

# 유전자와 역전파 알고리즘을 이용한 효율적인 윤곽선 추출

박 찬 란<sup>†</sup> · 이 웅 기<sup>††</sup>

## 요 약

유전자 알고리즘은 염색체 집단을 이용하는 탐색이므로 전역적인 최적해의 탐색 성능은 우수하여 최적해에 근접한 한점까지의 수렴속도는 빠르지만 지역적 탐색 메카니즘이 없기 때문에 최적해 근처의 탐색에서는 수렴 속도가 떨어지는 단점이 있고, 역전파 알고리즘은 개체 수준의 탐색이므로 지역적 미세조정의 탐색능력은 우수하지만 전역적 탐색기능이 없어 지역적 최적해로 수렴하는 경우가 있다.

본 논문에서는 수렴 속도가 향상된 윤곽선 추출을 위하여 유전자와 역전파 알고리즘을 병행해서 실행하는 윤곽선 추출 방법을 제안하였다. 윤곽선 추출 방법은 먼저 유전자 알고리즘을 이용하여 최적의 연결강도와 오프셋 값을 계산한다. 다음으로 이 값을 역전파 학습 알고리즘 학습의 파라미터의 초기값으로 한 반복 학습으로 최적의 윤곽선 구조를 추출하였다. 제안된 알고리즘은 유전자 알고리즘 또는 역전파 알고리즘 단독으로 실행한 경우보다 수렴속도가 향상된 결과를 보여 주었다.

## The Efficient Edge Detection using Genetic Algorithms and Back-Propagation Network

Chan-Lan Park<sup>†</sup> · Woong-Ki Lee<sup>††</sup>

## ABSTRACT

GA has a fast convergence speed in searching the one point around optimal value. But it's convergence time increase in searching the region around optimal value because it has no regional searching mechanism. BP has the tendency to converge the local minimum because it has global searching mechanism. To overcome these problems, a method in which a genetic algorithm and a back propagation are applied in turn is proposed in this paper.

By using a genetic algorithm, we compute optimal synaptic strength and offset value. And then, these values are fed to the input of the back propagation. This proposed method is superior to each above method in improving the convergence speed.

### 1. 서 론

영상에 존재하는 물체의 윤곽선 추출이란 물체와

배경 사이에 위치하는 경계선의 점들을 찾는 작업으로서 일반적으로 입력 영상에 대해 공간 미분을 실시하여 미분 영상을 만들고 그것을 이치화 한 후 다시 세 선화하는 순서로 처리한다. 기존의 윤곽선 추출 알고리즘[1,2,3]의 방법인 이치화 처리는 역치의 선택 방법이기 때문에 추출된 윤곽선의 중간에 끊어진 부분이

<sup>†</sup> 정 회 원 : 목포과학대학 전산정보처리과 교수  
<sup>††</sup> 정 회 원 : 조선대학교 전산통계학과 교수  
논문접수: 1998년 5월 19일, 심사완료: 1998년 9월 23일

발생하여 연결성이 떨어지거나, 세선화 처리에서는 세선화 한 선이 대상 물체의 경계선과 일치하지 않아 추출된 윤곽선이 물체의 정확한 위치를 나타내지 못하는 문제점이 있었다. 위의 문제점을 보완한 연결성이 높고 정확한 위치를 나타내는 효율적인 윤곽선 추출 방법으로서 최적화 알고리즘인 유전자 알고리즘을 이용한 윤곽선 추출 방법이나 역전파 알고리즘을 이용한 형상 인식 방법이 제안되었고[4,5,6], 이 방법의 윤곽선 추출 결과는 연결되고 물체의 정확한 경계선을 나타내기 때문에 우수한 결과를 얻을 수 있었다.

그러나 유전자 알고리즘은 전역적 탐색 알고리즘으로 시스템이 지역적 최소치(local minimum)에 빠지는 위험성이 작고 전역적인 최적해를 발견할 높은 가능성을 갖지만, 역전파 알고리즘의 미세 조정(local fine tuning) 기능인 지역적 탐색 메카니즘이 없기 때문에 최적해 근처에서의 탐색 횟수가 많아지는 원인이 되어 수렴 속도가 급격히 떨어지는 단점이 있다[7]. 또한 역전파 알고리즘의 신경망 학습에서 첫 번째 학습은 학습률과 연결강도, 오프셋 값을 임의로 설정한 값을 가지고 시작하고 목표하는 값과 학습의 출력 값과의 오차를 줄이는 방향으로 연결강도와 오프셋 값을 조정해 가면서 학습을 반복 실행하기 때문에 학습률과 연결강도와 오프셋의 초기값이 값이 얼마나 의미 있는 값으로 정하느냐에 학습시간이 좌우된다. 따라서 역전파 알고리즘이 보다 강력하고 빠른 탐색 성능을 갖기 위해서는 연결강도, 오프셋 값을 처음 학습 단계부터 적합한 값을 가지고서 신경망 학습을 시작하는 방법이 필요하다.

따라서 본 논문에서는 수렴속도가 향상된 윤곽선 추출을 위한 방법으로서 역전파 알고리즘(BP; Back Propagation)과 유전자 알고리즘(GA; Genetic Algorithms)을 이용하여 병행처리 하는 방법으로서 다음과 같이 제안하였다.

첫 번째 유전자 알고리즘을 이용하여 후보 윤곽선 구조의 연결강도, 오프셋 값을 구한다. 그런데 유전자 알고리즘은 전역적인 최적해의 탐색은 우수하지만 지역적 탐색 메카니즘이 없기 때문에 최적해 근처에서 탐색 회수가 많아지는 원인이 되므로 유전자 알고리즘의 탐색 속도를 향상시키기 위하여 인식이 최적해 근처에 이를 때까지 수렴하도록 하였고, 또한 연결되고 정확한 위치를 나타내는 윤곽선 구조를 탐색하기 위하여 유전자 알고리즘이 가지고 있는 연산자 외에

connection 연산자와 one\_pixel 연산자를 추가하였다.

두 번째로 역전파 알고리즘은 유전자 알고리즘에서 탐색한 후보 윤곽선 구조의 연결강도, 오프셋 값을 입력받아 역전파 알고리즘의 기울기 강하 기법(gradient descent method)을 이용하여 최적의 해로 미세 조정하는 학습을 실행하여 윤곽선 구조를 추출한다.

본 논문의 구성은 2장에서는 본 논문의 기본 개념인 유전자와 역전파에 대해 살펴보고, 3장에서는 유전자 알고리즘을 이용하여 후보 윤곽선 구조를 구성한 후 역전파 학습 알고리즘을 이용하여 윤곽선을 추출하는 방법인 시스템의 구성을 기술하였다. 4장에서는 제안된 방법의 실험 결과를 보이고, 끝으로 결론을 논하였다.

## 2. 유전자와 역전파 알고리즘

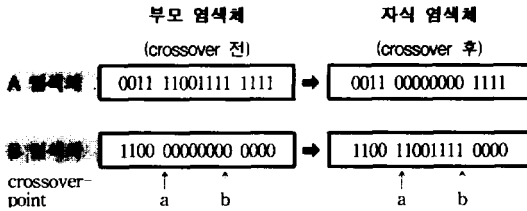
### 2.1 유전자 알고리즘

1975년대 Holland[8]가 제안한 최적화(optimization) 알고리즘의 하나인 유전자 알고리즘(GA, Genetic Algorithm)은 자연 생태계의 적자생존(survival of the fittest)의 원리를 모방한 이론으로서 정의된 함수  $Y = f(x)$ 에서 찾고자 하는 해  $x$ 를 병렬적이고 전역적인 탐색을 통해 찾는 알고리즘[9]이다. 유전자 알고리즘은 평가 함수(evaluation function), 유전 연산자, 그리고 문제의 해를 표현하는 염색체(chromosome)로 구성되며, 기본 연산자로는 선택(selection), 교배(crossover)와 돌연변이(mutation) 등이 있다.

selection은 다음 세대를 구성하는 염색체들을 모집단에서 선택하는 연산자이다. 선택 방법으로는 부모의 적합도(fitness)에 비례하여 높은 적합도 값을 가진 염색체는 다음 세대의 구성에서 선택 될 확률을 높게한 방법인 roulette wheel selection[9]과 각 적합도의 순위에 의해 선택되는 ranking 선택 방법 등이 있다.

