

다중 인공 신경망과 유전적 프로그래밍의 복합적 접근에 의한 공학설계 시스템의 개발

이경호*·연운석**

A Hybrid Approach to Multiple Neural Networks and Genetic Programming: A Perspective of Engineering Design Application

Kyung Ho Lee*·Yun Seog Yeun**

요 약

본 논문에서는 경사진 의사결정 트리(oblique decision tree)에 의해 몇 개의 영역으로 분할된 입력공간(input space)에서 우수한 성능을 발휘할 수 있도록 유전적 프로그래밍 트리들(genetic programming trees)과 연합된 다중 인공신경망 시스템을 개발하였다. 다중 인공신경망인 지역 에이전트들(local agents)은 분할된 영역을 책임지며, 유전적 프로그래밍 트리들로 구성된 경계 에이전트들(boundary agents)은 분할된 영역의 경계부분만을 담당하게 된다. 본 연방 에이전트 시스템을 이용하여 설계 초기단계의 정보 제한성을 극복하고, 선박 초기설계 단계에서 선박 중앙부 형상설계를 수행하여 범용 설계 시스템으로서의 유용성을 검증하였다.

Keywords : Federated Agents, Neural Network, Genetic Programming, Oblique Decision Tree

* 한국기계연구원 선박해양공학연구센터 선임연구원

** 대전대학교 기계설계학과 조교수

1. 서 론

함수근사(function approximation)의 전형적인 작업은 주어진 학습 데이터 집합의 정보를 이용하여 모든 가능한 입력 값에 대한 각각의 출력 값을 정확히 매핑시켜 주는 적절한 함수를 찾는 것이다. 이를 위해 nearest-neighbor 알고리즘, 통계학 모델, 사례기반 추론, feedforward neural network, 등의 다양한 방법론과 알고리즘이 개발되어 왔다. 그러나 하나의 방법론만을 사용하여 복잡한 공학 문제를 해결하기에는 많은 어려움이 수반되는데, 현재까지 이를 극복하기 위하여 여러 가지 방법론이 제시되어 왔다. 본 연구에서는 다음의 두 가지 관점을 적절히 결합한 새로운 접근을 시도하였다.

첫째는 서로 다른 원리에 기초한 서로 다른 접근법의 결과를 융합하는 방법이다(Zhang, Mesirov & Waltz, 1992). 둘째는 모듈화된 접근법이다. 즉 각각의 분할된 입력공간(input space)을 담당하는 generalizer를 구축하여 전체적으로 모듈화된 구조를 형성한다. 여기서 발생하는 문제점은 두 가지인데, 즉 입력공간의 적절한 분할 방법과 최적의 결과를 얻기 위한 각 generalizer들의 결합 전략 등이다. 모듈화된 다중 인공신경망에 대한 많은 연구가 최근까지 수행되고 있으며, 신경망을 분할하고 재 합성하는 다양한 접근 방식이 제시되고 있다(Happel & Murre, 1994; Jacobs & Jordan, 1991; 1993; Smieja, 1996; Waibel, 1989). 이러한 접근 방식을 크게 두 가지로 대별해 볼 수 있는데, 첫째가 입력공간(input space)을 몇 개의 공간으로 세분화하는 것이고 둘째가 그렇지 않은 경우이다.

본 연구에서는 입력공간을 자동적으로 세분화하는 방법은 다루지 않고, 다중 신경망의 결과

를 조합하여 최적의 결과를 도출할 수 있는 방안을 모색하였다. 이를 바탕으로 시스템의 성능을 보다 개선하기 위해서 다중 인공신경망과 유전적 프로그래밍의 트리(genetic programming tree : GP) (Koza, 1992; 1994)를 병행하여 사용할 수 있는 연방 에이전트 시스템(federated agent system)을 개발하였다. 아울러 본 시스템을 산적화물선(bulk carrier ship)의 형상설계에 적용하였고 그 결과를 검토하였다.

