

대규모 TSP과제를 효과적으로 해결할 수 있는 SOFM알고리즘

(An Efficient Algorithm based on Self-Organizing Feature Maps for Large Scale Traveling Salesman Problems)

金 宣 鍾*, 崔 興 文*

(Seon Jong Kim and Heung Moon Choi)

要 約

본 논문에서는 대규모 TSP과제의 해를 효과적으로 구할 수 있는 SOFM 알고리즘을 제안하였다. 제안된 알고리즘은 경쟁에서 이긴 뉴런을 학습에는 포함시키되 다음 경쟁에서는 제외시킴으로써 Angeniol 등의 알고리즘과는 달리 N개 도시 TSP문제를 학습 중 일정한 N개의 뉴런과 2N개의 연결세기만 가지고도 풀수 있을 뿐만 아니라, 뉴런을 삽입 또는 제거하는 별도의 알고리즘이 필요 없어 더욱 효과적이다. 시뮬레이션 결과, 제안된 알고리즘은 30개 도시나 1000개 도시 TSP에서 공히 기존의 알고리즘보다 속도를 약 30%정도 개선시킬 수 있었으며, 뉴런의 출력값을 직접 학습범위 적응조정에 이용함으로써 준최적해로의 수렴률도 크게 높일 수 있었다.

Abstract

This paper presents an efficient SOFM(self-organizing feature map) algorithm for the solution of the large scale TSPs(traveling salesman problems). Because no additional winner neuron for each city is created in the next competition, the proposed algorithm requires just only the N output neurons and 2N connections, which are fixed during the whole process, for N-city TSP, and it does not requires any extra algorithm of creation or deletion of the neurons. And due to direct exploitation of the output potential in adaptively controlling the neighborhood, the proposed algorithm can obtain higher convergence rate to the suboptimal solutions. Simulation results show about 30% faster convergence and better solution than the conventional algorithm for solving the 30-city TSP and even for the large scale of 1000-city TSPs.

1. 서론

TSP(traveling sales problem)와 같은 조합형

최적화(combinatorial optimization) 문제는 과학 및 공학분야에서 자주 나타나는 계산집약적인 문제 중의 하나이다. 특히, 산업이나 기술이 발달함에 따라 이와같은 최적화가 적용되는 VLSI 칩 배치문제, 영상처리 및 인식 등 그 문제의 크기가 대규모화 되는 경향이 있다. 지금까지 이러한 대규모의 최적화 문제를 해결하는 데는 유전 알고리즘(genetic algorithms)^[1,2], Christofides 알고리즘^[3] 그리고 신경

*正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)

接受日字: 1992年 2月 26日

회로망^[4,9]을 이용하는 방법등이 제시되고 있다. 이 중에서 대규모의 병렬성을 가지고 있는 신경회로망을 이용하여 대규모의 TSP를 해결하기 위한 연구가 특히 많이 진행되고 있다 신경회로망을 이용하여 TSP를 푸는 방법은 Hopfield 신경회로망을 적용하는 경우와 Kohonen의 SOFM(self-organizing feature map)을 적용하는 경우로 크게 구분될 수 있다.

Hopfield 등^[4]은 N개 도시의 TSP를 뉴런의 출력값으로 사상된 에너지함수로 표현함으로써 최적화를 시도하였다. 그러나 이 때는 N²개의 출력뉴런과 N⁴개의 연결세기의 수가 요구될 뿐만 아니라, 국부 최소점들(local minima)에 대한 처리가 어려워 수십개 이상의 도시를 갖는 대규모 TSP 과제를 효과적으로 해결하기는 불가능하다.

Angeniol 등^[5]은 대규모 TSP에 Kohonen의 SOFM을 적용하였다. SOFM의 모델에서는 출력뉴런을 1차원으로 배열하기 때문에 도시 수 N의 증가에 따라 출력뉴런의 수가 선형적으로 증가하므로 대규모 TSP문제의 해결이 용이하다. 그러나 학습중에 두개의 서로 다른 도시에 대해 승리하는 뉴런이 있으면 새로운 뉴런을 추가하기 때문에 출력뉴런수가 kN개가 요구된다.

