'95 대한전기학회 하계학술대회 논문집(1995. 7. 20~22)

SOFM(Self-Organizing Feature Map)형식의 Travelling Salesman 문제 해석 알고리즘

○석 진욱, 조 성원, 최 경삼
홍익대학교 전기·제어공학과

Self Organizing Feature Map Type Neural Computation Algorithm for Travelling Salesman Problem

Jinwuk Seok, Seongwon Cho, Gyung-sam choi
Dept. of Electrical & Control Eng., Hong Ik University.

Abstract

In this paper, we propose a Self Organizing Feature Map (SOFM) Type Neural Computation Algorithm for the Travelling Salesman Problem(TSP). The actual best solution to the TSP problem is computatinally very hard. The reason is that it has many local minim points. Until now, in neural computation field, Hopield-Tank type algorithm is widely used for the TSP. SOFM and Elastic Net algorithm are other attempts for the TSP. In order to apply SOFM type neural computation algorithms to the TSP, the object function forms a euclidean norm between two vectors. We propose a Largrangian for the above request, and induce a learning equation. Experimental results represent that feasible solutions would be taken with the proposed algorithm.

I. 서 론

Travelling Salesman Problem(TSP)은 N개의 도시를 모두 연결하는 최단의 경로를 찾아내는 문제로서, 계산 시간이 도시의 숫자 N의 지수함수로 증가하는 NP-Compete 문제중의 하나이다. 이 문제는 N 도시의 순서를 일부분 바꾸었을때 경로가 증가하게 되는 국소 최소점이 많이 있는 문제이다. 이렇게 국소 최소점이 많은 울퉁불퉁한 표면(Rugged Landscape)에서 전체적인 최소점을 찾는 문제는 변수공간 전체에 대한 검색을 하지 않는 한 불가능 하다. 따라서 근사적인 전체 최소점을 찾는데 주안점을 둔 몇가지 최적화 알고리즘으로는 첫번째로 담금질 (Simulated Annealing)방법이 있으며, 두번째로 신경회로망을 사용한 방법이 있다[3][8]. 이중 잘 알려진 Simulated Annealing방법을 제외한 방법은 최근에 패턴인식등에 많이 응용되는 신경회로망을 TSP에 적용하여 경로를 최적화게 학습을 시키는 것으로 Hopfield-Tank모델의 적용 예제로서 10 city이하의 간단한 문제에서는 항상 최적한 해답을 내은다. 반면 Hopfield-Tank모델은 국소 최소점에 매우 취약하여 city의 수가 클 경우 Feasible한 해를 구하기가 어려운 단점이 있다[8][9].

Hopfield-Tank모델이외에 신경회로망을 사용하여 TSP문제를 해결한 예는 SOFM(Self Organizing Feature Map)알고리즘과 Durbin-Wilshow알고리즘으로 알려진 Elastic Net 방법이 있다. SOFM알고리즘을 사용한 TSP문제해결은 SOFM알고리즘이 가지고 있는 Weight vector의 정렬특성을 이용한 해석(B.Angeniol, H.Ritter 등)으로서 다른 알고리즘에 비해 비교적 City수가 많은 문제에 대하여 전역 최소점 근처로 Weight Vector를 위치시킬 수 있는 장점이 있다. Elastic Net 방법은 SOFM알고리즘에 Weight Vector의 거리를 또 하나의 Energy 항으로 부가하여 보다 거리 최적화를 이루도록 하는 방법으로서 SOFM알고리즘의 Weight vector 정렬특성을 대수적으로 보다 명확히 한 특징이 있다. 하지만 SOFM알고리즘과 Elastic Net 두 알고리즘은 Weight Vector와 City Location으로 표현되는 입력 Vector가 동일 Dimension의 Vector Space에 위치하기 때문에 Kalman Filter적 특성, 즉 입력 Vector Cluster의 중심으로 Weight Vector가 위치하는 특징이 사라지지 않는다는 단점이 있다. 이경우 몇개의 city 중심에 Weight Vector가 위치하므로 좁은 영역에 다수의 city가 몰려 있을 경우 이때의 최적 경로를 구해내기가 어려울 뿐만 아니라 이러한 단점을 없애기 위해 Weight Vector를 많이 사용할 경우 불 필요한 Weight Vector를 지워가는 루처리가 또 다시 요구되며 연산시간의 연장, Network 효율의 저하를 피할 수 없을뿐 아니라 이 단점을 완전히 없앨 수도 없다[8][9].

