

717
726311
1
= 2

공학석사 학위논문

화물 컨테이너 운송을 위한
Tabu Search 기반의
차량경로계획

이 논문은 공학석사 학위논문으로 제출함



2003년 2월

부 경 대 학 교 대 학 원

산 업 공 학 과

장 동 원

장동원의 공학석사 학위논문을 인준함

2002년 12월 26일

주심 공학박사

옥영석



위원 공학박사

이운식



위원 공학박사

구평희



Tabu Search Based Vehicle Routing Planning for Freight Container Transportation

Dong-Won Jang

*Department of Industrial Engineering, Graduate School,
Pukyong National University*

Abstract

This thesis addresses vehicle planning and routing problems in freight container transportation systems where a number of loaded containers are to be delivered to their destination places and some empty containers are transported from locations with excess empty containers to shortage locations. The system under consideration is static in that all the transportation requirements are known at the beginning of a planning horizon. A two-phase vehicle planning and routing procedure is presented for the static container transportation system. In the first phase, a mixed integer programming model is presented to determine optimal flows between container handling centers and provide the minimum number of vehicles required. Based on the results from the first phase, a tabu search based procedure is proposed in the second phase where vehicle routings are determined with the least number of vehicles. Some simulation experiments are performed to compare the performance of the new procedure and some existing procedures under various container transportation settings. The experiments show that container transportation can be realized with less number of vehicles or less completion time in the new model compared with the existing procedures. The research result in this thesis is believed to be used in some other transportation environment such as AGV systems, unit-load transportation system, similar to the container transportation system.

keywords : tabu search, transportation, vehicle routing

목 차

제 1 장 서론

- 1.1 연구의 배경 및 목적1
- 1.2 연구의 방법 및 구성3

제 2 장 기존 연구현황

- 2.1 수송계획
 - 2.1.1 수송계획문제5
 - 2.1.2 차량배차문제5
 - 2.1.3 차량경로문제6
- 2.2 컨테이너 운송계획
 - 2.2.1 컨테이너 운송 개념과 문제정의12
 - 2.2.2 컨테이너 운송 최소차량대수 결정14
 - 2.2.3 컨테이너 운송계획 및 운영16
- 2.3 연구의 문제환경 정의
 - 2.3.1 컨테이너 운송시스템19

제 3 장 화물 컨테이너 운송을 위한 탐색적 해법

- 3.1 연구의 탐색적 해법 설명22
- 3.2 Tabu Search 이론23
- 3.3 공차운행 최소화 수리모형29
- 3.4 Tabu Search 기반의 차량경로계획 알고리즘31

제 4 장 수치예제 및 성능분석	
4.1 수치예제	38
4.2 성능분석	
4.2.1 기존의 방법과 비교분석	43
4.2.2 이동물량정보가 랜덤한 경우	46
4.2.3 확률적인 시간 분포를 가지는 경우	53
제 5 장 결론 및 발전방향	56
참고문헌	58
[첨부1] M_Size 10개 개별 데이터 결과	60
[첨부2] 확률적인 시간분포를 가지는 경우의 개별 실험결과	70
[부록1] AMPL 공차이동 최소화 수리모형	73
[부록2] Visual Basic6.0을 이용한 프로그래밍	82

그림 목차

<그림1> 수송계획문제의 개략적인 분류도.....	11
<그림2> 기본적인 컨테이너 운송	13
<그림3> 컨테이너 운송 문제 정의	13
<그림4> 컨테이너 수송문제의 개략적인 분류도.....	18
<그림5> 물류용기 공동화 사용하에서의 컨테이너 운송계획	20
<그림6> Tabu Search알고리즘 전체 개략도	23
<그림7> 메모리와 전략의 관계	26
<그림8> 가장 좋은 후보선택	27
<그림9> Tabu Search 기법 기본	28
<그림10> Busiest_V차량내 작업의 재삽입	35
<그림11> Busiest_V차량내의 작업삭제	36
<그림12> 알고리즘 흐름도	37
<그림13> 초기해 생성	40
<그림14> 생성된 최종해	41
<그림15> 각 반복횟수별 makespan의 변화	42
<그림16> 각 반복횟수별 완료시간 합의 변화	43
<그림17> Insertion방법 차량대수 적용시 makespan감소.....	50
<그림18> Insertion방법 480분 기준시 차량대수 절감 정도.....	50
<그림19> Tabu Search의 makespan을 운행시간으로 적용시 차량대수 절감 정도.....	51
<그림20> 확률적인 이동시간을 가질 경우의 makespan의 변화	55
<그림21> 확률적인 이동시간을 가질 경우의 완료시간 합의 변화	55

표 목차

<표1> 권역별 수출 컨테이너 화물 장치 현황	1
<표2> 적입된 컨테이너 이동물량과 빈 컨테이너 여유/부족 물량정보 유입유출표	38
<표3> 지점간 이동시간	39
<표4> Insertion 방법 차량대수 적용시.....	45
<표5> 차량운행시간이 500분인 경우	45
<표6> 차량 14대인 경우의 실험결과	46
<표7> 랜덤 생성 이동물량과 최소필요차량대수	47
<표8> 랜덤생성 데이터 평균-makespan기준, 첫물량 랜덤선택...	48
<표9> 랜덤생성 데이터 평균-makespan기준, 첫물량 근거리선택	49
<표10> 랜덤생성 40개 데이터 평균.....	52
<표11>확률적인 이동시간을 가지는 경우.....	54

제 1 장 서론

1.1 연구의 배경 및 목적

부산은 국내 물류의 최대 거점지역이다. 부산항은 지리적으로 유럽, 아시아, 북미를 연결하는 주 항로상에 위치하여 동북아 중심항만으로 중요한 역할을 수행하고 있다. 세계 100여개국 500여 항만과 교류하여 연간 13,000여척의 외국적 선박이 기항하고 있으며, 특히 세계 유수의 6개 항만과 자매결연을 맺고 항만운영에 관한 정보교류 및 친선을 도모하고 있다.

<표1>의 한국 컨테이너 부두공단의 2000년 말 자료에 따르면 권역별로 볼 때 부산권이 적 컨테이너와 공 컨테이너를 포함하여 총 474,648TEU의 수출 컨테이너를 처리한다. 총 651,139TEU의 수입 컨테이너를 포함하면, 이것은 국내에서 다른 지역보다 월등히 높은 수치이다.

<표1> 권역별 수출 컨테이너 화물 장치 현황

구분	합계	적 컨테이너 단위(TEU)			
		계	CY장치	CFS조작	공 컨테이너
합계	1,542,362	1,191,604	1,027,355	164,247	350,758
수도권	313,638	275,144	233,425	41,719	38,494
부산권	474,648	220,900	181,767	39,133	253,748
경남권	258,246	237,274	216,667	20,607	20,972
경북권	220,057	207,467	182,348	25,119	12,590
충부권	134,300	125,590	105,364	20,226	8,710
호남권	116,892	101,970	89,156	12,814	14,922
강원권	24,581	23,259	18,628	4,631	1,322

(<http://www.kca.co.kr>)

부산은 입항지역의 15개소 ODCY, 재송지역의 4개 ODCY, 수영지역의 9개 ODCY, 철도지역의 7개 ODCY, 기타 2개의 ODCY를 가지고 있으며, CY별 수출입(직/공별)내역을 살펴보면 수출 1,542,362 TEU, 수입 1,559,038TEU를 처리하고 있다(<http://www.kca.co.kr>).

본 연구는 이러한 화물창고, 터미널, 부두 등의 물류센터간의 컨테이너 운송에 있어서 보다 효율적으로 운송을 하기 위한 차량경로 계획에 중점을 둔다. 특히 화물이 적입된 컨테이너의 운송계획뿐만 아니라 목적지가 정해지지 않은 여유분의 빈 컨테이너 용기를 종합적으로 고려하여 최적의 운송계획을 수립한다. 이것은 물류합리화를 목표로 적용이 확대되고 있는 물류용기 공동화 사용의 화물컨테이너 운송체계로 볼 수 있고 수송효율 제고, 물류비 절감 그리고 효율적 운영 등의 이점을 낳는다.

최근 들어 물류관리의 중요성이 인식되어 기업경영에 적용되게 된 직접적인 원인은 물류 환경의 변화에 따른 것이다. 기업 간 경쟁이 심해진 상황에서 고속성장이 주춤하고 소비자의 기호가 다양화되면서 다품종 소량생산 체제로 이어지고, 판매전략에 있어서도 종전의 기업사이의 가격 경쟁력만으로는 불충분하여 고객 서비스 경쟁이 중요성을 더해가고 있다. 따라서 수배송 관리 및 재고관리 등을 통한 물류비 절감과 물류 서비스 제고를 통한 고객만족 경영이 기업의 주요한 부분으로 여겨지고 있다. 이러한 물류 환경의 변화에 적응하는 것은 효율적인 물류관리를 통해서만 가능한 것이므로 물류관리의 중요성이 한층 강조된다. 즉 고도의 물류 서비스를 제공함으로써 기업의 경쟁력을 강화시키는 것이다. 물류관리의 비효율성으로 인한 물류비의 상승은 제품의 원가를 상승시키고 결국 기업의 경쟁력을 상실시키기 때문이다.

수배송 물류는 장소적인 이동과 시간적인 극복을 통하여 재화와 용역의 효용가치를 증대시키는 행위를 의미한다(송계의 (1998)). 수배송 물류의 수단은 산업사회의 다양화에 따라 각종 수단이 출현하고 있는데, 크게는 자동차, 철도, 선박, 항공기 등으로 분류할 수 있으며, 그 형태로는 수송과 배송으로 나눌 수 있다. 수송은 재화와 용역을 물류 거점간에 이동시키는 행위를 말하고 배송은 재화와 용역을 물류 거점에서 고객 또는 폐기장소로 이동시키는 것을 말한다.

물류 활동에서 수배송이 차지하는 비중이 물류비용 측면에서 본다면 가장 크다. 특히 최근 들어서는 극심한 교통정체로 각 기업들의 수배송 효율이 크게 떨어지고 있고, 수요의 패턴변화에 따른 기업에의 소량, 다빈도, 긴급 수배송 요구가 점차 증가하고 있어, 물류비용의 절감차원뿐만 아니라 고객들에 대한 물류 서비스의 향상이라는 측면에서도 수배송의 효율화가 차지하는 비중은 크다.

1.2 연구의 방법 및 구성

본 연구의 목적은 앞 절에서 언급하였듯이 컨테이너 운송을 위한 효율적인 차량경로계획을 수립하는 것이다. 연구의 방법을 살펴보면 주어진 물량정보를 이용하여 수송계획법을 기초로 최소차량대수를 산출하고, 이러한 정보를 기반으로 Tabu Search 알고리즘을 적용한 새로운 탐색적 해법으로 최적의 차량운송경로를 도출한다. 그리고 유사 컨테이너 운송환경에서의 기존의 방법들과 비교실험을 통하여 알고리즘의 타당성을 검증한다.

본 연구의 구성은 2장에서 기존 연구현황을 다룬다. 2.1절에서는

수송계획문제를 설명하고, 차량배차부분과 차량경로부분으로 나누어 자세히 소개한다. 2.2절에서는 컨테이너 운송계획에 관해서 소개하고 2.3절에서는 본 연구에서 다루는 컨테이너 운송문제의 환경을 설명한다.

3장은 3.1절에서 본 연구가 제시한 탐색적인 해법에 관해서 설명한다. 3.2절은 연구에서 적용된 Tabu Search의 기본적인 이론을 설명하고, 이어 3.3절에서는 빈 컨테이너의 이동을 고려한 최소차량대수 산출법에 대해서 논한다. 3.4절에서는 최적의 차량경로를 도출하기 위한 Tabu Search 알고리즘의 자세한 내용을 기술한다.

4장은 제시한 Tabu Search 알고리즘의 성능을 분석한 장으로써 4.1절에서는 현장의 데이터로 구성된 기본예제를 Tabu Search 알고리즘에 적용한 수치예제 결과를 보여준다. 4.2절에서는 성능을 분석한 절로써 랜덤하게 생성한 물량정보를 이용한 경우, 어떠한 결과를 낳는지 비교실험을 통하여 평균적인 결과를 제시한다. 그리고 이동시간이 확률적인 시간분포를 가지는 경우의 실험결과를 비교 평가한다.

5장은 물류 부분에 있어서 컨테이너 물류의 발전방향을 제시하고, 앞으로의 연구과제에 대한 내용을 기술한다.

제 2 장 기존 연구현황

2.1 수송계획

2.1.1 수송계획문제

수송과 관련된 물류시스템은 (1)입지선정문제 (2)수송계획문제 (3)수송수단 설계문제 (4)터미널 설계문제 등 크게 4가지로 나누어 활발한 연구가 이루어지고 있다(김갑환 외2(1998)). 입지선정문제는 화물 또는 서비스의 공급을 원활히 할 수 있도록 관련 시설 또는 설비를 어디에 위치시킬 것인가를 다루는 문제이다. 수송계획문제는 수송수단을 통해서 화물을 수배송하는 문제이다. 수송수단 설계문제는 수송업무를 수행하는 운송업체, 기업 또는 공공기관의 물류팀 등이 어떻게 수송수단을 선택할 것인가 하는 문제이다. 터미널 설계문제는 터미널의 시설능력을 어떻게 추정할 것인가 하는 문제이다.

2.1.2 차량배차문제(Vehicle Dispatching Problem)

수송계획문제는 차량배차문제(Vehicle Dispatching Problem: VDP)와 차량경로문제(Vehicle Routing Problem: VRP)로 나누어 연구가 이루어지고 있다. 본 연구에서는 차량경로문제에 더욱 비중을 둔다. 차량배차문제(Vehicle Dispatching Problem: VDP)는 수송요구가 있을 때 어떤 차량에 배차할지를 다루는 dispatching문제로 최근 들어 수작업을 통한 노력과 시간을 단축시키기 위한 차량자동배차에 관한 연구들이 이루어져 GIS(Geographic Information System)

나 GPS(Global Positioning System)을 이용하여 편리한 사용자 인터페이스를 제공함으로써 수정작업의 편의를 도모하거나 수배송 업무 수행 중 발생하는 문제를 실시간으로 추적하여 대처하는데 주로 활용하는 수배송 지원시스템들이 있다. 실제기업에서 수행되고 있는 일반적인 배차계획 지원시스템은 각 물류센터에서 숙련된 배차담당자가 기초 데이터를 통한 가상배차 지원시스템을 장시간에 걸쳐 수정하여 배차계획을 확정하면 차량별 피킹 리스트를 통해 해당 물품을 오더피킹, 상차하여 배송하게 된다. 현실적으로 고려되어야 하는 여러 제약조건을 고려하여 임시차 배차알고리즘, 지입차 배차 알고리즘, 미배차 물량처리 알고리즘, 배차 완료조건 알고리즘, 기사배정 알고리즘 등이 최근 연구되었다(이명호 외2(2000)).

2.1.3 차량경로문제(Vehicle Routing Problem)

차량경로문제(Vehicle Routing Problem: VRP)는 수송요구가 있는 물량을 어떤 순서로 수송할지를 다루는 문제로 수송계획문제에서 가장 일반적인 것이다. 여러 제약환경에 따라 차량경로문제(Vehicle Routing Problem: VRP), 시간제약을 가지는 차량경로문제(Vehicle Routing Problem with Time Windows), 차량일정계획문제(Vehicle Scheduling Problem: VSP), 차량경로 일정계획문제(Vehicle Routing and Scheduling Problem: VRSP), 시간대에 따른 차량경로문제 등으로 구분되어 연구가 이루어져 왔으며 1959년 Dantzig와 Ranser에 의해 최초로 제기된 이후 꾸준한 관심의 대상이 되었다.

차량경로문제를 한 차량이 여러 도시를 방문하는 외판원문제

(Traveling Salesman Problem: TSP)와 여러 차량들로 수배송하는 차량경로문제로 구분하기도 한다(백시현과 김내현(1998)). 그러나 도시들의 집합에서 최단경로(Shortest Route)를 찾는 문제로 출발한 곳으로 되돌아오는 문제를 통칭해서 외판원문제(Traveling Salesman Problem: TSP)라고 한다. 이러한 외판원 문제는 고전적인 최적화 문제로 해밀턴 순환로(Hamilton Cycle)라고도 부르며 NP-Complete문제이다. TSP는 문제크기가 커지면 계산시간이 지수적으로 증가하기 때문에 최적화 모형으로는 해결하기가 어려워 많은 관련 연구들은 발견적 알고리즘(Heuristic Algorithm)의 제안에 초점을 맞추고 있다.

TSP의 목적은 각 연구의 문제상황 자체내에서 조정되는 결정변수의 최적값을 찾는 것으로 이러한 접근법은 다단계의 문제해법(Multistage Problem Solving)이라 부른다(김창은(1990)). TSP유형의 문제로는 도시에서 서로 다른 가게에 제품을 배달하는 경로(Delivery Route), 신문이나 우편물의 배달 또는 수거, 도로청소, PCB보드(Printed Circuit Board)에의 부품삽입, 차량의 경로선택 등이다.

차량경로문제(Vehicle Routing Problem)의 기본은 TSP로 차량의 대수에 따라 Single TSP와 Multi TSP로 구분이 되어진다. Single TSP문제에 대한 최적해법으로는 1963년 발표된 Little과 Murty의 분지한계법을 기본으로 한다. 이를 1971년 Bellmore 와 Malone이 NP-hard이므로 발견적 기법으로 개선시켰다. MTSP(Multi Traveling Salesman Problem)는 M명의 외판원과 N명의 고객이 있을 때 각 고객은 한 명의 외판원에 의해 반드시 한번만 방문될 때 M명의 외판원에 의한 총 여행거리를 최소화하는 방문경로를 찾는

것이다. Multi TSP는 차량집결지(depot)의 수에 따라 단일 출발지와 복수출발지로 나누어지고 단일출발지는 목적지에 따라 단일 목적지, 복수 목적지 등으로 구분되어진다.

배송센터 전체 문제는 MTSP문제로 볼 수 있으며 대리점이나 고객에게 차량이 할당된 후에는 TSP문제가 된다. 이는 1974년 Bellmore와 Hong에 의해 변환(MTSP-TSP)할 수 있음을 확인하였다. 제품의 배달과 더불어 용기나 팔레트를 회수하여야 하는 문제들이 있는데 주류나 음료회사의 배달량과 같은 부피의 병들을 의무적으로 회수해야하는 경우가 이에 해당한다. 대리점 또는 고객에 대한 제품 배달량과 회수해야 할 물량이 확정된 경우에 배달량을 모두 배달하면서 용기의 회수량을 최대로 하는 최단경로를 찾는 데, TSP 제약식과 함께 비선형 제약식을 포함하므로 최적해를 구하기는 어렵다. 그러므로 비선형 제약식을 제거한 TSP문제를 풀어 배달을 위한 최단경로를 얻은 후에 비선형 제약식을 만족하도록 하면서 앞에서 얻은 최단경로로부터 단계적으로 회수량을 증가시켜 가며 각 단계에 대한 경로를 찾는 기법을 이용한다(김내헌(1993)).

TSP는 환경의 제약에 따라 Bottleneck TSP, Time Windows를 갖는 TSP, Stochastic TSP 등으로 구분되기도 한다. 거리의 특성별로 TSP를 구분하면 유클리디안 거리를 이용한 연구, 렉티리니어 거리를 이용한 연구, 체비세프거리를 이용한 연구로 구분할 수 있다(백시현과 김내헌(1998)).

