

정보기술응용연구
제 3 권 제 3 호
2 0 0 1 년 9 월

유전적 프로그래밍을 이용한 응답면의 모델링 I : 방향도함수 기반의 Smoothing 기법

연윤석^{*}, 이욱^{**}

요 약

.....

본 논문은 최소한의 학습데이터를 사용하여 비선형의 응답면을 모델링할 수 있는 방안으로 유전적 프로그래밍(Genetic Programming, GP)의 사용을 모색하였다. 이때 대두되는 가장 큰 문제는 GP 트리가 부족한 학습 데이터 때문에 심한 Overfitting 현상을 보인다는 점이다. 이를 극복하기 위한 방법으로 DDBS (Directional Derivative-Based Smoothing) 기법을 제안하였고, 유용성을 검증하기 위해서 4 가지 응용 예를 보였다.

.....

*) 대전대학교 기계설계공학과

***) 대전대학교 전기공학과

1. 서론

기계 전기전자 등 다양한 분야에서 응답면(Response surface) 기법이 활용되고 있는데[1-4], 전통적으로 응답면을 모델링하기 위해 다항식(Polynomial)이 많이 사용되고 있다. 최근에는 응답면의 전역적(Global) 그리고 지역적(Local) 특성을 모두 고려할 수 있는 크리깅(Krigging)[1]을 사용하고 있는 추세이다. 또한 GP(Genetic Programming)[5]을 사용하여 응답면을 근사 시키고자 하는 시도도 이루어지고 있는데[6-7], 본 논문은 이러한 노력의 일환으로 GP의 사용 시 가장 큰 문제점으로 대두될 수 있는 Overfitting 문제를 회피함과 동시에 예측 성능의 향상을 도모할 수 있는 방안을 제시하고자 한다. GP는 유전적 연산자 (Genetic operators)들을 사용하여 최적 GP 트리 f_{GP} 의 구조를 생성하기 때문에 기존의 근사화 기법보다 뛰어난 성능을 보여 준다는 것은 이미 알려진 사실이다[5]. 특히 최적화 문제의 구속조건 식 (Constraint)들은 목적함수(Object function)들과 비교하여 비선형성(Nonlinear)이 크기 때문에 단순한 다항식을 사용한 근사는 그 예측 성능이 크게 떨어지게 된다. 이러한 경우 GP가 그 대안이 될 수 있는데, 일반적으로 응답면 모델링 시 사용되는 학습집합 (n 차원, Learning set) $L = \{ X_{1,i}, \dots, X_{n,i}, Y_i \}_{i=1}^m$ 의 크기(m)는 매우 작기 때문에 함수 근사화 입장에서 보면 낮은 차수(Order)의 다항식을 제외하고는 Overfitting은 필연적으로 발생하는 문제가 된다. 즉 학습 오류의 크기는 매우 작으나 학습집합에 포함되어 있지 않은 다른 데이터에 대한 출력 값이 실제 수치와 크게 차이가 나게 되는데, 이러한 현상은 GP에서 자주 발견되며, 경우에 따라서는 전혀 다른 값이 나오거나 그 크기가 무한대가 되는 심각한 문제가 종종 발생한다. f_{GP} 는 주어진 함수와 터미널 집합(Function and terminal set)에 정의된 원소들의 가능한 모든 조합에 의해 생성되기 때문에 비선형성이 매우 커서, 비록 학습집합의 데이터들에서는 원하는 거동을 나타내도 그 이외의 입력 데이터에서는 전혀 예상하지 못한 거동을 보일 수 있다. 예를 들면 f_{GP} 의 한 노드(Node)가 산술연산자의 하나인 '/'으로 구성되어 있고, 학습 데이터 이외의 다른 입력 데이터에 대한 하위 트리의 연산 결과가 0이고, 이 값이 '/'의 연산에서 0로 나누어지는 결과가 발생할 수 있다. 따라서 생성된 f_{GP} 의 전반적인 거동을 살펴보고, 바람직하지 못한 거동을 보이면 진화 과정에서 제거시킬 필요성이 대두된다. 응답면 기법을 활용하여 최적화 과정을 수행하는 주된 이유는 계산량이 많은 정확한 해석을 가급적 회피하고 비교적 합리적인 최적 해를 구하기 위한 것이라고 할 수 있다. 따라서 응답면의 모델링 시 필요한 학습집합의 크기는 최소화되는 것이 바람직하다. 매

우 작은 혹은 최소한의 크기를 갖는 학습집합을 사용한다는 것은 함수 근사화 관점에서 볼 때, 바로 Overfitting에 의한 예측 성능이 대폭적 저하를 의미하게 된다. 또 하나의 자명한 사실은 기존에 존재하는 어떠한 근사화 방법을 사용하여도, 부족한 학습 데이터로 만들어진 근사 모델은 원하는 정확도를 보일 수 없다는 것이다.

위에서 응답면을 모델링 시 발생하는 중요한 두 가지 문제점, 즉 응답면의 비선형성과, 학습집합의 크기가 작은 것에서 기인하는 Overfitting과 예측 성능의 저하를 언급하였다. 본 논문에서는 이와 같은 문제점을 완화하기 위한 방안으로 방향도함수 기반의 Smoothing (Directional Derivative-Based Smoothing, 이하 DDBS) 기법을 채용한 GP를 도입하고자 한다. DDBS의 기본 핵심은 f_{GP} 를 주어진 학습 샘플 $\vec{X}_i = (X_{1,i}, \dots, X_{n,i})$ 에서 Y_i 의 값에 근사 시키면서 동시에 그 점에서 근접한 다른 학습 샘플 $\vec{X}_j = (X_{1,j}, \dots, X_{n,j})$ 방향으로의 미분 값 즉 $\nabla f \cdot \vec{D}_{i,j}$ 와 $\nabla f_{GP} \cdot \vec{D}_{i,j}$ 의 값이 근접하도록 하는데 있다. ∇f 는 목적함수나 구속조건 식의 Gradient이고, $\vec{D}_{i,j}$ 는 $(X_{1,i}, \dots, X_{n,i})$ 에서 $(X_{1,j}, \dots, X_{n,j})$ 으로의 방향 벡터이다. 물론, 이런 방법은 주어진 샘플에서 방향 도함수의 값을 산출하기 위해 추가적인 또 다른 학습 샘플을 필요로 한다는 큰 단점이 있다. 따라서 $\nabla f \cdot \vec{D}_{i,j}$ 대신에 (\vec{X}_i, Y_i) 와 (\vec{X}_j, Y_j) 를 연결한 직선의 기울기 $s_{i,j}$ 를 사용하였다. f 가 선형이 아닌 이상 $s_{i,j}$ 는 $\nabla f \cdot \vec{D}_{i,j}$ 와 큰 차이를 보이는 것이 일반적이지만, DDBS는 f_{GP} 가 Overfitting이 되는 것을 방지하는 탁월한 효과가 있고, 또한 DDBS의 채용은 f_{GP} 가 f 의 전체적인 특징을 대체적으로 유지시키도록 하기 때문에 부족한 학습 샘플의 개수를 보충할 수 있다는 장점이 있다. 이와 같은 사실을 보이기 위하여 본 논문에서는 일반적인 GP와 DDBS를 채용한 GP에 대한 응답면 모델링을 수행하고 결과를 비교 검토하였다.

