

차량 경로 스케줄링 문제 해결을 위한 개미 군집 최적화 휴리스틱

홍 명 덕[†] · 유 영 훈^{**} · 조 근 식^{***}

요 약

차량 경로 스케줄링 문제(VRSPTW, the Vehicle Routing and Scheduling Problem with Time Window)는 여러 고객의 시간 제약과 요구량을 만족시키면서 최소 이동 비용을 가지는 경로를 구성하는 문제이다. 이 문제는 NP-Hard 문제이기 때문에 해를 산출하는데 시간이 오래 걸린다. 본 연구는 VRSPTW를 빠른 시간 내에 최근사해를 구하기 위한 멀티 비용 함수(Multi Cost Function)를 갖는 개미 군집 최적화(Ant Colony Optimization)을 이용한 휴리스틱을 제안하였다. 멀티 비용 함수는 각 개미가 다음 고객 노드로 이동하기 위해 비용을 평가할 때 거리, 요구량, 각도, 시간제약에 대해 서로 다른 가중치를 반영하여 우수한 초기 경로를 구할 수 있도록 한다. 본 연구의 실험결과에서 제안된 휴리스틱이 Solomon II 휴리스틱과 기회시간이 반영된 하이브리드 휴리스틱보다 효율적으로 최근사 해를 얻을 수 있음을 보였다.

키워드 : 차량 경로 문제, 조합 최적화, 개미 군집 최적화, 메타 휴리스틱

An Ant Colony Optimization Heuristic to solve the VRP with Time Window

Myung-Duk Hong[†] · Young-Hoon Yu^{**} · Geun-Sik Jo^{***}

ABSTRACT

The Vehicle Routing and Scheduling Problem with Time Windows(VRSPTW) is to establish a delivery route of minimum cost satisfying the time constraints and capacity demands of many customers. The VRSPTW takes a long time to generate a solution because this is a NP-hard problem. To generate the nearest optimal solution within a reasonable time, we propose the heuristic by using an ACO(Ant Colony Optimization) with multi-cost functions. The multi-cost functions can generate a feasible initial-route by applying various weight values, such as distance, demand, angle and time window, to the cost factors when each ant evaluates the cost to move to the next customer node. Our experimental results show that our heuristic can generate the nearest optimal solution more efficiently than Solomon II heuristic or Hybrid heuristic applied by the opportunity time.

Keywords : Vehicle Routing Problem, Combinatorial Optimization, Ant Colony Optimization, Meta Heuristic

1. 서 론

현대 산업사회에서 물류시장의 규모가 커지고 소비자의 요구수준이 높아짐에 따라 기업에서 차지하는 물류비용이 전체 비용 중에 상당수 차지하게 되었다. 따라서 기업들은 물류비용의 최소화를 위해 효과적인 배송정책을 수립하여 운영하는 것이 중요하다. 효과적인 배송정책이란 전체 운영 차량의 수와 운행거리를 최소화하며, 차량의 적재용량을 최

대한 활용하는 것이다.

이와 같이 각 고객의 시간 제약과 요구량을 만족시키면서 최소 이동 비용을 가지는 경로를 구성하는 문제를 차량 경로 스케줄링 문제(VRSPTW, Vehicle Routing and Scheduling Problem with Time Window)라고 한다[1]. 이 문제는 문제의 크기가 커질수록 계산 량이 지수 적으로 증가하게 되는 NP-Hard 문제이다[2, 3]. 이러한 문제의 해법은 완전 최적화 기법(Exact Optimization Method)과 휴리스틱 기법(Heuristic Method)으로 크게 구분된다. 완전 최적화 기법은 모든 가능한 경로를 조사해보는 방법으로 문제에 따라 최적 해를 구하는 시간이 오래 걸리거나, 최적 해를 구한다는 보장이 없을 수도 있다. 휴리스틱 기법은 모든 가능

[†] 준 회원: 인하대학교 정보공학과 석사과정
^{**} 정 회원: 인하대학교 컴퓨터정보공학과 전임강사
^{***} 총신회원: 인하대학교 컴퓨터정보공학과 교수
논문접수: 2010년 7월 28일
수정일: 2010년 9월 13일
심사완료: 2010년 9월 14일

한 경로를 조사하지 않고 다양한 지역 탐색(Local Search) 기법을 제어하여 여러 해를 생성하고 지능적으로 정보를 구축하여 최근사 해를 도출하는 방법으로 최근사 해(Nearest Optimal Solution)를 효율적으로 얻는 것을 목적으로 한다.

차량 경로 스케줄링 문제를 해결하기 위한 휴리스틱으로 Solomon의 삽입 기법(Insertion Heuristic)이 제안된 이후로 많은 해법들이 연구되고 있다.

Solomon의 삽입 기법은 특정한 기준 값을 계산하여 일부 구성된 부분경로에 아직 포함되지 않은 고객지점 들을 하나씩 추가하면서 해를 완성하는 기법이다[4].

Garcia et al.은 타부 탐색 기법(Tabu Search Heuristic)을 통한 해법을 제안하였다. 타부 탐색은 국부 지역 탐색(Local Search)에 빠지지 않기 위해 타부 목록(Tabu List)과 열망수준(Aspiration Criteria)에 의해 이웃 탐색을 관리하고 종료 조건이 만족할 때까지 반복 한다[5].

Thangiah et al.은 유전 알고리즘(Genetic Algorithm)을 통한 해법을 제안하였다. 유전 알고리즘은 문제를 염색체의 형태로 표현하고, 각 염색체가 문제를 해결하는데 얼마나 좋은지를 측정하기 위한 평가함수(Evaluation Function)와 유효한 해인지를 판단하는 적합함수(Fitness Function)를 결정하여, 매 세대마다 모집단(Population)을 운용하는 기법이다[6, 16].

Taillard et al.은 삽입 기법과 타부 탐색 기법을 사용하는 해법을 제안하였다. 이 방법은 타부 탐색 시에 초기 해를 삽입 기법으로 구한 후 하위 경로의 집합으로 나눈 뒤, 두 개의 하위 경로 간에 두 고객 노드들을 교환함으로써 해를 개선시켰다[7, 16].

Nazif et al.은 유전 알고리즘 기반의 OCGA(Optimized Crossover Genetic Algorithm)을 사용하는 해법을 제안하였다. 이 방법은 유전 알고리즘 내에서 선택(Selection) 연산자 다음에 최적화된 교차(Optimized Crossover) 연산을 수행하고 노드 교환을 수행하여 해를 개선시켰다[17].

El-Sherbeny은 이웃 해를 탐색하여 해를 향상 시킬 수 있는 여러 연산자(Operator)를 소개하고 있다. Relocate 연산자는 하나의 고객을 다른 차량으로 이동하여 이웃 해를 탐색하고, Exchange 연산자는 2개의 차량에서 2개의 고객을 서로 교환하여 이웃 해를 탐색하고, 2-Opt* 연산자는 차량의 한 부분을 다른 차량의 부분과 교환하여 이웃 해를 탐색하고, λ -interchange 연산자는 λ 개의 고객 노드 집합을 다른 차량의 고객 노드 집합과 교환하여 이웃 해를 탐색한다. 또한 그 외에 연산자들을 소개하고 있다[15, 18].

본 연구에서는 개미 군집 최적화(ACO, Ant Colony Optimization) 기법을 사용하여 VRSPTW를 해결하고자 하였으며, 실험을 통해 우수한 해를 도출할 수 있음을 보였다.

