

시간 제약을 가지는 차량 경로 스케줄링 문제 해결을 위한 기회시간 반영 하이브리드 휴리스틱

유영훈

인하대학교 일반대학원 정보공학과
(yhyu@estab.inha.ac.kr)

차상진

인하대학교 일반대학원 정보공학과
(aldehyde7@estab.inha.ac.kr)

조근식

인하대학교 공과대학 컴퓨터정보공학부
(gsjo@inha.ac.kr)

본 연구는 시간 제약조건을 가지는 차량 경로 탐색과 스케줄링 문제(VRSPTW, the Vehicle Routing and Scheduling Problem with Time Window)를 해결하기 위하여, 기회시간(Opportunity time)을 반영한 하이브리드 휴리스틱을 제안하였다. 기회시간은 각 고객 노드에서 요구하는 하역 서비스를 수행 한 뒤에도 남아 있는 여유시간이다. 제안된 휴리스틱은 기회시간에 대한 제약조건을 추가하고, 기회시간을 고려한 비용 평가함수를 삽입 전략에 적용함으로써 초기 해를 구하였다. 또한 고객 노드 교환에 의한 타부 탐색 전략에도 기회시간을 반영함으로써 해를 개선 시켰다. 마지막으로 지리적, 시간적, 용량적으로 다양한 데이터 유형들에 대하여 각각 효과적으로 최적 해를 구할 수 있는 초기 경로 생성 전략들을 소개하고 비교하였다. 본 연구의 실험에서는 제안된 휴리스틱이 Solomon I1 휴리스틱 보다 효율적으로 최근사 해를 얻을 수 있음을 보였다.

논문접수일 : 2009년 08월 25일 논문수정일 : 2009년 09월 05일 게재확정일 : 2009년 09월 18일 교신저자 : 차상진

1. 서론

차량 경로 문제(CVRP, Capacity Vehicle Routing Problem)는 각 고객의 배송 요구량을 최소 이동 비용으로 배송하기 위한 해를 찾는 문제로써 많은 연구자들에 의해 최적해를 찾기 위한 연구가 되어 오고 있다(고준택 외 2, 2009). 그러나, 보다 현실적인 문제에서는 고객의 요구량뿐만 아니라, 고객의 업무 시작과 종료시간 준수도 요구되고 있다. 이처럼, 각 고객에서의 시간 제약뿐만 아니라 요구량을 만족시키면서 최소 이동 비용을 가지는 경로를 구성하기 위한 문제를 VRSPTW(Vehicle Routing and Scheduling Problem with Time Window)라 한다(Bodin L., et al., 1983).

VRSPTW 문제에 대한 해를 찾는다는 것은 외판원 문제(TSP, Travel Salesman Problem)를 동시에 여러 개 푸는 문제와 유사하지만, 문제의 크기가 커질수록 계산량이 지수적으로 증가하게 되는 NP-Hard 문제이다(Lenstra and Rinnooy Kan, 1981; Savelsbergh M, 1984).

이러한 문제를 해결하기 위한 방법으로는 크게 완전 최적화 기법(Exact Optimization Method)과 메타 휴리스틱 기법(Meta Heuristic Method)이 있다. 완전 최적화 기법은 모든 가능한 경로를 조사해보는 방법이다. 대표적인 방법으로는 열 생성 계획법(column generation scheme)으로 LP(Linear Programming) 문제를 푸는 방법과 Branch and Bound 기법(이하 B&B)을 응용한 기법들이 있다(Kolen et al.,

1987; Desrochers et al., 1992; Bard J.F., et al., 2002; Irnich S., et al., 2005). 그러나 이 방법은 문제에 따라서 최적 해를 구하는 시간이 오래 걸리거나, 최적 해를 구한다는 보장이 없을 수도 있다. 메타 휴리스틱 기법은 최근사 해(Nearest Optimal Solution)를 효율적으로 얻는 것을 목적으로 한다. 이를 위해 다른 분야의 개념들을 결합하여 정보를 구축하고 학습함으로써 지능적으로 탐색공간을 탐색한다. 이때에 내부 휴리스틱들을 제어하여 반복적으로 생성된 해(iterative generation process)들 중 최근사해를 산출해 낸다. 이는 완전 최적화 기법에 비해 빠른 시간 내에 최근사 해를 얻을 수 있다는 장점이 있다(Ibrahim H. et al., 1996). VRSPTW 문제를 해결을 위한 대표적인 메타 휴리스틱으로는 solomon의 삽입 기법(Insertion Heuristic)과 Taillard의 타부 탐색 기법 등이 있다. Solomon은 VRP문제에 시간제약을 고려하기 위해 일반화된 할당 휴리스틱(generalized assignment heuristic)과 절약 휴리스틱(Saving Heuristic), 최이웃 탐색 휴리스틱(Nearest Neighborhood Search Heuristic)을 결합한 삽입 전략과 경로 초기화 전략들을 소개하였다(Clarke and Wright, 1964; Fisher M., et al., 1981; Solomon, 1987). Taillard는 solomon II 삽입 휴리스틱과 Glover의 타부 탐색 휴리스틱을 결합하였으며, 보정 저장소(Adaptive memory)를 사용한 CROSS 교환 기법을 제안하였다(Taillard et al., 1997).

본 연구는 VRSPTW 문제를 해결하기 위해, 기회시간을 고려한 비용 평가 함수를 적용한 삽입 전략을 수립하였다. 이는 고객의 종료시간 내에 고객 요구량을 배송할 수 있는 시간적 가능성 정도를 나타낸다. 또한, 초기 경로를 생성할 때, 고객 노드를 선택하기 위한 몇 가지 전략들을 소개하였으며, 각 전략에 대한 라우팅 결과를 보였다.