이 때 k는 뉴런제거 반복 횟수로서 Angeniol의 경우 k=3이었다. 따라서 연결세기의 수가 2kN개이고 새로운 뉴런을 삽입 또는 제거하는 특별한 알고리즘이 필요하기 때문에 학습속도도 느다.

본 논문에서는 SOFM을 대규모의 TSP에 적용함에 있어서 경쟁에서 이긴 뉴런은 학습에는 포함시키되 다음 경쟁에서는 제외시킴으로써 N개 도시 TSP 문제에 N개의 뉴런과 2N개의 연결세기만 가지고도 효과적으로 해를 구할 수 있는 알고리즘을 제안하였다. 제안된 알고리즘에서는 소요되는 뉴런의 수가 단지 N개의 적은 수로 고정되어 있을 뿐만 아니라 뉴런의 삽입 및 제거를 위한 별도의 알고리즘이 필요치 않아 수렴속도를 증대시킬 수 있다. 또한 뉴런의 실제 출력값을 학습범위(neighborhood) 적응조정에 이용하여 더 낮은 학습률을 이용할 수 있도록 함으로써 해의 성능도 개선시킬 수 있었다. 시뮬레이션을 통하여 제안된 방법과 기존의 방법을 비교검토 하였고 대규모 과제인 1000개 도시 TSP문제의 해도 구하였다.

II. 대규모 TSP를 위한 SOFM

전술한 바와 같이 SOFM을 이용하면 도시의 수에 일치비례하는 뉴런의 갯수만 가지고도 대규모 TSP

문제를 용이하게 해결할 수 있다. 그러나 Angeniol 등의 알고리즘에서는 경쟁학습에서 새로운 뉴런이 추가되기 때문에 출력뉴런수가 kN개, 연결세기의 수가 2kN개가 필요하고 새로운 뉴런을 삽입 또는 제거하는 별도의 알고리즘이 필요하다. 그러나 본 논문에서는 한번 승리한 뉴런을 학습에는 포함시키되 다음 경쟁에서는 제외시킴으로써 N개의 뉴런수와 2N개의 연결세기의 수만 가지고도 별도의 알고리즘 없이 효과적으로 대규모 TSP의 최적해를 구할 수 있도록 하였다.

1. 기존의 SOFM

먼저 기존의 SOFM 신경회로망에 대하여 살펴보면, Kohonen의 SOFM 신경회로망에서는 뉴런간의 거리에 따라서 정렬하려는 성질을 가지고 있으며 비지도 학습으로 뉴런 상호간에 on-center off-surround의 상호작용을 한다. 즉, 어떤 입력 패턴에 대해 연결세기의 값이 가장 유사한 뉴런이 승리뉴런이 되고, 승리뉴런 주위의 가까운 뉴런들은 입력패턴에 강하게 반응하여 학습되며 좀 더 거리가 먼 뉴런들은 입력 패턴에 대해 반응이 억제되어 멕시코 모자(Mexican hat) 형태의 상호작용을 하게 된다. 그림 1은 2개의 입력뉴런과 N개의 1 차원 배열을 갖는 출력뉴런으로 구성된 Kohonen의 SOFM 신경회로망 모델이다. 입력층과 출력층 사이의 연결세기가 완전결합(full connection)을 하고 있으며 모든 정보는 이 연결세기에 학습기억된다.

이러한 Kohonen의 SOFM 신경회로망은 승리뉴런과의 거리가 학습범위(neighborhood) G 내에 존재하는 모든 뉴런 j 에 대하여는

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t) (x_i - w_{ij}(t)) \tag{1}$$

로 학습시키고, 그렇지 않은 뉴런들은

$$w_{ij}(t+1) = w_{ij}(t) \tag{2}$$

로 학습시킨다. 여기서, x_i 는 입력값을, V_j 는 출력값이다. 그리고 w_{ij} 는 출력뉴런 j 와 입력뉴런 i 와의 연결세기를 나타낸다. 이러한 Kohonen의 학습방법에서는 학습률 η 를 시간 t 내에서 뉴런간의 거리와 무관하게 일정하게 함으로써 수렴속도가 느리다는 단점을 가지고 있다.

