

신경회로망 최적화 기법의 배경 및 응용

이 원돈* 이 석훈**

(*충남대 공대 전산학과 조교수, **통계학과 조교수)

1. 서 론

과학기술의 고속화에 따라 종래에는 인간이 더 잘 수행하던 지능을 요하는 일들이 점점 컴퓨터에 의해 처리되는 경우가 늘어나고 있다. 그러나 아직도 많은 분야에서 종래의 방식에 의한 한 개의 processor나 몇 개의 간단한 multiprocessor 시스템으로는 너무 많은 시간이 걸려서 처리할 수 없는 일들이 산재해 있다. 예를 들어 image를 인식하는 것이나 언어습득과 번역과 같은 문제들에서 인간의 뇌가 현존하고 있는 어떤 컴퓨터의 성능보다 능가하는 것을 알고 있다. 그 이유는 근본적으로 디지털 컴퓨터와 인간의 뇌 사이의 정보 처리 방식이 틀린 점에 기인한다. 대부분의 컴퓨터는 한 개 또는 몇 개의 main processor들이 자료들을 sequential하게 처리하는 데에 반하여, 인간의 뇌는 수천억 개의 neuron들이 정보를 병렬로 처리하고 있다. 따라서, 인간의 뇌에 대한 깊은 연구와 그에 따른 새로운 형태의 정보처리 기술의 개발이 시급히 요구되어지고 있다. 최근에 이와 같은 massive parallelism을 구현하기 위한 시도가 neural network를 중심으로 일어나고 있다. 여기에서 사용되는 algorithm들은 local information에 의하여 자료가 처리되므로 높은 parallelism을 실현한다. Neural network에는 여러가지 model들이 있으나 본고에서는 optimization model을 중심으로 그 배경 및 응용을 소개하고자 한다.

2. 본 론

큰 자유도와 conflicting constraint를 가지는 objective

function에서 global minimum을 찾는 문제는 이 objective function이 일반적으로 많은 local minima를 가질 수 있기 때문에 어려운 문제로 알려져 있다. 종래에는 optimization problem은 iterative improvement와 같은 방법을 많이 사용하였으나 이 방법은 objective function이 한 개 이상의 local minima를 가지고 있을 때에는 초기치의 설정에 따라 특정한 local minimum으로 수렴되기 때문에 일반적으로 global minimum point를 찾지 못하였다. 이런 문제의 해결 방안으로 objective function의 값을 선택하는데 있어서 near-optimal solution이 될 확률이 크도록 선택하는 것이 필요하며 또한 그와 같이 작성되어진 algorithm이 효과적으로 구현 되어질 수 있어야 한다. 지난 몇 년에 걸쳐 위의 두가지 criteria를 만족하는 해결책으로 simulated annealing 방법이 개발되어졌다.

1953년 Metropolis 등은 고체가 에너지 평형 상태로 변해가는 이러한 annealing 과정을 효율적으로 simulation할 수 있는 알고리즘을 발견하였다¹⁾. 그 후에 combinatorial optimization문제의 cost function을 최소화시키는 과정과 고체가 low energy ground state에 도달할 때까지 천천히 식히는 과정이 서로 기본적인 유사성이 존재함과 Metropolis 알고리즘을 적용함으로써 optimization 과정이 실현된다는 것이 Kirkpatrick이나 Cerny 등에 의해 발견되었다²⁾. Metropolis algorithm에서 energy를 cost로 바꾸고 온도의 값을 서서히 내리는 과정을 수행함으로써 Kirkpatrick과 그의 동료는 combinatorial optimization algorithm을 생각해 냈다. 그 이래로 여러 분야 특히 통신, 전산, OR, 전자공학 등의 분야에서 이 이

론의 역할이 증대되고 있다. 여기에서는 신경 회로망의 최적화 기법인 simulated annealing을 소개하기에 앞서, 종래의 combinatorial optimization technique을 살펴보고, 그것의 단점을 simulated annealing이 어떻게 해결하는가를 보고자 한다. Combinatorial optimization 문제를 해결하고자 할 때에는 두 가지의 approach가 있다. 그 하나는 global optimal solution을 찾는 optimal search가 있고 다른 하나는 locally optimal solution으로 만족하는 local search가 있다. 여기에서는 travelling salesman problem(TSP)을 중심으로 그 두 search technique들을 알아보고자 한다.

