

시간대 및 구역의존 차량이동속도를 고려하는 다목적차량일정문제: 일정계획해법과 전문가시스템*

Multiobjective Vehicle Scheduling Problem with Time and Area-Dependent Travel Speeds: Scheduling Algorithm and Expert System*

박양병**

Yang-Byung Park**

Abstract

This paper is concerned with the multiobjective vehicle scheduling problem with time and area-dependent travel speeds(MVSPTD), in which two conflicting objectives are explicitly treated and the travel speed between two locations depends on the passing area and time of day. The two objectives are the minimization of total vehicle travel time and the minimization of total weighted tardiness.

First, I construct a mixed integer linear programming formulation of the MVSPTD, and present a heuristic algorithm that builds the vehicle schedules based on the savings computed. The results of computational experiments showed that the heuristic performs very well. Finally, I propose an expert system for vehicle scheduling in the MVSPTD. Its whole process is executed under VP-Expert expert system environment.

1. 서론

최근의 통계자료[1]에 의하면 국내 기업의 수.배송비는 총물류비에서 약 70% 이상을 차지하고 있으며, 수.배송비중 육상 수송비가 약 66.5%로 가장 높은 것으로 나타나 있다. 게다가 이 비율은 지금까지 매년 증가추세를 보이고 있는 실정이다. 수.배송업무에서 최적의 차량일정을 수립하는 일은 비용의 절감은 물론 고객서비스의 질적 향상 및 생산시스템의 원활한 생산활동을 위해서 매우 중요하다.

그동안 다양한 형태의 차량일정문제(vehicle scheduling problems: VSP)에 대해 많은 연구[6]가 수행되어 왔다. 그러나 수행된 연구의 양과 질에 비추어 연구결과의 실제 적용이 크게 제한되어 온 것은 사실이다. 이에 대한 주요 이유를 다음과 같이 세가지로 정리할 수 있다.

- (i) 대부분의 VSP에서 두 지점간 차량의 이동시간을 항상 일정하게 가정하고 있다. 그러나 실제상황에서 차량의 이동속도는 이동시간과 이동구역에 따라 현저하게 변화하고 있다[2].

* 이 논문은 1995년도 한국학술진흥재단의 공모과제 연구비에 의해서 연구되었음.(1995. 9. 1 - 1997. 8. 31)

** 경희대학교 산업공학과 교수

- (ii) 대부분의 VSP에서 추구하는 목적은 단순히 차량의 총이동시간(거리)의 최소화이다. 그러나 최근의 물류환경은 고객서비스 수준향상이라는 또다른 목적의 추구를 요구하고 있다[3].
- (iii) 대부분의 VSP 해법은 설정된 가정을 토대로 수리모형을 풀거나 또는 절차적해법을 거쳐 해를 구한다. 따라서 실제 상황에서 발생할 수 있는 예외적인 문제, 가정의 완화, 일정계획 수립자의 주관적 판단 등이 해를 구하는 과정에 거의 반영되지 못하고 있다[14].

본 논문에서는 이와같은 문제점들을 고려하여 문헌상 처음으로 시간대 및 구역의존 차량이동속도를 고려하는 다목적차량일정문제(multiobjective vehicle scheduling problem with time and area-dependent travel speeds: MVSPTD)를 제안하고, 이에 대한 일정계획해법과 전문가시스템의 개발을 소개한다.

MVSPTD는 다음과 같이 정의된다. 차량들은 중앙차고지를 출발하여 지리적으로 산재해 있는 고객들의 알려진 수요를 모두 충족시킨 후 중앙차고지로 되돌아온다. 한 고객의 수요는 반드시 한대의 차량에 의해서 충족되어야 한다. 각 고객들은 납기와 배달시각의 상한(가장 늦을 수 있는 배달시각)을 지정하고 있으며, 고객의 중요도에 따라 순수납기지연(tardiness)에 대한 가중치가 달라진다. 차량의 지점간 이동속도는 차량의 이동시간대와 이동구역에 따라 달라진다. 각 차량의 차고지 귀환 허용시간(운전자의 근무시간)은 주어진다. 두개의 빈번히 상충하는 목적인 총차량운행시간의 최소화와 총가중순수납기지연의 최소화를 동시에 추구하며, 이들 목적은 분석자의 판단에 따라 그 상대적 중요도가 정해진다.

물류활동에서 고객의 서비스요구시각 준수는 매우 중요한 것으로 인식되고 있으며, 이러한 형태의 고객서비스는 유통, 서비스, 제조업 등 거의 모든 업종에서 전체 물류성과에 크게 영향을 미치는 것으로 나타나 있다[1, 3]. 따라서 MVSPTD는 실제로 여러 다양한 분야에서 발생할 수 있다. 예로서, 소·도매점(대리점)에 물품배달, 공장에 자재 및 부품배달, 가정에 물품배달, 방문 A/S, AGV 부품공급 등을 들 수 있다.

박양병과 송성현[2]은 차량속도가 이동구역과 시간대에

따라 변화하는 VSP에서 구역 및 시간대별 차량의 평균 속도를 추정하여 두 지점사이의 평균이동속도를 산정하는 모델을 제시하였다. Malandraki and Daskin[8]은 시간대에 따라 차량속도가 달라지는 차량경로문제(VRP)에 대해 nearest-neighbor 탐색을 토대로 한 발견적해법과 cutting planes를 이용한 발견적해법을 개발하였다. VSP의 특수모형인 traveling salesman problem에서 외판원의 이동시점에 따라 변화하는 이동비용을 고려한 연구는 지난 수년간 여러 학자들에 의해 수행되어 왔다. 주요 논문으로서 Wiel and Sahinidis[15], Balakrishnan et al.[5] 등이 있다.

Thangiah et al.[13]은 각 고객의 서비스 마감시간(time deadline)제약이 존재하는 상황에서 총차량대수와 총차량 이동시간 최소화의 두가지 목적을 추구하는 다목적차량 경로문제에 대해 genetic algorithm 접근방법을 이용한 발견적해법을 개발하였다. 서비스시간대를 하나의 제약조건으로 간주하면서 총차량이동시간 최소화와 같은 단일목적을 다루는 VSP나 VRP의 주요논문들은 Balakrishnan [4], Potvin and Rousseau[10], Solomon and Desrosiers [12] 등의 조사논문을 통하여 확인할 수 있다.

