

## 유전자 알고리즘 Genetic Algorithms

김 기 화\*

### 1. Genetic Algorithm 의 기본 개념

Genetic Algorithms(GA)은 다아윈이 주장한 자연진화의 법칙인 적자생존(survival of fittest)과 자연도태(natural selection)의 원리를 토대로 하여 정립된 탐색 알고리즘이다. 자연진화의 법칙에 의하면 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 생존할 확률을 크게 가지며 교배(crossover)와 돌연변이(mutation)의 과정을 통해 보다 좋은 방향으로 진화해 나가고 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화의 과정을 되풀이 하면 결국 주어진 환경에 가장 적합한 형질의 개체들로 형성될 것이다. GA는 이러한 생명체의 자연진화 법칙을 일반 최적화 문제에 적용한 방법으로, 설계영역에 다수의 설계점을 분포시켜 목적함수 값과 제한조건의 위반 정도에 따라 각 설계점에 적합성(fitness)을 부여한다. 적합성이 클수록 다음 단계인 교배와 돌연변이의 과정에 참여할 확률을 크게 하여(reproduction) 적합성이 좋은 설계점에 비슷한 형질을 가진 설계점들이 다음 단계에 보다 많이 형성되어 계산이 진행 될수록 전체 설계점들은 좋은 방향으로 탐색이 진행된다.

GAs는 1970년대에 Michigan 대학의 John Holland와 그의 연구진에 의해 개발되었다. 이들의 연구 목표는 자연계의 적응과정을 구체화 하고, 이러한 과정을 모방한 인공적인 소프트웨어를 설계하고자 하였다. 자연계는 어떠한 특별한 환경이나 상태에 국한되지 않고, 보다 다양한 환경에 대해 일률적인 적응과정을 가진다. 따라서 이들 연구진들은 광범한 응용분야에 대해 범용성이 뛰어난 시스템(robust adaptive system) 개발에 중점을 두어 연구를 수행하였다. 그 후 GAs에 대한 수많은 연구가 수행되었는데, 대부분 다음 3가지 분야로 분류될 수 있다:(1) GA 자체에 대한 연구, (2) GA를 이용한 최적화 문제, (3) classifier system을 통한 machine learning.

Genetic algorithm 이란 용어는 좁은 의미로는 위에서 언급한 Holland 와 그의 연구진들에 의해 소개 및 연구된 genetic algorithm을 의미하며, 이를 특히 canonical genetic algorithm 이라 부른다. 그 후 대부분의 GA에 대한 연구는 이 canonical GA를 기초로 하여 수행되었으며, 많은 변화 및 발전이 있었다. 따라서 넓은 의미로는 설계집단 형태의 탐색을 기초로 하여 설계공간의 탐색을 위해 도태와 결합의 과정을 사용하는 방법을 통칭한다.

\* 서울대학교 조선해양공학과 Post Doc. 공학박사

## 2. Genetic Algorithm의 특징

GA는 기존의 최적화 방법과는 다른 다음과 같은 특징이 있다.

### 2.1 설계변수로 코드를 사용

GA는 설계변수의 실제값을 직접 사용하는 대신 2진수 코드 형태의 문자열(string)을 사용한다. (예: 100101) 이러한 문자열 형태의 설계변수의 사용은 염색체에 유전인자가 나열해 있는 것과 같은 형태를 지니고 있어 교배와 돌연변이의 변환 과정을 수행하기가 매우 단순하고 편리하다. 또한 이러한 문자열은 이산적인 성질을 지니고 있어 정수 또는 이산적 설계변수를 포함하는 혼합형 최적화 문제에 효과적으로 사용할 수 있다.

이러한 코드의 사용은 GA의 가장 큰 특징(장점)으로 GA를 black box 로 사용할 수 있는 근거가 된다. 앞으로 설명할 설계영역 탐색의 변환과정은 모두 이러한 문자열 형태의 코드형태로서 이루어진다. 그리고 나서 실제 변수값을 취급하기 위해서는 이러한 코드의 정보를 실제 설계변수 공간의 값으로 변환하여 주기만 하면 된다. 예를 들어 보자. 실제 설계변수  $x$ 가 0에서 31까지의 값을 가지고, 2진수 문자열은 길이가 5라고 하자. 길이가 5인 문자열이 표현할 수 있는 설계점의 수는  $2^5=32$ 개로 이들을 적절히 실제 설계변수 공간으로 변환하여 주면 된다. 대부분의 변환과정에 사용되고 또한 단순한 선형 매핑을 사용하면 다음과 같다.

