論文98-35C-5-8

이동통신에서의 채널할당 신경망 알고리즘

(A Neural Network Algorithm for the Channel Assignment in Cellular Mobile Communications)

崔 光 皓 *, 李 康 彰 *, 金 竣 漢 *, 田 鈺 峻 **, 趙 鏞 範 *

(Kwang-Ho Choi, Kang-Chang Lee, Jun-Han Kim, Ok-June Jeon, and Yong-Beom Cho)

요 약

본 연구에서는 셀룰라 이동 통신의 효과적인 채널할당을 위한 신경망 알고리즘을 제안한다. 제안된 알고리즘은 대표적인 신경망 모델인 홉필드 네트워크를 근간으로 하여 셀간의 충돌이 발생하지 않게 최소수의 채널로 할당할 수 있도록 최적화 하였으며, 신경망의 장점인 병렬처리가 가능하도록 구현하였다. 알고리즘의 성능을 비교하기 위하여 7개의 벤치마크 문제(benchmark problems)에 적용하였다. 본 알고리즘은 7개의 벤치마크 문제에 적용한 기존의 알고리즘에 비해 약 27%에서 66%까지 수렴시간을 감소시켰으며 수렴율을 크게 개선하였다.

Abstract

This paper proposes a neural network algorithm for a channel assignment in cellular mobile communications. The proposed algorithm is developed base on Hopfield neural network in order to minimize the number of channel without a confliction between cells. To compare the performance of the proposed algorithm, we used seven benchmark problems selected from Kunz's and Funabiki's papers. Experimental results show that the convergence times are reduced from 27% to 66% compared with Kunz's and Funabiki's algorithm and convergence rates are improved to 100%.

Ⅰ. 서 론

셀룰라 이동 통신의 기본원리는 서비스 지역을 여러 개의 육각형 셀로 나누고 서로 인접하지 않은 셀 사이에 주파수를 재사용 함으로써 기존의 통신방법으로는 수용할 수 없었던 많은 통화요구를 충족시키는 것이다. 따라서 제한된 주파수를 가지고 많은 사용자가 사용할 수 있도록 주파수 채널을 할당하는 문제가 매우 중요하다. 초기의 아날로그 셀룰라 이동 통신에서는 셀마다 사용 가능한 채널의 개수를 미리 정해 놓는 고정채널 할당 방법을 사용하였다. 이 방법은 알고리즘 자체가 간단하고 시스템의 구현이 용이하기 때문에 개별 고정 채널할당, set별 고정 채널할당과 같은 알고리즘 개선과 셀을 더 작게 분할하는 방법 등을 통하여 오랫동안 사용되어 왔다. 그러나 급격히 증가하는 통신 요구량을 만족시키기에는 한계를 가지고 있으며 이를 대신하기 위해 call ordering, cell ordering, simulated annealing, channel borrowing, pattern-based op-

* 正會員, 建國大學校 電子工學科
(Dept. of Electronics Engineering, Konkuk University)
** 正會員, 陸軍士官學校 電子工學科
(Dept. of Electronics Engineering, Korea Military Academy)

※ 본 연구는 건국대학교 96년도 학술 진흥 연구비 지원에 의한 결과임.
接受日字:1998年2月4日, 수정완료일:1998年4月16日

timization, local-search, two-phase 등 여러 알고리즘이 제안되었고 channel borrowing과 같은 일부 알고리즘은 실제 시스템에 적용되어 사용되고 있다. 이들 알고리즘은 셀마다 특정개수의 채널을 할당하는 대신 채널 요구량에 맞추어 동적으로 채널을 할당해 준다. 하지만 이런 방법들 역시 국소적인 관점에서만 채널을 할당할 뿐 전체적인 채널 요구량의 분포를 고려하지 않기 때문에 효율적인 채널할당이 어렵다. 일반적으로 주파수 채널할당 문제는 graph-coloring 문제로 모델링[1][2] 될 수 있으며, graph-coloring 문제는 NP-complete(nondeterministic polynomial time complete) 문제로 알려져 있다[3]. NP-complete 문제는 지수적으로 증가하는 계산시간으로 인해 빠른 시간 안에 최적의 해를 찾기가 사실상 불가능하기 때문에 대부분의 알고리즘은 수학적인 방법과 휴리스틱한 방법을 이용하여 해결을 하게 된다. 최근에 채널할당 문제와 같은 NP-complete 문제를 해결하기 위한 방법으로 신경망을 이용하는 연구가 활발히 진행중이다[4-7]. 신경망을 이용하는 경우 전체적인 채널의 요구량을 고려하여 효율적인 할당을 할 수 있고, 병렬처리가 가능하므로 증가하는 채널요구량에 능동적으로 대응할 수 있다는 장점을 가지고 있다. 그러나 지금까지 제시된 신경망 알고리즘은 문제의 해결시간이 너무 길고 채널을 할당하지 못하는 경우가 자주 발생하여 실제의 시스템에 적용하기 어려운 문제점을 가지고 있다. 따라서 신경망 알고리즘의 문제 해결시간을 줄이고 수렴율을 개선시키는 노력이 필요하다.

본 논문에서는 대표적인 신경망 알고리즘인 홉필드 네트워크를 이용하여 이동 통신에서의 채널할당 문제를 효율적으로 처리할 수 있는 신경망 알고리즘을 제안하고, 모의 실험을 통해 그 결과를 기존의 신경망 알고리즘과 비교, 분석해 본다.

2장에서는 이동통신에서의 채널할당 문제를 정의하고 이를 모델링하기 위한 방법을 제시하며 3장에서 채널할당 문제를 해결하기 위한 기존의 신경망 알고리즘을 살펴보고 이를 개선한 새로운 신경망 알고리즘을 제안한다. 4장에서는 벤치마크 문제를 이용하여 제안하는 알고리즘의 성능을 분석하고 5장에서 결론을 내린다.

1. 이동 통신에서의 채널할당 문제

셀룰라 이동 통신(cellular mobile communication)은 전체 서비스 지역을 다수의 무선 기지국(base station)으로 분할하여 소규모의 서비스 지역인 셀(cell)들로 구성하고 이러한 무선 기지국들을 교환시스템으로 집중 제어하여 가입자가 각 셀 사이를 이동하면서도 통화를 계속할 수 있도록 하는 방식이다. 하나의 무선기지국에 의한 서비스 지역을 셀(cell)이라고 하며 이들 셀이 여러개 뭉쳐서 하나의 시스템에 의한 서비스 지역을 형성한다. 시스템의 설계편의상 이론적인 셀의 형태는 6각형의 형태이며 이러한 동일 형태의 셀로서 대상지역이 구성되어 있는 것으로 가정할 수 있다.