VSP나 VRP의 해를 구하는 과정에서 일정계획자의 전문적 지식이나 주관적 판단을 반영하는 기술에 대한 본격적인 연구는 전문가시스템과 컴퓨터기술의 발달에 따라 최근에서야 이루어지고 있다. Potvin et al.[9]는 서비스시간대 제약조건을 갖는 VRP의 해를 구하기 위한 interactive-graphic 컴퓨터시스템을 개발하였으며, 이 시스템을 기본으로 하여 해당 VRP의 특성에 가장 적합한 발견적해법을 선택하는 전문가시스템의 개발을 위한 기본틀을 제시하였다. Waters[14]는 차량일정계획 수립자의 주관적 판단과 선호를 어느 정도 반영할 수 있는 전문가시스템의 아이디어를 제안하였다. 지금까지의 VSP나 VRP를 위한 전문가시스템의 개발은 극히 초보적 수준에 머무르고 있다. 이것은 문제의 다양성과 지식베이스 구축을 위한 일반적 사실(facts)과 규칙의 부족함에 기인한 때문이라 볼 수 있다.

본 논문의 구성은 다음과 같다: 1절의 서론에 이어, 2절에서는 MVSPTD의 최적해를 구하기 위한 혼합정수 수리모형을 간단한 예제와 함께 제시한다. 3절에서는 MVSPTD의 근사해를 구하기 위해 개발한 BC-saving기

법을 소개한다. BC-saving기법의 성능을 평가하기 위한 계산실험의 내용이 4절에 정리된다. 5절에서는 BC-saving 기법을 기본으로 하여 MVSPD를 위한 전문가시스템의 개발이 논의된다. 끝으로, 6절에서 결론을 서술한다.

최근, 운행중인 차량의 관리를 위해 인공지능을 이용한 위치추적시스템(GPS)의 사용이 점차 확산되고 있는 실정이다. 그러나 배달문제와 같이 서비스대상 고객들에 대한 정보가 사전에 알려져 있는 경우, 차고지에서 또는 운행 중 고객들에 대한 차량일정을 미리 계획하는 일은 여전히 중요한 의사결정문제로 존재한다.

2. MVSPD 수리모형

2.1 수리모형

MVSPD에서는 두 지점사이의 이동시간이 차량의 출발시간대에 따라 달라지고 총차량운행시간 최소화와 총가중순수납기지연 최소화의 두개 목적을 함께 추구하기 때문에, 일반 VSP와는 다른 수리모형으로 구축된다. 출발시간대별 두 지점사이의 이동시간은 모두 사전에 알려져 있거나 또는 구역 및 시간대별 차량의 평균속도를 토대로 박양병과 송성헌[2]이 제안한 방법을 이용하여 추정할 수 있는 것으로 가정한다. 또한 이용가능한 차량수는 알려져 있으며, 차량의 유형은 모두 동일한 것으로 가정한다.

MVSPD 수리모형에서 사용되는 상수와 결정변수의 부호는 다음과 같다.

상수

- N = 차고지를 포함한 지점수(중앙차고지 = 지점 1)
- V = 이용가능한 차량수
- M = 시간대수
- l_i = 지점 i 의 납기
- A_i = 지점 i 의 배달시간 상한(가장 늦을 수 있는 배달 시간)
- c_i = 지점 i 의 배달량
- e_i = 지점 i 에서의 서비스시간
- R = 차량의 차고지 귀환시간 상한
- Q = 차량의 적재용량

t_{ijm} = 시간대 m 에 지점 i 를 출발하여 지점 j 까지의 이동시간

U_m = 시간대 m 에서 구간의 상한시간

w_i = 지점 i 의 가중치($w_i = 1, 2, 3$)

Φ = 총차량운행시간 최소화 목적의 비중($0 \leq \Phi \leq 1$)

B = 매우 큰 수

결정변수

$$x_{ijm} = \begin{cases} 1 & \text{만일 차량이 시간대 } m \text{에 지점 } i \text{를} \\ & \text{출발하여 } j \text{로 이동하면} \\ 0 & \text{그렇지 않으면} \end{cases}$$

a_i = 지점 i 에 차량도착시간

q_i = 지점 i 에 도착할 때 차량의 적재량

MVSPD는 이용가능한 차량수 V 만큼의 가상 도착차고지를 추가로 설정하여 아래와 같이 혼합정수 선형수리모형으로 구축할 수 있다. 가상 도착차고지의 위치는 중앙차고지와 동일하다. 이 모형은 Malandraki and Daskin [8]의 수식을 참고하여 발전시킨 것이다. 차량이 수거서비스를 수행하는 경우에는 이 모형에 대해 약간의 수정이 필요하다.

$$\text{Min } Z = \Phi \sum_{v=1}^V a_{N+v} + (1-\Phi) \sum_{i=2}^N w_i \max(a_i - l_i, 0) \quad (1)$$

subject to

$$\sum_{i=1}^N \sum_{m=1}^M x_{ijm} = 1, \quad j = 2, \dots, N+V \quad (2)$$

$$\sum_{j=2}^{N+V} \sum_{m=1}^M x_{ijm} = 1, \quad i = 2, \dots, N \quad (3)$$

$$\sum_{j=2}^{N+V} \sum_{m=1}^M x_{ijm} = V \quad (4)$$

$$a_j - a_i - Bx_{ijm} \geq t_{ijm} + e_i - B, \quad i = 1, \dots, N; j = 2, \dots, N+V; i \neq j; m = 1, \dots, M \quad (5)$$

$$a_j - t_{ijm} \leq U_m + B(1 - x_{ijm}), \quad i = 1, \dots, N; j = 2, \dots, N+V; i \neq j;$$

$$m = 1, \dots, M \quad (6)$$

$$a_j - t_{ijm} \geq U_{m-1} x_{ijm},$$

$$i = 1, \dots, N; j = 2, \dots, N+V; i \neq j;$$

$$m = 1, \dots, M \quad (7)$$

$$q_i - q_j - B \sum_{m=1}^M x_{ijm} \geq c_i - B,$$

$$i = 2, \dots, N; j = 2, \dots, N+V; i \neq j \quad (8)$$

$$q_j \sum_{m=1}^M x_{ijm} \leq Q, \quad j = 2, \dots, N \quad (9)$$

$$a_{N+v} \leq R, \quad v = 1, \dots, V \quad (10)$$

$$a_i \leq A_i, \quad i = 2, \dots, N \quad (11)$$

$$a_i \geq 0, \quad i = 2, \dots, N+V \quad (12)$$

$$q_i \geq 0, \quad i = 2, \dots, N \quad (13)$$

$$q_i = 0, \quad i = N+1, \dots, N+V \quad (14)$$

$$a_i = e_i = 0 \quad (15)$$

목적함수 (1)은 총차량운행시간과 총가중순수납기 지연의 합을 최소화한다. 두 목적의 상대적 중요도는 \emptyset 값에 의해 목적함수에 반영된다. 목적식에서 차량의 운행시간은 이동시간, 서비스시간, 그리고 대기시간을 포함한다. 대기시간은 차량이 이동속도가 좀 더 빨라지는 다음시간대까지 출발을 지연하는 경우에 발생한다. 제약식 (2)와 (3)은 각 고객이 한 차량에 의해 반드시 한번 방문되는 것을 보장한다. 또한 이 두 제약식과 x_{ijm} 은 정수라는 조건은 x_{ijm} 이 0 또는 1의 값을 갖게 해 준다.

