유전자 알고리즘 (Genetic Algorithm)

[목 차]

- Concept of Genetic Algorithm
- Terminology
- Genetic Operator
- Examples of Simple Genetic Algorithms
- Application for GA

Principal Heuristic Algorithms

- **Genetic Algorithms (Holland – 1975) : Today's issue**
  - √ Inspired by genetics and natural selection
- Simulated Annealing (Kirkpatrick – 1983)
  - √ Inspired by molecular dynamics – energy minimization
- Particle Swarm Optimization (Eberhart and Kennedy - 1995)
  - √ Inspired by the social behavior of swarms of insects or flocks of birds

These techniques all use a combination of randomness and heuristic "rules" to guide the search for global maxima or minima

What is a Heuristic?

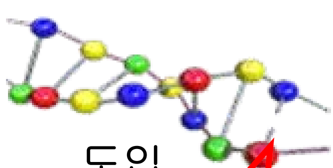
- A Heuristic is simply a rule of thumb that hopefully will find a good answer.
- Why use a Heuristic?
  - √ Heuristics are typically used to solve complex (large, nonlinear, nonconvex (ie. contain many local minima)) multivariate combinatorial optimization problems that are difficult to solve to optimality.
- Unlike gradient-based methods in a convex design space, heuristics are NOT guaranteed to find the true global optimal solution in a single objective problem, but should find many good solutions (the mathematician's answer vs. the engineer's answer)
- Heuristics are good at dealing with local optima without getting stuck in them while searching for the global optimum.

1) Concept of Genetic Algorithm

Concept of Genetic Algorithm (1)

- **Genetic Algorithm**
  - √ 생물학적 진화이론과 유전학에 기반
  - √ 우수한 형질의 개체가 자연계에 잘 적응하여 우수한 후손을 생성한다는 원리 이용
  - √ 전통적인 최적화 알고리즘과는 달리 도함수**(Gradient)**를 이용하지 않음
  - √ 이진수의 조합으로 구성된 개체**(individual 혹은 염색체)**들의 집단**(population)**을 가지고 선택**(selection)**

- **Genetic Algorithm**의 도입



Objective

대규모 조합의 최적화 문제
제한조건이 많은 이산화문제의 최적화

도입

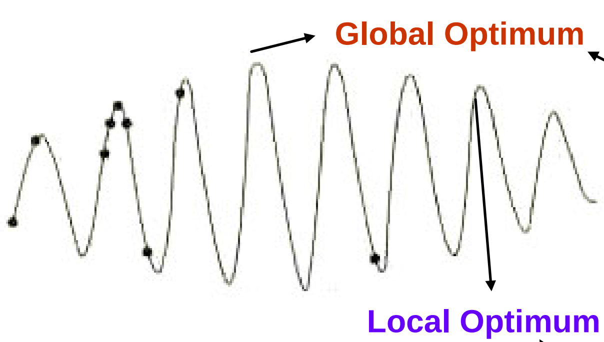
Holland (1975)
"Adaptation in Natural and Artificial Systems"

적용

Goldberg & Santani (1989)
: GA의 구조물설계적용

Concept of Genetic Algorithm (2)

- **Genetic Algorithm의 특징 및 장점**
  - √ 일반적인 공학적인 최적화 문제의 경우 그 현상이 비선형적인 거동을 보이는 경우가 많기 때문에 전통적인 함수의 구배를 이용하는 탐색기법에서는 **Local Optimum**에 도달하는 경우가 많이 존재함
  - √ **Genetic Algorithm**의 경우**,** 구배정보를 이용하지 않으며 주어진 설계 공간 전역에 대한 탐색을 수행하기 때문에 **Global Optimum**을 도출할 수 있는 확률이 구배법에 비해 많으며 초기치에 의존하는 경우가 없음
  - √ 특히 최적해는 알지 못하지만 평가는 할 수 있는 **Black Box**형태의 공학적 문제에서는 대단히 유용하게 적용될 수 있다**.**
  - √ **Ex.** **Maximize** $F(x)=-\cos(x)*\cos(x/20)$



*** GA Results**

Evaluation Count: 560

;Best Ref. Val = 0.987719237804

X=-3.133696794510

*** GBM Results**

Initial Point(12.)

Function Call 22

;F(x) = 0. 891295

X= 9.39938

*** GBM Results**

Initial Point(-4.)

Function Call 17

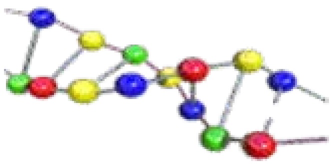
;F(x) =0. 987719

X= -3.13369

Concept of Genetic Algorithm (3)

- 분류

Evolutionary Algorithm
(진화 알고리즘)



Genetic Algorithm (GA)
- 가장 대중적으로 알려짐

Evolutionary Programming (EP)
(GA+Data Structure)

Evolution Strategies

Terminology

유전자좌(**Locus**)
: 염색체 상에서 유전자의 위치

염색체(**Chromosome**)
:**2**진수로 표현된 **String**

1 1 0 0 0 0 1 1

유전자형
(Genotype)

유전자(**Genes, Bit) : 2**진수로 표현된 값

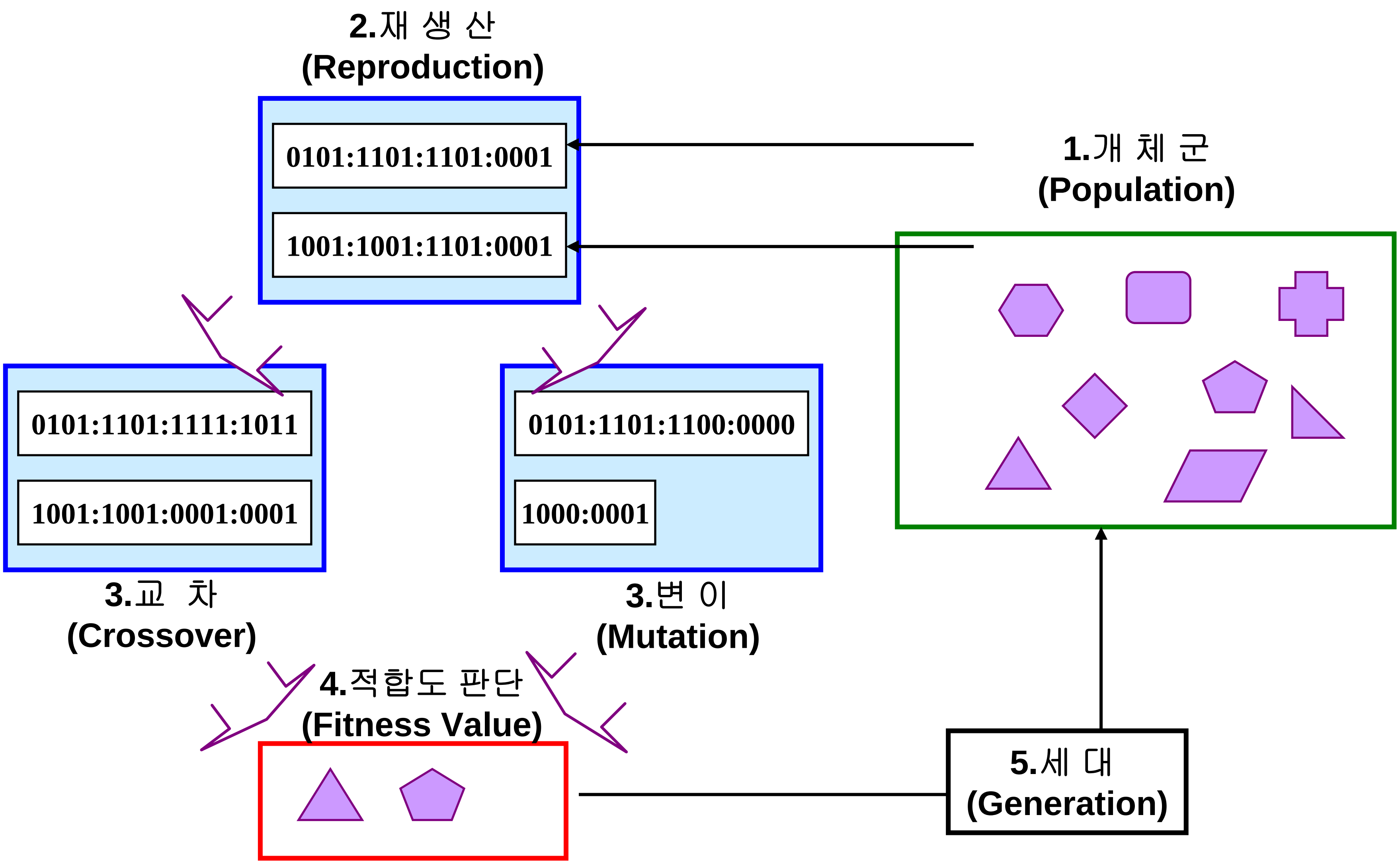
Decoding

Encoding

Solutions

표현형
(Phenotype)

Genetic Optimizer Terminology



Structure of Genetic Algorithm

```
Procedure GA
Begin
    t→0
    initialize P(t)
    evaluate P(t)
    while (not termination condition) do
        recombine P(t) to yield C(t)    // selection, crossover,
  mutation
        evaluate C(t)
        select P(t+1) from P(t) and C(t)
        t←t+1
     end
End
```

Encoding – Decoding

Genotype(유전자형)

Coded Domain

Phenotype(표현형)

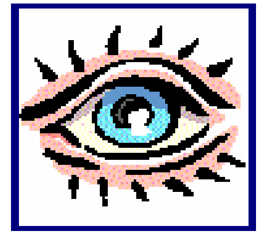
Decision Domain

Biology

UGCAACCGU
("DNA" blocks)

Expression →

← *Sequencing*



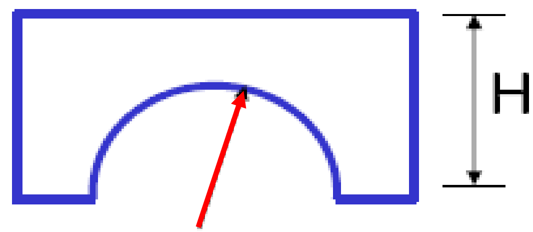
"Blue Eyes"