2. 유전적 프로그래밍

GP는 유전적 알고리즘(Genetic algorithm)의 일종으로 개체(Individual)가 트리 형태의 프로그램이 된다. 트리는 함수와 터미널 집합에 정의된 함수와 터미널의 조합으로 생성되는데, 문법적으로 올바르기만 하면 어떤 형태의 트리도 생성이 가능하다. GP에서 함수는 인자(Argument)를 갖고 그렇지 않으면 터미널로 분류된다. 유전적 알고리즘과 유사하게 GP도 보통 3 종류의 유전적 연산자를 사용한다. 즉 재생산(Reproduce), 교배(Crossover), 그리고 돌연변이(Mutation) 등이다.

진화과정의 주된 목표는 이 3 가지 유전적 연산자를 사용하여 트리의 적합도 함수(Fitness function) 값이 최적화 되도록 트리의 구조 자체를 변화시켜 최적의 트리를 생성하는 것이다. 일반적으로 GP의 다양한 응용을 위해서는 수치 파라미터 혹은 가중치(Weight)를 필요로 하는데, 보통 트리의 모든 노드에 가중치가 첨가되고, 노드의 출력은 항상 이 가중치를 곱하여 산출하게 된다.

2.1 가중치 추정

GP는 유전적 연산자를 사용하여 최적의 트리 구조를 찾는 것에만 중점을 두고 있는데, 응답면 모델링을 위해서는 최적의 트리 구조 탐색과 동시에 트리의 모든 노드에 부착되어 있는 가중치를 트리의 적합도 값이 보다 감소될 수 있도록 추정해야 한다. 이것은 매우 많은 계산량을 요구하기 때문에 가중 선형연상 기억 또는 선형연상 기억(Linear associative memory)을 채용하여 가중치 추정 시 필요한 계산량을 감축시키려는 연구가 수행된 적이 있다[8-10]. 그러나 응답면의 비선형성이 큰 경우는 선형연상 기억을 채용한 방법은 정확한 가중치의 탐색이 어렵고 더욱이 DBBS가 GP에 채용되는 경우 적용하기가 불가능하다는 문제점이 있다. 따라서 본 논문에서는 가중치 추정을 위해서 패턴 탐색기법 중 효율이 우수한 Hooke & Jeeves 방법[11]을 사용하였다. 개체군(Population)에 포함된 모든 트리의 가중치를 Hooke & Jeeves 방법을 사용하여 추정할 때 계산 시간을 절감하기 위하여 최대 허용 가능한 반복 횟수를 1000으로 제한하였다. 그리고 이 과정이 끝나면 가장 우수한 몇 개의 트리(5 개)를 선정하여 Hooke & Jeeves 방법을 다시 적용하여(최대 반복 회수 10000) 보다 정확한 가중치 산정을 도모하였다.

2.2 함수와 터미널 집합

사용된 함수 집합은 다음과 같다.

$$F = \{ +, -, *, /, \sin, \cos, \tan, \sin^{-1}, \cos^{-1}, \tan^{-1}, \sinh, \cosh, \tanh, \exp, \log, \sqrt{\cdot}, \text{pow} \}$$

위의 함수 집합을 사용하여 생성된 트리는 적합도 값을 계산 시 수학적 연산 오류가 종종 발생시키는데, 예를 들면 트리의 한 노드가 'log' 이고, 0 보다 작은 값이 사용되면 연산 오류가 발생한다. 이런 문제점을 갖고 있는 트리는 그 적합도의 값을 매우 큰 수로 만들어 다음 진화 과정에서 제외시킬 수 있다. 진화의 초기 단계에서는 연산 오류를 발생시키는 트리들은 전체 트리의 30% 이상이나 세대(Generation)가 지나갈수록 그 수가 점차적으로 줄어 거의 0%에 가까워진다.

터미널 집합에서는 n 개의 독립 또는 설계변수와 크기가 1 보다 작은 난수 (Random number)를 사용하였다.

3. 방향도함수 기반의 Smoothing

일반적으로 Smooth fitting이 이슈가 되는 경우는 학습집합 L에 노이즈(Noise)가 포함된 경우이지만, 응답면의 경우 L의 크기가 매우 작기 때문에 역시 큰 문제로 대두되게 된다. 특히 GP에서는 트리의 비선형성이 과도하게 커지는 경향이 있기 때문에 Smooth fitting 기법은 필수적인 사항이라고 할 수 있다.

3.1 정식화

Smooth fitting을 위해서는 먼저 트리가 어느 정도 Smooth한가하는 것을 측정할 수 있는 기준 ϑ_s (Smoothness measure)의 정의가 필요하다. DDBS는 Fig.1을 사용하여 다음과 같이 설명될 수 있다. 샘플 포인트 P_i (Y_i 는 제외)에서 가장 가까운 즉 Euclidean distance가 최소가 되는 p개의 샘플 포인트를 선정한다. 그러면 P_i 와 P_j 를 연결하는 직선의 식 $\vec{h}_{i,j}$ 는 (1)식과 같이 표현된다.

$$\vec{h}_{i,j} = \vec{X}_i + \mu(\vec{X}_j - \vec{X}_i) \quad (1)$$

where $0 \leq \mu \leq 1$

여기서 $\vec{h}_{i,j}$ 의 방향 벡터는 $\vec{D}_{i,j} = \frac{\vec{X}_j - \vec{X}_i}{|\vec{X}_j - \vec{X}_i|}$ 이 되고, $\Delta Y_{i,j} = Y_j - Y_i$ 라고 하면 두 점 (\vec{X}_i, Y_i) 와 (\vec{X}_j, Y_j) 를 지나는 직선의 기울기 $\vec{s}_{i,j} = \frac{\Delta Y_{i,j}}{|\vec{X}_j - \vec{X}_i|}$ 이 된다. q개의 서로 다른 μ 에 대한 $\vec{h}_{i,j}$, 즉 $\vec{h}_{i,j}(\mu_r)$ ($\mu_{i+1} = \mu_i + n/(q+1), n=0, \dots, q-1, \mu_0=0$)를 $\vec{h}_{i,j}^r$ 로 표기하면, $\vec{h}_{i,j}^r$ 에서 $\nabla f_{GP} = [\frac{\partial f_{GP}}{\partial x_1}(\vec{h}_{i,j}^r), \dots, \frac{\partial f_{GP}}{\partial x_n}(\vec{h}_{i,j}^r)]$ 이 된다. 따라서 (2)식과 같은 ϑ_s 을 정의할 수 있다.

$$\vartheta_s = 1/(mpq) \sum_{i=1}^m [\sum_{j=1}^p \{ \sum_{k=1}^q (\nabla f_{GP}(\vec{h}_{i,j}^k) \cdot \vec{D}_{i,j} - s_{i,j})^2 \}] \quad (2)$$

(2)식에서 알 수 있듯이, ϑ_s 을 최소화한다는 것은 직선 $\vec{h}_{i,j}$ 를 따라가면서 q개의 포인트에 대하여 ∇f_{GP} 를 계산하고, $\vec{D}_{i,j}$ 방향에 대한 방향 도함수를 구하여 그 값들이 $s_{i,j}$ 에 근접하도록 하는 것이다. 실제 계산상으로는 $\nabla f_{GP} \cdot \vec{D}_{i,j}$ 를 계산하기 위하여 ∇f_{GP} 를 계산할 필요는 없고 $\vec{D}_{i,j}$ 방향으로의 f_{GP} 의 변화율만 계산하여 사용하면 된다. 만일 f_{GP} 가 Overfitting이 되거나 학습 데이터이외의 점에서 급격한 Peak나 진동현상을 보이면, 당연히 ϑ_s 의 값은 매우 커질 가능성이 높다. 적합도 계산 시 가중치 추정을 통하여 이런 f_{GP} 의 ϑ_s 값을 줄여 주게 되는데, 이것이 충분하지 못할 경우 이 트리의 적합도 값은 다른 트리와 비교하여 상대적으로 커지게 되므로, 다음 진화과정에서 배제될 가능성이 대폭적으로 증가하게 된다.

