

자기조직화 신경망을 이용한 복수차량의 실시간 경로계획

이종태* · 장재진**

Realtime Multiple Vehicle Routing Problem using Self-Organization Map

Jongtae Rhee* · Jaejin Jang**

■ Abstract ■

This work proposes a neural network approach to solve vehicle routing problems which have diverse application areas such as vehicle routing and robot programming. In solving these problems, classical mathematical approaches have many difficulties. In particular, it is almost impossible to implement a real-time vehicle routing with multiple vehicles. Recently, many researchers proposed methods to overcome the limitation by adopting heuristic algorithms, genetic algorithms, neural network techniques and others.

The most basic model for path planning is the Travelling Salesman Problem(TSP) for a minimum distance path. We extend this for a problem with dynamic upcoming of new positions with multiple vehicles.

In this paper, we propose an algorithm based on SOM(Self-Organization Map) to obtain a sub-optimal solution for a real-time vehicle routing problem. We develop a model of a generalized multiple TSP and suggest an efficient solving procedure.

1. 서 론

기업의 물류 활동에서 차량 수배송 업무는 과학적이고 합리적인 방법을 통해 물류 비용의 절감과 함께 고객의 다양한 요구조건을 충족시킴으로써

양질의 서비스를 제공한다는 측면에서 기업 경영 상 매우 중요한 역할을 하고 있다. 이러한 차량 수배송 업무의 대표적인 문제가 차량경로계획문제(Vehicle Routing Problem : VRP)이다. 차량경로계획은 최소의 비용으로 주어진 차량에 작업경로

* 동국대학교 산업공학과

** 한국외국어대학교 산업공학과

를 할당한다[6, 7, 13]. Danzig와 Ramser[12]에 의하면 차량경로계획은 일반적으로 NP-hard 문제라고 알려져 있다. 만일 c 개의 수요지점이 있는 경우에 가능한 차량의 경로의 수는 $\frac{1}{2} c!$ 이다. 따라서 차량경로계획의 최적 해법을 응용 분야에 적용하는데에는 많은 시간적 제약이 있고 이런 시간적 한계는 실시간적 차량경로계획을 어렵게 만든다.

차량경로계획이 계산상 복잡한 문제이지만 응용 분야는 작업 순서 결정, 산업용 로봇 프로그래밍, 차량의 노선 결정, 근거리 통신망의 구성 등에 이르기까지 매우 다양하다[1, 7, 8, 13]. 그러므로 차량경로계획 문제에서 짧은 시간 내에서 좋은 해를 찾아내는 것은 매우 중요하다. 이러한 이유로 오래전부터 시간적인 제약을 해결하기 위해 휴리스틱 방법, 시뮬레이티드 어닐링, 유전자 알고리즘, 신경망을 이용한 방법 등이 논의되었다. 그러나 기존 연구에서는 차량의 대수가 한 대인 경우를 주된 대상으로 하고 있기 때문에 현실 적용상 실용적이지 못한 면이 있다.

본 연구에서는 신경회로망(neural network)의 일종인 자기조직화 신경망(Self-Organization Map : SOM)을 응용하여 복수대의 차량의 일반화된 차량경로계획을 실시간으로 세우는 알고리즘을 개발한다. 자기조직화 신경망은 해가 갖는 특징을 위상학적으로 표현하고, 그러한 특징을 최대한 만족시킬 수 있는 해를 적용적으로 구해나가는 방법이다. 이 방법은 최적화된 해를 제시하지는 않으나, 그 적용 범위가 넓고 일반적으로 우수한 해를 단시간에 제공한다는 특징이 있다.

2. 차량경로계획에 대한 고찰

2.1 순회 판매원 문제

순회 판매원 문제는 방문해야 할 도시들과 이들 사이의 거리가 주어졌을 경우, 순회 판매원이 여러 개의 도시를 방문하기 위해 어떤 특정한 도시를 출

발하여, 어떠한 도시도 두 번 방문함이 없이 모든 도시들을 거쳐 처음 출발한 도시로 되돌아오려고 할 경우에 총 여행 거리, 또는 비용이 최소가 되는 경로의 순서를 정하는 문제인데, 순회 판매원 문제를 확장한 한 형태가 차량경로계획이다. 기본 차량경로계획을 수식으로 표현하면 다음과 같다[2].

$$\min Z = \sum_{i=1}^c \sum_{j=1}^c h_{ij} x_{ij} \quad (2.1)$$

$$\text{s.t } x_{ij} = 0, 1 \quad \forall i, j \quad (2.2)$$

$$\sum_{i=1}^c x_{ij} = 1 \quad \forall j \quad (2.3)$$

$$\sum_{j=1}^c x_{ij} = 1 \quad \forall i \quad (2.4)$$

$$u_i - u_j + cx_{ij} \leq c - 1, \quad (2.5)$$

$$i = 2, \dots, c$$

$$j = 2, \dots, c, i \neq j$$

c : 도시의 개수

i, j : 방문할 도시 ($1, \dots, c$)

h_{ij} : 도시 i 와 j 사이의 거리

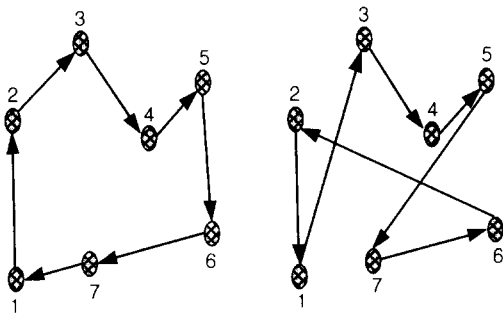
x_{ij} : 도시 i 에서 j 로의 방문여부 (1 : i 에서 j 로 방문함, 0 : i 에서 j 로 방문하지 않음)

u_i, u_j : 임의의 실수

여기서 식 (2.1)은 목적 함수식이며 운행 거리를 최소화하는 식이다. 식 (2.3)은 각 도시에서 도시 j 로 들어오는 것이 하나이어야 함을 나타내는 식이고, 식 (2.4)는 도시 i 로부터 다른 도시로 나가는 것이 하나임을 나타내는 식이다. 식(2.5)는 부분 경로가 생기는 것을 방지하는 제약식이며 출발 도시는 1번 도시라고 가정하였다. 본 식에서 두 도시간의 거리, h , 는 문제의 특성에 따라서 직각 거리나 유클리디안 거리가 될 수 있다.