2. 관련 연구

2.1 삽입 휴리스틱 응용

Solomon은 절약 알고리즘과 최근접 이웃 탐색 휴리스틱

에 의해 경로가 구성될 때 시간 제약 조건을 고려하는 기법이다[4, 16]. 삽입 기법은 일반적으로 다른 알고리즘들에 비해 빠르게 해를 구축할 수 있으나, 고객 노드를 선택하는 전략에 따라 생성된 해의 결과가 달라질 수 있으며, 지역 해에 빠질 수도 있다.

유영훈 외 2명은 기회시간을 반영한 2-opt기반의 타부 탐색 전략을 삽입 휴리스틱에 적용함으로써 Solomon의 삽입 휴리스틱의 단점을 극복 하였다[19]. 각 4가지의 초기 경로 생성 전략과 2가지 비용 평가 함수를 제안하여 삽입 전략에 의해 수행된 초기 해를 타부 전략을 사용하여 해를 개선시키고, 개선된 이웃 해들 중 가장 좋은 해를 최종 해로 선정하는 전략이다.

2.2 개미 군집 최적화

개미 군집 최적화는 개미들의 행동에 대한 메커니즘을 응용한 기법으로서 개미들이 정보를 교환할 때 사용하는 페로몬(Pheromone)을 이용하는 방법을 적용한 메타휴리스틱(Meta Heuristic)이다. 즉, 개미들이 먹이를 찾아 이동할 때, 자신이 이동한 경로 위에 페로몬을 분비하게 되며 다음 개미는 페로몬 흔적(Pheromone Trail)을 이용하여 다음 이동 경로를 선택하게 된다. 이때 자주 이용되는 경로는 페로몬이 축적되고, 이용되지 않는 경로는 페로몬이 감소하게 된다. 이렇게 축적된 페로몬 정보를 바탕으로 구성된 경로는 최적의 경로일 확률이 높기 때문에 이를 해로 산출하게 된다. 차량 경로 문제를 풀기 위해 이러한 개미 군집 최적화 기법을 응용한 연구들은 다음과 같다.

Bullnheimer et al.은 처음으로 개미 군집 최적화를 적용하여 인공 개미에 의한 초기경로구성 후 지역탐색 기법을 적용한 HAS(Hybrid Ant System)를 제안하였다[8].

Gambardella et al.은 두 개의 개미군집이 독립적으로 활동하여 각각 총 차량수와 총 차량 운행 거리를 최소화 하여 서로 정보를 교환하는 MACS-VRPTW(Multiple Ant Colony System for VRPTW)를 제안하였다[9].

Reimann et al.은 절약 알고리즘(Saving Algorithm)으로 초기경로구성 후 Sweep 알고리즘을 사용하여 임의로 여러 개의 하위문제로 분할하여 각각의 나뉜 문제를 다시 절약 알고리즘과 지역탐색을 이용하는 방법을 제안하였다[10].

Chengming et al.은 임의의 초기경로를 구하고 파레토 지역 탐색(Pareto Local Search)을 통해 해를 향상시키는 RPACS-VRPTW를 제안하였다[11].

또한 페로몬 흔적을 갱신하는 방법으로 Gambardella et al.은 페로몬 흔적 갱신 시에 지역 페로몬과 전역 페로몬으로 페로몬 흔적을 두 번 갱신하였고[9], 마찬가지로 이승관 외 1명이 외판원 문제(Travelling Salesman Problem; TSP)를 해결하기 위해 페로몬 흔적을 지역 갱신과 추가 강화값을 이용한 전역 갱신으로 나누어 갱신하였다[21].

기존 연구에서는 초기 경로를 먼저 구성하고, 반복(Iteration)적인 경로 구성을 통해 초기 해를 향상시키는 전략을 사용한다[8, 9, 11]. 또한 수행 속도 향상과 국부 지역

해(Local Minima)에서 벗어나기 위해, 일반적인 개미 군집 최적화 전략은 임의(Random)로 초기 경로를 구성한다. 하지만, 이는 부적절한 해가 페로몬 흔적에 적용될 수 있는 단점이 있다. 이를 극복하기 위해 다양한 해를 많이 구하여 페로몬 흔적을 갱신한다. 그에 따라 인공개미의 수와 경로 구성을 반복하는 과정이 많이 요구하게 된다.

본 연구에서는 홍명덕 외 2명이 제안한 개미 군집 최적화 기법을 기반으로 하여 초기 경로 구성 시 최근접 이웃 휴리스틱(NNH, Nearest Neighbor Heuristic)을 사용하고, 미 할당 노드를 선택 할 때 멀티 비용 함수(Multi Cost Function)를 통해 가장 높은 비용을 가진 노드를 선택한다[20]. 하지만 초기해 생성 시에 임의로 비용 가중치(Cost Weight)를 선택하기 때문에 매 해가 달라질 수 있는 단점이 존재한다. 따라서 본 연구에서는 우수한 해를 얻기 위해 비용 가중치를 적용하되 매 우수한 해를 도출 할 수 있도록 초기 사용될 비용 가중치를 별도로 지정하고 이후 임의로 비용 가중치를 선택하여 초기 해를 구축하는 전략을 사용하였다. 이로 인해 부적절한 해가 나오는 것을 방지하며, 반복 과정 시 페로몬 갱신 규칙을 적용함으로써 적은 수의 인공개미와 반복 과정을 통해 좋은 최적 근사 해를 구할 수 있다.

마지막으로 제안하는 휴리스틱의 평가는 삽입 휴리스틱과 기회시간 반영된 하이브리드 휴리스틱의 실험 결과를 이용하여 비교하고자 한다.

3. 시간 제약을 가지는 차량 경로 스케줄링 문제

3.1 문제 정의와 기본 제약 조건

VRSPWTW는 각 고객의 요구량과 이동거리와 함께 거점과 각 고객의 시간 제약을 준수하여 경로를 구성해야 하는 문제이다. 구성된 경로는 방향성 그래프(Direct graph) $G=(V, E)$ 로 나타낼 수 있다. 이때 모든 고객 노드들은 경로에 포함되어야 하며 오직 한번만 방문될 수 있다. V 는 고객 노드들의 집합이며, E 는 v_i 와 v_j 사이의 연결 간선(edge)들의 집합이다.

V : 고객 노드 v_i 의 집합, $i=1 \dots n$, $v_i \in V$,

단 v_0 는 거점,

E : 거점과 모든 고객 간 간선들의 집합,

$E = \{(v_0, v_1), (v_1, v_0), \dots, (v_{i-1}, v_i), (v_i, v_{i-1})\}$

n : 고객의 수

i, j : 거점과 고객 노드,

$(i=0, 1 \dots n), (j=0, 1 \dots n)$

단일 거점을 가지는 경로 $path$ 는 방향성 그래프 G 에서 거점(depot)에서 출발하여 고객 노드들을 순서대로 방문하고 거점으로 도착하는 순서를 나열함으로써 나타낼 수 있다. 구성된 경로 $path_k$ 는 차량 t_k 에 의해 운행하게 되며, 경로 상에서 첫 번째 방문지 v_0 와 마지막 방문지 v_n 는 거점과 동

일하다.