2. 지역 에이전트와 경계 에이전트에 의한 연방 에이전트 시스템

본 시스템은 분할된 각각의 입력공간을 담당하고 있는 인공신경망인 지역 에이전트(local agent)들로 이루어진 다중 인공신경망 시스템이다. 새로운 데이터가 입력되면 먼저 이에 해당하는 올바른 지역 에이전트를 찾는 것이 선행되어야 하는데, 이를 위해서 경사진 의사결정 트리(oblique decision tree) 방법에 기초한 OC1을 이용하였다(이경호 외, 1996; Murthy, 1994). 즉 입력 데이터가 주어지면, 이 데이터가 속한 분할된 입력공간 중의 하나를 OC1을 사용하여 예측한 후, 이 입력공간을 담당하고 있는 지역 에이전트를 수행하여 원하는 출력 값을 얻게 된다. 이 방법은 개념적으로 매우 단순하며 구현하기가 용이하나 다음과 같은 문제점을 갖고 있다. 입력 데이터가 분할된 입력공간들의 경계 근처에 위치하고 있다면, 의사결정 트리(decision tree)에 의해 지역 에이전트를 선정함에 있어 오류를 범할 가능성이 매우 높아진다. 이런 경우 지역 에이전트의 선정 오류로 인해 입력공간 전체를 하나의

신경망 모델로 학습시킨 경우보다 더 좋지 않은 결과를 가져올 수 있다. 이 문제를 극복하기 위해서 본 연구에서는 경계 영역만을 따로 담당하는 경계 에이전트(boundary agent)를 도입하였다. 경계 에이전트는 신경망 모델에 의해 구성될 수도 있다(이경호 & 연운석, 1997; Yeun & Lee, 1997). 그러나 경계 영역은 불연속이거나 몇 개의 다른 패턴이 동시에 존재할 수 있어, 신경망의 예측 성능은 크게 저하될 수 있다. 따라서 경계 에이전트로서 신경망 대신에 유전적 프로그래밍의 functional tree(이하 GP트리)를 사용하는 방안을 모색하였다. 본 논문에서는 GP 트리로 구성된 경계 에이전트, 인공신경망으로 구성된 지역 에이전트 그리고 경사진 의사결정 트리를 기반으로 한 퍼실리테이터(facilitator)가 함께 연방구조(federated architecture)를 구성하게 된다. 퍼실리테이터는 주어진 입력 정보를 바탕으로 의사결정 트리를 이용하여 적절한 지역 에이전트, 경계 에이전트를 선택해 주는 역할을 한다. Fig.1은 다중 인공신경망과 GP의 복합적 접근에 의한 연방 구조를 보여 주고 있다.

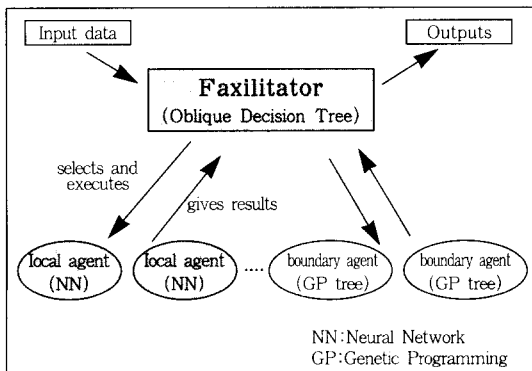


Fig.1 Federated Architecture of Local and Boundary Agents

3. 유전적 프로그래밍

유전적 프로그래밍 기법은 기존의 유전적 알고리즘에서 개체(individuals)로서 이진 스트링을 사용하는 대신 컴퓨터 프로그램을 표현하기 위한 트리 구조를 사용한다. 이 프로그램 트리는 함수 집합(function set)과 터미널 집합(terminal set)에 정의된 원소들의 조합으로 구성된다. 여기에서도 재생산(reproduction), 교배(crossover), 돌연변이(mutation), 등의 유전적 연산이 수행된다. 또한 연산 과정중의 트리의 크기와 구조는 유전적 연산에 의해 동적으로 변화하는데, 이렇게 얻어진 트리는 적합도(fitness) 함수에 의해서 평가된다.