순회 판매원 문제에서는 최소 거리의 경로가 최적의 경로라고 할 수 있는데, 좋은 해답이란 최적의 해답에 근사한 값을 말한다. 본 연구에서는 일반적인 경험에서 좋은 해의 대안으로 얻어진 원모

양의 단일 폐곡선 형태의 경로를 근사하여 최적 경로를 선정한다. 즉 [그림 1]의 가능해 중에서 (a)가 (b)에 비하여 일반적으로 바람직한 해를 나타내는 것으로 생각하고 모델을 설정한다. 또한 본 논문에서는 단순한 순회 판매원 문제를 일반화하여 복수의 차량의 수와 같은 갯수의 차량의 닫힌 경로를 구한다.



(a) 좋은 해 (b) 나쁜 해
[그림 1] 순회 판매원 문제 해의 예

2.2 차량경로계획에 대한 선행연구

최근에는 물류 부분의 중요성의 증대와 더불어 차량경로계획에 대한 연구가 활발히 진행중인데 문제 해결을 위한 접근 방법은 크게 최적해법, 계산 시간을 줄일 수 있는 발견적해법, 그리고 신경망을 이용한 지능적 접근법으로 구분될 수 있다.

최적해법(Optimal Algorithm)을 이용한 방법으로, Christofides, Mingozzi와 Toth[10]는 Dynamic Programming을 통해 최적해의 하한을 얻어내고 서비스 시간대가 거의 겹치지 않는 50개 노드 문제를 풀었다. 또한 이와 비슷한 방법으로 Baker[5]는 서비스시간대가 중복되는 50개의 노드를 갖는 문제를 효과적으로 풀었다. Desrosiers et. al[14]은 Ford Bellman-Moore의 동적계획법 알고리즘을 전형적인 최단거리문제로 일반화시킨 모형으로 구축하였고, 2,500개의 노드와 250,000개의 호를 가진 문제의 해를 CYBER 173 컴퓨터를 사용하여 1분만에 산출하였다. 이와 같은 최단거리문제에 대한 연구

는 이후 다중자원제약(multiple resource constraints)이나 서비스시간대 제약을 추가하여 VRPTW(Vehicle Routing Problem with Time Window)로 확장되었다. Dumas et al.[16]는 총이동거리(total travel distance)를 최소화하는 동적계획법 모형을 제안하였으며, Desrochers et al.[15]은 VRPTW를 set partitioning problem으로 모형을 수립한 후 LP(선형계획법)으로 이완하여 열생성기법(column generation method)으로 푸는 최적 알고리즘을 제시하였다. 한편 Pesant et al.[24]는 constraints logic programming 기법을 이용하여 서비스시간대에 대한 제약이 부과되지 않도록 구성한 다음 분지한계법을 적용하는 방법을 제시하였다. 한편, 최근 컴퓨터 관련 하드웨어와 소프트웨어의 발전에 힘입어 최적해법에 대한 연구가 새로운 전환기를 맞이함에 따라, 효과적으로 계산량을 줄일 수 있는 최적기법의 개발과 적용기술에 대한 연구가 비중있게 이루어지고 있다.

한편 차량경로계획은 NP-hard에 속하는 문제이기 때문에 최적화 해법을 적용하기에는 어려움이 많아 발견적 해법이 상대적으로 활발히 연구되고 있다. 근사해법 혹은 발견적해법으로는 기존 TSP 문제에 적용되던 경로개선절차(tour improvement procedure)를 위시한 Tabu search 방법[28], 유전 알고리즘 등과 같은 다양한 방법들이 적용되고 있다. Savelsberg가 제안했던 발견적해법[25]은 총거리비용을 최소화하는 호 교환(arc interchange) 방법이었으나 후에 시간개념을 포함한 연구[26]로 확장되었다. Gendreau et al.[19]는 일반적인 TSP를 위해 개발되었던 삽입방식(generalized insertion)의 발견적해법 GENIUS[17]를 TSPTW에 적용하였는데, 기존 연구에서 사용했던 문제들에 대해 근사해 혹은 최적해를 얻을 수 있음을 보였다.

Clarke와 Wright[11]는 두 차량이 서로 다른 두 지점을 방문하고 돌아오는 것보다 한 대의 차량이 두 지점을 방문하고 돌아오는 경우 발생하는 비용의 절약을 이용하는 절약기법(saving heuristic)을 제안하였다. Holmes와 Parker[21]는 절약기법을 개

량한 발견적 해법을 개발하였다. 이들은 절약기법으로 구한 경로내의 호(arc)중 해의 질을 떨어뜨리는 호를 가려내 제거하여 새로운 경로를 구성함으로써 개선된 해를 구한다. Altinkemer와 Gavish[3]는 수요지점간의 절약기법을 그대로 이용하면서 수요지점간의 결합에 matching 모형을 적용하여 해를 구하는 발견적 해법을 개발하였다. Christofides와 Eilon[9]은 Lagrangian 완화 기법에 의한 분지한계법을 개발하여 기본적인 차량경로계획에 차량의 최대 운행 거리 제한에 대한 제약 조건을 추가시킨 문제를 다루었다. 분지한계법은 최적해를 구할 수 있는 장점은 있으나, 수요지점이 늘어날 경우 계산 시간이 많이 걸리거나 불가능하다는 단점을 가지고 있다. Osman[23]은 시뮬레이티드 어닐링 개념을 도입하여 발견적 해법을 개발하였다. Gendreau et. al[18]은 Tabu search 개념을 차량경로계획에 도입하여 발견적 해법을 개발하였다. 이렇게 시뮬레이티드 어닐링과 Tabu search의 개념을 도입한 발견적 해법들은 최근에 보다 활발히 연구되고 있으며, 우수한 해를 낼 수 있으나 해법의 수행시간이 매우 오래 걸린다는 점이 단점으로 지적되고 있다.

