

시간제약과 하역장 용량제약이 있는 차량경로문제에 대한 타부탐색 알고리즘*

장희정** · 이경식*** · 최은정** · 박성수**

A Tabu Search Algorithm for the Vehicle Routing Problem with Time Window and Dock Capacity Constraints*

Heejeong Zang** · Kyungsik Lee***
Eunjeung Choi** · Sungsoo Park**

■ Abstract ■

We consider a vehicle routing problem with time window and dock capacity constraints (VRPTD). In most traditional models of vehicle routing problems with time window (VRPTW), each customer must be assigned to only one vehicle route. However demand of a customer may exceed the capacity of one vehicle, hence at least two vehicles may need to visit the customer. We assume that each customer has its own dock capacity. Hence, the customer can be served by only a limited number of vehicles simultaneously. Given a depot, customers, their demands, their time windows and dock capacities, VRPTD is to get a set of feasible routes which pass the depot and some customers such that all demands of each customer are satisfied. Since VRPTD is NP-hard, a meta-heuristic algorithm is developed. The algorithm consists of two procedures: the route construction procedure and the route scheduling procedure. We tested the algorithm on a number of instances and computational results are reported.

Keyword : VRPTW, Dock Capacity Constraints, Tabu Search

논문접수일 : 2004년 3월 22일 논문게재확정일 : 2005년 9월 20일

* 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2002-042-D10553).

** 한국과학기술원 산업공학과

*** 한국외국어대학교 산업정보시스템공학부

1. 서 론

차량경로문제(Vehicle Routing Problem ; 이하 VRP)는 차량의 기지에 해당하는 거점(depot)마디에서 출발한 차량이 배달(delivery)이나 수집(pick-up) 등의 서비스를 요구하는 고객마디를 특정한 순서대로 방문하고 다시 거점으로 돌아오는 최소 비용의 차량경로 집합을 결정하는 문제이다. 이 때 각 고객은 반드시 하나의 차량경로에만 포함되어야 하고 모든 차량경로의 구간에서 차량에 적재된 총 물량은 차량의 용량(vehicle capacity)을 초과할 수 없다. 비용은 이동 거리 또는 이동 시간 등으로 대체할 수 있으며, 비용에 차량의 운행 시 발생하는 운용비용을 고정비용으로 포함시킬 수 있다.

VRP 문제는 물류 분야에서 중요한 문제이며 현실 문제에 적용될 때 여러 종류의 제약조건이 존재한다. 이 중에서 VRP에 시간제약(time window constraints)을 추가한 문제를 시간제약이 있는 차량경로문제(Vehicle Routing Problem with Time Windows ; 이하 VRPTW)로 부른다. 여기서 시간 제약이란 차량이 고객마디를 방문할 수 있는 시간이 도착가능시각과 도착한계시각 사이로 한정되어 있음을 의미한다. 고객마디의 도착가능시각은 그 마디에서 서비스를 시작할 수 있는 가장 빠른 시각이며 도착한계시각은 가장 늦은 시각이다. 거점마디에서의 도착가능시각은 고객들에게서 물량을 수집하기 위해 차량이 출발 가능한 가장 빠른 시각을 의미하며 도착한계시각은 차량이 고객을 방문하고 거점으로 돌아와야 하는 한계 시각을 의미한다.

지금까지 연구되어 온 VRPTW 모형은 모든 고객마디는 반드시 하나의 차량경로에만 포함되어야 한다는 가정을 가지고 있다. 그러나 현실에서는 고객의 물량이 차량 한 대 이상이거나 여러 개의 거점으로부터 서비스를 받는 고객이 존재하며, 그런 경우 여러 차량이 그 고객을 방문하지만 만약 두 대 이상의 차량이 동시에 고객을 방문할 경우 하역장의 용량제약 때문에 모든 차량이 동시에 서비스를 할 수 없는 경우가 발생한다. 여기서 하역장이란

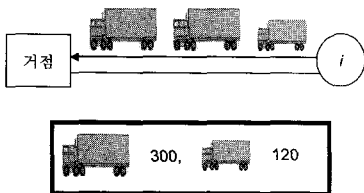
차량에 물량을 싣고 내리는 작업을 할 수 있는 장소로써 하역장 용량은 하역장에서 동시에 작업을 할 수 있는 차량의 대수로 정의된다. 즉, 하역장 용량 제약이란 마디에서 동시에 작업할 수 있는 차량의 수가 하역장 용량보다 작거나 같음을 의미한다. 또한 각 마디의 하역장에서 작업을 할 수 있는 차량의 최대 용량에는 한계가 있다고 가정하며 이를 하역장의 용적이라 부르기로 한다. 대표적인 예가 Beasley et al.[5]이 다룬 항공기 착륙 일정계획(Scheduling Aircraft Landing Problem)이나 선박 경로문제(Ship Routing Problem)이다. 그 외의 응용사례로 국내의 우편 운송망의 경우를 들 수 있다. 우편 운송망은 허브와 스포크 형태로서 하나의 교환센터를 중심으로 다수의 집중국이 존재하며 하나의 집중국은 관내에 다수의 집배센터를 포함하고 있다. 교환센터와 집중국 사이의 우편물량 흐름을 기간 운송망, 집중국과 집배센터 사이의 우편물량 흐름을 하부 운송망이라고 하는데 두 운송망에는 우편 물량이 차량 한 대 이상이어서 여러 차량경로에 포함되는 운송국들이 존재한다. 그리고 각 운송국에 시간제약과 하역장 용량제약이 존재하여 우편 운송계획 수립 시 이를 고려한 일정계획이 필요하다.

따라서 본 논문에서는 보다 현실적인 여건을 반영하기 위해 VRPTW에 하역장 용량제약을 고려한 시간제약과 하역장 용량제약이 있는 차량경로문제(Vehicle Routing Problem with Time Window and Dock Capacity Constraints ; 이하 VRPTD)를 제시하고 이 문제를 효과적으로 풀 수 있는 알고리즘을 제시하고자 한다. 실제 상황에서는 인력이나 공간의 문제로 하역장 용량이 1인 경우가 빈번히 발생하므로 본 논문에서는 VRPTD를 하역장 용량이 1인 문제로 제한해 수리모형으로 정식화하고 그에 대한 알고리즘을 구현한다.

VRP에 관한 연구는 오래 전부터 활발히 이루어져 왔으나[4] VRPTD에 관한 연구는 최근까지 거의 이루어지지 않았다. Beasley et al.[5]의 논문 등이 하역장 용량 제약을 고려한 문제를 다루나 이 경우는 차량경로가 고정되어 있어 일정계획 문제

(Scheduling Problem)만을 다루었다. 반면 VRPTD의 경우 차량경로 생성과 작업시각에 대한 일정계획을 동시에 결정해야 하므로 매우 해결하기 어려운 문제가 된다.

따라서 본 연구에서는 VRPTD를 사전처리 작업을 거친 후 차량경로 생성문제와 일정계획 문제의 두 문제로 구분하여 해결한다. 사전처리 작업이란 요구물량이 차량의 용량을 초과하는 고객마디에 대해 물량을 분할하고 일부 물량에 대해서는 직접경로를 생성하여 미리 차량을 할당하여 주는 과정이다. 어떠한 차량의 직접경로란 거점과 하나의 고객마디만을 왕복 운행하는 경로를 의미하며 거점으로 돌아오는 운송로에서는 그 차량의 적재율이 1이 되어야 한다. 물량을 분할할 때에는 고객마디의 요구물량과 하역장의 용적보다도 작거나 같은 용량을 가진 차량 중 용량이 큰 차량 유형 순으로 고려하며 이 때 직접경로로 수거하는 물량이 최대화 될 수 있도록 분할한다. 예를 들어 고객 i 의 물량은 800이고 하역장의 용적은 300이며 세 가지 유형의 차량이 있는데 유형 A의 용량은 500, 유형 B의 용량은 300, 유형 C의 용량은 120이라고 하자. 그러면 고객의 물량을 300, 300, 120, 80으로 분할하고 차량의 용량과 같은 앞의 세 물량에 대해서는 그 물량을 수집하기 위해 직접경로를 배정한다. 즉 두 대의 B차량, 한 대의 C차량에 해당하는 직접경로가 생성된다. 나머지 물량 80에 대해서는 차량경로 생성단계에서 서비스할 차량을 결정해준다. 이 예를 그림으로 표현하면 다음과 같다.