일반적으로 SOFM에서는 학습률 η 를 다음과 같이 뉴런간의 거리에 따른 가우스함수 형태로 변화시킴으로써 수렴속도는 개선될 수 있다. 즉,

$$\eta_j = \exp(-d_j^2/G^2)/\eta_0 \quad (3)$$

이다. 이 때 d_j 는 뉴런 j 와 승리뉴런간의 거리로서 직접 이웃하는 뉴런간에는 $d_j = 1$ 로 한다. 식 (3)은 Kohonen 에서와는 달리 그림 2에 도시 한 바와 같이 가우스함수를 적용하여 승리뉴런과 가까운 뉴런일 수록 더 강하게 학습시켜 전체적인 학습시간을 단축시키는 효과를 줄 수 있다. 그리고 식 (3)에서 η_0 는 학습률의 반영정도를 나타내는 상수로서 그 크기가 작으면 학습률이 크게 반영되어 수렴속도는 빨라지나 해의 성능이 저하되므로 이 값을 줄이는 데는 한계가 있으며 따라서 그에 대한 고찰이 필요하다.

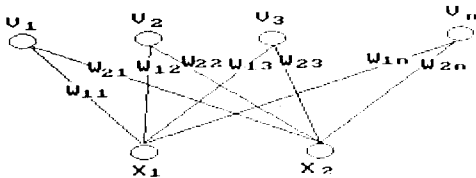


그림 1. 2입력 N출력 및 2N시냅스연결을 갖는 Kohonen의 SOFM 신경회로망 모델

Fig. 1. SOFM neural network model with 2 input and N output neurons and 2N connections.

2. 대규모 TSP를 위한 제안된 SOFM

이제 SOFM 신경회로망을 이용하여 Angeniol 등과는 달리 학습중에 항상 N 개로 일정한 출력뉴런만 가지고도 대규모 TSP 문제를 해결할 수 있는 제안된 알고리즘에 대하여 살펴보자. 잘 알려진 바와 같이 TSP에서는 각 도시를 단 한번씩만 방문하여 N 개 도시 모두를 가능한 짧은 거리로 방문하는 데 그 목적이 있다. 이러한 TSP 문제를 해결하기 위해서는 먼저 원하는 해를 뉴런의 출력으로 표시하여야 한다. TSP 문제를 SOFM신경회로망에 사상시키는데 있어서 승리뉴런은 해당도시를 대표하고 그 뉴런의 위치는 방문도시의 위치로 하면, 각 뉴런의 배열 순서가 곧 도시의 방문순서가 된다. 사상된 뉴런들을 1 차원으로 배열하고 SOFM의 반복적인 학습 알고리즘을 수행하면, 각출력뉴런들은 주어진 도시분포에 대해 최종적으로 최단거리로 순회할 수 있는 상태로 수렴한다. 이것은 도시간의 거리에 의존하는 TSP 문제를 뉴런간의 거리에 의존하는 SOFM에 사상시킨 것과 같다. 따라서 TSP문제 해결을 위한 SOFM 신경회로망 모델은 그 에너지가 뉴런간의 거리에 의존하도록 출력뉴런을 링으로 구성하면된다.

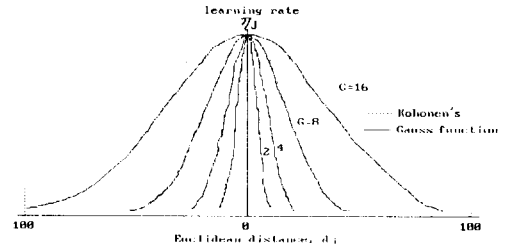


그림 2. 학습률의 변화

Fig. 2. Variation of the learning rate.