T : 차량 t_k 의 집합, $k=1 \dots n$

t_k^{capa} : 차량 t_k 의 적재 용량

u : 경로 상의 고객 노드 수

$path_k$: 경로, $k=1 \dots n$,

$\{path_k = t_k, (k=1 \dots n)$

$\{path_k = \{v_0, \dots, v_u, v_n\}, (v_u \in V, 1 \leq u \leq n)$

$depot = v_0 = v_n$: 거점과 처음 방문지와 마지막 방문지는 동일

이 문제에서 사용가능한 최대 차량 대수는 주어진 고객 노드 수 n 만큼 사용 할 수 있다. 그리고 모든 차량의 적재 용량을 동일하며, 식 (1)와 같이 구성된 경로에서의 고객의 요구량(Demand)의 총 합이 차량의 적재 용량 t_k^{capa} 보다 클 수 없다.

v_n^{demand} : 고객 노드 v_n 의 서비스 요구량

$t_1^{capa} = t_2^{capa} = t_3^{capa} = \dots = t_k^{capa}, (1 \leq k \leq n)$

$$t_k^{capa} \geq \sum_{i=1}^n path_{k,i}^{demand} \quad (1)$$

경로 $path_k$ 를 구성하기 위해서는 거점과 각 고객에 대한 시간 제약인 TW_i 내에 서비스를 제공해야만 한다. 따라서 식 (2)와 같이 이전 고객 노드 $i-1$ 에서 출발하여 i 에 도착할 경우, 고객 i 로의 도착시각 $path_{k,i}^{arrival}$ 는 고객 노드 i 의 서비스시작시각 v_i^{start} 보다 일찍 도착한다면 고객 노드 i 의 서비스시작시각 v_i^{start} 가 되며, 그렇지 않다면 고객 노드 $i-1$ 의 도착시각 $path_{k,i-1}^{arrival}$ 과 $i-1$ 의 서비스시간 $v_{i-1}^{service}$, 두 고객 노드 $i-1$ 과 i 사이의 이동 시간 $tr_{i-1,i}$ 을 더한 시간이 된다. 여기서 이동 거리와 이동 시간은 동일하게 가정한다.

v_i^{start} : 고객 노드 i 의 서비스 시작 시각

v_i^{end} : 고객 노드 i 의 서비스 종료 시각

TW_i : 고객 노드 i 의 시간 제약(Time window),

$TW_i = [v_i^{start}, v_i^{end}]$

$path_{k,i}^{arrival}$: 경로 $path_k$ 상의 고객 노드 i 의 도착시각

$path_{k,i}^{departure}$: 경로 $path_k$ 상의 고객 노드 i 의 출발시각

$path_{k,i}^{wait}$: 경로 $path_k$ 상의 고객 노드 i 의 대기시간

$v_i^{service}$: 고객 노드 i 의 서비스 시간

$tr_{i,j}$: 고객 노드 i 와 j 사이의 이동시간(이동거리)

$$path_{k,i}^{arrival} = \max \left\{ v_i^{start}, path_{k,i-1}^{arrival} + path_{k,i-1}^{wait} \right. \\ \left. + v_{i-1}^{service} + tr_{i-1,i} \right\} \quad (2)$$

$$v_i^{start} \leq path_{k,i}^{arrival} \leq v_i^{end}$$

경로 $path_k$ 상의 고객 i 의 대기시간 $path_{k,i}^{wait}$ 는 식 (3)과 같이 TW_i 상의 서비스시작시각 v_i^{start} 보다 차량이 일찍 도착할 경우에는 v_i^{start} 에서 도착시각 $path_{k,i}^{arrival}$ 을 뺀 시간이지만, 차량이 서비스시작시각 v_i^{start} 이후에 도착하는 경우에는 0이다.

$$path_{k,i}^{wait} = \max\{0, v_i^{start} - path_{k,i}^{arrival}\} \quad (3)$$

고객 i 의 출발시각 $path_{k,i}^{departure}$ 은 서비스시간 $v_i^{service}$ 으로 인해 TW_i 의 v_i^{end} 가 벗어나는 것을 허용하며, 이는 식 (4)와 같이 도착 시각 $path_{k,i}^{arrival}$ 과 대기 시간 $path_{k,i}^{wait}$, 서비스 시간 $v_i^{service}$ 의 합이라 할 수 있다.

$$path_{k,i}^{departure} = path_{k,i}^{arrival} + path_{k,i}^{wait} + v_i^{service} \quad (4)$$

그리고 거점 v_0 에서의 종료시간 v_0^{end} 내에 되돌아와야 하기 때문에, 식 (5)와 같이 마지막 고객 노드에서의 출발시간 $path_{k,u}^{departure}$ 과 다시 거점까지의 이동시간 $tr_{u,0}$ 을 합한 시간보다 크거나 같아야 한다.

$$v_0^{end} \geq (path_{k,u}^{departure} + tr_{u,0}) \quad (5)$$

최종적으로 식 (6)와 같이 구성된 모든 경로 $path_k$ 상의 고객의 수 u 의 합은 고객의 수 n 와 같아야 한다.

$$\sum_{k=1}^p path_k^u = n, \quad (p \text{는 구성된 경로의 수}) \quad (6)$$

3.2 목적 함수

본 연구의 목적 함수는 모든 고객에게 서비스를 수행 할 수 있는 차량 수와 함께 총 운행 시간, 총 이동거리를 최소화하도록 설계하는 것이다. 이들 목적함수에는 차량 수, 총 운행시간, 총 이동거리의 순서대로 우선순위를 가진다. 이것은 총 운행시간이 증가되었더라도 차량 수가 감소되었다면

해가 더 향상되었다는 것을 말한다. 식 (7)은 차량 수에 대한 목적함수이다.

$$\min(\sum_{k=1}^p t_k), \quad (1 \leq p \leq n) \quad (7)$$

식 (8)은 총 운행시간에 대한 목적함수이다. 여기서 $path_k$ 는 k 번 차량의 경로이고 u 는 경로상의 고객노드의 개수이다. 즉, 각 차량의 마지막 도착시간들에 대한 합을 최소화하는 것이다.

$$\min(\sum_{k=1}^p path_{k,u+1}^{arrival}), \quad (1 \leq p \leq n) \quad (8)$$

식 (9)은 총 이동거리에 대한 목적함수이다. 여기서 u 는 해당 차량 경로 상의 고객의 수이다.

$$\min(\sum_{k=1}^p \sum_{u=0}^z tr_{u,u+1}), \quad (1 \leq p \leq n, 1 \leq u \leq n) \quad (9)$$

4. VRSPTW를 위한 개미 군집 최적화 휴리스틱

4.1 제안 휴리스틱

(그림 1)은 개미 군집 최적화의 장점인 전역적 탐색정보 능력을 이용하여 해를 유도하는 최근접 이웃 휴리스틱을 적용한 알고리즘이다.

단계 1은 초기화 단계로서, 알고리즘 실행을 위해 비용가중치 목록(cwl)을 $CostWeightList()$ 을 이용하여 구하고, 초기 페로몬 흔적($pheromone\ trail$)을 초기화 한다.