[그림 1] 사전처리작업의 예

차량경로 생성단계에서는 사전처리 후 직접경로로 수거하지 않는 물량을 처리하기 위해 하역장 용

량 제약을 제거한 VRPTW를 풀게 된다. 일정계획 단계에서는 사전처리 작업으로 생긴 직접경로와 차량경로 생성단계에서 생성된 경로에 쓰이는 차량들이 각 경유지에서 서비스를 시작하는 시각을 정해주는 문제를 다룬다. 이 단계는 차량이 운행하는 경로를 작업(job)으로 보고 마디를 기계로 보았을 때 기계의 사용 가능한 시간(각 마디의 서비스 허용시간대)이 제한되어 있는 개별공정 일정계획(Job Shop Scheduling Problem, 이하 JSSP)과 유사하다.

VRPTW와 JSSP에 대해서는 폭넓게 연구되어 왔으며 메타 휴리스틱과 최적화 기법을 이용한 다양한 해법들이 제시되고 있다. Solomon[15]은 처음으로 VRP의 다양한 종류의 차량경로 생성 휴리스틱을 VRPTW에 적용할 수 있도록 일반화하였으며 Koskosidis et al.[14]는 최적화 기법을 둔 휴리스틱을 제시하였다. Brando and Mercer[7]은 타부탐색을 이용한 해법을 제시한 바 있으며 Tan et al.[18]은 타부탐색과 시뮬레이티드 어닐링, 유전 알고리즘을 제시하고 그 결과를 비교하였는데 Solomon이 제시한 여러 문제를 풀었을 때 타부탐색이 최적해에 근사한 해를 제공하였다.

Baker and McMahon[3], Fisher et al.[9] 등에 의해 JSSP를 해결하기 위한 최적화 기법 연구가 다양하게 이루어졌으나 문제의 복잡성(computational complexity)으로 인해 크기가 작은 문제들에만 적용이 가능하였다[14]. 따라서 다양한 휴리스틱 알고리즘이 개발되어 왔는데 그 중 작업목록 일정계획(List Scheduling)은 많이 쓰이는 휴리스틱 알고리즘으로써 여러 가지 작업 우선순위(Dispatching Priority) 규칙에 의해 기계에서 작업들간의 순서를 정하는 알고리즘이다. Su et al.[16]은 여러 가지 작업 우선순위 규칙을 기반으로 '세 가지의 우선순위 규칙(Three Dominance Rules)'을 세우고 작업의 마감시간(due date)이 있는 JSSP를 풀어 작은 크기의 문제에서는 80% 이상의 문제에서 최적해를 구했으며 대부분의 큰 크기의 문제에 대해서도 기존의 휴리스틱에 비해 좋은 해를 제공하였다.

본 연구에서는 이러한 결과들에 근거하여 VRPTD

를 해결하기 위해 사전처리 작업과 타부탐색 알고리즘을 개발하는데 앞에서 설명한 차량경로 생성 단계와 일정계획 단계가 타부탐색 알고리즘을 구성한다.

앞에서 언급한 바와 같이 차량경로 생성 단계에서는 하역장 용량제약을 제거한 VRPTW를 풀게 되는데 이 때 구해진 해를 가능해라 부르기로 한다. 일정계획 단계를 마친 후 사전처리 작업 후 생긴 직접경로들과 가능해를 구성하는 차량경로들이 본 논문에서 고려하는 모든 제약을 만족하면 이 가능해를 일정 가능해라 부르기로 하고 그렇지 않으면 일정 비가능해라 부르기로 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 논문에서 고려하고 있는 문제에 대해 더 자세히 기술하고 3장에서는 문제를 해결하기 위한 알고리즘을 구체적으로 설명하며 4장에서는 본 논문에서 제시한 알고리즘을 예제에 적용한 결과에 대한 논의를 하고 마지막으로 5장에서 결론을 제시한다.

2. 문제정의

VRPTD에 대한 알고리즘을 구현하는 데 있어 필요한 기본적인 입력정보에는 거점과 고객, 물량을 운송하기 위한 수단인 차량에 관한 정보 등이 있다. 구체적으로 기술하면 다음과 같다. 이 부분의 수리 모형은 부록 A의 식 A1~A20을 참고하기 바란다.

(1) 마디집합과 운송로 집합

마디집합 $V = \{0, 1, \dots, n\}$ 은 한 개의 거점마디와 n 개의 고객마디로 구성되며 마디 0은 거점을, 마디 $i \in V \setminus \{0\}$ 는 고객을 나타낸다. 각 고객마디는 요구물량을 가지고 있으며 물량이 차량의 용량보다 큰 고객이 존재한다. E 는 마디간의 운송로 집합을 의미하며 기호 t_{ij} 는 마디 i 에서 마디 j 로 가는데 걸리는 시간을, c_{ij} 는 마디 i 에서 마디 j 까지의 거리($(i, j) \in E$)를 비용으로 환산한 값과 같다.

(2) 서비스 유형

고객마디에서의 서비스 유형은 수집으로 한다. 고객마디의 하역장에서는 차량에 물건을 싣는 작업이 이루어지며 거점마디의 하역장에서는 수집을 마치고 거점으로 돌아온 차량에 적재된 물량을 내리는 작업이 이루어진다.

(3) 마디 $i \in V$ 의 서비스 허용시간대(Time windows)

기호 e_i 는 마디 $i \in V$ 의 도착가능시각을, 기호 l_i 는 도착한계시각을 나타낸다. 마디 i 에서 거점으로 이동할 때의 소요시간은 l_0 와 e_i 의 차보다 작거나 같고, 거점에서 마디 i 로의 이동 소요시간은 l_i 와 e_0 의 차보다 작거나 같다. 이를 수식으로 표현하면 다음과 같다.

$$\begin{aligned} t_{i0} &\leq l_0 - e_i, & \text{for } i \in V \setminus \{0\} \\ t_{0i} &\leq l_i - e_0, & \text{for } i \in V \setminus \{0\} \end{aligned}$$

(4) 마디 $i \in V$ 의 하역장 용량과 하역장 용적

일반적으로 하역장 용량은 마디마다 다를 수 있으나 서론에서 언급했듯이 본 연구에서는 모든 마디의 하역장 용량이 1인 경우의 문제를 다룬다. 하역장 용적은 Q_i 로 표시한다.

(5) 차량에 대한 정보

각 차량의 유형은 차량의 용량에 따라 구분되며 차량 유형이 $w (w=1, 2, \dots, m)$ 인 차량의 용량은 q_w 로 표시하고 $q_1 \leq \dots \leq q_m$ 이다. 사전처리 작업 시 각 유형별로 직접경로를 운송할 차량의 대수에는 제한이 없으나 차량경로 생성 시에는 각 차량 유형별로 사용 가능한 대수가 주어진다. 또한 각 유형 w 에 대해 차량을 한 대 사용할 때 드는 운용비용 h_w 가 주어진다. 이를 차량의 고정비용이라고 부르기로 한다. $Q_i \geq q_1$ for $i \in V \setminus \{0\}$, $Q_0 = q_m$, $h_1/q_1 \geq h_2/q_2 \geq \dots \geq h_m/q_m$ 이라고 가정한다. 또한 모든 차량의 속도는 같으며 모든 운송로에서 일정하

고 각 차량은 하나의 경로만을 운행하거나 운행하지 않는다.

(6) 서비스 속도 α 에 대한 정보.

물량을 싣거나 내리는 속도로서 ‘물량/단위시간’으로 표현된다. 단위시간은 t_{ij} 와 같은 단위를 사용한다.

위와 같은 입력정보와 가정이 주어졌을 때, VRPTD를 통해 얻고자 하는 정보는 차량경로 설정 및 차량 할당, 각 차량경로의 시간 스케줄로서 다음과 같다.

