



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

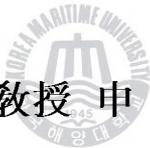
[Disclaimer](#)

物流學碩士 學位論文

효율적인 컨테이너 공로 운송을 위한
차량 경로 문제

**Vehicle Routing Problem for the Efficient
Container Road Transportation**

指導教授 申宰榮

The logo of Korea Maritime University is a circular emblem. It features a central stylized figure that resembles a ship's mast or a traditional Korean architectural element. The text "KOREA MARITIME UNIVERSITY" is written in a circular path around the top, and "해양대학교" (Korea Maritime University) is written in Korean at the bottom. The year "1945" is also visible within the emblem.

2010年 2月

韓國海洋大學校 大學院

物流시스템學科

吳 成 仁

本 論 文 을 吳 成 仁 의 物 流 學 碩 士 學 位 論 文 으 로 認 准 함 .

委 員 長 工 學 博 士 李 哲 榮 (인)

委 員 工 學 博 士 朴 眞 希 (인)

委 員 工 學 博 士 申 宰 榮 (인)



2009年 12月

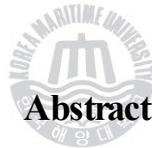
韓 國 海 洋 大 學 校 大 學 院
物 流 시 스템 學 科

吳 成 仁

Vehicle Routing Problem for the Efficient Container Road Transportation

Sung-Inn Oh

Department of Logistics
Graduate School of Korea Maritime University



At the present time, container transportation plays a key role in the international logistics. The efforts to increase the productivity of container logistics become essential for Korean trucking companies to survive in the domestic as well as global competition. This paper deals with VRP (Vehicle Routing Problem) of delivering container in Korean trucking industries. We examine the various characteristics of container transportation and then propose the mathematical models considering them. Several types of formulations are developed utilizing the standard formulations of well known TSP (Traveling Salesman Problem) and VRP. Heuristic algorithms and Tabu Search are also presented to solve the models. Efficiency of the proposed algorithms is verified through computational experiments.

목 차

ABSTRACT

제 1 장 서 론	1
1.1 연구의 배경 및 목적	1
1.2 관련 문헌 연구	2
1.3 논문의 구성	4
제 2 장 컨테이너 공로 운송의 특성	5
2.1 컨테이너	5
2.2 컨테이너 차량	5
2.2 컨테이너 공로 운송	6
제 3 장 문제 설정 및 수리적 모형	9
3.1 콤바인 차량만을 고려한 문제	10
3.1.1 문제 설정	10
3.1.2 수리적 모형	10
3.2 시간 창을 고려한 문제	13
3.2.1 문제 설정	13
3.2.2 수리적 모형	13
3.3 콤바인 차량과 셔틀운송만을 고려한 문제	13
3.3.1 문제 설정	13
3.3.2 수리적 모형	14
3.4 모든 차량의 형태를 고려한 문제	14
3.4.1 문제 설정	14
3.4.2 수리적 모형	14

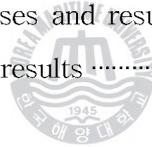


제 4 장 해법 및 실험	17
4.1 콤바인 차량만을 고려한 문제	18
4.1.1 해법	18
4.1.2 실험 및 분석	22
4.2 콤바인 차량과 셔틀운송만을 고려한 문제	25
4.2.1 해법	25
4.2.2 실험 및 분석	29
4.3 모든 차량의 형태를 고려한 문제	32
4.3.1 해법	32
4.3.2 실험 및 분석	37
 제 5 장 결 론	 39
 참고문헌	 40



표 목 차

<표 1-1> Literature summary	2
<표 2-1> 컨테이너 공로 운송 문제의 특성	9
<표 4-1> Experimental cases	22
<표 4-2> Experimental results	23
<표 4-3> Experimental cases with large sizes and results	24
<표 4-4> Data for transportation orders	29
<표 4-5> Travel time matrix	30
<표 4-6> 해법 수행 결과	30
<표 4-7> Random experimental cases and results	31
<표 4-8> Experimental cases and results	38



그 립 목 차

<그림 2-1> 컨테이너 공로 운송의 작업 흐름에 대한 예	8
<그림 4-1> 일반적인 Dial-A-Ride 네트워크	17
<그림 4-2> 콤바인 차량을 이용한 20-foot 컨테이너 운송 네트워크	18
<그림 4-3> 전체 알고리즘 수행 예	19
<그림 4-4> 40-foot 컨테이너 리퀘스트 클러스터	20
<그림 4-5> 20-foot 컨테이너 리퀘스트 클러스터 대상 집합 생성 예	21
<그림 4-6> 20-foot 컨테이너 리퀘스트 클러스터	21
<그림 4-7> 타부서치의 구조	27
<그림 4-8> 해법의 구조	33
<그림 4-9> 초기해 알고리즘	34
<그림 4-10> 해의 표현	34



제 1 장 서 론

1.1 연구의 배경 및 목적

전 세계적으로 글로벌 기업이 점차적으로 늘어남에 따라 국제무역의 수요와 중요성이 날로 증가하고 있다. 근래에 들어서 해상 운송 뿐만 아니라 컨테이너 트럭 운송도 국제 물류에서 중요한 역할을 담당하고 있다. 특히 수출입 의존도가 높은 우리나라에서 효율적인 수출입 컨테이너 운송체계를 구축하는 것은 매우 중요한 국가적 과제이다. 또한 국내 트럭 운송 회사는 국내 및 글로벌 경쟁에서 살아남기 위해 컨테이너 물류의 생산성 향상을 위한 노력이 필수적이다.

컨테이너 물류는 컨테이너 터미널에서의 물류와 컨테이너 운송 물류로 구분할 수 있다. 컨테이너 터미널에서의 물류 즉, 컨테이너 터미널의 설계 및 운영에 관해서는 학문적으로나 실제 사용 측면에서 비교적 많은 연구가 이루어져 왔다. 이에 반해 컨테이너 운송 분야는 높은 국가적 관심과 중요성에도 불구하고 학문적 연구가 지금까지 활발하게 수행되지 못했다. 컨테이너 물류의 생산성은 컨테이너 터미널에서의 물류와 컨테이너 운송의 동기화 정도에 따라 결정되므로 효율적인 컨테이너 운송에 대한 연구가 필요한 실정이다.

최근 국내에서 ODCY (Off-Dock Container Yard) 를 활용한 컨테이너 운송체계가 물류비 절감 및 원활한 교통 흐름 차원에서 점차적으로 On-Dock 서비스 체계로 전환되고 있는 실정이다. 그리고 ICD (Inland Container Depot) 의 본격적인 활용 및 해외 선진 컨테이너 운송업체의 국내 진출이 고려되고 있다. 따라서 컨테이너 운송에 대한 보다 많은 연구와 투자가 요구된다.

본 논문에서는 컨테이너 공로 운송에 대한 특성과 이를 토대로 문제를 체계적으로 접근하기 위한 수리적 모형을 제시한다. 제시되는 모형은 기존의 TSP (Traveling Salesman Problem)와 VRP (Vehicle Routing Problem)에서 활용되는 수리적 모형을 기반으로 개발하였다. 그리고 여러 알고리즘을 이용하여 본 연구에서 제시하고 있는 각각의 모형들에 대한 해법을 제안하고자 한다.

1.2 관련 문헌 연구

본 논문에서는 컨테이너 공로 운송에 대한 차량 경로 문제를 다룬다. 차량 경로 문제에서 특히 pick-up 지점과 delivery 지점 즉 리퀘스트를 가진 차량 경로 문제를 pick-up and delivery 제약을 가진 차량 경로 문제 (VRP_PD) 라고 한다. 그리고 VRP_PD에서 택시 운행과 같이 차량의 용량이 제한적인 문제를 Dial-A-Ride 문제 (DARP) 라고 부른다. 컨테이너 차량 경로 문제는 VRP_PD와 DARP의 문제 설정과 유사하다.

일반적인 VRP_PD와 DARP에 관한 기존 연구들은 국내·외에서 많이 이루어졌다. 일반적으로 차량 경로 문제는 NP-Complete 문제이기 때문에 문제의 규모가 조금만 커져도 최적해를 찾을 수 없다. 따라서 대부분의 논문에서 최적해에 가까운 값을 비교적 빠른 시간 내에 풀 수 있는 휴리스틱 기법을 많이 제시하였다. <표 1-1>은 각각의 관련 논문에서 사용된 방법론을 보여준다.



<표 1-1 > Literature summary

Variants	Methodology	Author and Year
VRP_PD	Additive bounding procedures and insertion algorithm	Vigo (1996)
	Tour-partitioning heuristic	Mosheiov (1998)
	Exact solution and Tabu search	Gendreau et al. (1999)
	Neighborhood search procedure	Beraldi et al. (2005)
	Initial solution based on constructive heuristic, followed by improvement procedures	Nagy and Salhi (2005)
	Tabu search algorithm	Fermin and Roberto (2006)
	Guided simulated annealing search	Urban (2006)
	Cluster-and-search heuristic	Ganesh and Narendran (2007)
	Local search with simple, complex and variable neighborhood and Tabu search	Bianchessi and Righini (2007)
	DARP	Parallel regret insertion heuristic
Local search and diversification strategy		Xiang et al. (2006)
Rejected-reinsertion heuristic		Luo and Schonfeld (2007)
Build an auxiliary graph and then solve an assignment problem on this graph		Roberto and Alberto (2007)

컨테이너 공로 운송에 대한 차량 경로 문제를 다룬 논문은 매우 적었다. 김갑환 외 2(1998)는 화물 수송문제를 대상으로 국내외 연구 현황과 적용 사례를 살펴보았다. 이론적인 해법보다는 적용분야를 중심으로 화물 수송 계획을 설명하였고 철도수송, 트럭 수송, 해운수송, 항공수송 등 수송 수단별로 구분하여 조사하였다. 이 연구를 통해 살펴보면, 컨테이너 터미널의 운영 및 설계에 대해서는 비교적 많은 연구가 수행되었으나 컨테이너 운송 분야에 있어서의 연구는 거의 수행되고 있지 않음을 알 수 있다.

Chung et al.(2007)은 컨테이너 운송의 형태, 컨테이너의 특성, 컨테이너 차량의 종류 그리고 컨테이너 차량의 운영 방식등과 같은 컨테이너 공로 운송의 특성을 설명하고, 이러한 컨테이너 공로 운송의 특성을 고려하여 다양한 모형을 수립하였다. 그리고 문제에 대한 해법으로 간단한 휴리스틱 알고리즘을 제시하여 국내 트럭 운송 회사에서 수집한 실제 문제 상황에 적용하였다. 하지만 이 논문은 각 문제에 대한 모형 수립에 비중을 두어 각각에 대해 효율적인 해법을 제시하지 않았다. 그리고 제시한 해법에 대한 유용성 검증을 하지 않고 현실 문제에 적용할 수 있다는 사실만 확인하였다.

신재영 외 1(2008)은 콤바인 새시 차량을 고려한 컨테이너 차량 경로 문제에 대한 휴리스틱 알고리즘 해법을 제시하였다. 휴리스틱 알고리즘은 리퀘스트 클러스터링을 통하여 원 문제를 m-TSP(multiple-Traveling Salesman Problem)로 변경할 수 있다는 것에 주안점을 두었다. 하지만 휴리스틱 탐색공간이 컨테이너 오더 수에 의해 결정되기 때문에 계산시간이 너무 커져서 실제 문제에 적용하기가 어려웠다. 신재영 외 2(2009)는 콤바인 새시 차량과 셔틀 운송을 고려하여 문제를 설정하였다. 그리고 컨테이너 공로 운송 문제를 체계적으로 접근하기 위한 수리적 모형과 타부서치를 이용한 휴리스틱 해법을 제시하였으며, 수치 실험을 통해 해법의 유용성을 검증하였다.

본 논문에서는 신재영 외 1(2008)과 신재영 외 2(2009)의 논문을 기반으로 국내 컨테이너 운송 현황을 고려하여 효율적인 차량 운송 계획에 대해서 연구한다.