Design

10010011110
(chromosome)

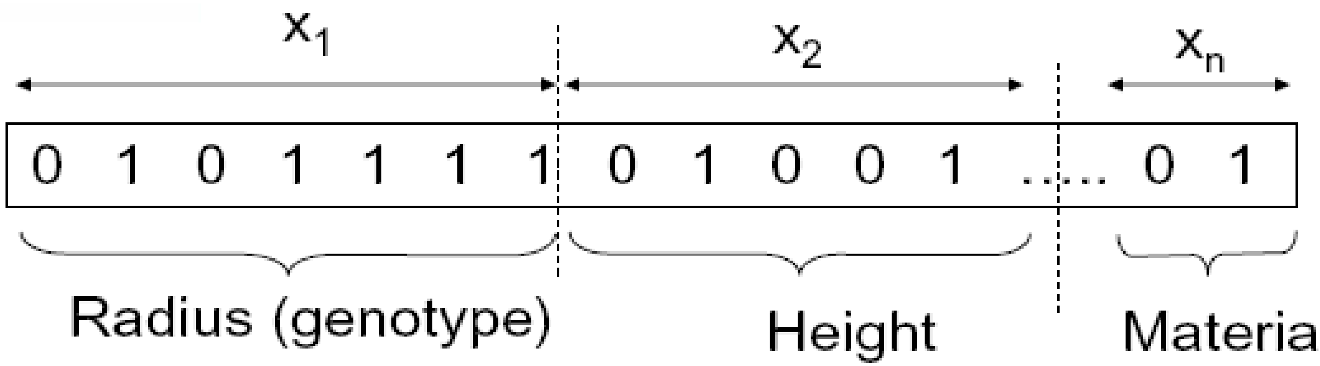
Decoding →

← *Encoding*



Radius R= 2.57[m]

Decoding



- **E.g. binary encoding of integers:**

$\boxed{10100011}$ ⇒ **Genotype(유전자형)**

$(1*2^7+0*2^6+1*2^5+0*2^4+0*2^3+0*2^2+1*2^1+1*2^0)$

$128 + 0 + 32 + 0 + 0 + 0 + 2 + 1 = \boxed{163}$

⇓

Phenotype(표현형)

Binary Encoding Issues

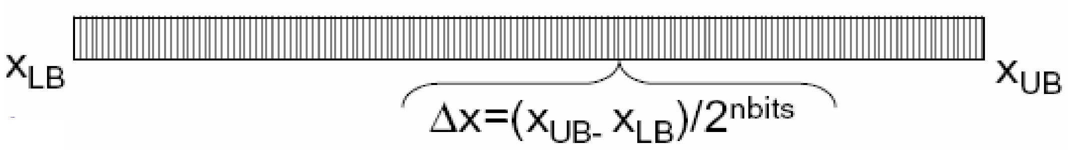
- Number of bits dedicated to a particular design variable is very important.

Number of bits needed:

$$nbits = \left\lceil \frac{\ln\left(\frac{x_{UB} - x_{LB}}{\Delta x}\right)}{\ln 2} \right\rceil$$

- Resolution depends on:
  - √ upper and lower bounds x_{LB}, x_{UB}
  - √ number of bits

$x \in \mathbb{R}$



Example

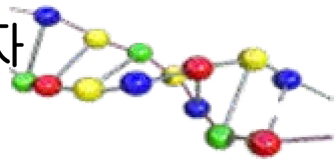
```
[G]=encode(137.56,50,150,8)
G = 1     1     0     1     1     1     1     1
[X]=decode(G,50,150,8);
X = 137.4510
```

So $\Delta x = (150-50)/2^8 = 0.39$

Loss in precision !!!

Genetic Operator

- GA에 적용되는 중요한 연산자

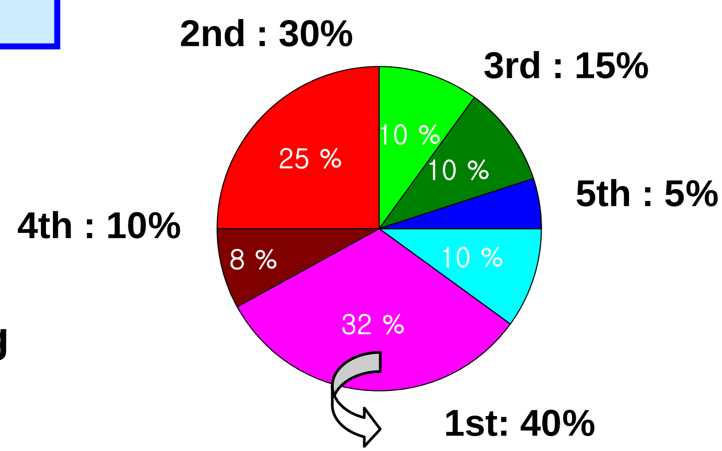


연산자	선 택(Selection)	교 차(Crossover)	변 이(Mutation)
기 능	-다음세대로 전달하기 위한 유전자의 선택	-선택된 염색체들을 조합,다음세대의 염색체를 생성	-유전자를 일정한 확률로 변화시키는 조작
효 과	-다음세대로 높은 적합도를 가지는 유전자의 특징을 전달	-수렴속도 가속화 -높은 최적치를 남길 가능성 부여	-전역적 탐색효과의 극대화
전 략	-적합도 비례전략 -순위전략 -토너먼트 선택전략 -엘리트 보존 전략	-단순 교차 -복수점 교차 -일점 교차	-정적변이 -동적변이

Genetic Operator : Selection (1)

- 선택 (Selection)
  - √ Goal is to select parents for crossover
  - √ Should create a bias towards more fitness
  - √ Must preserve diversity in the population

1. 순위전략(Selection according to RANKING)



• 적합도(fitness)의해 각 개체에 순위를 부여하여 그 순위에 의해 사전에 결정된 확률로 자손을 남김

(Better ranking has a higher probability of being chosen)

• 적합도 와 순위에 의해 부여되는 확률이 차이

This scheme tends to favor the fittest individuals in a population more than the ranking-scheme, faster convergence, but can also be a disadvantage.

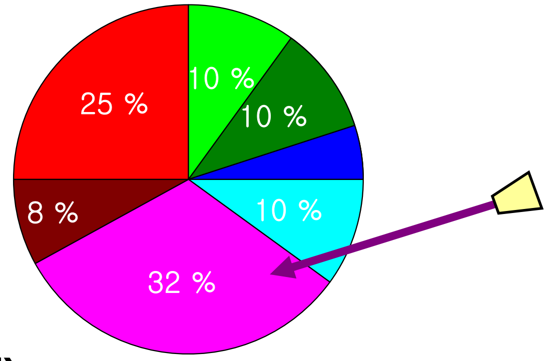
Genetic Operator : Selection (2)

2. 적합도비례전략(Proportional to FITNESS Value Scheme)

- 기본 모델(Classical Selection Model)
- 적합도에 비례 하여 자손을 남김

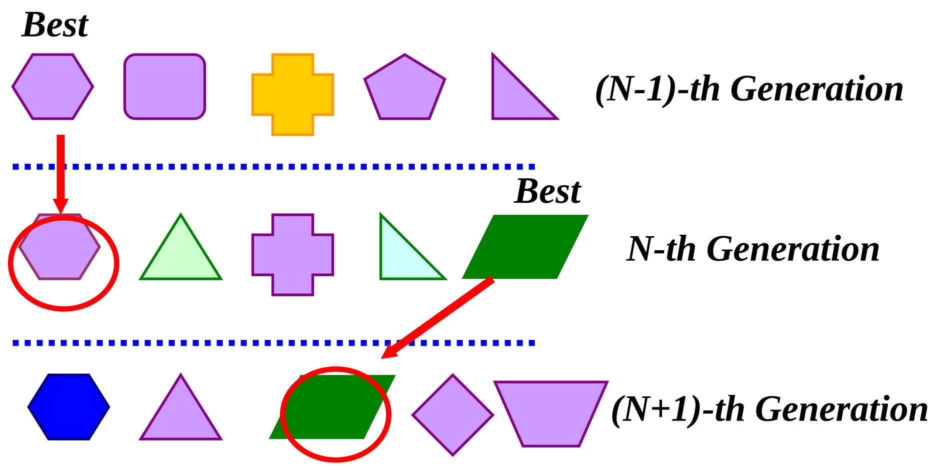
(Better fitness value has a higher probability of being Chosen)

- 룰렛(Roulette Model) , 몬테카를로 모델(Monte Carlo Model)



3. 엘리트 보존 전략

-각 집단 중에서 가장 적합도가 높은 개체를 다음세대로 그대로 넘김



Genetic Operator : Selection (2)

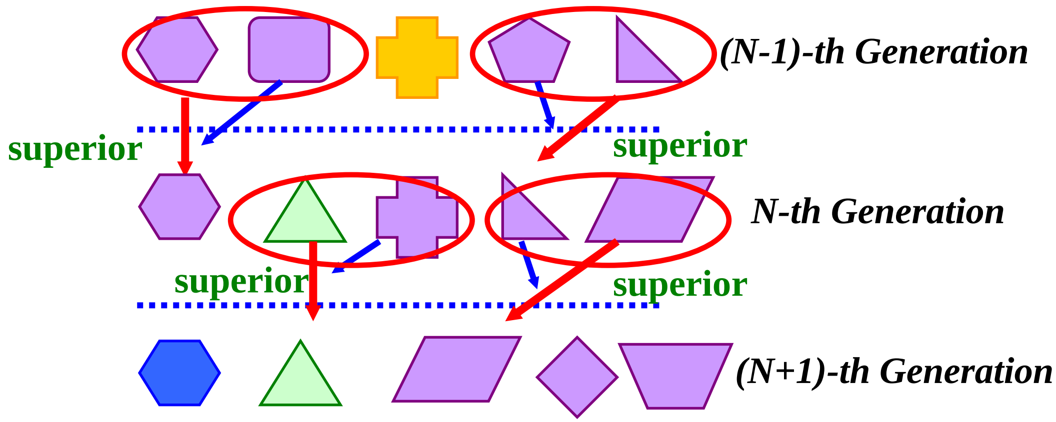
4. 토너먼트 선택전략(Tournament Selection)