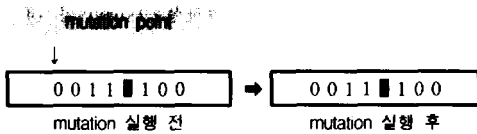
crossover는 두 염색체의 장점을 취하여 더 좋은 염색체를 만들어 주는 연산자로서 그 방법은 선택된 두 부모의 염색체에서 비트 위치를 1개 또는 2개를 임의로 선택하고 그 구간 사이의 염색체 비트들을 서로 교환한다. 이는 새로운 자손을 생성해 주는 연산자로서 탐색 문제의 관점에서 보면 제한된 공간에서 탐색 공간을 넓혀주는 역할을 한다. 다음의 (그림 1)은 선택된 두 부모 염색체의 두 구간 a, b 사이에 있는 비트들을

서로 교환함으로써 새로운 염색체를 생성해 주는 crossover의 실행 예를 나타낸 것이다.



(그림 1) crossover 연산  
(Fig. 1) Crossover operation

mutation은 임의로 선택된 위치의 비트 값이 0이면 1로 또는 1이면 0으로 대입되는 비트로 바꾸어 주는 연산자로서 집단에 존재하지 않는 새로운 해의 요소를 만들어 영역을 효과적으로 탐색하게 하고 국부적인 최소화 상태에서부터 벗어나게 해주는 연산자이다. 다음 (그림 2)는 mutation의 실행 예를 나타낸 것으로 임의로 선택된 5번째 비트에서 1이 0으로 바뀌게 된다.

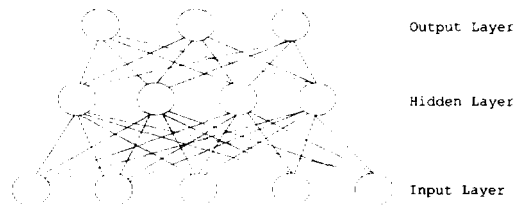


(그림 2) mutation 연산  
(Fig. 2) Mutation operation

## 2.2 역전파 알고리즘

1986년 David. E. Rumelhart에 의해 제안된 역전파 알고리즘은 교사학습(supervised learning) 방법에 의해 학습 자료들이 주어지면 원하는 값과 실제로 출력된 값 사이의 차이를 줄여나가는 방법이다. 학습 규칙은 학습을 진행하는 다계층 전방향(multilayer feed-forward)의 네트워크로서 일반화된 delta rule의 범칙으로 학습을 진행하며, 특히 계층 구조를 갖는 신경망의 학습에 이용되고 있다.

전형적인 역전파 학습 모델은 (그림 3)과 같이 입력층(input layer)과 출력층(output layer) 그리고 한 개 이상의 은닉층(hidden layer)으로 구성되며, 각 노드의 출력 값은 같은 노드로 되돌아오지 않는 비순환 구조로 되어 있다.



(그림 3) 역전파 알고리즘의 신경회로망 모델  
(Fig. 3) Neural network model of back propagation algorithm

역전파 알고리즘의 학습은 크게 두 단계로 나누어진다. 첫 단계는 입력층에 입력패턴을 제시하면 각 유닛에서는 망 입력 함수와 활성화 함수를 이용하여 계산된 값들이 은닉층을 거쳐 출력층에서 값을 출력하는 전방향 단계이고, 두 번째 단계는 요구되는 출력 값과 실제 출력 값과의 오차를 계산해서 이 오차 값을 출력층에서 입력층으로 역전파 시키면서 오차를 최소화 시키는 방향으로 각 노드들간의 연결 강도를 조절하는 후방향 단계이다. 이 두 단계는 시스템이 안정될 때까지 즉, 오류함수가 오류 기준치를 넘지 않을 때까지 계속 반복 실행에 의해 최적의 해를 탐색하게 된다. 역전파 학습 알고리즘의 단계는 다음과 같다.

단계 1 : 네트워크의 상태를 결정해 주는 입력층 I에서 은닉층 J로의 연결강도인  $W_{ij}$ 와 은닉층 J에서 출력층 K로의 연결강도  $W_{jk}$ 와 오프셋(offsets)  $\theta_j, \theta_k$ 를 작은 값의 임의의 수로 초기화한다.

단계 2 : 최초의 학습 패턴을 설정한다.

단계 3 : 학습 패턴인 입력 값  $I_i$ 를 입력층 유닛 I에 입력하여 출력된 값  $O_i$ 와 유닛 I에서 은닉층 유닛 J로의 연결 강도  $W_{ij}$ 와 은닉층 유닛 J의 오프셋  $\theta_j$ 를 가지고 식(2-1)을 이용하여 은닉층 유닛 J의 입력 값  $I_j$ 를 구하고, 식 (2-2)를 이용하여 유닛 J의 입력  $I_j$ 와 출력을 규격화 해주는 시그모이드 함수  $f$ 를 이용해 유닛 J의 출력  $O_j$ 를 구한다.

$$I_j = \sum W_{ij} \cdot O_i + \theta_j \quad (2-1)$$

$$O_j = f(I_j) \quad (2-2)$$

단계 4 : 단계 3에서의 은닉층 유니트  $J$ 의 출력  $O_j$ 와 은닉층에서 출력층으로의 연결강도  $W_{kj}$ , 출력층 유니트  $K$ 의 옵셋  $\theta_k$ 를 이용해 출력층 유니트  $K$ 의 입력  $I_k$ 를 식 (2-3)을 이용하여 구하고, 입력  $I_k$ 와 시그모이드 함수  $f$ 를 이용하여 출력층 유니트  $K$ 의 출력  $O_k$  값을 식 (2-4)를 이용하여 구한다.

$$I_k = \sum_j W_{kj} \cdot O_j + \theta_k \quad (2-3)$$

$$O_k = f(I_k) \quad (2-4)$$

단계 5 : 학습패턴의 목표 출력 값  $T_k$ 와 출력층의 출력  $O_k$ 와의 오차를 가지고서 은닉층  $J$ 에서 출력층 유니트  $K$ 로 연결되는 연결 강도와 출력층 유니트  $K$ 의 오프셋에 대한 오차  $\delta_k$ 를 다음의 식 (2-5)를 이용해 구한다.

$$\delta_k = (T_k - O_k) \cdot O_k \cdot (1 - O_k) \quad (2-5)$$

단계 6 : 단계 5에서 구한 오차  $\delta_k$ 와 은닉층에서 출력층의 연결강도  $W_{kj}$ 와 은닉층의 출력  $O_j$ 로부터 식 (2-6)을 이용하여 은닉층 유니트  $J$ 에 연결된 연결강도와 은닉층 유니트의 오프셋에 대한 오차  $\delta_j$  값을 구한다.

$$\delta_j = \sum_k \delta_k \cdot W_{kj} \cdot O_j \cdot (1 - O_j) \quad (2-6)$$

단계 7 : 단계 5에서 구한 출력층 유니트  $K$ 에서의 오차  $\delta_k$ 와 은닉층 유니트  $J$ 의 출력  $O_j$ 와 정수 상수  $\alpha$ 를 가지고 식 (2-7)을 이용하여 은닉층  $J$ 에서 출력층  $K$ 에 연결된 연결강도  $W_{kj}$  값을 수정하고, 오차  $\delta_k$ 와 정수상수  $\beta$ 를 가지고 식 (2-8)을 이용하여 출력층 유니트  $K$ 의 오프셋  $\theta_k$ 를 수정한다.