1.3 논문의 구성

본 논문의 구성을 살펴보면, 1장에서는 먼저 연구를 진행하게 된 배경과 연구목적에 관하여 언급하고, 연구목적과 관련된 기존 문헌 연구를 살펴보았다. 2장에서는 컨테이너 공로 운송의 특성을 설명하고, 3장에서는 이러한 특성을 조합하여 여러 가지 문제를 정의하고 분류된 문제 설정에 따라 수리적 모형을 제시한다. 4장에서는 각각의 수리적 모형에 대한 해법을 제시하고 실험 및 분석을 실시하며, 마지막으로 5장에서는 결론을 도출한다.



제 2 장 컨테이너 공로 운송의 특성

본 장에서는 본 연구에서 수행하고자 하는 효율적인 차량 운송 계획을 수립하기 위해 컨테이너 공로 운송의 특성과 업무 흐름에 대해서 살펴본다.

2.1 컨테이너

컨테이너는 그 크기에 따라서 20-feet 컨테이너와 40-feet 컨테이너로 구분된다. 특수한 경우 45-feet 컨테이너도 사용하지만 그다지 많이 사용되지 않아 전 세계적으로 20-feet 컨테이너와 40-feet 컨테이너, 두 종류의 컨테이너로 표준화되어 있다. 일반적으로 컨테이너 물량을 나타내는 단위로서 TEU (20-feet equivalent unit)를 사용하는데, 1 TEU란 20-feet 컨테이너 1개를 뜻한다. 따라서 40-feet 컨테이너 1개의 경우는 2 TEU로 환산한다. 그리고 컨테이너는 컨테이너 내에 들어 있는 화물의 속성에 따라 일반 컨테이너와 냉동 컨테이너 그리고 위험물이 담긴 특수화물 컨테이너로 구분할 수 있다. 특수화물 컨테이너는 특별 부속장비가 부착된 차량에 의해 운송이 이루어진다.

2.2 컨테이너 차량

컨테이너 차량은 머리 부분인 트랙터 (tractor)와 꼬리 부분인 채시 (chassis)로 구분된다. 운전석이 있는 트랙터는 컨테이너 차량을 움직이는 부분이며, 컨테이너가 적재되는 부분인 채시는 트랙터에 연결되어 움직이게 된다. 채시의 경우는 여러 형태가 있으나, 대표적인 것으로는 20-feet 컨테이너 1개만 실을 수 있는 미니채시, 20-feet 컨테이너 2개를 실거나 또는 40-feet 컨테이너 1개를 실을 수 있는 콤파인채시 그리고 40-feet 컨테이너 1개만을 실을 수 있는 구즈넥채시 등으로 구분된다.

국내의 경우 공 컨테이너의 소유주는 컨테이너를 운반하는 회사이며, 운송업체는 선박회사의 지시에 의해 컨테이너를 관리하게 된다. 또한 국내 대부분의 컨테이너 운

송업체들은 자차 (company-owned vehicle), 위탁차 (mandated vehicle), 용차 (rented vehicle)의 형태로 차량을 운영하고 있다. 자차는 운송회사가 직접 소유하고 있는 차량이고, 위탁차는 소유주가 따로 있으면서 계약에 의해 회사가 운영권을 갖고 사용하는 차량으로서 지입차량이라고도 한다. 그리고 용차는 그때그때 필요에 따라 빌려서 사용하는 차량을 말한다. 일반적으로 자차와 위탁차는 운송회사의 배차계획에 따라 운영되나, 용차는 완전히 용차의 소유주에 의해 운영된다. 이와 같이 세 가지의 차량은 소유주와 운영권자가 누구냐에 따라 구분되는 것으로서, 이러한 형태는 컨테이너 운송이 아닌 타 화물운송의 경우에도 유사하게 볼 수 있다.

특히 국내에서만 볼 수 있는 위탁차는 국내의 화물자동차운송사업이 면허제도로 운영되고 있는 것과 관련이 있다. 즉, 국내의 화물자동차운송사업에 관한 면허제도는 운송면허를 받은 사업자만이 화물운송용 차량을 보유할 수 있도록 독점적인 권한을 주고 있다. 이에 대다수 운수업자는 차량 보유에 관한 면허를 받은 뒤 자신이 직접 운영하지 않고 이를 운수업을 하는 다른 개인 또는 회사에게 일정한 대가 - 지입료 또는 위탁관리료 - 를 받는 형식으로 운영하며, 이를 지입제라고 한다. 이러한 지입제는 현재 국내에 보편화되어 있는 실정이다.



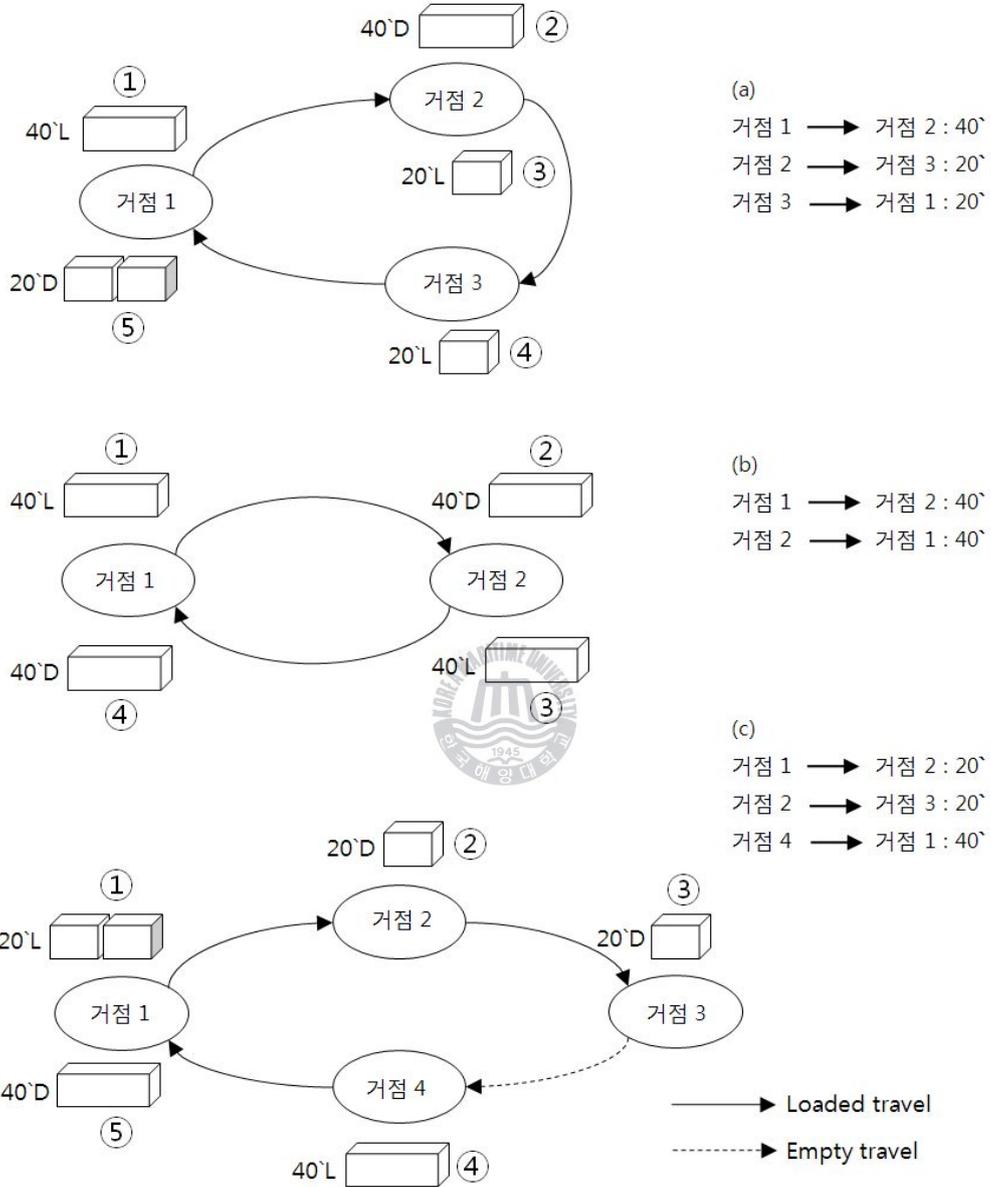
2.3 컨테이너 공로 운송

컨테이너 공로 운송은 컨테이너 운송업체의 관리방식에 따라 약간씩은 차이가 있으나, 일반적으로 서틀운송, 근거리운송, 장거리운송으로 구분할 수 있다. 근거리운송 및 장거리운송은 수출입 컨테이너를 컨테이너 터미널 또는 ODCY - 터미널의 임시 버퍼 기능 및 빈 컨테이너의 적재 기능을 담당 - 와 화주 사이의 운송이다. 즉 화주로부터 화물을 가져오거나 화주에게 화물을 보내는 운송을 말한다. 화주의 위치가 어느 정도 멀리 있는가에 따라 근거리운송 또는 장거리운송으로 구분한다. 일반적으로 화주가 도시 내에 있거나 인근 도시에 있을 경우를 근거리운송이라 하며, 화주가 아주 멀리 떨어져 있는 도시에 있는 경우 장거리 운송이라 한다. 반면에, 컨테이너 터미널과 ODCY 간의 컨테이너 운송, 터미널과 재래부두 간의 환적 (transshipment) 화물을 위

한 운송 그리고 철도수송을 위한 연결운송 등을 셔틀운송이라 한다. 셔틀운송은 전체 컨테이너 처리물량 중 약 20% 정도를 차지하며, 수출입 컨테이너 화물의 경우 각각 화물 마감 시간(cargo closing time)이나 휴일(free day) 내에서 작업이 이루어진다. 그리고 환적화물의 경우는 재래부두와의 연계 때문에 대부분의 물량 처리가 야간에 이루어진다.

근거리운송과 장거리운송의 경우는 화주와 대응되는 상황이므로 화주가 희망하는 시간 창(time window)을 만족해야 하지만 셔틀운송은 그렇지 않다. 실제 대부분의 컨테이너 운송업체에서는 셔틀운송을 위해 필요한 차량을 대략 결정하여 우선적으로 차량을 확보한 후 셔틀만의 배차 계획을 수립하여 운영한다. 따라서 운송회사에서 효율적인 셔틀 서비스 배차 계획을 수립하는 것도 중요한 의사결정 중 하나이다.

<그림 2-1>은 컨테이너 공로 운송의 업무 흐름에 대한 예를 보여준다. (a)는 한 차량이 거점 2와 거점 3에서 각각 20-foot 컨테이너를 상차한 후 거점 1에 두 개의 20-foot 컨테이너를 하차하는 형태의 운송이며, (c)는 반대로 거점 1에서 두 개의 20-foot 컨테이너를 상차한 후 거점 2와 거점 3에서 각각 20-foot 컨테이너를 하차하는 운송이다. (b)는 두 거점 사이에서 40-foot 컨테이너를 상·하차하는 운송이다. 그리고 (a)와 (c)는 콤바인 차량을 이용할 때 가능한 운송 형태이며, (b)는 40-foot 전용이나 콤바인 차량으로 운송이 가능하다.



<그림 2-1> 컨테이너 공로 운송의 작업 흐름에 대한 예

제 3 장 문제 설정 및 수리적 모형

2장에서 살펴본 컨테이너 공로 운송의 특성에 따라 기본적으로 컨테이너 공로 운송 문제에서 고려해야 하는 사항은 컨테이너 크기, 컨테이너 차량의 형태, 시간 창이 유무, 컨테이너 공로 운송의 유형 등이 있다. 그리고 앞서 언급한 것처럼 컨테이너 공로 운송 문제는 잘 알려진 차량 경로 문제 중 VRP_PD와 DARP의 문제 설정과 유사하다. 차량 경로 문제에서 목적함수로 생각해 볼 수 있는 것으로는 차량의 운행시간 최소화이다. 운행시간 최소화는 결국 가용 차량 대수의 최소화로 가능해진다. 따라서 본 논문에서 고려하는 컨테이너 공로 운송 문제의 특성은 <표 3-1>과 같다.