SOFM알고리즘과 Elastic Net 알고리즘이 이러한 단점을 가지는 가장 큰 이유는 Weight Vector가 존재하는 위상 공간이 기본적으로 입력 벡터공간이기 때문이다. 따라서 Weight Vector공간과 입력 벡터 공간을 분리하고 입력 벡터 공간과의 관계를 선형적으로 가지도록 하면 기존 SOFM 알고리즘이나 Elastic Net 알고리즘으로 TSP문제를 풀었을때의 단점을 해결 할 수 있다.

본 연구에서는 이러한 목적을 위해 먼저 City Location을 표시하는 Index Matrix를 도입하여 Weight Vector공간과 입력 벡터 공간을 분리하고, Index Matrix를 변화시키는 Permutation Matrix를 도입하여 Weight Vector와 City Location과의 관계를 설정하였다. 이때 Weight Vector들은 Permutation Matrix를 구하기 위한 연관 Paramter Matrix의 Component를 의미하게 되며, 여기에 기초하여 기존 Hopfield-Tank Model이나 Simulated Annealing방법등에서 사용한 Perfect Matching 형식의 Lagrangian대신 실제 Travelling 거리를 목적함수로 놓은 후 학습 규칙을 유도하였다. 본 연구에서는 또한 기존의 최급 강하법으로 유도된 학습 알고리즘이 가지는 약점을 보완하기 위해 이진 강화함수를 도입한 학습 알고리즘을 제안하였다.

II. TSP 문제를 위한 목적함수 유도

기존의 TSP문제를 위한 목적함수는 Perfect Matching 형식으로서 각 도시 사이의 거리 Matrix를 정의하고 몇 가지 등식 제한조건들을 포함하여 유도 되었다. 이러한 목적함수의 형태는 Hopfield-Tank 모델과 같은 신경망의 경우 매우 적절하게 적용될 수 있으나, SOFM과 같이 최소 자승법을 목적함수로 삼는 모델의 경우에는 적용하기 어려웠다. 따라서 TSP문제의 목적함수를 최소 자승법과 같은 형태로 만들기 위해서는 목적함수를 다음과 같이 국부 거리의 합의 형태로 놓아야 한다.

$$\sum_{j=0}^{n-1} \| x_{\phi(\vec{w}_{j+1})} - x_{\phi(\vec{w}_{j-1})} \|^2 \qquad (1)$$

식 (1)에서 n은 도시의 수, $\phi(\vec{w}^t_j)$는 Permutation 을 위한 Weight 벡터 $\vec{w}^t_j$에 의해 결정되는 도시의 index, w_{ij}^t는 Weight 벡터 $\vec{w}^t_j$의 i번째 성분으로서 시간 t에서 j번째 도시를 가리키는 Index가 i번째 도시를 가리키는 Index로 바뀔 확률에 비례하는 Paramter가 된다. $x_{\phi(\vec{w}_j)}$ 는 $\phi(\vec{w}^t_j)$번째 도시의 위치 벡터(x, y)성분을 가지며 다음으로 결정된다.

$$x_{\phi(\vec{w}_j)} = A \cdot \vec{p}^t_j \qquad (2)$$

식 (2)에서 Matrix A는 2×n Matrix 로서 도시의 위치 벡터(x, y)성분으로 이루어져 있으며 j번째 Column Vector는 j 번째 도시의 (x, y)성분을 가지고 있으며, $\vec{p}^t_j$는 $\phi(\vec{w}^t_j)$로 만들어지는 Index Matrix P의 j번째 Column 벡터 성분이다.