$$W_{kj} = W_{kj} + \alpha \cdot \delta_k \cdot O_j \quad (2-7)$$

$$\theta_k = \theta_k + \beta \cdot \delta_k \quad (2-8)$$

단계 8 : 단계 6에서 구한 은닉층 유니트  $J$ 의 오차  $\delta_j$ 와 입력층 유니트  $I$ 의 출력  $O_i$ , 정수 상수  $\alpha$ 를 가지고 식 (2-9)를 이용하여 연결강도  $W_{ji}$  값을 수정한다. 또 식 (2-10)을 이용해 은닉층 유니트의 오프셋  $\theta_j$ 를 수정한다.

$$W_{ji} = W_{ji} + \alpha \cdot \delta_j \cdot O_i \quad (2-9)$$

$$\theta_j = \theta_j + \beta \cdot \delta_j \quad (2-10)$$

단계 9 : 상기의 패턴을 학습패턴으로 한다.

단계 10 : 학습패턴이 종료할 때까지 단계 3으로 되돌아가 반복한다.

단계 11 : 학습의 반복 횟수를 갱신한다.

단계 12 : 학습이 반복 횟수가 제한 횟수보다 작으면 단계 2로 되돌아가 반복한다.

단계 13 : 종료

### 3. 윤곽선 추출을 위한 시스템 구성

#### 3.1 시스템의 전체적인 구성

최적의 윤곽선 구조를 추출하기 위한 단계는 먼저 유전자 알고리즘을 이용하여 후보 윤곽선 구조를 탐색하고 그 후보 윤곽선 구조를 이용하여 적합한 연결강도, 오프셋 값을 구한다. 이때 연결되고 정확한 위치를 나타내주는 윤곽선 구조를 탐색하기 위하여 유전자 알고리즘의 기본 연산자 외에 윤곽선 경로를 연결된 윤곽선 구조로 만들어 주는 connection 연산자와 윤곽선 구조가 중첩된 두꺼운 선으로 되어 있는 경우에는 최적의 위치에 있는 픽셀 1개만 윤곽선으로 만들어 주는 one\_pixel 연산자를 추가하였다.

다음 단계로 유전자 알고리즘의 탐색 결과인 후보 윤곽선 구조에 적합한 연결강도, 오프셋 값을 역전파 학습 알고리즘의 입력으로 하고 기울기 강하 기법의 알고리즘으로 오차를 줄여나가는 학습을 실행하여 최적의 윤곽선 구조를 추출한다.

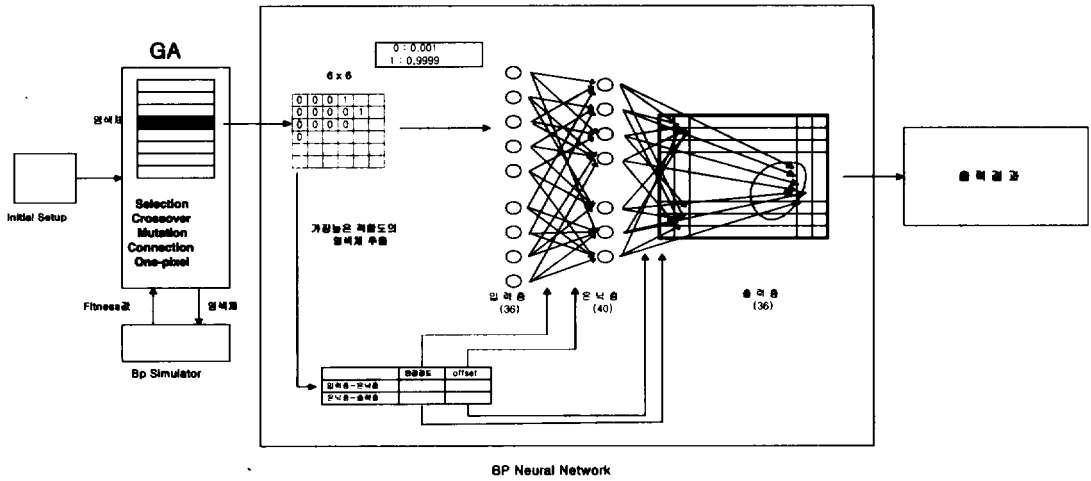
다음의 (그림 4)는 제안한 방법의 윤곽선 추출을 위한 시스템의 전체적인 구성을 나타낸 것이고, (그림 5)는 제안한 방법의 윤곽선 추출 과정을 나타낸 것이다.

단계 1 : Initial setup

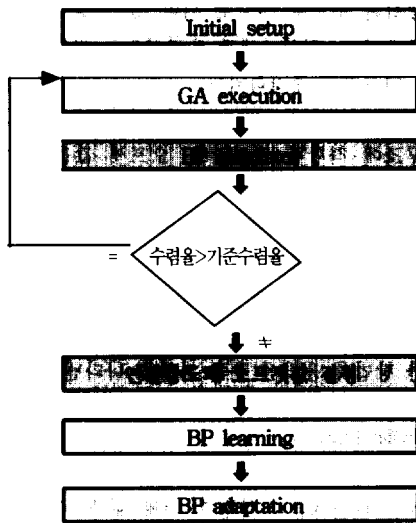
유전자 알고리즘의 selection, crossover, mutation, connection, one\_pixel 연산자의 확률 값과 역전파 알고리즘 학습에 관련된 초기 값을 입력하는 단계이다. 정보를 입력하지 않으면 내정값(default value)으로 설정된다.

단계 2 : GA execution

유전자 알고리즘의 실행으로 최적의 연결강도와 오프셋 값을 탐색하는 단계이다. 그런데 전역적 탐색 알고리즘인 유전자 알고리즘은 지역적 탐색 메커니즘이 없어 최적해 근처의 탐색에서는 수렴 속도가 급격히 떨어지는 원인이 되므로 탐색 속도를 향상시키기 위하여 GA-BP 방법을 적용하였다. 먼저 GA로 탐색한 각



(그림 4) 윤곽선 검출을 위한 시스템 구성  
(Fig. 4) System formation for edge detection



(그림 5) 제안된 방법의 윤곽선 추출 과정  
(Fig. 5) Procedure of edge detection in proposed method

염색체의 적합도 값을 BP simulator를 이용하여 정해진 cycle 동안 학습시킨 후 다음의 (3)절에서 정의한 평가함수를 이용하여 최적의 해를 윤곽선 구조를 평가한 적합도 값을 계산하여 GA로 넘겨주면, GA는 이 적합도 값에 따라서 selection을 실행하고, crossover, mutation 연산자를 이용하여 최적의 해를 탐색하는 과정을 반복한다. 이때 BP simulator는 학습에서 전체

자승 오차가 가장 작은 염색체의 연결강도와 오프셋 값을 write하여 다음 학습의 연결강도와 오프셋 값의 초기값으로 연결하여 이용한다.

이 과정을 기준 수렴율(예, 60%, 70%, 80%)에 도달할 때 까지 반복하고, 실행이 종료되면 가장 높은 적합도의 연결강도, 오프셋 값을 단계 5의 BP Neural Network의 파라미터로 넘겨주어 최적해로 수렴시키기 위한 지역적 탐색을 진행하도록 하였다.