3.1 GP 트리의 가중치 도입

GP를 실수 함수(real-valued function)의 근사에 사용하기 위해서는 수치 파라미터 추정 기법이 필요하다. 그것은 GP 알고리즘 자체가 원래 수치 계수 추정을 다루고 있지 않기 때문이다. 수치 계수 추정을 위해 제시된 방법 중 하나로서 Watson과 Parmee는 트리의 터미널 집합에 실수들로 구성된 집합을 포함시키고, 진화 과정중 적절한 실수를 선택하는 방식을 적용하여 유체가 흐르는 파이프에 대한 turbulent flow의 마찰계수 공식을 성공적으로 찾아냈다(Watson & Parmee, 1996). Sharman et al.는 가중치 산정을 위해서 트리의 모든 노드에 부과된 gating factor를 도입하였다. 이것은 수치 파라미터를 의미하며 적절한 값을 추정하기 위하여 비선형 최적화 방법인 SA(simulated annealing)를 적용하였다. 이 접근 방식은 비록 정확한 수치 파라미터의 값을 산정할 수 있으나 방대한 계산량

이 소요되는 문제점이 있다. GP에서 가중치가 반드시 전역 최적값(global optimum value)일 필요는 없다. 따라서 본 논문에서는 GP트리의 특정 함수와 터미널 노드에 인덱스 가중치(indexed-weight)를 첨가하고 이 값을 산정하기 위해서 언덕 오르기(hill-climbing) 탐색법을 사용하였다. Fig.2는 함수와 터미널을 포함한 GP 트리에 인덱스 가중치가 부여된 예를 보여주고 있다. 여기서 함수 와 터미널 w_i 에서 i 는 인덱스를 w_i 는 인덱스 가중치를 의미한다. 인덱스는 정수값을 취하고 가중치는 실수값을 갖는다. 만일 가중치들이 같은 인덱스 값을 가지고 있다면 그것이 서로 다른 함수나 터미널에 속한 것이라도 언덕 오르기 탐색 시 동시에 수정된다. 그러나 같은 인덱스를 갖는 인덱스 가중치라도 항상 동일한 값을 갖는 것은 아니다. 즉 각각의 가중치가 서로 다른 값으로 초기화되었다면 언덕 오르기 탐색에 의해 조정된 값도 서로 다른 값을 갖게 된다. 적절한 가중치를 찾기 위해 언덕 오르기 탐색을 적용할 때 탐색 중에 조정되는 가중치의 수는 인덱스의 수와 같게 되고, 일반적으로 인덱스의 수는 GP트리의 노드 수보다 작기 때문에 인덱스 가중치를 도입함으로써 가중치에 대한 탐색공간이 줄어들게 된다. Fig.2의 예에서 자세히 살펴보면 서로 다른 인덱스의 수는 6개이고, GP트리 노드의 총 수는 19개이다. 여기서 6개의 가중치만 언덕 오르기 탐색에 의해 조정된다. 앞서 언급한 Sharman et al.의 접근 방법과는 달리 조정 가능한 가중치의 수가 GP트리의 크기에 영향을 받지 않는다. 따라서 GP트리의 크기가 커지더라도 가중치를 결정하는데 필요한 계산비용은 크게 증가하지 않는다.