2개의 입력 뉴런에서 도시 s 의 좌표 (x_{s1}, x_{s2}) 를 입력 패턴으로, (w_{j1}, w_{j2}) 를 신경회로망 j 뉴런의 연결세기의 값으로 하면, 입력도시에 대한 뉴런 j 의 출력값 V_j 는

$$V_j = |x_{01} - w_{j1}| |x_{02} - w_{j2}| \quad (4)$$

가 되고, 경쟁에서 이긴 승리 뉴런 V_{jc} 는

$$V_j = \min_j V_j \quad (5)$$

으로 결정된다. Kohonen의 학습 알고리즘에서는 임의의 한 뉴런이 승리뉴런이 될 수 있다. 이로 인하여 그의 알고리즘에서는 한 뉴런이 두 도시입력에 대하여 모두다 승리하여 두개 도시를 대표할 수도 있다. 이런 경우에는 학습정도를 나타내는 수렴오차 E 를

$$E = \sum_{j=1}^N |x_{01} - w_{j01}| + |x_{02} - w_{j02}| \quad (6)$$

라 할 때, 수렴 오차 E 가 0이 되지 않는다. 이를 0으로 하기 위하여는 한 뉴런이 한 도시만을 대표하도록 하여야 한다.

Angeniol 등은 이를 해결하기 위하여 한 뉴런이 다른 도시 입력에서도 승리를 하면 새로운 뉴런을 삽입하여 승리를 넘겨 줌으로써 한 개 뉴런이 두 도시에 대하여 승리하지 못하게 하였다. 이렇게 함으로써 한 뉴런은 한 도시 입력에 대해서만 승리하고 수렴 오차도 0으로 수렴한다. 그러나 이 알고리즘을 이용하면 학습중에 뉴런의 복제로 인하여 출력뉴런의 수가 점차 증가함을 알 수 있다. 그리고 몇 번 계속하여 전체 입력도시에 대하여 한번도 승리를 하지 못한 뉴런은 출력뉴런에서 삭제되어 출력뉴런이 더 이상 증가하지 않도록 하였다. 따라서 이 Angeniol 등의 알고리즘에서는 학습중에 최대 kN 개의 출력뉴런이 요구될 뿐만 아니라, 이는 학습중 가변적이어서 사전에 출력뉴런수가 알려져 있지 않고, 뉴런의 삽입과

제거를 위한 별도의 알고리즘이 요구되는 등의 단점을 가진다.

그러나 본 논문에서는 N 개의 출력 뉴런을 각각의 도시에 대표시켜 한 도시를 대표하여 승리한 뉴런은 더 이상 다른 도시를 대표하지 못하도록 함으로써 식 (6)로 주어지는 수렴오차가 0이 되도록 하였다. 즉, 승리뉴런은 그 다음 학습에는 포함되나 경쟁에서는 제외되도록 하였다. 이렇게 하면 학습중에 계속 출력 뉴런수를 N, 시냅스 연결의 수를 2N으로 고정시킬 수 있다. 따라서 제안된 알고리즘에서는 출력뉴런 수와 시냅스 연결수가 적고 학습중에 고정되어 있을 뿐만 아니라 뉴런의 삽입 및 제거를 위한 별도의 알고리즘이 요구되지 않아 학습속도도 개선시킬 수 있다.

이렇게 얻어진 승리뉴런 j 를 중심으로 도시 s 에 대하여 모든 출력뉴런의 연결세기값을 학습률 η 만큼 다음과 같이 학습시킨다. 즉,

$$W_{jk} \leftarrow W_{jk} + \eta_j (X_{sk} - W_{jk}) \tag{7}$$

이다. 여기에서 학습률 η 는 식 (3)과 같다. 그러나 그림 2로 대표되는 학습률 η 를 그대로 사용하면 도시간의 거리가 뉴런간의 거리로 사상되어 도시간의 거리는 다르지만 그 도시가 대표하는 뉴런간의 거리는 서로 같게 될 수 있다. 따라서 학습률은 뉴런간 거리의 함수가 되어 승리뉴런을 중심으로 대칭으로 나타난다. SOFM은 대칭인 뉴런간의 거리에 따라 학습하기 때문에 비대칭인 도시간의 거리에 따른 효과적인 학습을 할 수 없다. 이러한 단점을 보완하려면 학습률에 도시간의 거리가 반영되어야 한다. 따라서 본 논문에서는 비대칭인 도시간의 거리를 반영하고 있는 실제 뉴런의 출력(output potential)에 따라 학습률을 조절할 수 있도록 하였다. 즉, 식 (4)에서 알수 있듯이 각 도시간의 거리는 뉴런의 출력으로 나타나므로 새로운 학습범위 G' 를