위 모형에서 각 차량은 중앙차고지를 출발하여 고객들을 서비스한 후 V 개의 가상 도착차고지($N+1, \dots, N+V$)중 하나로 귀환한다. 지점 1로부터 곧바로 가상 도착차고지로의 차량이동은 해당차량이 배달에 사용되지 않음을 의미한다. 특정의 가상 도착차고지에는 반드시 한대의 차량만이 도착한다. 이를 위해,

$$t_{i, N+v, m} = 0, \quad v = 1, \dots, V; m = 1, \dots, M \quad (16)$$

$$t_{ijm} = t_{im}, \quad i = 2, \dots, N; j = N+1, \dots, N+V; m = 1, \dots, M \quad (17)$$

의 설정과 함께 제약식 (3)과 (4)가 필요하다.

제약식 (5)는 지점 j 에서의 차량도착시간을 계산한다. $x_{ijm}=1$ 때 양변항의 차이는 차량의 대기시간이다. 제약식 (6)과 (7)는 지점 i 에서의 출발시간에 대한 시간대 m 을 정해준다. 제약식 (8)은 지점 j 에 도착하는(또는 지점 i 를 떠나는) 차량의 현재 적재량을 계산한다. 제약식 (9)와 (10)은 각각 차량의 적재량과 차고지 귀환시간 상한을 나타낸다. 제약식 (11)은 각 고객의 배달시간 상한을 지정한다. 불법경로(illegal subtours)의 제거는 제약식 (5)와 (8)에 의해 보장된다. 위 수리모형은 NP-hard 문제에 속해 문제의 규모가 커지면 최적해를 구하기가 매우 어려워진다.

2.2 예제

수리모형을 이용하여 MVSPD의 최적해를 구하기 위해 고객수가 6인 간단한 예제를 구축하였다. 서비스구역은 9개의 동일한 크기의 사각형 모양으로, 그리고 시간대는 2개로 나누었다. 차량의 적재용량은 50 그리고 차고

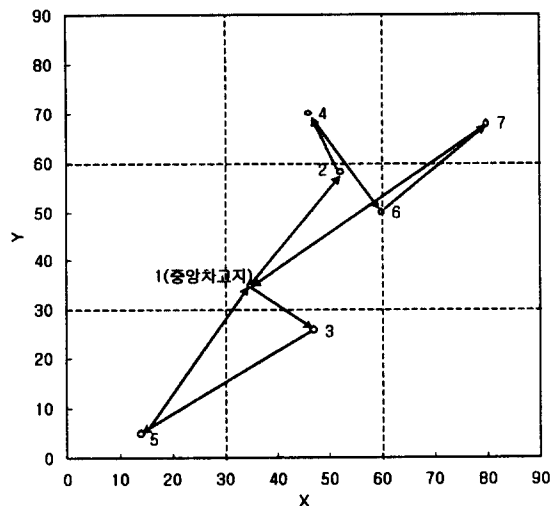


그림 1. 수리모형 예제의 최적 차량경로해

지 귀환 상한시간은 50으로 모든 차량에 대해 동일한 것으로 가정하였다. 고객 및 출발시간대별 지점간 이동시간 자료는 원하는 경우에 저자로부터 입수할 수 있다. 구축된 수리모형은 IBM호환 PC/586(36M RAM, 133MHz)에서 LINGO extended version[7]을 이용하여 실행되었다.

$\phi = 0.3$ 으로 하여 구해진 차량의 최적경로해가 그림 1에 그려져 있다. 총차량운행시간은 11.05 그리고 총가중순수납기지연은 0.00으로 계산되었다.

3. MVSPTD 발견적해법

MVSPTD의 근사해를 구하는 발견적해법을 개발하여 BC-saving기법(bicriteria-saving기법)이라 이름지었다. BC-saving기법은 한 루트의 마지막 방문지점 i 와 다른 루트의 첫번째 방문지점 j 를 연결함으로써 기대되는 차량운행시간의 saving(ST_{ij})과 가중순수납기지연의 saving(SD_{ij})의 합인 BC_{ij} 가 가장 큰 두 루트를 결합하면서 차량일정을 구축해 간다.

BC_{ij} 를 계산하는데 있어 두 saving의 상대적 중요도를 ϕ 로써 고려한다. BC_{ij} , ST_{ij} , SD_{ij} 를 구하는 식은 아래와 같이 정리된다. 모든 고객들의 서비스시간은 무시하여 차량의 고객지점 도착시간과 출발시간은 같은 것으로 가정한다. 부호의 정의는 수리모형에서와 동일하다.

$$BC_{ij} = \phi ST_{ij} + (1 - \phi)SD_{ij} \quad (18)$$

$$ST_{ij} = t_{i[a_i]} + t_{j[a_j]} - t_{ij[a_i]} + \beta \quad (19)$$

$$\beta = \begin{cases} 0 & \text{만일 } a'_j = a_j \\ \{TT(j) - t_{ij[a_i]}\} - \sum_{(m,n) \in P(j)} t_{mn[a'_m]} & \text{만일 } a'_j \neq a_j \end{cases}$$

$$SD_{ij} = \begin{cases} 0 & \text{만일 } a'_j = a_j \\ TD(j) - \sum_{(m,n) \in P(j)} \{w_m \max(a'_m - l_m, 0)\} & \text{만일 } a'_j \neq a_j \end{cases} \quad (20)$$

여기서

a'_j 턱두 루트를 결합함으로써 새롭게 결정되는 지점 j 도착(출발)시간($a'_j = a_j + t_{ij[a_i]}$)

$TT(j)$ = 두 루트의 결합전 지점 j 가 포함되어 있는 루

트의 차량운행시간

$TD(j)$ = 두 루트의 결합전 지점 j 가 포함되어 있는 루트의 총가중순수납기지연

$P(j)$ = 두 루트의 결합전 지점 j 가 포함되어 있는 루트에서 $(1, j)$ 경로를 제외한 모든 경로의 집합

MVSPTD를 위한 BC-saving기법에서는 이동시간대와 이동구역에 따라 차량이동속도가 달라지기 때문에 미리 모든 지점쌍에 대한 saving 값을 계산할 수 없다. 대신에, 두개의 루트의 결합을 시도할 때마다 대상 지점쌍들의 BC_{ij} 를 계산하여야 한다.

BC-saving기법의 절차를 요약하면 다음과 같다.

순서 1: 각 고객지점을 차고지와 연결하여 $N-1$ 개의 루트를 구축한다. 그리고 각 루트에 대해 차량운행시간과 총가중순수납기지연을 계산한다.