TSP적인 상태에서 문제의 답을 찾는 시스템적인 기법을 Dynamic Programming이라고 하고 그 동안 다양한 조건에서의 선형/비선형 목적 함수를 가진 문제를 풀기 위한 많은 알고리즘들이 개발되어져 왔다. 해법에 따라 최적화 수리모형과 휴리스틱

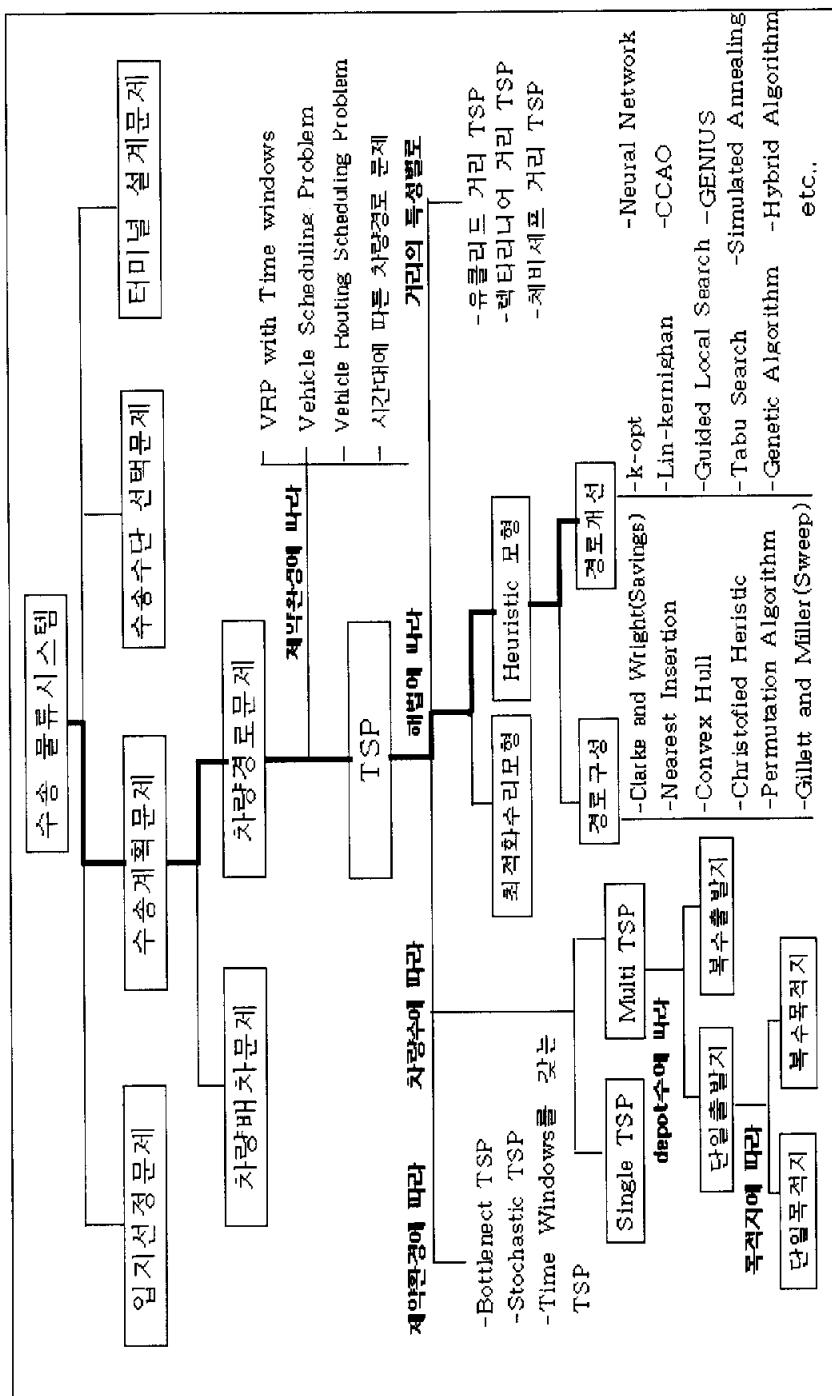
(Heuristic)모형으로 나뉘어지는데 최적화 모형은 문제의 크기가 커지면 해답을 얻기가 불가능한 NP-Complete 또는 NP-Hard문제이므로 이에 대한 대안이 휴리스틱(Heuristic)모형이다. 휴리스틱의 기본은 초기 실행 가능해(Initial Feasible Solution)에서 시작을 하고 각 반복구간에서 실행가능(feasible)하고 현재 목적함수 값보다 더 좋은 해로 진보하는 단계를 거치고 더 이상 목적함수를 향상시키는 실행 가능한 이웃해가 없으면 종료하는 구조로 이루어져 있다.

기존의 휴리스틱 방법들의 접근방식을 보면 좋은 초기해를 구성하는 방법, 좋은 해를 구성하도록 삽입하는 방법, 경로를 개선시키는 방법으로 최적해를 구하고 있다. 경로구성(Tour Construciton)방법과 경로개선(Tour Improvement)방법으로 나누어 알고리즘을 살펴본다. 경로구성방법은 빠른 시간에 국지탐색을 할 수 있는 장점이 있으나 경로개선방법보다는 해가 나쁘다. 그래서 보통 경로개선방법의 초기해로 경로구성방법을 사용한다.

경로구성은 Clarke 와 Wright의 Savings(1964), Nearest Insertion, Convex Hull(1998), Christofides Heuristic, Permutation Algorithm, Gillett 와 Miller의 Sweep(1960) 방법들이 유명하다(권상호와 강맹규(1999), 백시현과 김내현(1998)). 경로개선은 k-opt(1973), Lin-kernighan(1973), Guided Local Search(1999), Tabu Search, Genetic Algorithm, Branch and Bound(1963), Neural Network, Golden과 Stewart의 CCAO(1985) , Gendreau, Hertz 와 Laporte의 GENIUS(1992), Simulated Annealing, Hybrid Algorithm, MCH Algorithm(1999)등이 유명하다(권상호와 강맹규(1999), 백시현과 김내현 (1998)). 이외에도 여러 알고리즘들로 복합된 해법들이 있는데 휴리스틱 알고리즘의 사용은 좋은 해를 주지만 최적해를 보장

해 주지는 않는다. 따라서 휴리스틱은 보통 특정문제 구조에 맞춘 다항식 알고리즘이다. 휴리스틱을 세분화하면 Classical Heuristic과 Metaheuristic으로 분류할 수 있다. Classical Heuristic은 1960년과 1990년대 사이에 대부분 발전을 하였고, 현재의 대부분의 절차들이 Classical Heuristic에 속한다. 이러한 방법들은 적당한 계산시간 (computing time)내에서 좋은 해를 발견해 내고, 실제환경에서 조건에 변화가 있을 때 확장이 용이하다. 따라서 상업적으로 널리 이용되고 있다. Metaheuristic은 Classical Heuristic에 이어서 나온 것으로 정교한 근접 탐색법(search rules), 기억구조(memory structures), 해의 재조합 등 여러 기법이 있다. Classical Heuristic에 비해 계산시간이 증가하지만 우수한 해를 산출해낸다. 본 연구에서 사용된 Tabu Search기법은 Metaheuristic에 속한다.

앞서 설명한 수송계획문제에 관한 전제적인 내용은 <그림1>과 같이 개략적인 분류도로 표현할 수 있다.



<그림1> 수송계획문제의 개략적인 분류도

2.2 컨테이너 운송계획

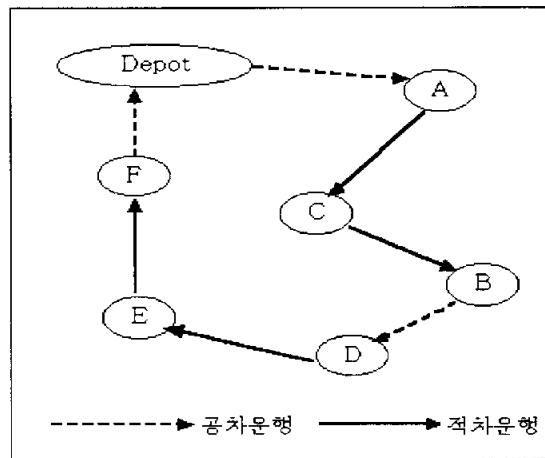
2.2.1 컨테이너 운송 개념과 문제 정의

수송계획 문제에서 또 다른 한 부분을 차지하고 있는 것은 컨테이너 운송계획이다. 1970년대 우리나라가 산업화에 박차를 가하게 된 이후로 수출입 물량의 증대와 더불어 컨테이너 화물의 증대는 불가피하게 그 규모가 커지게 되었고 그 중요성 또한 크게 대두되었다. 이에 컨테이너 화물 수송차량에 의한 물류의 수송문제에도 관심을 두게 되었다.

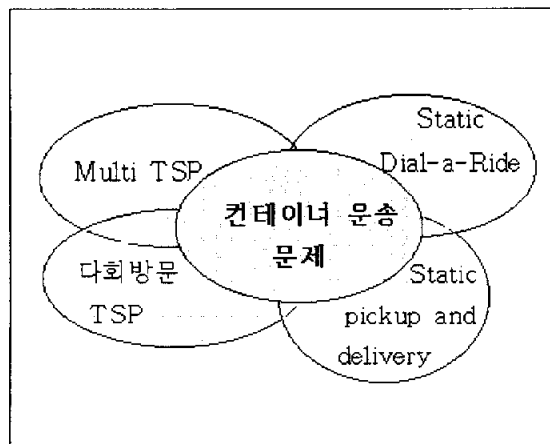
일반적인 컨테이너 운송이란 컨테이너에 미리 화물을 적입하여 운송하는 Unit Load 시스템의 일종으로 송하인으로부터 최종 수하인에 이르기까지 컨테이너로 화물을 운송하기 위해 물류의 전 구성요소를 결합하는 것을 말한다. 따라서 컨테이너에 적입된 내용물을 운송수단이 바뀌더라도 재적입이나 적출없이 그대로 운송이 가능하므로 경제성, 신속성, 안전성 면에서 큰 경제적 이익이 있다(이철영(1998)).

컨테이너 수송은 적용범위, 컨테이너의 규모, 컨테이너의 자재 형태, 수송방법과 같은 요소들로 인해서 매우 복잡한 문제이다(고창성과 신재영(2002)). 차량의 용량이 1이상이고 1회 방문이 그 기본을 이루고, 주어진 차량집결지(depot)에서 차량용량 제약을 만족하면서 여러 고객 노드들을 순회해야하는 일반적인 차량경로문제와는 달리 컨테이너 운송문제는 상차(pickup)지점과 하차(delivery)지점이 주어진 다수의 컨테이너 물량을 일정한 수의 트럭으로 다회 방문을 통해 수송하는 문제이다. 한번에 1개의 컨테이너만을 운송할 수 있는

며 여러 가지 실질적인 제약이 고려되기도 한다(김동희 외2(1997)). 따라서 컨테이너 운송계획은 Pickup and Delivery 문제이고, 차량용량이 1인 Multi TSP의 일종으로 사전에 모든 물량흐름의 정보가 있는 Static한 상황으로 Dial-a-Ride문제이자 Bidin et al(1983)이 분류한 트레일러를 트랙터에 연결하여 이동하는 트랙터-트레일러 운송체계라고 할 수 있다(고창성 외2(2000)). 기본적인 컨테이너 운송의 형태는 <그림2>와 같다. 컨테이너 운송문제는 <그림3>과 같이 표현할 수 있다.



<그림2> 기본적인 컨테이너 운송



<그림3> 컨테이너 운송 문제 정의

2.2.2 컨테이너 운송 최소차량대수 결정

컨테이너 운송에 있어서 다수의 물량을 일정수의 트럭으로 수송해야하기 때문에 차량대수의 결정이 중요한 부분을 차지한다. 대부분의 운송업체에서는 경험에 의해 대략 필요한 차량을 결정하고 있다. 그러나 AGV(Automated Guided Vehicle)시스템에서 활용되는 Maxwell모형이 최소의 차량대수를 결정하는데 있어서 이용가치가 높다(Maxwell과 Muckstadt(1982)). 작업장간 운행시간과 AGV이동자료를 이용하여 Loaded Vehicle의 이동대수와 Unloaded Vehicle의 이동대수를 산출하여 Vehicle의 수송능력시간을 나누어 최소 AGV 대수를 구하는 것을 컨테이너 운송 차량대수를 구하는데 동일하게 적용한다. 우선 총 차량 이동시간을 최소로 하는 수리모형은 다음과 같다

$$\text{Min} \sum_i \sum_j t_{ij}^a v_{ij} + \sum_i \sum_j t_{ij}^c x_{ij} \quad (1)$$

subject to :

$$\sum_j x_{ij} = nf(i) \quad \text{for all } nf(i) > 0 \quad (2)$$

$$\sum_k x_{ki} = -nf(i) \quad \text{for all } nf(i) < 0 \quad (3)$$

$$x_{ij} \geq 0 \text{ 인정수} \quad (4)$$

x_{ij} 차량이 공차로 장소 i 에서 장소 j 로 이동하는 횟수

v_{ij} 적입된 컨테이너를 싣고 장소 i 에서 장소 j 로 이동해야 하는 물량

t_{ij}^a 장소 i 에서 장소 j 로 적입된 컨테이너를 싣고 이동하는데 소요되는 시간

t_{ij}^c 장소 i 에서 장소 j 로 공차로 이동하는데 소요되는 시간

$nf(i)$ 차량의 순흐름

목적식(1)은 적재이동시간과 공차이동시간을 최소로 하는 식이고 제약식(2)(3)은 물량의 입출력에 있어서 양의 값을 갖는 지점은 공차이동이 시작되어 나가는 것을 의미하고 음의 값을 갖는 것은 이 지점으로 공차이동이 되도록 하는 것을 의미한다. 목적식의 최소이동시간에서 차량의 가용시간을 나누면 필요한 최소의 차량대수를 구할 수가 있다.

그러나 Maxwell모형에서는 각 차량이 가용한 시간 중 전혀 시간 손실이 발생하지 않는다는 가정과 모든 노드들을 연결시켜 총 운행시간을 계산하였으나 실제로는 한 대의 차량이 하루 운행시간 범위 내에서만 연결되며, 가상적으로 연결된 다음 노드로의 이동은 실제 발생하지 않는다. 이 노드는 새로이 할당되어야 할 차량의 출발 노드가 되는 것이다. 이러한 단점으로 인해서 컨테이너 운송문제를 Multi TSP로 변환하여 새로운 차량대수 결정 방법을 제시한 연구가 있다(고창성 외2(2000)). 이 방법은 현행 시스템의 각 지점사이의 컨테이너 수송요구 물량의 흐름을 나타내는 물리적 네트워크를 논리적 네트워크로 변환함으로써 수리적 모형을 구축하고, 기존의 Insertion Heuristic을 사용하여 소요 차량대수가 최소가 되는 해법 도출 과정을 제시하고 실제 현장자료에 기초한 예를 제시하였다. 또한 다른 연구에서는 차량이동시간 및 작업시간을 나타내는 확률변수의 평균치를 기준으로 하여 차량대수를 최소로 하는 수리적 모형을 제시하였다(고창성 외5(2001)).

하나의 센터와 센터에 연결된 여러 개의 터미널로 구성된 방사형 물류체계(Hub and Spoke)에서 최소차량대수를 구한 연구도 있는데 과잉터미널과 부족 터미널에 의하여 점유된 수송장비의 평균과 분산으로 각 과잉터미널과 각 부족 터미널의 평균과 분산을 구하여

수송 네트워크에서 필요한 수송장비의 분포를 이용하여 최소차량대수를 구하였다(Du와 Hall(1997)). 터미널의 수가 충분히 많을 경우 각 터미널이 점유한 수송장비의 합을 정규분포로 근사시킬 수 있는 점을 이용한 연구이다.

Du와 Hall(1997)의 연구에서 품질될 확률만을 고려하여 동일한 기간에 품질된 수송장비의 수를 고려하지 않은 점을 개선하여 다단계 재고시스템의 분산재고관리 정책을 채택하여 관련비용 즉, 빈 수송장비의 운행비, 재고유지비, 품질비와 수송장비 투자비를 최소화하는 수송장비의 보유대수 결정을 다룬 연구가 있다(서순근과 이병호(1998)).

2.2.3 컨테이너 운송계획 및 운영

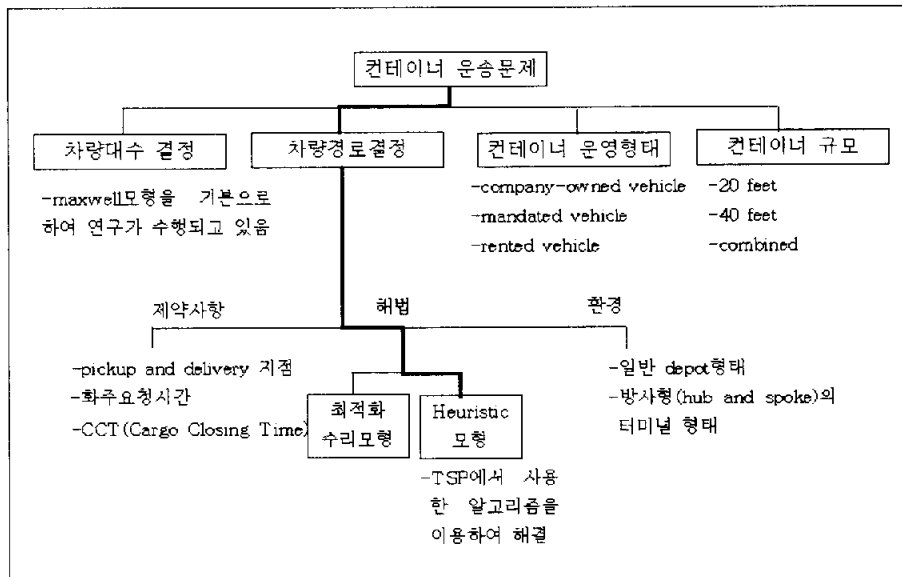
컨테이너 운송에 있어서 여러 제약사항을 살펴보면 Pickup and Delivery 지점, 화주요청시간, CCT(Cargo Closing Time: 선적선 접안 전 2시간)와 같은 실질적인 시간제약들이 주를 이룬다(김동희 외 2(1997)). 최근 컨테이너 차량의 형태를 고려하여 총괄수송계획이 연구되었다(고창성과 신재영(2002)). 컨테이너의 차량을 비용 발생면에서 분류를 하여 Company-owned truck, Mandated truck, Rented truck 3가지로 구분하여 Company-owned truck과 Mandated truck을 기준으로 수송요구와 차량형태를 고려하여 얼마나 운영할 것인지를 수리모형으로 제시하고 휴리스틱 해법을 이용하여 해결하였다. APP(Aggregate Production Planning)와 VRP(Vehicle Routing Problem)을 이용한 Solution Algorithm은 Tabu Search을 이용하여 최적해 또는 근사해를 구하고자 하였다.

컨테이너 운송계획에서도 차량배차에 관한 연구들이 있다. 보통 컨테이너 수송목적의 다수차량과 다수지점, 그리고 적하지점 및 순서제약과 각종 시간제약들을 만족시키면서 효율적인 연계수송을 통한 공차운행의 최소화를 위한 배차시스템이 목적이다. 김동희 외 2명의 연구에서 우선 순위규칙(Priority Dispatching Rule)을 적용하고 적절한 시점에 사용자 개입을 허용하는 사용자 대화식 접근법(User-Interactive Approach)을 사용하였다(김동희 외2(1997)). 경로 및 일정계획의 수행도중에 흔히 발생하는 동적사건을 반영하기 위하여 동적배차의 개념을 적용하여 Feedback구조를 취하고 사용자가 차량-물량 배차결과를 쉽게 볼 수 있도록 간트차트 기법을 도입하였다.