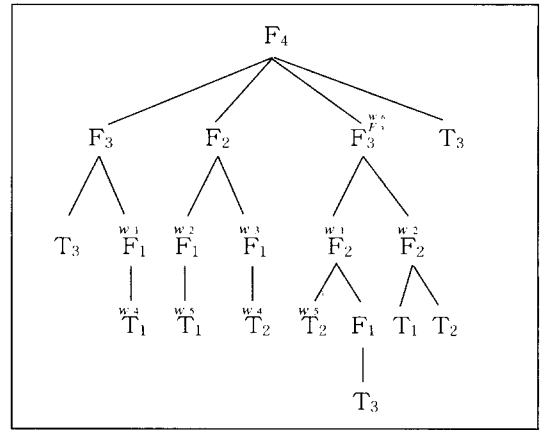


Fig.2 GP tree with indexed weights

3.2 적합도 함수(Fitness Function)

m 차원 입력벡터 X_i 와 1차원 출력벡터 y_i 로 구성된 학습 데이터 세트(X_i, Y_i)를 고려할 때, 한 세대에서의 각각의 GP트리에 대한 적합도는 다음과 같이 결정된다.

$$f = \sum_{i=1}^n |y_i - y(X_i)|^2 \quad (1)$$

여기서, f 는 적합도, n 은 샘플의 수, $y(X_i)$ 는 입력값 X_i 에서의 GP트리의 해(solution), 그리고 y_i 는 그때 요구되는 엄밀해(exact solution)를 각각 의미한다. 그러나 이러한 형태의 적합도 함수는 GP트리의 일반화(generalization)를 보장하지는 못한다. 즉, 테스트 데이터가 학습된 영역에 속하지 않을 때 좋은 결과를 주지 못하는 경우가 많다. 이러한 문제를 해결하기 위하여 본 연구에서는 (1)식에다가 다음과 같이 추가적인 항목을 첨가하였다.

$$f = \sum_{i=1}^n |y_i - y(X_i)|^2 + \alpha \left| \frac{\sum_{i=1}^n |Y_i|}{\sum_{i=1}^n |y_i|} - 1.0 \right| \quad (2)$$

where, $Y_i = \sum_{j=1}^M y(Z_{ij}) / M$

여기서 α 는 두 번째 항의 기여도를 나타내는 상수이고, $y(z_{ij})$ 는 X_i 와 ΔX_{ij} 와의 합으로 나타나는 z_{ij} 벡터에서의 GP트리의 해이다. Y_i 를 계산하기 위해서는 우선 X_i 주변 M개의 점 즉 M개의 z_{ij} 를 구해야 한다. z_{ij} 는 X_i 에 난수(random number)로 이루어진 벡터를 합하여 얻게 되는데, 이때 벡터 ΔX_{ij} 의 크기 $|\Delta X_{ij}|$ 는 미리 주어진 상수 값 θ_i 보다 같거나 작아야 한다. θ_i 는 모든 X_i 에 대하여 필요하므로 학습과정 전에 θ_i 를 결정하기 위한 전처리 과정이 요구된다. X_i 와 가장 가까운 X_p 를 선정하고 이 두 벡터들의 거리(d), 즉 $d = |X_i - X_p|$ 에 상수 μ (통상적으로 1 근처의 값)를 곱하여 θ_i 를 결정하게 된다. ΔX_{ij} 를 구성하고 있는 모든 원소들의 크기는 d/\sqrt{m} 보다 같거나 작게 된다. 이러한 방식으로 구한 M개의 z_{ij} 에 대한 GP 트리의 출력 값 $y(z_{ij})$ 를 계산하고 그 평균 값 Y_i 를 산출하게 된다. 보다 정확한 평균값을 산정하기 위해서는 M의 크기를 증가시키면 되지만 그에 따라서 계산량도 함께 증가하게 된다. Overfitting된 GP 트리에서의 Y_i 값은 그렇지 않은 GP 트리에 대한 경우보다 상대적으로 클 가능성이 높다. 따라서 overfitting된 GP 트리들은 진화과정 중에 도태될 확률이 증가하게 된다. 그러나 overfitting된 GP 트리의 경우라도 Y_i 값은 작을 수 있다는 단점이 있기 때문에 개선된 성능의 트리를 얻기 위해서는 수식 (2)의 파라미터 (α, M, θ_i) 값을 변화시켜서 여러 번 실험을 수행하여야 한다. GP에서 overfitting 현상이 일어나는 주요한 요인중의 하나는 함수 집합에 정의된 함수의 고유 특성에 기인한다고 볼 수 있다. 가령 사칙연산자 (+, -, *, /) 중의 하나인 "/"는 인자가 두 개인 함수인데, 다음과 같은 수학적 a/b 는 GP에서 (/ a b)로 표현된다. 이때 b의 값이 0에 가까우면 "/" 함수가 돌려주는 값은