TSP문제를 위한 제한 조건은 일반적으로 다음과 같다.

① 각 도시를 한번만 방문할 것
② 동시에 두 도시를 방문 할 수 없을 것
③ 방문하는 도시의 수가 도시의 전체 수 일것

이 제한 조건들은 Index Matrix에 모두 포함 되어 있으므로 Weight Matrix가 Index Matrix와 유사한 형태로 나타나면 제한 조건들은 만족되므로 제한조건 식은 다음과 같은 등식 제한조건식으로 유도 된다.

$$\sum_{i}^{n-1}\sum_{j}^{n-1}\| w_{ij}^{t} - p_{ij}^{t}\|^2 = 0 \qquad (3)$$

제한조건은 이외에도 $0 \le w_{ij}^{t} \le 1$이 필요하나 부분적으로 (3)식에 포함된다. 따라서 TSP를 위한 Largrangian은 다음과 같이 정의할 수 있다.

$$\mathcal{L} = \sum_{j=0}^{n-1}\|x_{\phi(\vec{p}_{j+1})} - x_{\phi(\vec{p})}\|^2 + \lambda\sum_{i}^{n-1}\sum_{j}^{n-1}\| w_{ij}^{t} - p_{ij}^{t}\|^2 \qquad (4)$$

식 (4)에서 λ는 Largrangian Multiplier이다.

Weight 벡터의 각 요소 w_{ij}^{t}와 Index Matrix의 i번째 Row, j번째 Coulmn사이의 관계는 다음과 같다.

$$p_{ij}^{t} = u(\sum_{k}^{n-1} M_{ik}^{m}\, w_{kj}^{t} - \varepsilon') = w_{kj}^{t} f(\vec{w}_j^{t}, m) \qquad (5)$$

여기에서 $M_{ki} = \delta(k-i)(1+\varepsilon) - \varepsilon$ 이며, ε, ε'는 매우작은 임의의 양수, m은 $\vec{w}_j^{t}$의 성분중 하나만 ε'보다 클때까지의 수이며 $f(\vec{w}_j^{t}, m)$는 가장 큰 $\vec{w}_j^{t}$의 성분중 하나에 대하여 $1/w_{kj}^{t}$의 값을 가지고 나머지에 대하여 0을 갖는 함수이다.

Ⅲ. SOFM형식의 학습 방정식 유도

학습을 위한 목적함수는 2절에서 구한 Largrangian을 사용하여 다음과 같은 Entropy함수로 놓는다.

$$H = -\sigma^2 \log_n \sum_k \exp(-\mathcal{L}_k/\sigma^2) \qquad (6)$$

이 함수는 Largrangian이 최소의 값을 가질때와 정보의 양이 가장 작도록 하는 각 k, 즉 전체 도시의 수에 대한 라그랑지안이 평균값에 접근 할때 가장 작은 값을 가지도록 하여 Energy Contour의 변화가 극소 최소점의 경우보다 비교적 작은 전역점 주변을 찾아갈 수 있도록 한다.

학습 방정식은 (6)식에 최급 강하법 (Gradient Descent Rule)을 적용하여 다음과 같이 유도한다.

$$\begin{aligned} w_{kj}^{t+1} &= w_{kj}^{t} - \frac{\partial H}{\partial w_{kj}} \\ &= w_{kj}^{t} - \eta\frac{\partial H}{\partial \mathcal{L}_j}\cdot\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}} \\ &= w_{kj}^{t} - \eta \log_n e \frac{\exp(-\mathcal{L}_j/\sigma^2)}{\sum_k \exp(-\mathcal{L}_k/\sigma^2)}\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}} \end{aligned} \qquad (7)$$

