제 4 장

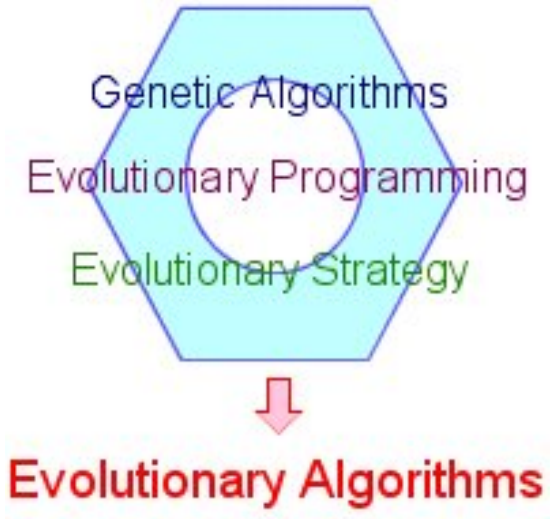
유전자 알고리즘
(Genetic Algorithm)

Genetic Algorithm

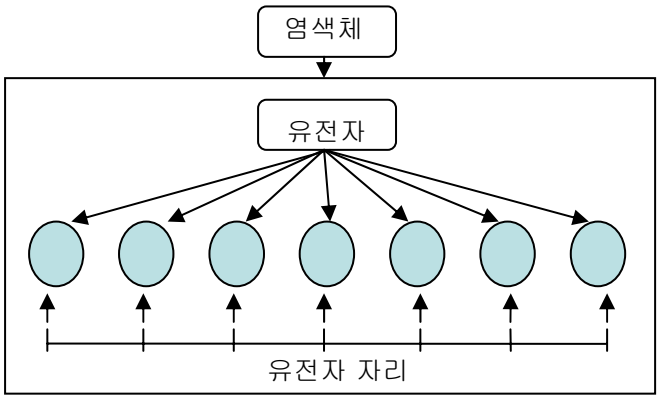
- 1975년 미국 미시간 대학의 John Holland 교수가 세포의 작용을 연구하던 중 제안한 것으로, 생물학계의 적자생존 원리 즉, 자연도태 원리를 기초로 한 최적화 알고리즘.
- 생물의 유전과 진화알고리즘을 공학적으로 모델화하여 문제해결이나 시스템의 학습등에 응용하려는 알고리즘
- 어떤 세대(generation)을 형성하는 개체(individual)들의 집합, 즉 개체군(population)중에서 환경에 대한 적합도(fitness)가 높은 개체가 높은 확률로 살아 남아 재생(reproduction)할 수 있게 되며 이때 교배(crossover) 및 돌연변이(mutation)로서 다음 세대의 개체군을 형성하게 되는 생물의 진화 과정을 인공적으로 모델링한 알고리즘
- 진화적 알고리즘(evolutionary algorithm)이라고도 함
- 2000년대 붐을 일으키고 있으며, Job shop scheduling, 훈련신경회로망, 이미지 특징 추출 및 인식, 최적화 문제등에 응용

진화적 알고리즘의 흐름

- 진화적 알고리즘(Evolutionary Algorithm, EA) 흐름
  - 유전자 알고리즘(Genetic Algorithm, GA)
    - 협의의 알고리즘으로 고정길이의 이진 스트링 염색체 사용
  - 유전자 프로그래밍(Genetic Programming, GP)
    - 트리 등을 염색체 표현에 사용, 90년대 개발
  - 진화적 프로그래밍(Evolutionary Programming, EP)
    - 그래프 등을 염색체 표현에 사용, 90년대 개발
  - 진화적 전략(Evolutionary Strategy, ES)
    - 실수의 값을 갖는 염색체 사용