채널할당 문제에서 가장 중요한 내용은 주파수의 재사용이다. 같은 주파수의 채널은 특정한 거리 이상 떨어져야만 재사용이 가능하다. 이 거리를 주파수 재사용 거리(frequency reuse distance)라고 하며 일반적으로 채널할당 알고리즘에서는 2셀 혹은 3셀 단위의 간격으로 가정한다. 그림 1에서는 주파수 재사용 거리를 2셀 단위로 가정하였을 때 주파수가 재사용 되는 예를 나타낸다. 그림 1에서 숫자는 각 셀에 할당된 채널을 의미한다.

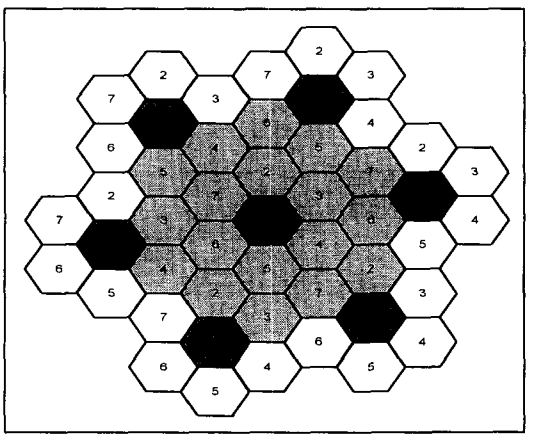


그림 1. 주파수 재사용의 예
Fig. 1. Example of frequence reuse.

채널할당 문제는 Gamst에 의해 정의되었으며[8], 다음에 채널할당 문제에서 사용되는 기본적인 용어를 나타낸다.

1) X = {x1, x2, ... , x_n }
: 셀 시스템으로 각각의 셀들의 유한 집합을 나타낸다.

2) M = (m_i)

: X에 대한 요구 벡터(requirement vector)로 각 셀당 필요한 채널 수를 나타낸다.

3) C = (C_{ij})

: X에 대한 Compatibility 매트릭스 (Compatibility matrix), 즉 셀 #i와 셀 #j간의 떨어져야 할 최소한의 주파수거리를 나타낸다.

4) F = (F_t)

: X에 할당된 주파수들을 나타낸다.

5) A, B

: 각각 이미 할당된 주파수, 차단된(Blocked) 주파수를 나타낸다.

위의 용어를 사용하면 채널할당 문제는 P=(X, M, C, A, B)와 F = (F_t)로 표시할 수 있다. 그리고 다음과 같은 조건들을 만족한다면 채널할당 문제 P에 대해 F는 주파수 할당이 가능하다.

$$\begin{aligned} &1) F_i - m_i && \text{for } all\ i \\ &2) |f - f'| \geq C_{ij} && \text{for } all\ i, j,\ f \in F_i,\ f' \in F_j \\ &3) A_i \subseteq F_i && \text{for } all\ i \\ &4) B_i \cap F_i = 0 && \text{for } all\ i \end{aligned} \quad (1)$$

셀룰라 이동통신에서의 채널할당 최적화문제는 Compatibility 매트릭스 C와 요구벡터 M이 주어졌을 때 전체 사용된 채널 수를 최소화하면서 충돌이 일어나지 않게 채널을 할당하는 것이다. Compatibility 매트릭스 C는 다음과 같은 제약 조건[9]을 이용하여 만들어진다.

1) Cochannel constraint(동일 채널 제약 조건) :
어떤 두 개의 셀에 동시에 같은 주파수를 할당해서는 안된다.

2) Adjacent Channel constraint(인접채널 제약 조건) :
주파수 스펙트럼에서 인접된 주파수들은 인접된 셀에 동시에 할당될 수 없다.

3) Cosite constraint(동일 지역 제약 조건) :
한 셀에 할당될 어떤 주파수 쌍은 주파수 스펙트럼 상에서 일정한 만큼의 거리가 떨어져 있어야 한다.

채널할당 문제의 간단한 예로서 그림 2와 같은 5셀 네트워크를 살펴본다.

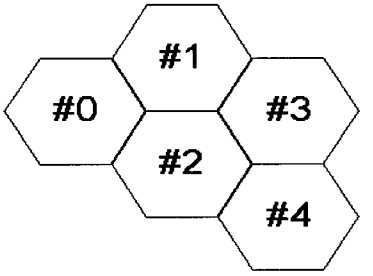


그림 2. 육각형 구조의 5셀 네트워크
Fig. 2. Five-cell network of hexagonal structure.

먼저 채널할당 문제를 수식화하기 위해 그림 2의 셀 형태와 채널할당을 위한 제약 조건으로부터 표 1-a와 같은 Compatibility 매트릭스 C를 만들어 내야 한다. 표 1에서 #0, #1, #2, #3, #4는 셀의 번호를, 표 1-a의 숫자 0, 1, 2는 각 셀간의 떨어져야할 주파수 거리를 의미한다. 따라서 Compatibility 매트릭스 C는 서로 간섭을 일으키는 셀 사이에 할당되는 주파수들은 주파수 거리 1만큼 떨어져 있어야 하며 한 셀에 할당되는 주파수들 사이의 간격은 주파수 거리 2가 되어야 함을 나타낸다. 그리고 표 1-b는 채널 요구 벡터로서 5개의 셀마다 요구된 채널의 수가 각각 2, 1, 1, 2, 3개임을 나타낸다. 마지막으로 사용 가능한 채널의 수는 6개로 제한하였다.

표 1. (a) Compatibility 매트릭스 C, (b) 요구 벡터 D
Table 1. (a) Compatibility matrix C, (b) Demand vector D.

(a)

	#0	#1	#2	#3	#4
#0	2	1	1	0	0
#1	1	2	1	1	0
#2	1	1	2	1	1
#3	0	1	1	2	1
#4	0	0	1	1	2

(b)

#0	#1	#2	#3	#4
2	1	1	2	3

위의 조건을 가지고 적당히 할당하면 그림 3과 같은 결과를 얻을 수 있었다.