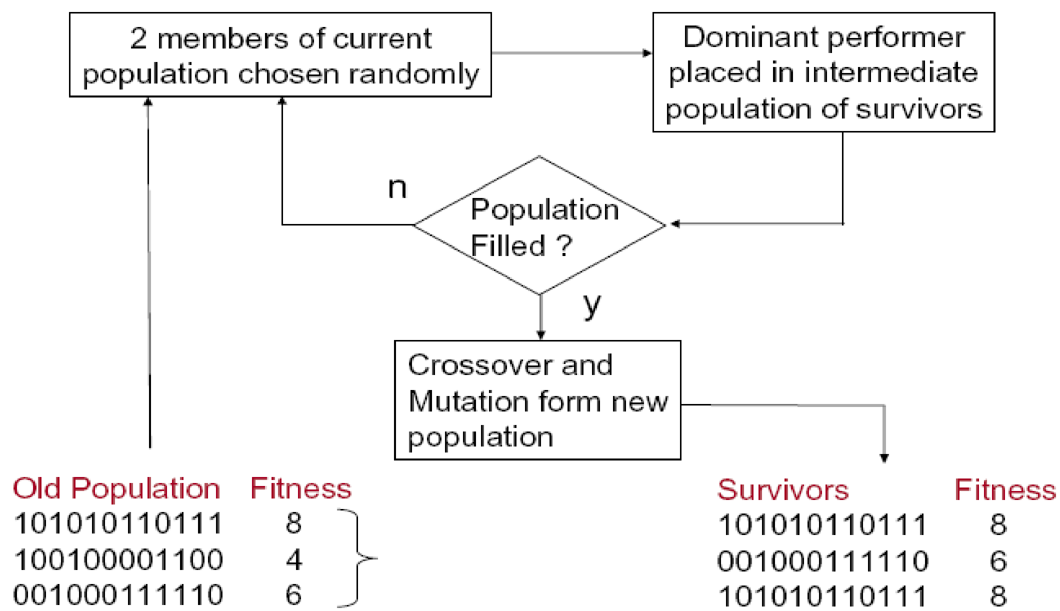
-임의의 수의 개체를 무작위 선택

(2 members of current population Chosen randomly)

-그 가운데 적합도가 높은 개체를 다음 세대로 넘김

(Dominant performer placed intermediate population of survivors)





Genetic Operator : Crossover

- 교차(Crossover)-일반적인 임의탐색 기법과 구별되는 큰 특징

1. 단순교차(Simple Crossover)

-하나의 교차위치 설정,

그 전후로 부모의 유전자형 교환

1 point

			exchange	
개체 A	1 0 0 1 | 1 1 1	→	1 0 0 1	0 0 0
개체 B	0 0 1 1 | 0 0 0		0 0 1 1	1 1 1

2. 복수점교차(Multi-point Crossover)

-교차위치가 복수인 경우

			exchange		
개체 A	1 0 | 0 1 1 | 1 1	→	1 0	1 1 0	1 1
개체 B	0 0 | 1 1 0 | 0 0		0 0	0 1 1	0 0

3. 일정교차(Uniform Crossover)

-마스크를 사용하여 어느쪽의 유전자를 받아 들일지 결정

가령 마스크의 비트가 **'0'**일 경우에는 그대로, **'1'**일 경우에는 두 부모의 유전자를 교환

filtering

개체 A	0 0 1 1 1 1
개체 B	1 1 1 1 0 0
mask	0 1 0 1 0 1
개체 A	0 1 1 1 1 0
개체 B	1 0 1 1 0 1

Genetic Operator : Mutation (1)

- 변이 (혹은 돌연변이, **Mutation)**
  - √ 유전자를 일정한 확률로 변화시키는 조작
  - √ 전역적 탐색효과의 극대화**(Maximize Global Search)**, 수렴 속도 가속**(Increase Convergence Rate)**
  - √ 집단의 다양성 증대 **(Increase Diversification of Population)**

- **Strategy**

1. 정적변이

-돌연변이의 확률을 일정하게 고정

2. 동적변이

-적응변이**(Adaptive Mutation)**

-돌연변이의 확률이 경우에 따라 변화

GAs versus Traditional Methods

Differ from traditional search/optimization methods:

- GAs search a population of points in parallel, not only a single point
- GAs use probabilistic transition rules, not deterministic ones
- GAs work on an encoding of the parameter set rather than the parameter set itself
- GAs do not require derivative information or other auxiliary knowledge - only the objective function and corresponding fitness levels influence search

References

- A. Zalzala, P.J. Fleming, “Genetic Algorithms in Engineering Systems” Control Engineering Series 55, The Institution of Electrical Engineers (IEE), 1997
- Gen, Mitsuo, Cheng, Runwei., “Genetic Algorithms and Engineering Optimization” **,** Wiley, New York , 2000.
- Back, Thomas, “Evolutionary Algorithms in Theory and Practice : Evolution Strategies, Evolutionary Programming, Genetic Algorithms”, Oxford University Press,Oxford, 1996.
- Michalewicz, Z., “Genetic Algorithm + Data Structures = Evolution Programs”, Spinger-Verlag, 1996.

* For Korean Students :

- 조성배 역, “GA의 기초이론, 공학응용 및 인공생명 – 유전자알고리즘”, 대청 컴퓨터월드, 1996년

2) Examples with Simple Genetic Algorithms

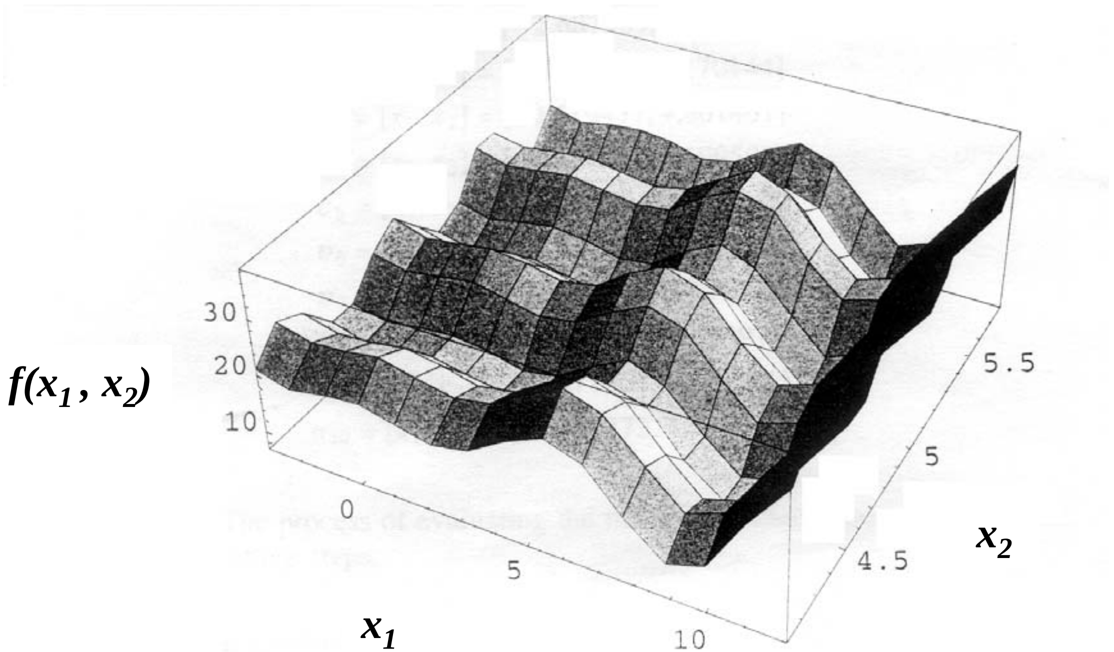
Examples with Simple Genetic Algorithms

- **Optimization Problem**

$$\textit{Maximize} \quad f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$$

$$-3.0 \le x_1 \le 12.1$$

$$4.1 \le x_2 \le 5.8$$



Object Function

Examples with Simple Genetic Algorithms

- Representations
  - √ Encode decision variables into binary strings.
  - √ The precision requirement implies that the range of domain of each variable should be divided into at least $[a_j, b_j]$ (domain of x_j is $(b_j - a_i) \times 10^4$ and the required precision is 4 after the decimal point.) size ranges.
  - √ Required bits(m_j)

$$2^{m_j - 1} < (b_j - a_j) \times 10^4 \le 2^{m_j} - 1$$

  - √ The mapping from a binary string to a real number

$$x_j = a_j + \text{decimal}(\text{substring}_j) \times \frac{b_j - a_j}{2^{m_j} - 1}$$

$$x_1 \qquad (12.1 - (-3.0)) \times 10^4 = 151000$$

$$2^{17} < 151000 \le 2^{18} \qquad m_1 = 18$$

$$x_2 \qquad (5.8 - 4.1) \times 10^4 = 17000$$

$$2^{14} < 17000 \le 2^{15} \qquad m_2 = 15$$

$$m = m_1 + m_2 = 18 + 15 = \underline{33}$$

Examples with Simple Genetic Algorithms

√ **The total length of a chromosome is 33 bits**

v_j 0000010101 00101001 1011110111 11110

	Binary Number	**Decimal Number**
x_1	0000010101 00101001	5417
x_2	1011110111 11110	24318

$$x_1 = -3.0 + 5417 \times \frac{12.1 - (-3.0)}{2^{18} - 1} = -2.687969$$

$$x_2 = 4.1 + 24318 \times \frac{5.8 - 4.1}{2^{15} - 1} = 5.361653$$

Examples with Simple Genetic Algorithms

- **Initial Population**
  - √ **Randomly generated as follows :**

$$v_1 = [000010101001010011011110111011100]$$
$$v_2 = [001110101110011000000010101001000]$$
$$v_3 = [110100100010011011000101000101101]$$
$$v_4 = [000010101001010011011110111000100]$$
$$v_5 = [001110101110011001100010101001000]$$
$$v_6 = [110100100010011011000101000101101]$$
$$v_7 = [000010101001010011011110110111011]$$
$$v_8 = [001110100110011010000010101001000]$$
$$v_9 = [101100100010011010000101000101001]$$
$$v_{10} = [011101100110010010010100110101000]$$

$$v_1 = [x_1, x_2] = [-2.687969,\ 5.361653]$$
$$v_2 = [x_1, x_2] = [0.474101,\ 4.170144]$$
$$v_3 = [x_1, x_2] = [10.419457,\ 4.661461]$$
$$v_4 = [x_1, x_2] = [6.159951,\ 4.109598]$$
$$v_5 = [x_1, x_2] = [-2.301286,\ 4.477282]$$
$$v_6 = [x_1, x_2] = [11.788084,\ 4.174346]$$
$$v_7 = [x_1, x_2] = [9.342067,\ 5.121702]$$
$$v_8 = [x_1, x_2] = [-01330256,\ 4.694977]$$
$$v_9 = [x_1, x_2] = [11.671267,\ 4.873501]$$
$$v_{10} = [x_1, x_2] = [-2.687969,\ 5.361653]$$