coded variables (genotype)	mapping	actual variables (phenotype)
00000	----->	0
00001	----->	1
-	-	-
11111	----->	31

설계변수의 수가 하나 이상일 때는 모든 설계변

수를 하나의 문자열에 차례대로 나열하여 준다. 즉 설계변수 벡터가  $X=(x_1, x_2, \dots, x_n)$ 일 때 이를 각 설계변수가 가지는 문자열의 길이가 5라고 하면 0100110001 --- 10011 같이  $5 \times n$ 의 길이를 가지는 하나의 문자열로 표현된다. 이는 다음과 같이 매핑된다.

$x_1$	$x_2$	---	$x_n$
01001	10001	---	10011

이와 같은 코드를 사용함으로써 해서 탐색 설계공간의 종류에 관계없이 GA를 사용할 수 있으며, 후에 또 언급되겠지만 설계공간이 코드로 표현됨으로써 해서 설계공간 중 탐색의 주 대상이 되는 부분은 코드 형태의 유사성(similarity)을 가지게 되어 매우 효과적인 탐색이 가능하다.

### 2.2 다수의 설계점들이 집단 탐색

대부분의 최적화 방법이 한점에서 한점에서의 이동에 의한 국부적인 탐색과정을 가지는 반면에, GA는 여러개의 설계점들이 집단(population)을 이루어 동시에 탐색을 행하여 보다 넓은 설계영역에 대한 정보를 활용함으로써, 전체 최적점에 수렴할 확률이 기존의 방법에 비해 상대적으로 매우 크다.

### 2.3 직접 탐색방법

GA는 목적함수와 제한조건의 값만을 사용하고 미분값이나 그 외의 다른 정보를 필요로 하지 않는 직접 탐색방법을 이용하므로 실제 구조물의 설계와 같은 복잡하고 다양한 환경의 최적화 문제에 적합하며 또한 기본 모델의 변경으로 인한 수정이 용이하다.

### 2.4 확률론적인 변환과정을 사용

일반적으로 최적화 방법에서는 탐색을 위한 변환과정을 확정론적인 방법을 사용한다. 즉 현 설계점에서 다음 설계점에서의 이동이 현재의 설계정보를 가지고 확정론적으로 결정된다. 이러한 탐색 방법은 설계 공간과 초기점이 주어지면 최종

결과가 결정되어 지므로, 초기점의 선정이 매우 중요한 요인이 된다. 하지만 GA는 탐색과정을 확률론적으로 행하므로 설계집단을 통한 탐색과 더불어 설계공간 전체에 대한 정보 활용이 매우 용이하다. 여기서 확률론적이란 개념은 뒤에서도 언급이 되겠지만 단순한 random search가 아니라, 보다 좋은 설계공간으로 탐색을 안내해 주는 도구로서 random choice를 사용한다는 것이다.

이와 같이 GA의 전체 전개 과정은 매우 단순하다. 하지만 다수의 설계집단과 도태, 교배 그리고 돌연변이의 과정이 합쳐진 결과는 매우 강력하다. 따라서 GA의 가장 큰 장점은 설계모델에 제약성(변수의 연속성, 미분값의 존재, unimodality 등)이 매우 적어 다양한 종류의 문제에 대해 범용성이 매우 넓다는 것이다. 이를 그림으로 도시하면 Fig. 1 과 같으며, 설계공간이 연속적으로 되어있고 국소 최적점이 존재하지 않는 경우를 제외하고는 기존의 최적화 방법에 비해 GA의 성능이 우수함을 알 수 있다.

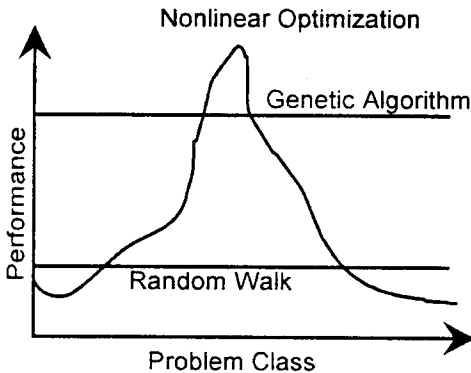


Fig.1 Genetic algorithm as a robust optimizer

### 3. Schema 정리

GAs는 다수의 설계점들이 설계영역을 탐색하며, 각 설계변수는 코드 형태의 문자열 구조를 가지고 있다. 이들 설계점들은 여러 변환과정을 통해 점차 좋은 설계점과 비슷한 형질의 설계점들이 보다 많이 생성되어, 결국 전체 설계점들은 보다