2.1 Branch and Bound Search : An Optimal Search

그림1과 같이 5개의 도시가 있고, 각각의 도시 사이의 거리가 우리에게 주어졌다고 하자. TSP는 각각의 도시를 단 한번씩 방문하여 가장 짧은 closed tour를 이루는 해를 발견하는 것이다.

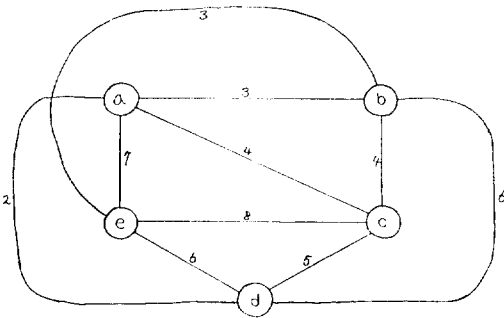


그림1 TSP의 예

Branch-and-Bound search는 tree search로 그 tree의 root는 모든 가능한 tour를 나타내며, 각각의 node는 어떤 특정한 edge를 tour에 포함시키느냐 않느냐에 따라 다시 두 개의 children을 가지게 된다. 따라서 각각의 node는 주어진 TSP의 특정 solution들의 set라고 볼 수 있다. 여기에서, 우리는 각각의 node에 주어진 constraint를 이용하여 그 node가 나타내는 solution들의 lower bound를 계산할 수 있다. 예를 들어 solution들 전체를 나타내는 root에서는 그 lower bound가 17.5임을 알 수 있다.

왜냐하면, 각각의 도시에서 가장 짧은 두 edge를 더한 것을 둘로 나누면, 어떤 tour도 그보다 더 작은 tour cost를 가질 수 없기 때문이다. 이러한 bounding heuristic을 이용하여 우리는 많은 node들을 제거할 수 있게 되어 효과적인 search를 하게 된다. 이와같이 하여, 그림2에서와 같은 tree search를 하게 되고, 우리의 optimal solution이 구해진다. 그러나, 이와같은 Branch-and-Bound method는 주어진 문제의 size가 커지면 매우 많은 computing time을 요구하게 된다.

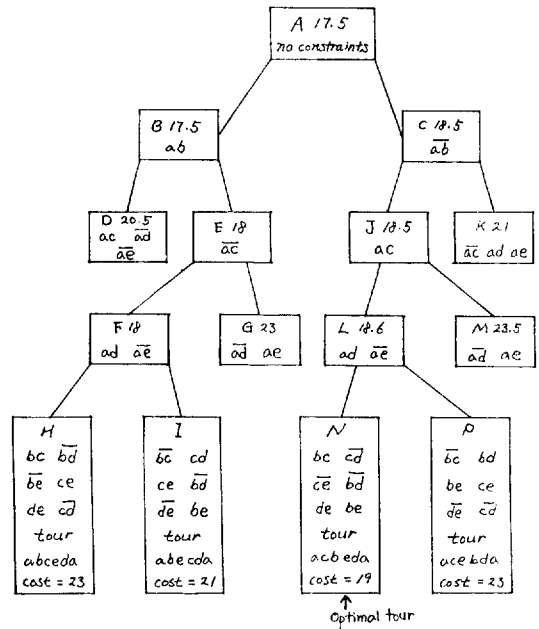


그림2 Branch-and-Bound Search Tree

2.2 2-Opting : A Local Search

Local search algorithm들은 optimal solution을 구하는 것이 exponential time을 요구하는 경우의 heuristic으로 가장 큰 효과를 보아왔다. 일반적으로 이 방법은 solution의 random guess로부터 출발하여 그 현재의 solution을 local transformation에 의하여 perturb하여 locally optimal solution에 이를 때까지 반복하는 것이다. 따라서 이 방법은 perturbation scheme을 쉽게 만들 수 있을 때 적용할 수 있게 된다. TSP에서 쓰이는 perturbation scheme의 하나인 2-opting은 임의의 두 edge