2. 관련 연구

2.1 완전 최적화 기법

VRSPTW 문제를 해결하기 위한 대표적인 완전 최적화 기법중 하나인 B&B 기법은 경로를 구성할 때마다, 고객 노드로부터 분기하여 얻을 수 있는 해에 대한 한계치인 bound를 계산한다. 만일, 그 한계치가 지금까지 찾은 최적 해보다 좋지 않을 경우엔 더 이상 노드를 탐색할 필요가 없기 때문에 해당 하위 고객 노드들을 제거(pruning)함으로써 탐색 공간을 줄일 수 있다(Desrochers et al., 1992). 최근에는 LP로 정의된 Cutting Plane을 B&B 기법에서의 노드 제거 전략에 적용한 Branch-and-Cut 연구(Bard J.F., et al., 2002)와 k-cycle 제거 전략으로 Branch-and-Price를 개선시킨 연구가 있었다(Irnich S. et al., 2005). 그러나, 이러한 기법들은 노드들을 제거할 수는 있지만, 고객 노드 수가 증가함에 따라 탐색 공간이 증가되기 때문에 해를 얻기 위해서는 여전히 많은 시간이 필요하다. 이러한 완전 탐색 기법들은 빠른 시간 내에 경로 탐색 스케줄링 문제를 해결해야하는 상황에서는 적합하지 않다. 따라서, 빠른 시간 내에 최근사 해를 찾기 위한 메타 휴리스틱 기법에 대한 연구가 필요하다.

2.2 삽입 기법(Insertion Heuristic)

VRSPTW 문제에서 경로를 구성하기 위한 휴리스틱 전략은 크게 순차 방법(sequential procedure)과 병렬 방법(parallel procedure)으로 나뉜다. 순차적인 방법은 모든 고객들이 스케줄 될 때까지 한 번에 하나씩 경로를 구성하는 방법으로써 해를 찾는 시간이 오래 걸린다. 병렬 방법은 모든 경로를 동시에 구성하는 방법으로써 Solomon의 삽입 기법이 대표적이다.

Solomon의 삽입 기법은 Clarke and Wright의 절약 휴리스틱(Saving Heuristic)과 최근접 이웃 탐색 휴리스틱(Nearest-Neighbor Heuristic)에 의해 경로가 구성될 때 시간 제약을 고려하는 삽입 전략이다. Solomon은 삽입 전략을 수행하는데 있어서, 경로 상에 고객을 삽입 할 경우의 거리 변화량과 대기시간이 반영된 출발시간 변화량에 각각 가중치를 부여한 삽입 평가 함수를 사용하였으며, 시간에 가중치를 부여한 비용 평가 함수인 I_1 (이하 Solomon I_1)이 더 우수하였음을 보였다(Clarke and Wright, 1964; Solomon M., 1987).

그러나, Solomon의 삽입 전략은 고객 노드를 선택하는 전략에 따라서, 해의 결과가 달라질 수 있으며, 지역 해(local minimum solution)에 빠질 수도 있다. 이러한 지역 해를 극복하기 위해 2-opt나 Or-opt들과 같은 고객 노드 교환 기법들을 적용시키기 위한 연구가 되어오고 있다(Or, 1976; Baker and Schaffer 1988; Potvin and Rousseau, 1995).

2.3 표준 타부 탐색 전략과 응용

Glover에 의해 정립된 표준 타부 탐색 전략(Standard Tabu Search Strategy)은 TSP와 같이 복잡한 최적화 문제를 해결하기 위해 사용될 수 있는 메타 휴리스틱이다(Glover, 1989; Glover, 1990; Glover et al., 1993).

타부 탐색은 극부 지역 탐색(local search)에 빠지지 않기 위해 주어진 종료 조건이 만족할 때까지 반복적으로 이웃 탐색을 수행한다. 새로운 이웃 해는 타부목록(Tabu List)과 열망수준(Aspiration Criteria)을 가지는 메모리 구조에 의해 관리된다. 타부 목록은 해의 이동정보를 일정기간 기억하기 위한 것으로서 타부 목록의 크기만큼 정보가 유지되며, 새로운 정보가 입력되면 가장 먼저 입력되었던 정

보가 제거되는 FIFO(First In First Out) 구조이다. 그러나 타부 목록의 사용은 더 좋은 해가 존재하는 곳으로의 이동을 방해 할 수도 있다. 이러한 한계를 극복하기 위해, 현재 탐색하고 있는 탐색 영역에서 벗어날 수 있도록 열망수준이 사용된다.

Taillard는 Solomon의 삽입 기법과 타부 탐색 전략 기법을 사용하여 VRSPTW 문제를 해결하였다(Taillard et al., 1997). 이 방법은 Solomon I_1 의 비용 평가 함수와 삽입 기법으로 초기 해를 구하고, 이를 하위 경로들의 집합으로 나눈 뒤, 보정 저장소(adaptive memory)에 저장한다. 거점을 기준으로 일정한 각도(degree)의 범위에 있는 두 개의 하위 경로를 보정 저장소에서 선택하고, 선택된 두 경로 상에 존재하는 두 고객 노드들을 교환함으로써 해를 개선시킨다(이를 CROSS 교환법이라 부른다).

Lau는 라우트에 미할당된 노드들을 보유 리스트(Holding List)로 구성하고 재위치(relocation) 전략으로 초기 해를 산출하였다. 산출한 해를 노드 교환(exchange)과 재위치 휴리스틱 기반의 타부 전략을 사용하여 해를 개선 시켰다(Lau H.C., et al., 2003).

본 연구에서는 기회 시간을 반영한 삽입 전략으로 생성된 각 경로들에 대하여, 기회시간을 반영한 2-opt기반의 타부 탐색 전략을 적용함으로써 해를 개선시키고자 하였다.