좋은 방향으로 이동하게 된다. 이러한 설계점들의 이동이 설계영역 전체에 걸쳐 얼마나 효과적으로 이루어 지는지 검토할 필요가 있다. GAs의 변환 과정은 코드형태인 문자열의 단순한 물리적인 조작에 의해 이루어 진다. 따라서 이들 문자열의 모양을 먼저 면밀히 검토할 필요가 있다. 예를 들어 적합성 함수가  $f(x)=x^2$ 와 같을때, 다음의 문자열에 대해 변수(x)와 적합성은 다음과 같은 값을 가진다.

No.	string	x	fitness
1	01101	13	169
2	11000	24	576
3	01000	8	64
4	10011	19	361

위의 표를 자세히 살펴보면 문자열의 형태가 설계변수나 적합성의 형태에 비해 유사성이 매우 뛰어난 것을 알 수 있다. 이를 보다 자세히 살펴보기 위해서\*, 즉 don't care symbol을 도입하여 보자. 위의 표에서 적합성이 가장 큰 두 문자열은 공통적으로 1\*\*\*\* 형태로 되어 있다. 이는 실제 설계변수 값 24, 19와 비교해 보면 그 형태적 유사성이 매우 우수하게 나타나 있음을 알 수 있다. 이와 같이 전체 설계집단을 구성하는 설계점들의 문자열의 형태가 목표로 하는 설계값(최적치) 방향으로 갈수록 유사성을 가진다는 것은 매우 주목할 현상이며, 이는 다음에 설명할 building block hypothesis의 근간이 된다.

#### 3.1 용어 정의

문자열의 표시로 2진수만을 사용한다고 가정하면 문자열을 구성하는 각 인수는 0 또는 1의 값만을 가지게 된다. 여기에 윗 절에서 언급했던 또 하나의 문자\*, don't care symbol을 정의 하자. 이는 정의대로 0 또는 1 어느 것이든 사용가능하다는 의미이다. 즉(0, \*, 1, \*)인 집합에 속하는 문자열은(0, 0, 1, 0), (0, 1, 1, 0), (0, 0, 1, 1) 그리고 (0, 1, 1, 1)의 4가지가 모두 포함된다.

그리고 다음의 2 개념을 정의하자.

Order of a schema H,  $o(H)$  ; the number of fixed positions

$$o(011*1**) = 4, \quad o(1***** ) = 1$$

Defining length of a schema H,  $\delta(H)$  ; the distance between the first and last specific string position.

$$\delta(011*1**) = 5-1 = 4, \quad \delta(1***** ) = 0$$

### 3.2 Effect of reproduction, crossover, mutation on schemata

설계집단 속에 존재하는 다수의 스키마들이 GA의 설계변경 인자인 reproduction, crossover 그리고 mutation 에 의해 어떠한 형태로 진화해 가는지 살펴보자.

#### 1) The effect of reproduction

시간 t에서 설계집단 A(t)에는 어떠한 스키마 H의 갯수를 m 이라고 가정하자. 이는 다음과 같이 표기된다,  $m(H,t)$ . 설계집단 내의 문자열은 reproduction 과정에서 자신의 적합성의 상대적 크기에 비례하여 복사된다. 따라서 시간 t+1 일 경우 스키마 H의 개수는 다음과 같이 표현된다.

$$m(H,t+1) = m(H,t) \cdot n \cdot f(H) / \sum f$$

$$= m(H,t) \frac{f(H)}{\bar{f}}$$

여기서 n ; population size

$f(H)$  ; average fitness of the string representing schema H at time t

$f_i$  ; fitness of i-th string

$\bar{f}$  ; average fitness of entire population ( $= \sum f_i / n$ )

이로부터 어떠한 스키마는 전체 설계집단의 적합성의 평균에 대한 그 스키마의 적합성의 평균값의 비 만큼 증가 성장한다는 것을 알 수 있다. 다시 말하면, 설계집단의 적합성의 평균값 이하의 스키마들은 다음 세대에서 그 수가 줄어들고, 이상인 스키마들은 늘어 남을 알 수 있다. 여기서 또 하나 살펴볼 것은, reproduction에 의해 설계집단 내에 존재하는 모든 스키마들은 그들의 적합성의