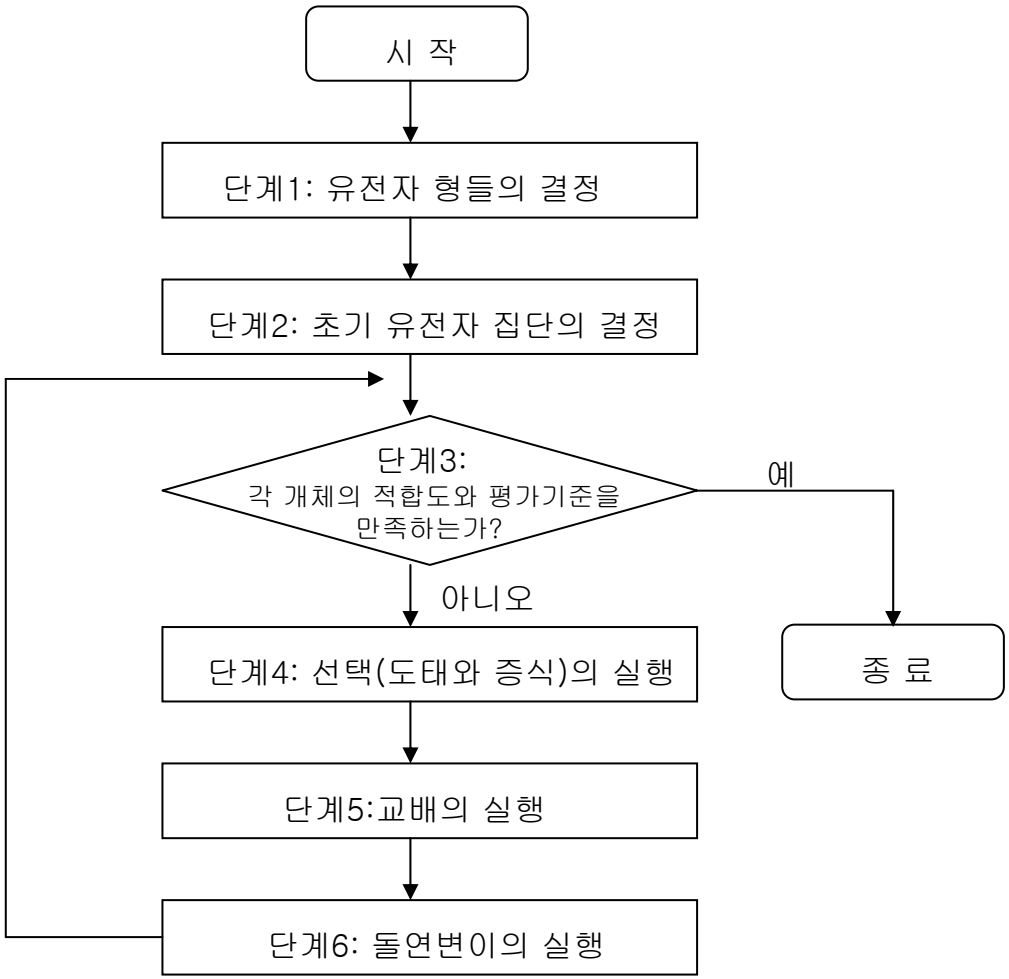
생물학과 GA 용어비교



<염색체, 유전자, 유전자자리>

생물학	유전자 알고리즘(GA)
개체(individual)	염색체에 의해 특징지어지는 자율적인 하나의 작은 집단
집단 (population)	집단 내의 개체의 수
유전자(gene)	개체의 형질을 규정하는 기본 구성요소 즉, 특성(feature), 형질(character) 등
염색체 (chromosome)	복수의 유전자 모임. 문자열(string)로 표현
대립 유전자(allele)	유전자가 갖는 특성 값(feature value).
유전자 자리(locus)	염색체상의 유전자의 위치 즉, 문자열의 위치 (string position)
적합도 또는 적응도(fitness)	유전자의 각 개체의 환경에 대한 적합의 비율을 평가하는 값 즉, 평가치로 최적화 문제를 대상으로 하는 경우 목적함수 값이나 제약조건을 고려하여 페널티 함수 값의 적응도로 설정된다.
코딩(coding)	표현 디코딩에서 유전자형으로 매핑하는 것
디코딩 (decoding)	유전자 형에서 표현형으로 역 매핑하는 것
유전자형 (genotype)	형질의 염색체에 의한 내부적으로 표현하는 방법으로 구조체(structure)로 표현
표현형 (phenotype)	염색체에 의해 규정된 형질을 외부적으로 표현하는 방법으로 파라메터 집합(parameter set), 대체해(alternative solution),디코드화를 위한 구조체(decoded structure)로 표현

GA 프로세스



a) 선택(selection) : 집단 중에서 적응도의 분포에 따라서 다음의 단계로 교배를 행하는 개체의 생존 분포를 결정한다. 적응도의 분포에 기초하고 있기 때문에 적응도가 높은 개체일수록 보다 많은 자손을 남기기 쉽게 된다.

b) 교배(crossover) : 2개의 염색체 사이에서 유전자를 바꾸어 넣어 새로운 개체를 발생시킨다.

c) 돌연변이(mutation) : 유전자의 어떤 부분의 값을 강제적으로 변화시킨다.

Characteristics

- 파라메터를 코딩한 것을 직접이용
- 점(point)이 아닌 다점(multi points, : 군(population)) 탐색 방법
- 탐색에 비용 정보(fitness function)를 이용하며, blind search수행 (미분값이나 다른 부가적인 지식을 요구하지 않음)
- 결정론적인 규칙이 없고 확률적 연산자를 사용하여 수행됨
- 다른 탐색방법이나 최적화 방법 보다 효율적

GA의 구성요소

- 개체
  - 풀고자 하는 문제의 변수값을 binary string으로 표현
    (예) 0100, 1011001
  - TSP 문제인 경우
    (3 4 1 7 2 5 8 6)으로 표현: 숫자는 도시의 위치
  - 각 유전자의 비트에 의미를 두고 디코딩시 해석함

선택메커니즘(Fitness Function)

- 목적 함수(objective function) 즉, 최적화 하고자 하는 함수는 각 개체의 적합도를 평가하는 기반이 된다.
- 목적함수의 값의 범위는 문제마다 다르기 때문에 보통 정해진 구간 사이의 양수 값을 갖도록 표준화된 값을 사용. 즉, 표준화하기 이전의 적합도의 값을 raw fitness라고 하며 표준화되어서 실제로 개체 선택의 기준이 되는 함수를 적합도 함수 (fitness function)라고 한다. 집단 중 다음단계로 교배를 수행하는 개체의 생존분포를 결정하는 것을 의미.
- raw fitness를 표준화(또는 스케일링)하는 방법
  - 선형 표준화(linear scaling)
  - σ절단(σ truncation) :계산된 적합도의 표준편차를 고려하는 방법
  - 거듭제곱 표준화(power law scaling)

선택 메커니즘 종류

- 자연선택(natural selection)현상을 모델링하여 잘 적응한 해들은 살아남고 잘 적응하지 못한 해들은 도태되도록 유도한다.
- 방법
  - 기본 모델 : 적합도 비례 선택(proportionate selection), 룰렛 선택법(roulette selection)
    - 각 개체 s_i 의 적합도 $f(s_i)(>0)$, i = 1, ..., N 의 총합을 구해, 각 개체 s_i의 선택 확률을 $p_i = \frac{f(s_i)}{\sum_{j=1}^{N} f(s_j)}$ 와 같이 정하는 방법
  - 엘리트 보존 선택(elitist preserving selection)
    - 확률에 따라 개체를 선택하여 교배 및 돌연변이의 결과로 특별히 좋은 해가 소실되는 것을 막기 위해 가장 좋은 해를 보존하여 다음 세대에 남기는 방법
    - 일반적으로 다른 선택 방법과 융합하여 사용가능

선택 메커니즘(cont.)