컨테이너 운송에서 빈 컨테이너의 효율적인 운영도 관심의 대상이다. 비록 육상운송의 빈 컨테이너 관리는 아닐지라도 빈 컨테이너의 효율적 관리를 위한 시뮬레이션 모형을 구축하고 이를 현장에서 쉽게 활용할 수 있도록 재배치 및 보충규칙을 설정하는 구체적 절차를 제시한 연구가 있다(서순근과 박준호(2001)). 화주의 컨테이너 수요요구에 질 좋은 서비스를 제공하기 위해서는 효율적인 컨테이너 운용기술이 필요하다.

순수 컨테이너를 다루는 연구는 아니지만 접근방식이 비슷한 몇 가지 경우를 예를 들면 재고관리 모델링을 이용한 Philip과 Sussman(1977)의 연구가 있으며 그들은 단일 터미널의 지역에서의 빈 철도차량의 수요를 만족시키기 위한 재고수준 결정 문제를 다루었다(김갑환 외2(1998)). 여러 개의 터미널을 동시에 고려하여 빈 차량의 배분문제를 다룬 연구는 Misra(1972)에 의해서 수송문제로 모형화되어 제시되었다. 그러나 이 모형은 확정적인 정보를 가정하였

는데 반하여 Jordan과 Turnquist(1983)는 미래의 철도차량의 수요, 철도차량의 공급, 철도차량운행 시간에 있어서의 불확실성을 고려한 모형을 제시하였다. 컨테이너 운송문제에 관한 개략적인 분류는 <그림4>와 같다.



<그림4> 컨테이너 수송문제의 개략적인 분류도

2.3 연구의 문제환경 정의

2.3.1 컨테이너 운송시스템

각 장소에는 계획기간 내에 운반해야 할 화물컨테이너가 있다. 컨테이너의 물량과 목적지는 정해져 있다. 또한 각 장소는 빈 컨테이너 물류용기가 필요한 수요지 인지, 남은 여유분이 있어서 공급할 수 있는 공급지 인지 정해져 있다. 적입된 화물 컨테이너를 공급지에서 도착지로 운반 완료한 후에, 그리고 빈 컨테이너 물류용기를 공급지에서 수요지로 운반 완료한 후에 차량은 공차상태로 되기 때문에 각 장소에서 차량이 적재된 물건을 운반하여 도착한 횟수는 그 장소에서 가용한 공차의 수를 나타낸다. 따라서 공급량이 도착물량보다 많은 장소에는 공차가 다른 장소에서 와야 하고, 도착량이 공급량보다 많은 장소는 공차를 다른 장소에 공급해 주는 역할을 한다. 기타 본 연구에서 다루는 물류시스템 환경은 다음과 같다.

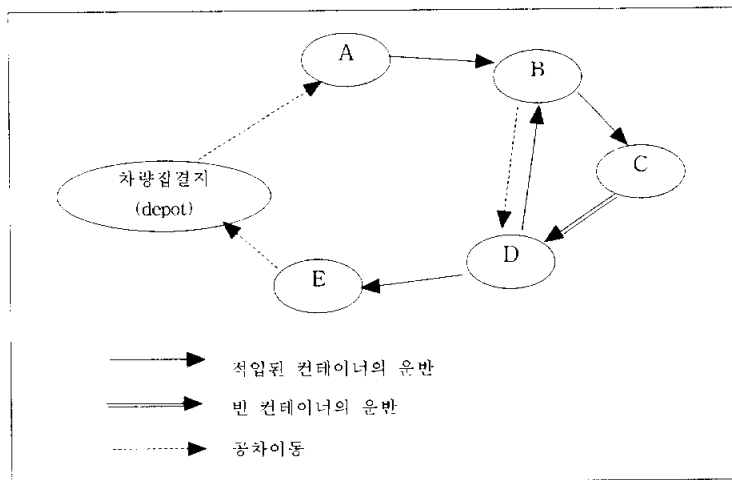
- ① 각 지점 사이에 운반되어야 할 적입된 컨테이너의 양은 알려져 있다.
- ② 각 지점에서 부족하거나 여유가 있는 빈 컨테이너의 양은 알려져 있다.
- ③ 각 지점사이의 이동시간은 알려져 있다.
- ④ 컨테이너는 계획기간 초에 준비되어져 있고, 도중에 수리가 없고 폐기도 없다.
- ⑤ 이동시 물량단위는 적입된 것과 빈 컨테이너의 구분 없이 차량 1대 분으로 한다.
- ⑥ 모든 차량의 수송용량은 1로 동일하고 차량은 여러 대이다.

⑦ 빈 컨테이너는 필요한 만큼 공급받을 수 있다.

(빈 컨테이너 총 공급량 $\geq$ 총 수요량)

⑧ 차량의 운반시간에는 순수 이동시간과 컨테이너 화물의 상차 및 하차 시간이 포함된다.

⑨ 모든 차량은 계획기간 초에 차량 집결지(depot)에 위치하고 있다.



<그림5> 물류용기 공동화 사용하에서의 컨테이너 운송계획

<그림5>은 본 연구에서 다루는 상황을 간단한 예로 설명하고 있다. 적입된 컨테이너의 운반은 $A \rightarrow B$, $B \rightarrow C$, $D \rightarrow B$, $D \rightarrow E$ 위치 간에 필요하고, 빈 컨테이너는 D에서 1개가 필요하고 모든 장소에 빈 컨테이너가 남아도는 경우의 차량 계획이다. 모든 적입된 컨테이너의 이동 물량과 공급지-도착지에 대한 데이터는 주어져 있으므로 차량의 이동시간은 공차의 이동시간에 영향을 받는다. <그림5>에서 장소 B를 보면 적입된 컨테이너를 싣고 들어오는 차량은 A와 D에서 2대인데 반하여 장소 B로부터 공급되는 적입된 컨테이너는 C로의 1대이므로, 장소 B에서는 여유있는 공차가 1대 있으며, 이 차량은 공

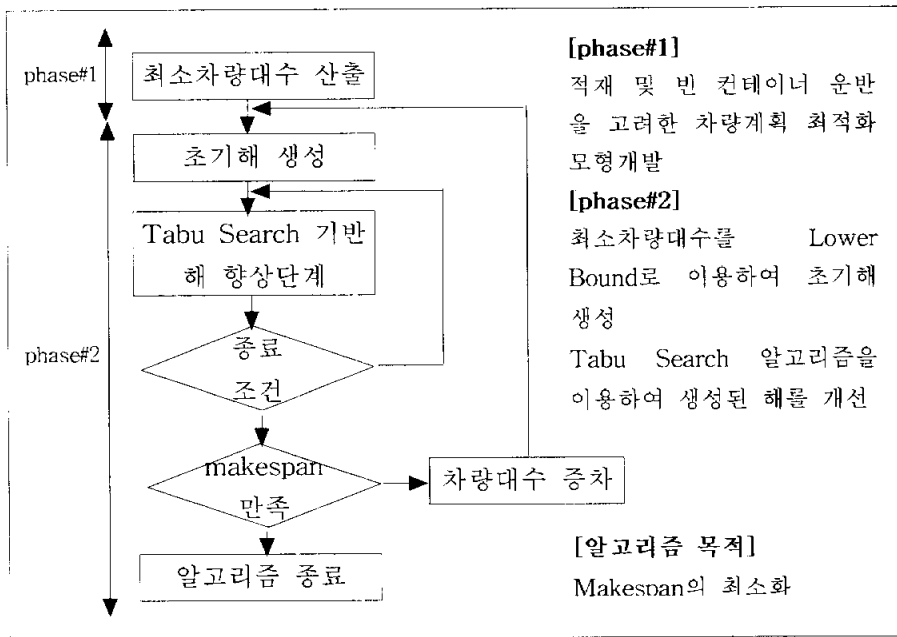
차를 필요로 하는 다른 장소로의 이동이 가능하다. 빈 컨테이너는 각 장소에서 공급량과 수요량은 갖고 있으나, 어디로부터 어디까지 운반하라는 조건은 없으므로, 빈 컨테이너의 운반시간과 이와 관련된 공차운행 시간 모두가 가변적이다. 따라서 빈 컨테이너의 운반은 공차운행을 최소화시키는 방안으로 활용 할 수 있다. 예를 들어 <그림5>에서 장소 C에는 하나의 적입된 컨테이너가 장소 B로부터 운반되지만 운반되어 나가야 할 화물 컨테이너는 없다. 따라서 다른 장소로 공차로 이동해야 하는데, 이때 장소 D에서 필요로 하는 빈 컨테이너를 실어 나른다면 차량 이동을 최소로 할 수 있다. 본 연구에서는 3.3절에서 이러한 상황을 고려하여 최소의 차량이동시간을 제공하는 차량계획에 대한 모형을 제시한다

제 3 장 화물 컨테이너 운송을 위한 탐색적 해법

3.1 연구의 탐색적 해법 설명

제시한 탐색적 해법의 개략적인 내용을 살펴보면 최적의 차량경로를 도출하기 위해서 주어진 물량데이터를 이용하여 공차운행을 최소로 하는 수리모형을 모델링하고 CPLEX패키지를 이용하여 최소필요차량대수를 구한다. 이렇게 구한 최소필요차량대수 정보를 기반으로 초기 차량경로를 구성하는 초기해 생성 알고리즘을 적용하고, 다음으로 생성된 초기해를 향상시키기 위해서 해 향상 알고리즘을 적용한다.

본 연구에서 제시한 Tabu Search 알고리즘은 주어진 차량대수를 이용하여 최대완료시간(makespan)을 구하는 것을 기본으로 한다. 그러나 만약 makespan이 주어진 경우라면 최소필요차량대수에서 알고리즘을 시작하여 makespan을 만족하기 위해 차량을 한 대씩 증가시키는 방법도 적용 가능하도록 구성되어 있다. 이때 차량대수는 최소필요차량대수가 아니라 사용자가 임의로 입력할 수가 있다. 알고리즘의 기본 목적은 makespan의 최소화에 있다. <그림6>은 Tabu Search 알고리즘 전체의 개략도이다.



<그림6> Tabu Search알고리즘 전체 개략도

3.2 Tabu Search 이론

현실에 있어서 문제의 해결방법은 조합최적화 문제로 해석할 수 있다. 그러나 이러한 경우의 단점은 그 조합이 너무 많은 경우 원하는 시간내에 풀 수가 없다는 것이다. 많은 연구자들은 이를 해결하기 위해 많은 영역에서 아이디어를 얻어왔다. 네트워크 흐름방법은 전기나 수력으로부터 아이디어를 얻었고, 시뮬레이티드 어닐링 방법은 금속의 표면 처리 과정의 냉각과정에 기초를 두었고, 유전알고리즘은 자연의 진화과정과 유전법칙을 모방했다. 본 연구에서 사용한 Tabu Search는 인간의 기억과정을 이용하였다. Tabu Search방법은 인공지능 분야에서 선택된 개념에 기초를 둔 기법이다.

Tabu Search는 복잡한 해 영역에서 좋은 해를 얻기 위해 탐색을 유도하는 일반적인 휴리스틱 과정이다. 또한 Tabu Search는 다른 휴리스틱 과정에서 사용할 수 있고 다른 휴리스틱 과정을 제어할 수 있는 메타휴리스틱이다. Tabu Search의 주요 요소 중 하나는 유연한 메모리 사용이다. 그것이 탐색과정에서 핵심적인 역할을 한다. 이러한 메모리 탐색방법을 더 세련되게 적용하면 특별한 문제를 효과적으로 풀 수 있다.

Tabu Search를 적용한 문제들을 부분적으로 살펴보면 다음과 같다. 이것은 Tabu Search가 여러 문제에 적용가능함을 알 수 있게 해준다.

작업자 스케줄링	기계 스케줄링
maximum satisfiability problems	형상인식
공간계획 및 건축설계	전화통신 경로배정
확률로직문제	job shop scheduling
neural network pattern recognition	computer channel balancing
convey scheduling	quadratic assignment problem
network topology problem	graph coloring
graph partition	nonlinear covering
maximum stable set problems	flow shop sequencing

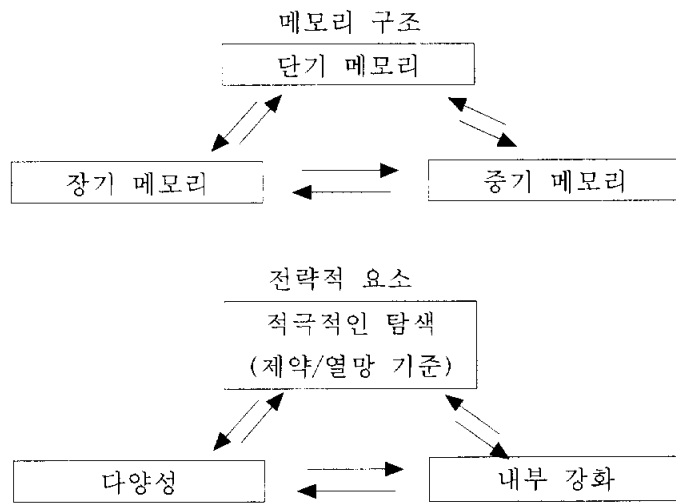
Tabu Search는 개선하는 움직임은 한쪽방향으로만 고정하지 않고 이미 지나왔던 지역 최적해에 다시 빠지는 일 없이 탐색을 계속하게 휴리스틱을 인도한다. Tabu Search는 일반적으로 T라고 일컬어지는 최대반복횟수를 사용함으로써 ‘제약적인 탐색’을 한다. 또한

3가지 기본적인 특징을 가지고 있다(김여근 외2(1997)).

- (1) 유연한 성질이 있는 컴퓨터 기억구조를 사용한다. 이는 고정된 기억구조방법(Branch and Bound)이나, 기억을 사용하지 않는 방법(Simulated Annealing, 임의로 찾는 방법)보다 더 자세하게 역사적 탐색정보와 기준개선을 이용하여 탐색한다.
- (2) 통제에 관련된 기술로 메모리구조에 관련한 메모리구조를 사용하여 통제조건과 열망기준(Aspiration Criteria)사이에서 자유롭게 탐색한다.
- (3) 강화(Intensifying)전략은 역사적으로 발견된 좋은 해의 장점을 이용하여 움직이게 하고, 다양화(Diversification, 다양성)는 탐색하지 않은 새로운 영역으로 다양하게 찾아가게 하는 것이다. 강화와 다양성의 탐색전략을 적용하기 위해서 메모리함수를 통합한다.

<그림7>은 메모리와 전략의 관계를 나타낸 것이다. 메모리를 세부분으로 나눈 것은 인위적인 것으로 특별히 중기 메모리와 강화, 장기 메모리와 다양성 전략간의 연결은 절대적이기 보다는 가변적이고 편리를 위해서 나눈 것이다.

Tabu Search 알고리즘을 설명하기 위해서는 Tabu List에 관한 설명을 하지 않을 수 없다. 이것은 최근에 방문한 해로 돌아가는 움직임을 방지하거나 벌금을 주는 메모리 구조를 이용함으로써 함정에 빠지지 않게 한다. Tabu 제약식의 기본적인 목표는 각 움직임이 매 단계별로 높은 개선을 하면서 지역해를 벗어나게 하는 것이다. 이러한 제약식이 필요한 것은 한 번 도달한 지역해에서 다음 움직임으로 선택된 제일 좋은 움직임은 다시 그 지역해로 돌아가기 때문이다. 혹은 개선이 없이 다른 점으로 이동했어도 다시 그 지역해



<그림7> 메모리와 전략의 관계(김여근 외2(1997))

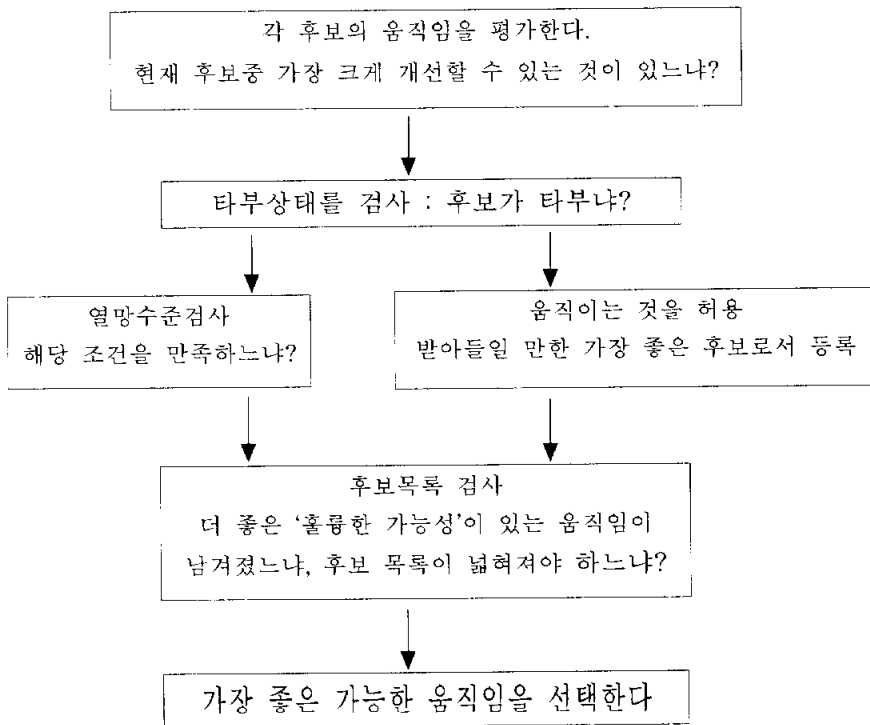
로 돌아간다. 일반적으로 Tabu 제약은 이러한 순환행동(cycle)을 방지하는 것이 목표이다.

다음으로 개선하기 위한 방법에서 최선의 후보를 선정하는 ‘가장 좋은 가능한’ 움직임을 찾는 것에 대해 설명한다. 이것은 현재해를 개선할 수도 있고, 하지 않을 수도 있다. 최적해나 혹은 최적해에 가까운 해로 개선하는데 가장 가능성이 높은 것으로 움직인다는 가정에 기초한다. Tabu Search 입장에서 이 가정은 개선기준이 된다. <그림8>은 가장 좋은 후보를 선택하는 중요단계를 상세하게 나타낸 것이다.

Tabu Search는 궁극적으로 좋은 해를 얻기 위한 Tabu 제약 조건이 도리어 좋은 해가 있을 수 있는 곳으로 움직이는 것을 방해할 수 있다. 이러한 한계를 극복하기 위하여 어떤 때는 Tabu 조건을 무시할 수 있는 새로운 조건이 필요할 때가 있다. 이러한 조건을 열망기준이라고 한다. 본 연구에서는 Tabu Search 알고리즘을 적용함

에 있어서 그 열망기준은 사용하지 않았다.

<그림9>은 기본 Tabu Search 기법을 정리한다. 초기 가능해를 생성하여 이를 현재해로 하고, 이로부터 이웃해 집합을 생성하여 타부이동이 아니거나 타부이동이지만 열망수준을 만족하는 이웃해에서 가장 좋은 해, 즉 최선해를 현재해로 하고, 이 이동속성을 타부 목록에 일정기간 저장하면서 이 과정을 종료조건이 만족될 때까지 반복하는 순서를 따른다.



<그림8> 가장 좋은 후보선택(김여근 외2(1997))

[단계 1] 초기화

- 1) 타부속성, 타부목록, 크기, 열망수준과 종료조건 등을 결정한다.
- 2) 초기 가능해를 구하여 현재해와 최선해로 둔다.
- 3) 타부목록을 비어둔다.

[단계 2] 이웃해 생성

현재해로부터 타부이동이 아니거나 타부이동이지만 열망수준을 만족하는 이동에 대해 이웃해를 생성한다.