평균값의 크기에 따라 증가하거나 소멸한다는 것이다. 이와 같은 reproduction의 효과를 고찰하기 위해 어떠한 스키마 H의 적합성의 평균값이 상수 전체 적합성의 평균 보다  $(1+c)$  배라고 가정하면, t+1 시간에서는

$$m(H,t+1) = m(H,t) \frac{(\bar{f} + c\bar{f})}{\bar{f}}$$

$$= (1+c) \cdot m(H,t)$$

시간 t=0에서 시작했다고 하면, 시간 t에서는

$$m(H,t) = m(H,0) \cdot (1+c)^t$$

가 된다. 이 식으로부터 reproduction의 효과는 전체 평균값보다 큰(작은) 평균값을 가지는 스키마는 세대가 계속됨에 따라 기하학적으로 증가(감소) 한다고 설명될 수 있다.

#### 2) The effect of crossover and mutation

Reproduction에 의해 좋은 스키마들이 보다 많이 다음 세대에 생성되었지만, 이는 단순한 복사의 효과만을 가지고 있으므로, 현 세대의 설계집단이 가지고 있는 설계영역 외의 다른 부분은 탐색하지 못하였다. 따라서 설계영역의 탐색의 확대를 위해서 crossover와 mutation 과정을 수행한다.

먼저 crossover가 스키마에 미치는 효과를 살펴 보기 위해 다음의 길이가 7인 문자열들을 예로 들어 보자.

$$A = 0111000$$

$$H_1 = *1****0$$

$$H_2 = ***10**$$

스키마에 대한 crossover의 효과를 살펴보자. 여기서는 가장 단순한 one-point crossover을 사용하였다. Crossover가 3번째와 4번째 문자열 사이에서 일어난다고 하고, 이를 '|' 로 표시하면 다음과 같다.

$$A = 011|1000$$

$$H_1 = *1*|***0$$

$$H_2 = *** | 1 0 **$$

$H_1$ 은 mating이 같은 스키마가 되지 않는 한 그 형태가 파손 된다. 하지만  $H_2$ 는 그 형태가 유지된다. 이와 같이 스키마 H가 교배에 의해 생존 또는 소멸될 가능성은 그 스키마의 defining length  $\delta(H)$ 와 전체 문자열의 길이( $l$ )의 비에 의존한다는 것을 알 수 있다. 즉  $H_1$ 이 교배에 의해 파괴될 확률은  $p_d = \delta(H_1) / (l-1) = 5/6$ 이 되고, 생존할 확률은  $p_s = 1 - p_d = 1/6$ 이 된다. 일반 스키마 H에 대해서도 생존할 확률은  $p_s = 1 - \delta(H) / (l-1)$ 이 된다. 교배가 확률  $p_c$ 의 확률로 행해진다고 하면 H가 생존할 확률과 교배가 행해진 후 그 스키마의 생성 개수 ( $m(H, t+1)$ )는 다음과 같다.

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1}$$

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - p_c \frac{\delta(H)}{l-1}]$$

여기서 부등호는 다른 스키마가 교배에 의해 스키마 H가 되는 경우(gains)와 교배를 위한 mating이 같은 스키마 끼리 행해지는 것을 제외 했기 때문이다. 이를 좀더 자세히 적으면 다음과 같다.

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - p_c \frac{\delta(H)}{l-1} (1 - \frac{m(H, t)}{n}) + p_c \cdot (\text{gains})]$$

이 식에서 스키마의 평균 적합성이 전체 설계집단의 평균 적합성 보다 크고, 짧은 defining length를 가지는 스키마들은 세대가 진행됨에 따라 그 수가 기하학적으로 증가함을 알 수 있다.

다음은 돌연변이의 효과를 살펴보자. 돌연변이는 확률  $p_m$ 로 비트 단위로 수행된다. 어떤 스키마 H가 돌연변이 과정에서 생존하려면 스키마의 문자열중 확정된 부분이 살아 남아야 되는데, 이들의 갯수는  $o(H)$  이다. 문자열중 어떤 한 allele가 살아 남을 확률은  $(1 - p_m)$ 이다. 따라서 스키마 H가 생존할 확률은  $(1 - p_m)^{o(H)}$ 가 된다. 일반적으로

돌연변이 확률  $p_m$ 은 매우 작은 값을 사용하므로 스키마의 생존 확률은  $[1 - o(H) \cdot p_m]$ 으로 근사될 수 있다.