이 외에 신경회로망을 이용한 기법들도 활발히 연구되고있다. Hopfield와 Tank[22]는 에너지 함수를 이용하여 에너지 함수를 줄여나가면서 최적해를 빠른 시간내에 구해내었다. 최근에는 차량경로계획과 같은 최적화 문제를 해결하기 위해서 적용적 신경회로망을 많이 사용한다. Angeniol et. al[4]은 추가-삭제기법(creation-deletion mechanism)에 의해서 근사 최적해에 도달하는 방법을 제안하였다. 이것은 한 노드가 두 개의 도시에 대해 승자가 될 때 노드를 추가하고, 3번의 순환에서 어떤 도시에 의해서도 승자 노드로서 선택되어지지 않으면 노드를 제거하는 방법이다. Somhom et. al[27]은 더 빠르고 좋은 학습을 위해 경쟁층의 노드를 도시의 수에 두 배로 구성하였다. 그리고 방지색인(inhibition index)을 주어 같은 노드가 계속 승자가 되는 것을 금하고 다른 노드에게 승자가 될 기회를

주어 효율적으로 학습하는 알고리즘을 개발하였다.

차량경로계획은 NP-hard에 속하는 문제이기 때문에 수리적인 방법으로 해결할 경우 시간적인 한계를 가지고 있다. 이를 극복하기 위해 여러 탐색적인 방법, 유전자 알고리즘, 신경망을 통한 알고리즘들이 개발되었다. 이 중에서 적응적 신경망에 관련된 여러 연구들은 한 대의 차량의 경로 결정에 대해 좋은 학습결과를 보여주고 있다. 기존의 대부분의 논문들이 차량이 한 대인 경우를 주된 대상으로 하고 있는데 비하여 차량이 두 대 이상의 일반화된 문제를 SOM으로 해결하기 위해서는 경쟁층의 구조를 차량수에 맞는 일반화된 모양으로 구성해야 하는 데, 이 경쟁층의 구조를 결정하기가 어렵다. 본 연구에서는 두 대 이상의 차량경로계획에 대해 차량수에 맞는 경쟁층의 구조를 구성하고 차량경로계획을 실시간으로 세우는 알고리즘을 개발한다.

3. SOM을 이용한 복수차량의 차량 경로계획

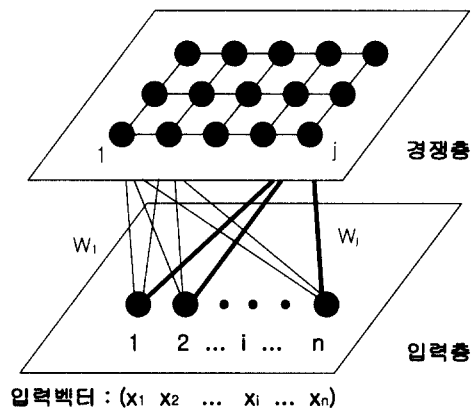
3.1 SOM 고찰

- t : 시간 (반복회수)
- c : 도시수
- P_i : i 번째 도시의 좌표값 (P_{1i}, P_{2i})
- m : 차량수
- j^* : 전체 승차 노드
- j_k^* : k 번째 원에서 승차 노드 ($k = 1, 2, \dots, m$)
- $w_j(t)$: 학습시간 t 에서의 j 노드의 가중치 백터 (w_{1j}, w_{2j})
- $\eta(t)$: 학습시간 t 에서의 학습률
- η_0 : 학습모수
- τ : 시간상수(time constant)
- n_j : 경쟁층 노드 j 의 이웃 함수 값
- b_k : k 번째 원의 이웃 폭의 범위

인공 신경망은 생물학적 처리 과정을 인공적으

로 구현한 것으로 응용 분야에 따라 다양한 형태의 모델들이 제시되었다. 본 연구에서 응용하려는 SOM은 Kohonen[20]에 의해 제시되었다. 이것은 신생아들이 눈의 초점을 맞추기를 배우는 과정에서 눈에 보이는 자극과 물체 및 모양들을 서로 연관시키는 자기조직화를 모델링 한 것이다. 여기서 자기 조직화란 입력패턴에 스스로 적응하는 것을 말한다. 일반적인 SOM의 구조는 [그림 2]와 같은 두 개의 층으로 이루어져 있다. 첫 번째 층은 입력층(input layer)이고, 두 번째 층은 경쟁층(competitive layer)이다. 방문 해야 할 도시를 상징적으로 나타내는 두 번째 층의 각 node(node j)에는 입력층의 노드들로부터 ($w_{j1}, w_{j2}, \dots, w_{jn}$)의 가중치 벡타가 연결된다. 여기에서 n 은 입력층의 node 개수이다. 또한 두번째 층의 노드들은 문제에 따라 요구되는 위상 (topology)을 반영할 수 있도록 배열된다. 본 연구에서 대상으로 하는 차량경로문제의 경우 입력노드는 2개 ($n = 2$)이며 이는 각 도시의 2차원 좌표를 나타낸다. 또한 이 경우 경쟁층의 노드 j 로 연결되는 가중치벡타(w_{j1}, w_{j2})도 2차원상의 좌표를 의미하게 된다. 이 가중치 벡타는 입력 벡터(x_1, x_2)가 입력층에 주어지면 경쟁학습에 의해서 수정된다. 경쟁학습에서는 노드들의 가중치벡타 중에서 입력벡터와 가장 거리가 가까운 승자 노드(winner node)와 승자노드의 이웃 노드(neighborhood node)들이 학습에 참여한다. 즉 학습 기간 중에 승자 노드의 가중치는 입력벡터를 향해서 현재의 값이 많이 변경되고 다른 노드의 가중치들은 경쟁층상에서 승자 노드와의 이웃한 정도에 따라 그 변화 정도를 결정하여 수정된다. 이러한 경쟁학습을 반복하고 학습이 어느 정도 진행이 된 다음에는 학습이 일어나는 이웃 노드의 범위를 점진적으로 줄여, 결국에는 승자 노드에 대해서만 학습을 하게하여 최종 해를 도출한다. 여기서 초기에 주어진 경쟁층의 위상적 구조는 신경망의 학습과정의 안내역할을 한다.