$$G' = G(V_j) = G(1 - V_j / 2) \tag{8}$$

라 정의하여 간접적으로 도시간의 거리를 반영시키면 학습률 η 는

$$\eta_j^* = \eta (V_j) = \exp(-d_j^2 / G'^2) / \eta_0 \tag{9}$$

로 되어 실제 도시간 거리가 반영될 수 있다. 이 때 뉴런간의 거리 d_j 는 출력뉴런이 링으로 구성되어 있으므로 다음과 같다.

$$d_j = \min(|j_c - j|, |j_c + (N-j)|, |j_c + (N-j_c)|) \tag{10}$$

따라서 제안된 알고리즘에서는 입력도시의 거리분포에 따라 학습률이 적응적으로 조절된다. 즉, 실제 출력값 V_j 가 작으면 현재 입력도시와 도시간의 거리가 가까워 학습률을 크게하여 많은 학습을 하지만 V_j 가 크면 현재 입력도시와는 도시간의 거리가 멀어 학습률을 작게한다. 이렇게 하면 학습률 η_j^* 는 도시간의 거리에 따라 적응조절되기 때문에 도시분포와 무관하게 좋은 해를 얻을 수 있다.

SOFM 알고리즘 초기에는 연결세기의 값이 자유롭게 변화하여 그림 4(a)와 같이 조그마한 원을 구성한다. 즉, 식 (9)에서 학습범위 G 가 클 때는 학습률 η 는 거리 d_j 에 크게 의존하지 않고, 모든 도시에 같은 확률로 학습되어 아직은 출력뉴런들이 어느 도시도 대표한다고 할 수 없는 상태가 된다. 그러나 점차 학습이 진행되면서 그림 4(c)와 같이 각 뉴런에 해당하는 연결세기의 값은 각각 어느 한 도시의 좌표값에 접근한다. 즉, 도시 s 의 좌표가 입력될 때, 입력좌표는 승리뉴런 주위의 출력뉴런의 연결세기 값에 학습된다. 모든 도시가 한번씩 입력되었으면 일반적으로 학습범위 G 를

$$G \leftarrow G (1 - \alpha) \tag{11}$$

로 감소시켜 수렴되도록 한다. 초기에는 출력뉴런들이 모든 입력도시에 대하여 반응하도록 하지만 점차 학습이 반복됨에 따라 학습범위는 좁아지고 결국에는 승리 뉴런 만이 한 도시 입력에 대하여 적응적으로 반응하여 결국 각 출력 뉴런들은 각각 하나의 도시를 나타내도록 학습된다.

제안된 알고리즘은 승리뉴런을 결정하는 과정이 Angeniol 등의 알고리즘과 달라 N 개의 고정되고 적은 뉴런을 가지고도 대규모 TSP 문제를 효과적으로 해결할 수 있을 뿐만 아니라 식 (10)과 같이 학습률을 실제 뉴런의 출력값에 따라 적응적으로 조절하였다.



그림 3. 반복횟수가 (a)5 (b)40 및 (c)60 일 때의 연결세기의 분포

Fig. 3. Distribution of connection weights after (a)5 (b)40 and (c)60 evolutions.