Examples with Simple Genetic Algorithms

- Evaluation
  - √ **Step 1.** Convert the chromosome's genetype to its phenotype.

$$x^k = (x_1^k, x_2^k), k = 1,2,,, pop\_size.$$

  - √ **Step 2.** Evaluate the objective function $f(x^k)$.
  - √ **Step 3.** Convert the value of objective function into fitness. For the maximization problem, <u>the fitness is simply equal to the value of objective function</u>

$$\boxed{eval(v_k) = f(x^k), k = 1,2,..., pop\_size.}$$

$$eval(v_1) = f(-2.687969,\ 5.361653) = 19.805119$$

$$eval(v_2) = f(0.474101,\ 4.170144) = 17.370896$$

$$eval(v_3) = f(10.419457,\ 4.661461) = 9.590546$$

$$eval(v_4) = f(6.159951,\ 4.109598) = 29.406122$$

$$eval(v_5) = f(-2.301286,\ 4.477282) = 15.686091$$

$$eval(v_6) = f(11.788084,\ 4.174346) = 11.900541$$

$$eval(v_7) = f(9.342067,\ 5.121702) = 17.958717$$

$$eval(v_8) = f(-01330256,\ 4.694977) = 19.763190$$

$$eval(v_9) = f(11.671267,\ 4.873501) = 15.159724$$

$$eval(v_{10}) = f(-2.687969,\ 5.361653) = 20.264971$$

Examples with Simple Genetic Algorithms

- **Selection**
  - √ **In most practices, <u>a roulette wheel approach</u> is adopted as the selection procedure.**

    1. Calculate the fitness value $eval(v_k)$ for each chromosome v_k :

    $$eval(v_k)=f(x), \quad k=1,2,\ldots,pop\_size$$

    2. Calculate the total fitness for the population:

    $$F=\sum_{k=1}^{pop\_size} eval(v_k)$$

    3. Calculate selection probability p_k for each chromosome v_k :

    $$p_k=\frac{eval(v_k)}{F}, \qquad k=1,2,\ldots,pop\_size$$

    v_k

    4. Calculate cumulative probability q_k for each chromosome v_k :

    $$q_k=\sum_{j=1}^{k} p_j, \qquad k=1,2,\ldots,pop\_size$$

Examples with Simple Genetic Algorithms

- Each time, a single chromosome is selected for a new population in the following way :
  - Step 1. Generate a random number r from the range[0,1].
  - Step 2. If $r \le q_1$, then select the first chromosome v_1 ; otherwise, select the k^{th} chromosome $v_k (2 \le k \le pop\_size)$ such that $q_{k-1} < r \le q_k$
- The probability of a selection p_k for each chromosome v_k (k=1,…,10) is as follows :

  p_1=0.111180, p_2=0.097515, p_3=0.053839

  …
- The cumulative probabilities q_k for each chromosome v_k is as follows :

  q_1=0.111180, q_2=0.208695, q_3=0.262534

  …
- Now we are ready to spin the roulette wheel 10 times. Let us assume that a random sequence of 10 numbers from the range [0,1] is as follows:

  0.301431 0.322062 0.766503

  …

Examples with Simple Genetic Algorithms

√ The first number $r_1 = 0.301431$ is greater than q_3 and smaller than q_4, meaning that the chromosome v_4 is selected for the new population; and so on. The new population consists of the following chromosomes :

$$\mathbf{v}'_1 = [100110110100101101000000010111001] \qquad (\mathbf{v}_4)$$

$$\mathbf{v}'_2 = [100110110100101101000000010111001] \qquad (\mathbf{v}_4)$$

$$\mathbf{v}'_3 = [001011010100001100010110011001100] \qquad (\mathbf{v}_8)$$

$$\vdots$$

Examples with Simple Genetic Algorithms

- Crossover
  - √ **One-cut-point method** : randomly selects one cut-point and exchanges the right parts of two parents to generate offspring.
  - √ Consider two chromosomes as follows, and the cut-point is randomly selected after the 17^{th} gene:

$$\mathbf{v}_1 = [1001101101\,001011\boxed{0100\,0000010111\,001}]$$
$$\mathbf{v}_2 = [0010110101\,000011\boxed{0001\,0110011001\,100}]$$

  - √ The resulting offspring by exchanging the right parts of their parents would be as follows:

$$\mathbf{v}_1' = [1001101101001011\boxed{00010110011001100}]$$
$$\mathbf{v}_2' = [0010110101000011\boxed{01000000010111001}]$$

  - √ The probability of crossover is set as $p_c = 0.25$ (25% of chromosomes undergo crossover.).

Examples with Simple Genetic Algorithms

- **Procedure : Crossover**

begin
  $k \leftarrow 0;$
  while $(k \le 10)$ **do**
    $r_k \leftarrow$ **random number from** $[0,1];$
    if $(r_k < 0.25)$ **then**
      select v_k **as one parent for crossover;**
    end
    $k \leftarrow k+1;$
  end
end

Assume that the sequence of random number is

0.625721	0.266823	0.288644
0.295114	0.163274	0.567461
0.085940	0.392865	0.770714
0.548656		

This means that the chromosomes v'_5 and v'_7 were selected for crossover.

- **Generate a random integer number pos from the range [1, 32]**

$$v'_5 = [10011011010010110100000000010111001]$$
$$v'_7 = [00011011010010110100000000010111001]$$

$$v'_5 = [10011011010010110100000000010111001]$$
$$v'_7 = [00011011010010110100000000010111001]$$

Examples with Simple Genetic Algorithms

- **Mutation**
  - √ Alters one or more genes with a probability equal to the mutation rate.
  - √ Assume that the 18th gene of the chromosome v_1 is selected for a mutation. Since the gene is 1, it would be flipped into 0.

$$v_1 = [100110110100101101000000010111001]$$

$$v_1' = [100110110100101100000000010111001]$$

The probability of mutation is set as $p_m = 0.01$, so 1% of total bit of population would undergo mutation.

  - √ Need to generate a sequence of random numbers r_k (k=1,…,330) from the range[0,1].

Bit_pos	*chrom_num*	*bit_no*	*random_num*
105	4	6	0.009857
164	5	32	0.003113
199	7	1	0.000946
329	10	32	0.001282

Examples with Simple Genetic Algorithms

- **Final population**

$$v'_1 = [100110110100101101000000010111001]$$
$$v'_2 = [100110110100101101000000010111001]$$
$$v'_3 = [001011010100001100010110011001100]$$

⋮

Corresponding decimal values of variables $[x_1, x_2]$

$$f(6.159951, 4.109598) = 29.406122$$
$$f(6.159951, 4.109598) = 29.406122$$
$$f(-0.330256, 4.694977) = 19.763190$$

⋮

√ **Just completed one iteration of genetic algorithm.**

The best chromosome in the 419th generation as follows:

$$v^* = (111110000000111000111101001010110)$$
$$\mathrm{eval}(v^*) = f(11.631407, 5.724824) = 38.818208$$
$$x_1^* = 11.631407$$
$$x_2^* = 5.724824$$
$$f(x_1^*, x_2^*) = \underline{38.818208}$$

3) Application of GA

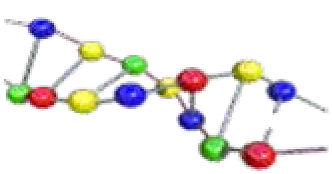
[목 차]

- High Bypass Ratio Jet Engine Design

Application of GA (1)