단계 3 : 연결강도와 오프셋 값 결정

앞에서 기술한 본문 2.2의 일반적인 역전파 알고리즘의 첫 번째 학습은 연결강도, 오프셋 값의 초기 값을 임의값으로 설정하여 실행하므로 전역적 최적해와의 오차가 크게 되어 학습시간이 오래 걸리는 원인이 된다. 따라서 초기 학습에 적합한 연결강도와 오프셋 값이 결정되는 것이 필요하다. 본 연구에서는 단계 2에서 탐색하면서 계속 write한 최적의 연결강도와 오프셋 값을 BP Neural Network 학습에서의 연결강도와 오프셋의 초기값으로 하였다.

단계 4 : BP Neural Network

GA의 탐색 결과인 후보 윤곽선 구조의 연결강도, 오프셋 값을 입력받아 최적의 해로 수렴시키기 위한 학습을 실행하여 최적의 윤곽선 구조를 완성시키는 단계이다. 학습을 위한 신경망의 구성은 입력층의 노드 수는 유전자 알고리즘의 탐색 결과인 후보 윤곽선 구조인 36개와 은닉층은 40개, 출력층은 36개의 노드로

구성하였다.

단계 5 : BP adaptation

완성된 신경망의 학습 결과를 테스트하는 단계이다.

3.2 유전자 알고리즘을 이용한 유곽선 구조 탐색

(1) 염색체(chromosome)의 구성

유전자 알고리즘을 이용하여 최적의 후보 유곽선 구조를 탐색하기 위한 염색체는 유곽선 구조 관계를 행렬로 구성하고 유곽선 픽셀은 1로하고, 유곽선이 아닌 픽셀은 0인 2진수의 값을 가지며 유전자 알고리즘을 적용시킬 직접적인 대상이 되는 연결구조 정보이다. 염색체의 크기는 메모리를 고려하여 100×100의 전체 영상에서 6×6 행렬을 세그먼트로 하였기 때문에 염색체 구조는 36개 비트인 유곽선 연결 구조와 각 염색체가 유곽선으로서 어느정도 만족하느냐를 평가함수에 의해 수치로 계산된 적합도 값으로 구성되었으며 다음의 (그림 6)과 같다.



(그림 6) 염색체의 구성  
(Fig. 6) chromosome formation

(2) 유전자 알고리즘 연산자

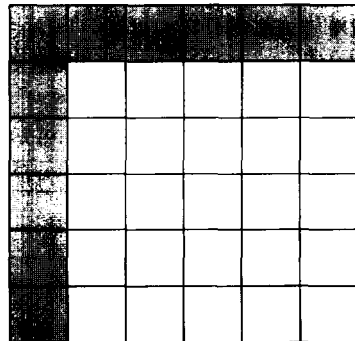
① 기본 연산자

비트 열로 구성된 염색체들은 연산자의 실행으로 최적의 유곽선 구조를 탐색한다. 본 논문에서 사용한 모집단(population) 수는 30개로 하였고 selection 연산자는 적합도의 비율에 따라 개체를 선택하는 roulette wheel selection 방법을 사용하였고, 확률에 따른 선택이므로 가장 좋은 개체가 선택에서 제외되는 경우도 있으므로 엘리트 보존 전략 알고리즘[10]을 사용하여 가장 좋은 적합도를 갖는 개체는 다음 세대에서 무조건 선택되도록 하였다.

Crossover 연산자는 2점 교배 (2-point crossover) 방법을 사용하였고, 또한 crossover와 mutation 연산자도 가장 좋은 적합도를 갖는 염색체가 crossover나 mutation 실행에 의해 변화되는 것을 막기 위하여 엘리트 보존 전략 알고리즘을 사용하였다. 또한 crossover와 mutation 확률은 각각 0.4로 하였다.

② Connection 연산자

연결되어 있는 유곽선 경로 사이에 유곽선이 아닌 즉, 0인 픽셀이 발견되면 그 픽셀을 1로 변환하여 연결된 유곽선 구조를 만들어 주는 연산자이다. connection 실행 과정은 1의 값을 가지면서 같은 방향의 기울기를 갖는 유곽선 경로 사이에 0인 픽셀이 발견되면 1로 수정해서 연결된 유곽선을 만들어 준다. 그런데 connection 연산자는 이웃 픽셀들과의 연결 관계 정보를 가지고서 평가하므로 세그먼트의 크기가 6×6 행렬이기 때문에 다음의 (그림 7)에서와 같이 빗금 친 부분인 위 부분과 좌측에 위치한 픽셀들은 자신의 위쪽의 이웃 픽셀들과 좌측의 이웃 픽셀들과는 연결관계를 비교할 수 없기 때문에 연결 관계 정보를 얻을 수 없다. 이를 해결하는 방법으로서 바로 앞 세그먼트의 결과인 유곽선 구조에서 최 우측과 하단의 12개 픽셀의 값을 배열에 보관해 놓고 현재의 위 부분과 최좌측 픽셀들의 연결 관계는 앞 세그먼트의 유곽선 구조의 값과 비교 평가한다.



(그림 7) 6×6 세그먼트에서 위쪽과 좌측에 위치한 픽셀  
(Fig. 7) Pixels located at left-upper side of 6×6 segment

connection 기능이 일어날 확률은  $P_c$ 에 의해 조절하도록 하였는데, 연결성을 높이기 위해  $P_c$ 는 0.8로 하였다. 알고리즘은 (그림 8)에 기술하였다. 여기에서 유전자 알고리즘의 염색체는  $g[2][6][6]$  ( $x = 0, 1, 2, \dots, 5, y = 0, 1, 2, \dots, 5$ ) 배열 값이고 중심이 되는  $x, y$  위치의 픽셀이 0인 경우  $x, y$  픽셀의 위와 아래에 위치한 픽셀 모두가 유곽선이거나,  $x, y$ 의 왼쪽과 오른쪽 픽셀 모두가 유곽선 인 경우, 또는  $x, y$ 의 대각선에 위치한 픽셀이 유곽선에 해당되면 중심픽셀  $x, y$ 는 유곽선 픽셀인 1로 수정한다.

```

for(x=0; x<6; x++)
{
  for(y=0; y<6; y++)
  {
    if(p < P0)

    {
      if(g2[x][y] == 0)
      {
        if((g2[x-1][y]==1 && g2[x+1][y]==1) || /* x, y 위치한 픽셀의 위와 아래의 픽셀이 윤곽선인 경우 */
            (g2[x][y-1]==1 && g2[x][y+1]==1) || /* x, y 위치한 픽셀의 좌우의 픽셀이 윤곽선인 경우 */
            (g2[x-1][y-1]==1 && g2[x+1][y+1]==1) || /* x, y 위치한 픽셀의 대각선에 위치한
            (g2[x-1][y+1]==1 && g2[x+1][y-1]==1) ) 픽셀이 윤곽선인 경우 */
            g2[x][y] = 1; 위의 경우에 해당되면 윤곽선 픽셀로 한다.
      }
    }
  }
}

```

(그림 8) Connection 연산자 알고리즘  
(Fig. 8) Connection operator algorithm

③ One\_pixel 연산자

기존의 윤곽선 추출 방법 중에서 sobel 방법은 윤곽선이 중첩된 여러 개의 점으로 추출되므로 실제 물체의 크기보다 더 크게 나타난다. 따라서 실제 영상의 물체 크기를 정확하게 나타내기 위한 윤곽선의 조건은 물체의 실제 경계에 해당하는 위치에 1개의 픽셀로 표시하는 것이다. one\_pixel 연산자는 윤곽선 구조가 여러 개의 픽셀로 중첩되어 있다면 그 중에서 가장 명암의 차가 큰 픽셀만 윤곽선으로 하고 그 이웃의 픽셀은

0으로 해서 1개의 선으로 구성된 윤곽선 구조를 만들어 주는 연산자이다. 중첩된 윤곽선 구조의 평가는 중심인 윤곽선 픽셀의 이웃 픽셀이 윤곽선인면서 윤곽선 방향을 의미하는 기울기 값은 중심 픽셀과 같지 않은 경우이다.