순서 2: 임의로 두개의 루트를 선정한 다음, 한 루트의 $(i, 1)$ 경로상 지점 i 와 다른 루트의 $(1, j)$ 경로상 지점 j 를 연결하여 한개의 루트로 결합하는데 따른 제약조건들(예: 차량귀환시간상한, 배달시각상한, 차량적재용량 등)의 만족여부를 조사한다. 제약조건을 만족하면 BC_{ij} 를 계산한다. 현재의 루트들에 대해 모든 가능한 지점쌍들의 BC_{ij} 를 계산한다. 만일 연결가능한 지점쌍이 하나도 존재하지 않으면, 순서 4로 이동한다.

순서 3: 가장 큰 BC_{ij} 값을 갖는 지점쌍을 선택하여 한개의 루트로 결합한 다음, 순서 2로 되돌아 간다.

순서 4: 한 루트에서 모든 인접지점간 쌍대교환에 따른 루트해의 개선을 계산하여 그중 가장 큰 개선을 제공하는 경로 (i, j) 의 순서를 교환한다. 더 이상의 개선이 나타나지 않을 때까지 이 과정을 반복한다.

순서 5: 모든 루트에 대해 순서 4의 과정을 적용한 후, 얻어진 루트일정해를 해로 하고 멈춘다.

위 절차에서는 묵시적으로 차량대수 최소화가 함께 고

려되고 있다. 명시적 두 목적인 총차량운행시간 최소화과 총가중순수납기지연 최소화는 ϕ 에 의해 조절된다. 만일 차량대수 최소화의 목적을 반영하지 않는다면, 순서 2에서 연결가능한 지점쌍의 조건으로서 $BC_{ij} > 0$ 을 추가하여야 할 것이다.

4. 계산실험

BC-saving기법의 일반적 성능을 평가하기 위해 총 8개의 실험문제를 구축하였다. 모든 실험문제에서 공통적으로 중앙차고지는 1개, 고객지점수는 100개, 차량용량은 100, 차량귀환시간은 100으로 설정하였다. 그리고 서비스대상지역은 편의상 모두 동일한 크기의 9개 정사각형 구역으로 나누었으며, 서비스시간대는 동일한 간격의 5개 구간으로 나누었다. 실험문제의 다양화를 위하여 고객의

집단화는 고객들이 특정구역에 밀집되어 있는 상태를 의미한다. 고객납기의 집단화는 0~20(10%), 20~50(30%), 50~80(50%), 80~100(10%), 그리고 고객가중치의 집단화는 1(20%), 2(50%), 3(30%)를 사용하였다. 표 1에 8개 실험문제의 고객특성이 요약되어 있다.

MVSPTD를 명시적으로 다루는 해법을 어느 문헌에서도 발견할 수 없었기 때문에, BC-saving기법의 비교대상 기법으로 Balakrishnan[4]이 soft 서비스시간대를 갖는 VRP를 위해 개발하여 그 성능을 입증한 NNS기법을 선택하였다. MVSPTD 실험문제에서 고객납기는 하한 및 상한값이 동일한 서비스시간대로 해석될 수 있기 때문에, NNS기법에서 연결가능한 고객을 선택하는 식 $PO(j)$ 는 아래와 같이 수정된다.

$$\text{Min } PO(j) = \phi t_{i|a_i} + (1 - \phi)w_i \max(a_j - l_j, 0) \quad (21)$$

표 1. 실험문제의 고객특성 요약

실험문제명	위치	납기	중요도	서비스시간
T000	일양분포	일양분포	일양분포	10
T001	일양분포	일양분포	집단화	10
T010	일양분포	집단화	일양분포	5
T011	일양분포	집단화	집단화	5
T100	집단화	일양분포	일양분포	10
T101	집단화	일양분포	집단화	10
T110	집단화	집단화	일양분포	5
T111	집단화	집단화	집단화	5

위치, 납기, 가중치(중요도)를 일양분포와 집단화의 2가지 형태로 다르게 구성하였다. 고객위치는 Solomon[11]의 실험문제 자료를 그대로 이용하였으며, 여기서 고객위치의

먼저, 문제 T000에서 총차량운행시간 최소화 목적의 비중(ϕ)을 0.0에서 1.0까지 0.1씩 증가하면서 BC-saving기법을 적용해 보았다. 그 결과가 표 2와 그림 2에 정리되어 있다. BC-saving기법의 적용에서 각 루트해의 개선을 위한 쌍대교환 반복회수는 최대 5회로 제한하였다. 표 2와 그림 2로부터, 두 목적값이 ϕ 값의 변화에 따라 대체적으로 잘 trade-off 되고 있음을 알 수 있다.

다음으로, 8개의 실험문제에 대해 $\phi = 0.0, 0.5, 1.0$ 으로써 BC-saving기법과 NNS기법을 적용하여 얻어진 결과를 비교해 보았다. 표 3에 두 목적값과 차량소요대수의 비교가 요약되어 있다. 절대비교치는 (BC-saving기법에 의한 해 * 100)/(NNS기법에 의한 해) 식에 의해 계산되었으며, 이것은 NNS기법에 의해 구해진 해를 100으로 할 때 BC-saving기법에 의해 구해진 해의 값을 의미한다. 평

표 2. 실험문제 T000에 대한 BC-saving기법의 적용결과

ϕ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
GTT	414.2	243.7	244.2	244.2	242.0	240.0	216.6	204.5	203.4	182.9	157.9
GTD	112.6	232.9	232.2	232.2	319.4	184.9	316.4	353.5	430.8	693.8	1760.2
NV	17	17	17	17	17	17	16	16	16	16	15