3. 시간 제약을 가지는 경로 탐색 스케줄링 문제

3.1 문제 정의와 기본 제약조건

VRSPTW는 각 고객의 요구량과 이동 거리와 더불어 차고지와 각 고객의 시간 제약을 준수하여

경로를 구성해야 하는 문제이다. 구성된 경로는 $n+1$ 개의 고객 노드 집합으로 구성된 연결 그래프인 $G=(V, E)$ 로 나타낼 수 있다. 이때 모든 고객 노드들은 경로에 포함되어야 하며 오직 한번만 방문될 수 있다. V 는 고객(노드 또는 배송지)들의 집합이며, E 는 v_i 와 v_j 사이의 연결 간선(edge)들의 집합이다.

- n : 고객의 수
- i, j : 고객 방문지, $i = 1 \dots n, j = 1 \dots n$
- V : 고객 방문지 v_i 의 집합, $i = 1 \dots n$
- $v_i \in V$
- E : 모든 고객 간 간선들의 집합
- $E = \{(v_0, v_1), (v_1, v_2), \dots, (v_i, v_{i+1})\}$

경로 $path$ 는 그래프 G 에서 고객 노드 v_i 와 고객 노드 v_j 간의 간선들의 집합들을 나열하거나 고객 방문지들을 순서대로 나열함으로써 나타낼 수 있다. 구성된 경로 $path_k$ 는 차량 t_k 에 의해 운행하게 된다.

- T : 차량 t_k 의 집합, $k = 1 \dots n$
- t_k^{capa} : 차량 t_k 의 용량
- u : 경로 상의 고객 수
- $path_k$: 경로, $k = 1 \dots n$
- $\begin{cases} path_k = t_k & \text{단, } k = 1 \dots n \\ path_k = \{v_0, \dots, v_u, v_{u+1}\} \end{cases}$
- 단, $v_u \in V, 1 \leq u \leq n$

이러한 경로는 출발지로 되돌아와야 하는 사이클을 가진다. 따라서, 첫 방문지 v_0 를 거점(*depot*)이라 한다면, 마지막 고객 v_z 를 방문하고 난 뒤의 방문지 v_{z+1} 은 거점과 동일하다

$$depot = v_0 = v_{z+1} \quad (1)$$

이 문제에서의 차량 대수는 주어진 고객 노드 수만큼 사용할 수 있지만, 구성된 경로에서의 고객 요구 총 용량은 경로에 투입되어야 하는 차량의 적재 용량보다 클 수 없다. 또한, 모든 차량의 적재 용량은 모두 동일한 것으로 가정한다.

$$t_0^{capa} = t_1^{capa} = t_2^{capa} = \dots = t_k^{capa} \quad \text{단, } 1 \leq k \leq n \quad (2)$$

경로 $path_k$ 상에 있는 고객들의 요구 용량(demand quantity)에 대한 합은 차량 t_k^{capa} 보다 클 수 없다.

$$t_k^{capa} \leq (v_{k1}^{capa} + v_{k2}^{capa} + \dots + v_{kz}^{capa}) \quad (3)$$

경로 $path_k$ 를 구성하기 위해서는 거점과 각 고객에 대한 시간 제약인 TW_i 내에 하역 서비스가 이루어져야 한다. 따라서, 이전 고객 방문지 $i-1$ 에서 출발하여 i 에 도착할 경우, 고객 i 로의 도착시간 a_i 는 고객 i 의 업무 시간보다 일찍 도착한다면 i 의 시작시간 s_i 가 되며, 그렇지 않으면 $i-1$ 으로의 도착시간과 $i-1$ 에서의 하역 서비스 시간 l_i , 두 고객 $i-1$ 과 i 사이의 이동시간을 더한 시간이 된다. 여기서, 두 고객 간의 이동 시간은 직접적인 연결 거리인 유클리디안 거리(Euclidean distance)를 사용하기 때문에 이동 거리와 동일한 것으로 가정한다.

- s_i : 고객 i 에서의 업무 시작 시각
- e_i : 고객 i 에서의 업무 종료 시각
- l_i : 고객 i 에서의 서비스 시간(하역시간)
- $tr_{i-1,i}$: 고객 i 와 j 사이의 이동시간(거리)
- TW_i : 고객 i 에서의 시간 제약(time window)

$$TW_i = [s_i, e_i]$$

- a_i : 라우트된 고객 i 로의 도착시각
 d_i : 라우트된 고객 i 에서의 출발시각
 w_i : 라우트된 고객 i 에서의 대기시간

$$a_j = \max\{s_j, a_{i-1} + l_{i-1} + tr_{i-1,i}\} \quad (4)$$

$$v_i^{TW(s_i)} \leq a_i \leq v_i^{TW(e_i)} \quad (5)$$

고객 i 에서의 대기시간 w_i 는 TW_i 상의 요구 시작 시각 s_i 보다 차량이 일찍 도착할 경우에는 s_i 에서 도착시각 a_i 를 뺀 시간이지만, 차량이 s_i 시각 이후에 도착할 경우에는 0이다.

$$w_i = \max\{0, s_i - a_i\} \quad (6)$$

고객 i 에서의 출발시각 d_i 는 하역시간 l_i 로 인해 TW_i 의 e_i 가 벗어나는 것을 허용하며, 이는 도착 시각 a_i 와 하역 시간 l_i , 대기 시간 w_i 의 합이라 할 수 있다.

$$d_i = a_i + l_i + w_i \quad (7)$$

또한, 거점 v_0 에서의 종료시간 e_0 내에 되돌아와야 하기 때문에, 마지막 고객 방문지에서의 출발 시간 d_u 와 다시 거점까지의 이동시간 $tr_{z,0}$ 을 합한 시간보다 크거나 같아야 한다.