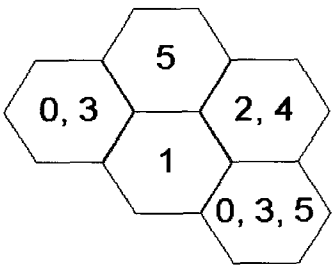
예를 들어 '셀 0'에 할당된 주파수와 같은 주파수를 $C_{01} = C_{10} = 1$ 이기 때문에 '셀 1'에 할당될 수 없고 최소한 주파수 거리 1만큼 떨어져 있어야 한다. 또 '셀 4'에 할당된 임의의 세 주파수는 $C_{44} = 2$이기 때문에 최소한 주파수 거리 2만큼 떨어져 있어야 한다.

따라서 그림 3과 같이 할당을 하게 된다. 위의 결과에서 6개의 이용 가능한 최소의 채널을 가지고 서로 간섭(Interference)을 일으키지 않으면서 재사용(Reuse)한 결과 각각의 셀에서 필요로 하는 총 9개의 채널을 할당할 수 있음을 알 수 있다.

셀의 개수 : 5
할당에 사용할 수 있는 채널 수 : 6
할당된 총 채널 수 : 9

셀 번호	채널 요구량	할당된 채널
0	(2)	: 0 3
1	(1)	: 5
2	(1)	: 1
3	(2)	: 2 4
4	(3)	: 0 3 5

(a)



(b)

그림 3. 5셀 네트워크에서의 채널할당 결과
Fig. 3. Channel assignment result in five-cell network.

Ⅱ. 신경망을 이용한 채널할당

1. 신경망을 이용한 채널할당 알고리즘의 선례

채널할당 문제를 해결하기 위한 신경망 알고리즘은 논문 [4-7] 등에 의해 제안된 바 있다. Kunz[4]는 셀과 채널 각각에 뉴런을 하나씩 대응시켜 '셀수×채널수' 개의 2차원 네트워크를 구성하였다. 각 뉴런은 홉필드와 탱크[10]의 모델을 적용하였으며 채널 요구량과 여러 제약 조건을 에너지의 함수로 나타내어 이를 계산하고 에너지가 최소화하도록 하였다. Kunz는 인접채널에 대한 제약 조건을 고려하지 않았지만 몇 가지의 예를 통해서 성공적으로 채널을 할당하였다. 그러나 Kunz의 모델은 몇 가지 문제점을 가지고 있다. Funabiki[5]는 Kunz의 알고리즘에서 다음과 같은 점을 지적하고 이를 개선하였다. 첫째 Kunz의 모델은 뉴런의 상태방정식에 속도가 느린 시그모이드 함수를 사용하였으며 decay항을 가지고 있다. 홉필드 네트워크에서 decay항은 일부 특정한 상태에서 시스템의 수렴을 방해하는 것으로 알려져 있다[11]. 둘째로 상태방정식에서 주어진 문제마다 다른 계수 값을 사용하여 일반적인 채널할당 문제에 적용하기 힘든 약점을 가지고 있다. 마지막으로 Kunz는 수렴빈도에 대해 언급하지 않았는데, 수렴빈도는 신경망 연구에 있어서 늘 논란의 여지를 가지고 있는 예민한 문제이다. Funabiki는 Kunz의 모델을 개선시켰다. 상태방정식에서 decay항을 제거하고 시그모이드 함수대신에 수렴속도가 빠른 히스테리시스 맥컬리-피츠 함수를 사용하였으며 8개의 벤치마크 문제를 테스트한 결과를 보여주고 있다. 그러나 Funabiki의 알고리즘은 쉬운 문제에서는 비교적 빠른 수렴속도와 높은 수렴빈도를 보이고 있지만 채널 요구량이 커지게 되면 수렴속도와 수렴빈도 양측에서 급격한 성능의 감소를 가져오고 있어 실제 시스템에 적용하기에 어려운 단점을 가지고 있다. 본 논문에서는 Kunz와 Funabiki의 알고리즘을 개선한 새로운 상태방정식을 구현하였다. Funabiki의 상태방정식에서 불필요하게 에너지의 변화를 막고 있는 saturation function을 제거하여 채널요구가 최대한 반영되도록 하고, 채널할당에 필요한 여러 제약조건들을 대부분 수식에 포함하였다. 또한 모의 실험을 통해 상태방정식에 가장 적절한 계수값을 미리 선정하여 채널할당 알고리즘의 모의 실험 과정에서 고정된 계수값을 사용하였다. 제안하는 알고리즘은 모의 실험 과정에서 휴리스틱한 방법을 사용하여 수렴율을 크게 개선시켰으며 7개의 벤치마크 문제에 적용하여 그 결과를 나타내고 있다.

2. 제안하는 신경망 채널할당 알고리즘

본 알고리즘에서는 홉필드 네트워크를 채널할당 문제에 적용시켰다. 홉필드 네트워크를 채널할당 문제에 적용하기 위해서는 문제에서 얻어진 코스트(cost) 함수와 제한 조건들(constraint)을 고려하여 에너지 함수를 만들어야 한다. 그러나 에너지 함수를 직접 만들기는 매우 까다로우며, 해당 문제에 적용할 때는 상태방정식으로 전환해야 한다. 따라서 뉴런의 상태방정식을 먼저 만들고 필요에 따라 이를 적분하여 에너지 함수를 얻는다.

홉필드 네트워크를 이용하여 최적화 문제를 해결하

기 위해서는 수학적 에너지 함수 E를 최소화해야 한다. 에너지 함수를 최소화하기 위해 제1차 오일러 방법을 사용하여 새로운 입력값 U(t+1)을 구하고 이를 뉴런의 입출력 함수에 의해 새로운 뉴런의 상태 V(t+1)을 구한다. 그리고 출력된 전체 시스템의 안정 상태를 조사하여 알고리즘을 종료하던가 다시 입력 변화값을 구하여 되풀이한다.

총 n개의 셀과 m개의 채널에 대한 채널할당 문제는 n × m 개의 뉴런이 필요하게 되고 이를 2차원 배열로 그림 4와 같이 신경망 네트워크를 구성할 수 있다.

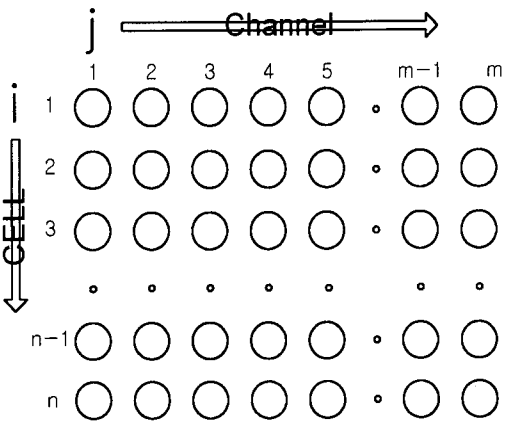


그림 4. 채널할당을 위한 2차원 신경망 배열
Fig. 4. Two dimensional neural network array for a channel assignment.