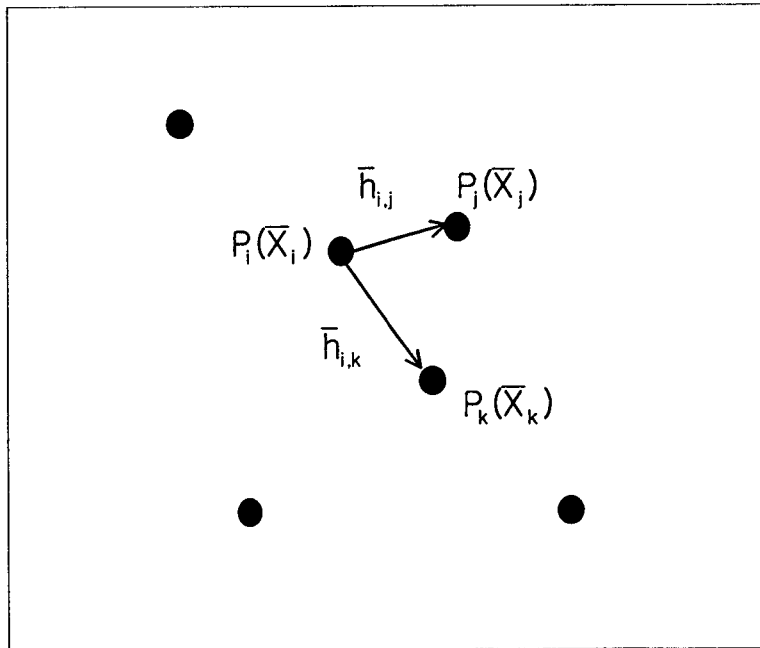


Fig.1. Line segments passing through two points.

3.2 적합도 함수

본 논문에서는 모든 설계변수의 값은 0.01과 1.0 사이에 오도록 스케일링

(Scaling) 하였고, Y_i 의 값도 0 과 2 사이에 오도록 스케일링 하였다. 즉 $0.01 \leq x_{i,j} \leq 1, 0 \leq Y_j \leq 2$ ($i=1, \dots, n, j=1, \dots, m$) 이다. 설계변수의 최소 값이 0이 아닌 0.01로 선택한 이유는 연산 오류를 발생시키는 트리를 조금이라도 감소시킬 가능성이 있기 때문이다. 본 3.2절의 모든 내용은 이렇게 스케일링된 학습집합에 대해서 논의된 것이다.

(3)식은 ϑ_s 을 포함하지 않은 GP의 적합도 함수를 정의한 것이다.

$$\vartheta = \vartheta_{MSE} \quad (3)$$

$$\text{where } \vartheta_{MSE} = 1/m \sum_{i=1}^m [f_{GP}(\vec{X}_i) - Y_i]^2$$

ϑ_{MSE} 는 f_{GP} 가 주어진 학습집합을 얼마나 잘 Fitting하는가를 나타낸 것이다. 만일 여기에 ϑ_s 를 추가하게 되면 적합도 함수는 다음 (4)식이 된다.

$$\vartheta = \vartheta_{MSE} + \lambda \vartheta_s \quad (4)$$

여기서 λ 는 ϑ_s 의 기여도를 결정하는 상수인데, λ 가 과도하게 크면 학습집합을 제대로 Fitting하지 못하며, 반대로 너무 작으면 Overfitting 현상이 발생한다. 따라서 적절한 λ 값의 결정은 매우 중요한데, 이것은 일반적으로 대단히 어려운 일이다. 노이즈(Noisy) 데이터의 Smooth Fitting 시 λ 값을 결정하기 위해서 사용되는 방법 중의 하나가 L-curve 기법이다[12]. 이 방법은 다수의 λ 값에 대한 ϑ_{MSE} 와 ϑ_s 의 값을 (4)식의 최적화를 통하여 구한 후 이것을 로그스케일로 그려 보면 'L'자 모양의 커브가 나타나고, 최적의 λ 는 'L'자의 코너 즉 L-curve의 곡률이 급격히 변화하는 지점에 해당하는 λ 가 된다. 이것은 또한 ϑ_{MSE} 와 ϑ_s 가 동시에 최소화되도록 λ 의 값을 결정해준다는 것과 같은 의미가 된다. 학습 데이터의 노이즈 포함 유무에 관계없이 GP에서는 L-curve방법이 거의 사용 불가능하지만[13], 동시에 ϑ_{MSE} 와 ϑ_s 가 최소화 되도록 고려한 (5)식과 같은 적합도 함수를 정의할 수 있다.

$$\vartheta = \vartheta_{MSE} + \vartheta_{MSE} \vartheta_s (\vartheta_{MSE} - 1/2 \vartheta_s)^2 \quad (5)$$

(5)식의 가장 큰 특징은 λ 가 식에 포함되어 있지 않다는 것이며, $\vartheta_{MSE} = 1/2 \vartheta_s$ 이면 (3)식과 동일하게 된다. 만일 ϑ_{MSE} 이 감소하고 그 대신 ϑ_s 이 증가하면 $(\vartheta_s - 1/2\vartheta_s)^2$ 에 의해 ϑ 의 값이 증가하게 된다. 즉 ϑ_{MSE} 와 ϑ_s 가 동시에 작아지도록 유도할 수 있다는 것을 의미한다. 그러나 ϑ_{MSE} 의 값이 매우 작으면 $\vartheta_{MSE}\vartheta_s$ 도 따라서 작은 값이 되므로 $(\vartheta_s - 1/2\vartheta_s)^2$ 이 기여하는 바가 감소하게 된다. 이런 경우 Overfitting 현상을 보일 수 있지만 많은 수치 실험의 결과를 관찰해 볼 때 대체로 그 효과는 미미하기 때문에 본 논문에서는 (5)식을 사용하는 것 이외에 Overfitting 방지를 위한 또 다른 전략은 사용하지 않았다.

4. 응용 예

Table 1은 본 논문의 응용 예에 공동적으로 사용된 DDBS GP의 주요 파라미터이다.

4.1 Rosenbrock's Function

Rosenbrock's valley function은 일명 바나나 함수라고도 불리는데, (6)식으로 정의되어 진다.