[단계 3] 현재해와 최선해 수정

- 1) 생성된 이웃해 중에서 가장 좋은 해를 현재해로 한다.
- 2) 현재해가 최선해보다 좋으면 현재해를 최선해로 둔다.

[단계 4] 타부목록 수정

새로운 현재해의 이동속성을 타부목록에 기록한다. 그리고 만약 타부목록에 저장된 타부속성의 수가 타부목록크기보다 크면 가장 먼저 기록된 타부속성을 삭제한다.

[단계 5] 종료기준 검사

만약 종료기준을 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

<그림9> Tabu Search기법 기본 (김여근 외2(1997))

3.3 공차운행 최소화 수리모형

2.3절에서 설명한 컨테이너 운송시스템의 상황에 따라 차량의 이동을 최소화 하는 수리모형은 다음과 같다.

Notation

[Decision Variables]

x_{ij} 차량이 공차로 장소 i 에서 장소 j 로 이동하는 횟수

y_{ij} 차량이 빈 컨테이너를 싣고 장소 i 에서 장소 j 로 이동하는 횟수

[Parameters]

v_{ij} 적입된 컨테이너를 싣고 장소 i 에서 장소 j 로 이동해야 하는 물량

f_i 계획기간 초에 장소 i 에 있는 공차의 수

g_i 계획기간 말에 장소 i 에 있어야 하는 공차 수

d_i 장소 i 에서 필요로 하는 빈 컨테이너의 수

s_i 장소 i 에서 공급해 줄 수 있는 빈 컨테이너의 수

t_{ij}^a 장소 i 에서 장소 j 로 적입된 컨테이너를 싣고 이동하는데 소요되는 시간

t_{ij}^b 장소 i 에서 장소 j 로 빈 컨테이너를 싣고 이동하는데 소요되는 시간

t_{ij}^c 장소 i 에서 장소 j 로 공차로 이동하는데 소요되는 시간

적입된 컨테이너 운반 상황에서 빈 컨테이너 운반을 고려한 차량의 총 운반시간은 아래와 같이 표현된다.

$$TR = \sum_i \sum_j t_{ij}^a v_{ij} + \sum_i \sum_j t_{ij}^b y_{ij} + \sum_i \sum_j t_{ij}^c x_{ij}$$

계산의 편의를 위하여 빈 컨테이너의 수요와 공급물량의 합을 동일하게 하기 위하여 가상적인 장소 r 을 도입한다. 가상장소 r 에서의 빈 컨테이너 수요는 $d_r = \sum_i s_i - \sum_i d_i$ 에 의해서 구해진다. 이때 가상적으로 빈 컨테이너를 한 장소에서 가상장소 r 로 이동시키는 작업이 있고 동일한 횟수만큼 공차를 가상장소 r 과 빈 컨테이너를 보낸 장소로 이동하는 것으로 가정한다. 모든 장소와 가상장소 r 간에는 이동시간을 모두 0으로 한다. 그러면, 가상적인 빈 컨테이너 운반은 차량의 이동시간과는 무관하게 된다. 장소 i 에서의 차량 순흐름 $nf(i) = \sum_j v_{ji} - \sum_k v_{ik} + f_i - g_i$ 와 공차이동과의 관계는 빈 컨테이너 운반에 영향을 받는다. 즉 빈 컨테이너의 수요처에서는 컨테이너를 배달 받은 후 유휴차량이 추가로 발생하며 이때 차량 순흐름은 증가하고, 반대로 공급처에서는 더 많은 공차가 필요하다. 빈 컨테이너의 운반을 고려한 차량계획의 목적은 차량의 총 이동시간을 최소로 하는 것이고, 이는 빈 컨테이너 운반과 공차운행 시간의 합을 최소화하는 것을 의미한다. 이때 수리모형은 아래와 같이 표현된다.

$$\text{Min } z = \sum_i \sum_j t_{ij}^a v_{ij} + \sum_i \sum_j t_{ij}^b y_{ij} + \sum_i \sum_j t_{ij}^c x_{ij}$$

Subject to :

- (1) $\sum_k x_{ik} = nf(i) + \sum_j y_{ji} - \sum_k y_{ik}$ $nf(i) + d_i - s_i \geq 0$ 인 모든 i 에 대하여
- (2) $\sum_j x_{ji} = -nf(i) - \sum_j y_{ji} + \sum_k y_{ik}$ $nf(i) + d_i - s_i < 0$ 인 모든 i 에 대하여
- (3) $\sum_k y_{ik} = s_i$ $s_i > 0$ 인 모든 i 에 대해
- (4) $\sum_j y_{ji} = d_i$ $d_i > 0$ 인 모든 i 에 대해

(5) $x_{ij}, y_{ij} \geq 0$ and 정수, for all i, j

식(1)은 빈 컨테이너의 수요처인 경우에 해당 장소에서 공차로 나가야 할 차량의 수는 순흐름과 빈 컨테이너를 배달한 차량 수의 합이어야 한다는 것을 보여준다. 빈 컨테이너를 배달한 차량 수를 더한 것은 빈 컨테이너를 배달한 차량도 해당장소에서 공차인 상태로 되기 때문이다. 식(2)에서는 순흐름과 빈 컨테이너의 수요량의 합이 음수인 경우 그 음수의 절대치만큼 해당 장소로 공차가 들어와야 하는 것을 나타낸다. 식(3)에서는 빈 컨테이너의 공급지는 공급량만큼 해당 장소로부터 빈 컨테이너를 싣고 나가는 차량의 수가 정의되며, 식(4)에서 빈 컨테이너의 수요처는 공급량만큼 해당 장소로 빈 컨테이너를 싣고 들어오는 차량의 수가 결정된다. 식(5)은 모든 결정변수는 비음의 정수 값을 가짐을 나타낸다.

상기의 수리모형은 차량운행시간이 최소화되도록 각 지점간의 차량운행횟수를 결정한다. 즉, 수리모형의 결과는 공차의 이동이 어디에서 어디로 얼마만큼의 양이 이동되어야 할지, 빈 컨테이너의 이동이 어디에서 어디로 얼마만큼의 양이 이동되어야 할지를 도출해준다. 그러나 여기서 얻은 해는 차량경로를 무시한 이상적인 상황에서의 결과로서 실제로는 달성하기 어려운 결과이다. 따라서 본 연구에서는 상기의 수리모형을 기반으로 최적의 차량경로를 수립한다.

3.4 Tabu Search 기반의 차량경로계획 알고리즘

3.3절에서 보여지는 수리모형 결과에 따라 차량의 총 운행시간을

하루운행 가능한 차량작업시간으로 나누면 운송작업을 완수하기에 필요한 최소차량대수를 구할 수가 있고 도출된 정보를 기초자료로 이용하여 차량경로를 계획한다. 차량경로를 계획하는 문제는 규모가 작은 경우에는 가능한 모든 경우를 조합하여 최적해를 구할 수가 있으나 문제의 규모가 커질수록 시간과 노력이 기하급수적으로 증가함에 따라 탐색적인 해법이 주로 연구되어 왔다(고창성 외2(2000), 김갑환 외2(1998)). 본 논문에서는 Tabu Search를 이용하여 컨테이너 운송 시스템에서의 차량경로계획을 수립한다. 새로운 알고리즘은 초기해 생성단계와 해 향상 단계로 이루어진다.

초기해 생성 단계는 근시안적인 방법을 이용하여 빠르고도 모든 차량이 비슷한 작업완료시간을 가지며 작업을 완료할 수 있도록 생성한다. 모든 작업 차량 중 현재 가장 빨리 작업이 완료된 차량에 공차이동시간이 가장 적은 작업을 할당한다. 작업 중에 동물이 발생하게 되면 makespan이 최소화 되도록 하기 위해서 작업시간이 큰 작업물량을 선정하여 할당한다.

생성된 초기해를 향상시키는 두 번째 단계에서는 먼저 최대의 완료시간을 가지는 차량을 선정한다. 선정된 차량내의 현재 계획된 작업순서를 조정해 봄으로써 완료시간을 줄여보도록 한다. 현재의 작업순서 내에서 더 이상 완료시간이 줄어들지 않는 경우는 가장 적절한 작업물량을 최소의 완료시간을 가지는 다른 차량으로 이동시킴으로써 전체 차량의 최대완료시간을 줄인다. 이와 같은 기본 개념을 이용한 Tabu Search 알고리즘의 단계별 세부내용은 다음과 같다.

☞ 초기해 생성 단계

[Step 0]

이용가능한 차량의 대수, 지점간 운반물량, 지점간의 시간 데이터를 수집한다.

[Step 1]

모든 작업물량 집합 Set_J 를 구성한다. 모든 차량의 집합 Set_V 를 구성한다. 차량의 완료시간은 depot를 출발해서 운반작업을 완료하고 depot로 복귀하는 시간까지 모두 포함한다. 각각의 차량의 완료시간은 0으로 초기화한다. 차량의 경로는 차량이 운반해야 할 작업물량의 순서이다. 모든 차량의 각 경로 역시 할당된 작업이 없는 상태로 구성한다.

[Step 2]

차량목록 Set_V 에 구성된 차량의 대수와 동일한 수의 작업물량을 랜덤하게 선택하여 각각의 차량에 첫 작업물량을 할당한다. 또는 차량목록 Set_V 에 구성된 각각의 차량에 차례로 근거리 작업물량을 할당한다. 선택된 작업물량들은 작업물량 집합인 Set_J 에서 삭제한다. 차량의 완료시간을 수정한다.

[Step 3]

작업이 할당된 모든 차량을 대상으로 최소완료시간을 가진 차량을 선정한다. 차량경로내의 마지막 작업과 가장 공차거리가 짧은 작업을 작업물량 집합에서 선정하여 차량의 마지막 작업 바로 뒤에 할당한다. 선택된 작업은 작업물량 집합에서 삭제하고, 차량의 완료시간 역시 수정한다. 동물이 발생할 경우는 작업시간이 큰 작업이 우선순위를 갖는다.

[Step 4]

모든 작업물량이 배정될 때까지 Step3을 반복한다.

초기해 생성단계를 수행하면 각 차량별 경로와 완료시간의 정보를 가진 초기해가 구성된다.

해 향상단계

[Step 0] 초기화

초기해를 Incumbent Solution으로 저장한다. 현재 총 반복횟수는 0로 초기화한다.

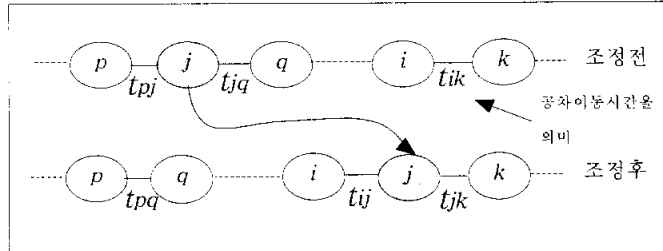
[Step 1] Busiest_V선정

최대 완료시간을 가지는 차량을 Busiest_V로 선정한다. 동물이 발생시에 랜덤하게 선택한다. 현재 반복횟수의 최대완료시간이 Incumbent Solution보다 작으면 Incumbent Solution을 교체 수정한다.

[Step 2] Busiest_V조정

선정된 Busiest_V차량내의 모든 작업에 대해서 작업순서를 변경시켜 완료시간을 줄이도록 한다. 작업 J_j 의 앞작업은 J_p , 뒷작업은 J_q 라고 하고, 작업 J_j 가 새로 삽입될 지점의 앞 작업은 J_i , 뒷작업은 J_k 할때 공차운행의 최소증가가 이루어지는 위치 $Min[(t_{ij}+t_{jk}+t_{pq})-(t_{ik}+t_{pj}+t_{jq})]$ 에 작업 J_j 는 재삽입된다. 동물이 발생시는 랜덤하게 선정되어 삽입된다. 수정된 경로에 대한 Busiest_V차량의 완료시간이 계산된다. <그림10>는 재삽입 되

는 과정을 나타낸다.



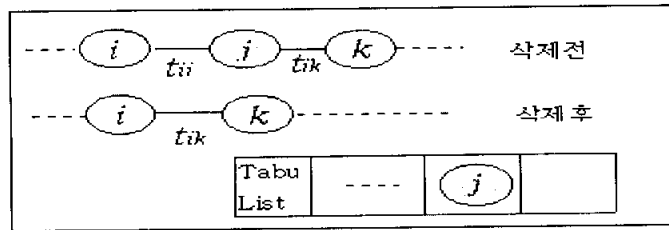
<그림10> Busiest_V차량내 작업의 재삽입

[Step 3] Busiest_V경로 Update

Step2의 결과로 인한 Busiest_V차량의 완료시간이 줄어들어 더 이상의 Busiest_V차량이 아닌 경우는 차량경로를 조정된 후의 순서로 업데이트 하고 Step1로 간다. 그 외의 경우는 반복횟수를 1회 증가시키고 Step4로 간다. 반복횟수가 T보다 크면 Step6으로 간다.

[Step 4] Busiest_V차량내의 삭제 작업 선정

Busiest_V차량내의 모든 작업에 대해서 다음 방법을 적용한다. 작업 J_j 의 앞작업을 J_i , 뒤작업을 J_k 라고 하자. $Max[(t_{ij} + t_{jk} - t_{ik})]$ 를 가지는 작업 J_j 를 선택한다. 선택된 작업은 Busiest_V차량내에서 삭제되고 Tabu List에 올라간다. 선정된 Busiest_V차량내의 모든 작업이 Tabu List에 있는 경우는 Feasible Neighbor가 없는 경우에 해당되어 Step6으로 간다. Feasible Neighbor란 Busiest_V차량내에서 선택할 수 있는 작업의 집합을 의미한다. <그림11>은 삭제작업을 선정하는 과정을 나타낸다.



<그림11> Busiest_V차량내의 작업삭제

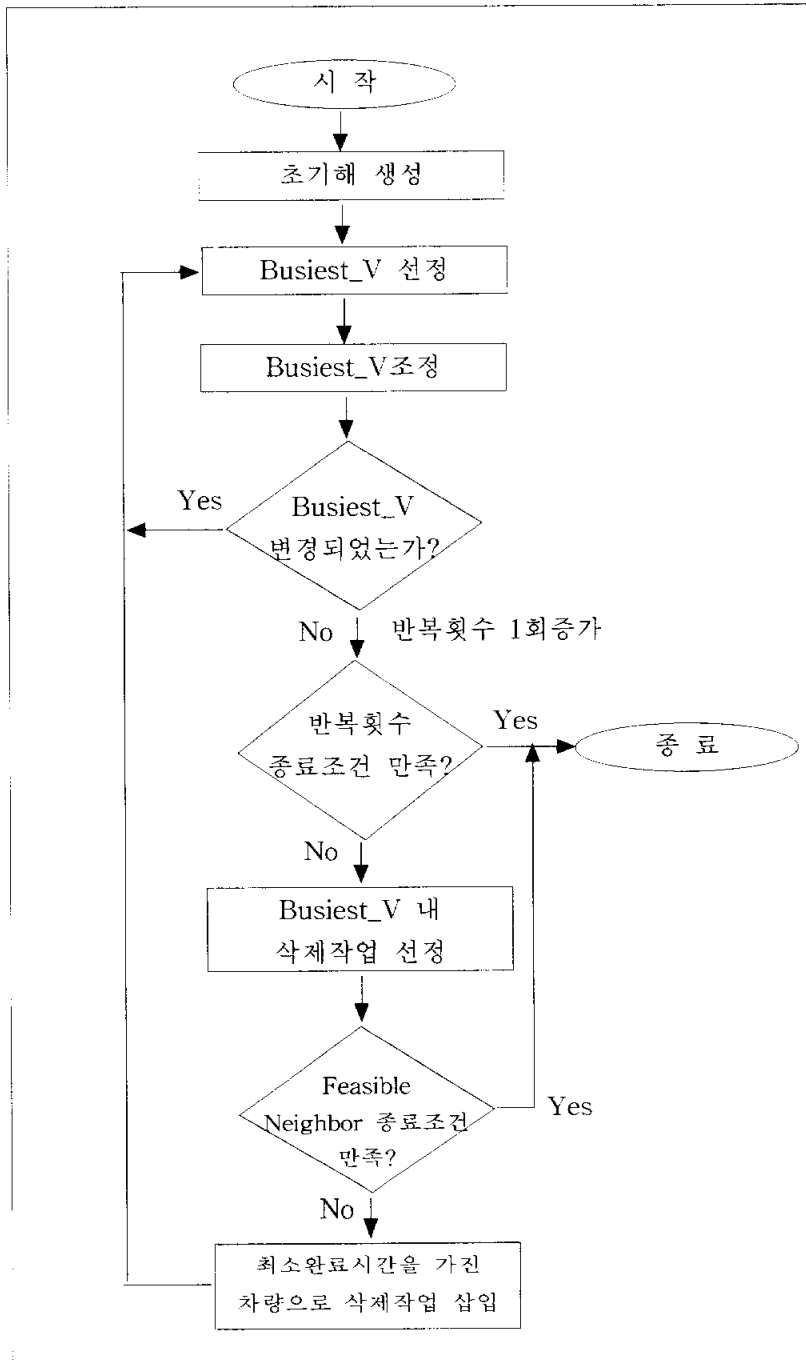
[Step 5] 최소 완료시간의 차량에 삽입

Busiest_V차량에서 삭제된 작업은 차량목록 중 최소완료시간을 가지는 다른 차량에 삽입되어진다. 삽입될 작업을 J_j 라고 하고, 삽입될 지점의 앞작업은 J_i , 뒷작업은 J_k 라고 할때 $Min[(t_{ij} + t_{jk} - t_{ik})]$ 인 지점에 삽입한다. 충돌이 발생시는 랜덤하게 선택한다. Step1로 간다.

[Step 6] 종료

주어진 총 반복횟수를 만족하는 경우 또는 Feasible Neighbor가 없는 경우에 종료된다.

초기해 향상단계를 수행하면 주어진 반복횟수만큼 각 반복구간마다 차량의 최대 완료시간이 구해진다. 각 반복구간마다 산출된 최대 완료시간 중 최소의 완료시간을 가지는 반복구간에서의 차량경로와 그때의 최대 완료시간이 Optimal Solution이 된다. <그림12>은 전체적인 알고리즘의 흐름도를 나타낸다.