따라서 중첩된 윤곽선 픽셀이 발견되면 중심 픽셀과 명암의 차를 비교하여 가장 명암의 차가 큰 픽셀을 1로 하고 이웃의 픽셀은 0으로 한다. 이에 대한 알고리즘은 (그림 9)와 같다.

```

for(x=0; x<6; x++)
for(Y=0; Y<6; Y+=)
{
  if (g2[x][y] == 1)
  {
    gradi = GRA(x,y); /* 중심인 x,y 위치 픽셀의 윤곽선 방향인 기울기 */
                    /* 값을 GRA함수를 이용해 구한다. */
    if(GRA(x-1,y-1) != gradi && g2[x-1][y-1]==1) /* x-1,y-1 위치의 픽셀이 */
    {
      /* 중첩된 픽셀인지 check */
      if(BRI(x,y) > BRI(x-1,y-1)) /* x,y 픽셀과 x-1,y-1 위치 픽셀의 명암의 차를 비교한다.*/
        g2[x-1][y-1] = 0; /* 명암의 차가 더 작은 픽셀은 0으로 한다. */
      else
        g2[x][y] = 0;
    }
  }
  else
  {
    if(GRA(x-1,y) != gradi && g2[x-1][y]==1) /* x-1,y 위치의 픽셀이 */
    {
      /* 중첩된 픽셀인지 check */
      if(BRI(x,y)>BRI(x-1,y)) /* x,y픽셀과 x-1,y 픽셀의 명암의 차를 비교한다.*/
        g2[x-1][y-1] = 0; /* 명암의 차가 더 작은 픽셀은 0으로 한다. */
      else
        g2[x][y] = 0;
    }
  }
}

```

(그림 9) One\_pixel 연산자 알고리즘  
(Fig. 9) One\_pixel operator algorithm

(3) 평가함수(fitness) 정의

평가함수는 염색체가 목표하는 윤곽선 구조에 어느 정도 만족하는지를 수치로 산정해주는 수식으로 목표하는 윤곽선 구조의 데이터는 원 영상의 디지털 값에서 다음 식 (3-1)을 이용하여 명암의 차를 구한다. 명암의 차가 임계값보다 크면 1로 임계값보다 작으면 0으로 구성된 이치화 데이터로 구성하였다.

$$e = \sqrt{\nabla I(x, y)^2} \tag{3-1}$$

x, y 위치의 픽셀에 대한  $\nabla I$ 는 다음 식 (3-2)와 같고  $\nabla I$ 는 영상의 일차 미분 값이다.

$$\nabla I(x, y) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{3-2}$$

각 염색체의 적합도(fitness) 값을 계산하기 위한 평가 함수는 원 영상의 2치화 데이터인 g1 배열의 비트 값과 유전자 알고리즘의 염색체인 g2 배열 비트와 각각 비교하여 비트 값이 일치하고 중심 픽셀과 기울기가 같은 방향인 이웃 픽셀도 윤곽선 픽셀로 되어 있으면 목표하는 윤곽선 구조에 접근하는 구조이므로 높은 값을 부여한다. 이에 대한 알고리즘은 (그림 10)과 같다. 여기서 GRA 함수는 x, y 위치의 픽셀이 윤곽선인 경우 윤곽선 방향인 기울기를 구해주는 함수로서 x, y 픽셀과 같은 기울기를 갖는 다음 방향의 픽셀 위치 값을 gradx와 grady에 주고, ga와 gb는 항목의 비중을 따라 유연성 있게 부여하는 가중치 값이다.

```

if(g1[x][y] == g2[x][y])
/* 원 영상의 2치화 데이터와 염색체의 x, y
  위치의 비트 값이 서로 일치하는지 비교 */
  fit1 = 1;
else
  fit1 = 0;
if(g2[x][y] == 1)
{
  gradx = GRA(x,y);
  grady = GRA(x,y);
  if(g2[gradx][grady] == 1)
/* x, y 위치의 비트가 윤곽선이면 자신과 연결되어
  있는 픽셀도 윤곽선 인지를 평가하는 부분 */
  fit2 = 1;
else
  fit2 = 0;
}
fitness += (ga * fit1) + (gb * fit2);
}
    
```

(그림 10) 평가 함수 알고리즘  
(Fig. 10) Evaluation funtion algorithm

3.3 BP Neural Network 학습 알고리즘

유전자 알고리즘의 실행은 탐색 속도를 향상시키기 위하여 정해진 인식을 만큼만 진행하도록 하였기 때문에 유전자 알고리즘의 결과인 후보 윤곽선 구조는 최적의 해로 수렴되지 못한 상태이다. 따라서 최적의 윤곽선 구조로 수렴시키기 위하여 역전파 학습 알고리즘으로 일정 주기 동안 학습시켜 최적의 윤곽선 구조를 추출한다.

역전파 알고리즘의 학습은 구하고자 하는 이상적인 출력과 실제 출력과의 오차를 최소화하기 위한 방법으로서 기울기 강하 기법<sup>[11]</sup>을 사용하였다.

$$\begin{aligned}
 e &= \sqrt{(|t_p - o_p|)^2} \\
 t_p &= \sum_{i=0}^n \sum_{j=0}^n g1[x][y].w[x][y] \\
 o_p &= \sum_{i=0}^n \sum_{j=0}^n g2[x][y].w[x][y]
 \end{aligned} \tag{3-3}$$

위의 식 (3-3)은 p번째 학습의 출력 값의 오차 e를 구하는 식으로  $t_p$ 는 목표하는 올바른 출력 값이고,  $o_p$ 는 역전파 학습 알고리즘의 출력 값이다. 여기서 g1 배열은 목표하는 윤곽선 구조의 비트 값이고, g2 배열은 학습에 의한 출력 값인 윤곽선 구조이다. w 배열은 다음의 <표 1>에 정의된 행렬의 비트 위치를 의미하는 위치 값 ( $w[x][y] = 2^n$  ( $x = 0, 1, 2, \dots, 5, y = 0, 1, 2, \dots, 5, n = 0, 1, \dots, n$ ))이다.

<표 1> 윤곽선 구조에서 행렬 w[x][y]의 위치 값  
(Table 1) Position value of matrix w[x][y] in edge formation

1	2	4	8	16	32
64	128	256	512	1024	2048
4096	8192	16384	32768	65536	131072
:	:	:	:	:	:

역전파 학습 알고리즘은 오차 e를 최소화하여 구하고자 하는 이상적인 윤곽선 구조로 최적화 하기 위한 알고리즘으로 오차 e 값이 최소 오차 한계 값 (t)보다 작을 때 까지 오차를 줄여나가는 학습 방법이다. 목표하는 윤곽선 구조 ( $t_p$ ) 값이 학습 결과 값 ( $o_p$ ) 보다 더 큰 경우는  $o_p$  배열의 윤곽선 구조가 앞 부분에 구성되어 있다는 것이므로 1의 비트를 적합한 위치의 뒤 부분으로 이동시켜야 목표하는 윤곽선 구조에 접근하게 되므로 오차 값에 해당하는 위치의 g2 배열의 0을



1로 하고, 반대로 0, 값이 더 큰 경우에는 1의 위치를 앞으로 이동시켜야 하므로 오차 값에 해당하는 위치의 g2 배열의 1을 0으로 한 다음 여기에서 발생하는 오차에 해당하는 앞부분의 비트는 1로 하면서 오차를 최소화 한다. 이에 대한 알고리즘은 다음의 (그림 11)에 나타내었다.

```

while(1)
{
오차 = abs(목표값-학습결과값);
if(오차 < 최소 오차 한계값)
break;
else {
if(목표값 > 학습결과값)
flag = 0;
else
flag = 1;
if(현위치값 == 오차 && 현 학습비트값 == flag) {
현재 학습비트를 반대로 바꿔준다;
goto ss;
}
else {
if(현 위치값 > 오차) {
열위치 값 -1 해서 1의 비트를 왼쪽으로 이동;
if (현 위치값 == flag)
현 학습 비트를 반대로 바꿔준다;
goto ss;
}
else {
열위치 값 +1 해서 1의 비트를 오른쪽으로 이동;
if (현 위치값 == flag)
현 학습 비트를 반대로 바꿔준다;
goto ss;
}
}
}
}
}
    
```