이상과 같이 대규모 TSP문제의 효과적으로 해결하기 위하여 본 논문에서 제안한 개선된 SOFM 알고리즘은 다음과 같이 정리될 수 있다. 즉,

- 단계 0 : 연결세기 값 w_{ij} , 승리뉴런의 학습률 η_0 , 학습범위 G_0 를 초기화 시킨다.
- 단계 1 : 식 (4)와 (5)로부터 입력 도시 좌표에 대한 출력 뉴런 j 의 유클리드 거리와 승리 뉴런을 구한다. 이 때 한 도시에 승리한 뉴런은 다른 도시와의 경쟁에서 제외한다.
- 단계 2 : 승리뉴런 j 를 중심으로 도시 s 에 대하여 모든 출력뉴런의 연결 세기값을 식 (7)과 같이 학습시킨다.
- 단계 3 : 한 도시 입력에 대하여 학습이 끝난 뒤 다른 도시 입력에 대하여 step 1을 반복한다. 모든 도시가 한번씩 입력되었으면 학습범위 G 를 식 (11)에서와 같이 감소시킨다.
- 단계 4 : 수렴오차한계 ϵ 보다 작아질때까지 위의 단계 1-4를 반복한다.

제안된 알고리즘은 승리 뉴런을 단계 1과 같이 결정함으로써 출력뉴런 수가 적고 학습중 일정하며 그 수렴속도도 빠르다. 또한 단계 2에서와 같이 학습률을 실제 도시간의 거리, 즉, 실제의 뉴런출력값에 따라 적응조절함으로써 도시분포와 관계없이 항상 양질의 해를 얻을 수 있다.

III. 시뮬레이션 결과 및 고찰

0-1사이의 임의의 값으로 발생시킨 30개의 도시 분포에 대하여 제안된 알고리즘을 이용한 TSP 문제를 시뮬레이션 하였다. 표 1은 제안한 방법과 Angeniol의 방법 각각의 필요한 출력 뉴런 수를 나타낸 것이다. 30개 도시 TSP문제에서 제안된 방법에서는 출력 뉴런수가 도시수와 같은 30개로 항상 고정되지만 기존의 방법에서는 각 파라메타에 따라서 소요되는 출력 뉴런 수가 달라질 뿐만 아니라 제거 알고리즘 적용 수 k 가 클수록 그 수는 더욱 더 증가된다.

그리고 표 2와 3에는 각각 G_0 및 η_0 에 따른 해의 성능과 수렴 속도를 나타내었다. 표에서 보는 바와 같이 제안된 방법이 기존 Angeniol 등의 방법보다 더 빠르게 수렴하고 그 해도 더 좋아짐을 알 수 있다.

그림 4와 5에는 제안된 방법과 기존의 방법에 있어

표 1. 30개 도시 TSP에 필요한 출력 뉴런의 수($k=3$)

Table 1. The number of required neurons where $k=3$ and $N=30$.

G_0	η_0	proposed	Angeniol's
10	2	30	69
10	30	30	65
10	60	30	70
30	2	30	75
30	30	30	78
30	60	30	79

표 2. 30개 도시 TSP에서 $\eta_0 = 30$ 일 때 G_0 변화에 따른 시뮬레이션 결과

Table 2. Simulation results for each G_0 where $\eta_0 = 30$ and $N=30$.

G_0	Proposed		Angeniol's	
	evolution	result	evolution	result
10	51	4.67368	59.5	4.88185
30	56	4.64781	63.8	4.87716
50	58	4.64777	68.2	4.87411

표 3. 30개 도시 TSP에서 $G_0=10$ 일 때 η_0 변화에 따른 시뮬레이션 결과

Table 3. Simulation results for each η_0 where $G_0=10$ and $N=30$.

η_0	Proposed		Angeniol's	
	evolution	result	evolution	result
2	20	6.35086	29.7	5.846244
30	51	4.67368	59.5	4.881849
60	69	4.65468	77.4	4.85116

서 반복횟수(evolution)와 경과시간 각각에 대하여 그 수렴 속도를 나타내었다. 처음에는 학습범위 G 가 크고 적응적인 학습으로 인하여 수렴오차가 크지만 학습범위가 좁아짐에 따라 더 빠르게 수렴함을 알 수 있다. 따라서 제안된 방법이 더 적은 뉴런을 사용하고 출력 뉴런의 삽입 및 삭제에 대한 별도의 알고리즘이 없기 때문에 30% 정도 더 빠르게 수렴됨을 알 수 있다.