식 (7)에서 $\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}}$를 구하기 위해 먼저 $\frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}}$를 구하면

$$\frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}} = -2(a_{1k}^2 + a_{2k}^2)(p_{kj+1}^{t} + p_{kj-1}^{t} - 2p_{kj}^{t}) - 2\lambda(p_{kj_n}^{t} - p_{kj}^{t})$$

$$\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}} = \frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}}\frac{\partial p_{kj}^{t}}{\partial w_{kj}^{t}} = \frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}} f(\vec{w}_j^{t}, m) \qquad (8)$$

식 (5)에서 $\frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}}$는 $\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}}$와 $f(\vec{w}_j^{t}, m)$ 함수의 특성 상 비례 관계에 있으므로 다음과 같이 근사화 할 수 있다.

$$\frac{\partial \mathcal{L}_j}{\partial p_{kj}^{t}} = \alpha\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}} \qquad (9)$$

식 (9)에 의해 $\frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}}$는 다음과 같이 유도 된다.

$$\begin{aligned} \frac{\partial \mathcal{L}_j}{\partial w_{kj}^{t}} &= -2\alpha(a_{1k}^2 + a_{2k}^2)(w_{kj+1}^{t} + w_{kj-1}^{t} - 2w_{kj}^{t}) \\ &\quad -2\lambda\alpha(p_{kj_n}^{t} - w_{kj}^{t}) \\ &= -2\alpha(a_{1k}^2 + a_{2k}^2)(w_{kj+1}^{t} + w_{kj-1}^{t}) - 2\lambda\alpha\, p_{kj_n}^{t} \\ &\quad +4(a_{1k}^2 + a_{2k}^2)\alpha\, w_{kj}^{t} + 2\lambda\alpha\, w_{kj}^{t} \\ &= -2\alpha(((a_{1k}^2 + a_{2k}^2)(w_{kj+1}^{t} + w_{kj-1}^{t}) + \lambda\, p_{kj_n}^{t}) \\ &\quad -(2(a_{1k}^2 + a_{2k}^2) + \lambda)\, w_{kj}^{t}) \end{aligned} \qquad (10)$$

식 (10)에서

$$x_j = \frac{(a_{1k}^2 + a_{2k}^2)(w_{kj+1}^{t} + w_{kj-1}^{t}) + \lambda\, p_{kj_n}^{t}}{2(a_{1k}^2 + a_{2k}^2) + \lambda} \qquad (11)$$

로 놓으면 학습 방정식은 다음과 같이 SOFM학습 방정식의 형태를 따른다.

$$w_{kj}^{t+1} = w_{kj}^{t} + \eta\, 2\alpha \log_n e \frac{\exp(-\mathcal{L}_j/\sigma^2)}{\sum_k \exp(-\mathcal{L}_k/\sigma^2)}(x_j - w_{kj}^{t}) \qquad (12)$$

식 (12)는 그러나 극소 최소점이 많은 울퉁불퉁한 표면(Rugged Landscape)에 대하여 극소 최소점에 도달했을때 Weight 값의 변화가 미소하므로 전역 최소점을 찾는데 상당한 문제가 있다. 따라서 식 (12)는 적절한 방법으로 이를 빠져 나갈 수 있도록 일부 수정 되어야 한다.

Ⅳ. 이진 강화 함수와 학습 방정식의 수정

이진 강화 함수는 $\vec{w}_j^{t}$를 보다 전역 최소점(Global Minimum Point)으로 가깝게 보내게 하는데 쓰인다. 이진 강화 함수를 이용한 적응 필터 알고리즘은 다음과 같다[6].

$$\begin{aligned} c_{k+1} &= c_k - \alpha\, u_k\, sgn[e_k] \\ e_k &= c_k \cdot u_k - d_k \\ u_k &= \varphi(x_k, x_{k-1}, \ldots) \end{aligned} \qquad (13)$$

식 (13)에서 φ는 관찰 과정(Observable Process) x_k를 데이타 과정 u_k로 보내는 사상이며, d_k는 요구 과정(Desirable Process), α는 일정 적응 이득(Constant Adaptive gain)이다.