본 연구에서는 이러한 SOM의 구조를 순회 판매원 문제에 응용한다. SOM을 순회 판매원 문제에

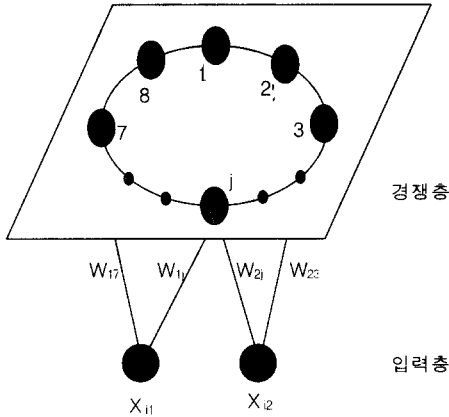


[그림 2] 일반적인 SOM구조

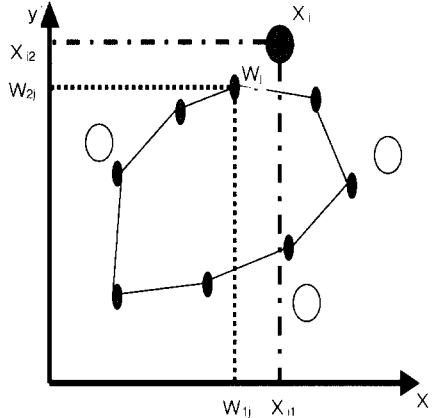
적용하기 위해 먼저 경쟁층의 노드 배열을 원모양으로 구성한다. 이것은 원 모양이 최소경로를 선택해야 하는 순회 판매원 문제의 경로를 가장 잘 표현하기 때문이다. 학습이 진행됨에 따라 입력층과 경쟁층 노드사이의 가중치들은 주어진 입력 도시의 좌표값과 유사한 값으로 변동된다. 결국 순회 판매원 문제를 SOM으로 해결하면 각각의 가중치 벡터들에 의해 주어지는 좌표들은 각 방문 목적도시들에 할당되게 되며 동시에 실제 도시의 지리적 배열 순서도 경쟁층에서 주어진 원 모양의 구조를 가능한 한 유지하게 되는데, 그 결과 전체 이동거리가 짧은 경로를 생성하게 된다. SOM에 의한 이 과정은 빠른 시간에 비교적 좋은 해를 도출한다.

3.2 차량이 한 대일 경우 SOM을 이용한 경로 계획

SOM은 경쟁학습을 통해 순회 판매원 문제의 기하학적인 해를 도출한다. 매 반복학습과정마다 방문해야 하는 도시들 중 하나가 무작위로 선택되며 그 도시의 좌표값(P_j)이 입력층에 입력된다. 입력층과 경쟁층의 도시, j , 사이의 가중치 벡터(w_j)는 2차원상의 실제 좌표값이다. w_{j1} 는 입력층의 첫 번째 노드와 경쟁층 j 사이의 가중치이며 w_{j2} 는 입력층의 두 번째 노드와 경쟁층 j 사이의 가중치이다. 이를 기하학적으로 표현하면 [그림 3-a]와 같이



(a) 경쟁학습



(b) 기하학적인 표현

[그림 3] 순회 판매원 문제의 경쟁학습과 기하학적인 표현

w_{1j} 는 2차원상의 x 좌표값이 되고, w_{2j} 는 y 좌표값이 된다. 이 때, 각 노드들의 가중치 벡터들은 처음에는 임의의 값으로 주어졌다가 학습이 진행됨에 따라 방문해야 하는 도시의 좌표값을 가지게 된다. 방문해야 하는 도시의 방문 순서는 경쟁층에서 원 모양으로 배열된 순서를 따른다.

방문해야 하는 목적지가 c 개인 순회 판매원 문제를 SOM으로 해결하기 위해서는 먼저 경쟁층에 $2c$ (도시의 2배)개의 node가 있는 원형구조를 정의한다. 노드를 $2c$ 개로 한 것은 학습의 효율을 향상하기 위해서이다[27]. 이 경우 학습이 완료된 이후 이 노드들 중 c 개는 각각 하나의 도시를 대표하게 되며 나머지 c 개는 무시된다. 처음의 가중치 벡터는 앞에서 말한 바와 같이 임의의 값으로 주어진다. 이 값들은 [그림 3-b]와 같이 2차원 평면상에 점으로 나타낼 수 있다. 각 학습단계에서는 방문해야 하는 도시들 중 한 도시를 무작위로 선택하여 선택된 도시와 현재의 신경망에 주어진 모든 node의 가중치들을 비교해서 식 (3.1)의 유클리드 거리 (D_j), 또는 다른 주어진 기준에 의한 거리가 가장 가까운 노드를 승자 노드 j^* 로 선택한다. 즉

$$D_j = | P_i - w_j | = \sqrt{(P_{1i} - w_{1j})^2 + (P_{2i} - w_{2j})^2} \quad (3.1)$$

$$D_{j^*} = \min_j (D_j) \quad (3.2)$$

각 반복학습과정에서 승자 노드와 주변의 이웃 노드의 가중치는 다음의 학습함수에 따라 수정된다.

$$w_j(t) = w_j(t-1) + \eta(t)n_j \{ P_i - w_j(t-1) \} \quad (3.3)$$

여기서 학습율 $\eta(t)$ 는 식(3.4)와 같이 시간에 따라 지수적으로 감소하는 함수이며 학습이 진행됨에 따라 가중치들의 변화량의 크기를 줄인다.

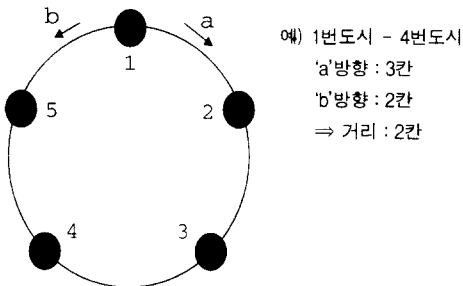
$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau}\right) \quad (3.4)$$

n_j 는 다음과 같은 가우스 이웃 함수(gaussian neighborhood function)이다.

