$$

이다. 그러면 $m \leq n$ 인 경우, 식 (14)는

$$\frac{\alpha \cdot T(f', x') + \beta \cdot (m + n - |G|)}{\alpha \cdot T(f', x') + \beta \cdot \max\{m, n\}} \leq \frac{\alpha \cdot T(f', x') + 2\beta n}{\alpha \cdot T(f', x') + \beta n} \quad (15)$$

가 됨을 성립한다. 또한 $m > n$ 인 경우에도, 식 (14)는

$$\frac{\alpha \cdot T(f', x') + \beta \cdot (m + n - |G|)}{\alpha \cdot T(f', x') + \beta \cdot \max\{m, n\}} \leq \frac{\alpha \cdot T(f', x') + 2\beta m}{\alpha \cdot T(f', x') + \beta m} \quad (16)$$

이다. 따라서 Lemma가 성립하게 된다. ■

4. 계산 결과

Kim과 Kim[6]의 논문에서 제시된 문제를 대상으로, 본 연구에서 제시한 하한 및 상한이 효율적임을 보이도록 하겠다. [6]에서는 TC의 출발지 야드를 0, 인접 야드베이간 이동시간 $\alpha=1$, 야드베이에서의 작업 준비시간 $\beta=2$ 에 대하여, 야드베이의 분포(ANBG), 작업일정의 수(NPT), 야드베이의 수(RL)를 고려한 실험을 제시하고 있다. [6]에서는 n 에 해당되는 RL은 25, 35, 45, m 에 해당하는 NPT는 10, 12, 14, 16, ANBG는 2.4, 2.8, 3.0, 3.4의 값을 가지며, 따라서 가능한 실험의 수는 $(RL \cdot NPT \cdot ANBG) = 4 \cdot 4 \cdot 3 = 48$ 가지이지만, 이 중에서 24가지에 대해서만 실제 실험을 수행하였다. 각 실험들을 구분하기 위해 (RL, NPT, ANBG)로 나타내기로 한다.

본 연구에서도 [6]과 동일한 실험 데이터를 이용하여, 본 연구에서 제안한 하한과 상한을 구하였다. 이를 위해 $Column_Generation_of_(\bar{P}(f, x))$ 에서의 선형 계획법 문제 $(\bar{P}(f))$ 의 최적해와 $Column_Generation_of_(\rho(f, x))$ 에서의 정수계획법 문제인 $(\rho(f))$ 의 최적해는 lp_solve 5.5.0.5라는 공개 소프트웨어를 사용하였고[10], Pentium-4 2.8GHz의 PC에서 C++로 코딩하여 실행하였다.

<표 1>은 Kim과 Kim[6]의 실험데이터에 대한 계산결과를 보여주고 있다. [6]에서 24가지 경우에 대해 제시한 실험결과는 <표 1>의 opt와 time1 열에 나와 있는데, opt 열은 $(\rho(f, x, y))$ 에 대한 최적해에서 “(TC의 운행시간, 야드베이의 방문 회수), 최적해”를 보여주고 있고, time1 열은 수행시간을 보여 주고 있다. [6]에서 언급하였듯이 실험결과를 보면 RL, NPT, ANBG의 값이 증가하면 실험시간이 증가하고 있으며, 실험 (25, 16, 3.0) 및 (25, 16, 3.4)의 경우에는 최적해를 구하지 못하고 있다. 이러한 실험결과를 보여주는 것은 [6]에서 제안한 해법이 $k \in G$ 에 대한 수송문제에서 모든 가능한 기저가능해의 탐색에 기반을 두고 있기 때문이다.

<표 1>에서 본 연구의 실험결과와 관련된 항목들을 살펴보자. 목적함수와 관련하여 S는 $s(x'', y'')$, S(IP)