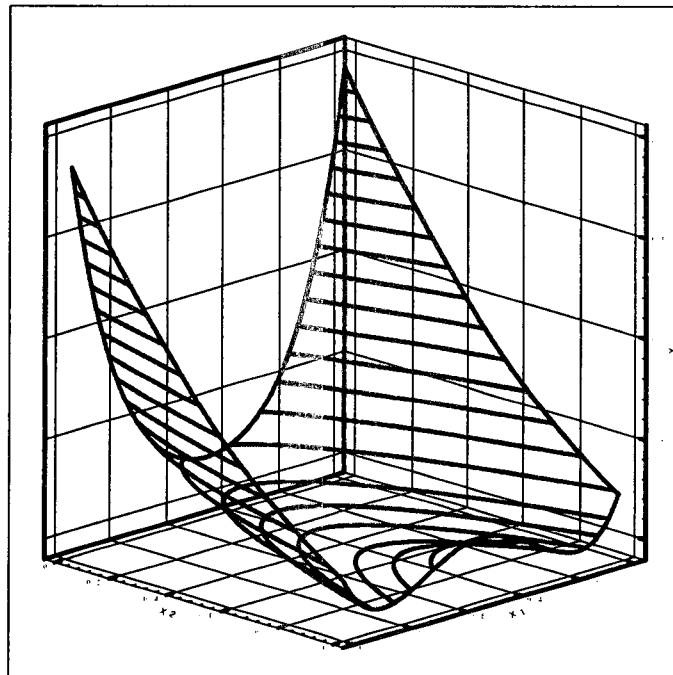
$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (6)$$

$$-2 \leq x_i \leq 2, i = 1, 2$$

학습데이터는 Grid 타입의 3x3=9 개를 사용하였고, 예측 성능을 측정하기 위하여 200x200 개의 테스트 데이터를 사용하였다. Fig. 2.a-c는 테스트 데이터에 대하여, 각각 (3)과 (5)식을 적합도 함수로 사용하여 구한 최적 f_{GP} 의 결과를 가시화한 것이다. Fig.2에서 확인할 수 있듯이 (3)식을 사용한 일반 GP의 경우는 원래 함수의 모양새를 크게 벗어나고 있는 반면 DDBS가 적용된 경우는 단지 9 개의 학습 데이터만을 사용하여도 그 결과가 원래 함수에 근접하고 있다. 일반 GP가 사용된 경우, 학습과 테스트에 대한 MSE(Mean Squared Error)는 각각 4.995E-11 이고 2.977E-1 이다. 반면 DDBS가 적용된 GP에 대해서는 각각 1.945E-6, 그리고 1.592E-2이다. Fig.3은 DDBS가 적용된 최적의 트리를 보인 것이다. "[]"안의 수치는 추정된 가중치이며 이 값은 노드의 출력 값에 항상 곱해지게 된다. atan은 \tan^{-1} 이다.

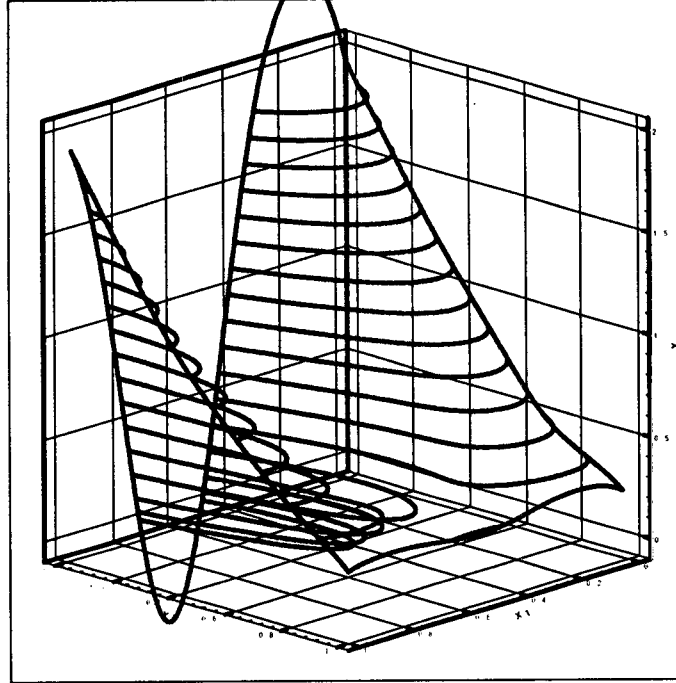
Table 1. Parameters used in the GP algorithm with DDBS.

Population size	500
Maximum allowable # nodes of trees	50
Allowable maximum generation	20
Selection method	Tournament with 30 trees
Reproduction probability	0.15
Crossover probability	0.7
Mutation probability	0.15
p in (2)	2
q in (2)	40

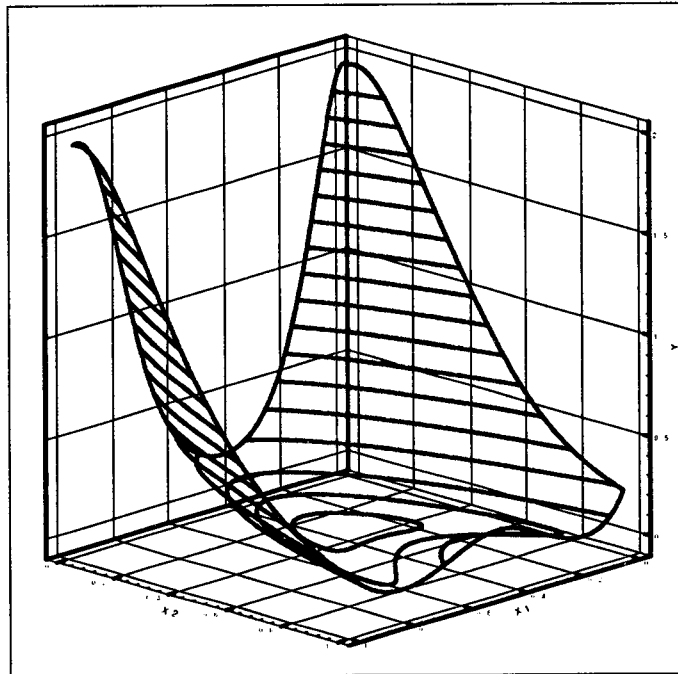


a. Rosenbrock's function.

Fig.2. The results of DDBS GP



b. GP without DDBS.



c. GP with DDBS.

Fig.2. The results of DDBS GP(Continued).

[1.962792968749992](cos[1.626171874999999](sin[1.909008789062494](+[1.196484375000026](sinh[1.218359374999995](sin[1.631445312500019](+[1.311450195312508](sinh[0.598541259765632](exp[-0.416339111328120]x2:5))[0.964062499999992](-2.348425292968752)(atan[0.174975585937501](exp[2.048852539062501]x1)))[1.799218749999982]x1))))[0.649609374999994](sin[2.11743164062505](+[1.271557617187505](sinh[0.699969482421882](exp[-0.378204345703118]x2))[1.384375000000000](-[2.346679687500002](atan[0.174975585937501](exp[2.061254882812501]x1))1.814575195312520)x1))))))

Fig.3. The optimal GP tree for the Rosenbrock's function.