여기에서 $u_k = 1$로 놓고 $d_k = c^* \cdot u_k + f_k$로 놓으면 f_k는 데이타 신호가 된다. c_k와 d_k가 동일 공간상에 존재한다면 이것은 확률적 최급 강하법의 한 종류가 된다. 이에 따라 위 방정식을 고쳐쓰면 다음과 같다.

$$\begin{aligned} w_{k+1} &= w_k - \rho(t)\, sgn[e_k] \\ e_k &= w_k - x_k \\ x_k &= w^* + n_k \end{aligned} \qquad (14)$$

식 (14)에서 x_k는 데이타 벡터 열, w^*는 w의 최적 평형 벡터 n_k는 오차 혹은 잡음 벡터 열 $\rho(t)$는 시간에 따라 1의 발생확률이 감소하는 랜덤변수 로서 다음을 만족한다.

$$\rho(t) \to 0 \quad as \quad t \to \infty$$

이진 강화 함수의 수렴성 판별은 또한 다음식에 의하여 구한다.

$$\begin{aligned} \lim_{n\to\infty}\sup \frac{1}{n}\sum_{k=1}^{n} E|e_k| &\le E|e^*| + E\rho(t)h \\ &\le E|e^*| \end{aligned}$$

이진 강화 함수를 학습 방정식에 적용하기 위하여 다음과 같이 Fokker-Plank 방정식의 형태로 수정한다. 이때 2차 편미분 항은 일반적인 Fokker-Plank 방정식과는 달리 3차 이상의 편 미분 항도 반영할 수 있다고 가정한다[4].

$$w_{kj}^{t+1} = w_{kj}^{t} - \alpha\frac{\partial H}{\partial w_{kj}} + \beta\frac{\partial^2 H}{\partial w_{kj}^2} \qquad (15)$$

이 방정식이 일반적인 Brownian 운동을 기술 할 수 있다면 극소 최소점에서 빠져 나갈 수 있는 충분한 에너지가 Weight Vector에 있는 것으로 생각할 수 있다. Fokker-Plank 방정식 에서 Brownian 운동은 1차 편미분 항과 2차 편미분 항의 부호가 같으므로 학습방정식은 다음과 같이 놓을 수 있다[4]. 또한 $(x_j - w_{kj}^{t})$항 앞의 적응 이득 부분은 시간에 대한 함수로 대치한다.

$$w_{kj}^{t+1} = w_{kj}^{t} + \alpha(t)(x_j - w_{kj}^{t}) + \rho(t)\beta\, sgn(x_j - w_{kj}^{t}) \qquad (16)$$

식 (16)에서 $\alpha(t)$는 t에 대한 단조감소 함수이며 β는 이진 강화 함수의 적응 이득 $sgn(\cdot)$은 1, 0 혹은 -1의 값을 갖는 Sigmoid 함수이다.

Ⅴ. 실험 결과

본 논문에서 제안한 알고리즘은 10 city ring형 배치, 10 city random 배치, 14 city random 배치, 20 city ring형 배치, 20 city random 배치들에 대하여 수행했다. 각 도시들의 위치 벡터들은

$C^2[0,1]$ 공간 상에 놓았으며 다음의 학습순서에 따라 실험을 수행 하였다.