- (1) 차량 별 경유지 및 경유지에서의 수집 물량
- (2) 모든 차량에 대해 각 경유지에서 서비스를 시작하는 시각과 마치는 시각에 대한 일정계획
- (3) 차량경로 별 운송 비용 및 총 운송비용

VRPTD의 목적은 총 운송비용을 최소화하는 것으로 사용하는 차량들의 고정비용(A1의 첫 항)과 운송거리의 합인 변동비용(A1의 둘째 항)의 합으로 표시한다. 본 연구에서 VRPTD에 대한 제약은 다음과 같다. 괄호 안의 내용은 내용에 해당하는 수리 모형의 식 번호를 의미한다.

- (1) 사용하는 차량경로를 통해 모든 운송물량이 처리되어야 한다(A4~A5).
- (2) 사전처리 작업 후 남은 물량은 반드시 하나의 차량으로 운송되어야 한다(A2~A3).
- (3) 모든 운송로에서 적재되어 있는 차량의 물량은 차량의 용량을 초과하지 않아야 한다(A7).
- (4) 사전 처리 후 차량경로 생성단계에서 사용되는 차량은 각 유형 별로 사용 가능한 대수의 한계를 초과하지 않도록 한다(A6).
- (5) 각 마디를 방문하는 차량은 그 마디의 시간제약을 만족해야 한다(A9, A13).
- (6) 모든 마디에 대해 그 마디를 방문하는 차량경로들은 하역장 용량제약 조건을 만족해야 한다.

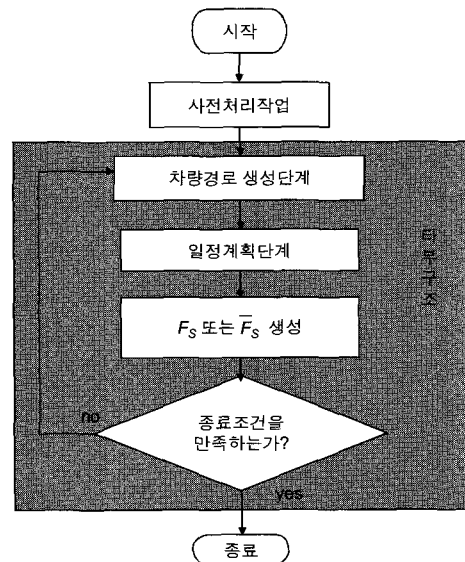
즉 어떤 마디의 서비스 가능한 모든 시간대에서 하역장을 점유하는 차량의 수는 한 대 이하여야 한다(A14~A16).

- (7) 모든 마디에 대해 마디를 방문하는 차량의 용량은 그 마디의 하역장의 용적보다 작거나 같아야 한다(A8).

1장에서 언급한 바와 같이 VRPTD는 VRPTW에 하역장 용량을 추가한 문제이다. VRPTW는 시간제약이 추가된 VRP의 일반적인(general) 형태로 NP-hard이다[4]. VRPTW는 하역장 용량이 무한대(∞)인 VRPTD와 같으므로 VRPTD는 VRPTW보다 더 일반적인 문제이고 따라서 VRPTD도 NP-hard에 속한다.

3. 알고리즘

전체 알고리즘의 흐름도는 다음과 같다.



[그림 2] 전체 알고리즘 흐름도

전체 알고리즘은 사전처리작업 단계와 타부탐색 단계로 나누어지며 타부탐색 단계는 차량경로 생성 단계와 일정계획 단계로 구성된다. 사전처리 과정

중 1장에서 설명한 물량을 분할하는 방법은 현실에서 많이 적용되고 있는 방법이다. 비용 면에서 이 방법의 효용성을 검증하는 것이 타당하나, 본 논문은 하역장 용량제약으로 인해 같은 마디를 경유하는 차량들의 일정계획이 필요하게 된 상황에 초점을 맞추고 있으므로 물량분할 과정은 이 방법을 적용하기로 한다. 차량경로 생성단계, 일정계획 단계의 내용 역시 1장에서 설명하였으며 3장에서는 타부탐색 알고리즘의 개념과 흐름에 대해 중점적으로 기술한다.

타부탐색은 Glover[11]에 의해 처음으로 소개되었고 지금까지 이론적으로 짧은 시간 안에 해를 구하기 힘든 많은 조합최적화 문제에 적용되어 왔다. 타부탐색은 메모리 기반 탐색 전략으로서 국소해에 머물지 않고 종료조건을 만족할 때까지 해를 찾는 과정을 계속한다. 그러므로 이 탐색 방법은 국소해와 같거나 더 좋은 해를 찾게 된다.

타부탐색은 현재의 해 S_c , S_c 의 이웃해 S_c' 생성 함수(neighborhood function) $f(S_c')$, 이웃해 집합 $N(S_c)$, 타부 목록의 내용, 타부 목록의 크기 H , 새로운 해의 선택방법, 타부탐색의 종료조건 등의 요소로 구성된다. 타부탐색은 S_c 에 이웃해 생성함수를 적용하여 목적함수를 개선할 수 있는 새로운 해를 찾고 이 새로운 해를 다시 S_c 로 놓고 이웃해를 생성해나가는 반복의 과정이다. 현재의 해에 이웃해 생성함수를 적용하여 구한 해의 집합을 이웃해 집합이라 부른다. 그러나 이런 방법은 여러 반복을 거치면서 이전의 해로 돌아가 순환(cycle)이 발생할 수 있다. 타부 목록은 이러한 해의 순환을 방지하기 위한 것으로 최근 H 번의 반복횟수에서 구한 해나 해의 이동(move)을 메모리에 유지한다. 이웃해 집합에서 새로운 해를 선택하는 방법에는 GB(global best)와 FB(first best)의 두 가지 방법이 있다. 목적함수가 최소화 문제의 경우 GB는 $N(S_c)$ 에 속하는 모든 이웃해를 탐색한 후 목적함수가 가장 많이 감소하거나 감소하는 해가 없는 경우 목적함수의 증가가 가장 적은 해를 선택하는 방법이다. FB방법

은 목적함수가 감소하는 해가 처음으로 생성되는 경우 그 해를 선택하고 개선되는 해가 없는 경우 목적함수의 증가가 가장 적은 해를 선택한다. 타부탐색의 종료조건은 여러 가지가 있는데 주로 현재의 반복횟수 IT_1 이 정해진 최대 반복횟수 M_1 을 넘었을 경우나 해의 개선이 없는 상태에서의 반복횟수 IT_2 가 정해진 최대 횟수 M_2 를 넘었을 경우 타부탐색을 종료한다.

타부탐색의 알고리즘에서 추가적으로 사용할 기호는 다음과 같다.

S_0	사전 처리 작업으로 생긴 직접경로들의 집합, $S_0 = \cup_{i \in V \setminus \{0\}} R_i$
$c(S_0)$	직접경로에 사용하는 차량의 운용비와 운송비용의 합
S	타부탐색의 차량경로 생성단계에서 구한 가능해
$c(S)$	S 에서 사용하는 차량의 운용비와 운송비용의 합
$N(S)$	S 의 이웃해 집합
S_b	현재까지 타부탐색의 차량경로 생성단계에서 구해진 $c(S)$ 가 가장 작은 S
F	$F = S_0 \cup S$
F_s	$S_0 \cup S_b$ 가 일정계획 단계를 거친 후 2장에서 언급한 모든 제약식을 만족할 때, $F_s = S_0 \cup S_b$
\bar{F}_s	$S_0 \cup S_b$ 가 일정계획 단계를 거친 후 하나의 제약식이라도 만족하지 못할 때, $\bar{F}_s = S_0 \cup S_b$
IT_1	현재 반복횟수
IT_2	가능해의 목적함수의 개선이 없는 상태에서의 현재 반복횟수
IT_3	일정계획이 가능하지 않은 상태에서의 반복횟수
M_1	정해진 최대 반복횟수
M_2	가능해의 목적함수의 개선이 없는 상태의 정해진 최대 반복횟수
M_3	일정계획이 가능하지 않은 상태에서의 정해진 최대 반복횟수

본 연구에서 구현한 타부탐색 알고리즘은 다음과 같다.