단계 2-1은 초기 경로(lb)를 구성 단계로서, $InitialSolution()$ 함수를 이용하여 미할당된 고객 노드들을 각 개미에 포함될 수 있는지 제약조건을 검사하여 포함 가능한 경우, 비용 함수를 통해 각 개미별 삽입 비용을 구한다. 구해진 비용 중 높은 비용을 갖는 노드를 결정하여 해당 개미의 경로에 포함하는 과정을 반복적으로 수행하여 경로를 구성한다. 즉, 각각의 개미들은 하나의 차량과 동일하다.

```

/* An ACO Algorithm for solving VRSPTW */
[Phase 1] Initialization
Initialize  $gb$ , which is a variable to store the best feasible solution, to  $\infty$ .
Create a  $pheromone\ trail$  of 2-dimensional array with the size of  $(n+1) \times (n+1)$ , and initialize this array to zero.
Create a  $cwl$ , which means the cost weight list, by calling a  $CostWeightList()$  function.

[Phase 2] Solution improvement
Repeat the following steps until  $cwl$  is empty.
Step 2-1. Choose a  $cw$  from  $cwl$  randomly, and remove a chosen  $cw$  from  $cwl$ .
        Generate a solution  $lb$  by calling  $InitialSolution(cw)$  function.
Step 2-2. Generate a improved solution  $lb^*$  by calling  $ImproveAlgorithm(lb)$ ,
        which improve  $lb$  by using the 2-Opt and relocate[12, 13].
Step 2-3. If  $lb^*$  is better then  $gb$ , replace  $gb$  with  $lb^*$ .
Step 2-4. Update the  $pheromone\ trail$  with  $lb^*$  and  $gb$ .
Return the best feasible solution  $gb$ .
    
```

(그림 1) 제안 알고리즘

단계 2-2는 경로 개선하는 단계로서, *Improve Alogrithm* () 함수를 이용하여 개미들이 구축한 경로를 2-opt와 Relocate 를 수행하여 초기해의 이웃해 집단을 생성 후 가장 우수한 이웃 해를 개선된 경로(*lb**)로 대체한다.

단계 2-3은 개선된 경로와 현재 까지 가장 우수 해(*gb*)를 차량 수, 총 운행 시간, 총 이동 거리를 비교하여 더 적은 차량, 운행 시간, 이동 거리를 이용한 해를 현재 까지 발견 된 우수 해로 대체한다.

단계 2-4는 페로몬 갱신 단계로서, 전 단계에서 개선된 경로와 현재까지 가장 우수한 해를 이용하여 페로몬 흔적을 갱신 한다.

마지막으로 종료조건 확인 단계로서, 멀티 비용 가중치 목록 개수를 모두 소진할 때까지 단계 2를 반복 수행한다.

다음은 각 단계에 대한 세부적인 설명이다.

4.2 초기화

초기 페로몬이 존재하지 않는 페로몬 흔적을 구성하고 경로 구성 시에 사용될 비용 가중치를 생성하는 과정이다. 페로몬은 두 고객 노드 사이에 값을 누적하므로 거점을 포함한 $n+1 \times n+1$ 의 2차원 배열을 페로몬 흔적으로 생성한다. 비용 가중치(*cw*)는 거리(α), 요구량(β), 시간(γ), 각도(δ)에 대한 가중치이다. 생성 방법은 다음과 같다. 우선 비용 가중치 목록을 저장할 수 있는 기억 장소를 생성하고, 각각 4개의 비용 가중치에 대해 0.4, 0.7, 1.0의 값을 갖도록 하여 반복하되 매 반복 시에 $\alpha, \beta, \gamma, \delta$ 을 포함한 하나의 비용 가중치 목록으로 저장한다. 이러한 과정을 총 3번 반복하게 되면 비용 가중치 목록의 수가 243개가 된다. 생성된 비용 가중치 목록은 이후 매 초기 경로 구성 시에 이용되어 다양한 해를 생성할 수 있으며, 경로 구성의 반복 횟수는 입력 된 비용 가중치 목록의 개수와 동일하게 된다.

4.3 초기경로구성에서의 비용정의

본 연구에서는 초기해 생성 시 거리, 요구량, 시간, 각도 비용을 고려하여 고객 노드를 경로 상에 추가해 가며 초기 해를 구축해 나간다.

첫 번째는 거리 비용으로 거점과 고객 노드의 거리 또는 고객 노드와 고객 노드 사이의 거리를 유클리디안 거리 (Euclidean distance)를 이용하여 구한다[14]. 거리가 가까울 수록 이동 시간(이동 거리)이 감소되기 때문에 거리가 가까울수록 유리하도록 하였다. 두 고객 노드 *i*와 *j*의 거리는 식 (10)을 통해 구할 수 있다.

v_i^y : 고객 노드 *i*의 *y*좌표

v_i^x : 고객 노드 *i*의 *x*좌표

$$tr_{i,j} = \sqrt{(v_i^x - v_j^x)^2 + (v_i^y - v_j^y)^2} \quad (10)$$

두 번째는 요구량 비용으로 각 고객 노드들은 주문정보에 따라 배송요구량이 정해져 있으며 차량은 배송 가능한 최대 적재량이 정해져 있다. 차량의 적재량을 넘지 않으면서 고

객의 배송요구량을 최대한 처리하기 위하여 배낭 문제를 해결하는 방법과 같이 고객의 배송요구량이 큰 것부터 점차 작은 순으로 차량에 할당을 하여 고객의 배송요구량을 최대한 처리할 수 있다면 사용되는 차량의 수를 감소시킬 수 있기 때문에 고객의 배송요구량이 클수록 유리하도록 하였다.

세 번째는 시간 비용으로 거점은 배송이 가능한 시간 제약이 존재하고, 고객 노드들은 배송되기를 원하는 시간 제약이 존재한다. 차량은 반드시 거점의 배송 시작 시간에 출발하여 배송 종료 시간 내에 도착하여야 하고, 고객 노드들의 배송 가능 시간 내에 도착하여야 한다. 만약 차량이 고객 노드들의 배송 가능 시간 이후에 도착한다면 제약조건을 위반하게 되어 해를 구할 수 없고, 배송 가능 시간 이전에 도착한다면 고객 노드들의 배송 가능 시간까지 대기시간으로 대기 한 후 배송하게 된다. 그러므로 상대적으로 같은 거리에 있다 하더라도 대기시간이 있는 고객 노드로 배송하기 보다는 대기시간이 없는 고객 노드로 배송하는 것이 일반적으로 좋다. 그러므로 서비스 가능 시간대가 좁거나, 서비스 시작 시각이 빠르거나, 서비스 종료 시각이 빠를수록 유리하도록 하였다.

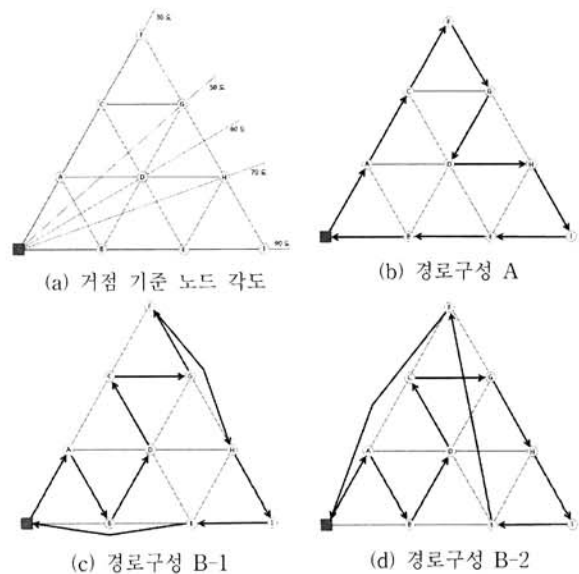
네 번째는 각도 비용으로 모든 고객 노드들은 식 (11)과 같이 거점을 기준으로 위치에 따른 각도 정보 v_i^c 을 가지게 된다. 그에 따라 특정 고객 노드에서 다음 고객 노드로 이동할 위치를 선택하고자 할 때 각도의 차가 적은 노드로 선택을 유도한다.