주 : GTT: 총차량운행시간, GTD: 총가중순수납기지연, NV: 차량소요대수

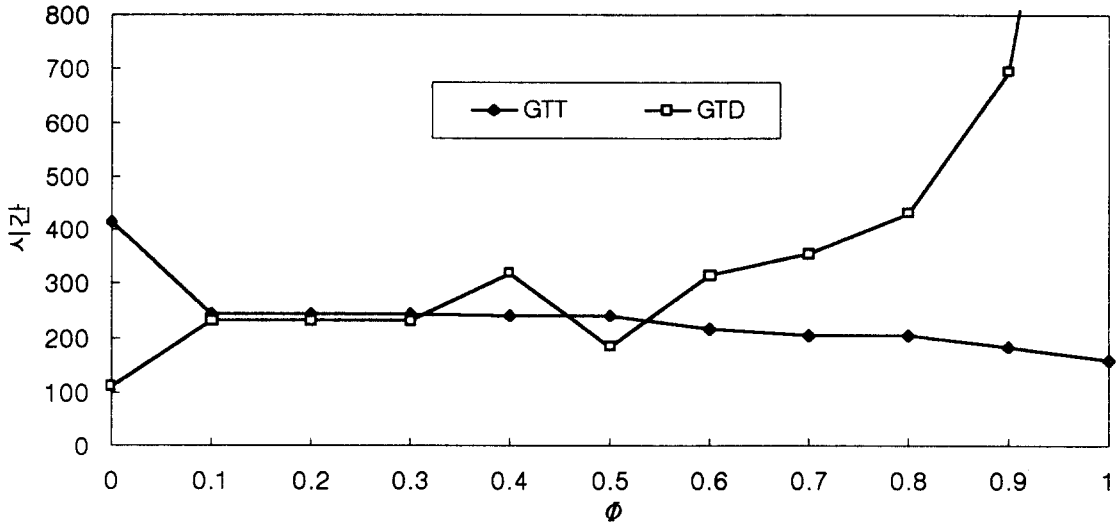


그림 2. 실험문제 T000에서 ϕ 의 증가에 따른 두 목적값의 변화

표 3. 8개 실험문제에서 BC-saving기법과 NNS기법의 성능비교

ϕ		0.0	0.5	1.0
T000	BC-saving기법	(414.2, 112.6, 17)	(240.0, 184.9, 17)	(157.9, 1760.2, 15)
	NNS기법	(486.5, 423.2, 16)	(347.3, 614.9, 16)	(308.9, 1661.1, 15)
T001	BC-saving기법	(383.8, 120.3, 16)	(212.7, 340.1, 17)	(170.2, 1784.3, 15)
	NNS기법	(469.9, 936.5, 16)	(320.2, 600.2, 15)	(312.0, 1268.0, 15)
T010	BC-saving기법	(371.9, 22.6, 12)	(154.0, 117.0, 14)	(152.2, 422.9, 13)
	NNS기법	(392.9, 332.1, 12)	(362.9, 293.9, 13)	(317.4, 768.9, 12)
T011	BC-saving기법	(450.0, 7.5, 12)	(150.9, 63.4, 13)	(149.2, 614.7, 12)
	NNS기법	(426.3, 812.3, 11)	(396.9, 550.7, 11)	(290.4, 853.8, 11)
T100	BC-saving기법	(376.0, 43.1, 16)	(213.9, 389.0, 17)	(164.4, 1234.3, 16)
	NNS기법	(396.9, 547.7, 15)	(392.1, 383.2, 16)	(196.1, 1913.3, 14)
T101	BC-saving기법	(375.4, 142.6, 16)	(235.3, 379.2, 17)	(147.3, 1621.7, 14)
	NNS기법	(489.1, 447.3, 16)	(404.9, 424.2, 16)	(240.5, 1592.4, 14)
T110	BC-saving기법	(287.7, 21.1, 11)	(111.7, 88.3, 14)	(91.9, 368.8, 14)
	NNS기법	(295.8, 324.5, 11)	(335.1, 340.3, 11)	(238.7, 739.4, 11)
T111	BC-saving기법	(339.2, 24.1, 12)	(110.0, 20.5, 15)	(96.6, 172.3, 15)
	NNS기법	(436.6, 561.5, 12)	(378.3, 292.1, 13)	(370.2, 712.8, 12)
평균절대비교치		(88.1, 4.2, 102.5)	(44.6, 21.9, 112.1)	(46.8, 59.1, 109.5)

주 (a, b, c)

a : 총차량운행시간, b : 총가중순수납기지연, c : 차량소요대수

균절대비교치는 8개 실험문제에서 구해진 절대비교치의 평균이다.

평균절대비교치로써도 알 수 있듯이, 거의 모든 경우의 문제에서 BC-saving 기법이 NNS 기법보다 더 짧은 총차량 운행시간과 총가중순수납기지연을 보여준다. 이것은 BC-saving 기법이 두 목적의 비중에 관계없이 일반적으로 NNS 기법보다 수행도가 더 뛰어남을 의미한다. $\phi = 0.0$ 때 BC-saving 기법의 총가중순수납기지연이 NNS 기법의 해에 비교하여 단지 평균 4.2%에 불과하다는 사실($\phi = 0.5$ 때 평균 21.9%, $\phi = 1.0$ 때 평균 59.1%)은 BC-saving 기법이 총가중순수납기지연 최소화 목적에 특히 매우 잘 작동한다고 말할 수 있다. 전체의 약 2/3 문제에서 BC-saving 기법의 해가 NNS 기법의 해보다 1~2대 정도의 더 많은 차량을 필요로 하고 있다. 이러한 사실은 실제 상황에 따라서 문제점으로 지적될 수 있다. 상세한 반복실험은 수행해 보지 않았지만, 표 3으로부터 BC-saving 기법의 성능이 실험문제에서의 고객특성에 의해 어떤 영향을 받는다는 사실은 발견할 수 없다.

5. MVSPD 전문가시스템

실제로 차량일정을 계획하는 상황에서는 수리모형이나 절차적해법에 포함되기 어려운 여러가지의 현실적인 문제들이 존재한다. 따라서 이들 방법이 그대로 사용되는 경우는 거의 드물며, 대부분의 경우에 일정계획자의 경험이나 직관을 토대로 한 주관적 의견을 반영하고 있다. 이 절에서는 MVSPD 환경에서 BC-saving 기법의 효능을 극대화하기 위해, 차량일정계획 전문가의 지식베이스를 기반으로 한 그리고 동시에 실제 상황적 요소들을 반영하는 전문가시스템(expert system for MVSPD: EXM)의 개발을 소개한다.

EXM의 구축을 위해 VP-Expert 전문가시스템 개발도구[16]를 이용한다. VP-Expert 전문가시스템 셸은 규칙을 이용한 지식표현방법과 후방향 추론방법을 사용하는 전문가시스템 프로그래밍언어이다.

5.1 EXM의 지식베이스 구축

EXM의 지식베이스는 문헌[9, 14]에 정리되어 있는 일

반 VSP에 관한 보편적인 지침과 MVSPD에 대한 일반 상식을 토대로 구축되었으며, 이들은 일정을 수립하기 전(pre-scheduling)과 BC-saving 기법을 적용하여 일정이 구해진 후(post-scheduling)에 대한 내용으로 나누어진다. 그러나 이들 지침과 상식들은 MVSPD의 다양한 실제상황에 기인하여 논쟁의 대상이 될 수 있음을 지적해 둔다. 이에 따라 지식베이스의 규칙을 EXM 사용자의 주관이나 상황에 따라 시스템과 사용자간의 interactive 과정을 통하여 용이하게 수정할 수 있게 하였다.