이상의 결과에서 어떠한 스키마 H가 재생, 교배, 돌연변이의 과정을 행한 후 다음 세대에서 예상되는 갯수는 다음의 식으로 표현된다.

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - p_c \frac{\delta(H)}{l-1}] [1 - o(H) \cdot p_m]$$

윗 식에서 cross-product 되는 항을 무시하면 다음과 같다.

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} [1 - p_c \frac{\delta(H)}{l-1} - o(H) \cdot p_m]$$

윗 식으로부터 길이가 짧고, 적은 차수 그리고 전체 평균 이상의 평균값을 가지는 스키마는 세대가 진행될수록 기하학적으로 그 수가 증가한다. 이것이 Genetic Algorithm의 근간 이론인 Schema 이론이다.

Schema theorem : Short, low order, above-average schemata receive exponentially increasing trials in subsequent generations.

### 3.3 The basic schema function and implicit parallelism

설계집단에서 어떠한 스키마가 형성될 수 있는 갯수를 살펴보자. 예를들어 먼저 길이가 4이고 차수가 1 과 2인 스키마는 다음과 같다.

#### order 1 schemata

0\*\*\* \*0\*\* \*\*0\* \*\*\*0  
1\*\*\* \*1\*\* \*\*1\* \*\*\*1

#### order 2 schemata

00\*\* 0\*0\* 0\*\*0 \*00\* \*0\*0 \*\*00  
01\*\* 0\*1\* 0\*\*1 \*01\* \*0\*1 \*\*01  
10\*\* 1\*0\* 1\*\*0 \*10\* \*1\*0 \*\*10  
11\*\* 1\*1\* 1\*\*1 \*11\* \*1\*1 \*\*11

이를 일반화 하면 길이가 L인 문자열에서 차수가 i 인 스키마는 0 과 1 의 구별이 없다면,  ${}_L C_i$  개가 형성될 수 있으나, 0 과 1 의 구별로 인해, i 차수의 스키마 각각은 2개의 스키마로 나타낼 수 있다. 따라서 길이가 L 이고 차수가 i 인 문자열이 표현할 수 있는 스키마의 수는  ${}_L C_i 2^i$ 이다.

그리고 문자열에서 차수가 i 인 스키마가 형성될 확률은 다음과 같다.

$$P(\text{single match of a schema of order } i) = \left(\frac{1}{2}\right)^i$$

그러면 크기가 n 인 설계집단 내에 차수가 i 인 스키마가 하나도 존재하지 않을 확률은 다음과 같다.

$$P(\text{no matches of an order } i \text{ schema}) = \left[1 - \left(\frac{1}{2}\right)^i\right]^n$$

크기가 n 인 설계집단에서 차수가 i 인 스키마가 적어도 하나 이상 형성될 확률은 다음과 같다.

$$P(\text{at least one success}) = 1 - \left[1 - \left(\frac{1}{2}\right)^i\right]^n$$

따라서 크기가 n 인 설계집단에서 길이가 L인 문자열로서 형성될 수 있는 스키마의 전체 갯수 S(n,L)는 다음과 같다.

$$S(n,L) = \sum_{i=0}^L {}_L C_i 2^i \left\{1 - \left[1 - \left(\frac{1}{2}\right)^i\right]^n\right\}$$

이 식을 schema function 이라 한다. 이 식을 이항정리(binomial theorem)를 이용하여 정리하면 다음과 같다.

$$S(n,L) = 3^L - \sum_{i=0}^L {}_L C_i 2^i \left\{1 - \left[1 - \left(\frac{1}{2}\right)^i\right]^n\right\}$$

설계집단의 크기 n=1 이면, schema function 은  $2^L$ 가 되고, n이 매우 커지면  $3^L$ 가 된다. 그리고 윗 식에서  $\left[1 - \left(\frac{1}{2}\right)^i\right]^n$ 항에 이항정리를 사용하면,

$$S(n,L) = 3^L - \sum_{i=0}^L {}_L C_i 2^i \sum_{j=0}^n {}_n C_j (-1)^j \left(\frac{1}{2}\right)^{ij}$$

가 되며, 이 식을 다시 이항정리를 사용하여 n에 대한 형태로 바꾸어 주면.