이러한 다양한 특성의 조합을 통해 컨테이너 공로 운송 문제는 여러 가지 문제로 분류할 수 있다. 본 절에서는 컨테이너 공로 운송 문제의 특성과 VRP_PD 및 DARP의 문제 설정을 고려하여 여러 가지 형태의 컨테이너 공로 운송 문제를 정의하고 수리적 모형을 제시한다.



<표 3-1 > 컨테이너 공로 운송 문제의 특성

Characteristics	Option
Container size	20-feet/40-feet
Type of container vehicle	20-feet only/combined/40-feet only
Time window	Without/with
Service coverage area	Shuttle/local/long distance
Objective function	Fleet size required/distance (time)

3.1 콤바인 차량만을 고려한 문제

3.1.1 문제 설정

먼저 본 논문에서 다른 문제의 기본이 되는 문제인 콤바인 차량만을 고려한 컨테이너 차량 경로 문제를 정의한다. 이 문제에서는 가용 차량 대수의 최소화를 목적함수로 설정하였다. 그리고 문제에서 고려한 가정 및 제약조건은 다음과 같다.

- 각 차량은 하나의 Depot 에서 출발하며, 동일한 Depot 로 돌아온다.
- 각 지점은 한 차량에 의해 한 번 방문된다.
- 각 차량은 한 리퀘스트 내의 pick-up 지점을 delivery 지점보다 먼저 방문한다.
- 콤바인 차량만을 고려하며 모든 차량은 동일하다.
- 각 차량의 일일 운행 시간은 모두 같고 제한되어 있다.
- 운송되는 모든 화물은 20-foot 나 40-foot 컨테이너이다.
- TEU 를 화물 단위로 하면, 모든 화물의 크기는 '1' 이나 '2' 로 표현될 수 있으며 콤바인 차량을 대상으로 하기 때문에 각 차량의 최대 적재 용량은 '2' 라고 가정한다.
- 다른 두 지점간의 운행 시간은 주어져 있다고 가정한다.

3.1.2 수리적 모형

모델링을 위해서 먼저 운송해야 할 n 개의 컨테이너가 있다고 가정한다. 그리고 각각의 컨테이너에 대한 pick-up 지점과 delivery 지점을 정의한다. 즉, 컨테이너 i 의 pick-up 지점을 노드 i 라 하면 컨테이너 i 의 delivery 지점은 노드 $n+i$ 라고 설정한다. 따라서 운송 대상 컨테이너가 n 개 이면 노드의 수는 $2n$ 개다. 모형에서 사용되는 *Parameter* 와 결정 변수는 다음과 같다.

Parameter

$N_1 = \{1, 2, \dots, n\}$: pick-up 노드 집합

$N_2 = \{n+1, n+2, \dots, 2n\}$: delivery 노드 집합

$N = N_1 + N_2 + \{0\}$

K : 콤바인 차량의 집합

s_j : 노드 j 에서 20-foot 컨테이너가 상차되면 1, 40-foot 컨테이너가 상차되면 2,
20-foot 컨테이너가 하차되면 -1, 40-foot 컨테이너가 하차되면 -2

T : 차량의 하루 최대 운행 시간

t_{ij} : 노드 i 에서 서비스(상·하차) 시간과 노드 i 에서 노드 j 까지 운행시간의 합

결정 변수



x_{ij}^k : 차량 k 가 노드 i 에서 노드 j 로 운행하면 1, 그렇지 않으면 0

t_i^k : 노드 i 에서 차량 k 의 도착 시간

w_i^k : 차량 k 가 노드 i 에서 출발할 때 차량 k 의 적재된 화물 크기

문제를 모델링하면 다음과 같다.

(P1)

$$\text{Minimize } \sum_{j \in N} \sum_{k \in K} x_{0j}^k \quad (1)$$

subject to

$$\sum_{j \in N} \sum_{k \in K} x_{ji}^k = 1 \quad j \in N, i \neq 0 \quad (2)$$

$$\sum_{j \in N} x_{ij}^k = \sum_{j \in N} x_{ji}^k \quad i \in N, k \in K \quad (3)$$

$$\sum_{j \in N} x_{0j}^k \leq 1 \quad k \in K \quad (4)$$

$$\sum_{j \in N} x_{ij}^k = \sum_{j \in N} x_{j,n+i}^k \quad i \in N_1, k \in K \quad (5)$$

$$\sum_{j \in N} x_{ij}^k = 1 \rightarrow t_i^k + t_{i,n+i} \leq t_{n+i}^k \quad i \in N_1, k \in K \quad (6)$$

$$x_{ij}^k = 1 \rightarrow t_i^k + t_{ij} \leq t_j^k \quad i \in N, j \in N, j \neq 0, k \in K \quad (7)$$

$$t_0^k = 0 \quad k \in K \quad (8)$$

$$t_i^k + t_{i0} \leq T \quad i \in N_2, k \in K \quad (9)$$

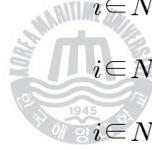
$$x_{ij}^k = 1 \rightarrow w_i^k + s_j \leq w_j^k \quad i \in N, j \in N, k \in K \quad (10)$$

$$w_0^k = 0 \quad k \in K \quad (11)$$

$$w_i^k \leq 2 \quad i \in N, k \in K \quad (12)$$

$$x_{ij}^k = 0 \text{ or } 1 \quad i \in N, j \in N, k \in K \quad (13)$$

$$t_i^k \geq 0, w_i^k \geq 0 \quad i \in N, k \in K \quad (14)$$



목적함수인 (1)은 컨테이너 운송에 사용된 전체 차량 대수의 최소화를 나타낸다. 제약조건 (2)는 반드시 하나의 차량이 노드 i 를 방문해야 한다는 의미이고, (3)은 네트워크 흐름보존법칙(flow conservation rule) 그리고 (4)는 차량의 가용성을 의미한다. (5)와 (6)은 pick-up 노드와 delivery 노드의 선후 관계를 나타내며, (7)은 차량 k 가 노드 i 에서 노드 j 로 운행할 경우 두 노드사이의 시간 제약조건을 의미한다. (8)과 (9)는 차량 k 의 운행 시간에 대한 제한을 나타낸다. (10)은 차량 k 가 노드 i 에서 노드 j 로 이동할 경우 각 노드에서 차량 k 에 적재된 화물의 크기에 대한 관계를 규정하는 것이다. (11)은 차량 k 가 시작 지점에서 빈 차량으로 출발한다는 것을 의미하며, (12)는 차량 k 의 최대 적재 용량 제한이다.

3.2 시간 창을 고려한 문제

3.2.1 문제 설정

가정 및 제약조건은 콤바인 차량만을 고려한 문제와 동일하며, 추가적으로 시간 창을 고려한 컨테이너 차량 경로 문제이다. 각각의 방문 지점은 특정한 시간 내에 차량의 방문을 요구하는 시간 창을 가진다고 가정한다.

3.2.2 수리적 모형

Parameter 와 결정 변수는 콤바인 차량만을 고려한 문제와 동일하다. 그리고 노드 i 에서 시간 창을 $[a_i, b_i]$ 라고 하자. 따라서 시간 창을 제약을 가진 문제의 모형은 (P1) 에 다음과 같은 제약 식을 추가하면 된다.

$$a_i \leq t_i^k \leq b_i \quad (1)$$

3.3 콤바인 차량과 셔틀운송만을 고려한 문제

3.3.1 문제 설정

실제 대부분의 컨테이너 운송업체에서는 셔틀운송을 위해 필요한 차량을 대략 결정하여 우선적으로 차량을 확보한 후 셔틀만의 배차 계획을 수립하여 운영한다. 따라서 셔틀운송만을 고려한 컨테이너 공로 운송 문제에 대한 고찰도 필요하다. 이 문제에서 고려되는 셔틀운송의 특성은 거점들 간의 운송이 빈번하게 발생하며, O/D 쌍이 같은 물량이 여러 개 발생 한다는 점이다.

셔틀운송만을 고려한다는 것을 제외하고는 문제 설정은 콤바인 차량만을 고려한 문제와 동일하다. 셔틀운송의 특성상 각 차량이 같은 지점을 여러 번 방문할 수도 있는

나, 같은 O/D 쌍을 가진 리퀘스트를 서로 다른 리퀘스트로 정의하면 각 차량은 각 지점을 한 번 방문한다고 볼 수 있다.

3.3.2 수리적 모형

Parameter 및 결정 변수 그리고 수리적 모형은 (P1) 과 동일하다.

3.4 모든 차량의 형태를 고려한 문제

3.4.1 문제 설정

20-foot 전용, 콤바인, 40-foot 전용 차량을 모두 고려한다는 것을 제외하고는 문제 설정은 콤바인 차량만을 고려한 문제와 동일하다. 모든 차량의 비용은 동일하며, 각 유형별 차량의 대수는 한정되어 있다고 가정하였다. 그리고 이 문제의 목적 함수는 가용된 차량의 총 운행시간(거리)의 최소화로 설정하였다.

3.4.2 수리적 모형

모델링을 위해서 20-foot 전용 차량, 콤바인 차량, 40-foot 전용 차량은 각각 k_1 , k_2 , k_3 개 있다고 가정한다. 그리고 차량의 인덱스는 각각 $K_1 = \{1, 2, \dots, k_1\}$, $K_2 = \{k_1 + 1, k_1 + 2, \dots, k_1 + k_2\}$, $K_3 = \{k_1 + k_2 + 1, k_1 + k_2 + 2, \dots, k_1 + k_2 + k_3\}$ 라고 정의한다. 모형에서 사용되는 *Parameter* 와 결정 변수는 다음과 같다.

Parameter

$N_1 = \{1, 2, \dots, n\}$: pick-up 노드 집합

$N_2 = \{n + 1, n + 2, \dots, 2n\}$: delivery 노드 집합

$$N = N_1 + N_2 + \{0\}$$

$K_1 = \{1, 2, \dots, k_1\}$: 20-feet 전용 차량 집합

$K_2 = \{k_1 + 1, k_1 + 2, \dots, k_1 + k_2\}$: 콤바인 차량 집합

$K_3 = \{k_1 + k_2 + 1, k_1 + k_2 + 2, \dots, k_1 + k_2 + k_3\}$: 40-feet 전용 차량 집합

$$K = K_1 + K_2 + K_3$$

V_k : 컨테이너 차량의 최대 적재 용량

s_j : 노드 j 에서 20 feet 컨테이너가 상차되면 1, 40 feet 컨테이너가 상차되면 2,
20 feet 컨테이너가 하차되면 -1, 40 feet 컨테이너가 하차되면 -2

T : 차량의 하루 최대 운행 시간

t_{ij} : 노드 i 에서 서비스(상·하차) 시간과 노드 i 에서 노드 j 까지 운행시간의 합

결정 변수

x_{ij}^k : 차량 k 가 노드 i 에서 노드 j 로 운행하면 1, 그렇지 않으면 0

y_i^k : 차량 k 가 노드 i 에서 서비스 하면 1, 그렇지 않으면 0

t_i^k : 노드 i 에서 차량 k 의 도착 시간

w_i^k : 차량 k 가 노드 i 에서 출발할 때 차량 k 의 적재된 화물 크기

문제를 모델링하면 다음과 같다.