v_i^c : 고객 노드 *i*의 거점(v_0) 기준 각도

$$v_i^c = -1 \cdot (Atan2(v_i^y - v_0^y, v_i^x - v_0^x)) \cdot 180/\pi, \quad (11)$$

$$(0 \leq u_i^c < 360)$$

(그림 2)와 같이 9개의 고객 노드를 가지는 문제를 가정하여 각기 다른 기준에 따라 다음에 이동할 노드를 선택하



(그림 2) 각도 차에 따른 노드 선택에 대한 경로

여 보았다. (그림 2)의 (a)와 같이 9개 노드 중 A, C, F는 거점기준 30도에 위치해 있고, G는 50도, D는 60도, H는 70도, B, E, I는 90도에 위치되어 있다 그리고 서로 인접해 있는 노드와의 거리는 일정한 경우를 가정하여 2가지 경로 구성을 진행하였다. (그림 2)의 (b)는 현재 고객 노드에서 다음 고객 노드로 이동시 가장 가까운 거리의 노드를 선택하고 만약 같은 거리의 노드가 2개 이상인 경우 각도의 차가 적은 쪽으로 경로를 구성하였다. 그 결과 (거점→A→C→F→G→D→H→I→E→B→거점)로 경로가 구성되었다. 각 인접한 고객 노드 간의 거리를 1이라고 할 때, 구성된 경로의 거리는 10으로 이 문제에서 나올 수 있는 최적 해를 구할 수 있었다. (그림 2)의 (c)와 (d)는 현재 고객 노드에서 다음 고객 노드로 이동시 가장 가까운 거리의 노드를 선택하고 만약 같은 거리의 노드가 2개 이상인 경우 각도의 차가 큰 쪽으로 경로를 구성하였다. 그 결과 각 경로의 거리는 (그림 2)의 (c)는 12, (그림 2)의 (d)는 13.7로 경로가 구성되었다. 두 가지 경로구성을 볼 때 동일 거리의 여러 고객 노드 중 각도차가 적은 고객 노드를 선택하는 것이 유리하기 때문에 제안 휴리스틱은 이를 반영하였다.

4.4 초기 경로 구성 전략

본 연구에서는 멀티 비용 함수가 적용된 최근접 이웃 휴리스틱을 이용하여 적은 수의 개미로 개미 군집 최적화의 특징을 사용하는 방법을 제안한다.

여러 가지 구성형 해법 중에서 비교적 개념이 쉬우면서 빠르게 해를 구축할 수 있는 효과적인 기법인 최근접 이웃 휴리스틱을 이용하며 다음에 방문할 노드를 결정하는 비용 함수를 설계하여 그 비용들 중 가장 큰 비용을 갖는 노드를 방문하는 과정을 반복하여 해를 구축한다.

우선 앞서 초기화 단계에서 생성한 비용 가중치 목록에서 임의로 하나의 비용 가중치를 가져와 각각 $\alpha, \beta, \gamma, \delta$ 에 대입하고 선택한 비용 가중치를 다음 초기 경로 생성 시에 사용되지 못하도록 비용 가중치 목록에서 제외한다. 단, 맨 처음 초기 경로 구성 시에는 $\alpha, \beta, \gamma, \delta$ 값을 모두 1로 대입하고 진행한다.

다음은 (그림 1)의 *InitialSolution()* 함수를 이용하여 초기 경로를 생성하는 과정을 설명한다. 처음 경로를 구축하거나 미할당된 노드를 기존 개미에 할당 할 수 없을 때, 미할당된 노드를 지정하여 새로운 개미를 추가하여야 한다. 이 때 미할당된 노드들에 대해 식 (12)를 이용하여 비용을 산출한다.

$$DtNCost_j = DC_{0,j} + WC_j + TC_j + \tau_{0,j} \quad (12)$$

$DtNCost_j$ 는 거점에서부터 노드 j 사이의 비용이다. $DC_{0,j}$ 는 거리 비용이며 식 (13)을 이용하고, WC_j 는 요구량 비용으로 식 (14)을 이용하고, TC_j 는 시간제약 비용으로 식 (15)를 이용하고, $\tau_{0,j}$ 는 거점에서 노드 j 사이의 페로몬의 양이다.

$$DC_{0,j} = (1 - \frac{tr_{0,j}}{\sum_{m \in \text{unsigned node}} tr_{0,m}}) \cdot \alpha, (0 < \alpha < 1) \quad (13)$$

$DC_{0,j}$ 는 거점에서 노드 j 사이의 거리 비용이고, $tr_{0,j}$ 는 거점에서 노드 j 사이의 거리이고, m 는 미할당된 노드 집합의 일부이고, α 는 거리 비용 가중치이다.

$$WC_j = (\frac{v_j^{\text{demand}}}{\sum_{m \in \text{unsigned node}} v_m^{\text{demand}}}) \cdot \beta, (0 < \beta < 1) \quad (14)$$

WC_j 는 노드 j 에 대한 요구량 비용이고, v_j^{demand} 는 노드 j 의 서비스 요구량이고 m 는 미할당된 노드 집합의 일부이고, β 는 요구량 비용 가중치이다.

$$TC_j = ((1 - \frac{v_j^{\text{start}}}{v_0^{\text{end}} - v_0^{\text{start}}}) + (1 - \frac{v_j^{\text{end}}}{v_0^{\text{end}} - v_0^{\text{start}}})) + (1 - \frac{v_j^{\text{end}} - v_j^{\text{start}}}{v_0^{\text{end}} - v_0^{\text{start}}}) \cdot \gamma, (0 < \gamma < 1) \quad (15)$$

TC_j 는 노드 j 에 대한 시간제약 비용이고, v_j^{start} 와 v_j^{end} 는 노드 j 의 서비스 시작과 종료 시각이고, v_0^{start} 와 v_0^{end} 는 거점의 서비스 시작과 종료 시각이고, γ 는 시간제약 비용 가중치이다.

식 (12)를 이용하여 산출된 비용 중 가장 큰 비용을 갖는 노드를 할당받는 새로운 개미를 추가한다. 이후 미 할당 노드를 기존 개미에 할당할 수 있는 경우 식 (16)을 이용하여 비용을 산출한다.

$$MNCost_{i,j} = DC_{i,j} + WC_j + AC_{i,j} + TC_j + \tau_{i,j} \quad (16)$$

$MNCost_{i,j}$ 는 각 개미의 마지막 노드 i 에서 미할당된 노드 j 사이의 비용이다. $DC_{i,j}$ 는 거리 비용이며 식 (17)을 이용하고, WC_j 는 요구량 비용으로 식 (14)를 이용하고, $AC_{i,j}$ 는 각도 비용으로 식 (18)을 이용하고, TC_j 는 시간제약 비용으로 식 (15)를 이용하고, $\tau_{i,j}$ 는 노드 i 에서 노드 j 사이의 페로몬의 양이다.

$$DC_{i,j} = (1 - \frac{tr_{i,j}}{\sum_{m \in \text{unsigned node}} tr_{i,m}}) \cdot \alpha, (0 < \alpha < 1) \quad (17)$$

여기서 $DC_{i,j}$ 는 노드 i 에서 j 사이의 거리 비용이고, $tr_{i,j}$ 는 노드 i 에서 j 사이의 거리이고, m 는 미할당된 노드 집합의 일부이고, α 는 거리 비용 가중치이다.

$$AC_{i,j} = (1 - \frac{(180 - \|v_i^c - v_j^c\| - 180)}{360}) \cdot \delta, (0 < \delta < 1), (0 \leq (v_i^c, v_j^c) < 360) \quad (18)$$

$AC_{i,j}$ 는 노드 i 에서 j 사이의 각도 비용이고, a_i 와 a_j 는 노드 i 와 j 의 거점 기준 각도이고, δ 는 각도 비용 가중치이다.