$$S(n,L) = 3^L - \sum_{j=0}^L {}_n C_j (-1)^j \left[1 + \left(\frac{1}{2}\right)^{j-1}\right]^L$$

로 schema function 이 표현된다. 이 식을 n=1인 경우 까지만 고려하면

$$S(n,L) \approx 3^L - 3^L + n 2^L = n 2^L$$

이 된다. 이 식은 근사식으로 이 식이 실제로 적용될 수 있는 설계집단의 크기를 살펴보자. 윗 식의 다음 항은  $-\frac{n(n-1)}{2} \left(\frac{3}{2}\right)^L$ 로 표현되며, 윗 식이 가지고 있는 오차량을 표현한다. 만일  $\epsilon$ 의 오차를 허용하는 스키마의 수를  $\hat{n}$ 이라 하고, 이때의 설계집단의 크기를  $\hat{n}$ 이라 하면,

$$\hat{n} \leq \frac{2\epsilon}{1+\epsilon} \left(\frac{4}{3}\right)^L + 1 < C \left(\frac{4}{3}\right)^L, \hat{S} = \hat{n} 2^L$$

으로 표현된다. 여기서 L을 소거 시켜  $\hat{S}$ 와  $\hat{n}$ 의 상관 관계를 보면

$$\hat{S} = \hat{n} 2^L > C_1 \hat{n}^{3.4} > C_1 \hat{n}^3$$

이 됨을 알 수 있다. 따라서 크기가 n인 설계집단에서 작동되는 스키마의 수는 설계집단 크기 n의 3승에 비례한다. 즉 차수가  $O(n^3)$ 이 된다. 이것이 GA 의 성능을 설명하는 매우 중요한 결과로 implicit parallelism 이라 한다. GA는 매 세대마다 설계집단의 크기(n) 만큼의 탐색 영역에 대한 연산을 수행하지만, 그 외의 어떠한 정보나 연산 없이 대략  $n^3$ 개의 스키마들이 동시에 작동되는 과정을 가진다.

#### 4. Building block hypothesis

Schema 정리에 의하면 GA가 실제로 작동되는

과정을 파악할 수 있다. 즉 GA는 길이가 짧고, 차수가 적으며, 적합성이 큰 스키마들이 탐색의 대부분을 차지하고, 이들의 재결합을 통해 새로운 스키마들이 형성되어, 보다 높은 적합성을 가지는 문자열을 형성하게 된다. 이러한 특별한 스키마들을 building block 이라 한다. 결국 어떠한 설계공간의 탐색을 위해, 예상될 수 있는 모든 조합을 다시 시도해 보는 대신에, 현재 탐색된 공간에 대한 정보 중에서 이러한 building block을 재결합하여 점차 보다 좋은 설계공간으로 탐색을 행하여 간다. 이와 같은 스키마의 탐색 성능을 보기 위해 실제로 스키마들이 차지하는 설계영역을 그림으로 보기로 하자.

다음 그림들은 어떠한 함수  $F(x)$  를 최대화 시키는 문제로서, 적합성은  $F(x)$  값과 같다고 하자. 길이가 6인 문자열 중 차수가 1 인 다음 스키마를 고려하여 보자. Fig. 2 의 빗금친 부분은 스키마 1\*\*\*\*가 형성하는 부분을 나타내고, 흰색 부분은 0\*\*\*\* 스키마의 영역이 된다. 그리고 Fig. 3 의 빗금침 부분은 스키마 \*\*1\*\*, 흰색 부분은 \*\*0\*\* 스키마가 된다. 각 경우에서 스키마의 적합성을 그림에서 직관적으로 비교해 보면 두 경우 모두 빗금친 부분이 높은 적합성을 가지고 있다. 따라서 스키마 정리에 의해 1\*\*\*\*와 \*\*1\*\*\* 스키마가 결합될 가능성이 높으며, 그러한 결합의 결과는 보다 좋은 탐색 공간을 확보할 가능성을 준다. 이 둘이 교배에 의해 새로 만들어진 스키마는 1\*1\*\*\* 이며 이는 Fig. 4 에 도시하였다. 여기서 윗 절에서 언급한 스키마의 설계영역 분할을 그림으로 확인할 수 있다. 즉 차수가 1인 스키마는 영역의 절반을 표현하고, 차수가 2인 스키마는 영역의 1/4 을 차지하므로, 차수가  $i$  인 스키마는  $1/2^i$ 만큼의 영역을 표현한다. 그리고 같은 차수에서도 고정된 위치에 따라, 그 스키마의 영역이 매우 다양하게 표현된다. GA는 이와 같이 다양하게 표현될 수 있는 스키마 중에서, building block의 재결합에 의해 최적점으로 탐색을 진행한다. 만일 이러한 building block의 결합에 의한 탐색의 방향이 최적점으로 방향과 다른 곳으로 움직이면, 최적점을 찾는 데 상당한 시간이 요구될 것이다. 이와 같이 설