- 기대치 선택법(expected-value selection)
  - 적합도 비례선택의 문제점은 개체군의 크기가 크지 않을 경우에 적합도가 정확히 반영되지 않을 가능성이 있다. 이런 문제점의 대안으로 제안된 것이 기대치 선택법이다. 적합도에 대한 각 개체의 확률적인 재생 개체수를 구하여 선택하는 것이 기본적인 방법이다
- 순위 선택법(ranking selection)
  - 적합도의 크기 순서에 따라 순위를 매긴 후 순위에 따라 다음세대에 자손을 남길 확률을 결정하는 방법
- 토너먼트 선택법(tournament selection)
  - 개체군 중에서 일정한 개수의 개체를 임의로 선택하여 그 중에 최고의 적합도를 가지는 개체를 다음 세대에 남기는 방법(토너먼트 방식).
  - 다음 세대의 개체수가 모두 찰 때까지 반복적으로 계속 수행
- 기타 GENITOR 알고리즘 등

GA 기본연산: Crossover

- 교배(crossover) :교배는 2개의 개체간에 염색체를 부분적으로 서로 바꿈으로써 새로운 개체를 생성하는 것이다. 이 때 부모의 형질이 자손에게 적절히 계승되어야 하며 이것을 형질 유전성 (character preservingness) 또는 교배 근접(crossover neighborhood)이라 한다.
  - ① 단순 교배(simple crossover)

$$p_1 = a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10} \qquad s_1 = a_1a_2a_3b_4b_5b_6b_7b_8b_9b_{10}$$
$$\updownarrow \qquad \rightarrow$$
$$p_2 = b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10} \qquad s_2 = b_1b_2b_3a_4a_5a_6a_7a_8a_9a_{10}$$

  - ② 복수점 교배(multipoint crossover) : 교배점이 2개 이상

$$p_1 = a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10} \qquad s_1 = a_1a_2b_3b_4b_5a_6a_7b_8b_9b_{10}$$
$$\updownarrow \quad \updownarrow \quad \updownarrow \qquad \rightarrow$$
$$p_2 = b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10} \qquad s_2 = b_1b_2a_3a_4a_5b_6b_7a_8a_9a_{10}$$

GA Operators(cont.)

– ③ 균일 교배(uniform crossover): 마스크에 의한 교배

$p_1 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$

$m = 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0$

$p_2 = b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10}$

→

$s_1 = a_1\ b_2 b_3 a_4 a_5 a_6\ b_7 a_8\ b_9 b_{10}$

$m = 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0$

$s_2 = b_1\ \underline{a_2 a_3} b_4 b_5 b_6\ \underline{a_7} b_8\ \underline{a_9 a_{10}}$

– ④ 부분 일치 교배(partially matched crossover : PMX)

A = 9 8 4 | 5 6 7 | 1 3 2 10 → 9 8 4 | 2 3 10 | 1 3 2 10

B = 8 7 1 | 2 3 10 | 9 5 4 6 → 8 7 1 | 5 6 7 | 9 5 4 6

→ A' = 9 8 4 | 2 3 10 | 1 6 5 7

→ B' = 8 10 1 | 5 6 7 | 9 2 4 3

GA Operators(cont.)

- ⑤ 순서 교배(ordered crossover : OX)

A = 9 8 4 | 5 6 7 | 1 3 2 10

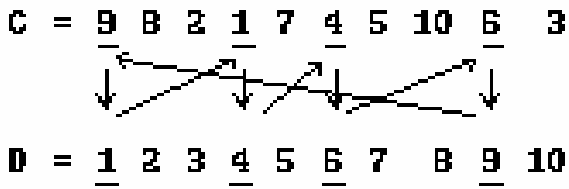
B = 8 7 1 | 2 3 10 | 9 5 4 6

→ 8 H 1 | 2 3 10 | 9 H 4 H

→ 2 3 10 | H H H | 9 4 8 1

→ 2 3 10 | 5 6 7 | 9 4 8 1

- ⑥ 주기 교배(cycle crossover : CX)



→ C' = 9 – – 1 – 4 – – 6 –

→ C' = 9 2 3 1 5 4 7 8 6 10

D' = 1 8 2 4 7 6 5 10 9 3

GA Operator: Mutation

- 개체의 각 유전자자리의 유전자에 대하여 일정한 돌연변이 확률 (pm) 을 적용하여 대립 유전자의 값으로 바꾸는 것
- 개체에 근접한 새로운 개체를 생성하는 국소적인 랜덤 탐색의 일종
- 또한 집단에서 잃어버린 유전형질을 복구하여 다양성을 유지하기 위한 수단으로도 사용됨. 이때 전형적인 돌연변이 확률은 0.05이하
- (예) 세 번째 비트인 a3 가 A3 로 바뀌었음

$$p = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \quad \rightarrow \quad s = a_1 a_2 \underline{A_3} a_4 a_5 a_6 a_7 a_8 a_9 a_{10}$$

Other GA Operators

- 치환(displacement): 염색체의 일부분을 다른 염색체의 일부분으로 대치하는 방법
- 역위(inversion) : 두 개의 역위점을 구하여 그 사이의 비트들의 순서를 반대로 바꾼다

$$p = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} \quad \rightarrow \quad s = a_1 a_2 a_3 \underline{a_8 a_7 a_6 a_5 a_4} a_9 a_{10}$$