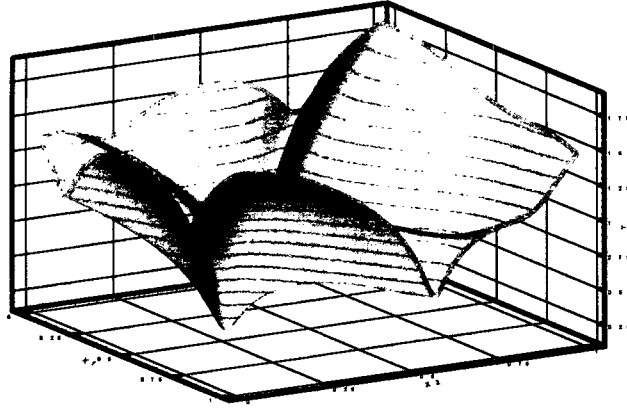
4.2 Goldstein-Price Function

$$\begin{aligned}
 f(x_1, x_2) &= (1 + (x_1 + x_2 + 1))^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \\
 &\cdot (30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) \\
 &- 2 \leq x_i \leq 2, i = 1, 2
 \end{aligned} \tag{7}$$

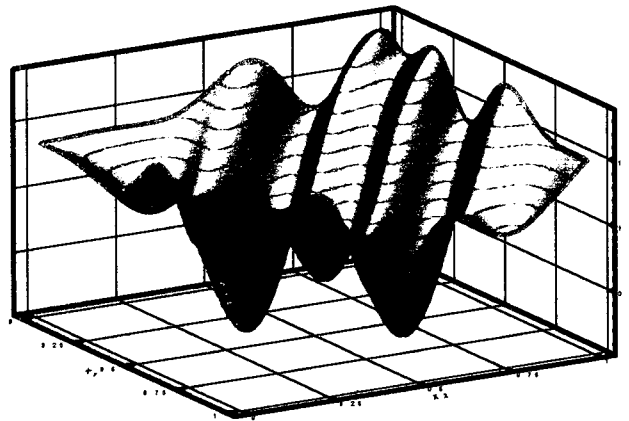
(7)식에 나타나 있는 Goldstein-price function은 비선형성이 매우 크고, 동시에 그 값의 변화가 대단히 크기 때문에 다음과 같은 로그 스케일을 사용하였다.

$$f_{\log}(x_1, x_2) = \log(1 + f(x_1, x_2)) \tag{8}$$

학습 데이터는 5x5 Grid 타입으로 생성하였고, 테스트 데이터는 200x200개이다. Fig. 4.a-c는 테스트 데이터에 대하여, 각각 (3)과 (5)식을 적합도 함수로 사용하여 구한 최적의 f_{GP} 의 결과를 가시화한 것이다.

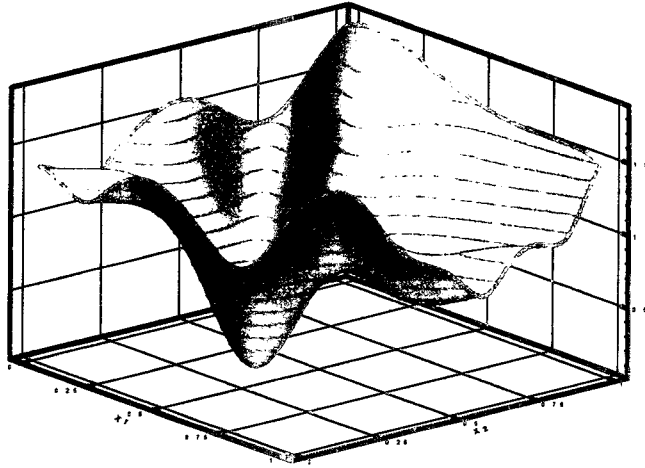


a. Goldstein-price function.



b. GP without DDBS.

Fig.4. The results of DDBS GP for the log-scaled Goldstein-price function(continued).



c. GP with DDBS.

Fig.4. The results of DDBS GP for the log-scaled Goldstein-price function.

일반 GP가 사용된 경우, 학습과 테스트에 대한 MSE는 각각 $1.736E-3$ 이고 $4.290E-2$ 이다. 반면 DDBS가 적용된 GP에 대해서는 각각 $1.468E-4$, 그리고 $1.147E-2$ 이다. Fig.4.b를 살펴보면 Goldstein-price function의 전체적인 윤곽은 유사하나 심한 Overfitting 현상을 보이고 있다. 반면 Fig.4.c는 Goldstein-price function과 비교하면 약간의 지역적인 차이는 있으나 전체적인 모양새는 일치하고 있으며, Overfitting 현상을 보이고 있지 않다. 본 예제를 통하여 DDBS GP는 Overfitting이 없이 비선형성이 큰 함수도 근사화 할 수 있음을 다시 확인할 수 있다.

4.3 Two-Bar Truss

Fig.5는 Two-bar truss를 보여주고 있고, (9)식은 트러스의 무게, (10)식은 좌굴에 관련된 구속조건, 그리고 (11)식은 항복응력에 관련된 구속조건 식이다[14].

$$f_1(x_1, x_2) = 2\rho\pi x_1 T(B^2 + x_2^2)^{1/2} \quad (9)$$

$$f_2(x_1, x_2) = \frac{\pi^2 E(x_1^2 + T^2)}{8(B^2 + T^2)} - \frac{P(B^2 + T^2)^{1/2}}{\pi T x_1 x_2} (\geq 0) \quad (10)$$

$$f_3(x_1, x_2) = \sigma_y - \frac{P(B^2 + T^2)^{1/2}}{\pi T x_1 x_2} (\geq 0) \quad (11)$$

여기서 $x_1 = D, x_2 = H$ 이며, $0.5in \leq x_1 \leq 5in, 5in \leq x_2 \leq 50in$ 이다. 또한 $B = 30in, T = 0.1in, 2P = 66,000psi, \rho = 0.3 lbs/in^3, \sigma_y = 60,000 psi$ 이다.