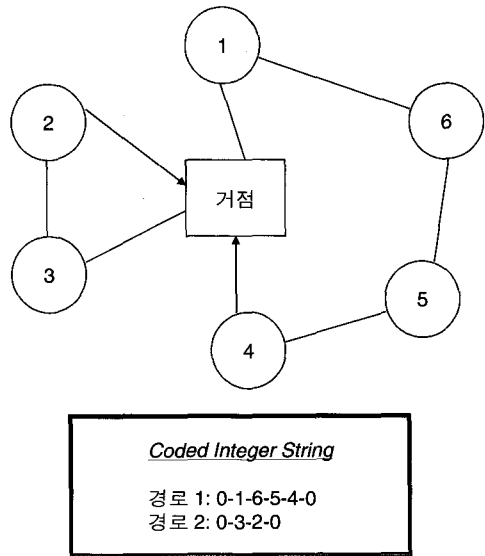
- [단계 1] 사전 처리 작업 후 남은 물량에 대해 초기 가능해 S_1 생성
- [단계 2] $S = S_1$, 일정계획 하부루틴을 부르고, $SCHED = 1$ 이면 $F_S = S_0 \cup S$, $SCHED = -1$ 이면 $\bar{F}_S = S_0 \cup S$ 로 한다.
- [단계 3] $S_b = S$, $IT_1 = 1$, $IT_2 = 0$, $IT_3 = 0$ 으로 초기화한다.
- [단계 4] 이웃 가능해 생성 하부루틴
 - [단계 4.1] S 에 속하는 모든 차량경로 쌍에 대해 2-interchange 방법을 이용하여 이웃 가능해 집합 $N(S)$ 를 생성한다.
 - [단계 4.2] GB 또는 FB에 따라 $N(S)$ 중 새로운 가능해 S' 를 선택한다. $S = S'$, $IT_1 = IT_1 + 1$ 으로 한다. 타부 목록을 수정한다.
- [단계 5] 일정계획 하부루틴을 부른다.
- [단계 6] $c(S) < c(S_b)$ 이고 $SCHED = 1$ 이면 $S_b = S$, $F_S = S_0 \cup S$, $IT_2 = 0$ 으로 한다. $c(S) < c(S_b)$ 이고 $SCHED = -1$ 이면 $S_b = S$, $\bar{F}_S = S_0 \cup S$, $IT_2 = 0$ 으로 한다. 그 외의 경우 $IT_2 = IT_2 + 1$ 로 한다.
- [단계 7] $IT_1 > M_1$ 이거나 $IT_2 > M_2$ 이거나 $IT_3 > M_3$ 이 아니면 [단계 4]로 간다.
- [단계 8] 알고리즘을 종료하고 F_S 와 \bar{F}_S 를 최종 일정 가능해와 최종 일정 불가능해로 한다.

[알고리즘] 타부탐색

[단계 1]과 [단계 4]는 차량경로 생성단계에 해당하며 사전처리 작업 후 남은 물량들에 대해 서비스할 차량경로를 정하여 주는 단계이다. 차량경로 생성단계에서는 2장에서 언급한 제약 중 (6)번만을 제외하고 나머지 모든 제약을 만족하는 차량경로집합을 생성한다. [단계 1]은 초기 가능해 생성단계로서 초기 가능해를 생성하기 위해 Solomon[15]이 제안한 Clarke and Wright[8]의 비용절약(Saving) 휴리스틱의 변형된 알고리즘을 사용한다. 비용절약 휴리스틱은 거점과 고객을 바로 연결하는 n 개의 직접경로로부터 시작하여 임의의 두 개의 차량경로를 합칠 경우 필요한 제약조건을 만족하면서 비용이 절약되면 두 개의 경로를 합쳐 새로운 차량경로를

만드는 과정을 반복하는 알고리즘이다.

[단계 4]는 이웃해 집합을 생성하여 새로운 가능해를 찾는 단계이다. 이웃해 생성함수 $f(S)$ 로는 λ -interchange[17] 기법을 사용하며 λ 값으로 2를 사용한다. $S = \{U_1, \dots, U_k, \dots, U_k', \dots, U_N\}$ 이 주어졌을 때, U_k 는 차량 k 를 사용하는 경로가 방문하는 고객마디의 집합이다. U_k 와 U_k' 사이에 2-interchange를 적용할 경우, 크기가 2보다 작거나 같은 U_k 의 부분집합 \tilde{U}_k 와 U_k' 의 부분집합 \tilde{U}_k' 를 서로 교환하여 새로운 이웃해 $S = \{U_1, \dots, U_k', \dots, U_k, \dots, U_N\}$ 를 생성한다. S 의 차량경로들 중 형상이 바뀐 경로들의 바뀌기 직전의 형상을 코딩된 정수 문자열(Coded integer string)로 표현하여 타부 목록에 저장하고 H 번의 반복회수(tabu list size) 후에 타부 목록에서 제외해야 한다. 형상을 저장하는 방법을 예로 설명하면 다음 그림과 같다.



[그림 3] 경로형상의 저장

이웃 가능해 중 타부 목록에 있는 차량경로의 형상과 같은 형상을 가지는 가능해는 타부가 되고 S' 후보에서 제외된다. 그러나 순환을 방지하기 위한 타부 제약은 현재의 해가 더 좋은 해로 움직이는

것을 방해할 수 있다. 이러한 한계를 극복하기 위해 열망기준(aspiration criteria)을 둔다. 즉 이웃 가능해가 타부 조건에 해당되어도 지금까지 발견된 가장 좋은 가능해보다 더 좋은 경우 타부 상태를 무시하고 새로운 가능해 S' 로 받아들여 순환을 방지하면서 더 좋은 가능해를 찾는 것을 가능하게 한다.

[단계 2]와 [단계 5]는 일정계획 단계에 해당하며 일정계획 하부부터 알고리즘은 Su et al.[16]의 세 가지의 우선순위 규칙을 이용한 알고리즘을 인용하였다. 이 알고리즘은 시기(stage)로 정의되며, 각 시기에서는 각 마디에서 작업할 한 대 이하의 차량을 정하게 되는데 작업할 차량이 두 대 이상인 경우 네 개의 우선순위 규칙에 의해 하나의 차량을 선택한다. 네 개의 우선순위 규칙은 Su et al.[16]의 세 가지의 우선순위 규칙에 각 마디에서의 서비스 허용 시간대 제약을 고려하기 위해 하나의 규칙을 추가하여 만든 규칙들이다. 이 규칙들은 마디에서의 비작업시간(idle time)을 가능한 줄이도록 일정을 계획할 수 있게 해준다. 하부부터 알고리즘의 자세한 절차는 부록 B와 Su et al.[16]의 논문을 참조하기 바란다.

[단계 2]와 [단계 6]의 *SCHED*는 F 가 하역장 용량 조건을 만족하도록 일정계획을 세웠을 때 각 마디의 허용 시간대 안에서 모든 서비스가 이루어질 수 있는 지를 나타내는 지표이다. $SCHED=1$ 은 F 의 일정계획이 모든 제약조건을 만족함을 의미하고 $SCHED=-1$ 은 그렇지 못함을 의미한다. 본 타부 알고리즘은 문제를 차량경로 생성단계와 일정계획 단계로 나누어 해결하는 방안을 취하고 있어 어떤 예제에서는 일정 가능해가 생성되지 않을 수 있다. 즉 차량경로 생성단계에서 각 경유지 마디에서 허용 시간대 제약을 만족했던 차량들이 일정계획 단계에서 같은 마디를 경유하는 다른 차량들로 인해 마디에서의 대기시간이 증가함으로써 시간 제약을 만족할 수 없는 경우가 발생한다. 그 경우에 예제가 원래 일정 가능해를 가질 수 있는 문제(이하 가능예제)이지만 알고리즘의 정밀성 부족으로 인해 일정 가능해를 생성하지 못한 것인지 확인해야 하

나 가능예제인지의 여부를 판별하는 문제 또한 다항시간 안에 해결할 수 없다. 따라서 본 알고리즘에서는 일정 가능해와 일정 불가능해를 따로 저장하여 종료 후 일정 가능해를 구하였는지의 여부를 확인한다.

타부탐색의 [단계 7]에서는 알고리즘 종료조건을 확인한다. 첫 번째와 두 번째 조건에 대해서는 앞에서 언급하였으며 마지막 조건은 S 가 연달아 M_3 번 보다 크게 일정 불가능해인 상태를 유지하면 알고리즘을 종료한다는 것을 의미한다.