매우 큰 값을 갖게 된다. 이는 곧 GP 트리의 학습 성능이 우수하여도 학습 데이터 점 근처를 벗어나면 "/" 함수에 의해 GP 트리의 값이 급격히 변화할 수 있다는 것을 의미한다. 만일 함수 집합에 "sqrt"가 있다면, GP에서는 프로그램 수행 오류를 방지하기 위해서 "sqrt"라는 함수는 실제 $\sqrt{(\quad)}$ 가 된다. 이것은 곧 GP 트리가 어떤 지점에서 원하지 않는 불연속적인 거동을 보일 수 있다는 것이다. 전술한 문제들은 실제 GP를 사용할 때 자주 관찰되며, overfitting 현상을 야기하는 주요 원인으로 작용한다. Y_i 를 도입한 의도는 함수 집합에 사용된 함수들이 GP 트리에서 바람직하지 못한 거동을 나타내는가를 판정하기 위해서이다. 즉 어느 영역에서 어떤 GP 트리의 값이 급격히 변화하면 Y_i 에 반영되어 이는 곧 적합도에 큰 영향을 끼쳐 이 GP 트리는 다음 세대에서 도태될 가능성이 매우 높아진다.

3.3 GP트리를 이용한 함수 근사의 예

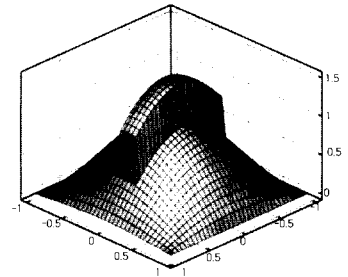
함수 근사에 사용될 GP의 적합성을 검증하기 위하여 SH(Skin-Head) 함수의 근사를 실험하였다. Fig.3(a)에서 보는 바와 같이 SH함수는 중앙부에 불연속 영역을 포함한 일반적인 수학적으로는 근사시키기가 어려운 함수이다.

여기서는 GP트리와 신경망을 이용한 결과를 비교하기 위하여 400개의 학습 데이터로 신경망을 학습하였다. 여기서 사용된 신경망 모델은 역전파 방법 대신에 빠른 학습 거동을 나타내는 conjugate-gradient 모델을 사용하였다(Adeli & Hung, 1995; 양영순 & 한상민, 1996). Fig.3(b)는 1200개의 테스트 데이터에 따른 신경망모델의 결과이다. 예상했던 것과 같이 중앙부의 불연속 구간의 근사에서 좋지 못한 결과를 보여주고 있

는 반면에 GP트리는 전 구간에서 매우 좋은 근사 결과를 보여주고 있다. 여기서 사용된 GP 트리에서는 x_1, x_2 가 모두 20개의 인덱스 가중치를 갖는 터미널 집합 $T = \{X_1^{w^{0-19}}, X_2^{w^{0-19}}\}$ 와 함수

$$F = \{ * , \% , + , - , iflte , \sin , \cos \}$$

로 구성된다. 여기서 iflte 함수는 총 네 개의 인자를 갖는데, 첫째 인자의 산정 값이 둘째 인자의 그것보다 크면 셋째 인자의 값을 돌려주고, 그렇지 않으면 넷째 인자의 값을 계산하여 돌려준다. Fig.4는 GP에 의해 추정된 SH함수의 근사해이다. 여기서 []에 나타난 실수값이 언덕 오르기 탐색에 의해 얻어진 인덱스 가중치를 의미하며, 함수나 터미널 노드 뒤 콜론(:)과 함께 나타난 정수는 인덱스이다. Fig.4에 나타나 있듯이 SH함수를 근사한 트리의 총 노드 수는 40개이며, 15개의 서로 다른 인덱스 가중치를 사용하고 있다.