High Bypass Ratio Jet Engine Design

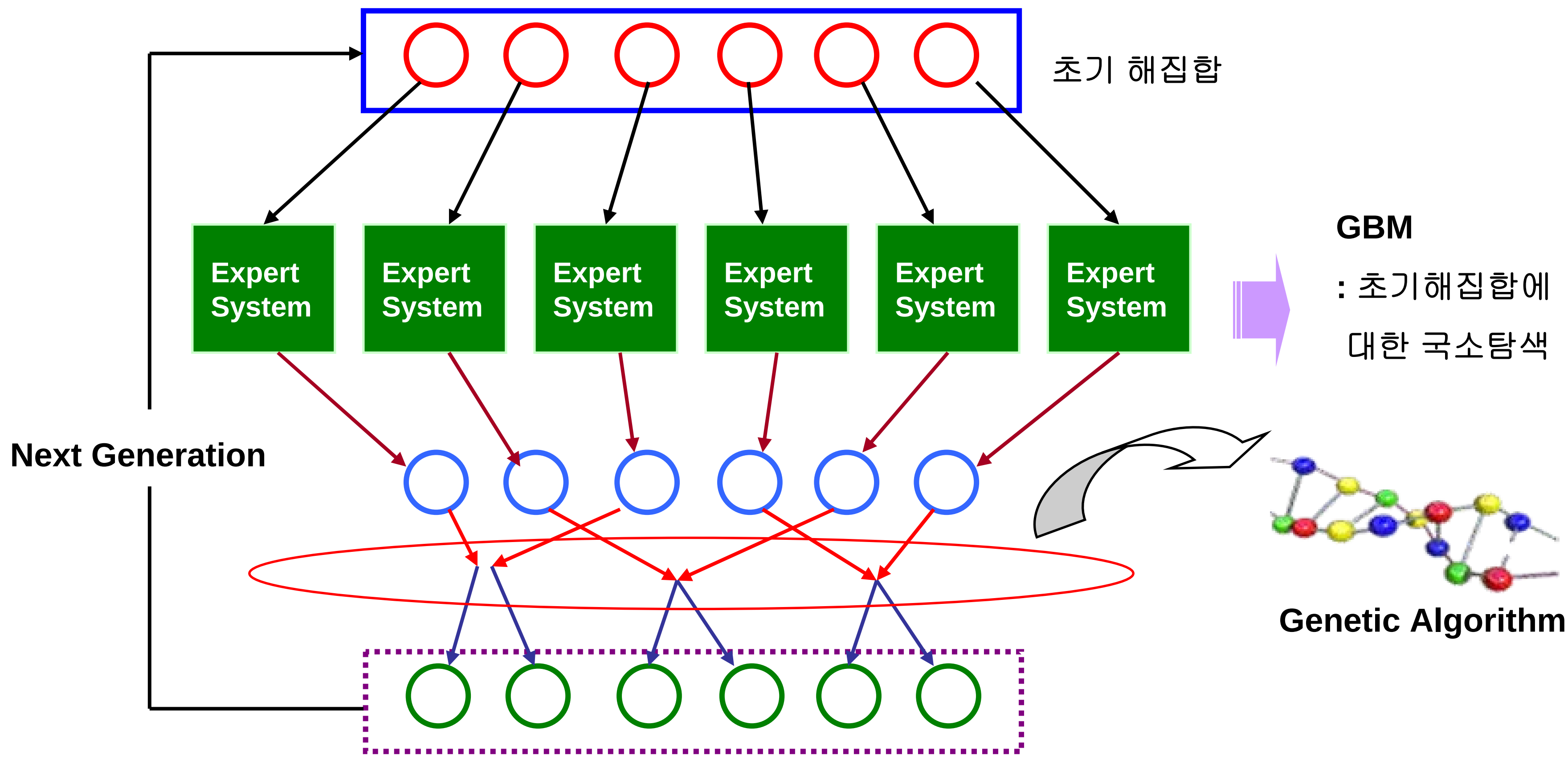
- √ **Powell 등, " EnGENEous : Domain independent, machine learning for design optimization," *Proc. Of ICGA-89*,1989**
- √ 이 문제의 경우, 가능한 해집합의 수는 **10**의 **387**승 개에 이름
- √ 통상의 방법으로는 두 명의 설계자가 한 달 이상의 작업을 요하는 문제



Genetic Algorithm + Expert System → EnGENEous

Application of GA (2)

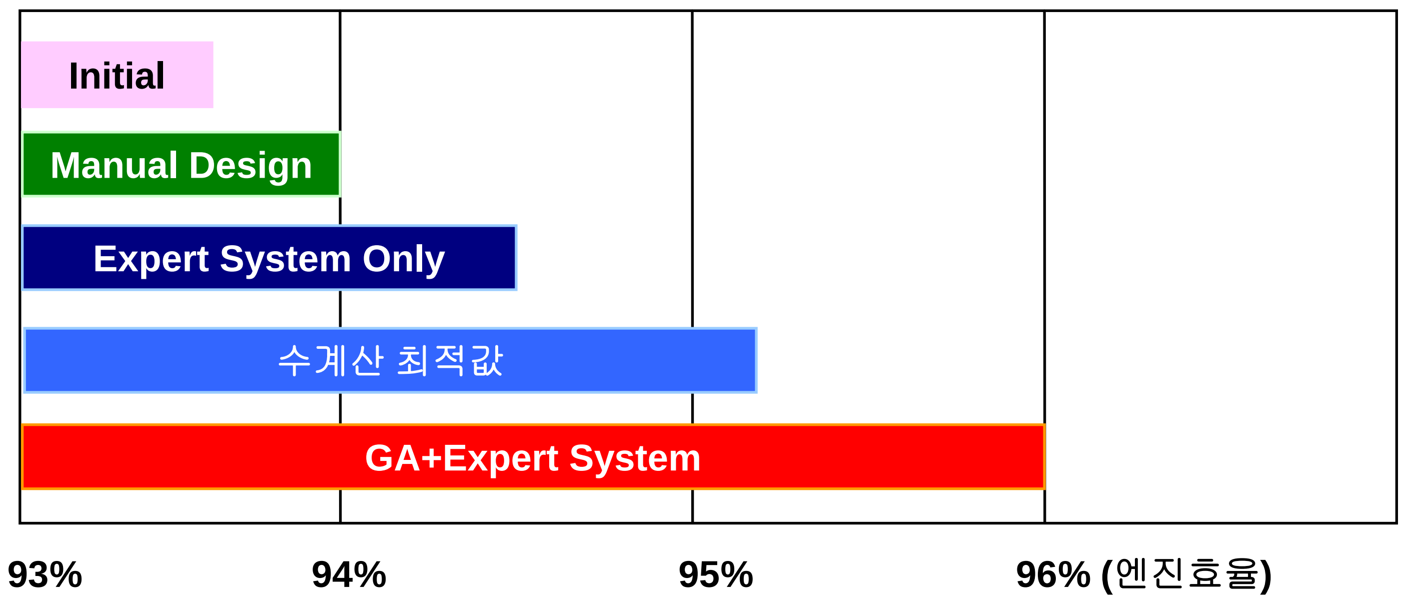
EnGENEous System



Application of GA (3)

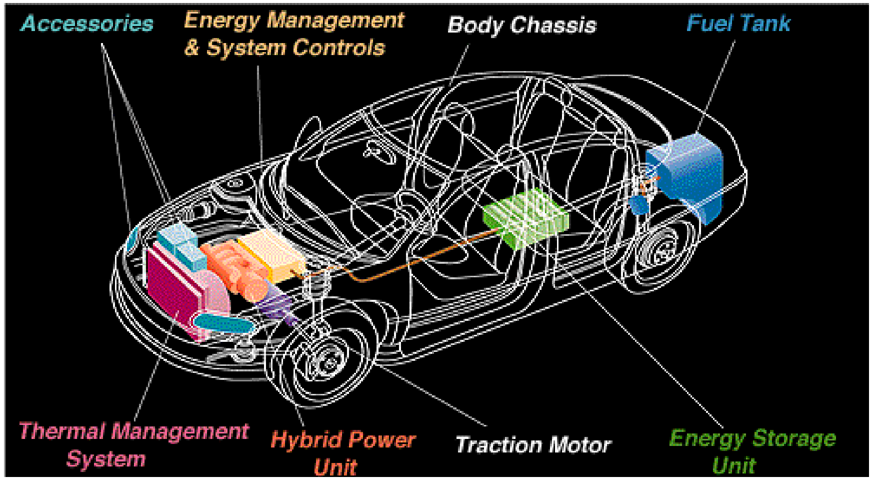
- **Results**

Method	적응도 향상 정도	Man-Month Ratio
인 간	1	2개월
Expert System Only	2	8일
GA(임의의 초기치)	3	17일
GA+Expert System	3	9일

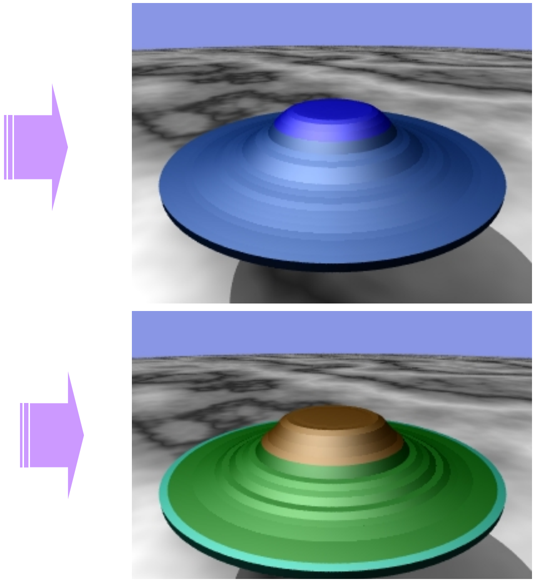


Application of GA (4)

- 플라이휠 설계
  - √ 형상의 용이한 설계를 위해 대개 플라이휠은 금속재료로 제작된다. 플라이휠은 스스로의 관성을 이용하여 속도의 변동을 조절할 뿐 아니라, 에너지를 저장하는 역할을 한다. 원판형의 플라이휠이 회전하게 되면 마찰손실이 존재하더라도 어느 수준 정도의 운동에너지를 생성하게 되고 모멘텀을 발생시킨다. 또한, 저장된 에너지는 기계적, 전기적 작용에 의해 방출된다. 플라이휠형 배터리는 화학적 배터리보다 에너지 저장능력이 뛰어나다. 특히, 크기가 작아질수록 이러한 플라이휠의 요구가 더욱 절실해지는데 전기자동차(**self-contained electric powered vehicle**)인 경우에서 그 우수성을 발견할 수 있다.