식 (16)을 이용하여 산출된 비용을 내림차순으로 정렬하여 상위 50%만을 선택한다. 각 선택된 비용의 노드에 앞 노드의 출발 시각을 비교하여 가장 빠른 것을 선택하여 최종적으로 개미에 할당한다. 이러한 과정을 반복하여 해를 구축해 나간다.

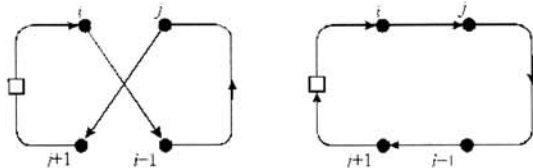
모든 노드가 할당되어진 경우 반복과정을 종료하며 이 때 구성되어진 개미의 수는 구축된 해의 차량수와 동일하게 된다.

4.5 경로 개선

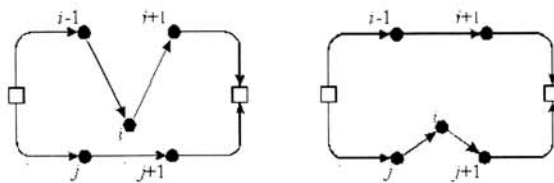
비용 함수가 반영된 최근집 이웃 휴리스틱을 이용하여 생성된 초기 해를 2-opt 알고리즘을 적용하여 해를 개선한다.

2-opt는 Russell이 제안한 방법으로 (그림 3)과 같이 구성된 경로에서 경로의 순서를 변경하여 해를 개선하는 방법이다[12, 15, 18]. 두 연결 ($i, i+1$)과 ($j, j+1$)를 (i, j)과 ($i+1, j+1$)로 교체하는 과정을 반복적으로 적용하여 해가 개선되지 않을 때까지 반복한다. 이 방법은 차량의 경로를 최적화하지만 차량수는 줄어들지 않는다.

다음으로 각 차량의 수를 줄이기 위한 Relocate를 적용한다. Relocate는 Prosser et al.이 제안한 방법으로 (그림 4)와 같이 Relocate는 단순히 하나의 경로($i-1, i, i+1$)에서 노드 (i)를 다른 경로($j, j+1$) 사이로 옮기는 과정을 반복적으로 적용하여 해가 개선되지 않을 때까지 반복한다[13, 15, 18]. 이 방법을 통해 차량의 수가 줄어들게 된다.



(그림 3) 2-opt 알고리즘



(그림 4) Relocate 알고리즘

4.6 페로몬 갱신 및 종료 조건 확인

매 반복 과정에서 두 번에 걸쳐 페로몬 흔적을 갱신한다 [9, 21].

첫 번째는 현재 개선된 해(Local Solution)를 식 (19)을 반영하여 페로몬 갱신을 한다.

$$\tau_{ij} \leftarrow ((1-\rho) \cdot \tau_{ij}^{old} + (\rho \cdot \Delta \tau_{ij}^{local}), (0 < \rho < 1) \quad (19)$$

두 번째는 현재 까지 구해진 가장 우수한 해(Global

Solution)를 식 (20)을 반영하여 페로몬 갱신 한다.

$$\tau_{ij} \leftarrow ((1-\rho) \cdot \tau_{ij}^{old} + (\rho \cdot \Delta \tau_{ij}^{global}), (0 < \rho < 1) \quad (20)$$

여기서 ρ 는 페로몬 증발률(Pheromone Evaporation)이다. 페로몬 증발률은 개미 군집 최적화 기법을 제어하기 위해 사용되는 매개변수(parameter)로써, 매 반복 때마다 개미가 이동한 경로에 축적되는 페로몬의 증가량과 기존 축적된 페로몬의 감소량을 결정하기 위해 사용된다.

알고리즘의 종료 조건은 초기화 과정에서 생성되는 비용 가중치 목록의 개수를 모두 소진 한 후 종료한다.

5. 실험 및 평가

실험은 Solomon[4]의 실험 데이터와 동일한 테스트 데이터 집합을 사용하였다.

실험 데이터는 R, C, RC 유형으로 구분되며 R은 노드의 분포도가 임의로 설정되었고 C는 노드의 분포도가 균집으로 설정되었고 RC는 R과 C를 혼합형으로 설정되어 있다. 또한 1, 2 유형으로 구분되며 1은 차량의 용적량이 작고 거점의 배송 시간이 짧으며 배송처의 시간 제약이 좁게 설정되어 있고, 2는 차량의 용적량이 크고 거점의 배송 시간이 길며 배송처의 시간 제약이 넓게 설정되어 있다. 모든 유형은 R1, C1, RC1, R2, C2, RC2로 크게 구분되며 총 56개의 데이터로 구성되고 6개의 유형별로 8~12개의 데이터를 가지며, 각 유형별 데이터 차이는 배송처의 시간 제약의 출현 빈도가 25%, 50%, 75%, 100%로 설정되거나 시간제약 길이가 모두 동일하게 설정 있다.

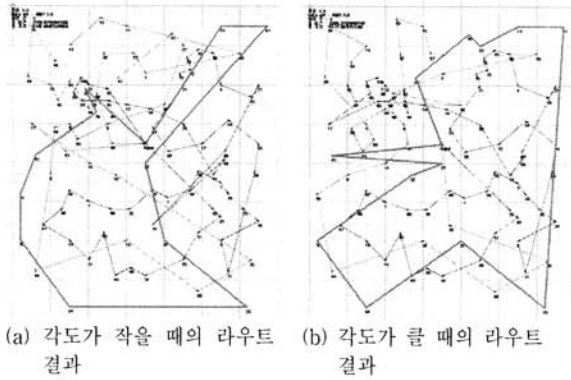
본 연구의 실험 환경은 Intel Pentium 4 2.80Ghz, MS Windows XP, RAM 1.75GB이며 C#으로 구현하였다.

<표 1>는 실험에서 사용될 페로몬 증발률(ρ)을 결정하기 위해 ρ 을 0.1부터 0.9까지 0.1씩 증가해 가며 실험하여 모든 실험데이터의 평균을 구하여 비교한 결과로 0.9의 값을 지정한 경우 가장 우수한 해를 도출하는 것을 알 수 있다.