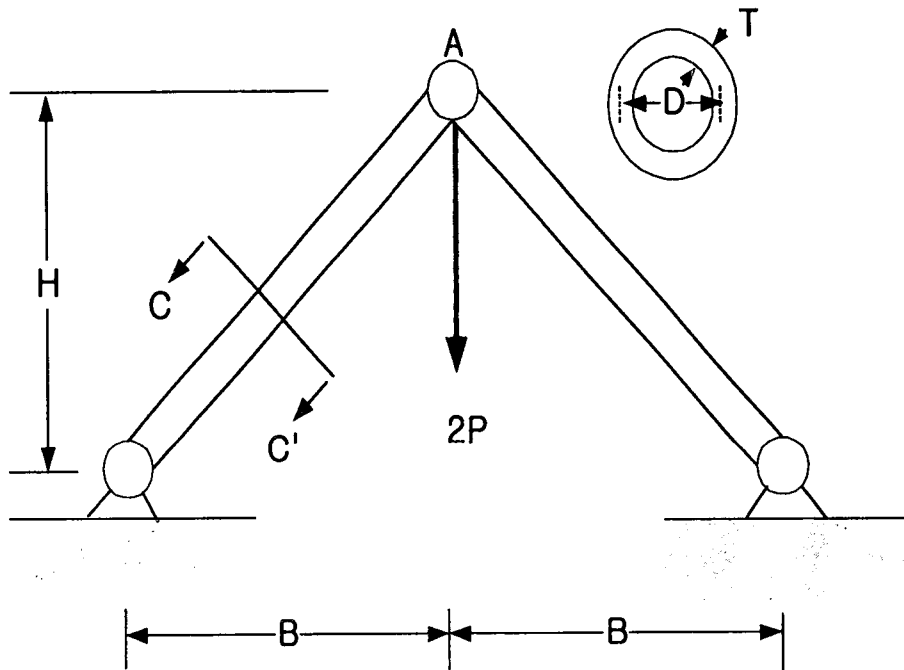
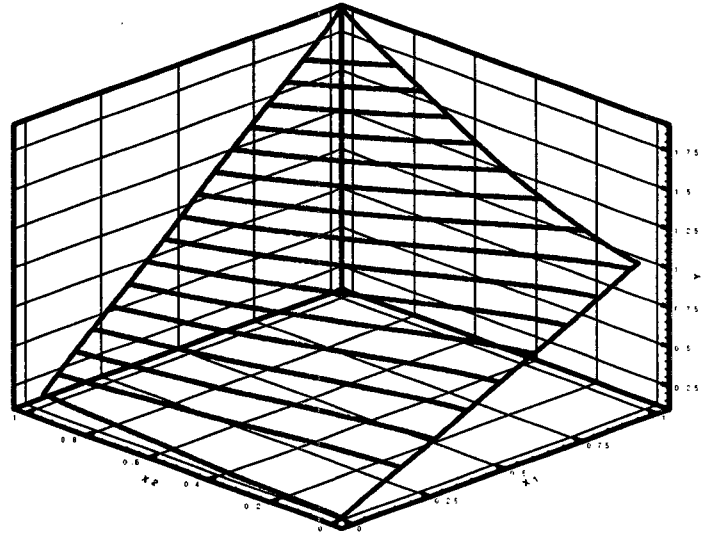
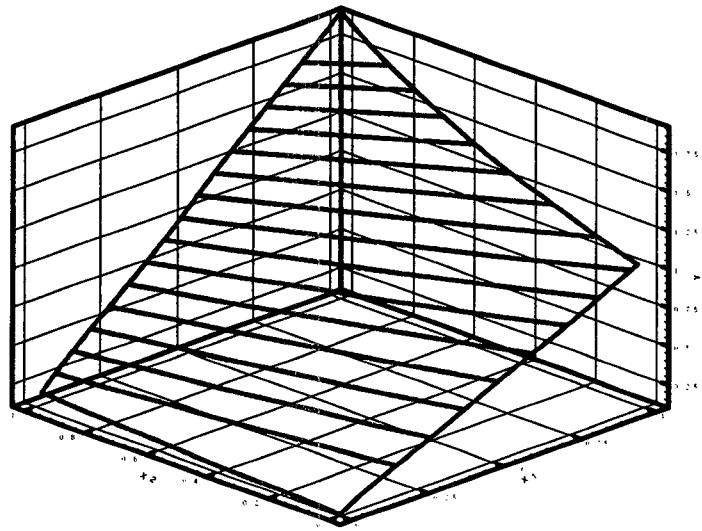


Fig. 5. Two-Bar Truss

(9)-(11)식을 근사하기 위해서 학습 데이터는 3x3 Grid 타입으로 생성하였고, 테스트 데이터는 200x200개이다. Fig.6-8은 DDBS가 채용된 GP의 결과를 가시화한 것이다. Table 2는 f_1, f_2 그리고 f_3 에 대한 DDBS GP의 학습 및 테스트 오류를 기록한 것이다.

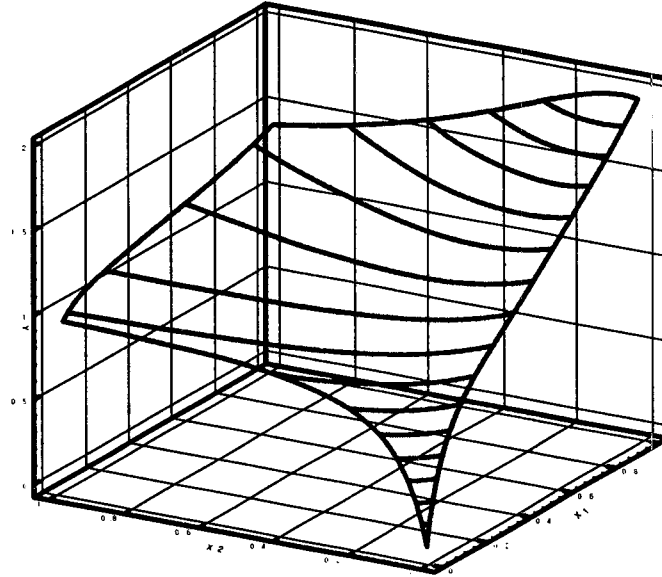


a. The original function.

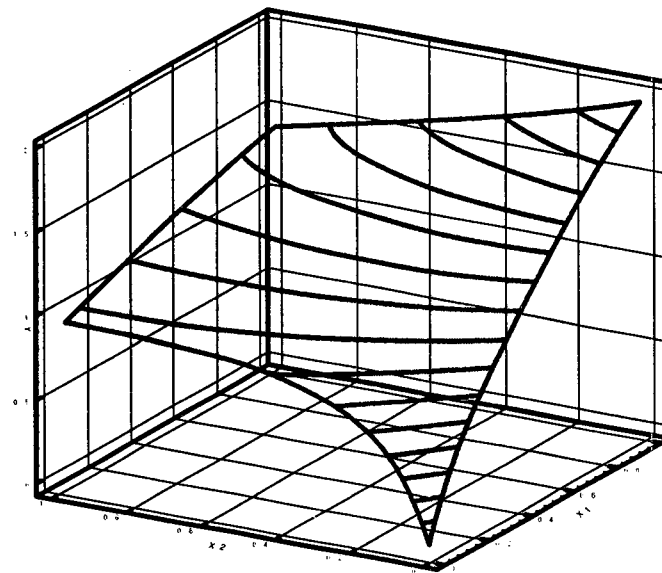


b. The results of GP with DDBS.

Fig.6. The weight of two-bar truss.

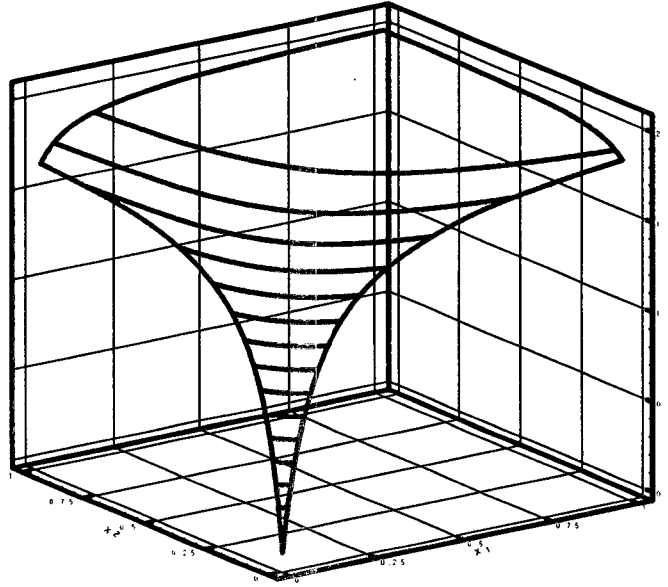


a. The original function.