$$n_j = \exp\left[-\frac{d(r_j, r_j)}{b}\right] \quad (3.5)$$

여기서 r_j 는 경쟁층 노드의 노드번호이고, r_{j^*} 는 승자 노드번호이다. 또한 $d(r_j, r_j)$ 는 node r_j

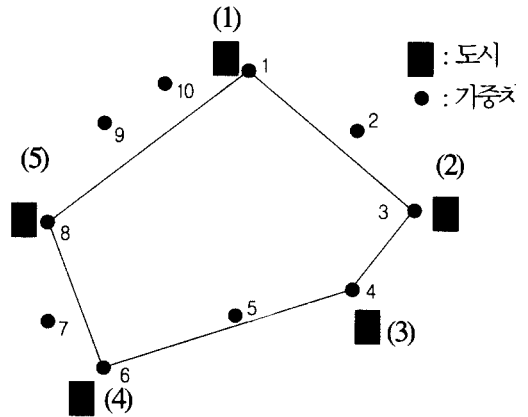
와 r_j 의 경쟁층에서의 위상학적 거리(여기서는 이웃 단계수)이다. [그림 4]와 같이 5개의 노드가 있을 때 1번 노드와 2번 노드간의 거리는 1이 된다. 그런데 만약 1번과 4번 노드간의 거리를 계산할 경우 'a' 방향으로 돌아가게 되면 거리가 3이 되지만 'b' 방향으로 가게 되면 거리가 2가 된다. 이런 경우에는 두 방향 중 작은 값인 2를 노드간의 거리로 본다. 식 (3.5)는 이웃 정도에 따라 학습 정도를 결정하는 식이다. 즉 경쟁층에서 승자 노드와 거리가 가까운 노드의 가중치에 대해서는 학습 정도를 크게 하고 승자 노드와 거리가 먼 노드의 가중치에 대해서는 학습 정도를 작게 한다. 모수 b 는 이웃 폭의 범위를 나타내는 변수이다. b 값이 크면 학습에 참여시키는 노드의 영역을 넓게 하는 것이고, b 값이 작으면 학습영역을 작게 하는 것이다. b 값은 시간이 지남에 따라 지속적으로 감소되는데, 이 값에 의하여 학습이 어느 정도 진행되었을 때는 승자 노드에 대해서만 학습하게 한다. 이러한 학습과정은 안정상태에 도달할 때까지 계속 반복된다. 안정상태라 함은 경쟁층의 각 노드의 가중치가 해당 도시의 좌표값에 근접한 상태이다.



[그림 4] 원의 구조에서 거리 구하는 예

마지막으로, 학습이 다 끝나면 각 도시들에 가장 가까운 가중치들을 할당하여 방문 경로를 정한다. 예를 들어 [그림 5]가 목적지가 5개인 순회 판매원 문제에 SOM을 이용한 학습결과라면 각 도시 근처에 있는 가중치들이 도시를 대표하게 된다. 즉 1번 가중치는 (1)번 도시를, 3번 가중치는 (2)번 도시

를, 4번 가중치는 (3)번 도시를 6번 가중치는 (4)번 도시를 8번 가중치는 (5)번 도시를 대표한다. 그러므로 방문 경로는 <1→2→3→4→5>가 된다.



[그림 5] 순회 판매원 문제의 SOM적용 예

3.3 SOM을 이용한 복수 차량의 일반화된 순회 판매원 문제

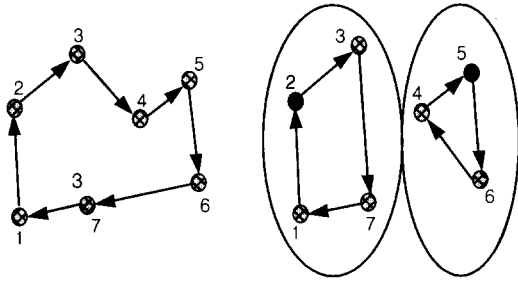
본 절에서 제안된 방법은 두 대 이상의 일반화된 차량 댓수에 대한 경로를 계획한다. 즉, 정해진 목적지에 대한 다중 순회 판매원 문제(Multiple Travelling Salesman Problem : MTSP)의 해를 도출한다. 이 경우는 차량이 한 대일 때와는 다르게 일반적인 순회 판매원 문제에 맞는 SOM의 경쟁층 구조를 구성해야 한다.

3.3.1 문제 해결 방안

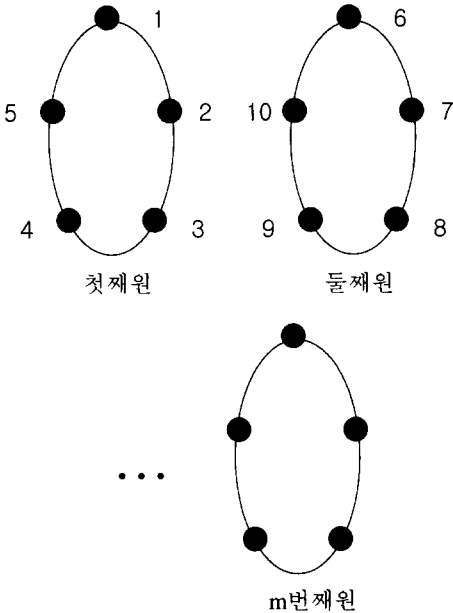
다수의 차량에 방문순서를 배정하는 방법은 [그림 6]과 같다. [그림 6]은 7개의 목적지를 갖는 문제인데, (a)는 차량이 한 대인 경우이고 (b)는 차량이 두 대인 경우이다.

(a)와 같이 차량이 한 대인 경우에는 경쟁층이 한 개의 원형 구조를 가지고, (b)와 같이 차량이 두 대인 경우에는 두 개의 원형 구조를 가진다. 차량이 m 대로 확장되면 m 개의 원형 구조를 가진다. 본 연구에서는 각 원의 노드의 수는 가급적 같은 수를

갖도록 한다. [그림 7] 즉, 도시의 수가 c 개 일 경우 각 원의 노드의 갯수는 $2 * c / m$, 또는 이 값에 가까운 값이 되도록 한다.