(그림 11) 후보 신경망의 역전파 학습 알고리즘  
(Fig. 11) Candidate neural network back-propagation learning algorithm

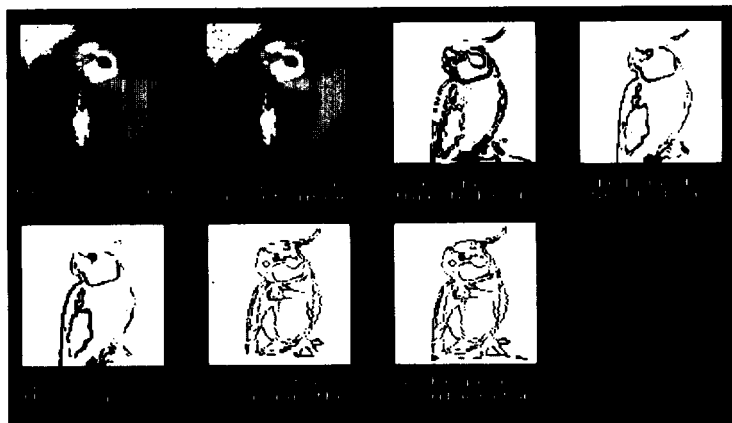
#### 4. 실험 및 결과

##### 4.1 윤곽선 영상 추출 결과

본 논문의 윤곽선 추출 실험은 IBM Pentium 기종으로 C 언어를 사용하였고, 사용한 영상의 크기는 100×100이며, 그레이 레벨이 0 ~ 255인 흑백 영상을 사용하였다. 또한 검색체의 크기는 메모리를 고려하여 100×100의 영상에서 6×6 세그먼트를 검색체로 하여 한 세그먼트씩 학습하고 학습의 전체 평균 오차가 임의의 오차 한계보다 작으면 학습을 끝나치고, 다음 영상으로 진행하여 마지막 세그먼트까지 완료하게 되면 윤곽선 추출 영상이 완성되게 된다.

본 실험이 사용한 영상은 다음의 (그림 12) bird 영상과 (그림 13)의 lena 영상 2가지이고, 기존의 윤곽선 추출 방법인 sobel, robert, kirsh 방법의 윤곽선 추출 결과와 유전자 알고리즘 단독으로 추출한 결과, 제안된 방법인 유전자와 역전파 알고리즘을 병행으로 실행한 윤곽선 추출 영상의 결과를 각각 비교하였다. 또한 제안된 방법의 효율적인 윤곽선 추출 결과를 검증하고 잡음 제거 효과를 확인하기 위해 원영상에 가우시안 방법으로 잡음을  $\sigma = 30$ 으로 첨부하였다.

다음의 (그림 12), (그림 13)에서 (c)의 영상은 sobel 기법을 이용한 결과로 잡음은 많이 제거되었지만 윤곽선이 두꺼운 선으로 추출되어 실제 영상의 물체보다 약간 크게 윤곽선이 나타났으며, 윤곽선이 끊어진 부분도 많이 발생하는 결과를 보였다.



(그림 12) (a) 원영상 (b) 잡음( $\sigma = 30$ )이 첨부된 영상 (c) sobel 방법 (d) Robert 방법  
(e) Kirsh 방법 (f) 유전자 알고리즘 방법 (g) 제안된 방법  
(Fig. 12) (a) Original image (b) Image with noise( $\sigma = 30$ ) (c) Sobel method (d) Robert method  
(e) Kirsh method (f) genetic algorithm method (g) Proposed method



(그림 13) (a) 원영상 (b) 잡음( $\sigma = 30$ )이 첨부된 영상 (c) sobel 방법  
 (d) Robert 방법 (e) Kirsh 방법 (f) 유전자 알고리즘 방법 (g) 제안된 방법  
 (Fig. 13) (a) Original image (b) Image with noise( $\sigma = 30$ ) (c) Sobel method  
 (d) Robert method (e) Kirsh method (f) genetic algorithm method (g) Proposed method

(d)와 (e)의 robert와 kirsh 방법의 결과도 윤곽선이 두껍게 추출되었으며, 또한 실체는 윤곽선 픽셀이지만 연결되어 있지 않은 독립된 픽셀이 잡음으로 평가되어 제거되기 때문에 끊어진 부분이 많이 나타났다.

또한 유전자 알고리즘 단독으로 추출한 결과인 그림(12). 그림(13)의 (f)와 제안한 방법의 결과인 (g)와의 윤곽선 추출 결과를 비교해 보면 두 방법 모두가 유전자 알고리즘에서 connection 연산자와 one\_pixel 연산자를 이용한 탐색 방법이므로 모두 연결성의 강도를 가지며, 실제 물체의 경계와 일치하는 윤곽선 결과로 두 방법의 윤곽선 추출 영상은 차이가 거의 없는 비슷한 결과를 나타내었다. 그러나 다음의 4.2 절에서 기술한 수렴시간의 비교에서 유전자 알고리즘의 단독 실행과 제안한 알고리즘과의 CPU 실행시간의 비교에서는 제안한 방법이 유전자 알고리즘 단독 실행보다 빠른 수렴시간의 결과를 나타내어 탐색시간의 성능에 효과가 있음을 알 수 있었다.

#### 4.2 수렴시간의 비교 실험 결과

윤곽선 추출에 대한 제안한 방법의 향상된 수렴속도의 성능을 검증하기 위하여 다음의 4가지 방법의 수렴율과 CPU 실행 시간을 비교하였다.

첫 번째는 유전자 알고리즘 단독으로 실행한 수렴시간의 결과이고, 두 번째는 역전파 알고리즘 단독으로 실행한 결과이고, 세 번째로는 유전자와 역전파 알고리즘을 병행으로 실행하였지만 유전자 알고리즘으로 후보 윤곽선 구조만 구하고 BP 학습에서의 연결강도와 오프셋 값은 임의의 값으로 설정하여 학습한 결과이고, 네 번째로는 제안한 방법인 유전자와 역전파 알고리즘을 병행으로 실행한 경우로 후보 유전자 알고리즘의 탐색결과인 후보 윤곽선 구조와 연결강도, 오프셋 값 모두를 역전파 학습의 입력으로 한 경우의 수렴시간을 실험해 보았다.

실험 결과에 대한 표는 실험 영상으로 사용한 bird 영상(㉓ 영상이라 지칭함)과 lena(㉔ 영상) 영상 2가지의 결과로 작성하였다.

유전자 알고리즘(GA) 단독 실행한 경우의 수렴 결과는 <표 6> <표 8>에 나타내었고 역전파 알고리즘(BP)의 단독 학습 결과는 <표 2> <표 3>과 같다. <표 4> <표 5>는 GA에서 탐색한 가장 높은 적합도 값을 가진 후보 윤곽선 구조만 BP의 입력층 유니트의 입력 값으로 하고 BP 학습에서 연결강도와 오프셋 값은 임의의 값으로 설정하여 학습한 결과로서, GA 실행으로 해에 접근한 후보 윤곽선 구조를 BP가 입력층 유니트의 입력으로 하였지만 BP 알고리즘의 학습은

입력층의 입력값과 임의로 설정된 연결강도와 오프셋 값과의 계산에 의해 출력 값이 산정되므로 BP 단독 실행과 별 차이가 없음을 보여주었다. 따라서 역전파 알고리즘 학습 시간의 성능을 향상시키는데는 연결강도와 오프셋 값의 초기 값이 적합한 값으로 설정되어야 효과가 있음을 알 수 있었다.