를 잡아서 그것들을 제거한 후, 그들의 end point들을 다시 연결하여 새로운 tour를 만드는 것이다⁴⁾. 그림3에서와 같이 (a)에서 perturbation에 의하여 (b)가 만들어지면 cost의 변화는 $\Delta C = C_f - C_i = 23 - 25 = -2$ 가 된다. 우리의 문제는 cost를 minimize하는 것이므로 우리는 (b)를 새로운 current solution으로 받아들일게 된다. 이 방법을 요약하면 아래와 같다.

Loop :

```

if  $\Delta C \leq 0$  then
    accept the move
else
    take another sample :
until no further improvement.
    
```

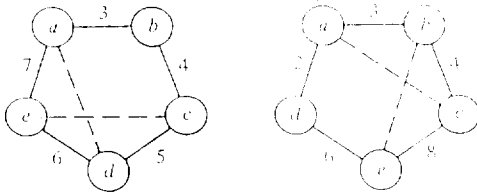


그림3 Local Search using 2-opting

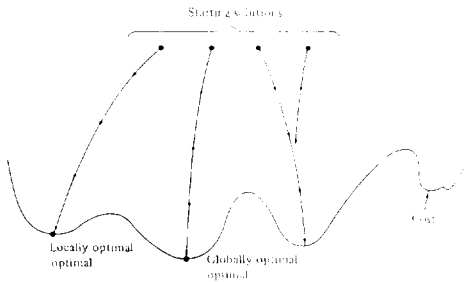


그림4 Local Search의 단점

이와같은 방법은 그 마지막 solution이 solution의 initial guess와 perturbation sequence에 따라 좌우되며, 주어진 sequence에 따라 iteration을 반복하여 어떤 local minimum point에 도착하게 되면 그 local minimum을 탈출할 방법이 없는 것이 가장 큰 문제로 지적되고 있다(그림4).

그러나, 이 방법은 configuration과 cost, function, 그리고 perturbation scheme들이 일반적인 응용에서 쉽게

정의가 될 경우에 광범위하여 이용될 수 있는 장점이 있다.

2.3 Simulated Annealing : A Neural Network Optimization Technique

Annealing이란 고체 물리학에서 고체를 가열하여 고체의 모든 분자가 자유롭게 임의로 배치되어 있는 액체 상태에서 서서히 온도를 내려 냉각시키면 모든 분자가 lattice의 low energy ground state에 가깝게 재배치가 되는 현상을 말한다. 높은 temperature에서 시작하는 annealing과정의 cooling단계는 다음과 같이 기술된다.

각 temperature T에서 고체는 thermal equilibrium에 도달하는데 이 평형은 Boltzmann 분포 $\text{prob}(E=e) = 1/Z(T) * \exp(-E/kT)$ 에 의해 에너지 E의 상태에서의 probability에 의해 특성되어진다. 여기서 Z(T)는 normalization factor인데 온도 T에 의존하는 partition function이다. 그리고 k는 Boltzmann 상수이고 $\exp(-E/kT)$ 은 Boltzmann factor이다. 온도가 감소함에 따라 Boltzmann distribution은 lowest energy를 갖는 상태에 집중되고 마침내 온도가 zero에 접근함에 따라 minimum energy상태만 non-zero probability를 가진다. 만약 cooling 이 너무 빠르면 즉 고체가 각 온도값에서 thermal equilibrium에 도달하는 것이 허용되지 않으면 무정형 고체상태로 응고하여 좋은 결과를 얻지 못한다.