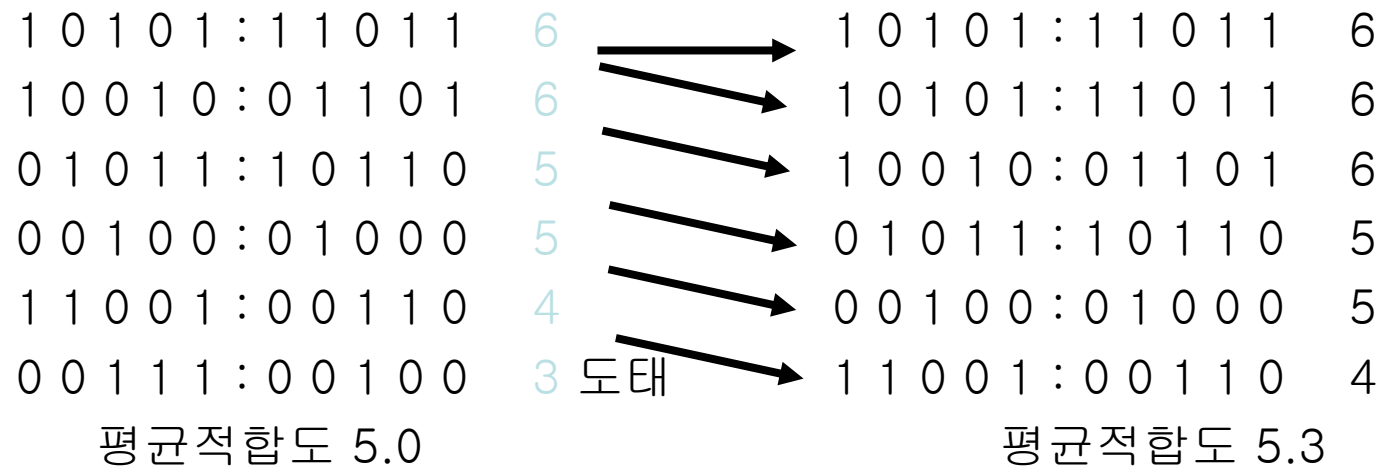
- 중복(duplication) :염색체상의 코드의 일부분을 중복시킨다.
- 추가(addition) :염색체상에 어떤 길이의 부분 문자열을 삽입시킨다. 이 결과 염색체의 길이가 길어지게 된다.
- 제거(deletion) : 염색체상에 어떤 길이의 부분 문자열을 제거시킨다. 이 결과 염색체의 길이가 짧아지게 된다.

GA의 장단점, 응용분야

- 장점
  - 복수개의 개체사이의 상호협력에 의한 해의 탐색이 가능함
  - 번거로운 미분연산이 불필요함
- 단점
  - 대상으로 하는 문제를 GA로 표현하는 방법상의 어려움발생
  - 개체수, 선택 메카니즘, 교배법 결정, 돌연변이 비율 등 선택해야 할 파라메터 수가 많다.
- 응용분야
  - 최적화, 자동 프로그래밍, 기계학습, 경제학, 면역체계, 생태학, 집단유전학, 진화와 학습, 사회적 시스템 등 많은 과학 및 공학 문제의 모델링에 사용되어 왔으며 앞으로도 사용 가능함

GA 예제

- 적합도 : 전반부는 0의 수, 후반부는 1의 수
- (적합도) (적합도)



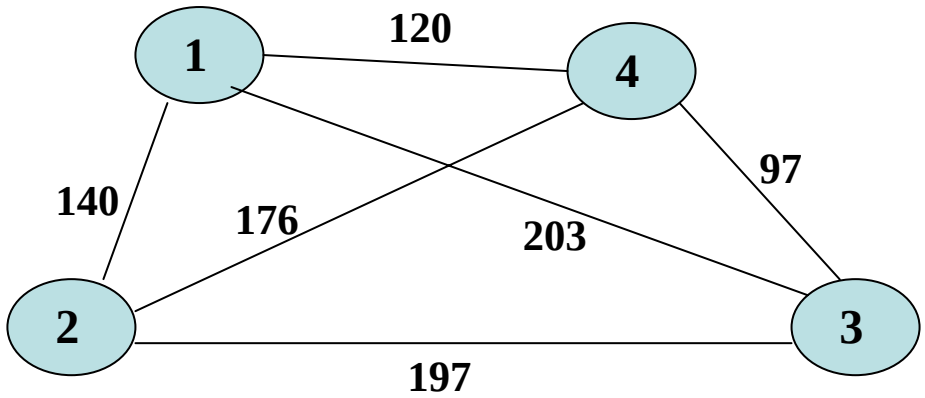
평균적합도 5.0 평균적합도 5.3

순위 1과 6, 2와 5, 3과 4를 교배 → 도태 → 교배 → 도태 ...

책 211 – 213

GA Example:TSP problem

- 정의:
  - n개의 도시와 도시 사이의 거리가 주어졌을 때, 어떤 도시에서 시작하여 모든 도시를 단 한번만 방문하고 원래의 출발지로 되돌아오는 최단 길이의 여행을 찾는 문제
  - 도시들의 가능한 방문의 모든 순열이 주어질 때 각 도시와 다음 도시와의 Euclidean distance의 합이 최소가 되는 여행을 선택하는 것
  - 탐색공간 = $\{T_1, T_2, \ldots, T_n!\}$= n!(크기)



1: 대전
2: 광주
3: 부산
4: 서울

<4개 도시사이의 직선거리>

TSP

T_1 = 1-2-3-4 D_1=555

T_2 = 1-2-4-3 D_2= 617

.....

T_{24} = 4-3-2-1 D_{24} = 555

- 최단경로는 T_1=T_6=T_8=T_{10}=T_{15}=T_{17}=T_{19}=T_{24}=555
- Euclidian Distance

$$D = \sum_{j=1}^{n} \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}$$

책 215-217

단순한 함수의 최적화

$$f(x) = x \cdot \sin(10\pi x) + 1.0$$

- 구간 [-1,2]에서 함수 f의 최대값 구하기

$$f'(x) = \sin(10\pi x) + 10\pi x \cos(10\pi x) = 0$$

$$\tan(10\pi x) = -10\pi x$$

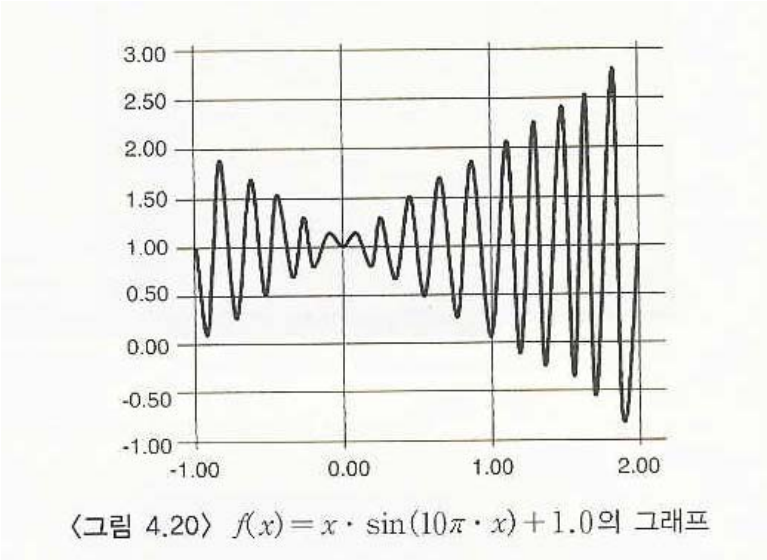
$$x_i = \frac{2i-1}{20} + \varepsilon_i$$