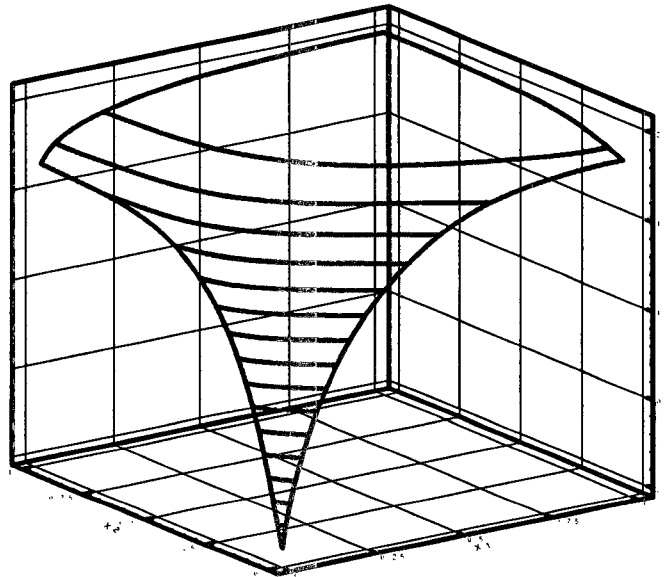


b. The results of GP with DDBS.

Fig.7. The bucking constraint of two-bar truss.



a. The original function.



b. The results of GP with DDBS.

Fig. 8. The stress constraint of two-bar truss.

Fig.6-8.에서 관찰할 수 있듯이 Two-bar truss 문제는 비교적 비선형성이 크지 않기 때문에, 단지 9 개의 학습 데이터로 DDBS GP는 목적함수와 구속조건을 거의 정확하게 모델링 하고 있다.

Table 2. The learning and test error of GP trees for two-bar truss.

GP tree with DDBS	Learning error(MSE)	Test error(MSE)
f_1	2.518E-6	2.29E-4
f_2	1.478E-7	3.450E-3
f_3	5.345E-9	3.394E-3

4.4 I-Beam

Fig.9는 I-Beam을 나타낸 것으로 설계변수의 개수는 4 개이다[15]. 목적함수는 보의 단면적인데 식이 너무 간단하여 본 논문에서는 근사화에서 제외 시키고 응력에 관련한 구속 조건식 (11)만을 DDBS GP로 근사화 시켰다.

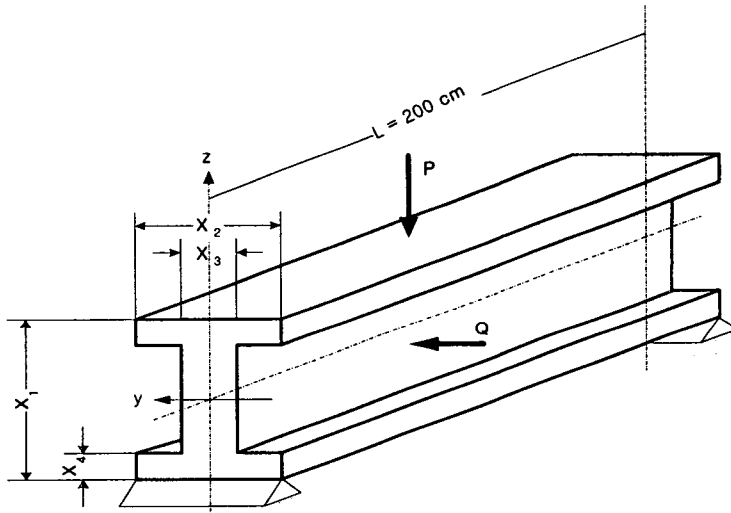


Fig. 9. I-Beam.

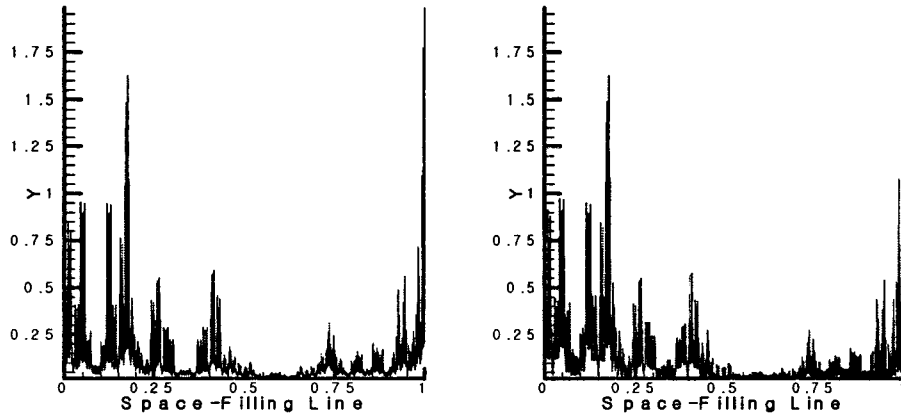
$$f(x_1, x_2, x_3, x_4) = \frac{M_y}{Z_y} + \frac{M_z}{Z_z} - \sigma_a (\leq 0) \quad (11)$$

$$\text{where } M_y = \frac{P}{2} \times \frac{L}{2}, \quad M_z = \frac{Q}{2} \times \frac{L}{2},$$

$$Z_y = \frac{1}{6x_1} \{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]\},$$

$$Z_z = \frac{1}{6x_2} \{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3\}$$

여기서 $10\text{cm} \leq x_1 \leq 80\text{cm}$, $10\text{cm} \leq x_2 \leq 50\text{cm}$, $0.9\text{cm} \leq x_3 \leq 5\text{cm}$,
 $0.9\text{cm} \leq x_4 \leq 5\text{cm}$, $\sigma_a = 16\text{kN/cm}^3$ 이다.



a. The original function.

b. The Results of GP with DDBS.

Fig. 10. The results of DDBS GP plotted by using the space-filling curve. The y-Axis represents the value of the constraint function of the I-Beam.

학습집합은 $3 \times 3 \times 3 \times 3 = 81$ 개의 Grid 타입이며, 테스트 데이터는 $14 \times 14 \times 14 \times 14 = 38416$ 개 이다. BBDS GP에서 생성된 최적 트리의 학습과 테스트 오류는 각각 $1.225\text{E}-4$, $4.126\text{E}-3$ 이다. 4 차원 문제이기 때문에 GP의 결과를 직접적으로 가시화하는 것은 불가능하지만 n 차원을 1 차원으로 일대일 매핑(Mapping)할 수 있는 공간채우기 곡선(Space-Filling Curve)의 일종인 Sierpinski's Curve[16]를 사용하여 Fig. 10에 그 결과를 가시화 하였다. 즉 4 차원 테스트 데이터 $\vec{X}_i = (x_{i,j})_{j=1}^4$ 를 $[0, 1]$ 의 1 차원 실수 라인(Line)의 한 점

으로 일대일 매핑 시키고 이에 대응하는 GP 트리의 결과를 도시화한 것이다. DDBS GP 트리의 학습과 테스트 오류는 각각 $1.227E-4$, $4.126E-3$ 이다. Fig.10을 살펴보면 역시 DDBS GP는 원래 함수의 모양에 크게 근접하는 최적의 트리를 생성함을 확인할 수 있다.

5. 결론 및 추후 연구 과제

본 논문에서는 비선형 함수 근사화에 우수한 성능을 보이는 GP를 응답면 모델링에 적용하기 위한 방안을 제시하였다. 응답면의 모델링은 최소한의 학습 데이터를 요구하므로, 이로 인하여 GP는 보통 Overfitting된 트리를 생성하게 된다. 이런 문제점을 극복하고자 제시한 방법이 DDBS이며, 4 절의 결과를 판단해보면 DDBS는 Overfitting 방지에 탁월한 성능을 보이고 있으며, 동시에 모델링하고자 하는 함수의 전체적인 특징을 GP 트리가 유지하는데 도움을 주는 것으로 보여 진다. 또한 DDBS를 적용하기 위한 적합도 함수를 제시하였는데, 이것은 σ 의 값을 산정할 필요가 없으며, σ_{MSE} 와 σ_s 가 동시에 최소화될 수 있는 형태를 취하고 있다. DDBS의 가장 큰 단점은 기존의 근사화 방법과 비교하여 많은 계산량을 요구한다는 것이다.