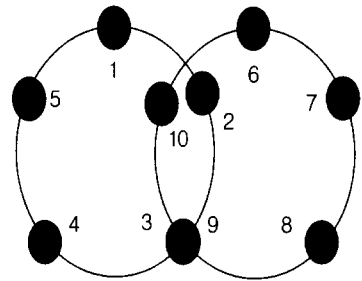
(a) 차량이 한 대 (b) 차량이 두 대
[그림 6] 순회 판매원 문제의 확장



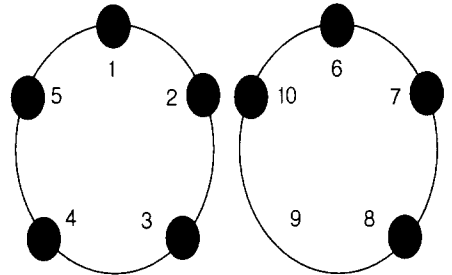
[그림 7] 차량이 m 대인 경우 경쟁층의 구조
예 : m 개의 원

그런데 일반적으로 차량이 m 대로 확장된 경우 route의 구조를 구성하는 것이 어렵다. 즉 영역이 애매한 목적지에 대해서는 어떤 차량에 할당하는 것이 좋은지 결정하기가 어렵다. [그림 8]과 같이 차량의 경로가 겹치는 것이 좋은 지, 적당한 거리

에 원을 배치시키는 것이 좋은 지 아니면 멀리 원을 구성하는 것이 좋은 지 쉽게 알 수가 없는 것이다. 본 연구에서는 승차노드가 있는 원의 노드들은 학습을 경험하지만 다른 원의 노드들은 학습을 하지 않도록 하여 이러한 문제를 해결하였다. 즉, 원들 간에 “Winner takes it all” 개념의 경쟁학습(Competitive learning)을 적용함으로써 노드들을 원들 간에 자동적으로 분배한다. 이 방법은 본 절차의 실제 응용을 용이하게 한다.



(a) 겹쳐진 원 구조



(b) 떨어진 원 구조

[그림 8] 차량이 두 대인 경우 원간의 거리

3.3.2 제안된 신경망의 학습단계

[단계 1] 초기화

초기화 단계에서는 반복회수, 도시수, 차량수 등의 모수와 가중치를 초기화하는 것이다. 경쟁층과 입력층간의 가중치는 임의의 값으로 초기화한다.

[단계 2] 네트워크의 초기화

경쟁층에 차량수만큼의 원형 모양의 구조를 구성한다.

[단계 3] 새로운 입력벡터 제시

방문해야 하는 목적지에 해당하는 도시 좌표들 중에서 무작위로 한 도시를 선택하고, 선택된 도시의 좌표값을 신경망에 제시한다.

[단계 4] 입력패턴과 가중치간 거리계산

제시된 입력패턴과 가중치의 거리를 식 (3.6)과 같이 유클리드 거리(euclidean distance), 또는 기타 미리 정해진 기준에 의한 거리를 이용하여 계산한다.

$$D_j = | P_i - w_j | \quad (3.6)$$

$$= \sqrt{(P_{1i} - w_{1j})^2 + (P_{2i} - w_{2j})^2}$$

[단계 5] 각 원의 승자 노드 선택

입력패턴과 가중치간의 거리가 계산된 후 각 원에서 D_j 가 가장 작은 승자 노드 $j^*_1, j^*_2, \dots, j^*_m$ 을 구한다. 이 때 각 원의 승자 노드를 선택할 때 방지색인을 주어 한 노드가 계속 승자가 되는 것을 방지한다. (해당 노드에 대해서는 계산을 안 함.) 즉 특정한 노드 이외에 다른 노드들에 대해서도 학습기회를 주어 학습효율을 높인다.

[단계 6] 전체 승자 노드 선택

$j^*_1, j^*_2, \dots, j^*_m$ 중에서 거리 D_j 가 가장 작은 j^* 를 전체 승자 노드로 선택한다.

[단계 7] 전체 승자노드가 있는 원의 가중치 조정
전체 승자노드가 포함된 원의 노드 j 에 대해 $w_j(t)$ 를, 시간에 따른 학습을 $\eta(t)$ 와 이웃 함수 n_j 를 학습식에 부과하여 조정한다. 학습식은 다음과 같다.

$$w_j(t) = w_j(t-1) + \eta(t)n_j(P_i - w_j(t-1)) \quad (3.7)$$

여기서 학습을 $\eta(t)$ 는 다음 식과 같이 시간에 따라 지수적으로 감소하는 함수이다.

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau}\right) \quad (3.8)$$

n_j 는 다음과 같은 가우스 이웃 함수이다.

$$n_j = \exp[-d(r_j, r_j)] \quad (3.9)$$

[단계 8] 만약 입력층의 모든 도시에 대해 경쟁층의 각각의 대응 노드를 선정할 수 있고 현 시점에서 각 도시의 위치와 이에 해당하는 노드의 가중치 벡타와의 거리가 오차범위 이내에 들어오면 학습을 종료한다. 그렇지 않으면 [단계 3]으로 가서 반복한다. 여기서 만약 모든 입력층의 도시에 대해 1:1로 대응하는 경쟁층의 노드를 선택할 수 없는 상태로 학습 과정이 수렴이 된다면 이는 학습의 실패를 뜻한다. 이때는 초기 가중치의 값을 변경시켜 학습을 다시 시작한다. 그러나 이러한 실패의 경우는 거의 발생하지 않는 것으로 알려져 있다.

[단계 9] 최종해 도출

학습이 끝나면 마지막으로 각 도시의 좌표와 가장 가까운 가중치에 이 도시의 정확한 좌표값을 할당하여 정확한 도시의 값을 해로 얻을 수 있도록 한다.

4. 모의실험과 결과분석

본 연구에서는 두 대 이상의 복수대에 대한 차량경로계획을 세우는 방법을 제시하였다. 본 절의 모의 실험에서는 차량이 두 대인 경우에 대해 실험하고, 도시수를 변화시키며 제안된 방법을 최적해와 비교하여 수행도를 보이고 이를 분석한다.

4.1 SOM의 최적해와의 근접도 분석을 위한

모의실험

실험은 도시수를 5개, 10개, 15개의 3가지로 하였고, 각 모형에서 차량의 출발점은 특정 도시에 있다고 가정하였다. 모든 차량이 같은 곳에서 출발을 하여야 하는 경우에도 이 개념을 약간의 수정을 통하여 적용 할 수 있다. <표 1>은 미국 LINGO System 사의 LINGO V. 6.0에 의해서 찾아낸 최적

헤이고, <표 2>는 도시수에 따른 SOM의 수행결과이다. SOM에 의한 실험결과를 얻는 데에는 매우 짧은 시간만이 소요되어 제안된 방법이 실시간의 차량경로선택에 사용될 수 있음을 알 수 있다.