그림 4에서 i, j번째 뉴런의 출력 V_{ij}는 채널 j가 셀 i에 할당되어 있는지를 나타낸다. 즉 V_{ij} = 1이면 채널 j가 셀 i에 할당되어 있음을 나타내고 V_{ij}=0이면 할당되지 않았음을 나타낸다.

에너지 함수를 만들기 위해서는 2장에서 언급한 채널 요구 제약 조건(channel requirement constraint), 동일지역 제약 조건(cosite constraint), 그리고 동일채널과 인접채널 제약 조건(cochannel and adjacent channel constraint)을 고려하여야 한다. 그러나 에너지 함수를 직접 만들기는 매우 까다로우므로 뉴런의 상태방정식을 먼저 만들고 필요에 따라 이를 적분하여 에너지 함수를 얻는다. 상태방정식은 채널 요구와 여러 제약 조건을 처리하는 항을 포함해야 하고 hill climbing과 같은 보조항이 추가되어야 하는데 대부분 모의 실험과 직관적인 경험에 의해 만들어야 한다. 채널할당 문제를 해결하기 위해 본 논문에서 제안한 상태 방정식은 다음과 같다.

$$\begin{aligned}\frac{dU_{ij}}{dt} = & -A\Big(\sum_{j=1}^{m} V_{ij} - m_i\Big) - B\sum_{j'=1}^{m} V_{ij'}\,\alpha_{ij'} - C\sum_{i'=1}^{n}\sum_{j'=1}^{m} V_{i'j'}\,\beta_{ij'j'} \\ & + Df\Big(\sum_{j'=1}^{m} V_{ij'} - m_i\Big)(1-V_{ij}) - Eg\Big(\sum_{j'=1}^{m} V_{ij'} - m_i\Big)V_{ij} \\ & + Fh\Big(\sum_{j'=1}^{m} V_{ij'}\,\alpha_{ij'} + \sum_{i'=1}^{n}\sum_{j'=1}^{m} V_{i'j'}\,\beta_{ij'j'}\Big)(1-V_{ij})\end{aligned} \tag{2}$$

여기서

$$\alpha_{ij} = \begin{cases} 1 & \text{if } j' \neq j \text{ and } j-(C_{ii}-1) \le j' \le j+(C_{ii}-1) \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{ij'j'} = \begin{cases} 1 & \text{if } i' \neq i \text{ and } c_{ii'} > 0 \text{ and } j-(C_{ii'}-1) \le j' \le j+(C_{ii'}-1) \\ 0 & \text{otherwise} \end{cases}$$

$$f(x) = \begin{cases} 1 & x < 0 \\ 0 & x \ge 0 \end{cases}, \quad g(x) = \begin{cases} 1 & x \ge 0 \\ 0 & x < 0 \end{cases}, \quad h(x) = \begin{cases} 1 & x = 0 \\ 0 & \text{otherwise} \end{cases}$$

여기서 A항은 채널 요구 제약 조건을 만족시키기 위한 항인데, 채널수의 요구를 만족시켜 주기 위해서는 셀 i에 대한 m개의 뉴런 중 총 m_i개의 뉴런이 1의 출력을 가져야 한다. 즉 셀 i의 합이 필요한 채널 수보다 크면 뉴런이 초과 활성화 됐다는 의미이므로 A항의 성분이 음의 값을 가져 V_{ij}의 값을 억제하도록 기여한다. 따라서 셀 i의 합이 채널 수 보다 작을 경우는 아직 필요한 만큼의 채널이 할당되지 않았다는 것을 의미하므로 A항의 성분이 양의 값을 가져 V_{ij}의 값이 활성화 되도록 기여한다. B항은 동일지역 제약 조건을 만족시키기 위한 항인데, 주파수 j로부터 C_{ii} 주파수 거리보다 작게 떨어진 주파수 j'가 셀 i에 이미 할당되어 있는 경우, 주파수 j는 셀 i에 할당되어서는 안된다. 또, C항은 동일채널 제약 조건과 인접채널 제약 조건을 나타낸 것으로 주파수 j로부터 주파수 거리 $C_{ii'}$만큼 떨어진 주파수 j'가 $c_{ii'}>0$이고 $i' \neq i$인 셀 i'에 이미 할당되어 있다면, 주파수 j는 셀 i에 할당되어서는 안된다. 만약 B항과 C항이 0이 아닌 값을 가지면 그것은 조건을 위반하게 되므로 V_{ij}에 음의 값을 증가시켜 V_{ij}가 억제되도록 하는 역할을 한다. 그리고 D항은 V_{ij}=0일 때 f함수 안의 값이 0보다 작으면 아직 할당할 채널이 남아 있음을 뜻하므로 U_{ij}의 값을 증가시켜 V_{ij} = 1이 되도록 뉴런을 활성화시키는 역할을 한다. E항은 V_{ij} = 1일 때 g함수 안의 값이 0보다 크면 U_{ij}의 값을 감소시켜 V_{ij}=0이 되도록 기여하여 뉴런을 억제시키는 역할을 한다. 또 F항은 B항과 C항이 0일 때, 즉 동일채널 제약 조건과 인접채널 제약 조건을 만족시킬 때 V_{ij}를 어느 정도 활성화시킬

수 있게 한다. 이 A, B, C, D항은 셀룰라 이동통신에서의 제약 조건들로부터 만들어 낼 수 있고, E항, D항, F항은 수렴 속도를 향상시키고 지역 최소점에 빠졌을 때 그곳을 빠져 나올 수 있도록 추가한 항이다.

뉴런의 입력은 제1차 오일러(Euler) 방법에 의해 새로운 값으로 갱신되며, 뉴런의 입출력 함수에 따라 갱신된 입력 값을 사용하여 뉴런의 출력상태를 결정한다. 신경망은 병렬처리를 기본으로 한다. 따라서 병렬처리를 하는 신경망 컴퓨터에서 실행하였을 때 가장 좋은 결과를 얻을 수 있다. 그러나 우리가 사용하는 일반적인 컴퓨터는 순차적(sequential)인 처리를 기본으로 하기 때문에 가상으로 병렬처리를 할 수 있는 알고리즘을 구현하여야 한다. 그 순서는 다음과 같다.

표 2. 신경망 구현을 위한 병렬처리 알고리즘

Table 2. Parallel algorithm for neural network.