계공간에서 형성되기 쉬운 building block의 결합에 의해 최적점이 구현되기 어려운 문제를 GA-deceptive problem 이라 하고, 이러한 문제의 대표적인 예가 격리된 최적점(isolated optima)을 가지는 문제, 즉 최적점이 가장 나쁜 점으로 둘러 싸인 문제를 들 수 있다. Building block hypothesis 를 정리하면 다음과 같다(Goldberg).

Building block hypothesis :

A genetic algorithm seeks near optimal performance through the juxtaposition of short, low-order, high-performance schemata, or building blocks.

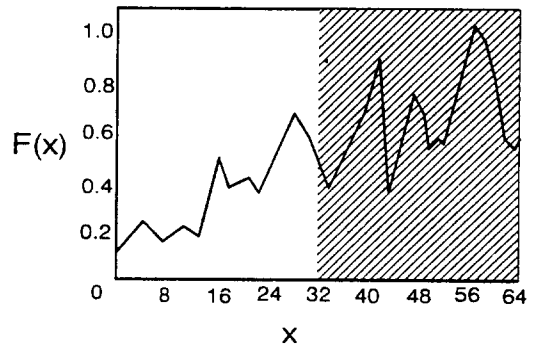


Fig.2 Sketch of schema

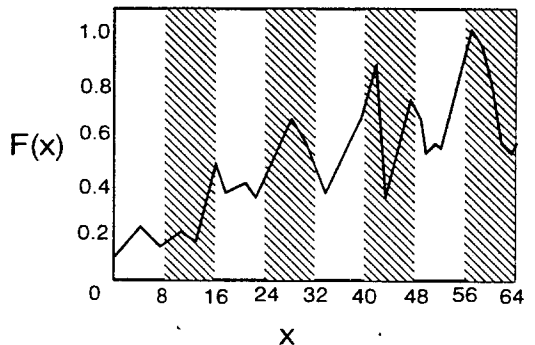


Fig.3 Sketch of schema

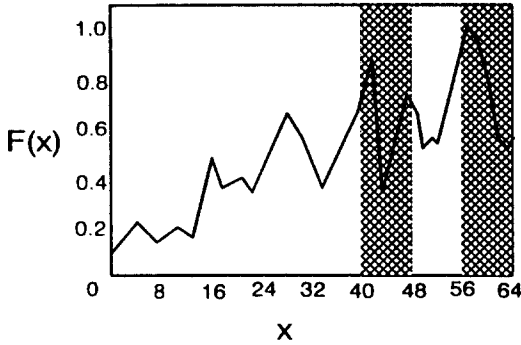



Fig.4 Sketch of schema

## 참 고 문 헌

1. 김기화, "Genetic Algorithm 에 의한 다목적함수 최적구조설계", 서울대학교, 조선해양공학과, 박사학위논문, 1994.
2. Goldberg, D.E., "Genetic Algorithms in search, optimization & machine learning", Addison-Wesley, 1989.
3. 北野 宏明, "유전적 알고리즘", 산업도서, 1993.
4. Holland, J.H., "Adaptation in natural and artificial systems", University of Michigan, Ann Arbor, MI, 1975. 

한국전산구조공학회는 96년도 가을 학술발표회 및 임시총회를 다음과 같이 개최하고자 하오니 회원 여러분의 많은 논문 발표와 참여 바랍니다.

개 최 일 : 1996년 10월 5일(토)

개최장소 : 제주대학교 공과대학

논문제목 제출 : 1996년 8월 31일까지

논문제출 기간 : 1996년 9월 14일까지

제출 및 문의처 : 학회 사무국(서울시 강남구 역삼동 635-4 과학기술회관 808호)

전화번호 (02)565-0035, FAX.(02)565-0036

추후 회원여러분께 논문발표계획서 및 논문작성요령을 보내드리오니 참조하시어 마감일과 논문작성 양식을 반드시 지켜주시기 바랍니다.