(P2)

$$\text{Minimize } \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} t_{ij} x_{ij}^k \quad (1)$$

subject to

$$y_i^k = 1 \rightarrow |s_i|y_i^k = V_k \quad i \in N, k \in K_1 + K_3 \quad (2)$$

$$\sum_{k \in K} y_i^k = 1 \quad i \in N_1 + N_2 \quad (3)$$

$$y_i^k - y_{n+i}^k = 0 \quad i \in N_1, k \in K \quad (4)$$

$$y_i^k = 0 \text{ or } 1 \quad i \in N, k \in K \quad (5)$$

$$\sum_{i \in N} x_{ij}^k = y_j^k \quad j \in N, k \in K \quad (6)$$

$$\sum_{j \in N} x_{ij}^k = y_i^k \quad i \in N, k \in K \quad (7)$$

$$\sum_{j \in N} x_{ij}^k = 1 \rightarrow t_i^k + t_{i,n+i} \leq t_{n+i}^k \quad i \in N_1, k \in K \quad (8)$$

$$x_{ij}^k = 1 \rightarrow t_i^k + t_{ij} \leq t_j^k \quad i \in N, j \in N, j \neq 0, k \in K \quad (9)$$

$$t_0^k = 0 \quad k \in K \quad (10)$$

$$t_i^k + t_{i0} \leq T \quad i \in N_2, k \in K \quad (11)$$

$$x_{ij}^k = 1 \rightarrow w_i^k + s_j \leq w_j^k \quad i \in N, j \in N, k \in K \quad (12)$$

$$w_0^k = 0 \quad k \in K \quad (13)$$

$$w_i^k \begin{cases} \leq 1 \\ \leq 2 \\ = 0 \text{ or } 2 \end{cases} \quad \begin{matrix} i \in N, k \in K_1 \\ i \in N, k \in K_2 \\ i \in N, k \in K_3 \end{matrix} \quad (14)$$

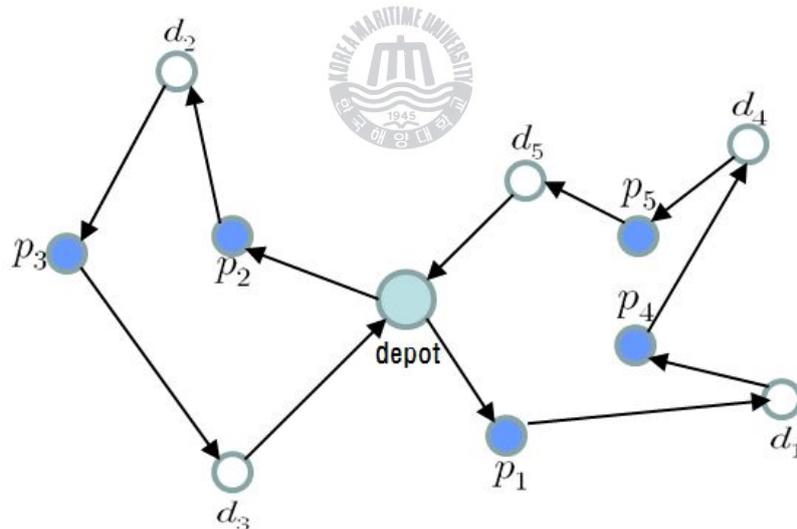
$$t_i^k \geq 0, w_i^k \geq 0 \quad i \in N, k \in K \quad (15)$$

목적함수인 (1)은 가용된 차량들에 대한 총 운행시간의 최소화를 나타낸다. 제약조건 (2) - (5)는 일반적인 할당 문제에 관한 식으로 각각의 리퀘스트는 한 차량에 할당되고, 각 차량은 차량의 최대 적재 용량을 초과할 수 없다는 것을 의미한다. 예를 들어 20-foot 컨테이너는 40-foot 전용 차량으로 운행할 수 없으며, 반대로 마찬가지다. (6) - (15)는 각 차량에 할당된 리퀘스트에 대한 방문 순서를 결정하는 제약 식으로 (P1)의 제약 식 내용과 동일하다.

제 4 장 해법 및 실험

일반적으로 차량 경로 문제는 NP-Complete 문제이기 때문에 문제의 규모가 조금만 커져도 최적해를 찾을 수 없다. 따라서 본 논문에서 제안한 모형에 대한 해법으로 휴리스틱 기법을 이용하였다. 그리고 시간 창을 고려한 문제는 다른 문제의 해법을 적용하면 해를 구할 수 있기 때문에 해법 및 실험에 대한 설명은 생략한다.

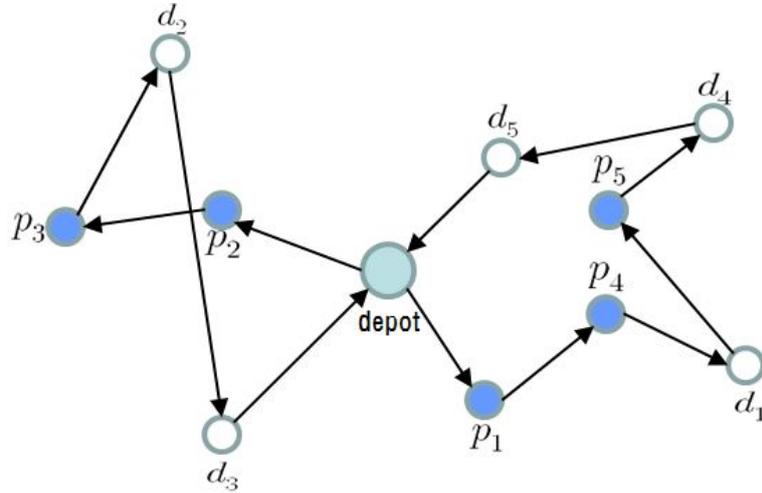
20-foot 전용 차량이나 40-foot 전용 차량을 이용한 차량 경로 문제는 일반적인 DARP의 네트워크 구조와 동일하다. 또한 콤바인 차량을 이용한 차량 경로 문제에서 40-foot 컨테이너의 운송도 마찬가지로이다. <그림 4-1>은 일반적인 DARP의 네트워크를 보여준다. 하지만 콤바인 차량을 이용한 차량 경로 문제에서 화물이 20-foot 컨테이너일 때 <그림 4-2>와 같이 다른 네트워크를 가질 수도 있다.



차량 1 : $depot \rightarrow p_1 \rightarrow p_4 \rightarrow d_1 \rightarrow p_5 \rightarrow d_4 \rightarrow d_5 \rightarrow depot$

차량 2 : $depot \rightarrow p_2 \rightarrow p_3 \rightarrow d_2 \rightarrow d_3 \rightarrow depot$

<그림 4-1> 일반적인 Dial-A-Ride 네트워크



차량 1 : $depot \rightarrow p_1 \rightarrow d_1 \rightarrow p_4 \rightarrow d_4 \rightarrow p_5 \rightarrow d_5 \rightarrow depot$

차량 2 : $depot \rightarrow p_2 \rightarrow d_2 \rightarrow p_3 \rightarrow d_3 \rightarrow depot$

<그림 4-2> 콤바인 차량을 이용한 20-foot 컨테이너 운송 네트워크

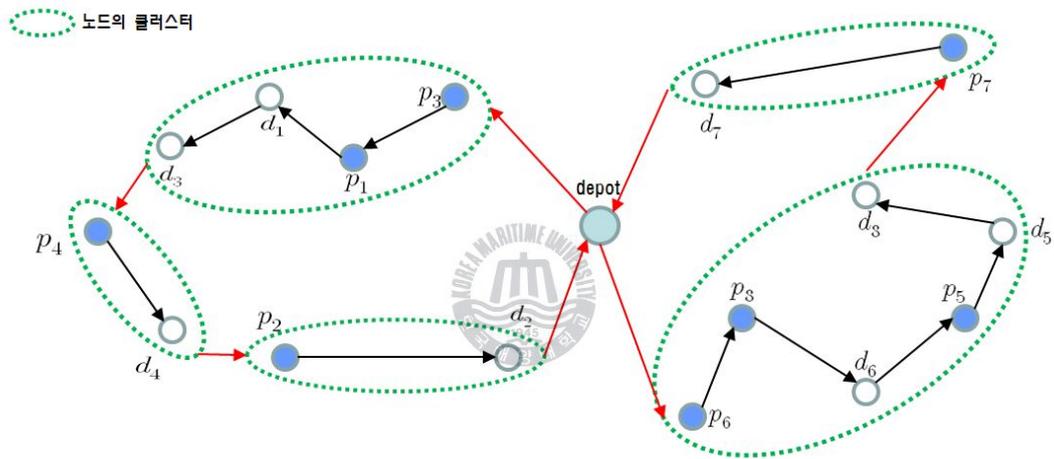
콤바인 차량의 적재 용량은 '2' 이기 때문에 40-foot 컨테이너의 운송에서 차량은 pick-up 노드를 방문한 후에는 반드시 delivery 노드를 방문해야 한다. 반면에 20-foot 컨테이너 운송에서는 <그림 4-2> 와 같이 리퀘스트 2개 내지 3개의 방문 순서가 꼬이는 경우도 발생할 수 있다. 본 장에서는 이러한 차량의 특성을 고려하여 3장에서 정의한 문제에 대한 해법으로 적합하고 효율적인 알고리즘을 제시한다.

4.1 콤바인 차량만을 고려한 문제

4.1.1 해법

본 문제에 대해 제시하는 해법은 크게 리퀘스트 클러스터 단계와 m-TSP 해법 단계로 나누어진다.

리퀘스트 클러스터 단계는 연속 방문 가능한 리퀘스트들을 클러스터하여 하나의 노드로 만드는 단계이다. 이렇게 콤바인 차량의 특성을 이용하여 리퀘스트를 클러스터하게 되면 기존의 제약조건에서 차량의 capacity 제약과 pick-up and delivery 제약이 없어지기 때문에 m-TSP 구조로 문제를 변경할 수 있다. 기존의 많은 논문에서 다양하고 효율적인 m-TSP에 대한 해법이 제시되었다. 본 논문에서는 *Insertion 알고리즘*을 이용하여 문제를 풀었으며, m-TSP 해법 단계에 대한 추가 설명은 하지 않는다. <그림 4-3>은 전체 알고리즘의 수행 예이다.



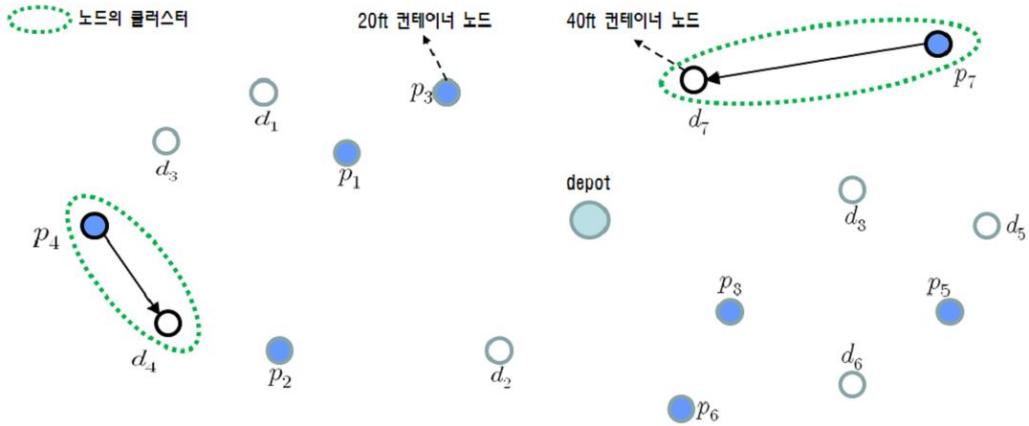
차량 1 : $depot \rightarrow p_3 \rightarrow p_1 \rightarrow d_1 \rightarrow d_3 \rightarrow p_4 \rightarrow d_4 \rightarrow p_2 \rightarrow d_2$

차량 2 : $depot \rightarrow p_6 \rightarrow p_8 \rightarrow d_6 \rightarrow p_5 \rightarrow d_5 \rightarrow d_8 \rightarrow p_7 \rightarrow d_7$

<그림 4-3> 전체 알고리즘 수행 예

<리퀘스트 클러스터 단계>

먼저 <그림 4-4>와 같이 모든 40-feet 컨테이너 리퀘스트 각각을 하나의 노드로 클러스터 한다.