$$x_0 = 0$$

$$x_i = \frac{2i+1}{20} - \varepsilon_i$$

$$x_{19} = \frac{37}{20} + \varepsilon_{19} = 1.85 + \varepsilon_{19}$$

$$f(x_{19}) \cong 2.85$$



〈그림 4.20〉 $f(x) = x \cdot \sin(10\pi \cdot x) + 1.0$의 그래프

(1) 표현방법

- 이진스트링 사용, 정밀도는 소수점 이하 6번째자리 사용
- 정의 길이 3 → 정의영역을 3x1000000개의 동일한 영역으로 분할
- 이진염색체가 22비트 필요

$$2097152 = 2^{21} < 3000000 \le 2^{22} = 4194304$$

- 염색체 (1000101110110101000111)는 숫자 0.637197

$$x' = (1000101110110101000111)_2 = 2288967$$

$$x = -1.0 + 2288967 \cdot \frac{3}{4194303} = 0.637197$$

- (0000000000000000000000) : -1.0
- (1111111111111111111111) : 2.0

(2) 초기 개체집단 생성

- 22비트의 임의로 초기화된 이진벡터

  v_1 = (1000101110110101000111),

  v_2 = (0000001110000000010000),

  v_3 = (1110000000111111000101)

(3) 평가함수

평가함수eval(x)= f(x)

x_1= 0.63719 → eval(x_1) = f(x_1) = 1.586345

x_2= -0.958973 → eval(x_2) = f(x_2) = 0.078878

x_3= 1.627888 → eval(x_3) = f(x_3) = 2.250650 ← 좋다

(4) 돌연변이

v_3 = (111000000011111111000101) ➔ (111010000011111111000101)

x'_3= 1.721638 → eval(x'_3) = f(x'_3) = −0.082257 ← 나빠짐

v_3 ➔ (111000000111111111000101)

x''_3= 1.630818 → eval(x''_3) = f(x''_3) = 2.3443555 ← 좋아짐

(5) 교배 v_2와 v_3, 위치는 5번째 유전자 이후

v_2 = (00000|011100000000010000)➔(00000|000001111111000101)

v_3 = (11100|000001111111000101)➔(11100|011100000000010000)

v'_2= −0.998113 → eval(v'_2) = f(v'_2) = 0.940865 ← 좋아짐

v'_3= 1.666028 → eval(v'_3) = f(v'_3) = 2.459245 ← 좋아짐

(6) 결과

150세대 후, v=(111001101000100000101), x=1.850773, f(x)=2.85

진화적 전략(ES)

- 1963년 독일 베를린대학의 학생. 실험 중에 최적화기법이 잘 적용이 되지 않자, 형상 파라메타를 돌연변이와 같이 임의로 변화시킬 생각
- 1977년 Schwefel, 목적함수를 해석적 형태로 만들 수 없었던 문제의 해결을 위하여 고안.

- 초기 : (1+1) = 하나의 부모 개체 → 표준정규분포를 이용한 돌연변이 → 대체 or 도태 → ...
  - 1/5만 성공
- 다수개체로 구성된 집단 개념을 도입함으로써 랜덤 교배, 돌연변이와 선택이라는 진화 원리를 사용할 수 있도록 한 전략 : $(\mu+\lambda)$
  - μ : 부모 개체수, λ : 자식 개체수

진화적 전략(ES)

- 진화적 전략은 개체의 평가방법 외에는 문제에 대한 정보를 거의 필요로 하지 않기 때문에 거의 모든 종류의 최적화 문제에 적용할 수 있음
- 선형 및 비선형 제약조건을 갖는 고차원, multi-modal의 비선형 문제를 해결할 수 있다.

- 진화적 전략은 무성생식에서의 유기체 발달을 시뮬레이션.
- 돌연변이와 선택만을 사용

진화적 전략(ES)

- 진화적 전략의 알고리즘

  [1단계 : 초기화]

  부모벡터의 초기화된 개체집단 $X_i(i=1, \ldots, \mu)$는 실행범위로부터 랜덤하게 선택된다.

  [2단계 : 돌연변이]

  자손벡터 $X_j(j=1, \ldots, \mu)$는 각각의 X_i의 구성에 대한 Gaussian의 랜덤인 변화를 기하는 것에 의해 각각의 부모 X_i로부터 만든다.

  [3단계 : 평가]

  X_i와 X_j의 적합도를 평가한다

  [4단계 : 선택]

  다음세대의 새로운 부모를 그들의 적응도로 기초한 개체집단으로부터 선택한다.

  [5단계 : 반복]

  단계2)로부터 (단계4)까지를 해가 수렴할 때까지 반복한다

진화적 프로그래밍(EP)

- 유전자 연산자로써 교배연산을 사용하지 않음.
- 돌연변이의 강도는 최적해에 가까울수록 감소 ← 최적해를 모르므로 강도를 줄일 수 없다. –대안→ 메타 진화 프로그래밍

- 메타 진화 프로그래밍 : 돌연변이에 사용되는 표준편차 값을 돌연변이 시킴. 해에 대한 돌연변이 강도를 조절하는 표준편차를 해와 함께 진화 시킴. → 진화과정에서 해에 대한 돌연변이 강도를 스스로 조정

- 확률적인 토너멘트

진화적 프로그래밍(EP)

진화적 프로그래밍의 알고리즘

[단계 1] 초기 집단을 임의의 값으로 초기화한다. 집단에서 개체 (해)의 숫자는 최적화 속도에 큰 영향을 주지만, 집단 크기를 얼마로 해야 적당한지에 대한 답은 없다.