설계변수의 차원이 증가할수록 기본적으로 필요로 하는 학습집합의 크기는 급격히 증가한다(Curse of Dimensionality). 이것은 응답면 모델링에 있어서 가장 큰 문제점 중에 하나인데, 고차원의 문제에 DDBS GP를 적용하기 위한 방안을 다음 두 가지로 생각해 볼 수 있다.

첫째, 설계변수의 차원에 비하여 매우 작은 학습집합으로 응답면을 모델링 한 후, 이로부터 최적 점을 근사적으로 구한다. 이 점에서 해석을 수행하여 학습 데이터를 추가하고 다시 응답면을 모델링 한다. 이 과정을 반복하면 관심이 있는 설계공간의 영역 즉 최적 점이 존재하리라고 판단되는 영역은 높은 정확도를 갖고 근사화 될 수 있다.

둘째, 전체 설계공간에 대한 응답면을 모델링 하는 것이 아니라, 탐색이 시작되는 근처만 크기가 작은 학습 집합으로 근사화하고, 이로부터 탐색 점을 최적 점이 있는 위치로 이동시키고, 현재 이동된 위치에서 근사식의 정확도가 떨어진다고 판단되면, 근처의 몇 개의 점에 대한 해석을 수행하여 새로운 학습집합을 얻고, 다시 응답면을 모델링 한다. 최적 점이 발견될 때까지 이 과정을 반복한다.

위의 두 가지 시나리오는 모두 초기에 찾아진 GP 트리의 구조를 반복적으로 변경하는 과정을 요구하는데, 전체 진화 과정을 다시 반복하는 것은 너무나 많은

시간과 계산량이 필요하게 된다. 따라서 전체 진화 과정을 반복하지 않고 GP 트리의 구조를 최적화할 필요성이 대두된다. 이를 위하여 몇 가지 전략을 수립할 수 있는데, 현재 이에 대한 연구를 수행하고 있다.

참고문헌

- [1]Simpson, T.W., Allen, J.K., and Mistree, F., "Spatial Correlation and Metamodels for Global Approximation in Structural Design Optimization", Proc. Of DETC98, ASME, 1998.
- [2]Malik, Z., Su, H., and Nelder, J., "Informative Experimental Design for Electronic Circuits", Quality and Reliability Engineering International, vol.14, pp.177-188, 1986.
- [3]Alotto, P., Gaggero, M., Molinari, G., and Nervi, M., "A Design of Experiment and Statistical Approach to Enhance the Generalized Response Surface Method in the Optimisation of Multi-Minimas", IEEE Transactions on Magnetics, Vol.33, No.2, pp.1896-1899, 1997.
- [4]Ishikawa, T. and Matsunami, M., "An Optimization Method Based on Radial Basis Function", IEEE Transactions on Magnetics, Vol.33, No. 2/II, pp.1868-1871, 1997.
- [5]Koza, J.R., Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, 1992.
- [6]Toropov, V.V. and Alvarez, L.F., "Approximation Model Building for Design Optimization using Genetic Programming Methodology", AIAA Paper 98-4769, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St.Louis, Part1, pp.490-498, 1998.
- [7]Toropov, V.V., and Alvarez, L.F., "Application of Genetic Programming and Response Surface Methodology to Optimization and Inverse Problems", In: Tanaka, M. and Dulikravich, G.S.(eds.), International Symposium on Inverse Problems in Engineering Mechanics(ISIP'98), Nagono City, Japan, pp.551-560, 1998.
- [8] Yeun, Y.S., Lee, K.H., and Yang, Y.S., "Function Approximations by Coupling Neural Networks and Genetic Programming Trees with Oblique Design Trees", AI in Engineering, Vol.13, No3, 1999.

- [9] Yeun, Y.S., Suh, J.C., and Yang, Y.S., "Function Approximation by Superimposing Genetic Programming Trees: with Applications to Engineering Problems", *Information Sciences*, Vol.122, Issue2-4, 2000.
- [10] 연윤석, "가중 선형 연상기억을 채용한 유전적 프로그래밍과 그 공학적 응용", *한국 CAD/CAM 학회 논문집*, Vol.3, No.1, pp.57-67, 1998.
- [11] Hooke, R. and Jeeves, T.A., "Direct search solution of numerical and statistical problems", *J. of ACM*, Vol.8, No.4, pp. 212-229, 1961.
- [12] Hansen, P.C. and O'Leary, D.P., "The Use of the L-curve in the Regularization of Discrete Ill-Posed Problems", *SIAM J. Sci. Stat. Comput.* Vol.14, pp. 1487-1503, 1993.
- [13] Yeun, Y.S., Lee, K.H., Han, S.M., and Yang, Y.S., "Smooth Fitting with a Method for Determining the Regularization Parameter under the Genetic Programming Algorithm", *Information Sciences*, Vol.133, Issue3-4, 2001.
- [14] Schmit, L.A., "Structural Synthesis-Its Genesis and Development", *AIAA Journal*, Vol.19, No, 10, pp. 1249-1263, 1981.
- [15] Osyczka, A., "Multicriteria Optimization for Engineering Design", In: Gero, J. S. (eds.), *Design Optimization*, pp. 193-227, Academic Press, Inc., 1985.
- [16] Perez-Cortes, J.C., Vidal, E. "The Extended General Spacefilling Curves Heuristic", *Int. Conf. on Pattern Recognition ICPR-98*, Brisbane, Australia, 1998.

Response Surface Modeling by Genetic Programming I: A Directional Derivative-Based Smoothing Method

Yun Seog, Yeun, Wook Rhee

Abstract

This paper introduces the genetic programming algorithm(GP), which can approximate highly nonlinear functions, as a tool for the modeling of response surfaces. When the response surfaces is approximated, the very small or minimal learning set should be used, and thus it is almost certain that GP trees will show overfitting that must be avoided at all costs. We present a novel method, called DDBS(Directional Derivative-Based Smoothing), which very effectively eliminates the unwanted behaviors of GP trees such as large peaks, oscillations, and also overfitting. Four illustrative numerical examples are given to demonstrate the performance of the genetic programming algorithm that adopts DDBS.

Key Words: response surface, genetic programming, overfitting, DDBS

◆ 저자소개 ◆

연운석 (Yun-Seog, Yeun)



서울대학교 대학원 조선해양공학과 박사를 취득하였으며, 현재 대진대학교 기계설계공학과 부교수로 재직 중이다. 관심분야는 Artificial Intelligence in Engineering Design, Evolutionary Computation 이다

E-mail : yeonyun@road.daejin.ac.kr

Tel : 031-539-1971(연구실), 02-470-4851(Home)

이욱 (Wook, Rhee)



숭실대학교 대학원 전기공학과 공학박사학위를 취득하였으며, 현재 대진대학교 전기공학과 부교수로 재직 중이다. 관심분야는 산업전자 응용제어, Evolutionary Computation 이다.

E-mail : rheewook@road.daejin.ac.kr

Tel : 031-539-1911