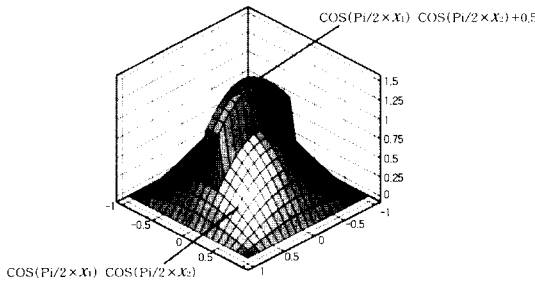
(c) GP tree solution

Fig.3 SH function Approximation

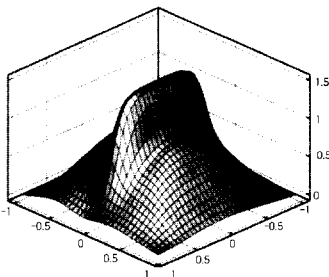
```
[0.999936]*(20 (iflte [0.9984161]*(20 [0.998416]*(20
(iflte (cos [1.57077]x:3) (cos (cos [1.5708151]y:17))
(iflte [0.9570269]x:5 [0.7689169]x:8[0.7689169]x:8
[1.9748527]y:12) (cos [1.6847957]y:18)) (cos
[1.5651845]y:17)) (cos [1.5710501]y:17)) (sin
[1.5560810]*(23 (cos [1.6993164]y:18) [1.2183294](%:36
[0.8231178]y:10 [1.2705649]y:15))) (cos
[1.5707755]x:3) [1.5188757]*(27 (cos [1.5708518]y:17)
[0.9889279] (%:34 (cos [1.2664022]x:9) (cos
[1.6920133]y:18)))) (cos [1.57085178]y:17))
```

Fig.4. GP tree for SH function

Table 1은 GP의 진화절차에 사용된 파라미터를 보여주고 있다. GP는 불연속성을 가진 함수의 근사에는 좋은 방법중의 하나이지만 최적의 트리를 생성하는데 있어서 Table 1에서 나타낸 파라미터들뿐 아니라 함수 집합, 터미널 집합들의 영향을 많이 받는다. 이러한 특성들 때문에 GP를 통해 요구되는 정확한 해를 얻기 위해서는 많은 실험과 경험, 그리고 문제영역에 대한 지식이 필요하다. 앞서서도 언급한 바와 같이 GP의 단점은 신경망에 비해 계산시간이 많이 요구된다는 것이다. 따라서 GP를 전 영역에서 사용하는 것이 아니라 제한된 필요한 좁은 영역, 예를 들어 불연속성을 가진 경계영역에서 사용하는 방법을 모색하였다. 본 논문에서는 신경망으로는 만족할 만한 결과를 얻을 수 없는



(a) original function



(b) neural network solution

경계영역을 따로 분리하여 GP트리를 사용하고, 나머지 영역은 신경망을 통하여 근사하는 방법을 사용하였다.

Table 1. Parameters used in evolving process

population size	700
maximum number of generation	30
initial depth of trees	3
allowable maximum depth of trees	7
selection method	tournament with 50 trees
reproduction probability	0.2
crossover probability	0.8
mutation probability	0.05
number of iterations for hill-climbing	30
α in equation(2)	1.0
M in equation(2)	10

4. 경사진 의사결정 트리(Oblique Decision Tree)에 의한 경계 데이터 생성

주어진 문제 영역에서 학습 데이터 집합을 $X\{X_i, Y_i\}$ 라 할 때 $X_i\{x_{i1}, \dots, x_{im}\}$ 는 독립변수들의 집합으로서 입력 값을 의미하며, $Y_i\{y_i\}$ 는 설계하고자 하는 출력을 의미한다. 여기서 각각의 y_i 에 대해 도메인 지식(domain knowledge)에 기초한 클래스 라벨(class label) c_i 가 부여된다. 따라서 출력 집합 Y 는 같은 클래스의 데이터를 포함하는 m 개의 subset(Y_j)으로 나눌 수 있으며, 여기에 맞추어 학습 데이터 집합 X 도 m 개

의 subset (X_j)로 세분화된다.