1) 적당한 계수값을 결정한다.(A, B, C,..등)
2) 각 뉴런의 입력 U_{ij}을 적당히 작은 값으로 랜덤하게 초기화한다.
3) 각 뉴런의 출력 V_{ij}를 사용하는 뉴런 모델의 입출력 특성에 따라 결정한다.
4) 사용되어지는 상태 방정식에 의해 새로 계산되는 각 뉴런의 입력변화치를 계산한다.
5) 제 1차 오일러 방법에 따라 각 뉴런의 새로운 입력 상태를 계산한다.
6) 시스템이 적당한 해답을 찾았는지 조사하여 알고리즘을 정지하거나 아니면 3)번부터 계속한다.

Ⅲ. 모의 실험 결과

알고리즘의 성능을 평가하기 위해 7개의 벤치마크 문제를 선택하였다. 문제1은 [4]에서, 문제2, 3, 4, 5, 6, 7은 [5]에서 선택하였다. [5]에서 사용된 데이터는 채널할당 문제의 알고리즘 성능을 평가하기 위해 벤치마크 문제로서 자주 사용된다[2,5,12].

Kunz는 핀란드 헬싱키의 한 지역을 모델로 하여 25셀을 비규칙적으로 분포시키고 각 셀당 필요한 통화밀도(Traffic density)와 셀 간에 서로 간섭을 일으키는지 여부에 대해 데이터를 미리 조사한 후에 최소한 73개의 채널이 필요함을 분석을 통해서 알아냈다. 표 3은 그가 분석한 25셀에서의 각 셀당 필요한 채널 수와 서로 간섭을 일으키는 셀 수 및 그 셀 번호를 나타낸 것이다. 표 4는 본 알고리즘에 적용할 수 있도록 추출해낸 요구 벡터와 Compatibility 매트릭스를 나타낸다.

표 3. Kunz's Data

Table 3. Kunz's Data.

Cell #	Channel #	Interf #	Interference to Cell #
0	10	11	1,2,4,6,7,8,9,11,12,13,14
1	11	10	0,2,4,67,9,11,12,13,14
2	9	13	0,1,3,4,5,6,7,8,9,0,11,12,13
3	5	11	2,6,7,8,9,10,11,12,22,23,24
4	9	10	0,1,2,9,10,11,12,22,23,24
5	4	5	2,6,7,8,9
6	5	11	0,1,2,3,5,7,8,9,10,11,12
7	7	12	0,1,2,3,5,6,8,9,10,11,12,23
8	4	11	0,2,3,5,6,7,9,10,11,23,24
9	8	17	0.1.2.3.5.6.7.8.10,11,12,13,14,15,21,23
10	8	19	2,3,4,6,7,8,9,12,13,14,15,17,18,19,20,21,22,23,24
11	9	11	0,1,2,3,4,6,7,8,9,12,13
12	20	17	0,1,2,3,4,6,7,9,10,11,13,14,15,16,17,18,19
13	7	14	0,1,2,4,9,10,11,12,14,15,16,17,18,19
14	7	14	0,1,4,9,10,12,13,15,16,17,18,19,20,21
15	6	10	4,9,10,12,13,14,16,17,18,19
16	4	6	12,13,14,15,17,18
17	5	8	10,12,13,14,15,16,18,19
18	5	11	10,12,13,14,15,16,17,19,20,21,22
19	7	10	10,12,13,14,15,17,18,20,21,22
20	6	6	10,14,18,19,21,22
21	4	9	9,10,14,18,19,20,22,23,24
22	5	8	3,10,18,19,20,21,23,24
23	7	8	3,7,8,9,10,21,22,24
24	5	6	3,8,10,21,22,23

표 4. (a) Compatibility 매트릭스 C1, (b) 요구 벡터 D1

Table 4. (a) Compatibility matrix C1, (b) Demand vector D1

(a)

2,	1,	1,	0,	1,	0,	1,	1,	1,	1,	0,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0
1,	2,	1,	0,	1,	0,	1,	1,	0,	1,	0,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0
1,	1,	2,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0
0,	0,	1,	2,	0,	0,	1,	1,	1,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	1,	1
1,	1,	1,	0,	2,	0,	0,	0,	0,	1,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0
0,	0,	1,	0,	0,	2,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0
1,	1,	1,	1,	0,	1,	2,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0
1,	1,	1,	1,	0,	1,	1,	2,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0
1,	0,	1,	1,	0,	1,	1,	1,	2,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	1
1,	1,	1,	1,	1,	1,	1,	1,	1,	2,	0,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0,	1,	0,	1,	0
0,	0,	1,	1,	1,	0,	1,	1,	1,	1,	2,	0,	1,	1,	1,	1,	0,	1,	1,	1,	1,	1,	1,	1,	1
1,	1,	1,	1,	1,	0,	1,	1,	1,	1,	0,	2,	1,	1,	1,	1,	0,	0,	0,	0,	0,	0,	0,	0,	0
1,	1,	1,	1,	1,	0,	1,	1,	0,	1,	1,	1,	2,	1,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0
1,	1,	1,	0,	1,	0,	0,	0,	0,	1,	1,	1,	1,	2,	1,	1,	1,	1,	1,	1,	0,	0,	0,	0,	0
1,	1,	0,	0,	1,	0,	0,	0,	0,	1,	1,	0,	1,	1,	2,	1,	1,	1,	1,	1,	1,	1,	0,	0,	0
0,	0,	0,	0,	1,	0,	0,	0,	0,	1,	1,	0,	1,	1,	1,	2,	1,	1,	1,	1,	1,	0,	0,	0,	0
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	1,	1,	1,	1,	2,	1,	1,	1,	0,	0,	0,	0,	0
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1,	1,	1,	1,	1,	1,	2,	1,	1,	0,	0
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1,	1,	1,	1,	1,	1,	1,	2,	1,	1,	0
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1,	1,	1,	1,	1,	1,	1,	1,	2,	1,	1
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1,	1,	1,	1,	1,	1,	1,	1,	1,	2,	1
0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	1,	0,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	2

(b)

10,11,9,5,9,4,5,7,4,8,8,9,10,7,7,6,4,5,5,7,6,6,4,5,7,5

[5] 에서 선택한 문제 2 ~ 7의 Compatibility 매트릭스와 요구 벡터를 표 5에서 보여주고 있다.