$$v_0^{TW(e_0)} \geq d_z + tr_{z,0} \quad (8)$$

3.2 목적 함수

시간 제약을 가진 차량 경로 스케줄링 문제의 최종 목적함수는 차량 수와 라우트된 전체 스케줄 시간, 총 이동거리, 총 대기시간을 최소화하는 것이

다. 또한, 이들 목적함수에는 차량 수, 스케줄 시간, 총 이동거리, 총 대기시간의 순서대로 우선순위를 가진다. 이는 전체 스케줄 시간이 좋으나 차량 수가 많은 결과보다는 비록 스케줄된 시간이 크더라도 사용된 차량수가 적은 결과가 더 좋다는 것을 의미한다. 먼저 차량 수에 대한 목적 함수는 다음과 같다.

$$\min \sum_{k=1}^p (t_k) \quad (11)$$

단, $1 \leq p \leq n$

스케줄 시간에 대한 목적 함수는 각 경로에서의 마지막 도착시간들에 대한 합을 최소화하는 것이다. 각 경로 $path_k$ 내의 방문지 개수를 z 이라 한다면 다음과 같다.

$$\min \sum_{k=1}^p (path_k^{a_{z+1}}) \quad (12)$$

단, $1 \leq p \leq n$

마지막으로 총 이동 거리에 대한 목적 함수는 다음과 같다.

$$\min \sum_{k=1}^p \sum_{u=0}^z tr_{u,u+1} \quad (13)$$

단, $1 \leq p \leq n, 1 \leq u \leq n$

4. 기회시간이 반영된 하이브리드 삽입 휴리스틱

4.1 제안 하이브리드 휴리스틱

본 연구에서 제안한 휴리스틱 절차는 <표 1>과 같다. 먼저, 주어진 차량과 고객 정보들로 고객 리스트와 거리 테이블을 생성하고, 초기 경로를 생성

<표 1> 제안 하이브리드 휴리스틱

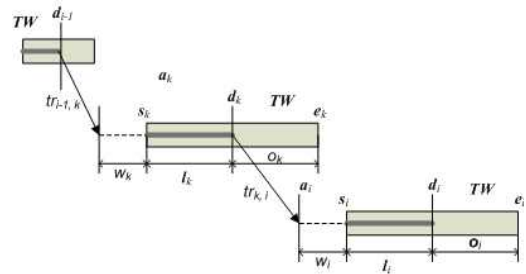
```

PROCEDURE solve()
read the vehicles  $T$  and customers  $V$ ;
make the distance table;
for each the initialization mode
    set the path initialization and the cost evaluation
    mode;
     $RouteSet \leftarrow Insertion(V, T)$ ;
    for each tour in  $RouteSet$ 
        tabu(tour);
    if ( $bestRoute > RouteSet$ )
         $bestRoute \leftarrow RouteSet$ ;
    print  $bestRoute$ 
    
```

하기 위한 전략과 비용 평가 함수 적용 전략을 설정한다. 각각 4가지의 초기 경로 생성전략과 2가지의 비용평가 함수를 사용하였다. 초기 경로 생성 전략으로는 기회시간을 최소로 가지는 고객 노드를 우선 선택하여 초기 경로를 생성하는 방법과 3시 방향을 기준으로 시계 반대방향으로 각도 회전(sweep)하여 형성된 클러스터의 중간에 위치한 고객 노드를 선택하는 방법(Gillet and Miller, 1974; 고준택 외 2, 2009), 고객의 요구 종료시간이 가장 일찍 끝나는 고객 노드를 선택하는 방법(Solomon M., 1987), 거점에서 가장 멀리 떨어진 고객 노드를 우선 선택하여 초기 경로를 구성하는 방법을 각각 적용하였다. 비용평가 함수로는 기회시간을 고려한 평가함수와 그렇지 않은 함수를 각각 적용하였다. 삽입 전략에 의해 수행된 결과를 가지고 타부 전략을 사용하여 해를 개선시키며, 개선된 해들 중 가장 좋은 해를 최종 해로 선정한다. 이때, 기회 시간은 삽입과 타부 탐색 전략의 제약조건 뿐만 아니라 삽입 평가함수로도 적용하였다.

4.2 기회시간 제약조건

경로 상의 최적의 위치에 미할당 고객을 삽입함으로써 전체 경로의 스케줄 시간에 영향을 줄 수 있는 삽입 비용 요소엔 이동거리($tr_{i,j}$), 대기시간(w)과 기



<그림 1> 고객 v_k 를 v_{i-1} 과 v_i 에 삽입할 경우, 경로 상에서의 기회 요소

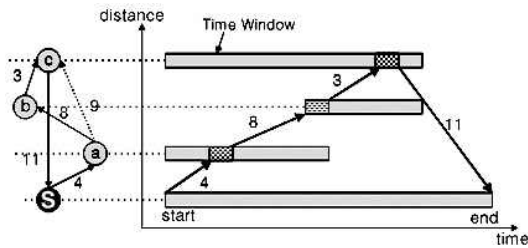
회 시간(r)이 있다. 본 연구에서는 Solomon이 고려한 비용 평가 요소(Solomon M., 1987)인 거리와 스케줄 시간뿐만 아니라 기회 시간도 함께 고려하였다.

<그림 1>에서 기회 시간 o_i 는 TW_i 상의 e_i 에서 고객 i 에 대한 하역을 종료한 뒤 출발하는 시각 d_i 를 뺀 시간이다. 기회 시간 o 의 작은 값을 우선 선택하여 삽입하면 이후 더 많은 고객 노드들을 경로 상에 삽입할 가능성이 있다는 것을 의미하기에 가능한 먼저 경로에 할당되어야 한다.