1. 각 Paramter들을 초기화한다. Matrix Pj_{ps}^t, P^t, W^t, PM_j^t 의 (i, j)번째 Component들은 다음과 같다.

$$p_{ij}^0{}_{ps} = p_{ij}^0 = w_{ij}^0 = pm_{ij}^0 = \delta(i-j) \qquad (17)$$

식 (17)에서 Pj_{ps}^t, 은 Epoch t에서의 j 번째 도시에 대한 Index Matrix로서 j 번째 Coulmn을 제외한 모든 요소는 P^t와 같다. PM_j^t는 j 번째 도시에 대한 Permutation Matrix로 Index Matrix를 변화 시키는데 사용된다.

2. city j에대한 w_{kj}^{t+1}를 모든 k에대하여 제안한 학습 방정식을 통해 구한다.
3. 학습 방정식을 통해 구해진 p_{kj}^{t+1}을 Normalize한다.

$$w_{kj}^{t+1} = \frac{w_{kj}^{t+1}}{\sum_k w_{kj}^{t+1}}$$

이로서 제한조건 $0 \le w_{ij}^t \le 1$이 만족된다.

4. $i = \arg\max_k w_{kj}^{t+1}$를 사용하여 j 번째 city에대한 Permutation Matrix PM_j^{t+1} 를 구한다. i는 다음 도시에 대한 Index, j는 현재도시에 대한 Index를 의미한다.
5. 현재의 Wj_{ps} Matrix에 대하여 다음을 구한다.

$$Pj^{t+1}{}_{ps} = PM_j^{t+1} \; P^t{}_{ps}$$

6. 이때의 Index Energy $E_j^{t+1} = \exp(-\mathcal{L}_j/\sigma^2)$를 $P_j^{t+1}{}_{ps}$ 에 대하여 구한다.
7. 전체 j에대하여 2~5를 반복한다.
8. 현재 Epoch t+1 에서의 최대 Index Energy E_{max}^{t+1}와 Index인 max를 구하여 그때까지의 Index Energy E_{max}^t 와 비교하여 다음에 따라 Index Matrix를 구한다.

$$\begin{cases} E_{max}^{t+1} \ge E_{max}^t & P^{t+1} = PM_{max}^{t+1} \; P^t \\ E_{max}^{t+1} < E_{max}^t & P^{t+1} = P^t \end{cases}$$

9. 이를 정해진 Epoch가 끝날때 까지 반복한다.

제안한 알고리즘에서 각각의 적용 이득은 다음과 같다.

$$\alpha(t) = 1 - \frac{\text{현재 } Epoch \text{ 수}}{\text{전체 } Epoch \text{ 수}}$$

$$\rho(t) = \begin{cases} x_1 \ge x_0 & 1.0 \\ x_1 < x_0 & 0.0 \end{cases}$$

$$\beta = 0.15 \qquad \lambda = 0.0025$$

위 식에서 x_1은 전체 Epoch 수에서 현재 Epoch 수 사이에 random 하게 선택된 수이며 x_0는 현재 Epoch 수의 1/10 사이에 Random하게 선택된 수이다. Ring으로 배치 된 도시의 경우 20 City까지 초기 Random한 배치에서 Ring 배치를 찾아 냈으며 $C^2[0,1]$ 공간 상에 City들이 Random하게 배치된 경우 그림과 같이 20 City 까지 최적 경로를 찾아냈다.

VI. 결 론

본 논문에서는 이진 강화 함수가 결합된 SOFM형식의 TSP문제 해결 알고리즘을 제안하고 이를 위한 Largrangian을 유도 하였으며 실험을 통해 제안한 알고리즘의 타당함을 보였다. 본 알고리즘은 도시의 수가 N으로 증가할 경우 연산시간은 거의 N^2배 증가하므로 비교적 빠르게 해를 구할 수 있으며 Largrangian Multiplier를 고려할 필요 없이 고정된 값으로도 Feasible한 해를 구할 수 있다. 제안한 알고리즘은 국소 최소점에 빠지더라도 이진 강화함수와 1차 편미분항의 상호 작용으로 국소 최소점을 빠져 나갈 수 있는 특징이 있다. 그러나 제안한 알고리즘은 언제나 최적해를 보장 할 수 있는 것은 아니며 비교적 최적해에 가까운 근사해를 보장 할 뿐이다. 그 이유는 국소 최소점에서 1차 편미분 항의 크기가 매우 작을 뿐 아니라 이진 강화 함수가 양자화 된 값만을 발생 시키므로 이에 의해 보다 넓은 범위에 대한 검색을 할 수 없기 때문이다. 이 경우 Epoch수를 보다 크게 할 경우 그 만큼 넓은 범위에 대한 검색을 행할 수 있다. 제안한 알고리즘은 TSP문제를 응용한 여러 문제에 모두 적용할 수 있으며 앞으로 알고리즘의 동적 특성을 분석하여 보다 최적해를 찾을 확률을 높이는 연구도 진행 될 것이다.