4. 실험결과 및 분석

본 논문에서 제시한 알고리즘은 C언어로 코딩화하였으며, Pentium III PC(800MHz)에서 실행되었다. 본 논문에서 사용한 타부탐색 알고리즘에서는 여러 번의 테스트를 거쳐 보다 효과적인 해를 제시할 수 있도록 파라미터 값을 정하였다. 지역해에 머물러 순환이 발생하는 것을 방지하기 위해 타부목록을 배열로 만들고 그 배열의 크기를 바꿔가면서 테스트를 한 결과 $H=8$ 로 정하였고, $M_1=5000$, $M_2=3000$, $M_3=1000$ 으로 하였다.

본 논문에서는 예제 문제를 (1) 고객의 수 n (10, 20 또는 50), (2) 차량 유형의 수 m , (3) 허용시간대의 크기와 서비스 시간의 평균비율 r (비작업시간 상대적으로 작음(tight, T) 또는 비작업시간 상대적으로 큼(loose, L)), (4) 같은 고객을 방문하는 차량의 최대 대수 R (S(1대), M(2대 이상))에 따라 분류한다.

<표 1>은 고객의 수가 10인 경우의 문제의 특성을 나타낸다. 고객의 수가 20, 50인 경우의 예제도 고객의 수가 10인 경우와 같은 형태를 가지며 문제 번호 11~20은 고객의 수가 20인 경우, 문제번호가 21~30은 고객의 수가 50인 경우이다. 본 연구에서 사용한 테스트 예제는 Solomon[15]의 예제 R1, R2를 바탕으로 만들었으며 차량 한대 이상의 요구물량을 갖는 고객을 형성하기 위해 1~5 사이의 난수

(Random Number)를 발생시켜 Solomon 예제의 고객의 요구물량에 곱하였다. 모든 예제의 고객의 도착가능시각과 도착 한계 시각은 각각 다르나 허용시간대의 절대적인 크기는 모두 같다. 그러나 고객마다 요구물량이 다르고 서비스 속도는 일정하므로 결과적으로 허용시간대의 상대적인 크기는 다르다고 할 수 있다. 고객 i 의 총 요구물량을 D_i 라고 할 때 허용시간대의 크기가 loose한 경우는 D_i/α 값의 평균이 허용시간대 크기의 50% 이하가 되도록, 한 tight한 경우는 60%~70% 사이가 되도록 서비스속도 α 를 조정한 경우이다. 고객을 방문하는 차량의 최대 대수가 1인 경우는 모든 고객이 하나의 차량 경로에만 포함이 되므로 이 경우는 거점의 하역장 용량만을 고려한다.

<표 1> 테스트문제

문제번호	n	m	r	R
1	10	1	T	S
2	10	1	L	S
3	10	1	L	S
4	10	3	T	S
5	10	3	L	S
6	10	1	T	M
7	10	1	L	M
8	10	3	L	M
9	10	3	T	M
10	10	3	L	M

본 논문에서 구현한 타부탐색 알고리즘의 성능을 평가하기 위해 부록 A의 VRPTD 수리모형을 이용하여 CPLEX 7.0으로 최적해를 구한 후 타부탐색의 결과와 비교하였다. <표 2>는 실험 결과를 나타내며 Z^* 는 3장의 수리모형에 의한 최적해의 목적함수 값과 $c(S_0)$ 의 합을 의미한다. $Z(F_s)$ 는 $c(F_s)$ 를, $Z(\bar{F}_s)$ 는 $c(\bar{F}_s)$ 를 의미한다. '수행시간 1'은 본 논문에서 구현한 알고리즘의 수행 시간이며, '수행시

간 2'는 CPLEX의 수행 시간이다. $GAP1$ 과 $GAP2$ 는 다음과 같이 계산한다.

$$GAP1 = 100\{Z(F_s) - Z^*\}/Z^*(\%),$$

$$GAP2 = 100\{Z(F_s) - Z(\bar{F}_s)\}/Z(\bar{F}_s)(\%)$$

본 논문에서 구현한 타부탐색 알고리즘은 모든 문제에 대해 일정 가능해를 생성하였으며 평균 $GAP1$ 이 3.5%로써 좋은 해를 제공하였다. 고객의 수가 20인 경우이고 $R=M$ 인 경우와 고객의 수가 50인 경우에 대해서는 최적해를 구하는 것이 불가능하여 구해진 해의 품질을 비교할 수 없었다. 그러나 최적해를 구한 예제의 경우와 같이 그 결과가 나쁘지 않을 것으로 예상된다. 초기 가능해와 타부탐색을 종료한 후 얻어진 일정 가능해를 비교해본 결과 고객의 수가 10일 경우 평균 6.8%의 개선이 있었고, 20인 경우는 21.28%, 50인 경우는 45.64%의 해의 개선이 있었다.

$GAP2$ 는 본 논문에서 제시한 VRPTD 문제의 필요성을 설명하기 위한 것으로 현실에서 하역장 용량제약이 존재하나 이를 무시하고 VRPTW로 풀었을 경우의 해와 하역장 용량제약을 고려하였을 때의 해의 품질을 비교하였다. 그 결과 고객의 수가 10인 경우는 평균 $GAP2$ 이 1.78%, 20인 경우 4.0%, 50인 경우 8.4%였고 대부분의 문제에서 VRPTW의 해와 VRPTD의 해가 같지 않았다. 따라서 본 논문에서 제시한 VRPTD가 현실적으로 의미가 있으며 현실 문제에서 하역장 용량제약이 존재할 경우 이를 고려한 모형이 바람직하다고 할 수 있다.

본 알고리즘의 평균 수행시간은 마디의 수가 10인 경우 14.8초, 20인 경우 31.8초, 50인 경우 115.9초였다. 마디의 수가 10인 예제에 대해서는 CPLEX가 더 빨리 해를 제공 하였지만 20인 경우부터는 본 알고리즘에 비해 CPLEX의 수행시간이 급격히 증가하는 것을 볼 수 있다. 이런 결과들을 종합해 볼 때 본 연구에서 구현한 알고리즘은 풀기 어려운 VRPTD에 대해 빠른 시간 안에 좋은 해를 제공할 것으로 판단할 수 있다.

〈표 2〉 테스트 결과

문제 번호	Z^*	$Z(F_s)$	$Z(\bar{F}_s)$	GAP1	GAP2	수행 시간1	수행 시간2
1	316	316	316	0.00	0.00	13.8	2.7
2	316	316	316	0.00	0.00	21.0	3.1
3	316	316	316	0.00	0.00	19.0	3.0
4	238	261	252	9.66	3.57	13.2	3.9
5	238	248	248	4.20	0.00	15.0	4.7
6	681	693	693	1.76	0.00	11.0	4.6
7	681	693	693	1.76	0.00	10.7	5.3
8	596	614	581	3.02	5.68	13.0	6.4
9	596	614	581	3.02	5.68	19.0	6.4
10	716	722	702	0.84	2.85	12.0	9.0
11	583	611	588	4.80	3.91	31.0	4927.0
12	520	545	525	4.81	3.81	39.2	5017.7
13	520	545	525	4.81	3.81	39.0	5103.2
14	493	551	535	11.76	2.99	27.5	5933.6
15	493	504	504	2.23	0.00	27.3	6638.3
16	*	1399	1329	*	5.27	41.0	*
17	*	1401	1352	*	5.98	29.0	*
18	*	1905	1838	*	9.61	28.3	*
19	*	1920	1894	*	1.37	29.7	*
20	*	1903	1838	*	3.54	26.4	*
21	*	1241	1186	*	26.12	65.3	*
22	*	1092	1080	*	13.99	143.5	*
23	*	1107	1089	*	11.26	143.0	*
24	*	1158	1135	*	11.88	58.6	*
25	*	1070	1059	*	3.98	132.5	*
26	*	2552	2538	*	9.15	97.7	*
27	*	2483	2483	*	0.00	101.3	*
28	*	1850	1850	*	0.00	144.0	*
29	*	1996	1955	*	7.02	129.0	*
30	*	1873	1856	*	0.97	143.9	*

5. 결 론

본 논문은 시간제약과 하역장 용량제약이 있는 차량경로문제(VRPTD)를 대상으로 하며 VRPTD는 총 운송비용을 최소화하면서 현실에서 발생할 수 있는 다양한 제약조건을 만족해야 한다.