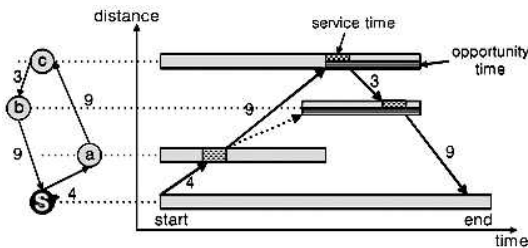
$$o_i = e_i - d_i \tag{14}$$

따라서, 경로 상에서 효율적으로 시간제약을 운영하기 위해서는 대기시간과 기회시간이 최소인 고객을 우선 삽입해야 할 필요가 있다. 경로 구성을 위해 기존 고객 $i-1$ 과 i 사이에 고객 k 를 삽입할 때, 경로 상에 삽입한 후의 고객 k 에 대한 도착 시각을 a'_k 이라 하고, 미할당 된 고객들 중에서의 가장 작은 기회 시간을 $\min(o_f)$ 이라 한다면, 추가 제약조건은 다음과 같다.

$$\begin{aligned}
 &(a'_k \leq e_k) \ \&\& \ (a'_i \leq a_i + \min(o_f) + w_i) \\
 &\text{단, } i=1 \dots n \ n \text{은 고객의 총 개수} \\
 &\quad f=k \dots n \tag{15}
 \end{aligned}$$



(a) 기회 시간 미 반영된 결과



(b) 기회 시간 반영된 결과

<그림 2> 기회시간 반영 여부에 대한 경로 생성 예제

<그림 2>에서, (a)는 고객 노드 1에서 이동하려는 다음 행선지로 거리와 대기시간이 짧은 2번을 선택했으며, 구성된 경로의 이동 비용은 26이 되었다. 그러나, 기회시간을 고려할 경우 이동할 다음 행선지로 3번을 선택할 수 있기 때문에 더 작은 이동 비용인 25로 경로를 구성할 수 있다.

4.3 제안 삽입 휴리스틱과 기회시간 비용 평가 함수

제안된 삽입 휴리스틱은 <표 2>에서와 같이 *RouteSet*을 초기화함으로써 시작된다. 이 초기화 과정은 먼저, 주어진 고객 노드들을 미할당 고객 리스트 V_u 에 저장한다. 이 리스트로부터 제 4.1절에서 소개한 초기 경로 생성 전략에 따라 경로 *path*를 구성한다. 경로 구성에 참여된 고객 노드들은 V_u 에서 제거된다. 이후, 현재 구성된 모든 *path*들에 대하여 V_u 에서 삽입 예상 비용이 가장 작은 고

<표 2> 제안 삽입 휴리스틱

```

PROCEDURE Insertion(  $V, T$  )
    //  $V$  : Customers,  $T$  : Vehicles
    Initialize RouteSet
    while exist unassigned customer in  $V$ 
        Create costTable;
        for each unassigned customer
            for each exist path in RouteSet
                if not satisfy the capacity of path.
                    then move to next path;
                for each visited customer  $k$  on path
                    if not satisfy the time window of path.
                        then cost is null,
                    else compute cost;
                        store the best cost on path to costTable;
                if the row of costTable is all null
                    then create new path to RouteSet
                else insert customer into path at index of best cost on costTable;
            remove an inserted customer from  $V$ ;
    end-while
    
```

객 노드 v 를 찾기 위해 제약조건과 삽입 평가함수가 고려된다. 식 (1)~식 (8)에 의해 주어진 제약 조건이 만족하는지 확인하며 모두 만족하면 삽입 예상 비용을 계산한다. 삽입 예상 비용은 2차원 배열인 *costTable*로 관리되며, 열(row) i 는 미할당 고객 노드 V_{u_i} , 행(column) p 은 $path_p$, 즉 차량 T 를 의미한다. *costTable*의 각 셀(cell)에는 열 V_{u_i} 를 행 $path_p$ 에 삽입할 경우에 대한 최소 삽입 비용이 저장된다. 예를 들어, $path_p$ 상의 고객 노드 v_{k-1} 과 v_k 사이에 미할당 고객 v_u 를 삽입 한다고 할 때, 예상되는 삽입 이동거리 증감 비용은 다음과 같으며,

$$\begin{aligned}
 \text{Travel Time Cost}_{pu} &= (tr_{k-1,u} + tr_{u,k}) - tr_{k-1,k}
 \end{aligned}$$

대기시간 증감 비용과 기회시간 비용은 다음과 같다.

$$\begin{aligned} \text{Wait Time Cost}_{pu} &= w_u + (w_f^{\text{old}} - w_f) \\ f &= u \dots \text{sizeOfpath}_p \\ \text{opportunity Time Cost}_{pu} &= \min(o_f) \\ f &= u \dots \text{sizeOfpath}_p \end{aligned}$$

최종적으로 costTable의 각 셀 <p, u>에 저장되는 비용 값은 다음과 같다.

$$\text{cost}_{pu} = \text{Travel Time Cost}_{pu} + \text{Wait Time Cost}_{pu} + \text{opportunity Time Cost}_{pu} \quad (16)$$

4.4 제안 타부 탐색 전략

삽입 휴리스틱의 결과인 각 경로는 제안된 타부 탐색 전략에 의해 다시 최적화된다. 본 연구의 타부 탐색 전략은 표준 타부 탐색 전략(Golver et al., 1993)과 2-Opt(Baker and Schaffer, 1988)를 응용하여 시간제약과 기회시간을 고려할 수 있도록 확장 되었다. <표 3>과 같이 경로 상의 두 고객 노드를 선택 한 뒤, 교환할 경로가 타부 리스트 상에 존재하지 않으면 식 (15)가 만족하는지 확인 하고, 만족하면 비용 평가 식 (16)에 의거 비용을 평가한다. 만일 이전 경로보다 비용이 감소되면 두 고객 노드를 교환하게 된다. tenure 값과 열망 수준 값에 의해 이웃 해들을 구하게 되고, 이러한 이웃해들 중 가장 좋은 해를 최종 해로 얻게 된다.