<그림 4-4> 40-foot 컨테이너 리퀘스트 클러스터

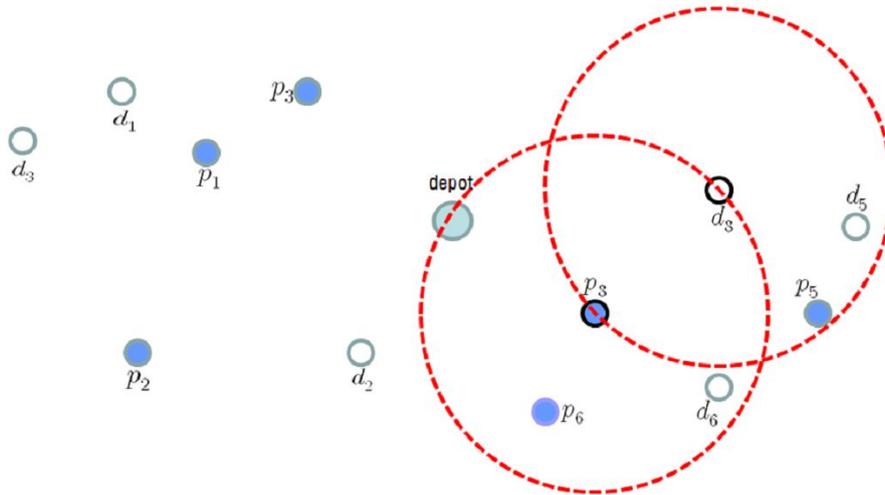
모든 20-foot 컨테이너 리퀘스트는 다음과 같은 단계로 클러스터 한다.

- Step 1 각각의 리퀘스트의 클러스터 대상 집합 (Q)을 생성
- Step 2 거리 단축 값이 최대인 것부터 Q 를 정렬
- Step 3 Q 에 동일한 리퀘스트가 포함되어 있으면 차량의 일일 최대 운행 시간 (T)을 초과하지 않는 범위 내에서 순서대로 클러스터. 나머지는 Q 에서 제거
- Step 4 Q 가 공집합이 되면 클러스터 되지 않은 리퀘스트 각각을 40-foot 컨테이너 리퀘스트의 클러스터처럼 하나의 노드로 클러스터

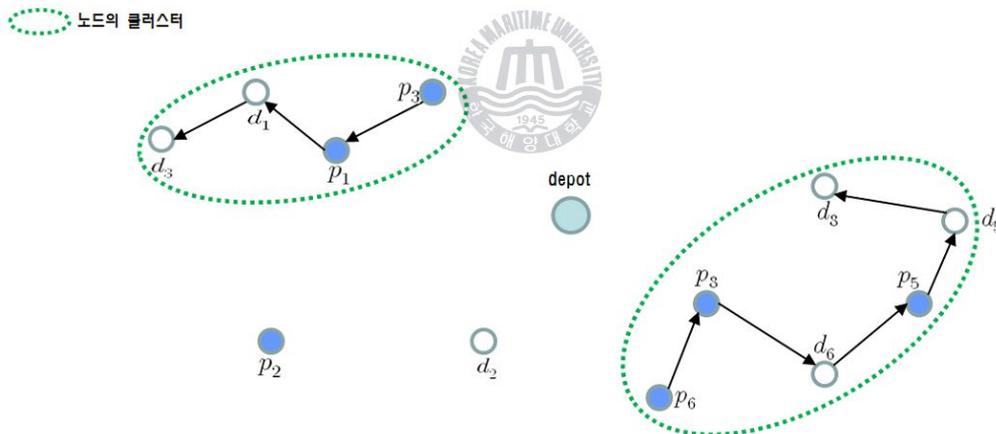
위에서 제시한 단계대로 <그림 4-4> 문제를 실행해 보면 다음과 같다.

먼저 <그림 4-5>와 같이 각각의 20-foot 컨테이너 리퀘스트에 대한 Q 를 생성한다. 클러스터 대상은 리퀘스트 내의 거리와 다른 리퀘스트와의 거리를 비교하여 후자가 전자보다 작으면 대상 집합으로 선택된다.

Step 1을 수행하면 클러스터 대상 집합 $Q = \{r_3r_1, r_8r_5, r_8r_6\}$ 를 생성할 수 있다. 그리고 Step 2를 수행하면 클러스터 대상 집합 $Q = \{r_8r_6, r_3r_1, r_8r_5\}$ 로 정렬되고 <그림 4-6>과 같이 리퀘스트가 클러스터 된다.



<그림 4-5> 20-feet 컨테이너 리퀘스트 클러스터 대상 집합 생성 예



<그림 4-6> 20-feet 컨테이너 리퀘스트 클러스터

마지막으로 Step 4를 수행하게 되면 리퀘스트 클러스터 단계가 종료된다. 리퀘스트 클러스터 단계를 통하여 새로 생성된 노드는 다음과 같다.

- 노드 1 : $p_4 \rightarrow d_4$ 노드 2 : $p_7 \rightarrow d_7$, 노드 3 : $p_3 \rightarrow p_1 \rightarrow d_1 \rightarrow d_3$,
 노드 4 : $p_6 \rightarrow p_8 \rightarrow d_6 \rightarrow p_5 \rightarrow d_5 \rightarrow d_8$ 노드 5 : $p_2 \rightarrow d_2$

4.1.2 실험 및 분석

위에서 제시한 해법을 다양한 문제사이즈와 상황에 적용시켜 분석을 실시하였다. 크기가 작은 문제는 <표 4-1> 과 같이 일일 발생하는 리퀘스트를 5개로 설정하였고, 20-foot 리퀘스트의 수를 다양하게 고려하였다. 그리고 차량의 일일 최대 운행 시간을 500/900분으로 설정하였으며, 각 리퀘스트 내의 운행시간 정도를 전체 리퀘스트의 20%가 200 보다 큰 경우(S)와 80%가 200 보다 큰 경우(L)로 나누어 설정하였다. 마지막으로 각 노드간의 운행시간은 50 에서 200 사이의 난수로 결정하였다.

<표 4-1> Experimental cases

Case	리퀘스트 수/ 20-foot 리퀘스트 수	T (min)	운행시간 정도	Case	리퀘스트 수/ 20-foot 리퀘스트 수	T (min)	운행시간 정도
1-1	5/2	500	S	5-1	5/4	500	S
1-2	5/2	500	L	5-2	5/4	500	L
1-3	5/2	900	S	5-3	5/4	900	S
1-4	5/2	900	L	5-4	5/4	900	L
2-1	5/2	500	S	6-1	5/4	500	S
2-2	5/2	500	L	6-2	5/4	500	L
2-3	5/2	900	S	6-3	5/4	900	S
2-4	5/2	900	L	6-4	5/4	900	L
3-1	5/3	500	S	7-1	5/5	500	S
3-2	5/3	500	L	7-2	5/5	500	L
3-3	5/3	900	S	7-3	5/5	900	S
3-4	5/3	900	L	7-4	5/5	900	L
4-1	5/3	500	S	8-1	5/5	500	S
4-2	5/3	500	L	8-2	5/5	500	L
4-3	5/3	900	S	8-3	5/5	900	S
4-4	5/3	900	L	8-4	5/5	900	L

전체 32가지의 경우에 대해 해법을 적용하여 실험을 실시하였으며, 각 경우별 가용 차량 대수, 해법의 수행 시간을 측정하였다. 최적해는 (P1)을 근간으로 Cplex를 이용하여 해를 구하였다.

실험 결과 <표 4-2>와 같이 모든 경우에서 해법의 해가 최적해와 일치하였으며, 20-foot 리퀘스트 수가 많고 차량 일일 최대 운행 시간(T)이 클수록 가용 차량 대수가 작아지는 것을 알 수 있다. 이는 20-foot 리퀘스트가 많을수록 클러스터 되는 리퀘스트가 많아지고, T 가 커지면 방문지가 많아질 수 있다는 사실을 고려하면 당연한 결과이다. 그리고 모든 경우에서 Cplex의 수행 시간이 평균 하루인 데 비해 제안한 휴리스틱 알고리즘의 실행 시간은 1s 미만으로 즉시 결과를 확인할 수 있는 수준의 짧은 계산 시간을 보였다.

<표 4-2> Experimental results

Case	휴리스틱 알고리즘	Cplex 최적해	Case	휴리스틱 알고리즘	Cplex 최적해
1-1	3	3	5-1	2	2
1-2	3	3	5-2	3	3
1-3	2	2	5-3	2	2
1-4	2	2	5-4	2	2
2-1	4	4	6-1	4	4
2-2	5	5	6-2	5	5
2-3	2	2	6-3	2	2
2-4	3	3	6-4	2	2
3-1	3	3	7-1	2	2
3-2	3	3	7-2	3	3
3-3	2	2	7-3	2	2
3-4	2	2	7-4	2	2
4-1	4	4	8-1	4	4
4-2	5	5	8-2	5	5
4-3	2	2	8-3	2	2
4-4	2	2	8-4	2	2

Cplex 를 이용하여 해를 구하기 어려운 크기가 큰 문제를 <표 4-3> 과 같이 랜덤하게 생성하여 해법을 적용해 보았다. 차량의 운행 시간은 480 분으로 가정하였고, 나머지 조건은 랜덤하게 주었다. 실험 결과 유의한 시간 내에 해를 얻을 수 있었다.

<표 4-3> Experimental cases with large sizes and results

Case	리퀘스트 수	운행차량 대수	계산시간 (sec)	Case	리퀘스트 수	운행차량 대수	계산시간 (sec)
1-1	500	36	66.90	2-5	800	211	1604.94
1-2	500	80	65.46	2-6	800	55	4281.36
1-3	500	36	281.82	2-7	800	126	4535.64
1-4	500	81	286.80	2-8	800	213	4001.16
1-5	500	133	300.72	3-1	1000	70	1220.34
1-6	500	36	694.32	3-2	1000	156	1248.12
1-7	500	81	642.24	3-3	1000	70	4674.48
1-8	500	137	609.90	3-4	1000	157	4670.28
2-1	800	56	461.04	3-5	1000	263	4401.78
2-2	800	126	477.60	3-6	1000	69	9997.56
2-3	800	56	1900.44	3-7	1000	158	10026.06
2-4	800	126	1856.40	3-8	1000	265	10152.48

4.2 콤바인 차량과 셔틀운송만을 고려한 문제

4.2.1 해법

기본적인 해법의 구조는 콤바인 차량만을 고려한 문제에 대한 해법 (*HAI*) 과 동일하다. 즉, 클러스터 단계와 m-TSP 해법 단계로 나누어진다. 클러스터 단계에서의 해법인 *HAI* 은 클러스터 대상 노드가 전체 노드이기 때문에 서치 영역이 커져서 계산 시간이 많이 걸렸다. 반면에 본 문제에서는 셔틀운송을 대상으로 하기 때문에 같은 pick-up and delivery 노드를 가진 O/D 쌍이 많이 발생한다. 따라서 *HAI* 과 달리 O/D 쌍을 클러스터 대상 노드로 지정하여 노드를 클러스터 한 후 클러스터 된 O/D 쌍에 따라 물량을 할당하는 과정을 모든 물량이 할당될 때까지 반복하는 방법으로 해법을 구성할 수 있다. 일반적으로 국내 셔틀운송에서 거점(O/D) 수는 10개미만으로 한정적이기 때문에 *HAI* 보다 계산시간을 많이 줄일 수 있다.

이와 같은 셔틀운송만의 특징을 고려하여 노드 클러스터 단계에 대한 해법으로 *HAI* 을 변형한 휴리스틱 알고리즘 (*HA2*) 을 이용하였다. 그리고 m-TSP는 4.1절과 마찬가지로 *Insertion* 알고리즘을 이용하여 문제를 풀었다. 또한, 메타 휴리스틱인 타부서치 알고리즘 (*TA*) 을 이용하여 새로운 해법을 개발하였다. 본 절에서는 먼저 노드 클러스터 단계 해법인 *HA2* 를 설명하고 타부서치의 개념과 전체 해법 과정인 *TA* 의 수행 절차에 대해 설명하고자 한다.