[단계 2] 개체는 새로운 집단으로 복제되고, 이 개체가 돌연변이 된다. 돌연변이의 강도는 부모 개체에 할당된 변화 요구에 의존한다.

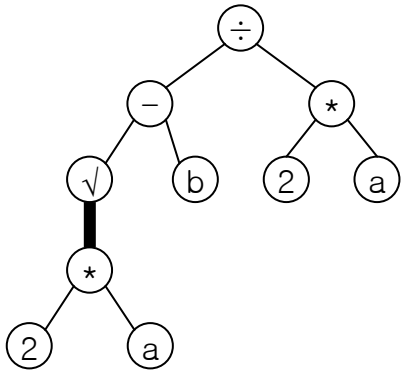
- 메타-진화프로그래밍에서는 목적변수의 돌연변이에 앞서 표준편차를 먼저 돌연변이 해야 한다.
- 돌연변이 연산자 : 표준 정규분포 돌연변이, 로그정규분포 돌연변이, 평균 돌연변이, 자기-적응성을 갖는 평균 돌연변이

[단계 3] 자식 개체는 적합도 평가를 받고, 확률적인 승자승 선택을 사용하여 다음 세대 집단을 구성한다

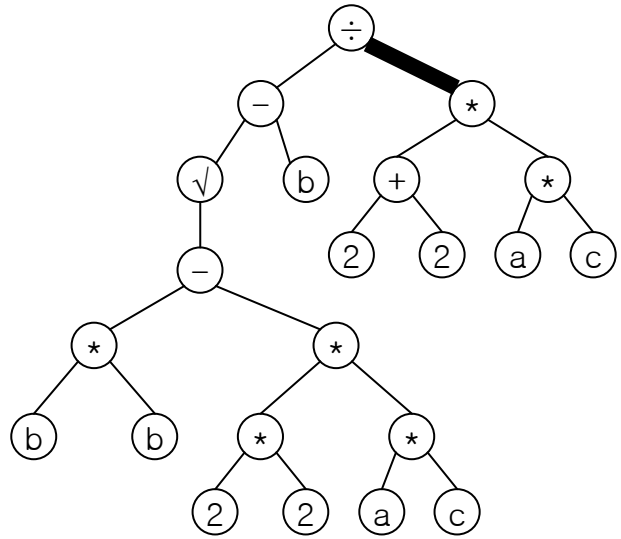
유전자 프로그래밍(GP)

- 컴퓨터가 인간에 의해 프로그램 되지 않은 상태에서 스스로 문제해결을 위한 프로그램을 자동 생성할 수 있도록 한 프로그래밍 전략
- 유전자 알고리즘에 근본을 두고있으나, 차이점은 유전자 프로그래밍이 프로그램으로 해석되는 동적인 트리 구조를 진화시키는데 있다. 즉, 유전자 알고리즘의 출력이 정량적인데 비해 유전자 프로그래밍의 출력은 컴퓨터 프로그램이다.
- GP를 정의하려면 프로그램을 구성할 원시함수와 터미널 집합의 정의가 필요.
  - 단말 노드는 심볼, 내부 노드는 함수
    - 단말노드 : 프로그램의 변수와 상수
    - 함수 : 프로그램의 기능을 구현, 산술연산, 수학연산, 논리연산, 문제에 따른 특수 함수

  - 교배 연산 : 하위 트리를 서로 교환 → 자식 개체는 부모와 다른 구조를 갖게 됨.
  - 돌연변이 : 노드를 변이, 트리구조를 다른 트리구조로 변이
- 프로그램은 모듈화되어야
- 가장 구현하기 어렵고 중요한 개념은 적합도 함수이다.
  - 예) 사격
- GP에서 가장 중요한 연산자는 교배연산이다.(예제 1, 2, 3 참조)
- GP활용분야
  - 최적해 분야, 자동 컴퓨터 프로그래밍 분야(로봇, 제어 등), 모델링, 예측, 데이터마이닝 등...

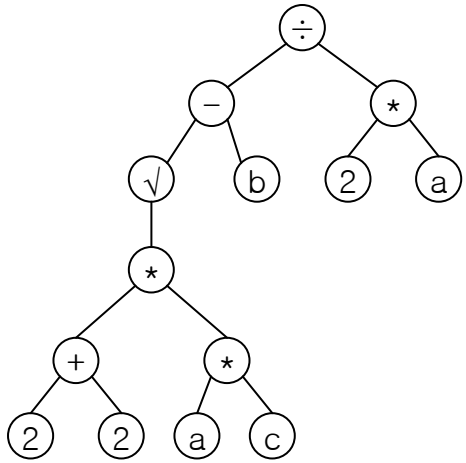


(÷(−√(*2a)b)(*2a))

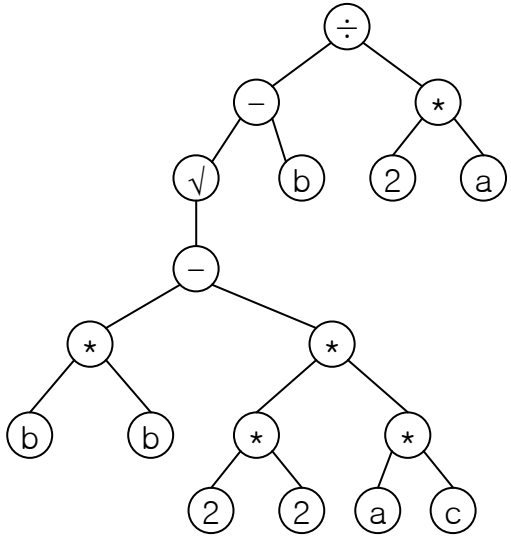


(÷(−√(−(*bb)(*(*22)(*ac)))b)(*2a))