5. 실험 및 평가

실험은 Solomon의 실험 데이터와 동일한 테스트 데이터 집합을 사용하였다(Christofides et al., 1979; Solomon M., 1987). <그림 3>과 같이 이 데이터는 100개의 고객노드가 지역적으로 군집된 C형, 임의 위치로 분산된 R형, C와 R이 혼합된 RC

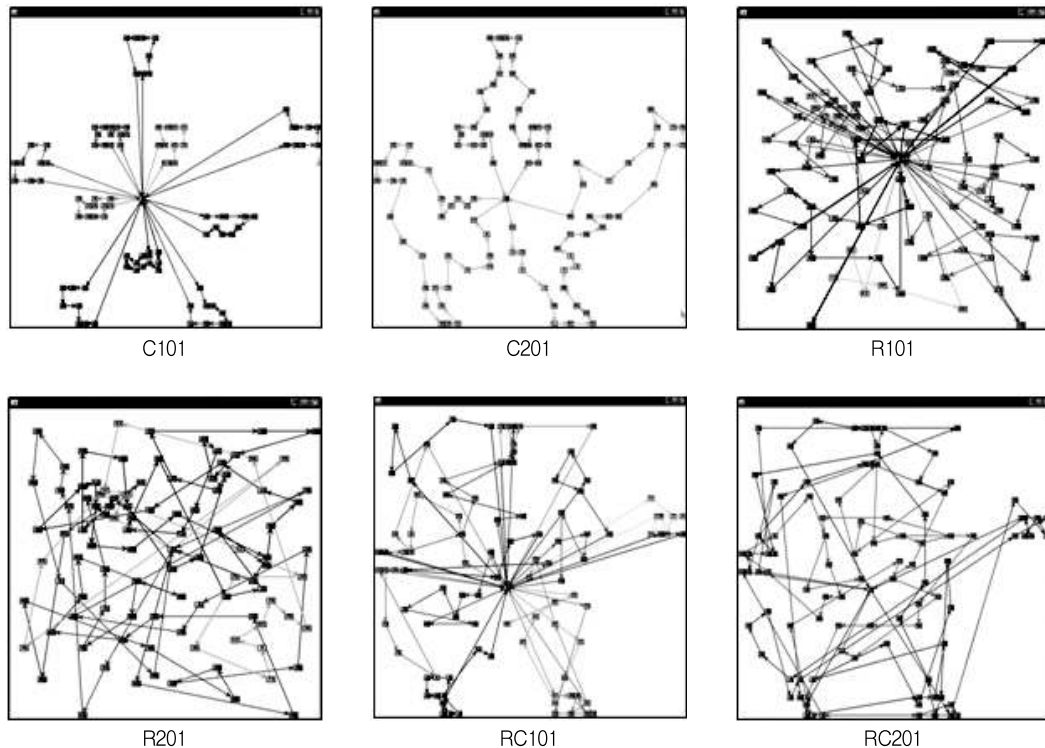
<표 3> 제안 타부 탐색 전략

```

PROCEDURE tabuSearch(path)
    //path : initial route
    Z ← path, Z* ← path;
    initialize tabu list, the number of iteration, and
    maximum tenure;
    while counter is less than iteration
    for moveScale ← 1 to the size Of Z
        while index i is less than the size of Z
            j = i+moveScale;
            optPath ← exchange the node of i and j
            on Z; //2-Opt
            if optPath exist on tabu list
                increase tenure by one on tabu list;
            else
                if optPath is not satisfied with the time
                constraint for each node
                    continue;
                if cost(optPath) < cost(bestPath)
                    save optPath to Z and bestPath;
                    i = i + 1;
            end-while /* for index */
        end-for /* for moveScale */
    if Z = Z*
        break; /* stop criterion of tabu */
    else if Z < Z*
        Z* ← Z;
    remove elements above maximum tenure from the
    tabu list;
    if the tabu list is full
        remove an element of the greatest tenure from
        the tabu list;
    add Z to tabu list, and initialize tenure with zero;
    increase counter by one;
    end-while
    return Z*
    
```

형으로 구분되며, 시간 제약의 범위가 짧고 차량 용적량이 작은 경우엔 1(C1, R1, RC1)로, 그렇지 않은 경우엔 2(C2, R2, RC2)로 표현된 6가지 유형을 가진다. 또한 각 유형별로 시간 제약의 출현 빈도가 25%, 50%, 75%, 100%인 데이터 집합을 가진다.

본 연구의 실험 환경은 Intel Core 2 Duo E7300 CPU, Window XP, RAM 2G이며, JAVA로 구현 하였다.



<그림 3> 라우트 스케줄 결과 그래프

<그림 3>은 본 연구의 실험에 대한 라우트 스케줄링 결과 그래프 중 일부이며, <표 4>는 Solomon II과 기회시간 평가 함수를 고려한 제안 휴리스틱에 대한 실험 비교 결과이다. 표시된 값은 각 유형별 모든 데이터 집합에 대한 평균값이다. 실험 결과는 비교적 지리적으로 임의 분산되어 있으며, 시간 제약 범위가 크고, 차량의 적재 용량이 큰 R2와 RC2형에서 Solomon II의 비용 평가함수보다 좋은 결과를 보였다. 이는 R2나 RC2와 같이 시간 제약을 고려해야 할 가변적 요인이 큰 경우에 기회시간을 고려한 비용 평가가 효과적임을 보여준다. 또한 비교적 빠른 실행시간 내에 해를 구하였다.