<휴리스틱 알고리즘>

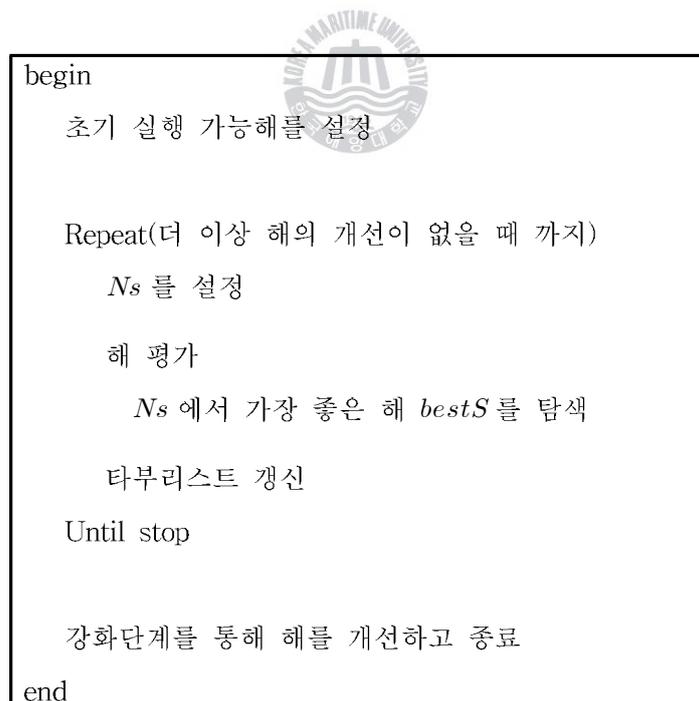
콤바인 차량만을 고려한 문제와 마찬가지로 *HA2* 를 이용하여 모든 노드를 클러스터 하여 차량의 capacity 제약과 pick-up and delivery 제약을 완화하여 m-TSP로 문제를 변형한다. 노드 클러스터 단계 해법인 *HA2* 는 다음과 같은 단계로 해법을 수행한다.

- Step 1 모든 40-foot 컨테이너 리퀘스트 각각을 하나의 노드로 클러스터
- Step 2 20-foot 컨테이너 리퀘스트 클러스터
- Step 2.1 20-foot 컨테이너 리퀘스트를 가지는 거점 수 (d) 에 의해 노드를 구성

$$N_1 = \{1, 2, \dots, d\}, N_2 = \{d+1, d+2, \dots, 2d\}, N = N_1 + N_2 + \{0\}$$
- Step 2.2 각각의 노드간의 리퀘스트에 대한 클러스터 대상 집합 (Q) 생성
- Step 2.3 거리 단축 값이 최대인 것부터 Q 를 정렬
- Step 2.4 Q 에 동일한 리퀘스트가 포함되어 있으면 차량의 일일 최대 운행 시간 (T) 을 초과하지 않는 범위 내에서 순서대로 클러스터. 나머지는 Q 에서 제거
- Step 2.5 Q 가 공집합이 되면 클러스터 되지 않은 리퀘스트 각각을 하나의 노드로 클러스터
- Step 2.6 결정된 클러스터 노드에 일치하게 컨테이너 물량을 최대한 할당
- Step 3 $d=0$ 이 될 때까지 Step 2 를 반복

<타부서치의 개념>

타부서치 (Tabu Search)는 인간의 기억과정을 이용한 것으로, 인공지능 분야에서 선택된 개념에 기초를 둔 기법이다. 최초 Glover 에 의해서 현재의 형태로 제시되었고, Hansen 에 의해서 처음으로 기본 아이디어가 제시 되었다. 타부 서치는 3개의 기본적인 특징이 있다. 첫 번째는 유연한 성질이 있는 컴퓨터 기억구조를 사용한다는 것이고, 두 번째는 타부리스트와 열망기준을 두고 제한적인 탐색을 하게 된다는 것이다. 마지막으로 다양화와 강화 전략을 통해서 탐색하지 않은 새로운 영역을 찾아가거나, 최적해가 있을 것으로 예상되는 지역을 집중적으로 탐색하는 등의 특징이 있다. 타부 서치는 이러한 특징을 유연하게 활용하여 빠른 시간 내에 최적해에 근접하는 해를 도출한다. <그림 4-7>은 TA 해법에서 사용되는 타부서치의 구조를 보여준다. N_s 는 이웃해 탐색 공간을 나타낸다.



<그림 4-7> 타부서치의 구조

<타부서치 알고리즘>

- Step 1 모든 40-foot 컨테이너 리퀘스트 각각을 하나의 노드로 클러스터
- Step 2 20-foot 컨테이너 리퀘스트를 가지는 거점 수 (d) 에 의해 노드를 구성
 $N_1 = \{1, 2, \dots, d\}$, $N_2 = \{d+1, d+2, \dots, 2d\}$, $N = N_1 + N_2 + \{0\}$
- Step 3 타부서치 단계
- Step 3.1 (초기해 단계)
- 각각의 노드 쌍을 클러스터 하여 초기해를 생성.
 Step 2.4 를 통해 현재 최적해 ($bestS$) 결정.
 타부목록 (tL) = Φ , 타부목록 크기는 3으로 설정.
- Step 3.2 (이웃해 탐색 단계)
- 각각의 노드 쌍 (r_i)을 현재 클러스터 집합에서 제거하고 다른 클러스터 집합에 삽입하는 방법으로 이웃해 (Ns) 탐색.
 $r_i \in tL$ 이면 이웃해 탐색 금지.
- Step 3.3 (해 평가 단계)
- Ns 를 *Insertion* 알고리즘을 이용하여 평가값 ($tempS$) 을 정의.
 $bestS < tempS$ 면 $bestS = tempS$, r_i 를 tL 에 등록하고 Step 3.2
 으로 이동. 그렇지 않으면 Step 3.4 로 이동.
- Step 3.4 (강화 단계 및 종료)
- $bestS$ 의 내부 경로 순서를 교환하여 개선하고 종료
- Step 4 결정된 클러스터 노드에 일치하게 컨테이너 물량을 최대한 할당
- Step 5 $d=0$ 이 될 때까지 Step 2, 3, 4 를 반복

4.2.2 실험 및 분석

Chung et al.(2007)에서 이용한 실제 데이터에 대해 수치 실험을 하였다. *HAI* 과 본 절에서 제시한 *HA2* 및 *TA* 를 해당 문제의 해법으로 수행하여 비교 분석하였다.

<표 4-4> 와 <표 4-5> 는 한 컨테이너 운송 회사로부터 얻은 자료이며, 각 거점별 운송 물량과 운행 시간을 나타낸다. 모든 물량은 하루에 처리되어야 하며, 운송되는 총 컨테이너 수는 1093 개, 거점은 8 개이다. 그리고 모든 차량은 콤팩트 새시 차량이며, 가용차량은 93 대, 각 차량의 최대 운행 시간은 8시간, 컨테이너의 상·하차 시간은 5분으로 가정하였다.

<표 4-4> Data for transportation orders

From To	업궁	수영	용당	신선대	자성대	감천	철도	U.T.C
업궁		30(40') 8(20')			1 3			48 15
수영	8 24			36 13	47 3			
용당					25 5	1 26		13 13
신선대		29 6			17 28		11 10	
자성대	26 15	8 20	48 3	28 6		23 16	46 17	3 18
감천			41 5		9 9			45 28
철도				7 15	38 6			
U.T.C	45 29		4 29		10 22	32 22		

<표 4-5> Travel time matrix

(unit: min)

From To	엄궁	수영	용당	신선대	자성대	감천	철도	U.T.C
엄궁	0	50	30	40	30	30	30	35
수영		0	30	40	30	35	35	35
용당			0	10	30	35	35	25
신선대				0	20	25	25	20
자성대					0	5	5	10
감천						0	5	15
철도							0	15
U.T.C								0

<표 4-6> 은 세 가지 다른 해법에 대한 수행 결과 및 계산시간을 보여준다. 서틀운송의 특성을 이용하여 탐색 영역을 줄인 결과 *HA1*의 계산시간 (5004.238s) 보다 *HA2*의 계산시간 (5.625s) 이 매우 빠른 것을 볼 수 있다. 그리고 *HA2*와 *TA*를 비교하면 계산시간은 더 늘어나지만 어느 정도 유의한 수준이고, 반면에 운행 차량 대수는 14대가 줄어들기 때문에 *TA*가 효율적이라고 말할 수 있다.

<표 4-6> 해법 수행 결과

해 법	차량 당 평균 운행시간 (hour)	운행 차량 대수	계산 시간 (sec)
<i>HA1</i>	7.464	67	5004.238
<i>HA2</i>	7.464	67	5.625
<i>TA</i>	7.616	53	308.762

<표 4-7> 과 같이 랜덤 생성한 문제에 대해 본 장에서 제시한 해법을 수행하였다. Case 1은 Cheng et al.(2007)의 문제와 동일한 상태에서 컨테이너 리퀘스트 수를 변

화시킨 문제이며, 나머지는 랜덤 생성한 문제이다. 랜덤 문제를 실험해 본 결과도 계산시간은 TA가 HA2보다 많이 걸리지만 해는 매우 개선되는 것을 볼 수 있다.

<표 4-7> Random experimental cases and results

Case	거점 수/ 컨 수	해법	운행차량 대수	계산시간 (sec)	Case	거점 수/ 컨 수	해법	운행차량 대수	계산시간 (sec)
1-1	-/1533	HA2	89	10.83	3-3	7/2038	HA2	109	17.41
		TA	70	6566.20			TA	81	2214.63
1-2	-/1712	HA2	99	12.92	3-4	7/1880	HA2	85	14.45
		TA	66	8636.75			TA	66	2694.61
1-3	-/2153	HA2	120	22.39	4-1	8/2272	HA2	109	24.91
		TA	89	7807.31			TA	81	9490.17
1-4	-/2710	HA2	158	41.56	4-2	8/2165	HA2	121	20.15
		TA	112	706.60			TA	91	11632.02
2-1	6/1474	HA2	78	6.97	4-3	8/2071	HA2	126	20.73
		TA	57	470.61			TA	92	9238.69
2-2	6/1261	HA2	55	5.66	4-4	8/2264	HA2	113	25.16
		TA	49	267.69			TA	89	7662.67
2-3	6/1443	HA2	89	5.98	5-1	9/2139	HA2	114	23.45
		TA	55	387.34			TA	92	33603.00
2-4	6/1382	HA2	59	5.59	5-2	9/2330	HA2	99	28.00
		TA	44	452.59			TA	82	28349.40
3-1	7/2014	HA2	114	21.34	5-3	9/2268	HA2	93	23.33
		TA	97	1722.38			TA	77	47113.36
3-2	7/2133	HA2	89	20.13	5-4	9/2394	HA2	120	26.58
		TA	67	2142.02			TA	91	39441.41

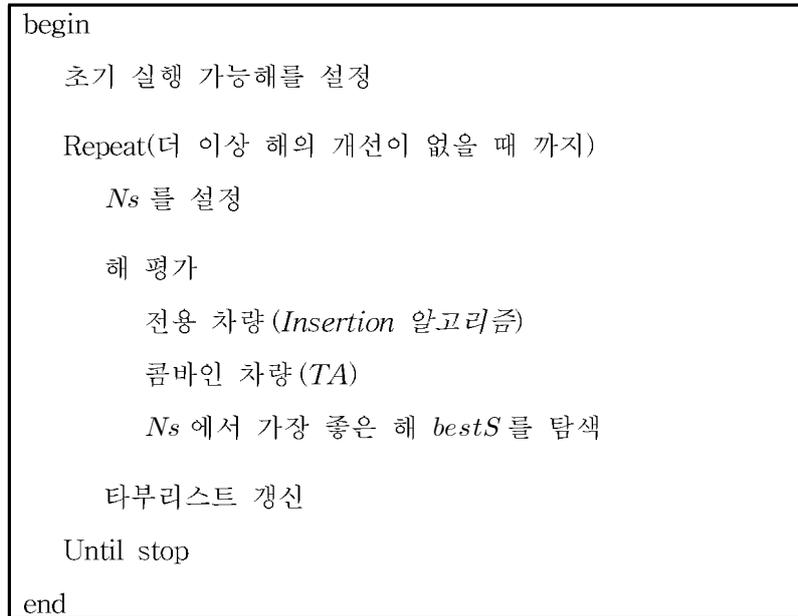
4.3 모든 차량의 형태를 고려한 문제

4.3.1 해법

모든 차량의 형태를 고려한 문제는 콤바인 차량만을 고려한 이전의 문제들과는 다르다. 모든 차량이 동일하다고 가정하면 모든 물량은 콤바인 차량에 할당하는 것이 효율적이다. 하지만 실제 현장에서 이용되는 차량의 대수가 유형별로 한정적이기 때문에 20-foot 전용 차량과 40-foot 전용 차량까지 고려하여 물량을 할당해야 한다. 그리고 전용 차량은 pick-up 노드를 방문한 다음 그 pick-up 노드와 대응되는 delivery 노드를 반드시 방문해야 하기 때문에 차량의 방문순서는 이전에 언급한 따라서 상기에서 언급한 클러스터 단계를 이용하여 문제를 m-TSP로 변형하는 것도 적합하지 않다.