(가) 서로 다른 크기와 구조를 갖는 부모 개체

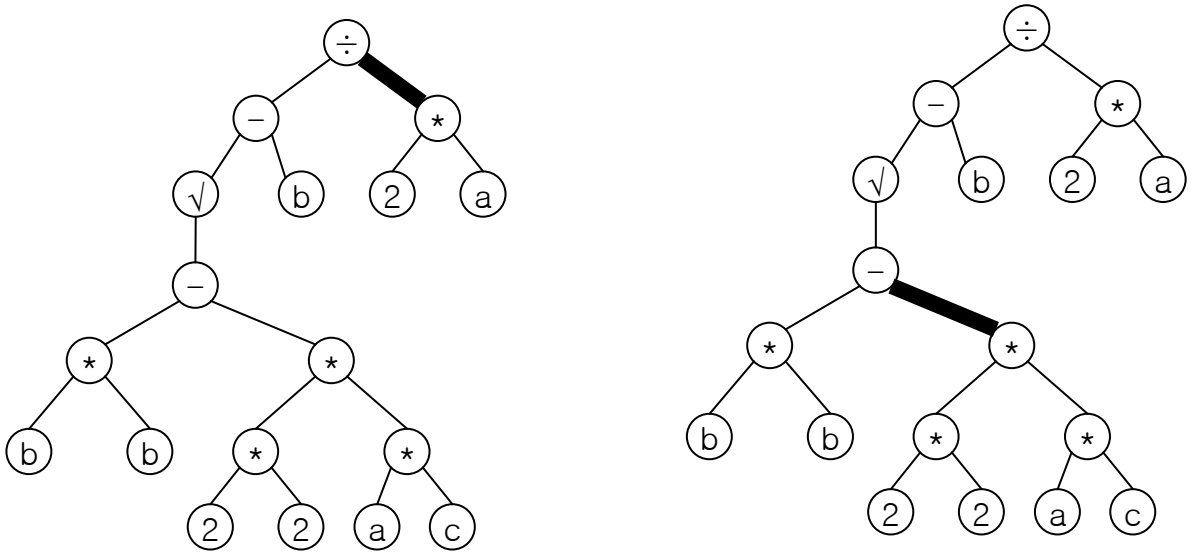


(÷(−√(*(*22)(*ac))b)(*2a))

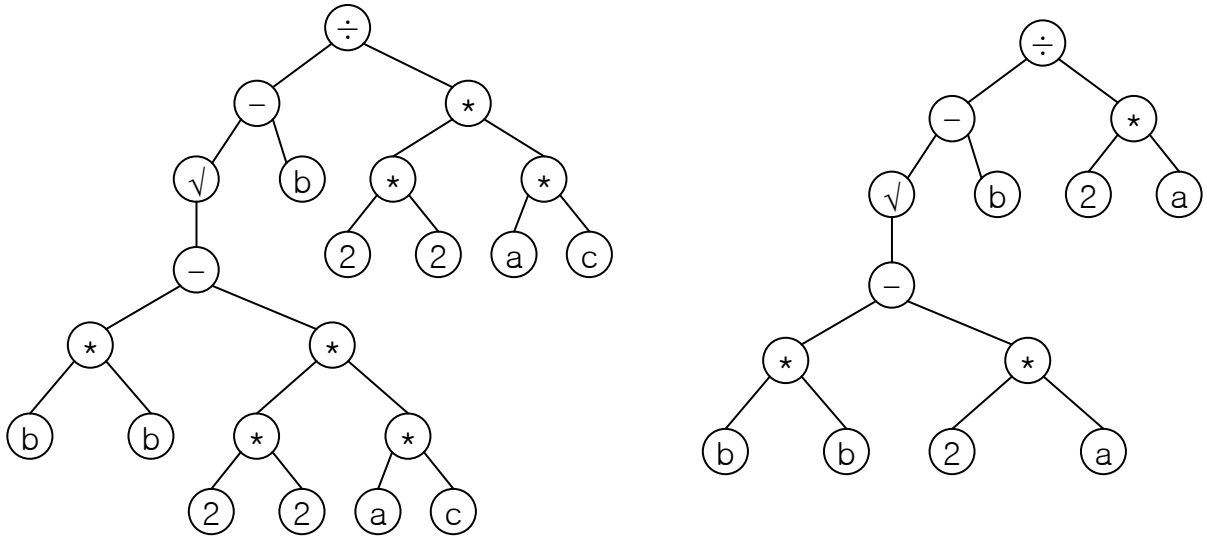


(÷(−√(−(*bb)(*(*22)(*ac)))b)(*2a))