<표 5>는 초기 경로 생성 전략과 비용 평가 함수별로 각 데이터 유형에서의 최적값 산출 빈도를

보여준다. 삽입 휴리스틱에서 초기 경로를 생성하기 위한 고객 노드 선택 전략은 고객 노드의 지리적 분포와 시간 제약 조건에 따라 최적값 산출에 영향을 주었다. 일반적으로 C1과 R1형 유형에서는 ED와 FD가 유리하였으며, C2와 R2형 유형에서는 초기 경로 생성 전략 SC와 WO가 유리하였다. 반대로 RC1형에서는 SC와 WO가 RC2형에서는 ED와 FD의 최적값 빈도가 더 많았다. 비용 평가 함수에 있어서는 기회 시간이 고려된 비용 평가 함수를 적용한 경우, 모든 데이터 유형에 대하여 최적값을 얻었다. 하지만, 초기 경로 생성 전략에 따라 다양한 데이터 유형에 대한 최적값을 얻을 수는 있었으나, 동일한 초기 경로 생성 전략으로 모든 유형에 대한 최적값을 얻을 수는 없었다.

<표 4> Solomon I1과 제안 휴리스틱의 결과 비교

데이터 유형(개수)	Solomon I1				제안 휴리스틱				
	차량수	스케줄시간	이동거리	대기시간	차량수	스케줄시간	이동거리	대기시간	실행시간 (ms)
C1(9)	10	10104.2	951.9	152.3	10	10116.282	1041.374	74.908	279.4
C2(8)	3.1	9921.4	692.7	228.6	3.25	9944.041	706.408	237.633	447.1
R1(12)	13.6	2695.5	1436.7	258.8	14	2779.972	1573.628	206.345	110.7
R2(11)	3.3	2578.1	1402.4	175.6	3.182	2503.477	1376.257	127.219	539.9
RC1(8)	13.5	2775	1596.5	178.5	13.875	2966.488	1819.588	146.9	103.6
RC2(8)	3.9	2955.4	1682.1	273.2	3.625	2872.654	1672.208	200.446	384.6

주) 각 유형에서의 모든 데이터 집합에 대한 평균값임.

<표 5> 초기 경로 생성 전략 및 비용 평가 함수별 최적값 산출 빈도

유형		초기 경로 생성 전략				비용 평가 함수 전략	
번호	데이터 집합 개수	ED	FD	SC	WO	DC	OC
C1	9	4	5	0	0	0	9
C2	8	0	2	3	3	0	8
R1	12	1	7	2	2	0	12
R2	11	3	1	3	4	0	11
RC1	8	2	1	3	2	0	8
RC2	8	2	3	2	1	0	8

ED(Earliest DeadLine) : 고객 노드의 요구 종료시간이 가장 일찍 끝나는 고객 선택

FD(Farthest from Depot) : 거점에서 가장 멀리 떨어진 고객 선택

SC(Sweep by capacity) : 미할당 고객 중 sweep으로 형성된 클러스터의 중간에 위치한 고객 선택

WO(Wait and Opportunity Time) : 거점으로부터의(distance + wait time + opportunity time)이 가장 작은 고객 선택

DC(Default Cost) : $cost = travel\ Time\ Cost + wait\ Time\ Cost$

OC(Opportunity Cost) : $cost = travel\ Time\ Cost + wait\ Time\ Cost + opportunity\ Time\ Cost$

6. 결론

많은 계산 복잡성을 가지는 VRSPTW문제를 해결하기 위해, 기존 Solomon I1 휴리스틱(Solomon M., 1987)을 개선하여 기회 시간을 고려한 비용 평가 함수를 반영한 하이브리드 삽입 휴리스틱을 제안하고 그 효율성을 보였다.

기회시간은 각 고객 노드에서 요구되는 하역 서

비스를 수행 한 뒤에도 남아 있는 여유시간이다. 이 기회시간이 반영된 추가 제약조건과 비용 평가 함수를 삽입 전략에 우선 적용함으로써 경로 생성을 위해 요구되는 차량 수를 줄였으며, 각 생성된 경로들에 대해서는 기회시간을 평가하여 타부 탐색을 수행함으로써 해를 개선 시켰다. 개선된 해는 고객 노드의 지리적 위치가 비교적 임의 분산되어 있으며, 시간 제약은 범위가 크고, 차량이 처리

해야할 용량이 클 경우에 나타났다. 또한 각 데이터 유형에 대한 초기 경로 생성 전략들을 평가하고 각 유형에 효과적인 초기 경로 생성 전략들을 보였다. 이는 VRSPWTW 문제를 해결하기 위한 삽입 전략뿐만 아니라 시간 제약 평가 함수를 사용하여 휴리스틱에서 기회시간을 반영한 비용 평가함수는 해 개선 효과를 기대할 수 있을 것이다. 하지만, 기회 시간을 반영하여 해를 개선시킬 수는 있더라도 모든 데이터 유형에서 최적값을 찾을 수는 없다. 따라서 이를 극복할 수 있는 휴리스틱에 대한 연구나 빠른 시간 내에 해를 찾을 수 있는 완전 최적화 기법에 대한 연구가 필요하다.