Simulated Annealing은 다음과 같이 요약되어진다.

```

Set  $x_0$  : initial solution
     $T_0$  : initial temperature.
     $T_f$  : final temperature.
     $i \leftarrow 0$ 
    
```

Loop :

- (i) perturb to make a candidate solution x_p :
- (ii) if $C_p \leq C_i$, set, $x_i \leftarrow x_p$
 else $\langle C_p \rangle C_i$
 $\text{prob}(x_i < -x_p)$
 $= \exp[-(C_p - C_i)/T_i]$:
- (iii) decrease T_i :
 set $x_{i+1} \leftarrow x_i$
 $i \leftarrow i + 1$
 until $T_i \leftarrow T_f$.

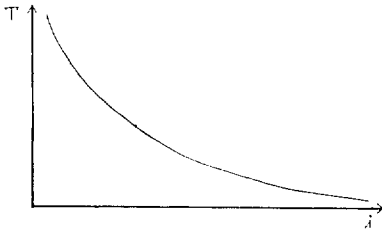


그림5 Annealing Schedule

이와같은 simulated annealing에서는 위의 (i)과정을 행함에 있어서 perturbation을 주는 coordinate를 임의로 선택하여 candidate solution을 형성하게 된다. 여기에서는 iteration이 반복됨에 따라 temperature T를 그림5와 같이 천천히 감소시켜야 하는데 이것을 annealing schedule이라고 한다. Temperature를 너무 급격히 감소시키면 각 temperature에서 equilibrium에 도달하지 못하게 되어 final solution의 quality가 떨어지게 된다. 여러 사람들이 optimal solution을 얻기 위한 엄격한 annealing schedule의 조건들을 제시하였으나^{56,67}, 이러한 엄격한 annealing schedule은 너무 많은 시간을 요구하기 때문에 구현하기가 어렵다. 따라서 보통은 응용 분야에 따라서 quasi-equilibrium에 이르도록 하는 여러 형태의 annealing schedule을 쓰고 있다²⁸⁶. Cost의 변화량 ΔC 는 local information에 의하여 계산될 수 있으므로 simulated annealing은 parallel하게 process할 수 있게 되어 parallel computer로 구현될 수 있어 큰 speed-up을 얻을 수 있게 된다. Candidate solution x_p 를 새로운 current solution으로 받아들일 확률은 ΔC 에 따라 그림6과 같이 나타내어진다. ΔC 가 negative이면 무조건 accept 하게 되므로 그 확률이 1이 되며, C 가 positive인 경우에는 temperature에 따른 Boltzmann factor의 값이 acceptance probability가 된다. T 가 무한대일 때에는 그 확률은 ΔC 에 관계없이 1이 되므로 모든 configuration이 equal probability를 가지는 random move상태가 된다. 그와 반대로 T 가 0에 가까울 때에는 ΔC 가 positive 일때에 그 확률이 0에 접근하므로 acceptance probability function은 step function의 형태를 지니게 된다. 이 때에는 cost가 작아지는 방향으로의 movement만 허용되므로 우리가 이미 본 local search에 해당하는 것임을 알 수 있다. T 가 positive한 값을 가지고 있을 때에는 perturbation에 의한 configuration의 cost가 현재의 solution

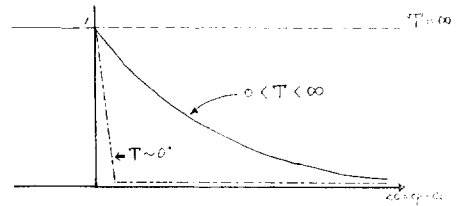


그림6 ΔC 에 따른 Acceptance Probability

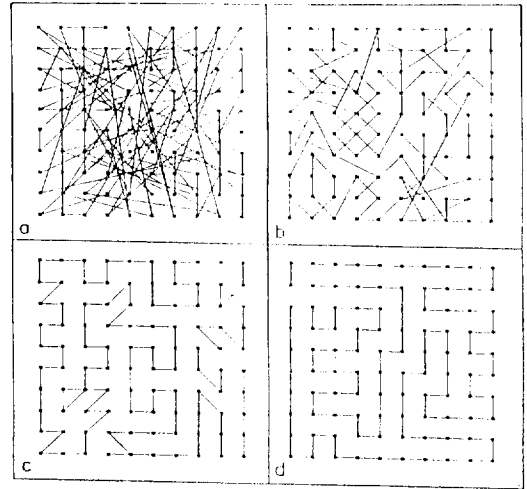


그림7 Simulated Annealing을 이용한 TSP 풀이의 예

의 것보다 높을지라도, 그것을 accept하는 probability를 가지게 되어 local minimum에서 탈출할 수 있는 길을 가지게 된다. 따라서 simulated annealing은 2-opting이 가지고 있는 장점을 살리면서 local minimum에 stuck 되는 단점을 보완한 것이라고 볼 수 있다.