고객의 수가 10, 20, 50인 문제에 대해 타부 탐색 과정에서 GB와 FB의 두 가지 방법을 적용하여 실행하였다. 고객 수가 10, 20인 경우에 한해 3장에서 제시한 수리모형으로 최적해를 비교한 결과 타부탐색은 평균 GAP이 4%이내인 좋은 해를 제공하였다. 또한 본 연구의 결과는 보다 더 현실적인 상황

을 고려한 VRPTD 문제 제기가 타당성이 있음을 보여주었다. 본 연구에서 제시한 알고리즘은 저자가 시도한 모든 예제에서 일정 가능해를 생성하여 우수한 성능을 보였다.

그러나 본 연구에서 제시한 알고리즘은 일정 가능해를 생성하지 못할 경우 본래 문제가 가능 문제인지의 여부를 판단할 수 없다는 단점이 있으므로, 추후 이 문제점을 해결 할 수 있는 방안에 대한 연구가 필요하다. 또한 본 논문에서는 모든 마디의 하역장 용량을 1이라고 가정하였으나, 마디마다 하역장 용량이 다른 문제에 대한 수리모형과 해법도 연구되어야 할 것이다.

참 고 문 헌

- [1] 서병규, 김종수, "자연 비용을 고려한 서비스 시간대가 존재하는 외판원 문제에 대한 발견적 해법", 『대한산업공학회지』, 제27권, 제1호(2001), pp.18-24.
- [2] Adams, J., E. Balas, and D. Zawack, "The Shifting Bottleneck Procedure for Job-Shop Scheduling," *Management Science*, Vol.34, No.3(1988), pp.391-401.
- [3] Baker, K.R. and G.B. McMahon, "Scheduling the General Job-Shop," *Management Science*, Vol.31, No.5(1985), pp.594-598.
- [4] Ball, M.O., T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, *Network Routing*, North-holland, Amsterdam, 1995.
- [5] Beasley, J.E., M. Krishnamoorthy, Y.M. Sharaiha, and D.A. Bramson, "Scheduling Aircraft Landings The Static Case," *Transportation Science*, Vol.34, No.2(2000), pp. 180-197.
- [6] Bodin, L. and B. Golden, "Classification in Vehicle Routing and Scheduling," *Networks*, Vol.11(1981), pp.97-108.
- [7] Brandao, J. and A. Mercer, "A Tabu Search Algorithm for the Multi-Trip Vehicle Routing and Scheduling Problem," *Euro. J. of Operational Res.*, Vol.100, No.1(1997), pp. 180-191.
- [8] Clarke, G. and J.W. Wright, "Scheduling of Vehicles from A Central Depot to A Number of Delivery Points," *Operations Res.* Vol.12, No.4(1964), pp.568-581.
- [9] Fisher, M.L., B.J. Lageweg, J.K. Lenstra, and A.H.G. Rinnooy Kan, "Surrogate Duality Relaxation for Job-Shop Scheduling," *Discrete Applied Mathematics*, Vol.5, No.1 (1983), pp.65-75.
- [10] Garey, M.R. and D.S. Johnson, *Computers and Intractability*, W.H. Freeman and Co., San Francisco, 1979.
- [11] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence," *Comput. Operations Res.* Vol.13, No.5(1986), pp.533-549.
- [12] He, Z., T. Yang, and A. Tiger, "An Exchange Heuristic Imbedded with Simulated Annealing for Due-Dates Job-Shop Scheduling," *Euro. J. of Operational Res.* Vol.91, No.1 (1994), pp.99-117.
- [13] Kim, Y.D., "A Comparison of Dispatching Rules for Job with Multiple Identical Jobs and Alternative Routings," *Int. J. Prod. Res.*, Vol.28(1990), pp.953-962.
- [14] Koskosidis, Y.A., W.B. Powell, and M.M. Solomon, "An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints," *Transportation Science*, Vol.26, No.2(1992), pp.69-85.
- [15] Solomon, M.M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Res.*, Vol.35, No.2(1987), pp.254-265.

- [16] Su, L.H., P.C. Chang, and E.S. Lee, "A Heuristic for Scheduling General Job Shops to Minimize Maximum Lateness," *Mathl. Comput. Modeling*, Vol.27, No.1(1998), pp. 1-15.
- [17] Tan, K.C., L.H. Lee, Q.L. Zhu, and K. Ou, "Heuristic Methods for Vehicle Routing Problem with Time Windows," *Artificial Intelligence in Engineering*, Vol.15, No.3 (2001), pp.281-295.

부록 A : 수리모형

제시할 수리모형은 사전처리 작업 후 고객의 물량을 차량의 용량보다 작거나 같게 분할하고 직접경로들을 생성한 후의 상황에 대한 모형이다. 우선 추가적으로 사용할 기호는 아래와 같다.

- G 운송망 그래프, $G=(V, E)$
- R_i 사전처리 작업으로 생긴 마디 $i \in V \setminus \{0\}$ 와 마디 0사이의 직접경로 집합, $R_i = \{R_{i1}, R_{i2}, \dots, R_{i|R_i|}\}$
- K_w 차량 유형이 w 이고 직접경로에 쓰이지 않은 차량의 집합, $w=1, 2, \dots, m$
- \overline{W}_i 마디 $i \in V \setminus \{0\}$ 에 대해 직접경로로 운행하는 차량들의 집합
- W 차량경로 생성단계에서 사용될 수 있는 모든 차량의 집합, $W = \bigcup_{w=1}^m K_w$
- vh_{ib} 직접경로 R_{ib} 에 사용되는 차량, $\overline{W}_i = \{vh_{i1}, \dots, vh_{i|R_i|}\}$, $i \in V \setminus \{0\}$
- \overline{W}_0 직접경로에 사용되는 모든 차량의 집합, $\overline{W}_0 = \bigcup_{i \in V \setminus \{0\}} \overline{W}_i$
- d_{ib} $b = \{0, 1, \dots, |R_i|\}$ for $i \in V \setminus \{0\}$, $b=0$ 이면 마디 $i \in V \setminus \{0\}$ 의 사전처리 작업 후 남은 물량, $b \neq 0$ 이면 차량 vh_{ib} 가 수집하는 물량
- s_{ib} 마디 i 에서 물량 d_{ib} 를 차량에 적재하는 데 걸리는 시간, $s_{ib} = d_{ib}/\alpha$, $b = \{0, 1, \dots, |R_i|\}$ for $i \in V \setminus \{0\}$

모형에서 사용하는 결정변수의 의미는 다음과 같다.

- x_{ij}^k 차량 $k \in W$ 의 경로구간을 나타내는 변수, 차량 k 가 마디 i 에서 j 로 갈 때 1, 그렇지 않은 경우 0. 단, $(i, j) \in E$
- $z_{kk'}$ 마디 $i \in V$ 에서 차량 k 가 차량 k' 보다 먼저 서비스를 시작할 때 1, 그렇지 않은 경우 0, $k \neq k'$, $k, k' \in W \cup \overline{W}_i$
- y_{ij} 사전처리 후 남은 물량을 수집할 때 마디 $i \in V$ 에서 마디 $j \in V$ 로 갈 때 차량에 적재되어 있는 물량, $(i, j) \in E$
- y_0^k 차량 $k \in W \cup \overline{W}_0$ 가 거점으로 돌아올 때 차량에 적재되어 있는 물량
- T_{ik} 차량 $k \in W \cup \overline{W}_i$ 가 마디 $i \in V$ 에서 서비스를 시작하는 시각

위의 기호들을 이용한 VRPTD 정수계획 모형은 다음과 같다.