참고문헌

[1] H.Ritter K.Schulten "Convergence Properties of Kohonen's Topology Conserving Maps : Fluctuations, Stability, and Dimesion Selection" Biol.Cybern. 60,59-71 (1988)
[2] T.Kohonen 'Self Organization and Associative Memory' Springer-Verlag 3rd printng 1989
[3] E. Domany J.L.van Hemmen K.Schulten (Eds.) 'Models of Neural Networks' Springer-Verlag 2nd printng 1992
[4] van Kampen Stochastic Processes in Physics and Chemistry North-Holland .1981. Armsterdam
[5] C. Darken J.Moody "Note on Learning Rate Schedule for Stochastic Optimization "
[6] A.Gersho "Adaptive Filtering with Binary Reinforcement" IEEE Trans.Inform. vol IT-30, 2,March 1984
[7] L. Botton P.Gallinari "A Framework for the Cooperation of Learning Algorithm" pp781 - 788
[8] S.Y.Kung Digital Neural Networks Prentice Hall, Englewood Cliffs, N.J. 1993
[9] H.Ritter K.Schulten T.Martinetz 'Neural Computation and Self-Organizing Maps' Addison Wesley Publishing Company, 1992.

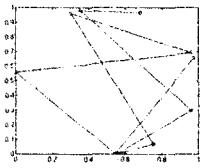

그림 1
10City Ring형 초기배치

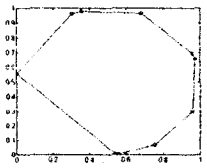

그림 2
100 Epoch 학습결과

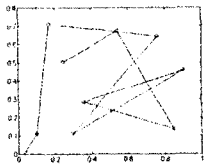

그림 3
10 City Random 초기 배치

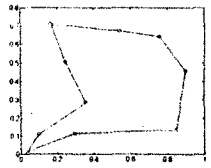

그림 4
100 Epoch 학습결과
총 길이 0.902072

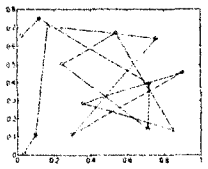

그림 5
14 City Random 초기위치

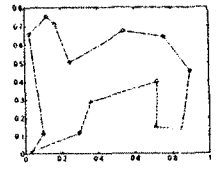

그림 6
400 Epoch 학습결과
총 길이 0.904554

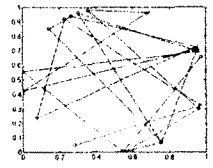

그림 7
20 City Ring형 초기배치

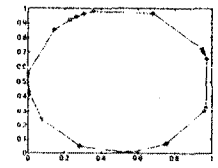

그림 8
400 Epoch 학습결과

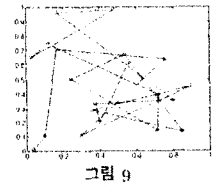

그림 9
20 City Random 초기배치

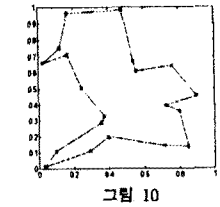

그림 10
400 Epoch 학습결과
총 길이 0.909072