참고문헌

- 고준택, 유영훈, 조근식, “한정 용량 차량 경로탐색 문제에서 이분시드 검출법에 의한 발견적 해법”, *한국지능정보시스템학회 지능정보연구 논문지*, 15권, 1호(2009), 1~12.
- Baker E. K. and Schaffer J. R., Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *American Journal of Mathematical and Management Sciences* Vol.6(1988), 261~300.
- Bard J. F., G. Kontoravdis, and G. Yu, “A Branch-and-Cut procedure for the vehicle routing problem with time windows”, *Transportation Science*, Vol.36, No.2(2002), 250~269
- Bodin L., B. Golden, A. Assad, and M. Ball, “Routing and Scheduling of Vehicles and Crews : The State of the Art. Comput”, *Opns. Res.* Vol.10(1983), 62~212.
- Christofides N., A. Mingozzi, and P. Toth, “The Vehicle Routing Problem. In Combinatorial Optimizations”, *John Wiley and Sons, New York*, 1979.
- Clarke G. and W. Wright, “Scheduling of Vehicles from a Central Depot to A Number of Delivery Points”, *Operational Research*, Vol.12(1964), 568~581.
- Desrochers M., J. Desrosiers, and M. M. Solomon, “A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows”, *Operations Research*, Vol.40(1992), 342~354.
- Fisher M., and R. Jaikumar, “A Generalized Assignment Heuristic for Vehicle Routing”. *Networks*, Vol.11(1981), 109~124.
- Gillet B., and L. Miller, “A Heuristic Algorithm for the Vehicle Dispatching Problem”, *Operations Research* Vol.22(1974), 340~349.
- Glover F., “Tabu Search Part I”, *ORSA Journal on Computing*, Vol.1, No.3(1989), 190~206.
- Glover F., “Tabu Search Part II”, *ORSA Journal on Computing*, Vol.2, No.1(1990), 4~32.
- Glover F., and E. Taillard, “A user’s guide to tabu search”, *Annals of Operations Research*, Vol.41 No.1(1993), 1~28.
- Ibrahim H. Osman, and James P. Kelly, *META-HEURISTICS : Theory and Applications*, Springer, 1 edition, March Vol.31, 1996.
- Irnich S. and Villeneuve D., The shortest path problem with k-cycle elimination ($k \geq 3$) : Improving a branch-and-price algorithm for the VRPTW. *INFORMS Journal of Computing* (2005).
- Kolen A., Rinnooy Kan A. H. G., and H. Trienekes, “Vehicle Routing with Time Windows”, *Operations Research*, Vol.35(1987), 266~273.
- Lau H. C., Sim M., and K. M. Teo, “Vehicle routing problem with time windows and a limited number of vehicles”, *Eur. J. Oper. Res.*, Vol.148(2003), 559~568.
- Lenstra, J. K., and Rinnooy Kan, A. H. G., “Complexity of vehicle routing and scheduling

- problems”, *Networks*, Vol.1(1981), 221~227.
- Or I., “Traveling Salesman-type Combinatorial Problems and their relation to the Logistics of Blood Banking”, Ph.D. thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL. 1976.
- Potvin J. Y. and J. M. Rousseau, “An Exchange Heuristic for Routing Problems with Time Windows”, *Journal of the Operational Research Society*, Vol.46(1995), 1433~1446.
- Savelsbergh M., Private Communication by Alexander Rinnooy Kan. 1984.
- Solomon Marius M., “Algorithms for the vehicle routing and scheduling problems with time window constraints”, *Operations Research*, Vol.35, Is., Vol.2(1987), 254~262.
- Taillard E. D., P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, “A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows”, *Te Vsportation Science*, Vol.31, No.2(1997), 170~186.

Abstract

Hybrid Heuristic Applied by the Opportunity Time to Solve the Vehicle Routing and Scheduling Problem with Time Window

Young-Hoon Yu^{*} · Sang-Jin Cha^{*} · Geun-Sik Jo^{**}

This paper proposes the hybrid heuristic method to apply the opportunity time to solve the vehicle routing and scheduling problem with time constraints(VRSPTW). The opportunity time indicates the idle time which remains after the vehicle performs the unloading service required by each customer's node. In this proposed heuristic, we add the constraints to VRSPTW model for the opportunity time. We also obtain the initial solution by applying the cost evaluation function to the insertion strategy considering the opportunity time. In addition, we improve the former result by applying the opportunity time to the tabu search strategy by swapping the customer's node. Finally, we suggest the construction strategies of initial routing which can efficiently acquire the nearest optimal solution from various types of data in terms of geographical condition, scheduling horizon and vehicle capacity. Our experiment show that our heuristic can get the nearest optimal solution more efficiently than the Solomon's I1 heuristic.

Key Words : VRSPTW(Vehicle Routing and Scheduling Problem with Time Window), Insertion Heuristic, Tabu Search, Hybrid Heuristic, Opportunity Time, 2-Opt Algorithm

* School of Information Engineering, Inha University

** School of Computer and Information Engineering, Inha University

저자 소개



유영훈

관동대학교 전자계산공학 학사, 인하대학교 전자계산공학 석사를 취득하였으며, 동 대학원의 정보공학과 박사과정을 수료하였다. 현재 인하대학교 컴퓨터정보공학과 강의전임교원으로 재직 중이다. 주요 연구분야는 인공지능, CSP, Constraint Programming, Meta-Heuristics, Intelligent Agent 등이다.



차상진

인하대학교 컴퓨터공학부 학사(2006), 인하대학교 대학원 정보공학과 석사(2009) 학위를 취득하였고, 현재 인하대학교 대학원 정보공학과 박사과정에 재학 중이다. 주요 관심분야는 Semantic Web, Agent, Knowledge Management, Collective Intelligence, Web Service, Meta-Heuristics 등이다.



조근식

인하대학교 전자계산학 학사, 미국 뉴욕 CUNY(City University of New York) 전자계산 석사와 박사를 취득하였다. 한국지능정보시스템학회 편집장(1998), 인하대학교 창업지원센터 소장(2000), 인하대학교 전산정보원장(2005), 한국지능정보시스템학회 회장(2008) 등을 역임하였다. 현재 인하대학교 컴퓨터정보공학부 교수로 재직하고 있으며, BK21 정보기술 사업 단장과 WCU 디지털 스토리텔링 플랫폼 사업 단장을 역임 중이다. 주요 연구분야는 인공지능, CSP, Intelligent E-Commerce System, Semantic Web 등이며, AI Magazine, Expert System with Application, Journal of Organizational Computing and Electronic Commerce, LNCS, LNAI 등 다수의 학술지에 논문을 게재하였으며, 다수의 특허를 보유하고 있다.