<그림12> 알고리즘 흐름도

제 4 장 수치예제 및 성능분석

4.1 수치예제

본 연구의 4장에서는 제시한 새로운 알고리즘을 예제에 적용하여 실제로 현장에서 사용하고 있는 방법과 기존의 연구에서 발표된 방법과 비교 평가한다. 적입된 컨테이너의 이동물량과 빈 컨테이너의 여유와 부족의 양이 <표2>과 같이 주어져 있다. 이동시간은 <표3>과 같이 주어져 있다. A부터 L까지는 각 지점을 의미하게 되고 각 지점사이의 이동물량 정보와 계획기간 초와 계획기간 말 차량집결지(depot)에 있어야 할 차량 수, 빈 컨테이너의 여유와 부족의 정보, 차량 순흐름등의 정보를 보여준다. 본 기본예제에서는 depot와의 시간은 0으로 하고 있다. 이는 하루업무 시작 시는 미리 상차지점으로 이동하여 대기하고 있음을 의미하고, 하루 업무 종료 후에는 depot

<표2> 적입된 컨테이너 이동 물량과 빈 컨테이너 여유/부족 물량 정보 유입유출표

To From	A 1	B 2	C 3	D 4	E 5	F 6	G 7	H 8	I 9	J 10	K 11	L 12	출 출 발	초 기 빈 차	말 기 빈 차	수 요 / 공 급	NF(i) NF(i)- (수요/공급)	총필요공차 (수요/공급)
A 1					15			47	2				64			-10	-40	-30
B 2					28			5	2				35			-30	-35	-5
C 3					22								22			10	-10	-20
D 4									1				1			0	-1	-1
E 5	3		10										13			10	52	42
F 6													0			-5	0	5
G 7													0			-5	0	5
H 8	21		2										23			40	33	-7
I 9								4					4			0	1	1
J 10														15			15	15
K 11															15		-15	-15
L 12																-10	0	10
총도착	24	0	12	0	65	0	0	56	5	0	0	0						

<표3> 지점간 이동시간

장소	A 1	B 2	C 3	D 4	E 5	F 6	G 7	H 8	I 9	J 10	K 11	L 12
A 1	-	50	30	35	40	35	30	30	30	0	0	0
B 2		-	30	35	40	35	30	35	35	0	0	0
C 3			-	5	10	25	30	35	35	0	0	0
D 4				-	5	20	25	30	30	0	0	0
E 5					-	15	20	25	25	0	0	0
F 6						-	10	15	15	0	0	0
G 7							-	5	5	0	0	0
H 8								-	5	0	0	0
I 9									-	0	0	0
J 10										-	0	0
K11											-	0
L 12												-

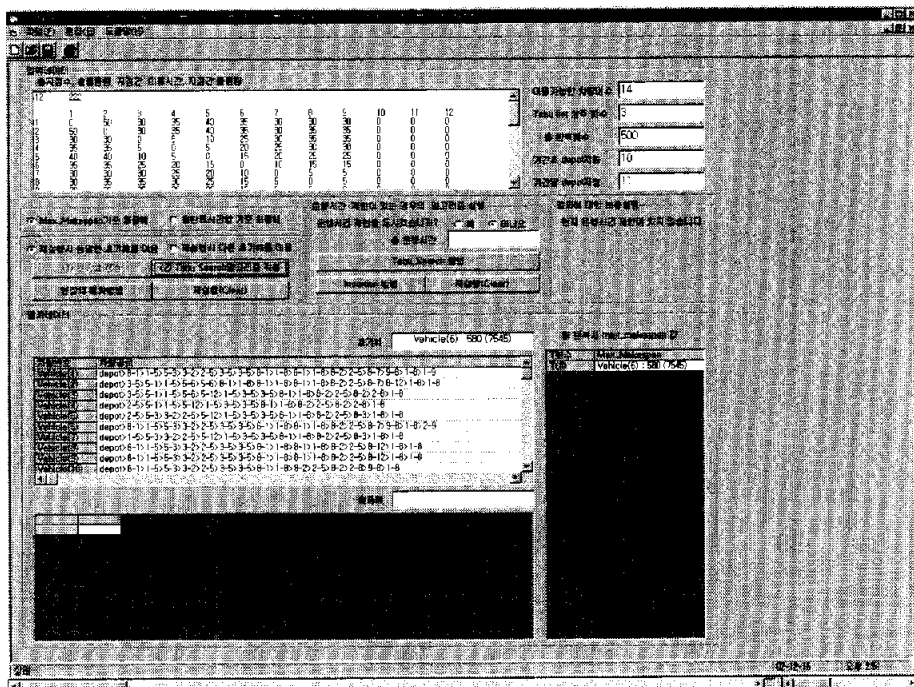
로 돌아가는 시간은 포함하지 않는 것을 의미한다. 알고리즘상 차량 운행시간에는 depot에서 출발과 depot로의 복귀시간도 모두 포함하게 되어있으므로 기본예제 외의 다른 상황에서 depot와의 거리가 있는 경우도 알고리즘은 동일하게 적용된다.

3.3절에서 제시한 수리모형을 최적화 해법도구인 CPLEX패키지를 이용하여 도출한 결과, 총 차량 이동시간이 6655분으로 차량이 운행 가능한 시간을 하루 8시간(480분)으로 볼 때 약 13.86대로 계획기간(하루)동안 필요한 최소차량대수는 14대임을 알 수가 있다. 이는 차량대수의 하한값(Lower Bound)을 의미한다. 본 연구에서 수치계산은 차량대수와 이동물량수를 계산할 때를 제외하고 모두 사사오입한다.

프로그램에 대해서 설명을 하면, Text형태의 데이터를 입력받도록 구성되어있다. 이용 가능한 차량대수, Tabu List 상주횟수, 총 반복

횟수, 기간초/말 depot지점 등의 파라미터 정보를 입력받을 수 있는 란이 구성되어있다. 이러한 파라미터는 실험 방법에 따라 입력해야 하는 경우도 있고, 그렇지 않은 경우도 있다.

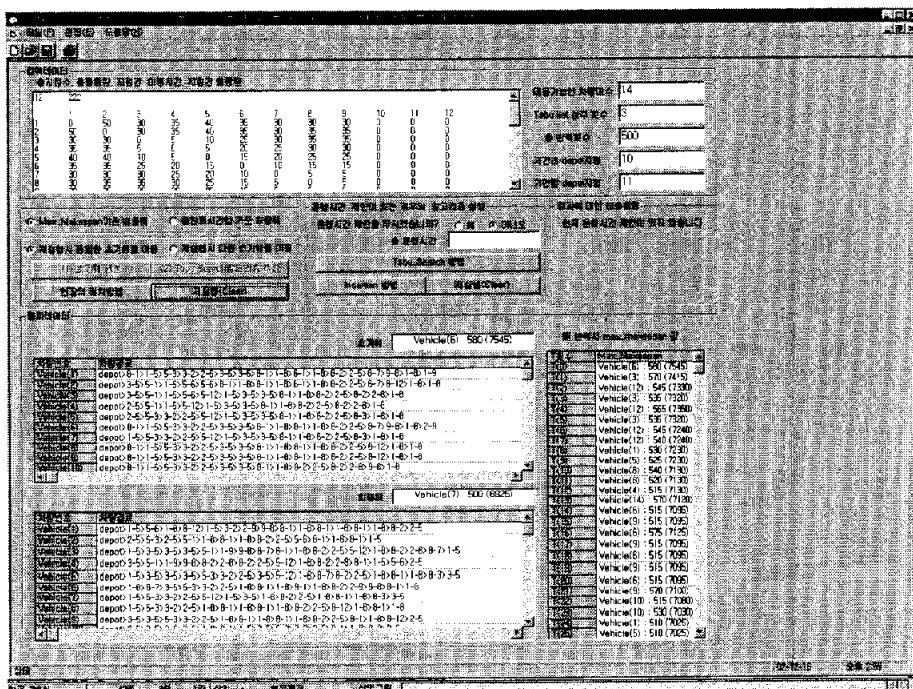
14대의 차량대수를 이용하여 초기해를 생성한다. 초기해를 생성할 경우는 차량대수 정보, 기간초/말 depot지점 정보만 있으면 가능하고, Tabu Search 알고리즘을 이용한 해 향상을 하기 위해선 Tabu List 상주횟수와 총 반복횟수를 추가로 입력해야 한다. 기본예제의 실험은 주어진 차량대수를 이용하여 makespan을 구하는 것으로 한다. 최종해를 도출시에 각 반복횟수별 makespan을 비교대상으로 한다. <그림13>는 생성된 초기해를 보여준다. 초기해의 makespan은 580분으로 총 운행시간은 7545분임을 알 수 있다.



<그림13> 초기해 생성

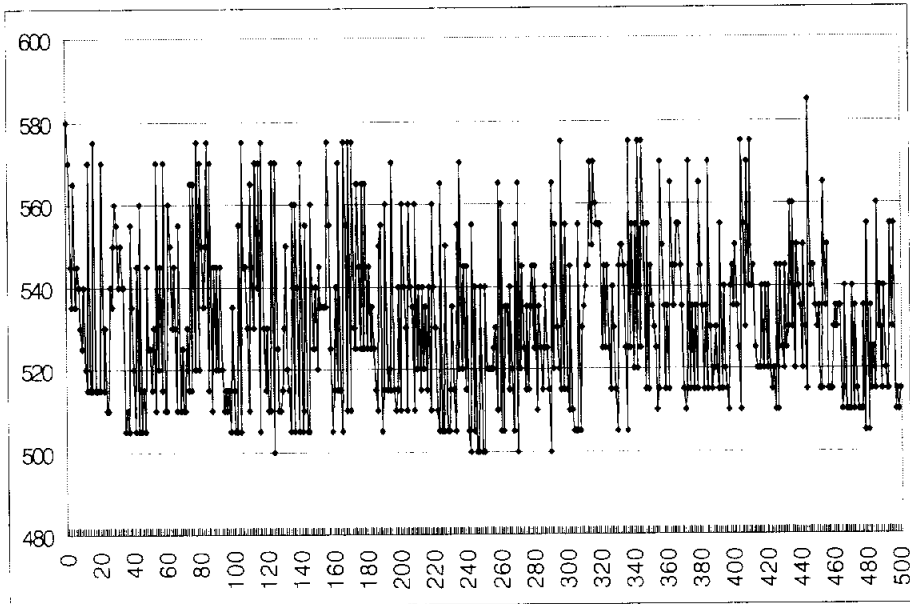
옵션사항으로, 비교대상을 완료시간의 합을 기준으로 할 수도 있다. 그리고 Tabu Search 알고리즘 적용시 재실행을 할 경우, 동일한 초기해를 이용하거나, 다른 초기해를 새로 생성할 수 있도록 구성하였다. Insertion방법이나, Tabu Search 알고리즘에 시간제약을 두는 경우, 차량대수가 어떻게 변화 되는지에 대한 정보를 제공하는 결과보충설명란도 구성되어있다. 초기해와 최종해가 어떻게 변화되는지를 알 수 있도록 Grid를 이용하여 나누어 놓았고, 각 반복구간별 결과값을 알 수 있도록 매 구간 makespan과 완료시간의 합을 나타내었다.

생성된 초기해를 Tabu Search를 이용한 해 향상 알고리즘을 적용한다. 여기서 Tabu List 상주횟수는 3회, 총 반복횟수는 500으로 한다. 상주횟수는 여러 번의 실험결과를 통해 얻었다.



<그림14> 생성된 최종해

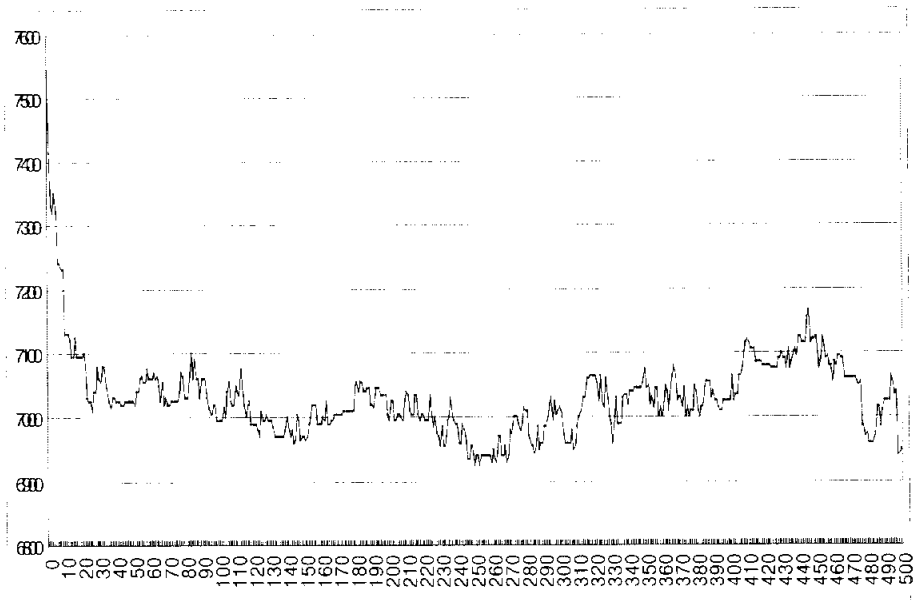
<그림14>는 도출된 최종해를 보여준다. 최종해의 makespan은 500분, 완료시간의 합은 6925분임을 알 수 있다. 각 반복횟수마다 생성된 makespan과 완료시간의 합 결과는 <그림15><그림16>을 통해서 지역해에 빠지지 않고 최종해를 얻는 과정을 볼 수가 있다.



<그림15> 각 반복횟수별 makespan의 변화

기본예제의 Insertion방법 적용실험 화면과, Tabu Search 알고리즘에 시간 제약을 두고 실험한 경우의 예는 [부록2]를 참고하기 바란다.

기본예제에서 이용한 현장의 데이터는 각 지점과 depot사이와의 이동시간을 0으로 가정하고 있다. 사용된 예제는 이동시간이 0임으로 인해서 초기해를 생성할 경우, 첫물량을 할당할시에 랜덤하게 선택이 되거나 또는 근거리의 물량이 할당되거나 그다지 큰 의미를



<그림16> 각 반복횟수별 완료시간 합의 변화

가지지 못한다. 따라서 최종해를 도출시에 최적해에 보다 근접할 수 있는 방법의 선택은 성능분석의 절에서 보다 자세히 다루도록 한다.

4.2 성능분석

4.2.1 기존의 방법과 비교분석

컨테이너 물류운송부분의 차량경로 연구에서 논문에서 제시한 환경과 동일한 조건에서 연구되어진 논문은 많지 않다. 따라서 이와 가장 유사한 환경의 컨테이너 물류운송에 관한 해법과 실제로 현장에서 적용되는 방법을 이용하여 비교해본다. 비교 실험하게 될 해법

인 Insertion 방법은 계획기간 동안 차량운행시간이 주어지고 그에 따라 용량과 시간제약을 고려한 실행가능성 여부를 통과하고 수송요구 작업을 할당받은 상태에서 추가적으로 수송요구 작업을 받을 때 총 운행시간 증가분이 적은 위치에 작업이 삽입되는 방법을 따른다(고창성 외5(2002)). 즉, 용량과 시간조건을 만족한 작업물중에서 적차 운행시간과 공차운행시간을 모두 고려하여 총 운행시간 증가분이 가장 적은 작업을 선택하는 것이다. 두 번째 비교 실험하게 될 실제 현장에서의 방법은 가장 가까운 각 상차지점으로 이동을 하여 작업물을 하차한 후에 상차할 작업물이 있으면 작업물을 운송하고 하차한 지점에 상차할 작업물이 없으면 가장 가까운 상차지점으로 이동하는 방법을 따른다. 다시 말하면, 근거리 물량을 먼저 수송하는 것을 의미한다.

4.1절의 수치예제에서 언급을 하였지만, Tabu Search 알고리즘 적용시에 필요한 Tabu List 상주횟수는 여러 번의 실험결과 상주횟수는 3회에서 좋은 해를 산출하였다. 이에 상주횟수는 3회, 총 반복횟수는 500회로 실험을 진행하였다. 또한 수치예제의 depot와의 거리가 0이라는 가정상, 초기해 생성방법은 랜덤하게 첫물량을 할당하는 방법을 사용하기로 한다. Insertion 방법 적용시에 필요한 차량운행시간은 480분으로 한다. <표4>의 결과를 보면 Insertion 방법은 차량운행시간이 480분일 때 16대의 차량으로 모든 작업을 완료한다. 따라서 16대의 차량으로 Tabu Search 알고리즘을 적용한 결과, Tabu Search 알고리즘이 Insertion방법에 비하여 최대완료시간(makespan)은 9.38%, 완료시간의 합은 6.38% 해가 향상된 것을 알 수 있다. 현장에서의 방법보다는 최대완료시간은 15.53%, 완료시간의 합은 8.43%로 해가 향상되었다.

<표4> Insertion방법 차량대수 적용시

	차량대수	makespan	완료시간 합
Tabu Search	16	435	6900
Insertion	16	480	7370
현장의 방법	16	515	7535

수리모형으로 도출된 최소차량대수인 14대로 작업을 완료할 때 Tabu Search 알고리즘은 상주횟수 3회, 총 반복횟수 500회일때 500분의 최대완료시간을 가진다. 동일한 상황에서 비교를 하기 위해 500분을 차량운행시간으로 한 Insertion방법의 실험결과는 <표5>과 같다. <표5>의 결과를 보면 Insertion방법은 차량운행시간이 500분일때 14대의 차량으로 계획기간내에 작업을 완료하기에는 불가능하다. 차량이 1대 증차된 15대로 작업을 완료할 수 있다.

<표5> 차량운행시간이 500분인 경우

	차량대수	makespan	완료시간 합
Tabu Search	14	500	6925
Insertion	15	500	6780

동일한 14대의 차량으로 모든 작업을 완수할 경우의 실험결과는 <표6>와 같다. <표6>의 결과를 보면 Tabu Search 알고리즘은 Insertion방법보다는 최대완료시간이 7.4%, 완료시간의 합은 4.94% 해가 향상되었고, 현장의 방법보다는 최대완료시간은 13.79%, 완료시간의 합은 8.22% 해가 향상된 것을 알 수 있다.

<표6> 차량 14대인 경우의 실험 결과

	차량대수	makespan	완료시간 합
Tabu Search	14	500	6925
Insertion	14	540	7285
현장의 방법	14	580	7545

4.2.2 이동물량정보가 랜덤한 경우

현재 비교실험을 하고 있는 Insertion방법은 현장의 방법의 개선 방법 중 하나이므로 현장의 방법은 제외하고 Insertion방법만을 가지고 비교한다. 기존의 논문에서 적용한 현장의 데이터를 이용한 경우 뿐만 아니라 랜덤한 다른 상황의 경우를 분석하여 제시한 알고리즘의 성능을 알아본다.

알고리즘의 성능을 알아보기 위해 랜덤하게 10개의 이동물량 데이터를 생성한다. 랜덤하게 데이터를 생성시 편리를 위하여 몇가지 가정을 한다. 가정사항은 다음과 같다.

- ① 데이터별 지점수 동일
- ② 데이터별 지점간 이동시간 동일
- ③ 데이터별 빈 컨테이너 수요량, 공급량 동일

기본예제에서 제시한 물량보다 2배 내외의 많은 이동물량을 생성한다. <표7>은 생성된 데이터의 이동물량 정보와 최소필요차량대수를 나타낸다.

<표7> 랜덤생성 이동물량과 최소필요차량대수

데이터 번호	실 지점수	이동물량(적컨+빈컨)	최소필요차량
M_Size 1	20	195	12
M_Size 2	20	307	17
M_Size 3	20	318	18
M_Size 4	20	322	17
M_Size 5	20	341	19
M_Size 6	20	391	22
M_Size 7	20	394	19
M_Size 8	20	396	21
M_Size 9	20	438	19
M_Size 10	20	371	22

랜덤 생성된 데이터를 분석한 결과는 <표8>과 같다. 각 데이터들의 개별 결과 정보는 [첨부1]을 참고하기 바란다. 먼저 초기해 생성시 랜덤한 방법을 선택한 경우를 살펴보도록 한다. 모든 이동물량을 완수하는 것을 기준으로 볼 때 Tabu Search 알고리즘이 평균적으로 3.4대의 차량대수 절감을 보인다. 그러나 makespan만을 비교해보면 Tabu Search 알고리즘이 Insertion방법에 비하여 3.32% 정도 더 많이 절린다. Insertion방법시 산출된 차량대수로 동일하게 이용한 경우는 Tabu Search 알고리즘이 평균적으로 10.4%정도 makespan의 감소를 보인다. 480분이란 차량운행 시간제약을 두고 실험을 한 경우 makespan은 시간제약으로 인해 거의 비슷하게 산출되고, 차량대수는 Tabu Search 알고리즘이 2.2대의 절감효과가 있다. Tabu Search 알고리즘을 이용하여 산출된 makespan을 차량운행 제한시간으로 할 때 Tabu Search 알고리즘이 평균적으로 2.4대의 차량절감효과가 있다.