본 연구에서는 설계하고자 하는 파라미터의 효과적인 추정을 위해서 설계 영역이 입력 공간의 어떤 영역에 속하는지를 먼저 찾아낸 다음, 그 영역에 맞게 학습된 지역 에이전트인 서브네트워크를 수행하여 결과를 얻게 된다. 이러한 방법은 기존의 다중 인공 신경망 시스템에서 최종 출력값을 얻기 위해 얻어진 서브네트워크들의 결과를 합성해야 하는 필요성이 없기 때문에 매우 단순하면서도 우수한 결과를 보여준다. 여기서 학습 데이터의 영역이 어떤 영역에 속하는지를 알기 위해서 집합 $\{x_{i1}, \dots, x_{im}, c_i\}$ 을 대상으로 경사진 의사결정 트리(oblique decision tree)를 생성하였다. 이 트리의 생성을 위하여 OC1이라는 귀납적 분류 시스템(inductive classification system)을 사용하였다. Fig.5는 2차원 공간에서 경사진 의사결정 트리에 의해 나누어진 입력 데이터 공간의 예이다.

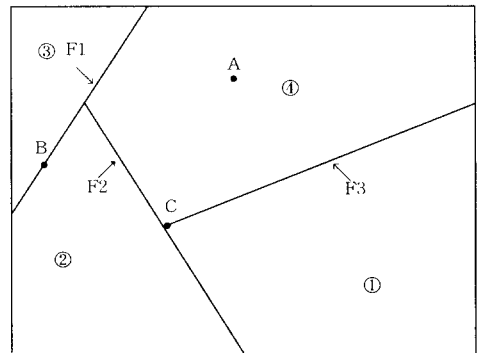


Fig.5 A Partitioned Input Space by Oblique Decision Tree in 2-D Space. F1, F2, and F3 are Hyperplane Equations of Decision Tree.

Fig.5에서 입력 점이 점A 근처에 주어졌을 때는 명백히 이 점은 영역4에 속함을 알 수 있으나 점B의 경우는 영역2에 속하는지 영역3에

속하는 지 판별하기 곤란하다. 더욱이 점C의 경우는 영역 1, 2, 또는 4 속할 수 있다. 이러한 문제를 해결하기 대안으로 다음과 같이 두 가지 방법을 고려해 볼 수 있다.

첫째는 Fig.6과 같이 신경망을 분류된 영역 이외에 hyper-plane을 따라 경계 영역만 담당하는 경계 에이전트를 도입하는 방법이다. 입력 데이터가 hyper-plane의 경계선 근처에 위치하면 지역 에이전트 대신에 이 경계 에이전트를 사용하게 된다. 이러한 경계 에이전트로서 GP가 사용될 수 있으나 경계 영역이 커지거나 복잡한 패턴을 갖게 될 때, 요구되는 정확도를 갖는 GP 트리를 얻기 위해서는 매우 많은 시간과 노력이 요구된다.