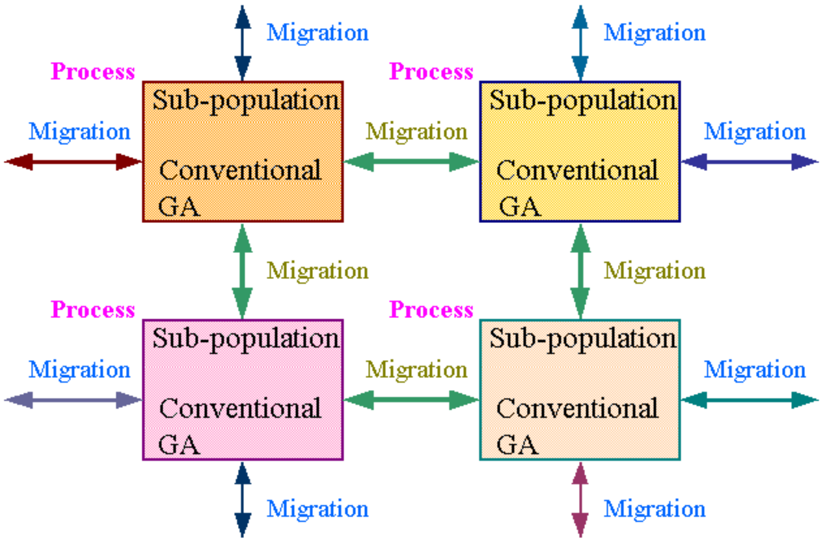
Hybrid 전기자동차



GA에 의해 최적화된 복합재료 플라이휠의 **3**차원 형상,

Application of GA (5)

- 병렬 유전자알고리즘을 이용한 헬리콥터 블레이드의 공력/공력소음 최적화
  - √ 헬리콥터와 같이 회전날개 항공기에서는 로터시스템과 와류의 충돌에 의해 비행성능에 지장을 주게되고 또한 회전유동에 의한 소음이 발생한다. 저진동, 저소음(**BVI :blade-vortex interactions)**의 로터 및 블레이드를 설계하기 위해 유전알고리즘이 적용되고 있다. 전산유체역학적 해석방법에 의해 유전자알고리즘 설계해를 탐색하는 경우에는 계산시간 및 비용이 엄청나게 증가하므로 이러한 단점을 최적화기법에서 해결하기 위해 병렬형 유전알고리즘을 개발하여 전역설계의 생성뿐 아니라 병렬연산기술을 활용하여 설계기간을 단축시킬 수 있다.



병렬유전자알고리즘을 이용한 각 **Processor**간의 **Data** 교환

Simulated Annealing

2006. 4. 1

Principal Heuristic Algorithms

- Genetic Algorithms (Holland – 1975)
  - √ Inspired by genetics and natural selection
- **Simulated Annealing (Kirkpatrick – 1983) : Today's issue**
  - √ Inspired by molecular dynamics – energy minimization
- Particle Swarm Optimization (Eberhart and Kennedy - 1995)
  - √ Inspired by the social behavior of swarms of insects or flocks of birds

These techniques all use a combination of randomness and heuristic "rules" to guide the search for global maxima or minima

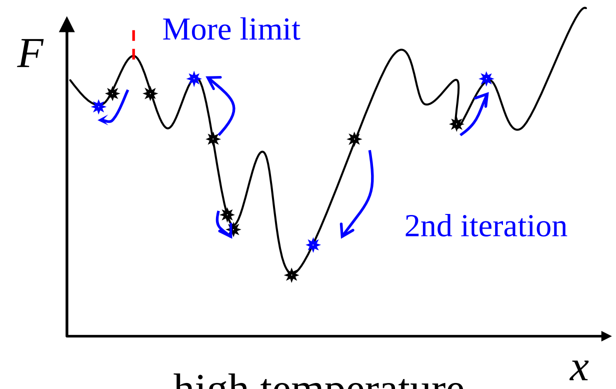
Origin of Simulated Annealing (SA)

- **Definition:** A heuristic technique that mathematically mirrors the cooling of a set of atoms to a state of minimum energy.
- **Origin:** Applying the field of Statistical Mechanics to the field of Combinatorial Optimization (1983)
- Draws an **analogy** between the cooling of a material (search for minimum energy state) and the solving of an optimization problem.
- **Original Paper** Introducing the Concept
  - √ Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Volume 220, Number 4598, 13 May 1983, pp. 671680.

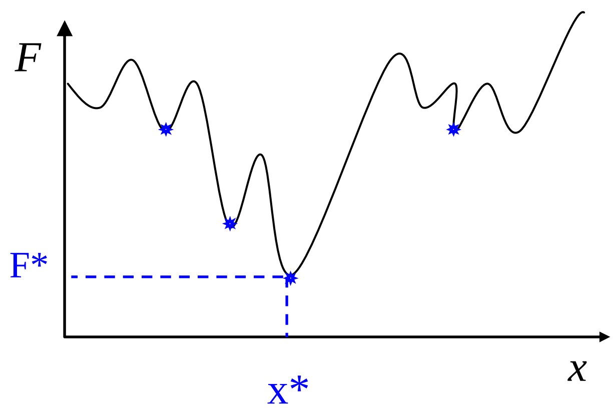
Simulated Annealing(SA)

- **Like GA, SA is a stochastic process.**

SA mimics the physical process of heating then slowly cooling a metal during processing to receive thermal/structure stresses.



high temperature



low temperature

√ **Randomly generates design points**

√ **Downhill point movement always o.k**

√ **Uphill movement depends on the statistics (90% No, 10%Yes)**

√ **More limit changes ?**

√ **High/low temperature criterion & steps ?**

Simulated Annealing(SA)

1) Start with initial selection of point, $\vec{x}$

2) Pick an initial high temperature *T*

3) For each design

$$\vec{x}_{new} = \vec{x}_{old} \pm \vec{p}$$

Move limit

4) If $F(\vec{x}_{new}) \pi F(\vec{x}_{old})$ $\vec{x}_{new}$ is accepted, next $\vec{x}$

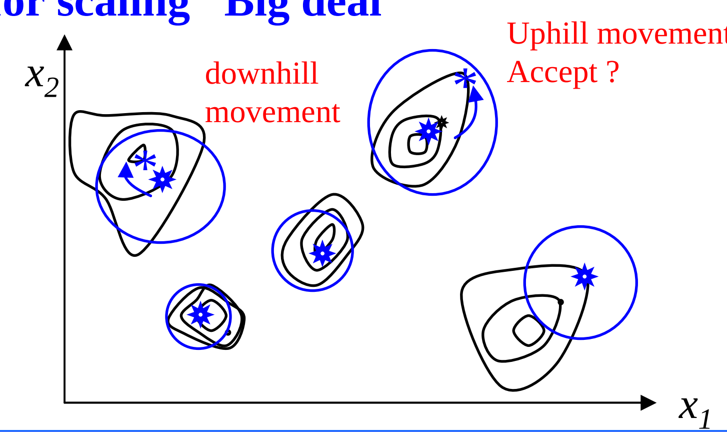
Else, accept an"uphill move"with the following probability

$$P_{accept} = \exp\left[\frac{-|F(\vec{x}_{new}) - F(\vec{x}_{old})|}{KT}\right]$$

simulated annealing temp

Constant used for scaling "Big deal"

Get next $\vec{x}$



5) T_{new} = 80% of T_{old}

Repeat the process

Accepting a New Current Solution

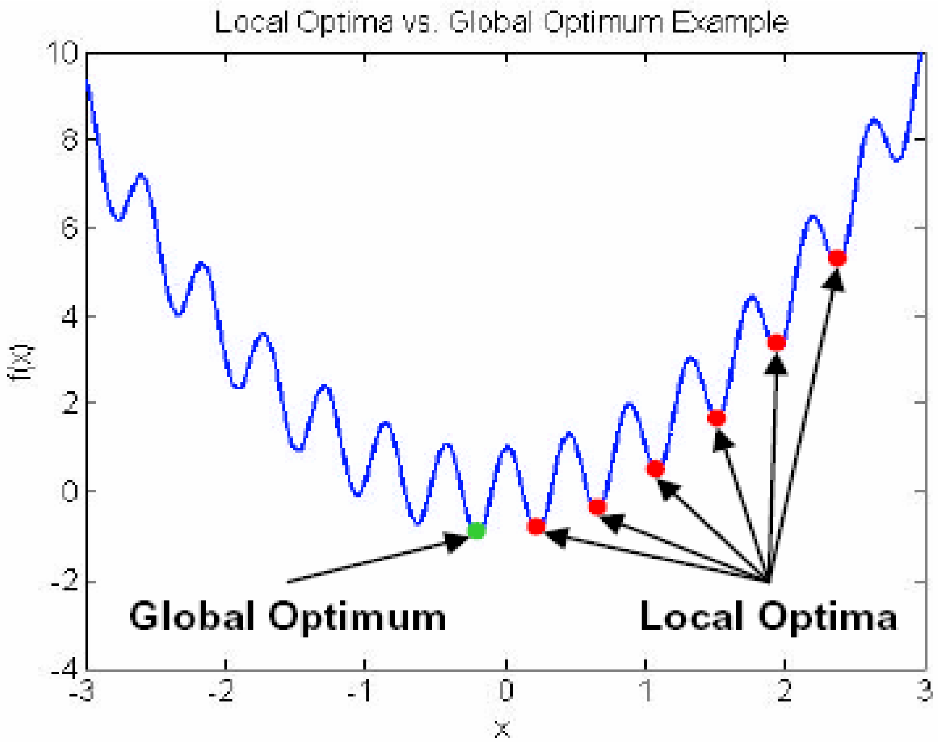
- Why move to a worse current solution?
  - √ To avoid getting trapped in a local optimum.
- Local Optimum
  - √ A solution is locally optimal if there is no neighbor who has a better objective function value.
- Global Optimum
  - √ A solution is globally optimal if there is no other solution in the entire feasible trade space that has a better objective function value.
  - √ Note: We are only talking about single objective problems.