<표8> 랜덤 생성 데이터 평균-makespan기준, 첫물량 랜덤선택

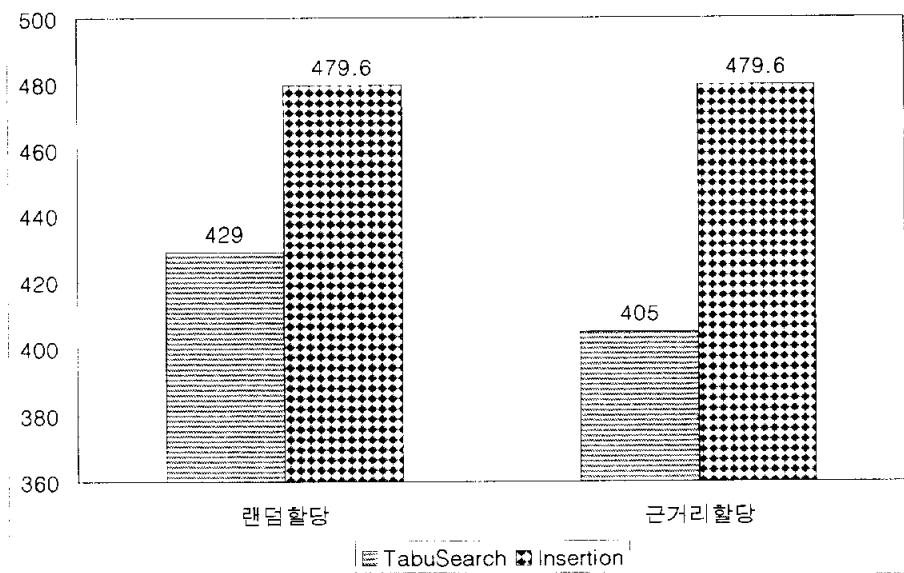
M_Size 1~10 (이동물량 : 347.3 최소차량대수 : 18.6)				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.6	495.5	9190.2	
Insertion 방법	22	479.6	9987.2	
현장의 방법	18.6	553.7	9519.9	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	429.8	9379.2	10.4% 향상
Insertion 방법	22	479.6	9987.2	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19.8	472.7	9291.1	2.2대 절감
Insertion 방법	22	479.6	9987.2	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.6	495.5	9190.2	2.4대 절감
Insertion 방법	21.1	495.3	9938.9	

다음으로 초기해 생성시 근거리 물량을 선택하여 할당하는 방법의 결과를 살펴보도록 한다. <표9>를 보면, 모든 이동물량을 완수하는 것을 기준으로 볼 때 동일하게 Tabu Search 알고리즘이 평균적으로 3.4대의 차량대수 절감을 보인다. 또한 makespan을 비교해보면 Tabu Search 알고리즘이 Insertion방법에 비하여 1.02% 정도 감소되었다. Insertion방법시 산출된 차량대수로 동일하게 이용한 경우는 Tabu Search 알고리즘이 평균적으로 15.6%정도 makespan의 감소를 보인다. 이것은 랜덤하게 첫물량을 할당할 때 보다는 성능이 향상됨을 보여준다. 480분이란 차량운행 시간제약을 두고 실험을 한 경우 Tabu Search 알고리즘이 makespan의 2.46% 감소와 차량대수

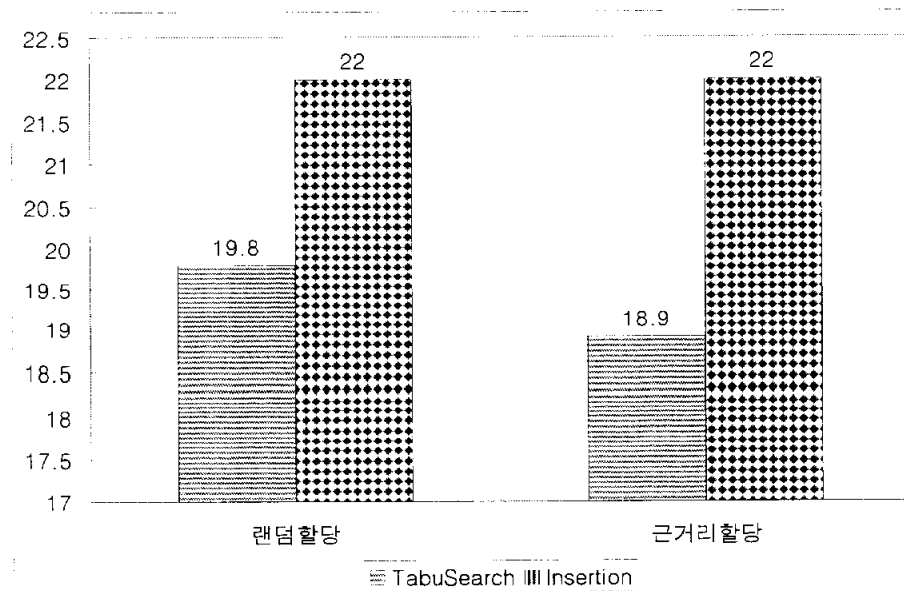
<표9> 랜덤생성 데이터 평균-makespan기준, 첫물량 근거리선택

M_Size 1~10 (이동물량 : 347.3 최소차량대수 : 18.6)				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.6	474.7	8772.9	
Insertion 방법	22	479.6	9987.2	
현장의 방법	18.6	553.7	9519.9	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	405	8734.6	15.6% 향상
Insertion 방법	22	479.6	9987.2	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.9	467.8	8786.4	3.1대 절감
Insertion 방법	22	479.6	9987.2	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.6	474.7	8772.9	3.7대 절감
Insertion 방법	22.3	473.6	9998.8	

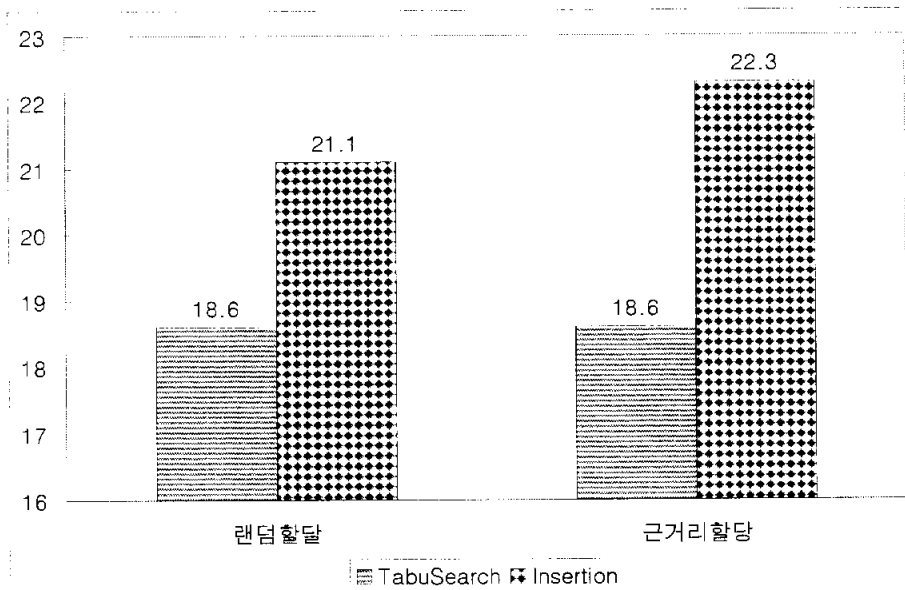
3.1대의 절감효과가 있다. Tabu Search 알고리즘을 이용하여 산출된 makespan을 차량운행 제한시간으로 할 때 Tabu Search 알고리즘이 평균적으로 3.7대의 차량절감효과가 있다. <그림17>에서 <그림19>는 초기해 생성방법에 따른 결과를 그림으로 비교한 것이다.



<그림17>Insertion방법 차량대수 적용시 makespan 감소



<그림18>Insetion방법 480분 기준시 차량대수 절감 정도



<그림19>Tabu Search의 makespan을 운행시간으로 적용시
차량대수 절감 정도

이 실험으로 판단할 수 있는 것은 물량정보가 랜덤하게 변화하더라도 Tabu Search 알고리즘이 기본예제의 경우와 같이 기존의 방법과 비교하여 상대적으로 성능이 나음을 보여주고, 또한 depot와의 거리가 주어진 경우는 첫물량을 할당할시에 근거리의 물량을 할당하는 것이 보다 최적해에 근접한 해를 이끌어 낸다는 것이다.

따라서, 근거리 물량할당 방법을 이용하여 초기해를 구하고 최종해를 찾는 방법을 적용하여 랜덤하게 이동물량을 40개 생성하였다. 랜덤하게 생성된 데이터를 이용하여 결과를 분석해 보면, 일반적으로 주어진 물량을 모두 완수하는데는 Tabu Search 알고리즘을 이용한 방법이 3.1대의 절감을 보인다. 초기해 생성시 근거리 물량 할당방법을 사용함으로써 makespan 역시 3.2% 감소됨을 알 수

<표10> 랜덤생성 40개 데이터 평균-makespan기준, 첫물량 근거리선택

M_Size 1~40 (이동물량 : 335.58 최소차량대수 : 18.23)				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.2	464	8405.3	
Insertion 방법	21.3	479.1	9607.4	
현장의 방법	18.2	534.1	9105.8	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21.3	399	8407.6	15.6% 향상
Insertion 방법	21.3	479.1	9607.4	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.5	458.9	8415.7	2.9대 절감
Insertion 방법	21.3	479.1	9607.4	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18.2	464	8405.3	4.4대 절감
Insertion 방법	22.6	462.7	9631.8	

있다. Insertion방법시 이용되는 차량대수를 적용한 결과를 살펴보면, Tabu Search 알고리즘을 이용한 방법이 15.6%정도 makespan이 감소됨을 알 수 있다. Insertion방법시 적용된 480분이란 차량운행시간을 적용하여 차량대수를 비교해 보면, Tabu Search 알고리즘을 이용한 방법이 2.9대의 차량대수 감소를 보인다. Tabu Search 알고리즘을 적용하여 도출된 makespan을 차량운행시간으로 적용할 때는 4.4대의 차량대수 절감을 보인다. 랜덤하게 생성된 40개의 데이터 결과에 대한 유의수준 5%일 때의 신뢰구간을 구함으로써 결과의 신뢰도를 알아본다. 모집단이 정규분포를 따른다고 가정하고 동일한 차량대수를 이용한 경우의 makespan을 기준으로 신뢰구간을 구한

다. 유의수준 5%일 때의 Tabu Search방법의 makespan신뢰구간은 [393.23, 404.97]이고, Insertion방법의 makespan신뢰구간은 [478.80, 479.45]이므로 makespan의 신뢰구간 폭이 좁고 서로 영향을 받지 않으므로 유의수준 5%인 경우 40개의 데이터 평균결과를 신뢰할 수 있다. 실험의 결과로 볼 때, 제시한 해법은 기존의 방법과 비교해서 상대적으로 보다 나은 해를 도출해 줄 수 있다.

그러나 이 외에 프로그램을 실행하는데 있어서 실험시간은 상대적으로 Tabu Search 알고리즘이 Insertion방법에 비하여 많이 걸린다. 전체 반복횟수를 많이 할수록 시간은 상대적으로 더 늘어난다. 본 성능분석을 위한 실험에서는 적게는 5분 내외에서 많게는 30분 내외로 사용자의 조정에 따라, 컴퓨터의 성능에 따라 가변적이다.

4.2.3 확률적인 시간 분포를 가지는 경우

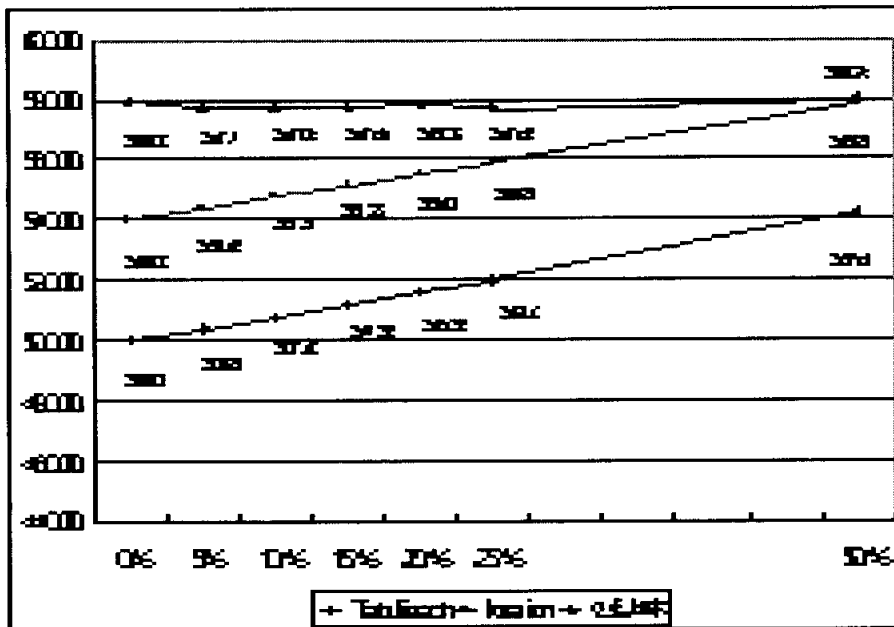
기본예제에서 제시한 이동시간은 확정적이다. 그러나 현실적으로 시간은 주변의 상황에 따라 가변적이다. 따라서 본 실험은 보다 현실적인 환경아래에서 알고리즘의 성능을 분석하기 위해서 시간을 가변적으로 둔다. 이러한 실험의 수행결과는 알고리즘의 강건성 정도를 알 수 있게 해준다. 기본예제에서 사용한 이동시간은 평균이동시간으로 하고, 각 이동시간은 삼각함수를 따른다고 가정하였다. 각 이동시간은 평균이동시간에 대해 5%~25%까지 변동이 있다고 가정을 한다. 즉, 5%의 시간 변동이 있는 경우, 평균이동시간이 50분일 때, 평균은 50분, 최소값은 47.5분, 최대값은 52.5분으로 그 범위안에서 이동시간은 가변적인 것이다. 각각의 방법에 대하여 변동 5%~25%일 경우를 실험하였다. 기본예제에서 depot와의 이동시간이 0이

므로 현장의 방법 첫물량 할당과 Tabu Search 알고리즘의 초기해 생성시 첫물량 할당은 랜덤선택으로 한다. 실험횟수는 각 10회로 하였다. <표11>은 10회 실험의 평균이다.

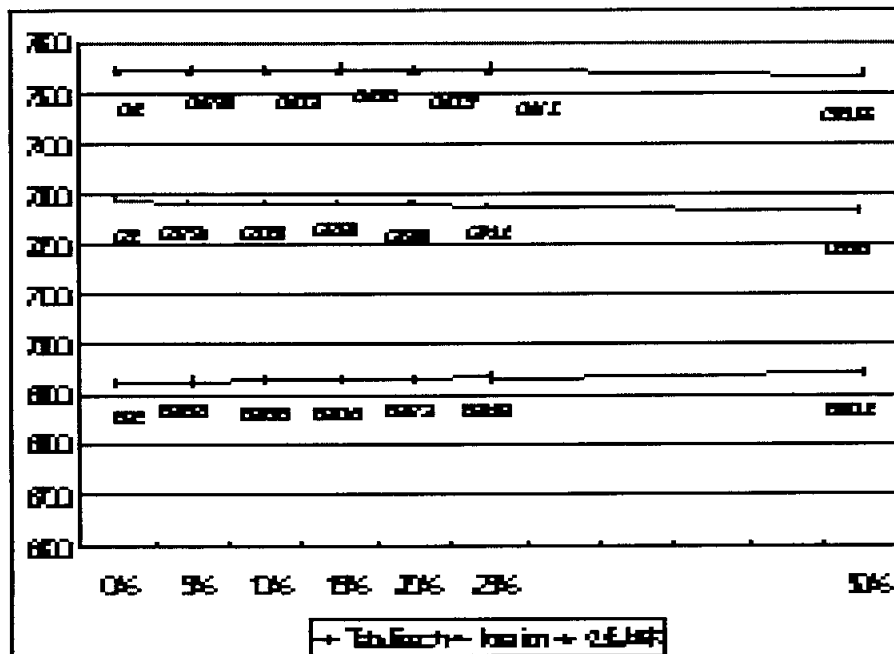
<표11> 확률적인 이동시간을 가지는 경우

makespan 평균						
	0%	5%	10%	15%	20%	25%
Tabu Search	500	503.85	507.70	511.56	515.58	519.73
Insertion	540	543.76	547.53	551.29	555.05	558.81
현장배차	580	577.11	577.04	577.47	578.39	577.48
완료시간의 합 평균						
Tabu Search	6925	6926.82	6928.63	6930.45	6932.27	6934.08
Insertion	7285	7282.95	7280.90	7278.84	7279.04	7274.74
현장배차	7545	7545.86	7543.22	7546.89	7543.52	7547.10

<그림20><그림21>을 살펴보면, 변동이 커질수록 Tabu Search 알고리즘의 수행도는 저하됨을 볼 수 있다. 그러나 확률적인 시간변동이 있더라도 Tabu Search 알고리즘이 다른 방법에 비해서 상대적으로 우수함을 알 수가 있다. 현장의 배차방법은 차량경로가 정해지지 않은 상태에서 그때의 상황에 따라 차량운행시간이 최소인 차량에 물량을 할당하여 차량경로가 결정되므로 makespan의 큰 변동은 없다. 그러나 Tabu Search 알고리즘과 Insertion 방법은 차량경로가 정해진 상황에서 생기는 시간변동이기에 때문에 평균을 기준으로 변동이 클수록 그 makespan이 조금씩 늘어남을 알 수가 있다. 그러나 시간변동이 50%이상인 경우는 Insertion 방법이나 현장의 배차방법은 makespan에서 큰 차이를 볼 수가 없다. 그러나 운행시간의 합을 보면 Insertion 방법이 현장의 배차방법보다는 효율적임을 알 수가 있다. 개별 실험에 대한 결과는 [첨부2]를 참고하기 바란다.



<그림20> 확률적인 이동시간을 가질 경우의 makespan의 변화



<그림21> 확률적인 이동시간을 가질 경우의 완료시간 합의 변화

제 5 장 결론 및 발전방향

본 연구는 화물 컨테이너 운송을 위한 Tabu Search기반의 차량 경로계획에 대해서 다루었다. 특히 공동으로 사용되는 빈 컨테이너의 운반을 고려하여 화물 컨테이너 운송체계를 통합적으로 다루었다. 기존연구현황을 살펴봄으로써 수송계획에 관한 전반적인 연구들을 개략적으로 정리해 보았고, 특히 본 연구와 직접적으로 관계가 있는 컨테이너 화물운송계획에 관한 기존의 연구들을 조사 정리하였다.

본 연구에서는 컨테이너 화물운송 시스템에서 효율적인 차량운행을 위하여 Tabu Search를 기반으로 한 차량경로계획 절차를 제시하였다. 제시된 절차는 2단계로 구성되었다. 우선 1단계에서는 Maxwell 모형을 기초로 최적화 수리모형을 수립하여 최소차량대수를 산출한다. 1단계의 결과를 이용하여 2단계에서는 Tabu Search 알고리즘을 기반으로 주어진 시간 내에서 최소의 차량으로 컨테이너를 운송할 수 있도록 경로를 설정한다. 제시된 절차는 여러 컨테이너 운송환경 하에서 제시된 해법과 비교 분석하여 그 효율성과 타당성을 검증하였다. 실험에서는 물량정보를 랜덤하게 변경시키고, 확률적인 시간을 따르게 하여 여러 가지 변경된 상황 아래에서도 본 연구가 제시하는 알고리즘이 얼마나 유용하고, 효율성있고, 강건성을 갖는지 실험하고 결과를 분석하였다. 결론적으로 본 연구에서 제시한 알고리즘이 기존의 유사환경에서 제시된 해법보다 상대적으로 우수함을 보였다.