그림 6은 시간변화에 따른 학습범위를 나타낸 것이

다. 실제 뉴런의 출력을 학습범위에 반영하였기 때문에 뉴런간의 거리에 따라 시간이 지남에 따른 학습률이 비대칭이 되어 적응적으로 학습범위가 좁아져서 효과적인 제어가 가능함을 알 수 있다.

그림 7은 대규모 TSP 과제를 해결할 수 있는 제안된 알고리즘과 Angeniol의 알고리즘, Christofide 알고리즘 및 GA(genetic algorithms)에 대한 해의 결과분포이다. 제안된 알고리즘은 준최적해를 얻을 확률이 92%로 해의 질이 가장 높음을 알 수 있었다.

그림 8은 대규모 과제인 1000개 도시 TSP문제에 대한 시뮬레이션 결과이다. 이것은 $\alpha = 0.01$ 로 하였을 때 IBM PC-486에서 약 20시간이 걸렸고 총거리는 10,592로서 좋은 결과를 얻었다. 또한 처리속도를 높이기 위하여 $\alpha = 0.2$ 로 하였을 때는 1시간 20분만에 12,392에서 13,626사이의 총거리의 해답을 얻을 수 있었다. 따라서 제안한 알고리즘을 이용하면 출력뉴런 및 연결세기의 수가 적고 수렴속도도 개선 시킬 수 있을 뿐만 아니라 최적해의 질도 높일 수 있음을 알 수 있다. 또한 Angeniol의 경우에 비하여 출력뉴런 수가 적고 학습중에 고정적이기 때문에 하드웨어 구현이 용이하고 대규모 과제에도 효과적인 적용을 기대할 수 있다.

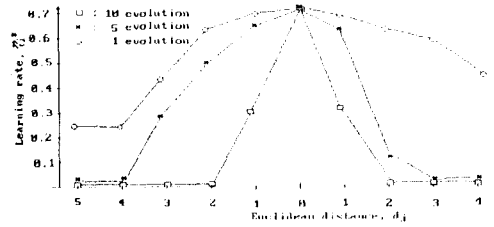


그림 6. 학습범위의 효과적인 제어를 위한 학습률의 변화

Fig. 6. The variation of learning rate during the ordering process.

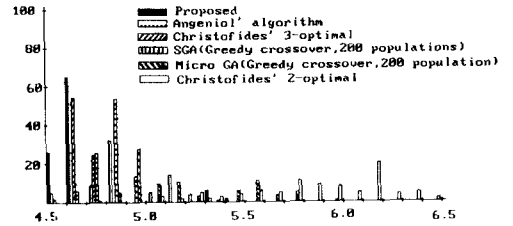


그림 7. 대규모 TSP를 해결하기 위한 각 알고리즘의 해의 분포도

Fig. 7. The histogram of tour lengths when each algorithm is applied for large scale TSP.

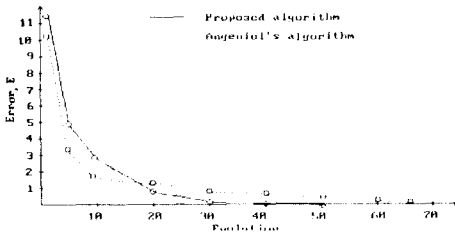


그림 4. 반복횟수에 대한 수렴 속도
Fig. 4. The convergence versus evolutions.

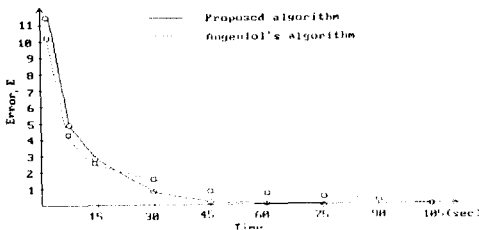


그림 5. 경과시간에 대한 수렴 속도
Fig. 5. The convergence versus elapsed time.