Accepting a New Current Solution (Continue)

- Local Optima vs. the Global Optimum
- Example

  $f(x) = \cos(14.5x - 0.3) + (x + 0.2)x$

  Minimize $f(x)$

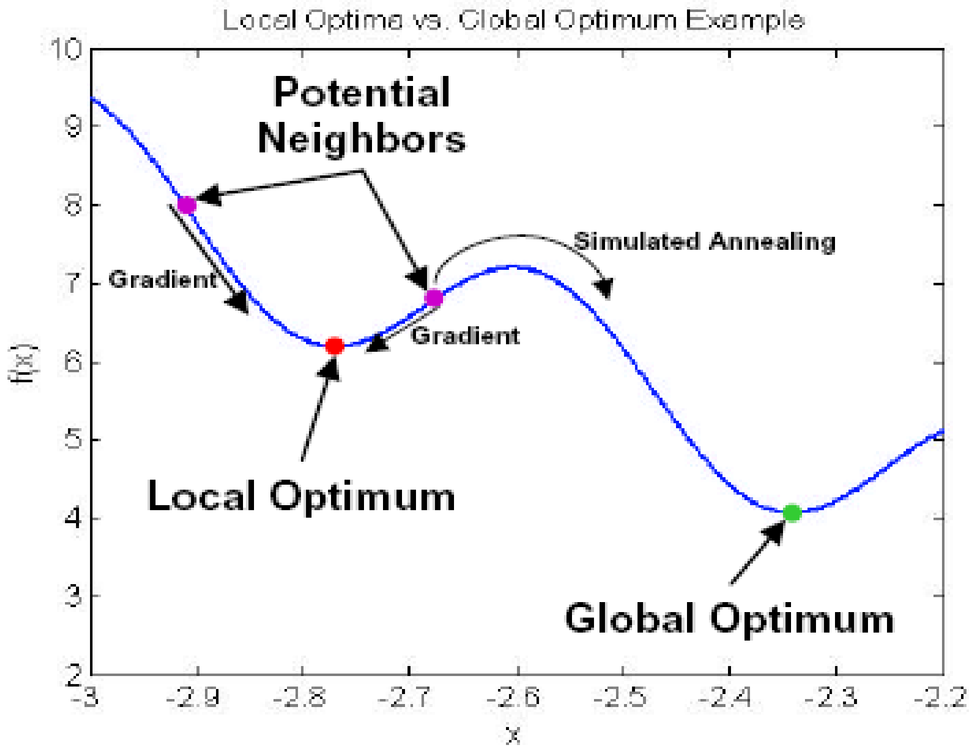


Accepting a New Current Solution (Continue)

- Local Optima vs. the Global Optimum
- Example

  Minimize $f(x) = \cos(14.5x - 0.3) + (x + 0.2)x$

  Subject to $-2.2 \geq x \geq -3$



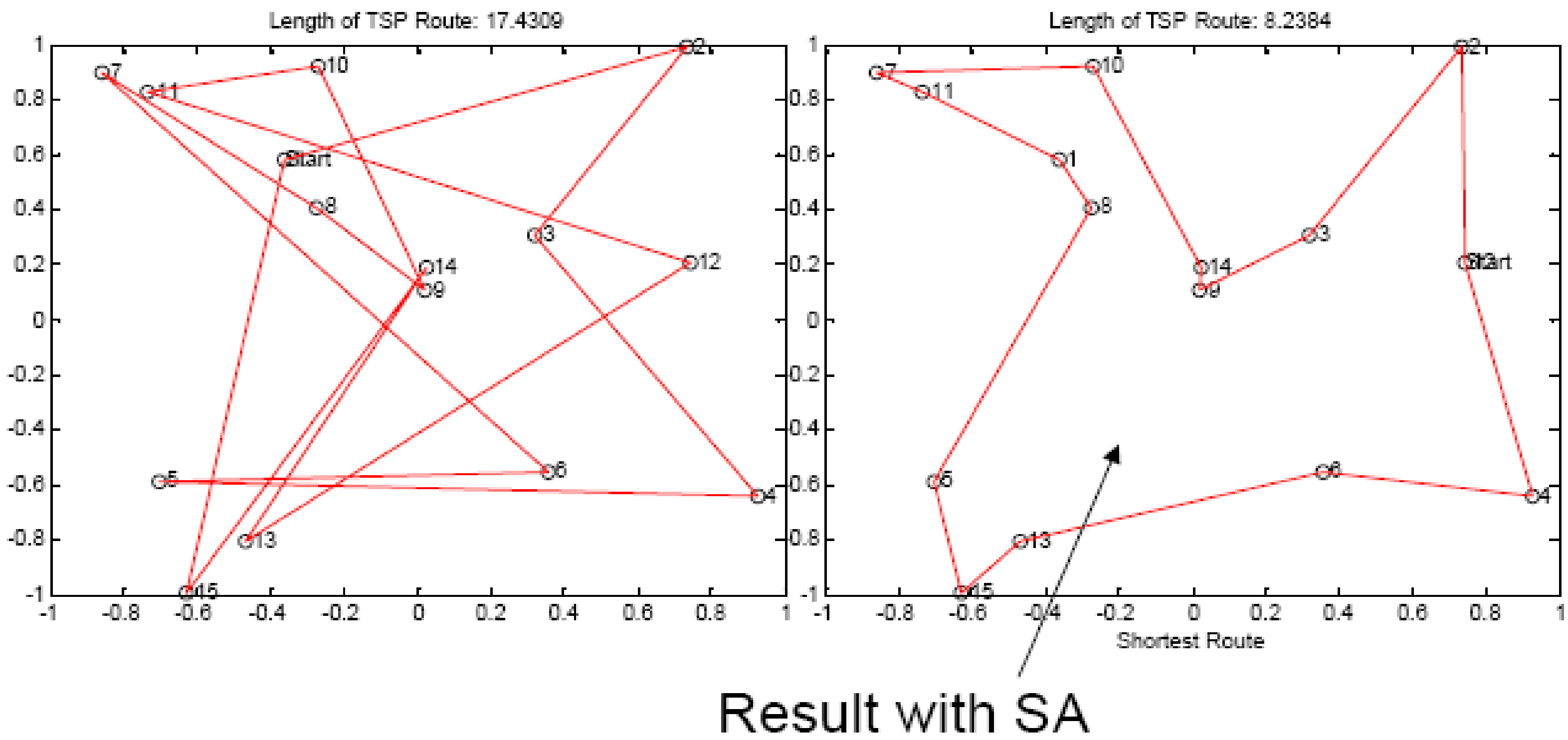
Summary: Steps of SA

- **The Simulated Annealing Algorithm**

1) Choose a random X, select the initial system temperature, and outline the cooling (ie. annealing) schedule
2) Evaluate $E(X_i)$ using a simulation model
3) Perturb X_i to obtain a neighboring Design Vector (X_{i+1})
4) Evaluate $E(X_{i+1})$ using a simulation model
5) If $E(X_{i+1}) < E(X_i)$, X_{i+1} is the new current solution
6) If $E(X_{i+1}) > E(X_i)$, then accept X_{i+1} as the new current solution with a probability $e^{(-\Delta/T)}$ where $\Delta = E(X_{i+1}) - E(X_i)$.
7) Reduce the system temperature according to the cooling schedule.
8) Terminate the algorithm.

Example : Traveling Salesman Problem

- Initial (Random) Route
  Length: 17.43
- Final (Optimized) Route
  Length: 8.24



Research in SA

- Alternative Cooling Schedules and Termination criteria
- Adaptive Simulated Annealing (ASA)
  - √ determines its own cooling schedule
- Hybridization with other Heuristic Search Methods
  - √ GA, Tabu Search ...
- Multiobjective Optimization with SA

Summary of stochastic Methods (GA,SA)

- **suitable for problems with many local minima**
- **design space can be non-smooth and discontinuous**
- **can handle discrete variables (*x*'s)**
- **Function calls can be large compared to path –building methods (be careful not to exceed grid search number of functions)**

SA code : simman.f

- simman.f
  - √ **Simple SA code**
  - √ **Format : FORTRAN 77**
  - √ **Optimization problem type : Unconstrained Optimization Problem Only**
  - √ 단**, penalty function**을 이용**, pseudo-objective**형태로 전환하여 **Constrained Problem**을 해결하는 방식이 가능
  - √ **Reference : Goffe, Ferrier and Rogers, "Global Optimization of Statistical Functions with Simulated Annealing," Journal of Econometrics, vol. 60, no. 1/2, Jan./Feb. 1994, pp. 65-100.**

SA code : simman.f

- **Input Parameter**

parameter	type	definition
N	Integer	Number of Design Variables
X(I)	real	Initial Value Vector
LB(I)	real	Lower Boundary Vector
UB(I)	real	Upper Boundary Vector