<표 1> 도시수에 따른 최적해

		모형 A	모형 B	모형 C
도시수		5	10	15
경로	차량1	2-3-4	2-3	2-14
	차량2	1-5	1-9-8-7-5-4-6-10	1-15-4-3-13-12-5-7-6-8-9-11-10
거리	차량1	11903	8858	3802
	차량2	6454	19925	38616
총 거리		18357	28783	42418

<표 2> 최적해와 비교한 SOM의 수행결과

		모형 A	모형 B	모형 C
도시수		5	10	15
경로	차량1	2-3-4	1-10-6-3-2	1-10-9-8-11-15-14-2
	차량2	1-5	4-9-8-7-5	3-4-5-6-7-12-13
거리	차량1	11903	14568	24638
	차량2	6454	19829	24621
총 거리		18357	34397	49529
최적경로의 거리		18357	28783	42418
최적해와의차 (%)		0%	16%	13%

위에서 제시된 연구 결과는, 그 정확도에 있어서는 최적해와 다소의 차이가 있으나, 문제의 크기(방문 도시 수)가 커질 경우에도 그 오차 규모가 커지지 않는 특징을 보여준다. 또한 본 접근 방법은 해의 도출 시간이 짧아 빈번한 수배송 경로 설정이 필요한 경우에서 실시간 경로 작성이 가능하며, 복수의 차량에 대한 모델 적용이 용이하다.

4.2 도시수에 따른 SOM의 수행도 평가

일반적으로 차량경로계획의 수리적 해법의 필요 시간은 도시수에 따라 지수적으로 증가하는 것으로 알려져 있다. 하지만 신경망의 일종인 SOM을 적용할 경우 도시 수가 증가하더라도 단계수는 실험 적으로 polynomial 하게 증가한다. <표 3>에

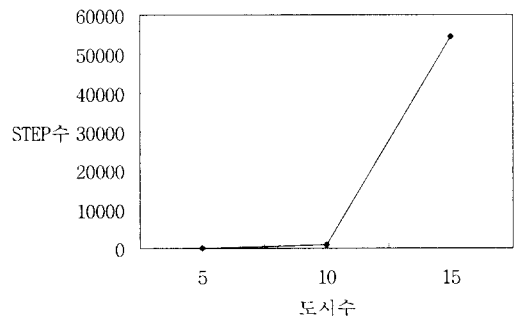
도시수에 따른 수행도를 LINGO에 의한 수리적인 방법과 비교하여 평가하였다. 여기서 수리적인 방법의 단계는 한 번의 선형계획문제의 해 발견과정을 의미하며, SOM의 반복수는 모든 노드에 대해 학습을 1회 수행하는 것을 의미한다.

<표 3>에서 알 수 있듯이 수리적인 방법으로 해를 도출하는 경우 도시수가 증가함에 따라 단계 수가 지수적으로 증가한다. 실제로 도시수가 20개 이상으로 커지면 수리적인 방법에 의해 해를 찾는 것은 거의 불가능하다. 이에 반해 SOM을 이용한 경우 도시수가 증가하더라도 반복 수행수는 실험 적으로 polynomial하게 증가한다.

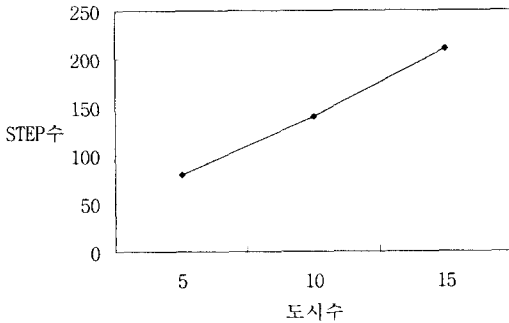
<표 3> 수리모형과 SOM의 도시 수 증가에 따른 난이도 비교

	모형 A	모형 B	모형 C
도시수	5	10	15
수리모형의 제약식의 개수	57	208	457
수리모형의 변수의 개수	58	218	478
수리적인 방법의 단계수	41	874	54515
SOM의 반복 수행수	80	140	210

[그림 9]는 도시 수에 따른 LINGO의 단계 수를 보이고 있다. 이 그림에서 도시 수가 증가함에 따라 복잡도는 지수적으로 증가하는 것을 알 수 있다. [그림 10]은 도시 수에 따른 SOM의 반복 수행 수를 나타내고 있다. 모의실험결과를 통하여 알 수 있듯이, SOM에 의한 방법은 수리적인 방법과 달



[그림 9] LINGO의 도시수에 따른 단계수



[그림 10] 도시 수에 따른 SOM의 반복 수행 수

리 도시수가 증가하더라도 반복 수행 수는 실험적으로 polynomial하게 증가한다. 또한 이 문제도 신경망칩이 개발된다면 병렬분산연산이라는 장점으로 인하여 도시 수에 상관없이 서로 비슷한 시간 내에 차량 경로계획을 세울 수 있게 된다.

5. 결론 및 추후 연구과제

차량경로계획을 필요로하는 현실적 환경은 매우 다양하다. 따라서 가장 단순한 형태의 순회 판매원 문제 모형으로는 이러한 현실을 반영하기 어렵기 때문에 보다 일반적인 방법이 필요하다. 차량경로계획을 수리적으로 풀기 위해서는 모델링을 해야 하는데 도시 수와 차량이 증가할수록 식이 복잡해진다. 또한 차량경로계획은 NP-hard에 속하는 문제이기 때문에 도시가 20개 이상만 되어도 현실적으로 계산하는 것이 불가능해진다. 본 연구는 이러한 일반화된 차량경로계획을 위해 신경망의 일종인 SOM을 이용해서 실시간으로 일반화된 차량경로계획을 세우는 알고리즘을 개발하였다. SOM을 적용할 경우 사용자가 차량경로계획을 하기 위해 수리적인 복잡한 모델링을 할 필요없이 간편하게 차량경로계획을 할 수 있다.