$$\begin{aligned}
 \text{Min} \quad & \sum_{w=1}^m \sum_{k \in K_w} \sum_{j \in V \setminus \{0\}} h_w x_{0j}^k + \sum_{k \in W} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}^k & A1 \\
 \text{s.t.} \quad & \sum_{k \in W} \sum_{j \in V} x_{ij}^k = 1 & \text{for } i \in V \setminus \{0\} & A2 \\
 & \sum_{\{j|(i,j) \in E\}} x_{ij}^k - \sum_{\{j|(j,i) \in E\}} x_{ji}^k = 0 & \text{for } i \in V, k \in W & A3 \\
 & \sum_{\{j|(i,j) \in E\}} y_{ij} - \sum_{\{j|(j,i) \in E\}} y_{ji} = d_i & \text{for } i \in V \setminus \{0\} & A4
 \end{aligned}$$

$$\begin{aligned} \sum_{i \in V \setminus \{0\}} y_{0i} &= 0 & \text{A5} \\ \sum_{k \in K_w} \sum_{i \in V \setminus \{0\}} x_{0i}^k &\leq |K_w| & \text{for } w=1, \dots, m & \text{A6} \\ y_{ij} &\leq \sum_{k \in K_w} x_{ij}^k q_w & \text{for } (i, j) \in E, w=1, \dots, m & \text{A7} \\ \sum_{k \in K_w} \sum_{(j|(i, j) \in E)} x_{ij}^k q_w &\leq Q_i & \text{for } i \in V \setminus \{0\}, w=1, \dots, m & \text{A8} \\ \sum_{(j|(i, j) \in E)} e_i x_{ij}^k &\leq T_{ik} \leq \sum_{(j|(i, j) \in E)} l_i x_{ij}^k & \text{for } i \in V, k \in W & \text{A9} \\ T_{ik} + s_{j0} + t_{ij} - T_{jk} &\leq (1 - x_{ij}^k) M & \text{for } (i, j) \in E, k \in W & \text{A10} \\ y_0^k &= \sum_{j \in V \setminus \{0\}} \sum_{(j|(i, j) \in E)} x_{ij}^k d_{j0} & \text{for } k \in W & \text{A11} \\ y_0^k &= d_{ib} & \text{for } i \in V \setminus \{0\}, k = vh_{ib} \in \bar{W}_i & \text{A12} \\ e_i &\leq T_{ik} \leq l_i & \text{for } i \in V, k \in \bar{W}_i & \text{A13} \\ T_{ik} + s_{ib} - T_{ik'} &\leq (1 - z_{kk'}^i) M & \text{for } b = \{0, 1, \dots, |R_i|\}, i \in V \setminus \{0\}, \\ & & \text{If } b=0, k \in W \text{ \& } k' \in \bar{W}_i. \text{ Otherwise,} & \\ & & k = vh_{ib} \text{ \& } k' \in W \cup \bar{W}_i \setminus \{vh_{ib}\} & \text{A14} \\ T_{0k} + y_0^k / \alpha - T_{0k'} &\leq (1 - z_{kk'}^0) M & \text{for } k, k' \in W \cup \bar{W}_0 & \text{A15} \\ z_{kk'}^i + z_{k'k}^i &= 1 & \text{for } i \in V, k \neq k', k, k' \in W \cup \bar{W}_i & \text{A16} \\ z_{kk'}^i + z_{k'k}^i &\leq \sum_{j|(i, j) \in E} x_{ji}^k & \text{for } i \in V, k \in W, k' \in \bar{W}_i & \text{A17} \\ x_{ij}^k &\in \{0, 1\} & \text{for } (i, j) \in E, k \in W & \text{A18} \\ z_{kk'}^i &\in \{0, 1\} & \text{for } i \in V, k \neq k', k, k' \in W \cup \bar{W}_i & \text{A19} \\ y_{ij}, y_0^k, T_{ik} &\geq 0 & \text{for } i \in V, (i, j) \in E, k \in W \cup \bar{W}_i & \text{A20} \end{aligned}$$

M is a very large number.

제약식 A2~A10은 사전처리 후 남은 물량에 대한 차량경로 형성을 의미한다. 식 A10은 같은 차량이 방문하는 마디 간의 서비스 시작시각에 대한 적합성을 의미한다. 식 A11과 식 A12는 변수들 간의 논리 제약식으로서 식 A11은 사전처리 후 남은 물량을 수거한 차량에 대해, 식 A12는 직접경로에 쓰이는 차량에 대해서 차량이 거점으로 돌아올 때 차량에 적재되어 있는 물량에 대한 식이다. $Z_{kk'}^i$ 는 하역장 용량제약을 표현하기 위해 본 VRPTD 모형에서 추가한 변수이며 이 변수와 관련된 식 A14~A16은 하역장 용량 제약, 식 A17은 이 변수와 다른 변수와의 논리적 적합성을 의미한다. 식 A14는 거점을 제외한 마디에 대해, 식 A15는 거점에 대해 각 마디를 지나가는 차량들의 우선 순위에 따른 작업 시작 시각에 대한 연관관계를 나타낸다. 식 A16은 같은 마디를 지나가는 임의의 두 대의 차량 중 어느 차량이 우선권을 갖는지 결정되어야 함을 의미한다. 식 A18과 식 A19는 해당 변수가 이진 변수임을, 식 A20은 비음 실수 변수임을 의미한다.

위의 수리모형에서 $Z_{kk'}^i$ 변수는 두 차량 간의 우선순위만을 의미하므로, 하역장 용량이 1보다 큰 경우 이 변수를 이용하여 하역장 용량제약을 표현할 수 없다. 이 경우 마디의 허용시간대를 단위시간으로 나누고 각 단위시간에 작업하는 차량이 마디의 하역장 용량보다 작거나 같다는 식을 도입하여 해결할 수 있다.

부록 B : 일정계획 하부루틴(F 에 대해 일정계획을 수립한다)

일정계획 하부루틴에서 추가로 사용할 기호는 다음과 같다.

a Stage를 의미하는 지표

AN_k 차량 $k \in W \cup \overline{W}_0$ 의 출발지인 거점을 제외하고 k 가 경유하는 운송로와 마디를 워크(walk)로 표현한 집합이다. $AN_k = \{we_{k1}, wv_{k1}, \dots, we_{k, 2|AN_k|}, 0\}$ 으로 we_{kj} 는 k 가 j 번째로 방문하는 운송로를, wv_{kj} 는 마디를 의미한다. 마지막 원소(element)는 거점이다

s_{ik} 차량 $k \in W \cup \overline{W}_i$ 가 마디 i 에서 소요하는 서비스 시간, $k \in W$ 이면 $s_{ik} = s_{k0}$, $k = v_{h_{ib}} \in \overline{W}_i$ 이면 $s_{ik} = s_{ib}$

ar_k^i 마디 i 를 방문하는 차량 $k \in W \cup \overline{W}_i$ 가 마디 i 에서 서비스를 시작할 수 있는 가장 빠른 시각. $i = wv_{k1}$ 에 대해 $ar_k^i = \max\{e_i, t_{0i}\}$, $i = wv_{k,j}$ 이고 $i' = wv_{k,j+1}$ 에 대해 $ar_k^i = \max\{e_i, ar_k^i + t_{ii'} + sk_{ik}\}$

p_k 차량 $k \in W \cup \overline{W}_0$ 가 모든 경유지의 허용 시간대를 지키면서 거점으로 돌아오는 시각을 지연시킬 수 있는 시간의 최대값, $p_k = l_0 - ar_k^0$

SN_k^a Stage a 일 때, 차량 $k \in W \cup \overline{W}_0$ 가 Stage $(a-1)$ 까지 서비스한 마지막 마디가 wv_{kj} 이면 차량경로가 $SN_k^a = AN_k - \{we_{k1}, wv_{k1}, \dots, wv_{kj}, we_{k,j+1}\}$

g_k^a Stage a 일 때, 차량 $k \in W \cup \overline{W}_0$ 의 SN_k^a 의 첫 번째 원소이며 만약 $SN_k^a = \emptyset$ 이면 $g_k^a = -1$ 로 둔다

VS_i^a Stage a 일 때, 마디 i 를 방문할 차례에 있는 차량 집합, $g_k^a = i$ 이면 $k \in VS_i^a$

sv_i^a Stage a 일 때, 마디 i 에서 선정된 차량, $sv_i^a \in VS_i^a$

T_k^a Stage a 일 때, 차량 $k \in \cup_{i \in V} VS_i^a$ 가 g_k^a 에서 서비스를 시작할 수 있는 최초 가능 시각

B_k^a Stage a 일 때, 차량 $k \in W \cup \overline{W}_0$ 가 아직 방문하지 않은 마디를 방문하기 위한 이동시간과 서비스 작업 시간의 합, $B_k^a = \sum_{(i,j) \in SN_k^a} t_{ij} + \sum_{i \in SN_k^a} s_{ik}$

A_i^a Stage a 일 때, 마디 i 의 하역장 이용이 가능한 시각, $A_i^{a+1} = \max\{e_i, T_{sv_i^a}^a\} + s_{ik}$ for $k = sv_i^a$

p_k^a Stage a 일 때, 차량 $k \in \cup_{i \in V} VS_i^a$ 가 서비스 시작 시각을 지연시킬 수 있는 최대 시간의 양. $k = sv_i^a$ 이면 $p_k^{a+1} = p_k^a - (T_k^a - T_{ki})$, 그렇지 않은 경우 $p_k^{a+1} = p_k^a - (T_k^{a+1} - T_k^a)$

절차는 다음과 같다.