표 5. Funabiki의 예제
(a) Compatiblility 매트릭스 C2 (b) Compatiblility 매트릭스 C3 (c) Compatiblility 매트릭스 C4 (d) 요구 벡터 D2 (e) 요구 벡터 D3

Table 5. Funabiki's examples
(a) Compatiblility matrix C2 (b) Compatiblility matrix C3 (c) Compatiblility matrix C4 (d) demand vector D2 (e) demand vector D3

(a)

5,1,1,0,0,	1,1,1,1,0,	0,0,0,1,1,	1,0,0,0,0,	0
1,5,1,1,0,	0,1,1,1,1,	0,0,0,0,1,	1,1,0,0,0,	0
1,1,5,1,1,	0,0,1,1,1,	1,0,0,0,0,	1,1,1,0,0,	0
0,1,1,5,1,	0,0,0,1,1,	1,1,0,0,0,	0,1,1,0,0,	0
0,0,1,1,5,	0,0,0,0,1,	1,1,0,0,0,	0,0,1,0,0,	0
1,0,0,0,0,	5,1,1,0,0,	0,0,1,1,1,	0,0,0,0,0,	0
1,1,0,0,0,	1,5,1,1,0,	0,0,1,1,1,	1,0,0,1,0,	0
1,1,1,0,0,	1,1,5,1,1,	0,0,0,1,1,	1,1,0,1,1,	0
1,1,1,1,0,	0,1,1,5,1,	1,0,0,0,1,	1,1,1,1,1,	1
0,1,1,1,1,	0,0,1,1,5,	1,1,0,0,0,	1,1,1,0,1,	1
0,0,1,1,1,	0,0,0,1,1,	5,1,0,0,0,	0,1,1,0,0,	1
0,0,0,1,1,	0,0,0,0,1,	1,5,0,0,0,	0,0,1,0,0,	0
0,0,0,0,0,	1,1,0,0,0,	0,0,5,1,1,	0,0,0,0,0,	0
1,0,0,0,0,	1,1,1,0,0,	0,0,1,5,1,	1,0,0,1,0,	0
1,1,0,0,0,	1,1,1,1,0,	0,0,1,1,5,	1,1,0,1,1,	0
1,1,1,0,0,	0,1,1,1,1,	0,0,0,1,1,	5,1,1,1,1,	1
0,1,1,1,0,	0,0,1,1,1,	1,0,0,0,1,	1,5,1,1,1,	1
0,0,1,1,1,	0,0,0,1,1,	1,1,0,0,0,	1,1,5,0,1,	1
0,0,0,0,0,	0,1,1,1,0,	0,0,0,1,1,	1,1,0,5,1,	1
0,0,0,0,0,	0,0,1,1,1,	0,0,0,0,1,	1,1,1,1,5,	1
0,0,0,0,0,	0,0,0,1,1,	1,0,0,0,0,	1,1,1,1,1,	5

(b)

7,2,1,0,0,	1,2,2,1,0,	0,0,0,1,1,	1,0,0,0,0,	0
2,7,2,1,0,	0,1,2,2,1,	0,0,0,0,1,	1,1,0,0,0,	0
1,2,7,2,1,	0,0,1,2,2,	1,0,0,0,0,	1,1,1,0,0,	0
0,1,2,7,2,	0,0,0,1,2,	2,1,0,0,0,	0,1,1,0,0,	0
0,0,1,2,7,	0,0,0,0,1,	2,2,0,0,0,	0,0,1,0,0,	0
1,0,0,0,0,	7,2,1,0,0,	0,0,2,2,1,	0,0,0,0,0,	0
2,1,0,0,0,	2,7,2,1,0,	0,0,1,2,2,	1,0,0,1,0,	0
2,2,1,0,0,	1,2,7,2,1,	0,0,0,1,2,	2,1,0,1,1,	0
1,2,2,1,0,	0,1,2,7,2,	1,0,0,0,1,	2,2,1,1,1,	1
0,1,2,2,1,	0,0,1,2,7,	2,1,0,0,0,	1,2,2,0,1,	1
0,0,1,2,2,	0,0,0,1,2,	7,2,0,0,0,	0,1,2,0,0,	1
0,0,0,1,2,	0,0,0,0,1,	2,7,0,0,0,	0,0,1,0,0,	0
0,0,0,0,0,	2,1,0,0,0,	0,0,7,2,1,	0,0,0,0,0,	0
1,0,0,0,0,	2,2,1,0,0,	0,0,2,7,2,	1,0,0,1,0,	0
1,1,0,0,0,	1,2,2,1,0,	0,0,1,2,7,	2,1,0,2,1,	0
1,1,1,0,0,	0,1,2,2,1,	0,0,0,1,2,	7,2,1,2,2,	1
0,1,1,1,0,	0,0,1,2,2,	1,0,0,0,1,	2,7,2,1,2,	2
0,0,1,1,1,	0,0,0,1,2,	2,1,0,0,0,	1,2,7,0,1,	2
0,0,0,0,0,	0,1,1,1,0,	0,0,0,1,2,	2,1,0,7,2,	1
0,0,0,0,0,	0,0,1,1,1,	0,0,0,0,1,	2,2,1,2,7,	2
0,0,0,0,0,	0,0,0,1,1,	1,0,0,0,0,	1,2,2,1,2,	7

(c)

7,2,1,0,0,	1,2,2,1,0,	0,0,0,1,1,	1,0,0,0,0,	0
2,7,2,1,0,	0,1,2,2,1,	0,0,0,0,1,	1,1,0,0,0,	0
1,2,7,2,1,	0,0,1,2,2,	1,0,0,0,0,	1,1,1,0,0,	0
0,1,2,7,2,	0,0,0,1,2,	2,1,0,0,0,	0,1,1,0,0,	0
0,0,1,2,7,	0,0,0,0,1,	2,2,0,0,0,	0,0,1,0,0,	0
1,0,0,0,0,	7,2,1,0,0,	0,0,2,2,1,	0,0,0,0,0,	0
2,1,0,0,0,	2,7,2,1,0,	0,0,1,2,2,	1,0,0,1,0,	0
2,2,1,0,0,	1,2,7,2,1,	0,0,0,1,2,	2,1,0,1,1,	0
1,2,2,1,0,	0,1,2,7,2,	1,0,0,0,1,	2,2,1,1,1,	1
0,1,2,2,1,	0,0,1,2,7,	2,1,0,0,0,	1,2,2,0,1,	1
0,0,1,2,2,	0,0,0,1,2,	7,2,0,0,0,	0,1,2,0,0,	1
0,0,0,1,2,	0,0,0,0,1,	2,7,0,0,0,	0,0,1,0,0,	0
0,0,0,0,0,	2,1,0,0,0,	0,0,7,2,1,	0,0,0,0,0,	0
1,0,0,0,0,	2,2,1,0,0,	0,0,2,7,2,	1,0,0,1,0,	0
1,1,0,0,0,	1,2,2,1,0,	0,0,1,2,7,	2,1,0,2,1,	0
1,1,1,0,0,	0,1,2,2,1,	0,0,0,1,2,	7,2,1,2,2,	1
0,1,1,1,0,	0,0,1,2,2,	1,0,0,0,1,	2,7,2,1,2,	2
0,0,1,1,1,	0,0,0,1,2,	2,1,0,0,0,	1,2,7,0,1,	2
0,0,0,0,0,	0,1,1,1,0,	0,0,0,1,2,	2,1,0,7,2,	1
0,0,0,0,0,	0,0,1,1,1,	0,0,0,0,1,	2,2,1,2,7,	2
0,0,0,0,0,	0,0,0,1,1,	1,0,0,0,0,	1,2,2,1,2,	7