(그림 5)는 유리한 각도 비용을 결정하기 위해 두 고객 노드의 각도의 차가 작거나 클수록 유리하도록 하여 페로몬 증발률 0.9로 지정하고 실험 데이터 R207로 비교한 그래프

<표 1> 페로몬 증발률(ρ) 변화량에 따른 결과 비교

증발률	차량수	총운행시간	총이동거리
0.1	8.339	4864.504	1327.234
0.2	8.339	4858.998	1326.315
0.3	8.304	4839.772	1323.452
0.4	8.375	4858.033	1329.698
0.5	8.339	4847.866	1319.046
0.6	8.339	4857.243	1329.566
0.7	8.321	4842.735	1319.053
0.8	8.375	4860.931	1331.485
0.9	8.268	4833.161	1315.480



(그림 5) 각도비용에 따른 라우팅 스케줄링 결과 그래프

이다. 대부분의 실험 데이터에서는 각도의 차이에 따른 결과가 동일하였으나, (그림 5)의 (a)는 사용된 차량수가 3대, 총 운행 시간은 2189.972, 총 이동 거리는 1182.894이었고, (b)는 사용된 차량수가 3대이며 총 운행 시간은 2204.375, 총 이동 거리는 1177.008이었다. 위 결과에서 볼 때 각도의 차이가 작을수록 비용을 높게 평가한 것이 보다 유리한 해를 도출할 수 있었다.

<표 5>은 이 논문에서 제시된 목적함수와 동일한 목적함수를 갖는 Solomon II[4]와 기회시간이 반영된 하이브리드 휴리스틱[19]와 제안된 휴리스틱에 대한 실험 비교 결과이다. 표시된 값은 각 유형별 모든 데이터 집합에 대한 평균 값이다. 실험의 결과는 고객 노드의 분포도가 거점을 중심으로 고르게 퍼져있으며 차량의 용적량이 크고 배송처의 시간 제약이 넓게 설정되어 있는 R2와 RC2형에서 다른 휴리스틱에 의해 구해진 결과보다 더 좋은 결과를 보여주고 있다.

각 데이터별 결과 분석을 위하여 식 (21)을 이용하여 거점의 배송 시간 길이 대비 모든 고객 노드의 평균 시간 제약 길이를 비율로 구하였다. 평균 시간 제약 비율이 클수록 경로 구성 시 탐색 가능한 공간이 상대적으로 증가하게 된다.

$$\text{평균시간제약비율} = \frac{\sum_{i=1}^n (v_i^{\text{end}} - v_i^{\text{start}})}{v_0^{\text{end}} - v_0^{\text{start}}} \quad (21)$$

<표 2>, <표 3>, <표 4>은 C, R, RC 유형별로 기회시간이 반영된 하이브리드 휴리스틱과 제안 휴리스틱에서 사용된 차량수를 비교한 결과이다.

<표 2>에서 볼 때 평균 시간 제약 비율이 다른 유형에 비해 큰 유형인 C103, C104, C203, C204에서 좋지 않은 결과를 보였는데 이는 C 유형의 데이터에서 고객 노드의 분포는 균집화를 이루는데 시간 제약 조건이 넓어져 상대적으로 다른 균집으로 고객을 서비스하게 될 수 있는 가능성이 높기 때문이다.

<표 3>에서 볼 때 평균 시간 제약 비율이 다른 유형에 비해 클수록 다양한 경로 탐색이 가능하여 차량수가 줄어드는 것을 볼 수 있다. 이는 앞서 C 유형과는 다르게 모든 고

<표 2> 시간 제약 비율에 따른 차량 수 결과 비교

유형	비율	기회 시간	제안 휴리스틱	유형	비율	기회 시간	제안 휴리스틱
C101	4.92	10	10	C201	4.72	3	3
C102	26.40	10	10	C202	27.70	3	4
C103	47.60	10	11	C203	50.60	4	4
C104	69.00	10	11	C204	73.50	4	4
C105	9.84	10	10	C205	9.44	3	3
C106	12.60	10	10	C206	14.40	3	3
C107	14.60	10	11	C207	18.10	3	3
C108	19.70	10	10	C208	18.90	3	3
C109	29.10	10	10				

<표 3> 시간 제약 비율에 따른 차량 수 결과 비교

유형	비율	기회 시간	제안 휴리스틱	유형	비율	기회 시간	제안 휴리스틱
R101	4.35	21	20	R201	11.60	4	4
R102	25.00	18	18	R202	32.90	4	4
R103	44.80	16	16	R203	54.20	3	3
R104	64.50	12	13	R204	75.10	3	3
R105	13.00	15	16	R205	24.00	3	3
R106	31.50	14	14	R206	42.20	3	3
R107	49.10	13	12	R207	60.30	3	3
R108	66.70	11	11	R208	78.30	3	3
R109	25.60	13	13	R209	35.00	3	3
R110	37.60	12	13	R210	38.30	3	3
R111	40.50	12	13	R211	47.20	3	3
R112	51.10	11	11				

<표 4> 시간 제약 비율에 따른 차량 수 결과 비교

유형	비율	기회 시간	제안 휴리스틱	유형	비율	기회 시간	제안 휴리스틱
RC101	12.50	17	16	RC201	12.50	4	4
RC102	29.8	15	15	RC202	33.20	4	4
RC103	46.90	13	13	RC203	53.90	3	4
RC104	64.40	12	11	RC204	74.70	3	3
RC105	22.60	16	17	RC205	23.20	4	5
RC106	25.00	13	13	RC206	25.00	4	3
RC107	36.80	13	12	RC207	36.40	4	3
RC108	46.80	12	12	RC208	49.20	3	3

객 노드가 거점을 중심으로 고르게 분포되어 있기 때문이다. 그리고 같은 차량수를 사용했지만 <표 5>에서와 같이 R2 유형의 경우 총 운행거리가 더욱 우수하게 나온 것을 확인할 수 있었다.

<표 4>에서 볼 때 R 유형과 유사하게 평균 시간 제약 비율이 다른 유형에 비해 클수록 차량수가 줄어드는 것을 볼 수 있었다. 이는 RC 유형이 C 유형과 같이 고객의 노드가 거점을 중심으로 균집을 이루면서 동시에 몇몇 노드가

<표 5> 다른 휴리스틱들과 제안 휴리스틱의 결과 비교

데이터 유형 (개수)	Solomon II[4]			기회시간이 반영된 하이브리드 휴리스틱[19]			제안 휴리스틱		
	차량수	총운행시간	총이동거리	차량수	총운행시간	총이동거리	차량수	총운행시간	총이동거리
C1(9)	10.0	10104.2	951.9	10.000	10116.282	1041.374	10.333	9967.898	962.686
C2(8)	3.1	9921.4	692.7	3.250	9944.041	706.408	3.375	9804.537	761.403
R1(12)	13.6	2695.5	1436.7	14.000	2779.972	1573.628	14.167	2645.087	1524.395
R2(11)	3.3	2578.1	1402.4	3.182	2503.477	1376.257	3.182	2429.753	1329.510
RC1(8)	13.5	2775.0	1596.5	13.875	2966.488	1819.588	13.625	2728.545	1646.625
RC2(8)	3.9	2955.4	1682.1	3.625	2872.654	1672.208	3.625	2776.616	1602.639

고르게 분포되어 있기 때문에 해 탐색 경로의 경우의 수가 증가했기 때문으로 보인다. 또한 RC1과 RC2 유형에서 기회 시간 반영 휴리스틱과 제안 휴리스틱 간에 사용된 차량수의 차이가 크게 나지 않는 것을 볼 수 있었으며, RC1 유형은 차량수와 총 운행거리가 우수하였고, RC2 유형은 차량수는 동일하고 총 운행거리가 우수하였음을 보였다.