본 논문에서는 차량경로계획의 여러 유형 중에서 차량이 두 대 이상인 일반화된 형태를 대상으로 하여 모델을 개발하고, 차량이 두 대인 경우에 대하여 SOM의 성능을 검토하였다. 본 실험에서 SOM을

이용한 방법은 도시수가 증가하더라도 빠른 시간에 최적해에 비교적 근사한 차량경로계획을 세웠다. 본 연구에서는 비교적 규모가 작은 문제에 대해 제안된 방법의 타당성을 검증하였다. 따라서 이를 유통산업의 현실에 맞는 규모의 문제에 대해 적용하여 실시간적인 차량경로계획을 세울 수 있는 솔루션으로 개발할 예정이다. 특히 신경망 이외의 다른 방법론 중 주목을 받고 있는 유전자알고리즘과의 비교연구를 수행할 예정인데 유전자알고리즘의 경우 반드시 원의 형태를 따르지 않는 경로해 중에서 우수해를 도출해 줄 것으로 예상되므로 해의 특징비교와 적용가능성에 대한 분석도 행할 예정이다.

기타 추후 연구로는 두 대 이상의 m대의 차량에 대한 수행도를 검사하고, AGV 경로 설정, 근거리 통신망 설계와 같은 응용 문제에 적용하는 문제를 고려하고 있다. 또한 차량 경로계획의 기존 문헌에서 다루어지고 있는, 각 도시마다 요구된 방문 시간이 있는 time window를 가진 문제에 대한 연구와 차량의 운반 capacity에 제한이 있는 경우에 대한 연구도 수행하고자 한다.

참 고 문 헌

- [1] 권영탁, 이충세, 「유전자 알고리즘을 이용한 TSP」, 충북대학교 석사학위논문, 1997.
- [2] 박순달, 「OR(경영과학)」, 민영사, 1992.
- [3] Altinkemer, K. and B. Gavish, "Parallel Savings Based Heuristic for the Delivery Problem", *Operations Research*, Vol.39(1991), pp.456-469.
- [4] Angeniol, B., DeLa. Croix Vaubois G and J.LeTexier., "Self-organising feature maps and the travelling salesman problem", *Neural Networks*, Vol.1, No.4(1988), pp.289-293.
- [5] Baker, E.K., "An Exact Algorithm for the Time-constrained Traveling Salesman Problem", *Opns. Res.*, Vol.31(1983), pp.938-945.

- [6] Bodin, L.D., "Twenty Years of Routing and Scheduling", *Operation Research*, Vol.38, No.4(1990), pp.571-579.
- [7] Bookbinder, J.H. and K.E. Reece, "Vehicle Routing Consideration in Distribution System Design", *European Journal of Operation Research*, Vol.46(1990), pp.322-332.
- [8] Breedam, A.V., "Improvement heuristics for the Vehicle Routing Problem based on Simulated Annealing", *European Journal of Operational Research*, Vol.86(1995), pp.480-490.
- [9] Christofides, N. and S. Eilon, "An Algorithm for the Vehicle Dispatching Problem", *Operational Research Quarterly*, Vol.20, No.3 (1969), pp.309-318.
- [10] Christofides, N., A. Mingozzi and P. Toth., "An Algorithm for the Time Constrained Traveling Salesman Problem", Report IC-OR-81/12, Imperial College of Science and Technology, London, 1981.
- [11] Clarke, G. and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol.12, No.4(1964), pp.568-581.
- [12] Dantzig, G.B. and J.H. Ramser, "The Truck Dispatching Problem," *Management Science*, Vol.6, No.1(1959), pp.80-91.
- [13] Desrochers, M., J.K. Lensta, and M.W.P. Savelsbergh, "A Classification Scheme for Vehicle Routing and Scheduling Problem", *European Journal of Operational Research*, Vol.46(1990), pp.322-332.
- [14] Desrosiers, J., F. Soumis and M. Sauve, "Lagrangian Relaxation Methods for Solving the Minimum Fleet Size Multiple Traveling Salesman problem with Time Windows", *Mgmt. Sci.*, Vol.34(1988), pp.1005-1022.
- [15] Desrochers, M., J. Desrosiers, and M. Solomon. "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows", *Opns. Res.*, Vol.40(1992), pp.342-354.
- [16] Dumas, Y., J. Desrosiers, E. Gelinas and M. Solomon., "An Optimal Algorithm for the Traveling Salesman Problem with Time Windows", *Opns. Res.*, Vol.43(1995), pp.367-371.
- [17] Gendreau, M., A. Hertz and G. Laporte., "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem", *Opns. Res.*, Vol.40(1992), pp.1086-1094.
- [18] Gendreau, M., A. Hertz and G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science*, Vol.40, No.10(1994), pp.1276-1290.
- [19] Gendreau, M., A. Hertz, G. Laporte and M. Stan., "A Generalized Insertion Heuristic for the Traveling Salesman with Time Windows", *Opns. Res.*, Vol.43(1998), pp.330-335.
- [20] Kohonen, T., "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, Vol.43(1982), pp.59-69.
- [21] Holmes, R.A. and R.G. Parker, "A vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme," *Operations Research Quarterly*, Vol.27, No.1(1976), pp.83-92.
- [22] Hopfield, J.J. and D.W. Tank, "Neural computation of Decisions in optimisation problems", *Biological Cybernetics*, Vol. 52(1985), pp.141-152.
- [23] Osman, I.H., "Metastrategy Simulated Annealing and Tabu search Algorithms for the Vehicle Routing Problem", *Annals of Oper-*

- ations Research, Vol.41(1993), pp.421-451.
- [24] Pesant, G., M. Gendreau, J. Potvin and J. Rousseau., "An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows", *Transportation Science*, Vol.32, No.1(1998), pp.12-29.
- [25] Savelsbergh, M., "Local Search in Routing Problems with Time Windows", *Ann. Oper. Res.*, Vol.4(1985), pp.285-305.
- [26] Savelsbergh, M., "The Vehicle Routing Problem with Time Windows : Minimizing Route Duration", *ORSA J. Computing.*, Vol. 4(1992), pp.146-154.
- [27] Somhom, S., A. Modares and T. Enkawa, "A Self-organising model for the travelling salesman problem", *Journal of Operational Research Society*, Vol.53(1997), pp.919-928.
- [28] Thangiah, S.R., "Genetic Algorithms, Tabu Search and Simulated Annealing Methods for Vehicle Routing Problems with Time Windows", *Practical Handbook of Genetic Algorithms : Complex Structures*, L. Chambers (Ed.), CRC Press, 1998.