(d)

8,25, 8, 8, 8,15,18,52,77,28,13,15,31,15,36,57,28, 8,10,13, 8

(e)

5, 5, 5, 8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25

표 6은 위에서 언급한 Compatibility 매트릭스와 요구 벡터를 조합하여 만든 7개의 문제를 보여주고 있다.

표 6. 알고리즘의 성능비교를 위한 벤치마크 문제

Table 6. Benchmark problems for comparison of algorithm.

문제 #	셀의 개수	요구되는 채널 개수	Compatibility Matrix C	Demand Vector D
1	25	73	C1	D1
2	21	381	C2	D2
3	21	533	C3	D2
4	21	533	C4	D2
5	21	221	C2	D3
6	21	309	C3	D3
7	21	309	C4	D3

요구되는 채널의 수가 미리 정의되어 있어야 함을 주의해야 한다. 요구되는 채널의 수란 전체 셀룰라 시스템에서 사용 가능한 채널의 수를 말하는 것으로 [2] 와 [12] 에 의해 채널의 최소 요구량이 정의되어 있다. 만약 사용 가능한 채널의 수가 충분히 많다면 채널할당이 쉬운 것은 자명한 사실이다. 그러나 사용 가능한 채널의 수가 최소 요구량보다 적다면 문제는 해결이 불가능해진다. 따라서 채널할당 알고리즘의 성능을 비교하는 방법으로 필요한 채널의 요구량을 비교하는 방법과 채널을 할당하는데 걸리는 계산시간을 비교하는 방법, 두 가지가 있다[2].

본 논문에서 사용하는 채널 요구량은 [2] 와 [12] 에 의해 최소 요구량임이 증명되어 있고 특히, 문제2 ~ 7의 채널 수는 [8] 에 의해 이론적으로 최소 채널 요구량임이 증명되어 있다. 본 알고리즘에서는 각 문제마다 최소 채널 수가 주어진 상태에서 계산시간을 비교하게 된다.

본 알고리즘에서는 계산시간을 최소화하기 위해 휴리스틱한 방법을 사용하였다. 채널할당 알고리즘의 동작시 채널 요구량이 가장 많은 상위 10%의 요구 벡터를 먼저 할당한 후, 나머지 요구를 할당하는 방법을 선택하였다. 신경망을 이용하여 채널을 할당할 경우, 채널 요구량이 제일 많은 곳에서의 할당시간이 전체 할당시간에서 차지하는 비중이 매우 높음으로 이곳을 먼저 할당해 주면 알고리즘의 수렴속도를 매우 빠르게

증가시킨다. 위의 데이터를 가지고 모의 실험한 결과를 Funabiki의 결과와 비교하였는데, 반복회수는 1000번으로 제한하여 그 이상일 경우에는 수렴에 실패한 것으로 간주하였으며 각 문제마다 100번씩의 모의 실험을 하였다. 이를 표 7에 나타내었다.

표 7. 모의 실험 결과 성능 비교
Table 7. Summary of simulation results.

문제 #	Kunz		Funabiki		제안한 알고리즘	
	평균반복회수	수렴율	평균반복회수	수렴율	평균반복회수	수렴율
1	2450	-	294.0	9%	73.9(216.6)	100%(50%)
2	-	-	147.8	93%	66.89	100%
3	-	-	117.5	100%	76.59	100%
4	-	-	100.3	100%	70.66	100%
5	-	-	234.8	79%	81.28	100%
6	-	-	85.6	100%	60.57	100%
7	-	-	305.6	24%	105.41	100%

모든 경우에 있어서 수렴율이 Funabiki의 결과보다 우수했으며 27%에서 66%까지 평균반복회수를 크게 줄일 수 있었다. 이는 Funabiki가 모의 실험을 위해 만든 상태방정식이 지나치게 복잡하고 에너지 함수를 만들 때 문제의 요구조건을 제대로 반영하지 못했기 때문인 것으로 생각된다. 제안한 알고리즘의 상태방정식의 경우 현재의 뉴런상태와 여러 제약조건에 따라 수렴 속도를 빠르게 하기 위한 항과 지역최소점에 빠졌을 때 빠져 나올 수 있는 항을 포함하고 있으며 1번 반복했을 경우 변화할 수 있는 ΔU값의 최대, 최소 값을 크게 해줌으로서 수렴시간과 수렴율을 향상시킬 수 있었다. 그러나 문제1번의 경우 ΔU값을 크게 해줄 경우에는 평균 반복횟수와 수렴율이 각각 216.6과 50%에 머물렀으나 ΔU 값의 최대, 최소 값을 15%정도로 줄인 경우에는 73.9와 100%로 매우 뛰어난 결과를 보이고 있다. 이는 ΔU 값이 클 경우 문제를 해결하는 속도는 좋아지지만, 일부 문제에서 나타난 것과 같이 해를 놓치는 경우가 있는 것으로 생각된다.