본 문제에 대한 해법은 먼저 각 차량별로 물량을 할당하고, 각 차량에 할당된 물량에 대해 차량의 방문순서를 결정하는 방법으로 구성된다. 20-foot 전용 차량은 20-foot 컨테이너 물량만을 40-foot 전용 차량은 40-foot 컨테이너 물량만을 처리할 수 있다. 또한 이러한 전용 차량은 적재용량 제약에 의해서 각 차량에 할당된 pick-up 노드와 delivery 노드는 하나의 노드로 클러스터 할 수 있다. 따라서 전용 차량에 할당된 노드를 방문하는 순서는 상기에서 언급한 *Insertion 알고리즘*에 의해 간단하게 해를 구할 수 있게 된다. 반면에 각각의 콤바인 차량에 할당된 노드의 방문순서는 *Insertion 알고리즘*으로 초기해를 구성한 다음 4.2절에서 설명한 *TA* 해법을 이용하여 해를 구할 수 있다.

이와 같은 개념을 이용하여 본 논문에서는 모든 차량의 형태를 고려한 문제에 대한 해법으로 타부서치 알고리즘을 개발하였다. 평가함수는 차량의 총 운행시간의 최소화이다. <그림 4-8>은 타부서치를 이용한 해법의 구조를 설명한 것이다. N_s 는 이웃해 탐색 공간을 나타낸다.



<그림 4-8> 해법의 구조



<초기해 설정>

앞서 언급한 것처럼 모든 차량이 동일하다고 가정하면 콤바인 차량에 모든 물량(리퀘스트)을 할당하는 것이 전용 차량에 리퀘스트를 할당하는 것에 비해 더 효율적이다. 따라서 초기해를 설정할 때 리퀘스트를 콤바인 차량에 먼저 할당한 후 20-foot 컨테이너 리퀘스트는 20-foot 전용 차량에 40-foot 컨테이너는 40-foot 전용 차량에 할당한다. 그리고 각 차량은 하루 최대 운행 시간이 한정적이기 때문에 할당된 리퀘스트는 그 시간 내에 모두 처리할 수 있어야 한다.

본 문제에서 초기해를 설정하는 알고리즘은 <그림 4-9>와 같다. L_k 는 차량 k 에 할당된 리퀘스트를 모두 처리하는 데 걸리는 최대 시간이며, t_{ij} 는 리퀘스트 r_i 에 대응되는 pick-up 노드와 delivery 노드 사이의 거리이다. 그리고 T 는 차량의 하루 최대 운행 시간이며, θ 는 임의의 리퀘스트를 차량에 할당 할 때 그 차량의 운행시간 증가 값이다.

```

begin
  임의의 차량 ( $k$ )을 선택
  임의의 리퀘스트 하나 ( $r_i$ )를 선택된 차량에 할당
   $L_k = t_{ij}$ 
  Repeat(더 이상 할당할 리퀘스트가 없을 때 까지)
     $\theta$  가 최소인 리퀘스트를 선택
    If  $L_k + \theta \leq T$  then
      해당 리퀘스트를  $k$  에 할당
       $L_k = \text{Min}(L_k + \theta)$ 
    else stop;
  Until stop
end

```

<그림 4-9> 초기해 알고리즘



해는 <그림 4-10> 과 같이 각 차량에 할당된 리퀘스트로 구성된다. 차량은 유형별로 각각 2대씩 있다고 가정 (k_t : 20-foot 전용 차량, k_c : 콤바인 차량, k_f : 40-foot 전용 차량) 하고 리퀘스트는 20-foot 컨테이너 12개 {1, 2, ..., 12} 와 40-foot 컨테이너 8개 {13, 14, ..., 20} 으로 20개가 있다고 가정한다.

$$P = \begin{bmatrix} k_t 1 \text{ 에 할당된 리퀘스트} \\ k_t 2 \text{ 에 할당된 리퀘스트} \\ k_c 1 \text{ 에 할당된 리퀘스트} \\ k_c 2 \text{ 에 할당된 리퀘스트} \\ k_f 1 \text{ 에 할당된 리퀘스트} \\ k_f 2 \text{ 에 할당된 리퀘스트} \end{bmatrix} = \begin{bmatrix} 1, 8, 12 \\ 3, 4, 9, 11 \\ 2, 7, 14, 17 \\ 5, 6, 10, 20 \\ 13, 16, 19 \\ 15, 18 \end{bmatrix}$$

<그림 4-10> 해의 표현

위와 같이 초기해 알고리즘을 이용하여 문제에 대한 초기해를 결정한다. 실험을 통해 타부목록의 크기는 리퀘스트 수의 10% 로 설정하였으며, 타부목록은 해의 이동이 발생하면 갱신되고, 가장 먼저 등록된 순서대로 제거되는 방법으로 관리한다.

<이웃해 탐색 공간 결정>

N_s 는 타부서치의 탐색 공간 크기로서 작게 결정할 경우 탐색시간이 빠르지만 더 좋은 해를 얻을 가능성이 낮아지고, 크게 결정할 경우 더 좋은 해를 얻을 가능성이 높지만 탐색하는데 많은 노력이 필요하다. 따라서 N_s 의 크기를 결정하는 것은 타부서치의 계산 시간과 정확성에 중요한 요소라고 할 수 있다.

본 문제에서는 아래와 같은 방법으로 N_s 를 결정하였다. 현재해는 <그림 4-10> 에 표현된 것과 같다고 가정하였다.

Step 1 리퀘스트 인덱스에 따라 하나의 리퀘스트를 선정한다.

예) $k_t1 = (1\ 8\ 12)$ $k_c1 = (2\ 7\ 14\ 17)$ $k_f1 = (13\ 16\ 19)$
 $k_t2 = (3\ 4\ 9\ 11)$ $k_c2 = (5\ 6\ 10\ 20)$ $k_f2 = (15\ 18)$

Step 2 선정된 리퀘스트가 이동할 수 있는 모든 위치를 N_s 로 결정한다.

예) 아래와 같이 가능해인 경우를 N_s 로 한다.

$k_t1=(8\ 12)$	$k_c1=(2\ 7\ 14\ 17)$	$k_f1=(13\ 16\ 19)$	가능해
$k_t2=(1\ 3\ 4\ 9\ 11)$	$k_c2=(5\ 6\ 10\ 20)$	$k_f2=(15\ 18)$	
$k_t1=(8\ 12)$	$k_c1=(1\ 2\ 7\ 14\ 17)$	$k_f1=(13\ 16\ 19)$	불가능해
$k_t2=(3\ 4\ 9\ 11)$	$k_c2=(5\ 6\ 10\ 20)$	$k_f2=(15\ 18)$	
$k_t1=(8\ 12)$	$k_c1=(2\ 7\ 14\ 17)$	$k_f1=(13\ 16\ 19)$	가능해
$k_t2=(3\ 4\ 9\ 11)$	$k_c2=(1\ 5\ 6\ 10\ 20)$	$k_f2=(15\ 18)$	
$k_t1=(8\ 12)$	$k_c1=(2\ 7\ 14\ 17)$	$k_f1=(1\ 13\ 16\ 19)$	불가능해
$k_t2=(3\ 4\ 9\ 11)$	$k_c2=(5\ 6\ 10\ 20)$	$k_f2=(15\ 18)$	
$k_t1=(8\ 12)$	$k_c1=(2\ 7\ 14\ 17)$	$k_f1=(13\ 16\ 19)$	불가능해
$k_t2=(3\ 4\ 9\ 11)$	$k_c2=(5\ 6\ 10\ 20)$	$k_f2=(1\ 15\ 18)$	

위의 절차에서 N_s 중 불가능해가 발생하는 경우를 살펴보면 20-foot 컨테이너 리퀘스트가 40-foot 전용 차량에 할당된 경우나 그 반대인 경우, 그리고 차량의 하루 최대 운행 시간을 초과하는 경우가 있을 수 있다. 다시 말하면, 리퀘스트 1은 20-foot 컨테이너 리퀘스트이기 때문에 40-foot 전용 차량인 k_{f1} 과 k_{f2} 로 처리할 수 없다. 그리고 차량 k_{c2} 는 리퀘스트 1을 추가로 처리할 경우 차량의 하루 최대 운행 시간을 초과하기 때문에 $k_{c2} = (1\ 2\ 7\ 14\ 17)$ 은 불가능해라고 할 수 있다.

<평가 함수>

평가 함수는 차량의 총 운행 시간이다. 앞서 언급한 것처럼 전용 차량은 *Insertion 알고리즘*에 의해 전용 차량 전체 운행 시간의 최적 값을 구할 수 있다. 반면에 콤바인 차량은 4.2절에서 제시한 *TA* 해법을 이용하여 차량 방문 순서를 개선하는 방법으로 콤바인 차량 전체 운행 시간의 최적 값을 구할 수 있다. 따라서 차량의 총 운행 시간은 전용 차량의 총 운행 시간과 콤바인 차량의 총 운행 시간의 합으로 구할 수 있다.

<해의 탐색 절차>

초기해를 시작으로 하여 N_s 의 탐색을 계속 하게 된다. N_s 에서 도출된 값들 중 가장 좋은 값이 현재까지 도출된 가장 좋은 해보다 더 좋으면 현재까지의 최적해, 타부 리스트를 갱신한다. 반면에 현재까지 도출된 가장 좋은 해보다 좋은 경우가 없으면 다음 N_s 로 이동하여 탐색을 계속하게 된다. 탐색의 종료는 모든 N_s 의 실행가능 경우를 탐색한 후에도 더 이상 현재까지 도출된 가장 좋은 해가 갱신이 되지 않을 경우 종료한다. 탐색의 종료 후 가장 좋은 해를 도출하는 리퀘스트 할당과 차량 방문 순서를 최적 리퀘스트 할당과 차량 방문 순서로 선정을 한다.

4.3.2 실험 및 분석

위에서 제시한 해법을 다양한 문제사이즈와 상황에 적용시켜 분석을 실시하였다. <표 4-8> 과 같이 일일 발생하는 리퀘스트는 200 / 300 / 400 / 500 개로 설정하였고, 대기하고 있는 차량의 수를 차량의 유형별로 다양하게 고려하였다. 그리고 차량의 일일 최대 운행 시간을 480 분으로 설정하였으며, 각 노드간의 운행시간은 랜덤하게 발생하였다.

전체 100 가지의 경우에 대해 해법을 적용하여 실험을 실시하였으며, 각 경우별 가용 차량 대수, 차량의 총 운행시간, 해법의 수행 시간을 측정하였다. 측정한 결과는 <표 4-8> 과 같다.

각 리퀘스트 별로 실험 결과를 살펴보면 해법 수행 시간은 각각 평균 2.5948, 3.7638, 7.6069, 15.4245 초로 빠른 계산시간을 보였으며, 리퀘스트 수가 증가할수록 계산시간이 증가하는 것을 알 수 있다. 그리고 분석 결과 모든 경우에서 보유하고 있는 콤바인 차량의 대수가 적을수록 목적함수인 차량의 총 운행시간이 증가하였다. 이는 모든 차량이 동일하다고 가정하면 전용차량을 이용하는 것보다 콤바인 차량을 이용하는 것이 차량의 운행시간 측면에서 더 효율적이라는 것을 알 수 있는 결과이다.