는 $S(f', x', y')$, T(IP)는 $T(f', x')$, LB는 $\alpha \cdot T(f', x') + \beta \cdot S(x'', y'')$, UB는 $\alpha \cdot T(f', x') + \beta \cdot S(f', x', y')$ 를 나타내고 있다. 그리고 itr는 반복수, cut 1과 cut 2는 $Column_Generation_of_(\bar{P}(f, x))$ 에서 식 (V3)과 식 (V2)의 추가 횟수, cut 3은 $Column_Generation_of_(\rho(f, x))$ 에서 식 (V2)의 추가 횟수, gap1은 S(IP)-S, time2는 수행시간을 나타내고 있다. 본 연구에서는 48가지 모든 경우에 대하여 실험을 실시하였다. 계산 결과에서 평균수행시간은 약 2초 정도이고 최대수행시간은 12초로서 매우 안정적이며, [6]과 달리 수행시간은 RL, NPT, ANBG의 값과 관계없이 일정하였다. 또한 “UB-opt”열에서 보듯이 [6]에서 최적해를 구한 22개의 문제에서 본 연구에서 제안한 상한인 UB는 (25, 10, 3.4)와 (25, 16, 2.4)의 2가지 경우를 제외하고 모두 최적해와 동일한 값을 보여주고 있다. 그리고 48가지 문제에 대한 상한 UB와 하한 LB의 평균 겹은 1.48%이며 최대 겹은 4.19%를 보여주고 있어서, 상한과 하한의 차이가 매우 적음을 알 수 있다. 이러한 결과를 볼 때, 본 연구에서 제안한 하한 및 상한은 수행시간 및 효율성을 고려할 때 현실 문제에서의 적용이 용이함을 알 수 있다.

5. 결 론

본 연구에서는 기존에 알려지지 않았던 TCRP의 복잡성이 NP-complete임을 보였고, 이를 위한 새로운 정식화 방안을 제시하였다. 제안된 정식화의 특성 분석을 통해 효과적인 하한과 상한을 제시하였으며, 이 하한과 상한의 차이가 이론적 및 실험적으로 매우 작다는 것을 보였으며, 따라서 현실문제에도 쉽게 적용이 가능한 해법임을 보였다.

향후의 연구방향으로는 보다 다양한 현실 문제에 대한 해법의 적용 가능성을 확인하여야 할 것이다. 기존 [6]의 연구에서 제시한 가정들 중에서 현실을 반영하지 못하는 부분을 알아보고, 이러한 가정들을 완화한 문제에 대해서도 본 연구에서 제안한 해법을 적용할 수 있도록 하여야 할 것이다. 그리고 연구에서 제외한 유효부등식이 얼마나 효율적인지에 대한 이

론적 결과를 도출하고, 그 외에 다양한 유효부등식을 찾아서 보다 효과적인 해법의 개발에 활용하여야 할 것이다.

〈표 1〉 계산 결과

RL	NPT	ANBG	opt	time1	S	S(IP)	T(IP)	LB	UB	itr	cut1	cut2	cut3	gap1	UB-opt	UB-LB	time2
25	10	2.4	(133,17),167	<1	17	17	133	167	167	5	1	0	1	0	0	0	2
		2.8	(100,19),138	1	19	19	100	138	138	9	2	4	0	0	0	0	4
		3.0	(129,19),167	19	18	19	129	165	167	4	1	0	0	1	0	2	2
		3.4	(119,20),159	54	20	22	118	158	162	13	3	7	0	2	3	4	11
	12	2.4	(134,18),170	4	18	19	132	168	170	6	2	1	1	1	0	2	3
		2.8	(82,17),116	164	16	17	82	114	116	6	3	0	0	1	0	2	3
		3.0	(80,18),116	725	16	18	80	112	116	5	2	0	0	2	0	4	2
		3.4	(96,23),142	614	21	23	96	138	142	9	1	7	1	2	0	4	5
	14	2.4	(142,21),184	8	21	21	142	184	184	4	1	0	0	0	0	0	1
		2.8	(106,22),150	282	21	22	106	148	150	7	6	0	0	1	0	2	2
		3.0	(118,22),162	1580	21	22	118	160	162	6	2	2	2	1	0	2	3
		3.4	(125,24),173	13872	22	24	125	169	173	6	8	0	0	2	0	4	3
	16	2.4	(130,21),172	307	20	23	128	168	174	9	8	1	1	3	2	6	5
		2.8	(106,23),152	7859	21	24	104	146	152	8	8	0	0	3	0	6	4
		3.0	-		22	26	139	183	191	10	10	0	6	4	-	8	8
		3.4	-		25	27	130	180	184	6	8	0	0	2	-	4	4
35	10	2.4			17	17	159	193	193	9	0	6	0	0		0	5
		2.8			19	19	164	202	202	15	0	12	0	0		0	12
		3.0			18	19	148	184	186	4	1	0	0	1		2	2
		3.4			20	22	154	194	198	5	2	0	0	2		4	3
	12	2.4	(139,18),175	4	18	18	139	175	175	4	2	0	0	0	0	0	2
		2.8	(160,20),200	39	19	21	158	196	200	7	1	0	5	2	0	4	3
		3.0	(166,20),206	126	20	20	166	206	206	3	0	0	0	0	0	0	1
		3.4	(170,24),218	357	23	24	170	216	218	6	3	0	0	1	0	2	3
	14	2.4			18	19	150	186	188	4	2	0	0	1		2	<1
		2.8			20	21	158	198	200	5	2	0	0	1		2	<1
		3.0			21	23	182	224	228	7	6	0	0	2		4	1
		3.4			22	24	164	208	212	8	3	4	0	2		4	1
	16	2.4			20	22	160	200	204	8	9	1	0	2		4	1
		2.8			23	24	166	212	214	11	6	1	7	1		2	2
		3.0			23	26	178	224	230	11	11	2	2	3		6	2
		3.4			23	27	187	233	241	12	4	10	6	4		8	6
45	10	2.4			16	17	165	197	199	4	1	0	0	1		2	<1
		2.8			17	19	212	246	250	6	3	0	0	2		4	<1
		3.0			19	19	196	234	234	4	1	0	0	0		0	<1
		3.4			19	22	198	236	242	7	2	1	1	3		6	1