Simulated annealing을 2dimensional grid TSP에 적용하였을 때 temperature가 감소됨에 따라 optimal solution에 접근해감이 그림7에 보여진다.

2.4 Sampling Schemes

Simulated annealing의 배경인 Metropolis algorithm은 원래 statistical mechanical quantity를 계산하기 위하여 제안되었다. Classical statistical system의 molecular configuration x 에 따른 energy $E(x)$ 의 평균값은

$$\langle E \rangle = \int_{x \in \mathcal{X}} E(x) p(x) dx \text{로 표현한다.}$$

로 표현된다.

여기에서 probability density $p(x)$ 는 Boltzmann distribution으로 주어지게 된다.

$$p(x) = 1/Z(T) * \exp[-E(x)/kT]$$

여기에서 partition function $Z(T)$ 는

$$Z(T) = \int_{x \in \mathcal{X}} \exp[-E(x)/kT] dx$$

이와같은 $\langle E \rangle$ 값을 crude하게 측정하기 위해서는 random number를 이용하여 system의 particle들이 uniform하게 distribute되어 random position에 있도록 하면 된다. 그런 후 각각의 configuration에서의 $E(x)$ 와 $p(x)$ 를 측정 한 후 $E(x)p(x)$ 의 곱을 많은 configuration에 대하여 구한 후 평균하면 될 것이다. 그러나, 이와같은 계산은 다음의 두가지 이유로 인하여 불가능하다.

첫째로, probability density $p(x)$ 가 Boltzmann factor $\exp[-E/kT]$ 에 비례하므로 $E(x)$ 에 따라 매우 급격하게 변한다. 따라서, 매우 낮은 energy $E(x)$ 를 가지는 configuration 들만이 significant한 $E(x)p(x)$ 값을 가지게 된다. 다시 말하여, configuration space에서 매우 제한된 일부만이 정말로 중요한 지역이며 나머지 configuration space는 우리의 average계산에 중요하지 않다. 따라서 몇몇의 configuration들만을 sample했을 때에는 알맞은 작은 energy를 가진 configuration들이 매우 적거나 아주 없을 것이다. 이러한 문제는, sampling을 important region에서 집중적으로 하는 이른바 "importance sampling"으로 해결할 수 있게 된다.

위의 예에서, sample configuration들을 uniformly random 한 방법으로 고르지 않고, 구하는 값에 significant한 contribution을 하는 configuration들이 많이 골라질 수 있는 distribution $\pi(x)$ 에 따라 sample하면 된다. Simulated annealing에서 쓰고 있는 distribution $\pi(x)$ 는

$$\pi(x) = w \exp[-E(x)/kT] \text{로 나타내어진다.}$$

로 나타내어진다.

두번째로, 좀더 심각한 문제는 probability density $p(x)$ 를 계산할 수 없다는 것이다. 왜냐하면, configuration space가 방대하여 configuration integral $Z(T)$ 의 값을 모르기 때문이다. 이와같은 문제는 system의 successive sample configuration이 independent sample들이 아니라 Markov chain을 만들면 해결될 수 있다. 이렇게 될 때

에는 successive configuration들의 $p(x)$ 값들의 ratio만을 계산하면 되어 $Z(T)$ 를 계산할 필요가 없게 된다.