- **Internal Parameter for SA**

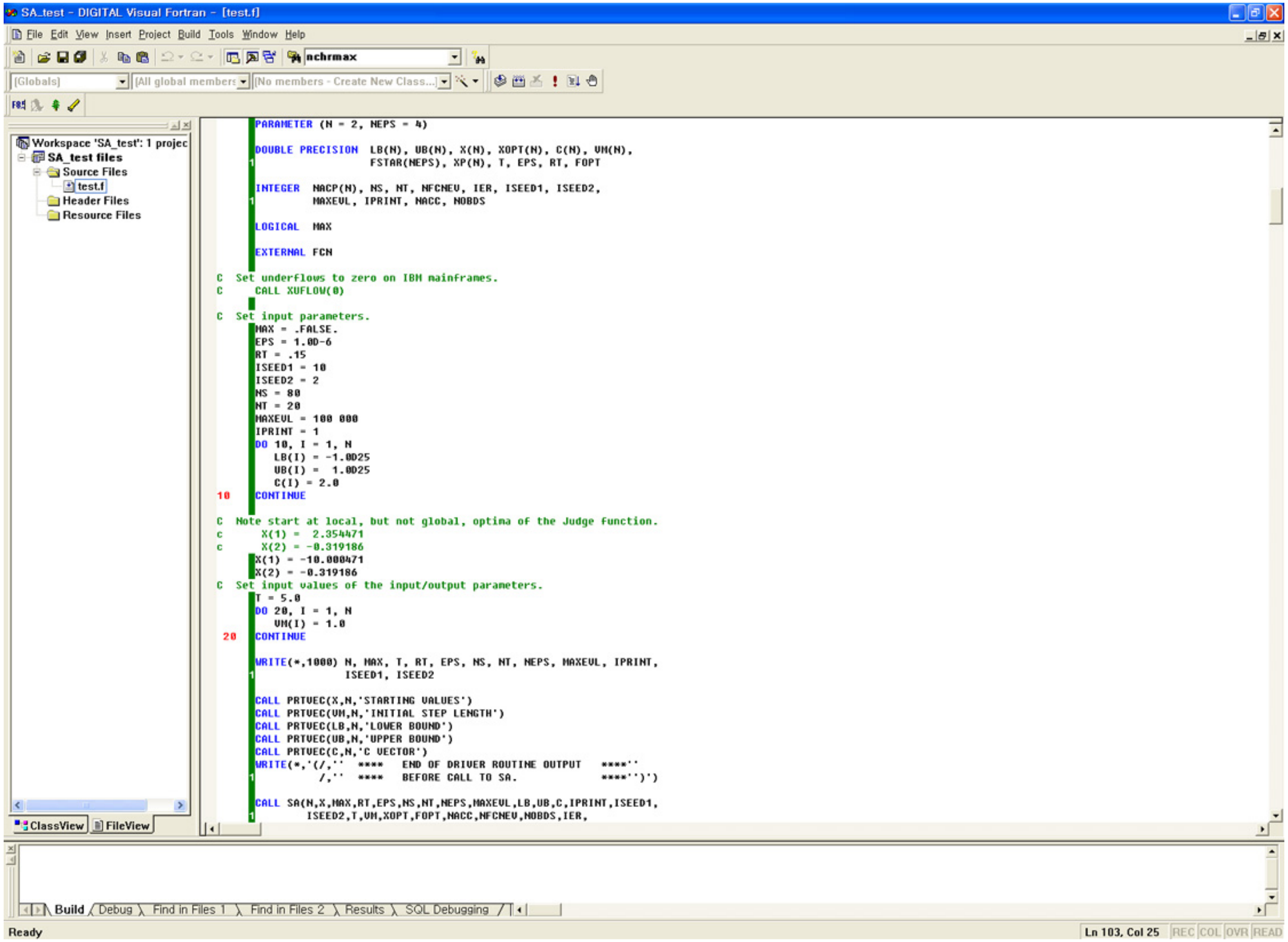
parameter	type	definition(default)
RT	real	Temperature Reduction Factor (0.1~0.95)
MAX	logic	True = max. / False = min.
EPS	real	Error tolerance for termination(1.0E-06)
NT	integer	Number of iterations before temperature reduction(20)
NEPS	integer	Number of final function values used to decide upon termination(4)
MAXEVL	real	The maximum number of function evaluations(100000)

√ **parameter RT**

– **The temperature reduction factor. The value suggested by Corana et al. is .85**

SA code : simman.f

- **Program Layout**



SA code : simman.f

- procedure

File : simman.f

Step 1 : parameter definition

No. of Design Variable

Problem Type , tolerance, RT

NT, NS,

Upper & Lower Bound

Initial Value

```
      PARAMETER (N = 2, NEPS = 4)

      DOUBLE PRECISION  LB(N), UB(N), X(N), XOPT(N), C(N), VM(N),
     1                  FSTAR(NEPS), XP(N), T, EPS, RT, FOPT

      INTEGER  NACP(N), NS, NT, NFCNEV, IER, ISEED1, ISEED2,
     1         MAXEVL, IPRINT, NACC, NOBDS

      LOGICAL  MAX

      EXTERNAL FCN

C  Set underflows to zero on IBM mainframes.
C     CALL XUFLOW(0)

C  Set input parameters.
      MAX = .FALSE.
      EPS = 1.0D-6
      RT = .15
      ISEED1 = 10
      ISEED2 = 2
      NS = 80
      NT = 20
      MAXEVL = 100 000
      IPRINT = 1
      DO 10, I = 1, N
         LB(I) = -1.0D25
         UB(I) =  1.0D25
         C(I) = 2.0
10    CONTINUE

C  Note start at local, but not global, optima of the Judge function.
c      X(1) =  2.354471
c      X(2) = -0.319186
      X(1) = -10.000471
      X(2) = -0.319186
```

SA code : simman.f

File : **simman.f**

Step 2 : objective definition

```
      SUBROUTINE FCN(N,THETA,H)
C  This subroutine is from the example in Judge et al., The Theory and
C  Practice of Econometrics, 2nd ed., pp. 956-7. There are two optima:
C  F(.864,1.23) = 16.0817 (the global minumum) and F(2.35,-.319) = 20.9805.

      DOUBLE PRECISION THETA(2), H

      DOUBLE PRECISION Y(20), X2(20), X3(20)
```

Object Function

```
      H = 0.0
      DO 100, I = 1, 20
         H = (THETA(1) + THETA(2)*X2(I) + (THETA(2)**2)*X3(I) - Y(I))**2
     1        + H
100   CONTINUE

      RETURN
      END
```

SA code : simman.f

Step 3

√Compiling and running the system:

Data from Optimization Processing

Optimal Solution

: Optimum X vectors

Optimal Function Value,

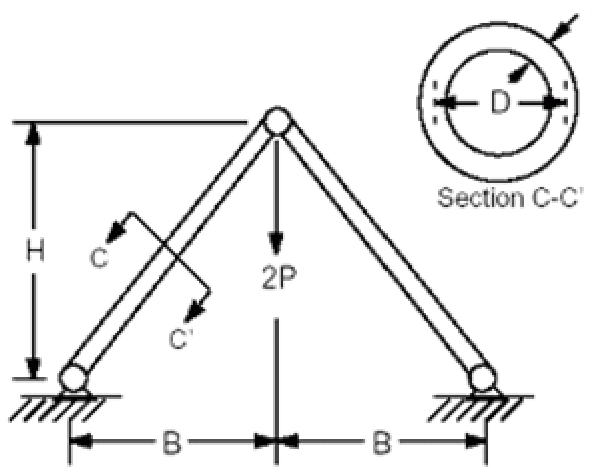
Optimum Results

"C:₩KwonSuJeon₩MYPROJECT₩SA_test₩Debug₩SA_test.exe"

```
                        STEP LENGTH (VM)
0.88893E-04  0.50756E-04


INTERMEDIATE RESULTS BEFORE NEXT TEMPERATURE REDUCTION

 CURRENT TEMPERATURE:              0.43249E-08
 MIN FUNCTION VALUE SO FAR:     16.0817306860283473
 TOTAL MOVES:                        3200
    DOWNHILL:                         743
    ACCEPTED UPHILL:                  758
    REJECTED UPHILL:                 1699
 TRIALS OUT OF BOUNDS:                  0
 NEW MINIMA THIS TEMPERATURE:           2

                        CURRENT OPTIMAL X
0.86479       1.2357

                        STEP LENGTH (VM)
0.26183E-04  0.20625E-04


 SA ACHIEVED TERMINATION CRITERIA. IER = 0.


 ****   RESULTS AFTER SA   ****

                        SOLUTION
0.86479       1.2357

                        FINAL STEP LENGTH
0.26183E-04  0.20625E-04

OPTIMAL FUNCTION VALUE:   16.08173068603
NUMBER OF FUNCTION EVALUATIONS:          38401
NUMBER OF ACCEPTED EVALUATIONS:          18037
NUMBER OF OUT OF BOUND EVALUATIONS:          0
FINAL TEMP:  0.4324879820095E-08  IER:   0
Press any key to continue
```

2-Bar Truss Design : Structural Optimization

- **Description**
  - √ **The symmetric 2-bar truss design shown in below has been studied by several researchers**
  - √ **Balling and Clark(1992), Schmit, (1981), Sobieszczanski-Sobieski et al(1982)**



  - √ **The objective of this optimum problem is to <u>minimize the weight of truss system</u> subject to behavioral constraints**

  - √ **Related parameters**

- B = 30 in

- t = 0.1 in

- ρ = 0.3 lbs/in3

- σ_y = 60,000 psi

- E = 30E6 psi

2-Bar Truss Design : Structural Optimization

- **Formulation for Optimization**
  - √ **Minimize** $W(X)=2\rho\pi Dt(B^2+H^2)^{1/2}$
    - Minimize the weight of truss system
  - √ **Subject to**

$$g_1(X) = \sigma_e - \sigma \geq 0$$

    - the first constraint prevents failure due to Euler buckling

$$g_2(X) = \sigma_y - \sigma \geq 0$$

    - the second constraint prevents failure due to yield stress

Where,

$0.5 \leq D \leq 5.0$ (in) $\Rightarrow X(1)$: mean tube diameter

$5.0 \leq H \leq 50.0$ (in) $\Rightarrow X(2)$: height of the truss

√ The resulting optimum value from (Schmit, 1981) for W(x) is 19.8 lbs.

- $W^* = 19.8$ lbs (at $D^* = 2.47$ in , $H^* = 30.15$ in)

SA Application : 2-Bar Truss Design

- **Algorithm과 initial Value에 따른 결과 비교**

r_p = 1.0E2 , initial values X(1) =0. , X(2)= 0.

Program	DOT					simman	true
	Constrained			Unconstrained			
Algorithm	MMFD	SLP	SQP	BFGS	F-R	SA	**-**
X1 (D)	2.481	2.476	2.476	4.558	4.558	-	**2.47**
X2 (H)	29.870	29.992	30.00	15.031	15.031	-	**30.15**
Opt. val. (W)	19.800	19.800	19.800	28.828	28.828	-	**19.8**
Num. of Iter.	9	20	11	3	3	-	**-**
Num. of fun. eval.	77	69	45	24	24	-	**-**

r_p = 1.0E2 , initial values X(1) =1. , X(2)= 5.

Program	DOT					simman	true
	Constrained			Unconstrained			
Algorithm	MMFD	SLP	SQP	BFGS	F-R	SA	**-**
X1 (D)	3.462	2.480	2.474	3.588	3.588	2.452	**2.47**
X2 (H)	17.588	29.990	30.041	17.820	17.820	30.582	**30.15**
Opt. val. (W)	22.690	19.800	19.800	23.601	23.601	19.804	**19.8**
Num. of Iter.	5	20	10	3	3	-	**-**
Num. of fun. eval.	26	66	40	26	26	2201	**-**