<표 6> <표 7>과 <표 8> <표 9>는 본 논문이 제안한 두 알고리즘을 병행한 방법으로서 먼저 GA에서 해에 근접한 운곽선 구조와 연결강도, 오프셋 값을 탐색한 후 이 모든 값을 BP Neural Network 첫 번째 학습의 파라미터로 하여 학습을 실행한 수렴 결과로서 GA 실행에서 각각의 수렴율에 따른 결과를 BP의 입력으로 하여 학습하였을 때 전체시간에 대한 성능을 비교하였다.

<표 6> <표 7>을 보면 GA 실행 결과인 후보 운곽선 구조와 연결강도, 오프셋을 BP의 초기 값으로 입력받아 학습을 하는 시간대별 수렴시간을 비교해보면, 93% 수렴도를 기준으로 하였을 때 GA 수렴율이 0.8일 때의 (탐색시간 42초) 후보 운곽선 구조와 연결강도, 오프셋 값을 BP의 파라미터의 초기값으로 정하여 학습(93% 수렴의 학습 시간 10초)한 결과인 운곽선 추출의 전체 소요 시간은 52초이고, BP 단독으로 운곽선을 추출한 시간인 54초와 비슷한 탐색시간을 나타내어 별 효과가 나타나지 않았다. 또한 0.5, 0.6, 0.7, 0.8의 적합도의 경우 초기 10, 50회 학습에서는 수렴율의 차이가 나타났지만 학습 횟수가 100회 이상이 진행됨에 따라 별다른 차이는 보이지 않았다.

그러나 99.5%의 수렴을 하기 위해서는 GA에서 50% 수렴한 결과인 후보 운곽선 구조와 연결강도, 오프셋 값을 BP의 파라미터로 하여 병행 실행한 경우에는 전체시간이 4분 12초(7초+4분5초)의 탐색시간이 걸렸는데 BP 단독 실행 결과는 약 7분이고, GA 단독 실행 결과는 6분 이상이 소요되어 총 수렴시간을 비교해보면 제안한 방법이 수렴 속도의 성능에서 더 우수함을 알 수 있었다.

다음의 (그림 14)의 그래프를 가지고 GA의 수렴율에 따른 수렴시간을 비교해보면, GA의 수렴율이 0.6의 경우에는 전체시간이 4분17초(12초+4분5초)가 소요되었고, 0.8의 경우의 전체 시간은 4분 47초(42초+4분5초)의 탐색시간이 소요되었다. 이 결과로 GA에서 80% 수렴한 결과를 역전파 알고리즘이 입력받아 학습을 실행

하는 경우는 초기 학습에서 수렴율은 제일 높지만 99.5% 수렴하기 위한 전체적인 소요시간을 본다면 4분 47초로 GA가 50% 수렴한 결과를 BP의 파라미터로 하여 학습을 실행한 CPU의 전체 실행시간인 4분 12초보다 더 오래 걸렸다.

따라서 제안한 방법에서 가장 효과적인 탐색 성능을 보인 경우는 GA에서 50%에서 수렴한 결과인 후보 운곽선 구조와 연결강도, 오프셋 값을 BP의 입력으로 학습 하였을 때 가장 효과적이었다.

결론적으로 GA와 BP 알고리즘의 단독 실행보다는 제안한 방법인 GA에서 적절한 연결강도와 오프셋 값을 탐색하여 BP의 입력으로 하여 학습하는 방법이 탐색시간의 성능을 향상시킬 수 있었고, 또한 가장 빠른 탐색의 운곽선 추출의 결과는 GA의 50%나 60% 수렴한 결과인 연결강도, 오프셋 값을 BP의 입력으로 하여 학습한 경우가 전체적인 시간을 비교해 보면 상대적으로 수렴 시간이 가장 빠른 결과를 나타내었다.

1) 역전파 알고리즘 단독 실행의 학습 수렴 결과

<표 2> 역전파 알고리즘의 단독 학습 결과 (㉔ 영상 결과)  
<Table 2> The result of learning using only back propagation (Image ㉔)

학습횟수	20	50	100	250	400
CPU 시간	20초	54초	1분 39초	4분 4초	7분 42초
수렴율(%)	80.6	92.7	97.65	99.03	99.78

$$\text{수렴율} = \frac{\text{목표하는 운곽선 픽셀의 위치와 일치하는 픽셀수}}{\text{전체 픽셀 수}}$$

<표 3> 역전파 알고리즘의 단독 학습 결과 (㉕ 영상 결과)  
<Table 3> The result of learning using only back propagation (Image ㉕)

학습횟수	20	50	100	250	400
CPU 시간	20초	57초	1분 48초	4분 21초	7분 53초
수렴율(%)	81.5	91.4	97.72	98.92	99.69

2) 유전자 알고리즘 + 역전파 알고리즘 병행 실행의 예 (단, 연결강도, 오프셋값은 임의값으로 하고 후보 운곽선 구조만 입력층의 입력으로 한 경우)

〈표 4〉 유전자 알고리즘 + 역전파 알고리즘 실행의 학습결과 (㉔ 영상 결과)

〈Table 4〉 The result of learning using both genetic algorithm and back propagation (Image ㉔)

학습횟수	20	50	100	250	400
CPU 시간	19초	52초	1분 39초	4분 1초	7분 36초
수렴율(%)	81.79	93.5	98.45	99.1	99.87

〈표 5〉 유전자 알고리즘 + 역전파 알고리즘 실행의 학습결과 (㉕ 영상 결과)

〈Table 5〉 The result of learning using both genetic algorithm and back propagation (Image ㉕)

학습횟수	20	50	100	250	400
CPU 시간	20초	54초	1분 39초	4분 4초	7분 42초
수렴율(%)	80.4	92.6	98.24	99.10	99.72

3) 제안한 방법

(후보윤곽선과 연결강도, 오프셋 값 모두 입력층의 입력으로 한 경우)

〈표 6〉 유전자 알고리즘 수렴 과정 (㉔ 영상 결과)

〈Table 6〉 The converging process of genetic algorithm (Image ㉔)

세대수	10세대	20	40	80	120	150	200	500
적합도 수렴율	0.31	0.51	0.62	0.774	0.803	0.837	0.854	0.952
탐색시간	4	7	12	27	42	54	1분19초	6분2초

〈표 7〉 역전파 알고리즘 학습 실행 과정 (㉔ 영상 결과)

〈Table 7〉 The learning process of back propagation (Image ㉔)

학습횟수	10	50	100	250	400	
적합도 수렴율	0.2	81	91.4	96.4	98.9	99.1
	0.3	83	92.7	97.2	99.1	99.12
	0.4	84	93.6	98.1	99.2	99.2
	0.5	88.2	97.6	98.9	99.5	99.5
	0.6	90.3	98.2	99.2	99.52	99.63
	0.7	91.7	98.4	99.5	99.54	99.62
	0.8	93.2	98.7	99.31	99.57	99.65
CPU 시간	10초	53초	1분48초	4분5초	7분43초	

〈표 8〉 유전자 알고리즘 수렴 과정 (㉕ 영상 결과)