다시 말하면 $\pi(x)$ 를 limit distribution으로 갖는 Markov chain을 만들고자 하는데 이를 위하여는 다음의 condition을 만족하는 transition matrix $\|P_{ij}\|$ 를 만들어야 한다.

$$\begin{aligned} p_{ij} &\geq 0 \quad \forall i, j \\ \sum_j p_{ij} &= 1 \quad \forall i \\ \sum_i \pi_i p_{ij} &= \pi_j \end{aligned}$$

위 세번째 condition은 Markov chain이 irreducible 하고 aperiodic할 때 unique limit distribution이 존재한다는 데서 나오게 된다. Perturbation을 주어 next configuration을 generate하는 것에 해당하는 transition matrix가 $\|p_{ij}^*\|$ 일 때, Metropolis가 제안한 sampling Markov chain의 transition matrix element, p_{ij} 는 다음과 같다.

$$p_{ij} = \begin{cases} p_{ij}^* & \pi_j > \pi_i, j \neq i \\ p_{ij}^* \pi_j / \pi_i & \pi_j < \pi_i, i \neq j \end{cases}$$

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}$$

위의 세 condition들을 만족하는 Barker가 제안한 transition probabilities는 아래와 같다.

$$p_{ij} = p_{ij}^* \pi_j / (\pi_i + \pi_j) \quad j \neq i$$

$$p_{ii} = 1 - \sum_{j \neq i} p_{ij}$$

이 Barker의 sampling 방법은 Boltzmann machine과 그의 여러 simulated annealing 응용에서 사용되었다⁹⁾.

여기에서 중요한 점은 각각의 transition probability는 π_i 자체를 계산할 필요가 없이 그 ratio π_j/π_i 와 같은 형태만을 계산하면 된다는 것이다. 따라서, configuration integral을 계산해야하는 문제를 우회할 수 있게 된다.

2.5 응용의 예 1 : A Combinatorial Problem

선박 design에 있어서 equipment들의 초기 배치 문제를 생각해보자¹⁰⁾. n개의 equipment들을 m개의 가능한 location들에 $(m > n)$ 배치할 때의 cost는

$$Cost = \sum_i \sum_j \sum_k \sum_l B(klij) X(ik) X(il) + \sum_i \sum_k A(ik) X(ik) + \sum_i \sum_k X_{ik} = 1, X_{ik} = 1 \text{ or } 0 \quad (k = 1 \dots n; i = 1 \dots m)$$

로 표현되는데 여기에서 $B(klij)$ 는 equipments와 l 을 각각 i 와 j locations들에 놓을 때의 관계 cost이고 $X(ik)$ 는 equipment k 가 location i 에 배치됐을 때엔 1, 그렇지 않을 때엔 0의 값을 가진다. 따라서, 이 문제는 n 개의 distinct balls를 minimal cost를 가지게 되도록 m 개의 box에 넣는 문제와 같게 된다.

이러한 문제에서 $m=20, n=15$ 일 때에 관해 simulated annealing을 적용하였다¹⁴. Perturbation은 현재의 solution에서 임의의 한 equipment를 임의의 빈 location으로 이동하는 것으로 주었다. Temperature는 55,000에서 10,000까지 낮추어주었다. 온도가 낮추어지면서 low cost에 해당하는 configuration이 찾아짐을 그림8에서 볼 수 있다.

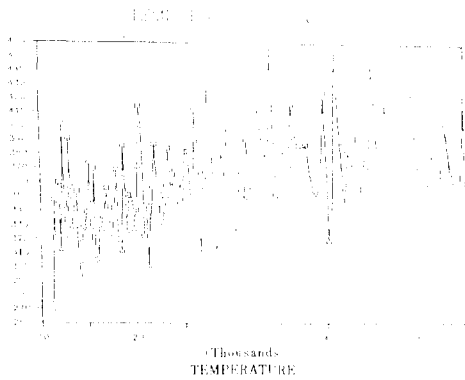


그림8 Shipbuilding Design의 예

2.6 응용의 예 2 : A Continuous Optimization

여기에서는 linear regression model에서 optimal design을 하는 데에 있어서의 응용을 살펴보고자 한다. 이 문제는 모든 n 개의 design point들의 집합 $\{w/w = (x_1, x_2, \dots, x_n) : x_i \in [-1, 1] \text{ for all } i\}$ 위에서 정의되는 real-valued function $\Phi(w) = |M(w)|$ 를 최대로 하는 point를 찾는 것이다¹². 여기에서 design space는 $X = [-1, 1]$ 이며 $M(w) = \sum f(x)f'(x)$ 이고 $f(\cdot)$ 는 design space $[-1, 1]$ 위에서의 vector-valued function이다.