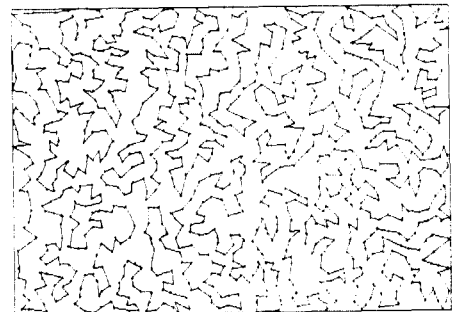


그림 8. 600x300 내의 분포한 1000개 도시 TSP 문제에 대한 $\alpha = 0.01$ 일 때의 시뮬레이션 결과(총길이:10,592)
Fig. 8. Simulation results for 1000-city TSPs. The path length is 10,592 in a 600x300 rectangle for $\alpha = 0.01$.

IV. 결론

본 논문에서는 SOFM을 대규모의 TSP에 적용함에 있어서 경쟁에서 이긴 뉴런은 학습에는 포함시키되 다음 경쟁에서는 제외시킴으로써 N개 도시의 TSP에서 N개의 뉴런과 2N개의 연결세기만 가지고도 효과적으로 최적해를 구할 수 있는 알고리즘을 제안하였다. 30개의 TSP 문제에 대하여 시뮬레이션 해본 결과, 제안된 방법에서는 기존의 방법보다 30%이상의 속도개선을 이루었고 뉴런의 실제 출력값을 직접 학습범위 제어에 적용함으로써 준최적해를 얻을 확률이 총 시도의 92%로서 해의 질도 높였다. 또한 대규모 과제인 1000개 도시 문제에 대하여도 만족할 만한 시뮬레이션 결과를 얻었다. 따라서 제안한 알고리즘을 이용하면 출력뉴런 및 연결세기의 수를 적은 수로 고정시킴으로써 수렴속도를 개선할 수 있을 뿐만 아니라 해의 질도 개선시킬 수 있음을 알 수 있다. 또한 학습중에 출력 뉴런 수가 고정되어 하드웨어 구현이 용이하여 대규모 과제에 더욱 효과적으로 적용할 수 있다.

參考文獻

- [1] D. E. Goldberg, *Genetic Algorithm in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, 1989.
- [2] W. M. Spears and K. A. DeJong, "Using neural networks and genetic algorithms as heuristics for NP-complete problems," *IJCNN*, Washing-

- ton D.C., vol.1, pp. 118-121, Jan. 1990.
- [3] N. Christofides and S. Eilon, "Algorithm for large-scale traveling salesman problems," *Opl. Res. Quart.*, 23, pp. 511-518, 1972.
- [4] J.J. Hofield and D.W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol.5, pp.141-152, 1985.
- [5] B. Angeniol et al, "Self-organizing feature maps and the traveling salesman problem," *IEEE Neural Network*, vol.1, pp.289-293, 1988.
- [6] J. A. Freeman and D. M. Skapura, *Neural networks - algorithms, application and programming techniques*, Addison-Wesley Publishing Company, Massachusetts, 1991.
- [7] T. Kohonen, *Self-organization and associative memory*, Third Edition, Springer-Verlag, Berlin, 1989.
- [8] M. Goldstein, "Self-organizing feature maps for the multiple traveling salesman problem(MTSP)," *INNC* Paris, France, vol.1, pp.258-261, 1990.
- [9] Y. Idan and R. C. Chevallier, "Handwritten digits recognition by a supervised Kohonen-like learning algorithm," *IJCNN*, Singapore, vol. 3, pp. 2576-2581, Nov. 1991.

著者紹介



金 宣 鍾(正會員)

1965年 3月 29日生. 1989年 2月 경북대학교 전자공학과(공학사). 1991年 2月 경북대학교 대학원 전자공학과(공학석사). 1991年-현재 경북대학교 대학원 전자공학과 박사과정중. 주관심분야는 병렬처리, 다중컴퓨터망, 신경회로망 및 신경망 컴퓨터 등임

崔 興 文(正會員) 第 30卷 B編 第 5號 參照

현재 경북대학교 전자공학과 교수