(1) pre-scheduling 지침 및 상식

- ① 기준 이동시간대의 인접한 고객지점들을 결합하여 지점수를 줄인다. 이때 결합된 지점의 위치와 납기의 결정은 평균치 또는 분석자의 주관적 판단에 근거한다.
- ② 도로조건(예: 일방통행, 공사) 및 특정시간대의 도로상태(예: 행사, 정체구간)에 따라 운행금지 경로를 설정하여 $t_{im} = \infty$ 로 한다.
- ③ 많은 고객들의 납기가 지나치게 이른 시각에 몰려 있는 경우는 납기를 적절히 조정한다. 한가지 방법으로, 차량귀환시각 상한치의 $1/x$ 이하의 납기를 요구하는 고객수가 전체고객수의 일정비율 이상일 때는 이들 고객중 인접고객들과의 가장 짧은 평균 이동시간이 기준시간 이상이면서 $w_i = 1$ (또는 ≤ 2)인 고객에 대해서 납기를 지연한다.
- ④ 인접고객과의 가장 짧은 평균이동시간이 아주 긴 고객의 배달은 별도의 방법(예: 일반운송업자, 소포 등)을 강구하도록 하고 일정계획 대상에서 제외한다.
- ⑤ 특급고객들을 최우선적으로 방문하도록 한다. 한가지 방법으로, 납기가 차량귀환시각상한의 $1/y$ 이하이고, 납기와 배달시각상한의 차이가 매우 작고, 그리고 $w_i = 3$ 의 조건을 동시에 만족하는 고객을 특급고객으로 간주한다.

(2) post-scheduling 지침 및 상식

- ① 차량의 적재용량때문에 아주 이른 시각에 귀환하는 차량들의 루트를 한 차량이 연속으로 서비스하는 것을 고려한다. 결과적으로, 차량소요대수는 감소하나

총가중순수납기지연은 증가할 것이다.

- ② BC-saving기법의 적용과정에서 최종적으로 결합된 두 루트에 대한 BC_i가 음수인 경우 두 루트를 결합하기 전(즉, 차량대수가 마지막으로 1대 감소하기 전)의 해를 검토한다. 이것은 차량소요대수를 1대 증가함으로써 분석자 관점에서 더 개선된 해를 얻을 수도 있기 때문이다. 이러한 검토를 루트결합 진행과정의 역으로 계속한다.
- ③ 차량적재용량, 차량회환시각상한, 고객배달시각상한 등의 제약조건을 약간씩 완화하여 얻어진 해를 검토한다.
- ④ 차량의 적재량이 아주 적은 경우에 해당루트의 제거를 시도한다. 한가지 방법으로서, 차량의 적재율이 기준이하인 경우 해당루트상 고객지점들을 모두 각각의 인접루트에 포함시키는 것을 고려한다. 결과적으로 차량대수를 1대 감소할 수 있다.
- ⑤ 일반 VSP 해의 특성을 토대로 구해진 해를 수정한다. 이를테면, $\emptyset \approx 1$ 인 경우의 루트해에 명백히 교차경로가 존재하면 이를 꽃잎모양으로 펼치거나,

$\emptyset \approx 0$ 인 경우에 가중순수납기지연이 일정크기 이상인 지점들의 방문순서를 루트 앞쪽으로 조정한다.

- ⑥ 두 루트가 이동시간상 인접해 있는 경우에 이들 루트에 대해 지점간 쌍대교환을 적용하여 해의 개선을 도모한다.

이상의 지침과 상식들을 "IF <조건> THEN <결론>"의 규칙으로 표현하여 지식베이스에 저장한다.

5.2 EXM의 구성

EXM의 구성은 pre-scheduling 및 post-scheduling 지식베이스, BC-saving기법 FORTRAN 프로그램, 제약조건조사 FORTRAN 프로그램, 목적값계산 FORTRAN 프로그램, 루트그래프작성 C 프로그램, 입력자료(고객, 차량, 지점간 이동시간) ASCII 데이터파일, 그리고 차량일정해 ASCII 데이터파일을 포함하여 총 7개의 모듈로써 이루어져 있다. EXM의 구성과 구성요소들의 상호관련을 도시하면 그림 3과 같다. EXM의 진행과정을 흐름도로 나