이 design space가 real이므로, 이 문제에 맞는 perturbation scheme이 필요하다. 따라서 perturbation은 현재의 n 개의 design point w_c 의 각각의 individual point $x_i \in w_c$ 를 다음과 같이 옮김으로써 수행된다.

$$x_{it} = x_i + \alpha \zeta_i$$

여기에서는 maximum allowable displacement이고 ζ_i 는 -1과 1사이의 random number이다. 그리고 X 의 boundaries들은 reflecting한다고 생각한다. 여기에서 중요한 점은 annealing을 하는 동안, α 값을 서서히 감소시킨다는 것이다. 왜냐하면 α 는 perturbation scheme에 의해 design point가 restrict 되는 accuracy 정도를 나타내기 때문이다.

$f'(x_i) = [1, x_i, x_i^2, x_i^3, x_i^4, x_i^5]$ 이고 $n=6$ 일 때 각각의 α 와 T 에서 equilibrium에 도달된 후에 $[-1, 1]$ 사이에서 0.1의 interval씩 6개의 design points들의 relative frequencies를 관찰하면 그림9와 같다. Points 1과 -1이 먼저 나타나고, 다음에 ± 0.765 , 그 다음에 ± 0.285 가 나타나는데 이것들은 5th Legendre polynomial의 근으로서 optimal design points들에 해당한다.

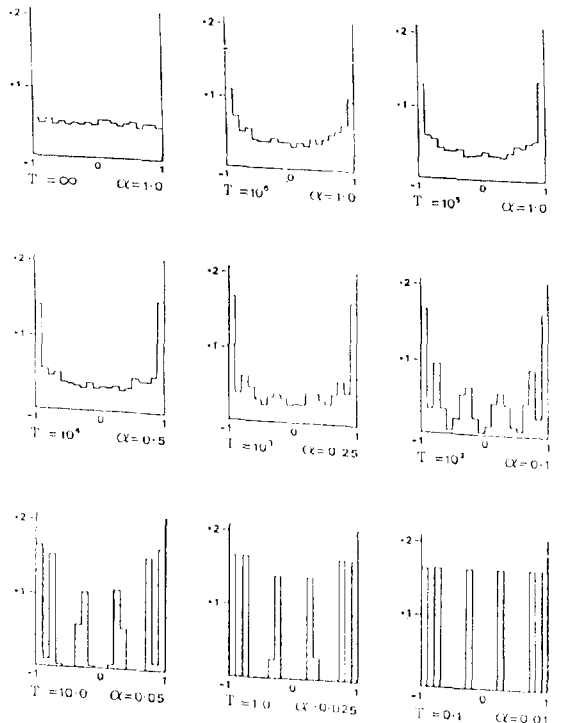


그림9 Histograms Summarizing the System at Equilibrium for Specified Values of α and T

3. 결론

Neural network optimization technique인 simulated

annealing은 기존의 iterative improvement방법의 단점을 Boltzmann distribution에 의한 acceptance probability를 가지고 local minimum에서 탈출할 수 있는 기회를 제공함으로써 그 문제를 해결하였다. 이 방법은 많은 경우의 응용분야에 쓰일 수 있으며 이미 본바와 같이 discrete한 경우와 continuous optimization 문제 등에 응용되었다.

응용분야는 computer-aided circuit design에서 placement¹³⁾, routing¹⁴⁾ 등이 있으며 그 외에도 logic minimization¹⁵⁾, Boltzmann machine¹⁶⁾, numerical analysis¹⁶⁾, biology¹⁷⁾, magnetics¹⁸⁾, code design¹⁹⁾ 등 매우 다양한 분야에서 simulated annealing을 사용하여 문제를 풀고 있다. Simulated annealing은 traditional algorithms들과 많이 비교되었으며 적어도 비슷하거나 더 좋은 결과를 가져온 경우가 많다.