	12	2.4	(197,19),235	4	19	19	197	235	235	6	5	0	0	0	0	0	<1
		2.8	(202,20),242	57	19	20	202	240	242	6	4	0	0	1	0	2	1
		3.0	(174,21),216	162	20	21	174	214	216	6	5	0	0	1	0	2	1
		3.4	(225,24),273	500	22	24	225	269	273	7	6	0	0	2	0	4	1
	14	2.4			20	21	218	258	260	7	6	0	0	1		2	1
		2.8			20	22	203	243	247	12	2	7	0	2		4	1
		3.0			19	21	196	234	238	7	2	1	3	2		4	2
		3.4			23	25	223	269	273	7	7	0	0	2		4	1
	16	2.4			20	21	218	258	260	6	7	0	0	1		2	1
		2.8			21	22	228	270	272	7	11	0	0	1		2	1
		3.0			23	25	205	251	255	11	8	4	3	2		4	12
		3.4			25	28	251	301	307	11	6	3	3	3		6	3

참 고 문 헌

- [1] I. Adler, A. J., Hoffman, R. Shamir, "Monge and feasibility sequences in general flow problems," *Discrete Applied Mathematics*, Vol.44(1993), pp.21-38.
- [2] M., Bazaraa and J. Jarvis, *Linear Programming and Network*, John Wiley and Sons, (1992), pp.574-585.
- [3] T., Çezik and O. Günlük, "Reformulating linear programs with transportation constraints-With applications to workforce scheduling," *Naval Research Logistics*, Vol.51, No.2 (2003), pp.275-296.
- [4] C.F., Daganzo, "The crane scheduling problem," *Transportation Research B*, Vol.23 (1989), pp.159-175.
- [5] K.H. Kim, J.S., Kang, and K.R. Ryu, "A beam search algorithm for the load sequencing of outbound containers in port container terminals," *OR spectrum : Quantitative Approaches in Management*, Vol.26, No.1(2004), pp.93-116.
- [6] K.H., Kim and K.Y. Kim, "An optimal routing algorithm for a transfer crane in port container terminals," *Transportation Science*, Vol.33, No.1(1999), pp.17-33.
- [7] K.Y., Kim and K.H. Kim, "Heuristic algorithms for routing yard-side equipment for minimizing loading times in container terminals," *Naval research logistics*, Vol.50 No.5(2003), pp.498-514.
- [8] W.C., Ng and K.L. Mak, "Yard crane scheduling in port container terminals," *Applied Mathematical Modelling*, Vol.29 (2005), pp.263-276.
- [9] R.I., Peterkofsky and C.F. Daganzo, "A branch and bound solution method for the crane scheduling problem," *Transportation Research B*, Vol.24(1990), pp.159-172.
- [10] http://groups.yahoo.com/group/lp_solve/