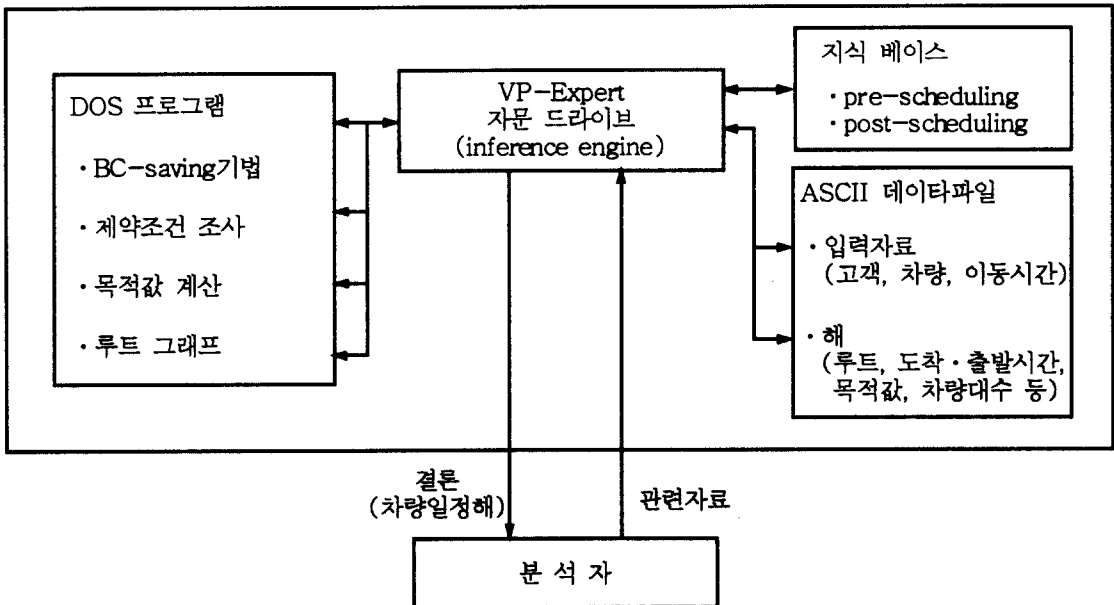


그림 3. EXM의 구성도

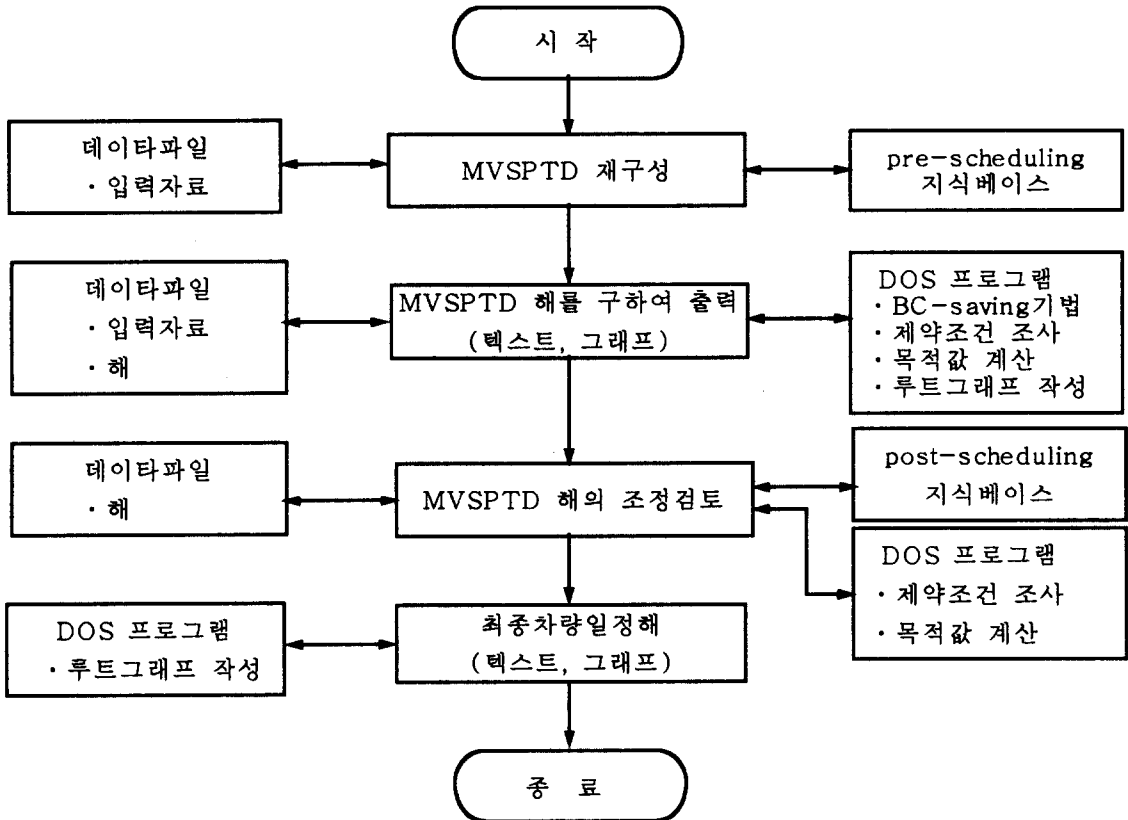


그림 4. EXM의 진행과정

타내면 그림 4와 같다. EXM의 전과정은 VP-Expert 전문가시스템 환경에서 실행된다.

EXM은 첫번째 단계에서 pre-scheduling 지식베이스의 추론과정에 따라 주어진 문제를 재구성한다. 이때 분석자는 지식베이스의 IF 조건문에서 사용되는 변수값의 입력을 요구받을 수도 있다. EXM은 분석자와의 원활한 인터페이스를 위해서 분석자에게 질문 및 선택메뉴에 대한 자세한 설명을 화면에 제공한다. 두번째 단계에서는 BC-saving기법, 제약조건조사, 목적값계산, 그리고 루트그래프작성을 위한 DOS 프로그램들을 함께 실행하여 구해진 차량일정해를 화면에 텍스트와 그래프로 출력한다. 세번째 단계에서는 post-scheduling 지식베이스의 추론과정에 따라 두번째 단계에서 구해진 차량일정해의 조정을 모색한다. 이 과정에서 EXM은 제약조건조사와 목적값계산 프

로그래밍의 실행을 필요로 한다. 끝으로, 루트그래프작성 프로그램을 실행하여 최종차량일정해를 텍스트와 그래프로써 화면에 출력한다.

추후, 저자는 별도의 논문을 통하여 EXM의 내용을 상세히 소개할 것이다.

5.3 EXM의 실행에

EXM의 적용을 위해서 4절의 실험문제 T000의 고객지점수를 50개, 차량용량을 100, 차량귀환시각을 60으로 축소하여 새롭게 예제를 구축하였다. 이 예제에 대한 EXM의 적용은 IBM PC호환 586 노트북 컴퓨터(16M RAM, 120MHz)상에서 실행되었다.

그림 5는 EXM에서 이동시간상 인접한 지점들을 한개

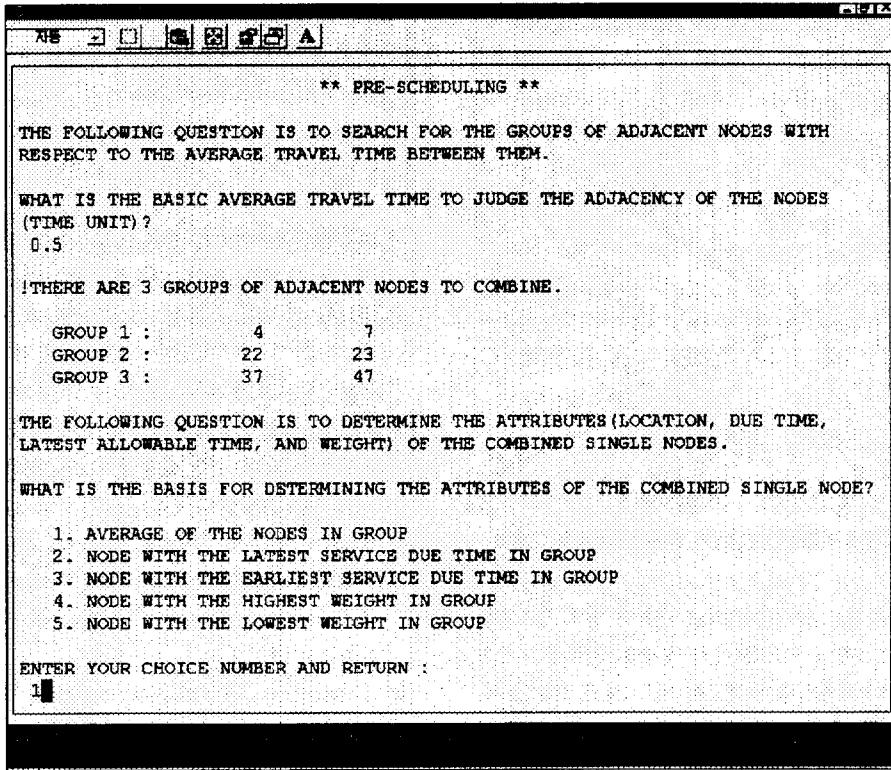
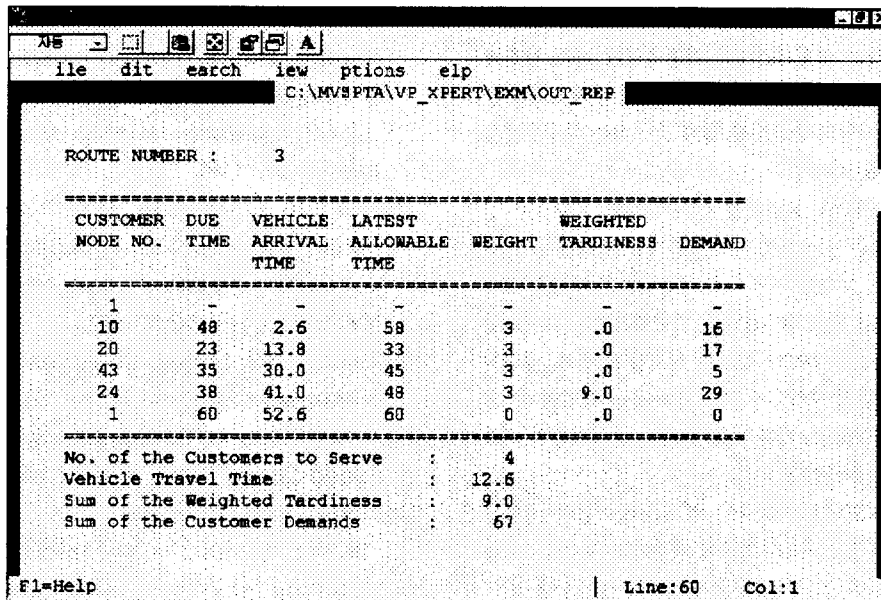