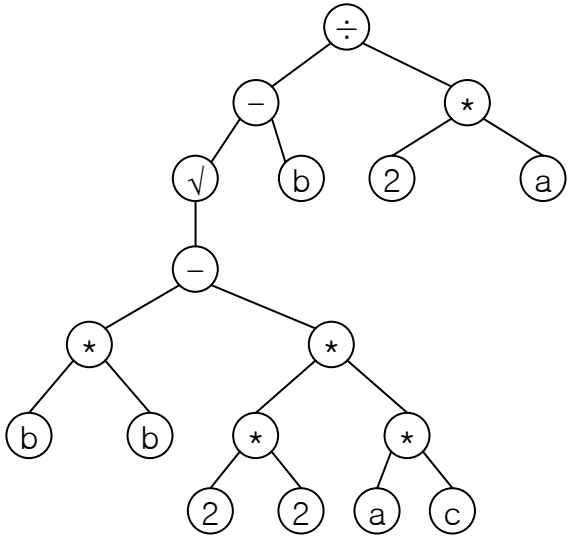
(나) 교배에 의해 생성된 자식 개체



(가) 동일한 부모 개체

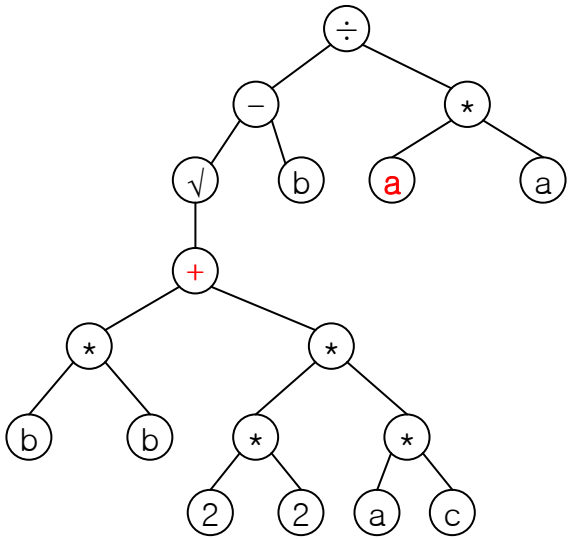


(나) 교배에 의해 발생된 자식 개체

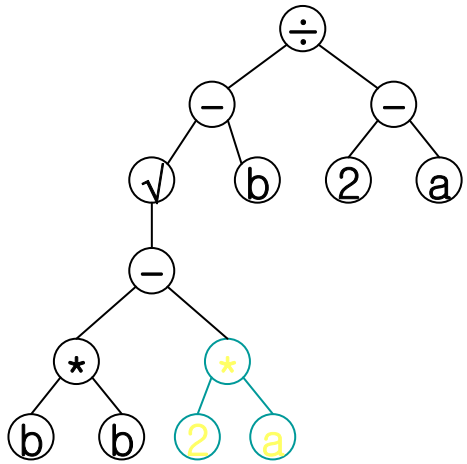


(÷(−√(−(*bb)(*(*22)(*ac)))b)(*(*2a))

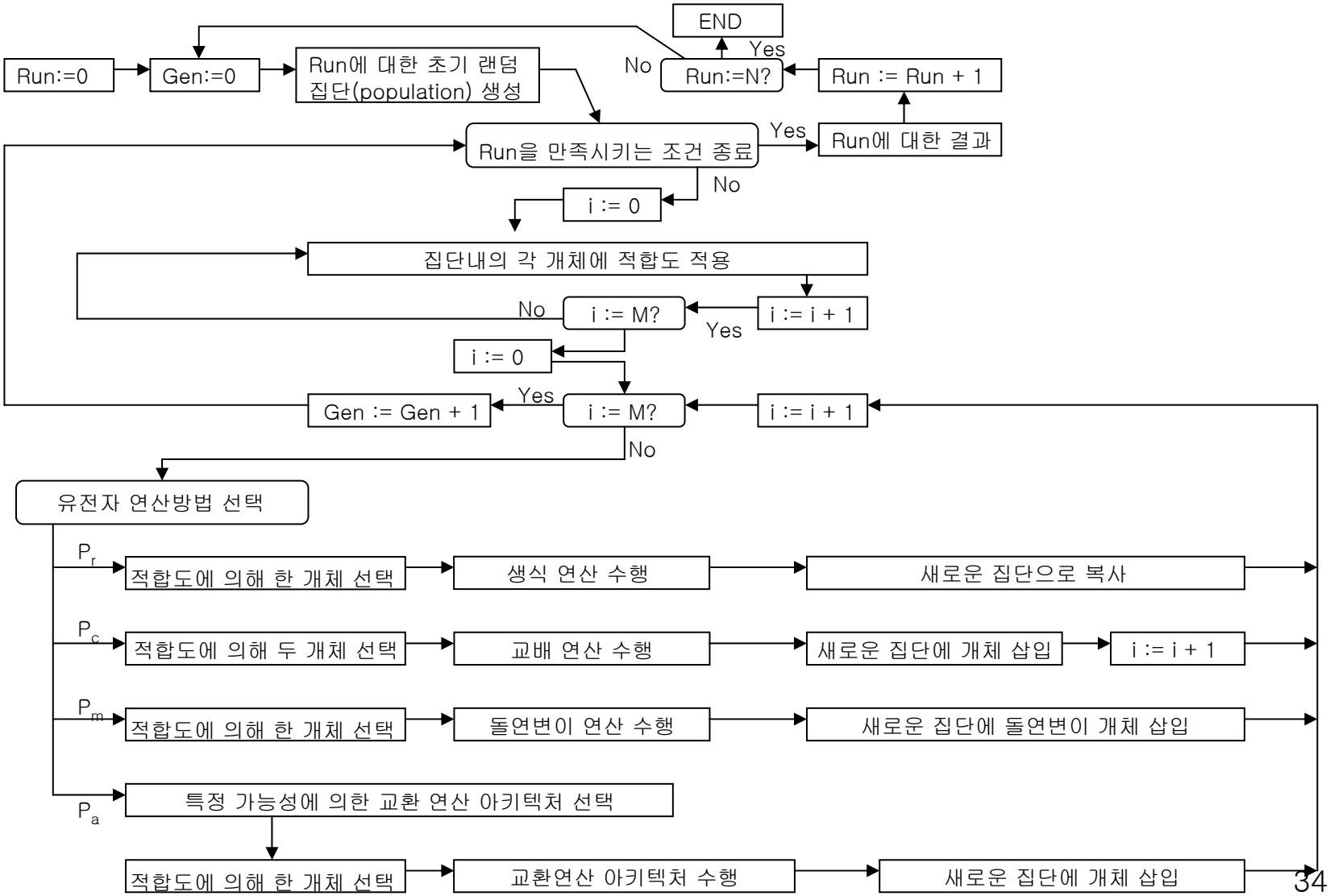
(가) 원본 개체



(÷(−√(+(*bb)(*(*22)))b)(*aa))



(÷(−√(−(*bb)(*(*2a))b)(*2a))

(나) 돌연변이 된 개체

유전자 프로그래밍 문제풀이과정