<표 4-8> Experimental cases and results

Case	리퀘스트 수	운행차량 대수			총 운행거리	계산시간 (sec)	Case	리퀘스트 수	운행차량 대수			총 운행거리	계산시간 (sec)
1-1		9	10	9	3273	2.3291	3-1		17	20	18	8471	5.0322
1-2		9	10	9	3410	2.4531	3-2		17	20	18	9651	6.7968
1-3		8	10	9	4048	1.5468	3-3		17	20	18	9653	7.0781
1-4		9	10	9	3611	1.9365	3-4		17	20	18	9982	7.0634
1-5		9	10	9	3561	2.5312	3-5		17	20	18	9223	8.5000
1-6		8	11	8	3182	3.0312	3-6		16	21	17	9448	8.8896
1-7		8	11	8	3674	4.9843	3-7		16	21	17	8019	10.5322
1-8		8	11	8	3312	2.0458	3-8		16	21	17	9140	11.7959
1-9		7	11	8	3884	4.0312	3-9		16	21	17	9870	6.7968
1-10		8	11	8	3665	2.7500	3-10		16	21	17	9424	5.0781
1-11		9	9	10	3980	2.6572	3-11		17	19	17	9485	5.4072
1-12		9	9	9	4287	1.9531	3-12		17	19	17	10101	7.7968
1-13	200	9	9	10	5040	1.9687	3-13	400	17	19	18	9917	6.3750
1-14		10	9	10	4491	2.1718	3-14		17	19	18	9037	5.3906
1-15		10	9	10	2019	2.9853	3-15		17	19	18	9680	7.6396
1-16		10	8	9	3828	1.9687	3-16		18	18	18	8759	10.0634
1-17		9	8	10	3709	2.3583	3-17		18	18	18	9774	6.9521
1-18		9	8	10	4663	2.8583	3-18		19	18	19	6170	7.3134
1-19		9	8	10	5215	2.0146	3-19		18	18	19	9290	9.4531
1-20		9	8	10	4513	3.6718	3-20		19	18	18	7013	5.2666
1-21		10	7	11	4426	3.8896	3-21		18	17	18	9659	7.4218
1-22		10	7	11	3054	2.4052	3-22		19	17	18	8279	5.6083
1-23		10	7	10	4356	2.4687	3-23		18	17	19	10088	11.0468
1-24		11	7	11	4325	1.6083	3-24		19	17	19	6915	5.7656
1-25		11	7	11	3907	2.2500	3-25		18	17	19	9613	11.1084
2-1		13	15	13	3404	4.8438	4-1		21	25	22	10297	13.7959
2-2		12	15	12	6867	4.7969	4-2		21	25	22	11028	15.6563
2-3		12	15	13	7393	3.6719	4-3		21	25	22	11184	16.2022
2-4		12	15	12	7750	3.9688	4-4		21	25	22	11921	14.9531
2-5		13	15	13	3977	5.7813	4-5		21	25	22	10436	18.4697
2-6		13	16	13	5783	3.2969	4-6		20	26	21	12216	9.7646
2-7		13	16	13	7554	5.5928	4-7		20	26	21	11054	14.5167
2-8		13	16	13	4895	3.5469	4-8		20	26	21	12459	15.3584
2-9		13	16	13	3310	5.3281	4-9		20	26	21	12097	15.7031
2-10		13	16	12	5501	5.3438	4-10		21	26	21	8640	19.7734
2-11		13	14	14	8054	2.8281	4-11		22	24	22	5863	13.5938
2-12		13	14	14	6835	3.3428	4-12		21	24	21	10728	20.4219
2-13	300	13	14	14	6861	6.4697	4-13	500	22	24	22	7177	16.4688
2-14		13	14	13	5243	6.4531	4-14		21	24	22	7256	20.9531
2-15		13	14	14	7341	4.3138	4-15		22	24	22	5863	13.7667
2-16		11	17	12	6596	4.1719	4-16		20	27	20	11666	12.5469
2-17		12	17	12	6731	5.7188	4-17		20	27	21	11258	12.5928
2-18		12	17	12	3700	7.0489	4-18		20	27	20	11896	18.4053
2-19		12	17	11	6182	3.5147	4-19		20	27	21	12508	12.3281
2-20		11	17	12	6416	4.0615	4-20		20	27	20	12531	10.4678
2-21		14	13	14	5993	4.1104	4-21		22	23	23	12065	10.5615
2-22		13	13	14	7821	3.0928	4-22		23	23	23	9355	14.7197
2-23		13	13	14	6428	4.7031	4-23		22	23	23	8037	20.0938
2-24		13	13	14	7497	4.3916	4-24		22	23	22	12585	13.1865
2-25		13	13	14	6187	6.5000	4-25		23	23	22	5293	21.3115

제 5 장 결 론

국가 간 무역의 장벽이 사라진 오늘날 국제물류에 대한 관심은 계속 증대되고 있다. 따라서 국제물류의 수단으로 활용되는 컨테이너 물류에 대해 보다 많은 연구가 요구되고 있는 실정이다. 컨테이너 물류는 컨테이너 터미널에서의 물류와 컨테이너 운송 물류로 구분된다. 컨테이너 물류의 생산성은 이 두 가지 물류의 동기화 정도에 따라 결정되므로 컨테이너 운송 또한 매우 중요하다. 그리고 컨테이너 물류의 SCM 차원에서 특히, 근거리운송과 서틀운송의 효율적인 계획수립과 관리가 필요하다. 하지만 컨테이너 터미널 물류에 관한 연구에 비해 컨테이너 운송 분야의 학문적 연구는 매우 미약한 실정이다.

이에 따라 본 논문에서는 효율적인 컨테이너 공로 운송을 위한 컨테이너 차량 배차 계획에 대하여 연구하였다. 컨테이너 공로 운송의 특성에 따라 다양하게 문제를 설정하였으며, 각각의 문제에 대한 수리적 모델과 해법을 제시하였다. 제안된 해법은 다른 컨테이너 운송 문제의 서브 문제에 대한 해법이 될 수 있다. 또한, 실험 및 분석을 통해 해법의 효율성을 검증하였다. 특히, 국내 운송회사에서 수집한 자료를 이용하여 결과를 분석해 보았다는 점에 큰 의의가 있다고 할 수 있다.

향후 본 논문에서 고려하지 못한 제약들을 추가하여 더욱더 현실감 있는 해법을 개발할 수 있을 것이다. 그리고 모든 차량의 형태를 고려한 문제에서는 제안한 해법과 최적해의 비교 분석을 통해 해법에 대한 유용성을 검증하지 못하였다. 따라서 Cplex 나 ILOG 를 이용하거나 최적해를 구하는 알고리즘을 개발하여 유용성을 검증하는 단계가 필요하다. 또한 운송회사의 차량 운영방식(자차, 지입차, 용차)을 고려한 컨테이너 공로 운송 문제에 대한 연구 역시 필요할 것으로 보인다.

참 고 문 헌

고창성, 정기호, 신재영(2000), "컨테이너 셔틀운송을 위한 차량 대수 결정", 한국경영과학회, 17권 2호, pp.87-95

고창성, 정기호, 신재영(2004), "동적 수요와 세 가지 차량형태를 고려한 총괄 컨테이너 운송계획", IE Interfaces, 17권 1호, pp.71-77

고창성, 정기호, 신재영, 김홍배, 양천석, 박성찬(2002), "추계적 작업환경에서 컨테이너 운송 차량 대수 결정", 한국생산관리학회, 12권 3호, pp.17-32

김갑환, 고창성, 신재영(1998), "화물 수송 계획 및 운영의 정량적 모형에 대한 조사 연구", 대한산업공학회, 11권 1호, pp.1-14

신재영, 오성인(2008), "Combined chassis 트레일러를 이용한 컨테이너 차량 경로 문제", 한국항해항만학회 2008 공동학술대회, 논문집, pp.155-156

신재영, 오성인, 박종원(2009), "컨테이너 셔틀 서비스를 위한 차량 경로 문제의 근사적 해법", 한국항해항만학회, 33권 8호, pp.583-588

윤원영, 류숙재(2007), "내륙 운송 체계 하에서 컨테이너의 최적 운송관리에 관한 연구", 한국항해항만학회, 31권 2호, pp.207-209

Beraldi, P., Ghiani, G., Laporte, G. and Musmanno, R.(2005), "Efficient neighborhood search for the Probabilistic Pickup and Delivery Travelling Salesman Problem", Networks, 45, pp.195-198

Bianchessi, N. and Righini, G.(2007), "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery", *Computers & Operational Research*, 34, pp.578-594

Chung, K. H., Ko, C. S., Shin, J. Y., Hwang, H. and Kim, K. H.(2007), "Development of mathematical models for the container road transportation in Korean trucking industries", *Computers & Industrial Engineering*, 53, pp.252-262

Cordeau, J. F. and Laporte, G.(2003a), "A tabu search heuristic for the static multi-vehicle dial-a-ride problem", *Transportation Research Part B*, 37, pp.579-594

Cordeau, J. F. and Laporte, G.(2003b), "The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms", *4OR*, 1, pp.89-101

Cordeau, J. F. and Laporte, G.(2007), "The dial-a-ride problem: models and algorithms", *Annals of operations research*, 153, pp.29-46

Dethloff, J.(2001), "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up", *OR Spektrum*, 23, pp.79-96

Diana, M. and Dessouky, M. M.(2004), "A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows", *Transportation Research Part B*, 38, pp.539-557

Dumas, Y., Desrosiers, J. and Francois, S.(1991), "The pickup and delivery problem with time windows", *European Journal of Operational Research*, 54, pp.7-22

Fermin, A. T. M. and Roberto, D. G.(2006), "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers & Operations Research*, 33, pp.595-619

- Ganesh, K. and Narendran, T. T.(2007), "CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up", *European Journal of Operational Research*, 178, pp.699-717
- Gendreau, M., Laporte, G. and Vigo, D.(1999), "Heuristics for the traveling salesman problem with pickup and delivery", *Computers & Operations Research*, 26, pp.699-714
- Luo, Y. and Schonfeld, P.(2007), "A rejected-reinsertion heuristic for the static Dial-A-Ride Problem", *Transportation Research Part B*, 41, pp.736-755
- Mosheiov, G.(1998), "Vehicle routing with pick-up and delivery: tour-partitioning heuristics", *Computers & Industrial Engineering*, 34, pp.669-684
- Nagy, G. and Salhi, S.(2005), "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, 162, pp.126-141
- Roberto, W. C. and Alberto, C.(2007), "An effective and fast heuristic for the Dial-a-Ride problem", *4OR*, 5, pp.61-73
- Urban, K. P.(2006), "A guided simulated annealing search for solving the pick-up and delivery problem with time windows and capacity constraints", *International Journal of Logistics*, 9, pp.369-381
- Vigo, D.(1996), "A heuristic algorithm for the Asymmetric Capacitated Vehicle Routing Problem", *European Journal of Operational Research*, 89, pp.108-126
- Xiang, Z., Chu, C. and Chen, H.(2006), "A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints", *European Journal of Operational Research*, 174, pp.1117-1139

사랑하는 저의 가족에게
이 논문을 바칩니다.



감사의 글

지난 5년간 아낌없는 가르침과 훈계로 저를 지도해 주신 신재영 교수님께 진심으로 무한한 존경과 감사를 드립니다. 또한 바쁘신 일정에도 불구하고 저의 논문 심사를 맡아 세심한 지도를 아끼지 않으신 이철영 교수님과 박진희 교수님께도 깊이 감사드립니다. 그리고 석사 과정을 마치기까지 많은 가르침을 주신 곽규석 교수님, 남기찬 교수님, 신창훈 교수님, 김환성 교수님께도 감사드립니다.

이 논문을 쓰는 내내 많은 관심과 조언을 해주신 창윤이형과 수현이형, 그리고 항상 저에게 힘이 되어 준 면수형과 창원이형에게도 감사의 말을 전합니다. 대학원 시절 함께 해왔던 물류정보실험실 가족들, 종호형과 순철, 종원, 상진, 용이에게도 감사의 마음을 전합니다.

이제 작은 결실을 맺는 하나의 과정을 거치면서 주위 모든 분들에게 감사함을 다시금 느끼며, 앞으로도 변함없는 애정과 지도로 이끌어 주시길 바람과 함께 이 논문을 바칩니다.