그림 5. EXM에서 pre-scheduling 규칙의 입력 및 실행에



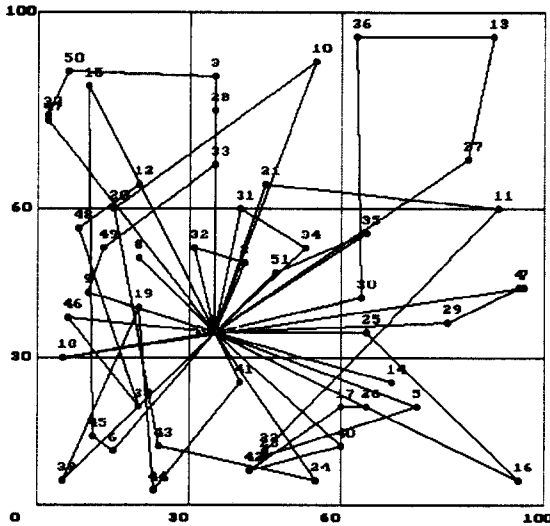


그림 6. EXM에서 최종차량일정해의 화면에(텍스트와 그래프)

지점으로 결합하는 pre-scheduling 규칙의 입력 및 실행 예를, 그리고 그림 6은 EXM에서 예제에 대해 $\phi = 0.5$ 때 구해진 최종차량일정의 텍스트 및 그래프 해를 보여 준다. 텍스트해 화면은 총 16개의 차량루트중 3번 루트의 일정해를 예로서 보여준다.

6. 결 론

본 논문에서는 이동시간대와 이동구역에 따라 변화하는 차량속도를 고려하면서 총차량운행시간의 최소화와 총가중순수납기지연의 최소화를 함께 추구하는 차량일정문제(MVSPTD)를 다루었다.

먼저, MVSPTD의 최적해를 구하기 위한 혼합정수 선형수리모형을 간단한 예제풀이와 함께 소개하였다. 그리고 수리모형의 계산상 어려움을 극복하면서 MVSPTD에 쉽게 적용할 수 있는 BC-saving 발견적해법을 개발하였다. Balakrishnan의 NNS기법과의 비교실험 결과, 제안된 BC-saving 기법의 수행도가 매우 뛰어나함을 확인할 수 있었다. 특히, BC-saving기법은 총가중순수납기지연의 최소화목적에 효과적이었다.

끝으로, MVSPTD 환경에서 BC-saving기법의 효능을 극대화하기 위해서 차량일정계획 전문가의 지식베이스를 기반으로 한, 그리고 동시에 실제 상황적 요소들을 반영

하는 EXM이라는 전문가시스템의 개발을 실행예와 함께 소개하였다. 추후, 저자는 별도의 논문을 통하여 EXM의 내용을 상세히 소개할 것이다.

참고문헌

- [1] 김문환, "물류관리를 위한 로지스틱스 시스템 구축방안", *ie 매거진*, 제2권, 제4호, pp.32-37, 1995.
- [2] 박양병, 송성현, "구역 및 시간의존 차량스케줄링문제: 차량속도 추정모델과 차량스케줄링 해법", *대한산업공학회지*, Vol.22, No.3, pp.517-532, 1996.
- [3] 유병우, 김동혁, "업종별 물류성공에 영향을 미치는 물류활동요인에 관한 실증연구", *로지스틱스연구*, 제3권, 제1호, pp.1-37, 1995.
- [4] Balakrishnan, N., "Simple Heuristics for the Vehicle Routing Problem with Soft Time Windows", *Journal of Operational Research Society*, Vol.44, No.3, pp. 279-287, 1993.
- [5] Balakrishnan, N., Lucena, A. and Wong, R.T., "Scheduling Examinations to Reduce Second-Order Conflicts", *Computers and Operations Research*, Vol. 19, pp.353-361, 1992.
- [6] Desrochers, M., Lenstra, J.K. and Savelsbergh, M.W. P., "A Classification Scheme for Vehicle Routing and Scheduling Problem," *European Journal of Operational Research*, Vol.46, pp.322-332, 1990.
- [7] *LINGO Optimization Modeling Language*, Lindo Systems, Inc., 1994.
- [8] Malandraki, C. and Daskin, M.S., "Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms", *Transportation Science*, Vol.26, No.3, pp.185-200, 1992.
- [9] Potvin, J. and Lapalme, G. and Rousseau, J., "A Microcomputer Assistant for the Development of Vehicle Routing and Scheduling Heuristics", *Decision Support Systems*, Vol.12, pp.41-56, 1994.
- [10] Potvin, J. and Rousseau, J., "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows", *European*

- Journal of Operational Research*, Vol.66, pp.331-340, 1993.
- [11] Solomon, M.M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", *Operations Research*, Vol.35, No.2, pp. 254-265, 1987.
- [12] Solomon, M.M. and Desrosiers, J., "Survey Paper: Time Window Constrained Routing and Scheduling Problems", *Transportation Science*, Vol.22, No.1, pp. 1-13, 1988.
- [13] Thangiah, S.R., Vinayagamoorthy, R. and Gubbi, A. V., "Vehicle Routing with Time Deadlines Using Genetic and Local Algorithms", *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA, pp. 506-513, 1993.
- [14] Waters, C.D.J., "Expert System for Vehicle Scheduling", *Journal of Operational Research Society*, Vol. 41, No.6, pp.505-515, 1990.
- [15] Wiel, R.J.V. and Sahinidis, N.V., "Heuristic Bounds and Test Problem Generation for the Time-Dependent Traveling Salesman Problem", *Transportation Science*, Vol.29, No.2, pp.167-183, 1995.
- [16] *VP-Expert: User Manual*, Paperback Software International, 1987.
-
- 97년 8월 최초 접수, 97년 10월 최종 수정