[단계 S.1] $SN_k^1 = AN_k$ 로 두며 모든 $i \in V$ 와 $k \in W \cup \overline{W}_0$ 쌍에 대해 $g_k^1 = i$ 이면 VS_i^1 에 k 를 원소로 넣는다.

[단계 S.2] $k \in \cup_{i \in V} VS_i^1$ 에 대해 $p_k^1 = p_k$, $B_k^1 = \sum_{(i,j) \in SN_k^1} t_{ij} + \sum_{i \in SN_k^1} s_{ik}$, $T_k^1 = ar_k^i$ for $i = wv_{k1}$ 으로 둔다. $i \in V$ 에 대해 $A_i^1 = \max\{e_i, t_{0i}\}$ 으로 두고 $a=1$ 로 한다.

[단계 S.3] 모든 $k \in W \cup \overline{W}$ 에 대해 $g_k^1 = -1$ 이면 [단계 S.8]로 간다. 그렇지 않은 경우 [단계 S.4]로 간다.

[단계 S.4] $VS_i^a \neq \emptyset$ 인 마디의 하역장에서 서비스를 수행할 차량을 선정한다.

case 1 : $VS_i^a = \{k\}$ 인 경우, $sv_i^a = k$ 로 하고 [단계 S.5]으로 간다.

case 2 : $|VS_i^a| \geq 2$ 인 경우, 다음의 4가지 우선권 규칙에 의하여 차량을 선정한다. 마디 i 를 방문할 차례인 모든 차량의 쌍 k 와 k' 에 대해,

- 규칙 1(시간제약 규칙) : $p_k^a = 0$, $k \in VS_i^a$ 인 k 가 가장 우선한다. 만약 그런 차량이 두 대 이상이면 [단계 S.9]로 가고 그런 차량이 존재하지 않으면 규칙 2, 3, 4 중 해당하는 규칙을 적용한다.
- 규칙 2 : $T_k^a \leq T_{k'}^a$ 이고 $A_i^a < T_{k'}^a$ 인 경우 적용한다. (1) $s_{ik'} + B_{k'}^a > B_k^a$ 이고 $T_k^a + s_{ik} > T_{k'}^a$ 이고 $T_k^a + s_{ik} + B_k^a > T_{k'}^a + s_{ik'} + B_{k'}^a$ 를 만족하거나 (2) $s_{ik'} + B_{k'}^a < B_k^a$ 이고 $T_k^a + s_{ik} > T_{k'}^a$ 를 만족하면, k' 가 먼저 서비스 할 우선권을 갖고 그렇지 않으면 k 가 우선권을 갖는다.
- 규칙 3 : $T_k^a < T_{k'}^a$ 이고 $T_k^a \leq A_i^a \leq T_{k'}^a$ 를 만족하거나 $T_k^a \leq T_{k'}^a$ 이고 $A_i^a \leq T_{k'}^a$ 를 만족하는 경우 적용한다. (1) $s_{ik'} + B_{k'}^a > B_k^a$ 이고 $A_i^a + s_{ik} > T_{k'}^a$ 이고 $A_i^a + s_{ik} + B_k^a > T_{k'}^a + s_{ik'} + B_{k'}^a$ 를 만족하거나 (2) $s_{ik'} + B_{k'}^a < B_k^a$ 이고 $A_i^a + s_{ik} > T_{k'}^a$ 를 만족하면, k' 가 먼저 서비스 할 우선권을 갖고 그렇지 않으면 k 가 우선권을 갖는다.
- 규칙 4 : $T_k^a < T_{k'}^a$ 이고 $A_i^a < T_{k'}^a$ 를 만족하거나 $T_k^a = T_{k'}^a$ 이고 $A_i^a \leq T_{k'}^a$ 를 만족하는 경우 적용한다. (1) $B_{k'}^a > s_{ik} + B_k^a$ 를 만족하거나 (2) $B_k^a < s_{ik} + B_{k'}^a$ 이고 $s_{ik} + B_{k'}^a > s_{ik'} + B_k^a$ 를 만족하면, k 가 먼저 서비스 할 우선권을 갖고 그렇지 않으면 k' 가 우선권을 갖는다.

위의 규칙에 의해 우선권이 가장 높은 차량을 sv_i^a 로 두고 [단계 S.5]으로 간다.

[단계 S.5] 차량 $k \in W \cup \overline{W}_0$ 에 대해 SN_k^{a+1} , B_k^{a+1} , p_k^{a+1} 를 수정하고, 마디 $i \in V$ 에 대해 A_i^{a+1} , VS_i^{a+1} 를 수정한다. $k \in VS_i^a$ 이고 $k = sv_i^a$ 이면, $T_{ik} = T_{ko}^a$, $T_k^{a+1} = \max\{T_k^a + s_{ik} + t_{i'}$, $A_i^{a+1} + t_{i'} = g_k^{a+1}\}$ 으로 한다. $k \in VS_i^a$ 이고 $k \neq sv_i^a$ 이면 $T_k^{a+1} = \max\{T_k^a, A_i^{a+1}\}$ 으로 한다. [단계 S.6]로 간다.

[단계 S.6] 모든 i 에 대해 $T_k^{a+1} > l_i$ 인 차량 $k \in VS_i^{a+1}$ 이 존재하지 않으면 [단계 S.7]로 가고 그렇지 않은 경우 [단계 S.9]으로 간다.

[단계 S.7] $a = a+1$ 로 하고 [단계 5.3]로 간다.

[단계 S.8] $SCHED = 1$, $IT_3 = 0$ 으로 하고 알고리즘을 종료한다.

[단계 S.9] $SCHED = -1$, $IT_3 = IT_3 + 1$ 로 하고 알고리즘을 종료한다

일정계획 알고리즘은 모든 차량이 모든 고객마디에서 서비스를 시작할 수 있는 가장 빠른 시각 T_k^a 을 ar_k^i for $i = wv_k$ 으로 둔 상태에서 시작한다. 각 stage의 [단계 S.4]에서 어떤 마디를 방문하는 차량이 두 대 이상일 경우 4가지의 우선권 규칙에 의해 하나의 차량을 선택한다. 일정계획 알고리즘의 규칙 2, 3, 4는 Su et al.[16]의 Three Dominance Rules과 같다. 그러나 이들만 고려할 경우 고객마디에서의 시간제약에 대한 조건은 검사할 수 없다. 따라서 본 연구에서는 이를 위해 규칙 1을 추가하였다. [단계 S.5]에서는 선택되지 않은 차량들에 대해 다음 stage에서 서비스를 시작할 수 있는 가장 빠른 시각 T_k^a 을 뒤로 미루게 된다. 이 때 각 차량경로의 서비스 시작시각이 미뤄질 수 있는 최대 한계인 p_k^a 를 초과하지 않아야 하며 이 때 미뤄진 시간의 양은 $p_k^a - p_k^{a+1}$ 와 같다. 또한 위의 기호설명에서 정의한대로 SN_k^{a+1} , B_k^{a+1} , A_i^a , VS_i^{a+1} 도 업데이트 한다. [단계 S.6]에서는 다음 stage에서 시간 제약을 만족하지 못하는 마디가 생기게 되면 S 를 일정 비가능해로 두고 IT_3 을 증가시키며 일정계획 알고리즘을 종료하는 것을 의미한다.