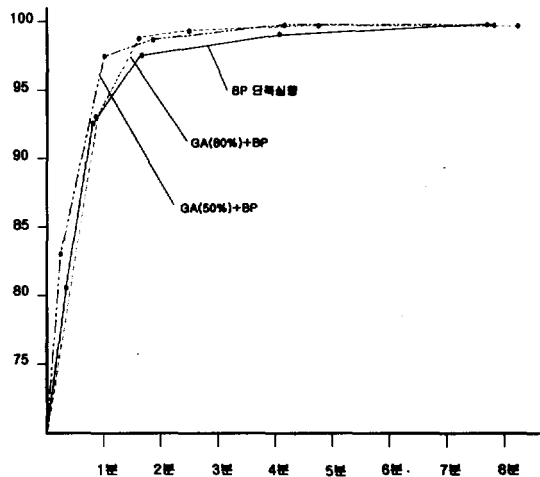
〈Table 8〉 The converging process of genetic algorithm (Image ㉕)

세대수	10세대	20	40	80	120	150	200	500
적합도 수렴율	0.27	0.59	0.724	0.781	0.843	0.845	0.857	0.962
탐색시간	5	14	24	38	47	1분 2초	1분 32초	6분 15초

〈표 9〉 역전파 알고리즘 학습 실행 과정 (㉕ 영상 결과)

〈Table 9〉 The learning process of back propagation (Image ㉕)

학습횟수	10	50	100	250	400	
적합도 수렴율	0.2	80	89.2	95.3	97.8	99.3
	0.3	81.4	91.3	96.1	98.4	99.2
	0.4	82.7	92.4	98.1	99.2	99.3
	0.5	87.6	97.1	98.9	99.6	99.5
	0.6	90.03	97.9	99.1	99.6	99.6
	0.7	92.1	98.4	99.6	99.62	99.7
	0.8	93.7	97.9	99.44	99.6	99.71
CPU시간	11초	54초	1분46초	4분2초	7분38초	



(그림 14) 윤곽선 추출 결과의 수렴율과 CPU 시간의 비교 (㉔영상 탐색 결과)

(Fig. 14) The converging rate and CPU time in edge detection. (Image ㉔)

## 5. 결 론

정확한 윤곽선을 추출하기 위한 조건은 원 영상의 물체와는 관계없는 잡음이 제거되고 윤곽선 구조가 연결성을 지니며 실제 물체의 경계와 일치하는 정확한 위치에 1개의 픽셀로 나타나는 것이다. 위의 조건을 만족하는 효율적인 윤곽선 추출 방법으로서 최적화 알고리즘인 유전자 알고리즘을 이용한 윤곽선 추출 방법이나 역전파 알고리즘을 이용한 형상인식 등에 관한 방법들이 제안되어 우수한 윤곽선 추출 결과를 얻을 수 있었다. 그러나 유전자 알고리즘은 최적해 부근에서의 지역적 미세 조정 기능이 없기 때문에 최적해 근처에서의 탐색회수가 많아지는 원인이 되고, 또한 역전파 알고리즘은 연결강도, 오프셋의 초기 값에 따라서 출력값과 목표하는 값과의 오차가 결정되므로 초기 학습에 적합한 연결강도와 유니트 값을 결정하는 것이 학습시간을 줄일 수 있는 조건이 되었다.

본 논문에서는 수렴속도가 향상된 윤곽선 추출을 위하여 역전파 알고리즘과 유전자 알고리즘을 병합한 방법을 제안하였다.

윤곽선 추출 방법은 먼저 유전자 알고리즘을 이용하여 연결강도, 오프셋 값을 구하였다. 이때 연결되고 정확한 윤곽선 구조로 만들어 주기 위하여 connection 연산자와 one\_pixel 연산자를 추가하였다. 다음으로 유전자 알고리즘의 탐색 결과를 역전파 알고리즘의 학습 매개변수의 초기 값으로 입력하여 학습시킨 후 최적의 윤곽선 구조를 추출하였다.

제안한 방법은 유전자 알고리즘 단독 실행 결과와 비교해서 보면 윤곽선 추출 영상은 거의 비슷한 결과를 나타내었지만, 수렴 시간의 성능 비교에서는 유전자와 역전파 알고리즘 단독 실행했을 때보다 제안한 두 방법의 병행한 방법이 수렴 시간의 성능에서 더 우수함을 나타내었다. 또한 제안한 방법에서 가장 수렴 시간이 빠른 경우는 GA에서 50% 수렴하고, 그 결과를 BP의 파라미터로 하여 학습을 실행하여 최적해에 수렴한 경우로 유전자 알고리즘 단독 실행보다 약 20% 더 빠른 수렴시간을 나타내었다.

앞으로의 연구 방향은 유전자 알고리즘을 이용하여 최적의 은닉층의 개수를 탐색하여 역전파 알고리즘의 최적의 신경망을 설계할 수 있도록 하는 방법과 유전자 알고리즘의 기존의 연산자 외에 탐색 기능을 향상시킬 수 있는 새로운 연산자에 대한 연구가 있어야 할 것으로 사료된다.

## 참 고 문 헌

- [1] A. Rosenfeld and A. C. Kak, Digital Picture Processing, New York Academic, Vol.1 and 2, 1982.
- [2] V. Torre and T. A. Poggio, "On Edge Detection," IEEE Trans. Patt. Anal. Machine Intel., Vol. PAMI-8, No.2, pp.147-163, Mar., 1986.
- [3] T. Peli and D. Malah, "A study of Edge Detection Algorithms," Computer Graphics Image Processing, Vol.20, pp.1-21, 1982.
- [4] 김수겸, "에지의 구조적 특성과 유전자 알고리즘을 이용한 에지추출", 조선대학교 박사학위 논문, 1996.
- [5] 배대규, "유전자 알고리즘을 이용한 윤곽선 추출", 정보과학회, 논문지(B), 제24권 제10호, 1997.
- [6] Jay F. Winkler. B. S. Manjunath, "Genetic Programming for Object Detection," "Genetic Programming 1997: proceeding of the second annual conference," pp.300-335, 1997.
- [7] H. Kitano, "Empirical studies on the speed of convergence of neural network training using genetic algorithms.," In Proc. 8th JMIT National Conf. in Artificial Intelligence, Vol.2, pp.789-796, 1990.
- [8] J.H.Holland, "Adaptation in natural and artificial systems," The Univ. of Michigan Press, 1975.
- [9] D. E. Goldberg, Genetic Algorithms in Search Optimization & Machine Learning, Addison. Wesley, 1989.
- [10] 조성배 역, "유전자 알고리즘", 대청컴퓨터월드, pp.33.
- [11] 임영희. 박대회, "기울기 강하 유전 연산자.", 한국정보과학회 추계 학술발표 논문집, 제21권 2호, pp. 339-342, 1989.
- [12] D. H. Vallard and C. M. Brown, Computer Vision, Prentice-Hall, 1982.
- [13] S. Sarker and K. L. Boyer, "On Optimal Infinite Impulse Response Edge Detection Filters," IEEE Transac. on PAMI, Vol.13, No.11, pp.699-714, November 1986.
- [14] H. Kitano, "Empirical studies on the speed of convergence of neural network training using genetic algorithms," In Proc. 8th JMIT National

Conf. in Artificial Intelligence, Vol.2, pp.789-796, 1994.

- [15] James A. Freeman and David M. Skapura, "Neural Networks : Algorithms, Applications, and Programming Techniques," Addison-Wesley Publishing Company, 1992.
- [16] Marilyn McCord Nelson and W.T. Illingworth, "A Practical Guide to Neural Nets," Addison-Wesley Publishing Company, 1991.



### 박 찬 란

1984년 2월 조선대학교 전자계산  
학과 졸업(이학사)

1989년 2월 조선대학교 대학원 전  
자계산학과 졸업(이학석사)

1995년 9월~현재 조선대학교 대학  
원 전산통계학과 박사과정

1991년 3월~현재 목포과학대학 전산정보처리과 조교수  
관심분야 : 영상처리, 신경망, 인공지능

### 이 응 기

조선대학교 자연과학대학 전산통계학과 교수