본 논문에서 제안한 방식은 다음과 같은 특징을 가지고 있다. 첫째, 신경망을 이용하여 병렬처리가 가능하다는 점이다. 특히 채널할당 문제와 같이 문제의 크기가 커질 경우 계산량이 크게 증가하는 문제일 경우 병렬처리의 가능성은 실제문제의 적용에 있어 매우 중요한 의의를 가진다. 순차적인 처리를 하는 일반 컴퓨터가 아닌 병렬로 동작할 수 있는 하드웨어 연산기만 개발한다면 문제의 크기에 따라 연산기의 수를 조절하므로 시스템의 확장이 매우 용이해진다. 둘째, 홉필드 네트워크를 이용하여 채널할당 문제를 해결하였지만 항상 전역 최소점에서 해를 구하는 것은 아니다. 이 문제의 경우 조건만 만족한다면, 즉 채널이 모두 다 할당 됐다면 더 좋은 해답이 있더라도 구하지 않는다는 점이다. 이는 전역최소점을 목표로 할 경우 채널이 모두 할당됐음에도 불구하고 계속 채널을 할당해야 하는 불필요함을 제거하여 빠른 채널할당을 유도할 수 있어 실제적인 응용에 적합하다. 셋째, 부분적으로 채널 요구량이 변했을 때 전체적인 채널 재배치를 한다는 점이다. 이는 장점과 단점을 모두 가지고 있다. 장점은 전체적으로 채널을 재배치함으로 보다 부분적인 재배치보다 효율적으로 채널을 할당할 수 있다. 그리고 정적인 채널할당과 동적인 채널할당의 차이가 없으므로 제안한 알고리즘을 그대로 동적 할당 알고리즘으로 사용할 수 있다는 점이다. 단점은 채널이 요구 될 때마다 재배치를 해야 함으로 빠른 연산시간이 필요하여 실제적인 문제에 적용하는데 장애가 될 수 있다.

Ⅳ. 결 론

본 연구에서는 셀룰라 이동 통신의 효과적인 채널할당을 위한 신경망 알고리즘을 제시하였다. 총 n개의 셀과 m개의 사용 가능한 채널을 가지고 최적의 채널할당을 위해 n×m개의 뉴런들을 상호 연결시킨 홉필드 신경망을 이용하였다. 채널할당 문제는 주어진 주파수의 범위 내에서 통화요구를 최대로 만족시켜주기 위해 채널을 할당하는 것인데, 문제의 크기가 커질수록 계산량이 지수적으로 증가하여 일반적인 순차처리 방식의 알고리즘으로는 해결하기 힘든 경향을 가지고 있다. 그러나 이러한 문제에 신경망을 이용할 경우 고도의 병렬 처리 능력으로 빠르게 최적의 해를 찾을 수 있게 된다. 제안한 알고리즘의 성능을 비교하기 위하여 7개의 벤치마크 문제에 적용하여 모의 실험을 하였으며, 동일한 벤치마크 문제에 적용한 기존의 알고리즘에 비해 약 27%에서 66%까지 문제해결을 위한 수렴시간을 감소시켰으며 수렴율을 크게 개선하였다.

참 고 문 헌

[1] W. K. Hale, "Frequency Assignment:

Theory and Application," *Proc. IEEE*, vol. 68, no. 12, pp. 1497-1514, DEC. 1980.

[2] Wei Wang and Craig K. Rushforth, "An Adaptive Local-Search Algorithm for The Channel-Assignment Problem (CAP)," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 3, Aug. 1996

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, LA : Freeman, 1979

[4] D. Kunz, "Channel Assignment for Cellular Radio Using Neural Networks," *IEEE Transactions on Vehicular Technology*, vol. VT-40, pp. 188-193, Feb. 1991.

[5] N. Funabiki and Y. Takefuji, "A Neural Network Parallel Algorithm for Channel Assignments in Cellular Radio Network," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 430-437, Nov. 1992.

[6] P. T. H. Chan, M. Palniswami, and D. Everitt, "Neural Network Based Dynamic Channel Assignment for Cellular Mobile Communication System," *IEEE_Transactions on Vehicular Technology*, vol. VT-43, pp. 279-288, May 1994.

[7] Enrico Del Re, Romano Fantacci, and Luca Ronga, "A Dynamic Channel Allocation Technique Based on Hopfield Neural Networks," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 1, Feb 1996.

[8] A. Gamst, "Some Lower Bounds for A class of Frequency Assignment Problems," *IEEE Transactions on Vehicular Technology*, vol. VT-35, pp. 8-14, Feb. 1986.

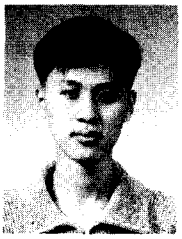
[9] A. Gmast and W. Rave, "On Frequency Assignment in Mobile Automatic Telephone Systems," *Proc GLOBECOM '82* 1982, pp. 309-315.

[10] J. J. Hopfield and D. Tank, "Neural Computation of Decisions in Optimization Problems," *Biological Cybernetics*, vol. 52, pp. 141-152, 1985

[11] Y. Takefuji and K. C. Lee, "Artificial Neural Networks for Four-Coloring Map Problems and K-colorability Problems," *IEEE Transaction Circuit and Systems*, vol. 38, pp. 326-333, Mar 1991.

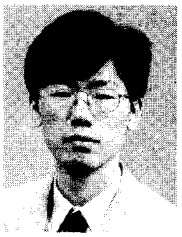
[12] K. S. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel Assignment in Cellular Radio," *Proc. 39th IEEE Vehicular Technology Conf.*, May 1-3, 1989, pp. 846-850.

저 자 소 개



崔 光 皓(正會員)

1997년 건국대학교 전자공학과 공학사. 1997년 ~ 현재 건국대학교 전자공학과 석사과정. 관심분야는 로봇 시스템 설계, ASIC설계, Neural Network

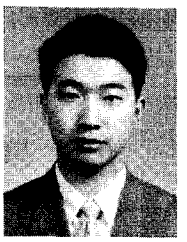


金 竣 漢(正會員)

1994년 경원대학교 전자공학과 공학사. 1996년 건국대학교 전자공학과 공학석사. 1996년 ~ 현재 한국 MJL. 관심분야는 VLSI/CAD Design

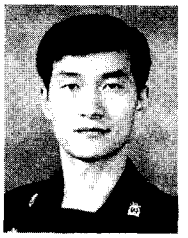
趙 鏞 範(正會員)

1981년 경북대학교 전자공학과 공학사. 1988년 Univ. of South Carolina 공학석사. 1992년 Case Western Reserve University 공학박사. 1992년 ~ 현재 건국대학교 전자공학과 부교수. 관심분야는 Neural Network, VLSI Design



李 康 彰(正會員)

1996년 건국대학교 전자공학과 공학사. 1998년 건국대학교 전자공학과 공학석사. 1998년 ~ 현재 (주)건아기전. 관심분야는 영상신호처리, ASIC 설계



田 鈺 峻(正會員)

1986년 육군사관학교 전자공학과 공학사. 1990년 Case Western Reserve University 공학석사. 1997년 ~ 현재 육군사관학교 전자공학과 강사. 관심분야는 인공지능/신경회로망, C4I 통신체계