* 이 연구의 일부는 1988년도 문교부 학술연구조성비에 의한 자유공모과제로 선정되어 보조를 받았습니다.

참 고 문 헌

- 1) Metropolis, N., A. Rosenbluth, M.Rosenbluth, A. Teller and E.Teller, Equation of State Calculations by Fast Computing Machines, J. of Chem. Physics, 21, 1953, 1087-1092
- 2) Kirkpatrick, S., C.D. Gelatt Jr. and M.P.Vecchi, Optimization by Simulated Annealing, IBM Research Report RC 9355, 1982.
- 3) Cerny, V., Thermodynamical Approach to the Travelling Salesman problem : An Efficient Simulation Algorithm, J.Opt. Theory Appl. 45, 1985, 41-51.
- 4) Lin, S., and B.W.Kernighan, An Effective Heuristic Algorithm for the Travelling Salesman Problem, Oper. Res., 21, 19730, 498-516.
- 5) Geman, S. and D.Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, IEEE Proc. Pattern Analysis and Machine Intelligence, PAMI-6, 1984, 721-741.
- 6) Anily, S. and A.Federgruen, Probabilistic Analysis of Simulated Annealing Methods, Graduate School of Business, Columbia University, New York, preprint, 1985.

- 7) Mitra, D., Romeo, F. and A.L.Sangiovanni-Vincentelli, Convergence and Finite-Time Behavior of Simulated Annealing, Proc. 24th Conf. on Decision and Control, Ft. Lauderdale, December 1985, pp.761-767.
- 8) Nahar, S., S. Sahni and E.Shragowitz, Experiments with Simulated Annealing, Proc. 22nd Des. Automation Conf., Las Vegas, June 1985, pp.748-752.
- 9) Hinton, G.E., T.J. Sejnowski and D.H. Ackley, Boltzmann Machines : Constraint Satisfaction Networks that Learn, Department of Computer Science, Carnegie- Mellon University, Pittsburgh, Technical Report No.CMO-CS-86-119, May 1984.
- 10) Warczewski, A., Layout and Location Problems - The Quantitative Approach, Building Science, Volumn 7,1972.
- 11) Lee Won Don, Simulated Annealing Applied to Shipbuilding Design, Proceedings of the First International Neural Network Society(INNS) Conference, 1988.
- 12) Haines, L., The Application of the Annealing Algorithm to the Construction of Exact, Optimal Designs for Linear-Regression Models, Technometrics, Vol.29, 4, November, 1987.
- 13) Rothman, S., Circuit Placement by Simulated Annealing at IBM, presented at the Workshop on Statistical Physics in Engineering and Biology, Lech, July 1986.
- 14) Leong, H.W. and C.L.Liu, Permutation Channel Routing, Proc. IEEE Int. Conference on Computer Design, Port Chester, October 1985, pp.579-584.
- 15) Fleisher, H., M.A.Tavel and D.B.Martin, Decomposition of Logic Functions by Simulated Annealing, IBM Corporation, Poughkeepsie, unpublished manuscript, 1984.
- 16) Armstrong, B.A., A Hybrid Algorithm for Reducing Matrix Bandwidth, Int. J. for Num. Methods in Eng., 20, 1984, 1929-1940.
- 17) Lundy, M., Application of the annealing algorithm to combinatorial problems in statistics, Biometrika, 72, 1985, 191-198.

18) Lyberatos, A., P. Wohlfarth and R.W.Chantrell,
Simulated Annealing : An Application in Fine
Particle Magnetism, IEEE Trans. Magnetics, MAG-
21(1985) 1277-1282.

19) Beenker, G.F.M., T.A.C.M. Claasen and P.W.C.
Hermens, Binary Sequences with Maximally Flat
Amplitude Spectrum, Philips J. of Research, 40,
1985, 289-304 : Corrigendum, 40, 1985, 399.