본 연구에서는 모든 실질적인 시간과 용량제약에 관한 부분까지 다루고 있지는 않다. 컨테이너 유통경로는 물량이 증가할수록, 그

유통경로 또한 복잡해져가므로, 실질적인 제약사항을 추가해야하는 연구가 앞으로 필요하다. 따라서 화물 Pickup과 Delivery 시간제약이 있는 Time Window를 가진 경우의 차량경로문제로 확장 할 수 있을 것이다. 또한 본 연구에서 가정한 Full Load, 즉 물품 및 빈 물류용기의 운반 단위를 동일하게 하고, 차량의 운반 Capacity도 한 단위로 가정하고 있는 부분을 20feet 컨테이너 2개를 한꺼번에 운반할 수 있도록 변형시켜 하나의 차량으로 1개 이상의 물량을 운반할 수 있도록 확장하는 연구가 필요하다. 이러한 문제에 대한 연구는 추후의 연구과제로 남겨 놓는다.

참고 문헌

1. 강인선(1996), 입지선정 및 차량경로문제, *한국공업경영학회지* 제 19권, 제37호.
2. 고창성, 정기호, 신재영(2000), 컨테이너 셔틀운송을 위한 차량대수 결정, *경영과학* 제17권, 제2호 pp.87-95.
3. 고창성, 신재영(2002), Aggregate Transportation Planning Considering Three Types of Container Vehicles, *한국경영과학회/대한산업공학회 '02춘계공동학술대회*
4. 고창성, 박성찬, 정기호, 신재영, 하태영, 양천석(2001), 추계적 특성을 고려한 컨테이너운송 차량대수 결정, *한국경영과학회/대한산업공학회 '01춘계공동학술대회* pp.205-208.
5. 고창성, 정기호, 신재영, 김홍배, 양천석, 박성찬(2002), 추계적 작업환경에서 컨테이너 운송차량대수 결정, *한국생산관리학회* 제12권, 제3호, pp.17-32
6. 김갑환, 고창성, 신재영(1998), 화물수송계획 및 운영의 정량적 모형에 대한 조사 연구, *산업공학* 제 11권, 제1호 pp.1-14.
7. 김동희, 이창호, 김봉선(1997), 컨테이너 화물수송을 위한 차량배차 의사 결정지원시스템, *Journal of the Korean Institute of Industrial Engineers* Vol.23, No.2,
8. 김창은(1990), Survey of Traveling Salesman Problem, *공업경영학회지* 제13권 제22호.
9. 김내현(1993), 배달과 회수를 고려한 차량경로 문제, *산업공학회지* 제16권 제28호
10. 김여근, 윤복식, 이상복 공저(1997), 메타휴리스틱, pp.261-286, pp.410-411, 영지문화사

11. 권상호, 강맹규(1999), 비대칭 외판원 문제에서 3-Opt를 응용한 새로운 발견적 알고리즘, *공업경영학회지* 제22권 제52호
12. 백시현, 김내현(1999), 외판원 문제에 대한 효율적인 새로운 경험적 방법개발, *공업경영학회지* 제22권 제51호.
13. 서순근, 이병호(1998), 방사형 물류체계에서 수송장비의 보유대수 결정과 분배정책, *산업공학* 제11권 제1호, pp.56-66
14. 서순근, 곽준호(2001), 해상수송망에서 공컨테이너의 재배치 및 보충정책 수립을 위한 시뮬레이션모형의 개발, *IE Interfaces* Vol.14, No.3, pp.255-262,
15. 신해웅, 강맹규(1991), 다회방문을 허용하는 차량경로문제의 발견적 해법, *공업경영학회지* 제14권 제24호.
16. 송계의(1998), 물류경영론, 2판, 5장1절, 6장, 문영사, 서울.
17. 이명호, 김내현, 신재율(2000), 자동배차 지원시스템의 발견적 알고리즘 설계, *산업경영시스템학회지* 제23권 제58집.
18. 이철영(1998), 항만물류시스템, pp372-399, 효성출판사.
19. <http://www.kca.co.kr/> 한국컨테이너 부두공단 홈페이지
20. Yafeng Du, Randolph Hall(1997), Fleet sizing and empty equipment redistribution for center-terminal transportation networks, *Management Science* Vol.43, No.2.
21. W.L.Maxwell, J.A.Muckstadt(1982), Design of Automatic Guided Vehicle Systems, *IIE transactions*, Vol.14, No.2, pp.114-124

[첨부1] M_Size 10개 개별 데이터 결과

M Size 1 (이동물량 : 195 최소차량대수 : 12) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	461	5509	
Insertion 방법	13	480	5940	
현장의 방법	12	587	6889	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	13	427	5519	
Insertion 방법	13	480	5940	
11.04% 향상				
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	461	5509	
Insertion 방법	13	480	5940	
1대 절감				
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	461	5509	
Insertion 방법	14	461	5921	
2대 절감				

M_Size 1 (이동물량 : 195 최소차량대수 : 12) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	455	5417	
Insertion 방법	13	480	5940	
현장의 방법	12	587	6889	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	13	436	5601	9.17% 향상
Insertion 방법	13	480	5940	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	455	5417	1대 절감
Insertion 방법	13	480	5940	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	12	455	5417	2대 절감
Insertion 방법	14	453	5937	

M_Size 2 (이동물량 : 307 최소차량대수 : 17) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	499	8434	
Insertion 방법	21	479	9470	
현장의 방법	17	553	9058	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	415	8639	13.36% 향상
Insertion 방법	21	479	9470	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	476	8515	3대 절감
Insertion 방법	21	479	9470	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	499	8434	3대 절감
Insertion 방법	20	499	9412	

M_Size 2 (이동물량 : 307 최소차량대수 : 17) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	493	8334	
Insertion 방법	21	479	9470	
현장의 방법	17	553	9058	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	397	8278	17.12% 향상
Insertion 방법	21	479	9470	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	466	8335	3대 절감
Insertion 방법	21	479	9470	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	493	8334	4대 절감
Insertion 방법	21	493	9450	

M Size 3 (이동물량 : 318 최소차량대수 : 18) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	493	8845	
Insertion 방법	21	480	9455	
현장의 방법	18	542	9129	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	431	9020	10.20% 향상
Insertion 방법	21	480	9455	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	473	8914	2대 절감
Insertion 방법	21	480	9455	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	493	8845	2대 절감
Insertion 방법	20	493	9465	

M_Size 3 (이동물량 : 318 최소차량대수 : 18) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	473	8468	
Insertion 방법	21	480	9455	
현장의 방법	18	542	9129	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	400	8354	17.67% 향상
Insertion 방법	21	480	9455	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	473	8468	3대 절감
Insertion 방법	21	480	9455	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	473	8468	3대 절감
Insertion 방법	21	472	9452	

M_Size 4 (이동물량 : 322 최소차량대수 : 17) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	499	8405	
Insertion 방법	19	480	8787	
현장의 방법	17	617	9420	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	454	8580	5.42% 향상
Insertion 방법	19	480	8787	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	479	8508	1대 절감
Insertion 방법	19	480	8787	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	499	8405	1대 절감
Insertion 방법	18	499	8689	

M_Size 4 (이동물량 : 322 최소차량대수 : 17) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	482	8110	
Insertion 방법	19	480	8787	
현장의 방법	17	617	9420	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	433	8183	9.79% 향상
Insertion 방법	19	480	8787	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	18	452	8091	1대 절감
Insertion 방법	19	480	8787	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	17	482	8110	2대 절감
Insertion 방법	19	481	8757	

M_Size 5 (이동물량 : 341 최소차량대수 : 19) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	501	9441	
Insertion 방법	22	479	10008	
현장의 방법	19	497	9082	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	439	9585	8.35% 향상
Insertion 방법	22	479	10008	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	20	479	9504	2대 절감
Insertion 방법	22	479	10008	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	501	9441	2대 절감
Insertion 방법	22	501	10055	

M_Size 5 (이동물량 : 341 최소차량대수 : 19) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8768	
Insertion 방법	22	478	10008	
현장의 방법	19	497	9082	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	401	8661	
Insertion 방법	22	478	10008	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8768	
Insertion 방법	22	478	10008	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8768	
Insertion 방법	23	466	10193	

M_Size 6 (이동물량 : 391 최소차량대수 : 22) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	511	11130	
Insertion 방법	27	480	12232	
현장의 방법	22	559	11371	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	27	428	11501	10.83% 향상
Insertion 방법	27	480	12232	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	24	473	11296	3대 절감
Insertion 방법	27	480	12232	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	511	11130	3대 절감
Insertion 방법	25	510	12304	

M_Size 6 (이동물량 : 391 최소차량대수 : 22) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	486	10619	
Insertion 방법	27	480	12232	
현장의 방법	22	559	11371	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	27	391	10492	18.54% 향상
Insertion 방법	27	480	12232	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	23	474	10772	4대 절감
Insertion 방법	27	480	12232	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	486	10619	5대 절감
Insertion 방법	27	486	12330	

M_Size 7 (이동물량 : 394 최소차량대수 : 19) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	484	9155	
Insertion 방법	22	480	9890	
현장의 방법	19	528	9521	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	425	9307	11.46% 향상
Insertion 방법	22	480	9890	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	20	469	9297	2대 절감
Insertion 방법	22	480	9890	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	향상정도
Tabu Search	19	484	9155	2대 절감
Insertion 방법	21	484	9806	

M_Size 7 (이동물량 : 394 최소차량대수 : 19) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	466	8809	
Insertion 방법	22	480	9890	
현장의 방법	19	528	9521	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	399	8714	16.88% 향상
Insertion 방법	22	480	9890	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	466	8809	3대 절감
Insertion 방법	22	480	9890	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	466	8809	3대 절감
Insertion 방법	22	463	9875	

M_Size 8 (이동물량 : 396 최소차량대수 : 21) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	506	10545	
Insertion 방법	27	479	12128	
현장의 방법	21	540	10258	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	27	406	10840	15.24% 향상
Insertion 방법	27	479	12128	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	23	462	10576	4대 절감
Insertion 방법	27	479	12128	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	506	10545	4대 절감
Insertion 방법	25	506	12062	

M_Size 8 (이동물량 : 396 최소차량대수 : 21) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	478	9902	
Insertion 방법	27	479	12128	
현장의 방법	21	540	10258	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	27	383	9782	20.04% 향상
Insertion 방법	27	479	12128	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	478	9902	6대 절감
Insertion 방법	27	479	12128	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	21	478	9902	6대 절감
Insertion 방법	27	476	12102	

M_Size 9 (이동물량 : 438 최소차량대수 : 19) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	500	9468	
Insertion 방법	22	480	10190	
현장의 방법	19	577	9771	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	438	9585	8.75%
Insertion 방법	22	480	10190	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	20	480	9539	2대 절감
Insertion 방법	22	480	10190	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	향상정도
Tabu Search	19	500	9468	2대 절감
Insertion 방법	21	499	9982	

M_Size 9 (이동물량 : 438 최소차량대수 : 19) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8824	
Insertion 방법	22	480	10190	
현장의 방법	19	577	9771	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	404	8857	15.83% 향상
Insertion 방법	22	480	10190	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8824	3대 절감
Insertion 방법	22	480	10190	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	19	467	8824	4대 절감
Insertion 방법	23	467	10120	

M_Size 10 (이동물량 : 371 최소차량대수 : 22) - 랜덤선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	501	10970	
Insertion 방법	26	479	11772	
현장의 방법	22	537	10700	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	26	435	11216	9.19% 향상
Insertion 방법	26	479	11772	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	24	475	11253	2대 절감
Insertion 방법	26	479	11772	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	501	10970	3대 절감
Insertion 방법	25	501	11693	

M_Size 10 (이동물량 : 371 최소차량대수 : 22) - 근거리 선택				
모든 이동물량 완수기준				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	480	10478	
Insertion 방법	26	479	11772	
현장의 방법	22	537	10700	
Insertion방법의 차량대수를 적용한 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	26	406	10424	15.24% 향상
Insertion 방법	26	479	11772	
Insertion 방법의 차량운행시간(480분)기준인 경우				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	480	10478	4대 절감
Insertion 방법	26	479	11772	
Tabu Search의 makespan을 차량운행 제한시간으로 적용				
방법	차량대수	makespan	완료시간의 합	비 고
Tabu Search	22	480	10478	4대 절감
Insertion 방법	26	479	11772	

[첨부2]확률적인 시간분포를 가지는 경우의 개별 실험결과

Tabu Search makespan					
	5%	10%	15%	20%	25%
Sim_1	502.27	504.14	506.21	508.28	510.35
Sim_2	503.11	506.22	509.33	512.44	516.80
Sim_3	506.92	513.84	520.76	527.68	534.61
Sim_4	505.14	510.29	515.43	520.58	525.72
Sim_5	503.88	507.76	511.65	515.53	519.41
Sim_6	505.51	511.03	516.54	522.05	527.56
Sim_7	503.25	506.51	509.76	513.02	516.32
Sim_8	502.49	504.98	507.47	509.95	512.44
Sim_9	501.03	502.05	503.19	505.92	508.64
Sim_10	505.10	510.20	515.29	520.39	525.49

Tabu Search 완료시간의 합					
	5%	10%	15%	20%	25%
Sim_1	6914.51	6904.01	6893.52	6883.02	6872.53
Sim_2	6928.58	6932.16	6935.74	6939.33	6942.91
Sim_3	6930.30	6935.60	6940.90	6946.20	6951.50
Sim_4	6931.58	6938.17	6944.75	6951.33	6957.92
Sim_5	6917.17	6909.34	6901.51	6893.68	6885.85
Sim_6	6933.37	6941.74	6950.11	6958.49	6966.86
Sim_7	6923.71	6922.42	6921.13	6919.84	6918.55
Sim_8	6927.71	6930.42	6933.14	6935.85	6938.56
Sim_9	6927.04	6929.08	6931.12	6933.16	6935.20
Sim_10	6934.19	6943.38	6952.57	6961.76	6970.95

Insertion 방법 makespan					
	5%	10%	15%	20%	25%
Sim_1	543.71	547.42	551.13	554.85	558.56
Sim_2	541.62	543.25	544.87	546.50	548.12
Sim_3	541.93	543.87	545.80	547.74	549.67
Sim_4	540.11	540.22	540.33	540.44	540.55
Sim_5	543.93	547.85	551.78	555.71	559.64
Sim_6	545.19	550.38	555.57	560.76	565.95
Sim_7	544.14	548.28	552.41	556.55	560.69
Sim_8	546.05	552.10	558.14	564.19	570.23
Sim_9	547.82	555.63	563.45	571.27	579.08
Sim_10	543.13	546.25	549.38	552.50	555.63

Insertion 방법 완료시간의 합					
	5%	10%	15%	20%	25%
Sim_1	7277.89	7270.79	7263.68	7256.58	7249.47
Sim_2	7270.48	7255.96	7241.43	7226.91	7212.38
Sim_3	7294.43	7303.86	7313.29	7322.72	7332.15
Sim_4	7282.73	7280.46	7278.20	7275.93	7273.66
Sim_5	7274.08	7263.16	7252.24	7241.32	7230.40
Sim_6	7291.54	7298.08	7304.62	7311.17	7317.71
Sim_7	7289.56	7294.12	7298.67	7303.23	7307.79
Sim_8	7294.42	7303.85	7313.27	7322.70	7332.12
Sim_9	7271.74	7258.48	7245.21	7231.95	7218.69
Sim_10	7282.60	7280.20	7277.80	7275.40	7273.00

현장배차 makespan					
	5%	10%	15%	20%	25%
Sim_1	581.59	579.98	583.07	583.10	581.11
Sim_2	573.17	571.83	569.06	571.06	571.90
Sim_3	584.86	584.19	584.39	583.37	582.03
Sim_4	540.76	539.31	539.63	542.67	541.13
Sim_5	583.06	583.46	585.14	583.49	584.19
Sim_6	583.81	584.35	583.79	583.27	580.04
Sim_7	579.50	580.48	580.00	580.79	579.02
Sim_8	581.91	584.93	585.63	588.96	587.40
Sim_9	580.90	579.24	584.38	584.23	585.64
Sim_10	581.55	582.63	579.64	583.00	582.31

현장배차 완료시간의 합					
	5%	10%	15%	20%	25%
Sim_1	7599.46	7588.65	7609.84	7594.31	7570.98
Sim_2	7438.20	7436.20	7436.00	7445.86	7460.24
Sim_3	7637.20	7636.42	7628.76	7633.62	7602.58
Sim_4	7354.22	7348.67	7337.38	7337.37	7350.54
Sim_5	7486.75	7495.02	7499.19	7497.44	7514.61
Sim_6	7551.04	7551.76	7556.36	7558.44	7519.24
Sim_7	7597.30	7605.23	7611.95	7587.44	7642.45
Sim_8	7652.09	7652.73	7656.20	7657.61	7673.92
Sim_9	7563.29	7561.49	7584.41	7594.68	7578.65
Sim_10	7549.00	7556.04	7548.84	7528.43	7557.76

[부록1] AMPL 공차이동 최소화 수리모형

1. 수리모형 모델링 파일을 프로그래밍한다.

File Name : fleet.mod

#파라미터

param N; # 노드수

param Ta{i in 1..N, j in 1..N};

#장소i에서 장소j로 화물 컨테이너를 싣고 이동하는데 소요되는 시간
(적재/하역시간포함)

param v{i in 1..N, j in 1..N};

#화물 컨테이너를 싣고 장소i에서 장소j로 이동하는 횟수

param Tb{i in 1..N, j in 1..N};

#장소 i에서 장소j로 빈 컨테이너를 싣고 이동하는데 소요되는 시간

param Tc{i in 1..N, j in 1..N};

#장소 i에서 장소j로 공차로 이동하는데 소요되는 시간

param NF{i in 1..N}; #장소 i에서의 차량 순흐름

param s{i in 1..N}; #장소 i에 공급해 줄 수 있는 빈 컨테이너의 수

param d{i in 1..N}; #장소 i에서 필요로 하는 빈 컨테이너의 수

#결정변수

var y{i in 1..N, j in 1..N}>=0;

#차량이 빈 컨테이너를 싣고 장소i에서 장소j로 이동하는 횟수

var x{i in 1..N, j in 1..N}>=0;

#차량이 공차로 장소i에서 장소 j로 이동하는 횟수

2. 수리모형 데이터 파일을 프로그래밍한다.