6. 결론 및 향후연구

계산 량이 지수 적으로 증가하는 VRSPTW 문제를 해결 하기 위해, 멀티 비용 가중치가 적용된 최근접 이웃 휴리스 틱을 이용한 향상된 개미 군집 최적화를 제안하고 그 효율 성을 보였다. 멀티 비용 가중치를 통해 매 경로를 구성할 때 마다 해당 개미가 어떤 비용에 더 큰 비중을 두는지를 다양하게 하여 여러 초기 해를 얻을 수 있었다. 그리고 비 용 중 거점 기준 두 지점의 각도 차가 작을수록 유리하게한 각도 비용의 유용성을 확인 할 수 있었다. 또한 제안 휴리 스틱에서 고객 노드의 위치가 임의 분산 되어 있고, 차량의 용적량이 크고, 거점의 배송 시간이 길며, 시간 제약이 넓게 설정된 데이터 유형에 대해 페로몬 증발률을 0.9로 하였을 때 가장 유리하였음을 보였다. 하지만 모든 데이터 유형에 서 최적 값을 찾을 수 없었는데, 배송처의 시간 제약의 출 현 빈도가 낮아질 수록 그리고 군집 화된 문제의 경우 평균 시간제약 발생비율이 높을수록 해가 좋지 않았음을 확인할 수 있었다.

향후 연구 방향으로 이를 극복할 수 있도록 해를 향상 시 킬 수 있는 다른 경로 개선 알고리즘을 적용하거나 멀티 비 용 가중치가 적용된 최근접 이웃 휴리스틱이 보다 다양하고 우수한 초기 해를 가질 수 있도록 균형적인 비용 가중치 생 성 방법에 대한 연구가 필요하다. 또한 현실적인 차량 경로 문제의 해결을 위하여 이기종 차량을 고려한 문제, 다회전 차량 운행 문제, 다중 거점 문제, 차량 진입 제한 문제, 배달 과 수거가 혼합된 문제, 고객 노드간의 거리를 현실적인 도 로 상황을 고려하여 고속도로와 일반도로로 구분하여 실제 거리로 처리하는 문제 등 보다 현실 상황에서 발생될 수 있 는 제약을 추가하여 문제를 해결하는 해법에 대한 연구가 필요하다.

참 고 문 헌

[1] L. Bodin, B. Golden, A. Assad, and M. Ball, "Routing and Scheduling of Vehicles and Crews: The State of the Art," Computer and Operations Research, Vol.10, No.2, pp.63-211, 1983.

[2] J. K. Lenstra, and A. H. G. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," Networks, Vol.11, No.2, pp.221-227, 1981.

[3] M. W. P. Savelsbergh, "Vehicle routing and computer graphics," Centre for Mathematics and Computer Science, Amsterdam, Note OS-N8402, 1984.

[4] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," Operations Research, Vol.35, No.2, pp.254-265, 1987.

[5] B.-L. Garcia, J.-Y. Potvin, and J.-M. Rousseau, "A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints," Computers and Operations Research, Vol.21, No.9, pp.1025-1033, 1994.

[6] S. R. Thangiah, K. E. Nygard, and P. L. Juell, "GIDEON: A genetic algorithm system for vehicle routing with time windows," Artificial Intelligence Application, Proceedings., 7th IEEE Conference on., pp.322-328, 1991.

[7] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "A Tabu Search heuristic for the Vehicle Routing Problem with Soft Time Windows," Transportation Science, Vol.31, No.2, pp.170-186, 1997.

[8] B. Bullnheimer, R. F. Hartl, and C. Strauss, "Applying the Ant System to the Vehicle Routing Problem," MIC97, pp.1-12, 1997.

[9] L. M. Gambardella, E. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, New Ideas in Optimization," McGraw-Hill, London, UK, pp.63-76, 1999.

[10] M. Reimann, K. Doerner, and R. F. Hartl, "D-ants: Saving Based Ants Divide and Conquer the Vehicle Routing Problems with Time Window," Computer and Operations Research, Vol.31, No.4, pp.563-591, 2004.

[11] Q. Chengming, C. Shoumei and S. Yunchuan, "Using Ant

Colony System and Local Search Methods to Solve VRPTW," IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, pp.478-482, 2008.

[12] R. A. Russell, "An Effective heuristic for the M-tour traveling salesman problem with some side conditions," Operations Research, Vol.25, No.3, pp.517-524, 1977.

[13] P. Prosser, and P. Shaw, "Study of greedy search with multiple improvement heuristics for vehicle routing problems," Working paper, university of Strathclyde, Glasgow, Scotland, 1996.

[14] M. Zargayouna, F. Balbo, and G. Scemama, "A multi-agent approach for the dynamic VRPTW," In Proceedings of the International Workshop on Engineering Societies in the Agents World, Saint-Etienne, Springer Verlag, 2008.

[15] O. Braysy, and M. Gendreau, "Vehicle Routing Problem with Time Windows, Part I - Route Construction and Local Search Algorithms," Transportation Science, Vol.39, No.1, pp.104-118, 2005.

[16] O. Braysy, and M. Gendreau, "Vehicle Routing Problem with Time Windows, Part II - Metaheuristics," Transportation Science, Vol.39, No.1, pp.119-139, 2005.

[17] H. Nazif, and L. S. Lee, "Optimized Crossover Genetic Algorithm for Vehicle Routing Problem with Time Windows," American Journal of Applied Sciences, Vol.7, No.1, pp.95-101, 2010.

[18] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," Journal of King Saud University - Science, Vol.22, No.3, pp.123-131, 2010.

[19] 유영훈, 차상진, 조근식, "시간 제약을 가지는 차량 경로 스케줄링 문제 해결을 위한 기회시간 반영 하이브리드 휴리스틱", 지능정보연구, 제15권, 제3호, pp.129-142, 2009.

[20] 홍명덕, 유영훈, 조근식, "차량 경로 스케줄링 문제 해결을 위한 멀티 비용 함수를 갖는 개미 군집 최적화 기법 기반의 휴리스틱", 제 33회 한국정보처리학회 춘계학술발표대회 논문집, 제 17권, 제1호, pp.314-317, 2010.

[21] 이승관, 정태충, "Ant Colony System에서 효율적 경로 탐색을 위한 지역개신과 전역개신에서의 추가 강화에 관한 연구", 한국정보처리학회 논문지, 제10-B권, 제3호, pp.237-242, 2003.



홍명덕

2006년 인하공업전문대학 컴퓨터정보과(전문학사)
 2008년 서울디지털대학교 컴퓨터공학부(학사)
 2005년~2008년 구우정보기술(주) 연구원
 2008년~현 재 인하대학교 정보공학과 석사과정

관심분야: AI, Semantic Web, Ontology, Ant Colony Optimization, Vehicle Routing Problem



유영훈

2000년 인하대학교 전자계산공학(석사)
 2001년 (주)소프트웨어벨리 대표
 2010년 인하대학교 정보공학과(박사)
 2008년~현 재 인하대학교 컴퓨터정보공학과 전임강사

관심분야: AI, Constraint Satisfaction Problems, Resource Optimization, Meta-Heuristic



조근식

1982년 인하대학교 전자계산학 학사
 1985년 Queens College, CUNY M.A.
 1991년 City University of New York Ph.D.
 1991년~현 재 인하대학교 컴퓨터정보공학과 교수
 2006년~현 재 BK21 지능형 유비쿼터스 물류 기술 연구 사업단장

관심분야: AI, Semantic Web, Intelligent Agent System