File Name : fleet.dat

param N := 12; #가상노드 포함

#적재이동시간

param Ta:	1	2	3	4	5	6	7	8	9	10	11	12 :=
1	0	50	30	35	40	35	30	30	30	0	0	0
2	50	0	30	35	40	35	30	35	35	0	0	0
3	30	30	0	5	10	25	30	35	35	0	0	0
4	35	35	5	0	5	20	25	30	30	0	0	0
5	40	40	10	5	0	15	20	25	25	0	0	0
6	35	35	25	20	15	0	10	15	15	0	0	0
7	30	30	30	25	20	10	0	5	5	0	0	0
8	30	35	35	30	25	15	5	0	5	0	0	0
9	30	35	35	30	25	15	5	5	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0;

#이동횟수

param v:	1	2	3	4	5	6	7	8	9	10	11	12 :=
1	0	0	0	0	15	0	0	47	2	0	0	0
2	0	0	0	0	28	0	0	5	2	0	0	0
3	0	0	0	0	22	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	1	0	0	0
5	3	0	10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0	0	0	0
8	21	0	2	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	4	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0;

#빈 컨테이너 이동시간

param Tb:	1	2	3	4	5	6	7	8	9	10	11	12 :=
1	0	50	30	35	40	35	30	30	30	0	0	0
2	50	0	30	35	40	35	30	35	35	0	0	0
3	30	30	0	5	10	25	30	35	35	0	0	0
4	35	35	5	0	5	20	25	30	30	0	0	0
5	40	40	10	5	0	15	20	25	25	0	0	0
6	35	35	25	20	15	0	10	15	15	0	0	0
7	30	30	30	25	20	10	0	5	5	0	0	0
8	30	35	35	30	25	15	5	0	5	0	0	0
9	30	35	35	30	25	15	5	5	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0;

#공차 이동시간

param Tc:	1	2	3	4	5	6	7	8	9	10	11	12 :=
1	0	50	30	35	40	35	30	30	30	0	0	0
2	50	0	30	35	40	35	30	35	35	0	0	0
3	30	30	0	5	10	25	30	35	35	0	0	0
4	35	35	5	0	5	20	25	30	30	0	0	0
5	40	40	10	5	0	15	20	25	25	0	0	0
6	35	35	25	20	15	0	10	15	15	0	0	0

7	30	30	30	25	20	10	0	5	5	0	0	0
8	30	35	35	30	25	15	5	0	5	0	0	0
9	30	35	35	30	25	15	5	5	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0;

#차량순흐름

param NF :=

1	-40
2	-35
3	-10
4	-1
5	52
6	0
7	0
8	33
9	1
10	15
11	-15
12	0;

#공급해 줄수 있는 빈컨테이너 수

param s :=

1	0
2	0
3	10
4	0

5	10
6	0
7	0
8	40
9	0
10	0
11	0
12	0;

#필요한 빈 컨테이너 수

param d :=

1	10
2	30
3	0
4	0
5	0
6	5
7	5
8	0
9	0
10	0
11	0
12	10;

3. RUN파일을 프로그래밍 한다.

File Name : fleet.run

```
reset;                #현재 입력된 데이터를 초기화시킨다.
model fleet.mod;      #모델링 정보를 입력한다.
data fleet.dat;       #모델링 정보에 따른 실험데이터를 입력한다.
option solver cplexamp.exe; #Solver Engine으로 CPLEX를 이용한다.
option cplex_options 'nodes=500000 backtrack=1';
                    #AMPL에서 CPLEX의 파라미터를 조정한다.
                    #Branch and Bound의 탐색노드수를 500000으로 조정한다.
                    #소요시간을 단축하기 위한 파라미터를 조정한다.

solve;
#현재 입력된 모델링 정보와 데이터 정보를 이용하여 문제를 계산한다.
display TR;           #목적함수의 시간을 보여준다.
display _solve_time;  #문제를 계산하는데 소요된 시간을 보여준다.
display x;            #결정변수의 값을 보여준다.
display y;            #결정변수의 값을 보여준다.
```

4. 배치파일을 프로그래밍한다.

File Name : fleet.bat

```
ampl fleet.run >fleet_result.txt #run파일을 실행시켜 텍스트화면으로 결과를 보여준다.
```

5. 배치파일을 실행시킨 결과는 다음과 같다.

CPLEX 6.0.2: nodes=500000

backtrack=1

CPLEX 6.0.2: optimal solution; objective 6655

16 iterations (9 in phase I)

TR = 6655

_solve_time = 0.0400576

x [*,*]

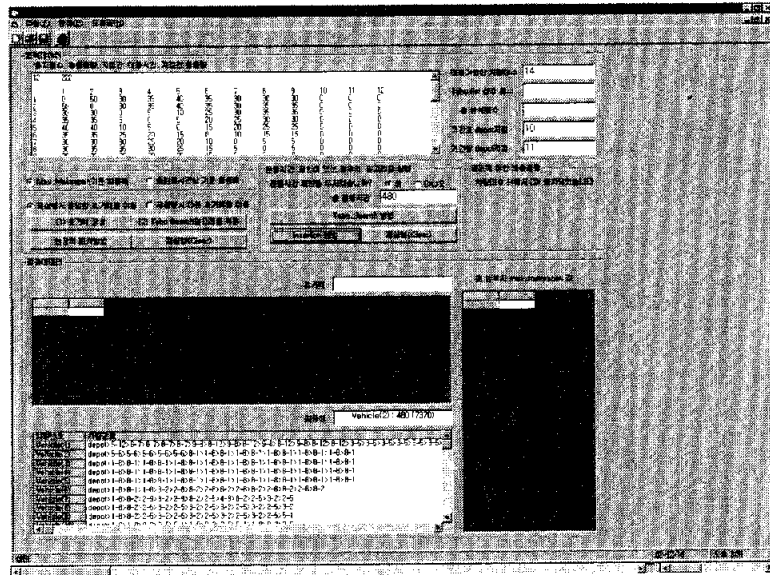
:	1	2	3	4	5	6	7	8	9	10	11	12	:=
1	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	
5	1	5	20	1	0	0	0	0	0	0	15	0	
6	4	0	0	0	0	0	0	1	0	0	0	0	
7	0	0	0	0	0	0	0	5	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	1	0	0	0	0	
10	15	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	
12	10	0	0	0	0	0	0	0	0	0	0	0	;

```

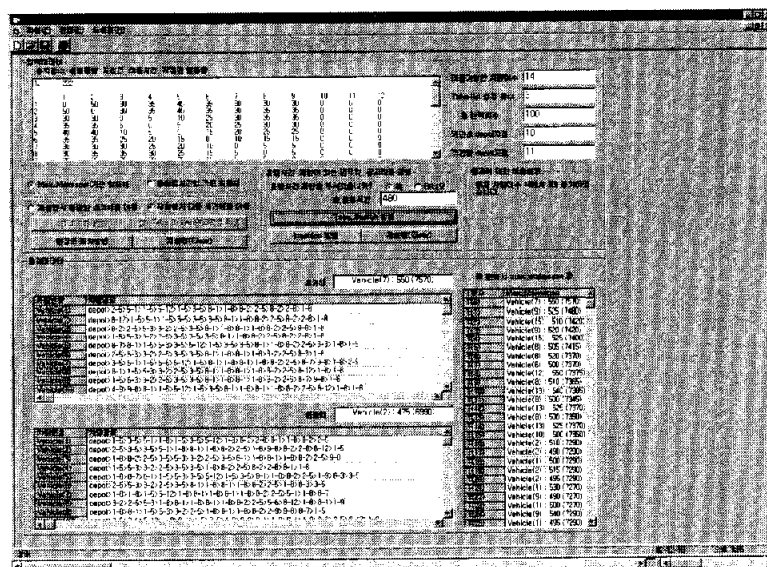
y [*,*]
:   1   2   3   4   5   6   7   8   9  10  11  12   :=
1   0   0   0   0   0   0   0   0   0   0   0   0
2   0   0   0   0   0   0   0   0   0   0   0   0
3   0  10   0   0   0   0   0   0   0   0   0   0
4   0   0   0   0   0   0   0   0   0   0   0   0
5   0   0   0   0   0   5   0   0   0   0   0   5
6   0   0   0   0   0   0   0   0   0   0   0   0
7   0   0   0   0   0   0   0   0   0   0   0   0
8  10  20   0   0   0   0   5   0   0   0   0   5
9   0   0   0   0   0   0   0   0   0   0   0   0
10  0   0   0   0   0   0   0   0   0   0   0   0
11  0   0   0   0   0   0   0   0   0   0   0   0
12  0   0   0   0   0   0   0   0   0   0   0   0 ;

```


Insertion방법 (480분 적용 화면)



Tabu Search알고리즘 시간제약 만족을 위해 증차되는 경우
(예제 파라미터 : 480분 시간제약, 상주횟수 3회, 총반복횟수 100회)



프로그램 코딩 일부분

Option Explicit

```
Public fMainForm As frmMain
Public Total_Vehicle_Num As Long '총 차량대수
Public Tabu_Num As Long 'tabu list에 상주횟수
Public T As Long '총 반복횟수
Public Sdepot As Long '기간초 depot지점
Public Edepot As Long '기간말 depot지점
Public Spot_Num As Long '총 지점수
Public Total_Con_Num As Long '총 컨테이너 물동량
Public Spot_Time(2000, 2000) As Long '지점 간 이동시간
Public Job_Complete_Time(9999) As Long
'작업하나를 수행하는데 걸리는 시간
Public Job_Info(9999) As String '작업정보
Public Job_Prev(9999) As Long '작업의 앞작업
Public Job_Succ(9999) As Long '작업의 뒤작업
Public Count_Num(9999) As Long '차량에 할당된 작업의 갯수
Public Vehicle_Num(2000, 2000) As String
'차량번호와 차량에 할당된 작업정보
Public Vehicle_Num_2(2000, 2000) As Long
'차량번호와 차량에 할당된 작업번호
Public Assigned_Job(9999) As Long
'초기해생성시 작업이 모두 할당되는지 여부확인시이용
Public Vehicle_Makespan(9999) As Long
'차량의 makespan
Public W As Long
```

```

Public Min_Makespan_Vehicle_Num As Long
Public Optimal_Sol(2000, 2000) As String '이하 optimal sol정보들
Public Optimal_Sol_2(2000, 2000) As Long
Public Optimal_Sol_Makespan(9999) As Long
Public Optimal_Sol_Count_Num(9999) As Long
Public Optimal_Sol_Max_Makespan As Long
Public Optimal_Sol_Total_CT As Long '완료시간의 총합
Public StoreIni_Sol(2000, 2000) As String '이하 초기해 정보들
Public StoreIni_Sol_2(2000, 2000) As Long
Public StoreIni_Sol_Makespan(9999) As Long
Public StoreIni_Sol_Count_Num(9999) As Long
Public StoreIni_Sol_Total_CT As Long
Public iteration As Long '반복횟수
Public Busiest_Vehicle As Long '가장 busiest한 차량번호
Public Tabu_List(9999) As String 'tabu list에 올려진 작업정보
Public Tabu_List_2(9999) As Long 'tabu list에 올려진 작업번호
Public Tabu_Count(9999) As Long 'tabu list 상주횟수
Public Tabu_index As Long
Public Tabu As Long
Public Tabu_OK As Long
Public Compare As Long
Public Move_Job As String '옮겨질 작업
Public Move_Job_2 As Long
Public Use_SameIni_Sol As Long
Public Add_Vehicle As Long
Public A As Long '할당여부 확인에 사용
Public Forced_End As Long
Public Limite_Time As Long

```

```
Public Number As Long
Public Check_Same As Long
Public Check_Same_Ok As Long
```

```
Private Sub Command1_Click()
```

```
    '초기해 생성시 명령버튼
```

- '1) 초기화를 시킨다.
- '2) 데이터를 입력받아 배열에 저장한다.
- '3) 초기해를 생성한다.
- '4) 초기해 생성한것을 그리드로 보여준다.

```
Dim i As Long
```

```
If rtfText.Text <> "" And Text1.Text <> "" And Text2.Text  
<> "" And Text3.Text <> "" And _Text4.Text <> "" And  
Text5.Text <> "" Then
```

```
    Call Initialization    '초기화시킨다.
```

```
    Call Input_Data        '데이터를 입력받는다.
```

```
    Call Make_Ini_Sol      '초기해를 생성한다.
```

```
    Call Show_Ini_Sol      '초기해를 보여준다.
```

```

If Option1.Value = True Then
    '초기해를 저장해둔다.
    For i = 1 To Total_Vehicle_Num
        StoreIni_Sol(i, StoreIni_Sol_Count_Num(i)) =
            Vehicle_Num(i, Count_Num(i))
        StoreIni_Sol_2(i, StoreIni_Sol_Count_Num(i)) =
            Vehicle_Num_2(i, Count_Num(i))
        StoreIni_Sol_Count_Num(i) = Count_Num(i)
        StoreIni_Sol_Makespan(i) = Vehicle_Makespan(i)
        StoreIni_Sol_Total_CT = Optimal_Sol_Total_CT
    Next i

    Command1.Enabled = False
ElseIf Option2.Value = True Then
    Command1.Enabled = False
End If

Else
    MsgBox "입력데이터가 부족합니다. 입력데이터를 모두 채우
        십시요"
End If

End Sub

Private Sub Command2_Click()
    'Tabu Search알고리즘을 적용하여 초기해를 향상시킨다.

    '1)최대 makespan을 가진 차량을 찾는다
    '2)가장 busiest한 차량의 makespan을 조정해본다.
    '옮겨질 작업과 옮겨질 위치를 찾는다.

```

'3)다른 차량으로 작업을 move할지 안할지 비교작업을 수행한다.

'4)다른 차량에 삽입시 삽입할 작업과 삽입될 지점을 찾는다.

```
Dim i As Long
```

```
Forced_End = 0
```

```
If Command1.Enabled = False Then
```

```
    Use_SameIni_Sol = 0
```

```
    Call Search_Max_Makespan
```

```
        '최대 makespan을 찾는다. busiest_vehicle정보도출
```

```
    Call Busiest_Vehicle_Adjustment
```

```
        '가장 busiest한 차량 조정작업
```

```
Do
```

```
    If Compare = 1 Then      '반복횟수 증가없이
```

```
        Call Search_Max_Makespan
```

```
        Call Busiest_Vehicle_Adjustment
```

```
    ElseIf Compare = 2 Then '반복횟수 증가
```

```
        Forced_End = 0
```

```
        Call Select_Move_Job_Implement
```

```
        '제거될 작업의 선택 remove position 정보도출
```

```
        Call Insert_Move_Job_Implement
```

```
        '삽입될 위치를 선택
```

```
    If Forced_End <> 0 Then
```

```
        iteration = T
```

```
        Call Show_Final_Sol
```

```
Label14.Caption = "Fesible Neighbor가 없어 강제 종
```

료 됨"

ElseIf Forced_End = 0 Then

iteration = iteration + 1

Call Check_Optimal_Sol

'반복횟수가 증가되면 current_sol과 optimal_sol 비교한다.

End If

End If

Loop Until iteration = T

Call Show_Final_Sol '최종해를 보여준다.

Command2.Enabled = False

Command3.Enabled = True

Else

MsgBox "생성된 초기해가 없습니다. 초기해를 먼저 생성시키고
클릭하십시오"

End If

End Sub

감사의 글

유명한 석학이 되고자 했던 것도 아니었고, 다만 공부가 좀더 하고 싶었던 제 욕심을 채우기 위해 지난 시간동안 가족들을 힘들게 했던 것 같습니다. 이제 감사의 글로 논문을 마무리하면서 제일먼저 부모님의 얼굴이 떠오릅니다. 가정형편이 넉넉지 못하여 대학원을 진학하려던 저를 그렇게 말리셨는데 힘들게 했던 가족에게 죄송한 마음과 함께 그래도 지난 2년동안 열심히 했다고, 이렇게 한편의 논문을 완성했다고 말씀드리고 싶습니다.

6년의 학교생활을 하면서 학부시절 한번이라도 힘들다고 느낀 적도 없었고, 선배들과 함께 어울려 밤샘 프로젝트 하는 그것마저도 저에게는 즐거움 그 자체였습니다. 그러나 대학원의 2년은 저에게 많은 생각을 갖게 해준 시간이었습니다. 공부가 힘들었다는 것이 아니라, 스스로 무엇을 해야 하는가를 깨닫게 해준 자기성찰의 시간이 아니었나 봅니다. 가끔씩 무리한 일정이 잡히거나, 혼자서 모든 일을 다 해야했던 그때를 지금 생각해 보면 조용히 잘 이겨냈고, 그것은 앞으로의 삶에 있어서 많은 밑거름이 되리라 생각합니다.

논문완성까지 세심한 배려를 아끼지 않으셨던 구평희 교수님께 감사드립니다. 교수님께서 배려해주신 덕분에 무사히 대학원을 졸업할 수 있었다 생각합니다. 부족한 부분이 보일 때도 화내지 않으시고 차근차근 가르침을 주시던 교수님, 교수님을 뵈는 지난 4년간 많은 가르침에 감사드립니다. 그리고 학부시절 갓 입학한 새내기때 대학생활의 바른길을 지도해주신 김병남 교수님, 엄격함 속에서도 잔잔한 정을 물씬 느끼게 해주시던 권혁무 교수님, 수업시작 전 좋은 글귀 한 구절을 읊어주시던 오수철 교수님, 수업시간 특유의 카리스마를 지니고 계신 이운식 교수님, 온화한 미소로써 학생들을 이끄시는 특이한 매력의 옥영석 교수님, 수업시간 학생들과 많은 대화를 이끌어 내셨던 김수용 교수님, 푸름이 느껴졌던 고시근 교수님, 젊은 학문의 김영진 교수님까지 모든 교수님들께 고개숙여 감사의 인사를 드립니다.

연구실에서 본의 아니게 내 눈치를 보게 만들었던 선배, 동기, 후배들에게 지면을 통해서나마 고마움을 전하고 싶습니다. 지난 4년의 연구실 생활 동안 먼저 졸업한 선배와 동기들의 얼굴도 하나하나 생각이 나고, 지금 연구실원들의 모습도 파노라마처럼 지나갑니다. 모두에게 좋은 동기, 좋은 선배의 기억으로 남고 싶습니다. 특히, 프로그래밍을 하다가 막힐 때, 가려운 부분을 잘 긁어주던 한섭 선배에게 고마움을 전하고 싶습니다.

사랑하는 나의 친구들! 친구들이 있었기에 외롭지 않게 생활을 한 것 같습니다. 나의 영원한 단짝 주야~ “언제나 현주 네가 내 옆에 있어서 힘이 되었던다. 쌍둥이처럼 모든 생활의 리듬이 같아 들어 있어도 하나라는 느낌을 지울 수가 없구나.” 내가 학생이라고 툭하면 술사라고 해도 군말없이 술 한잔 사주던 분길~ “분길아, 내가 여러 가지 문제로 힘들어 약해질때 네가 많이 다독겨줘서 힘을 얻었던다. 나중에 내가 힘들면 내가 너의 힘이 되어줄게” 항상 전화할 때가 나밖에 없다는 쑥맥 덩어리 혜진~ “첫 월급 탔다고 몸보신 시켜준 너의 맘이 이뻐서 내가 첫 월급타면 두배로 몸보신 시켜줄꺼야” 내가 무엇을 힘들어하는지 벗어나고 싶어하는지 함께 느껴주는 커피친구 수지~ “초등학교 단짝 친구를 여태 잘 만나고 있는 것도 하나의 행복이야, 커피향도 좋지만 우리도 이제 해맑은(?) 피부를 생각할 나이가 되지 않았어? 녹차로 바꿔볼까?” 나이가 들어가도 귀여움이 살포시 묻어나는 연정이 언니~ “언니가 필요할 때, 가끔씩 어리광 부리고 싶을 때 언니가 있어서 얼마나 다행이었는데 줄 몰라요. 언니의 조언도 언제나 고맙구요” 유아원 코흘리개들 뒤흔다꺼리에 정신이 없는 연정이~ “초등학교 운동장을 뛰어다니던 그때가 그림다. 나날이 이빠지는 널 보며 나도 덩달아 이빠지는 것 같아” 언제나 내편이 되어주고 힘들 때 나를 지켜준 사랑하는 나의 친구들이 있었기에 힘들지 않았습니다. 모두 행복하길 두손 모아 간절히 기도합니다. 앞으로 학문을 계속할 기회가 생기면, 공부에만 전념하고픈 열정을 가슴속에 가져봅니다. 사랑하는 나의 부모님, 교수님, 친구들 모두 감사합니다. 행복하세요.

장 동원 드림