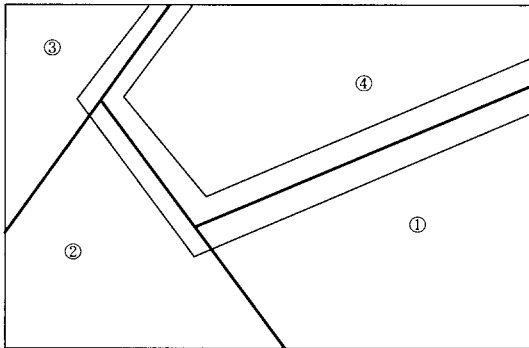


Fig.6 A Boundary Regions of Partitioned Input Space

둘째 방법은 tile 영역의 도입이다. 여기서 tile 영역은 분할된 입력 데이터 영역에서 모서리(edges)와 정점(vertexes)으로 구성된 hyper-plane 주위의 영역을 말하는 것으로 각각의 tile 영역에서 GP 트리가 학습되어 사용된다. Fig.7은 분할된 2차원 입력 공간에서 영역4 주위에 정의된 tile 영역의 예를 보여주고 있다. 영역4 주위에는 그림과 같이 3개의 tile 영역이 존재하고, 이들 각각에 대해서 학습된 GP 트리가 존

재한다. Fig.7 (b)에서 알 수 있듯이 tile 영역은 d_1, d_2, d_4, d_4' 등을 사용하여 정의해 주고 있다. 앞에서 설명되어 있듯이 의사결정 트리는 hyperplane 식의 값을 계산하여 데이터가 속한 영역 또는 클래스를 판별하는데, $d_i(i=1,2,4,4')$ 의 크기는 학습 데이터에 대한 hyperplane 식들의 값을 계산하고 이것을 토대로 결정할 수 있다. 만일 주어진 데이터에 대한 hyperplane 식의 절대 값이 d_i 보다 작으면 이 데이터는 tile 영역에 속하게 된다. d_i 의 크기를 산정하기 위해서 0에서 1 사이의 값을 갖는 두 종류의 gating ratio인 **R**과 **B**를 도입하였다. Fig.7(b)에 표시된 d_1, d_2, d_4 의 크기는 각각 $R \cdot N_i(i=1,2,4)$ 값을 기준으로 결정할 수 있다. 여기서 N_i 는 영역 1, 2, 4에 속한 학습 데이터의 수를 의미한다. 즉 $R \cdot N_i$ 의 반올림한 정수 값 $n_i(\text{round}(R \cdot N_i))$ 는 N_i 영역에 속한 학습 데이터 중 tile 영역에 포함될 데이터의 수가 된다. 이때 N_i 에 속한 모든 학습 데이터에 대하여 hyperplane 식의 절대값 계산하고, 이 값을 오름차순으로 소팅(sorting)한다. 소팅된 이 값들 중 n_i 번째 값이 바로 d_i 의 크기가 된다. 이와 마찬가지로 d_4' 의 크기도 $B \cdot N_4(N_4$ 는 영역4의 학습 데이터 수) 값을 이용하여 구할 수 있다. Gating ratio **B**는 입력 데이터가 Fig.7 (b)의 점 P와 같은 정점 주위에 주어져 두 개의 영역 이상에 속하게 될 때 사용됨으로 통상적으로 **R** 보다는 큰 값을 갖게 된다. 이러한 gating ratio의 결정은 주어진 응용 도메인과 분할된 입력 데이터 영역의 특성에 따라서 그 값을 적절히 선정해야 한다. 만일 입력 데이터 영역이 2차원 이상일 경우에는 앞서 언급한 방법과 유사하게 tile 영역을 나누지만 모서리나 정점으로 인한 복잡성 때문에 이를 위한 특별한 고려가 이루어져야 한다. 입력 데이터가 주어질 때, 항상 이

데이터가 세분된 영역의 경계에 근접하는지 검사하고, 충분히 경계에 접근되어 있다고 판단되면 일반 지역 에이전트 대신에 해당 tile 영역을 담당하는 경계 에이전트를 수행하여 그 결과를 얻게 된다. 본 논문에서는 지역 에이전트, 경계 에이전트, 그리고 퍼실리테이터를 채용한 연방 에이전트 시스템(federated agent system)을 구축하고 선박 중앙단면의 형상설계에 적용하여 본 방법의 유용성을 검증하였다.

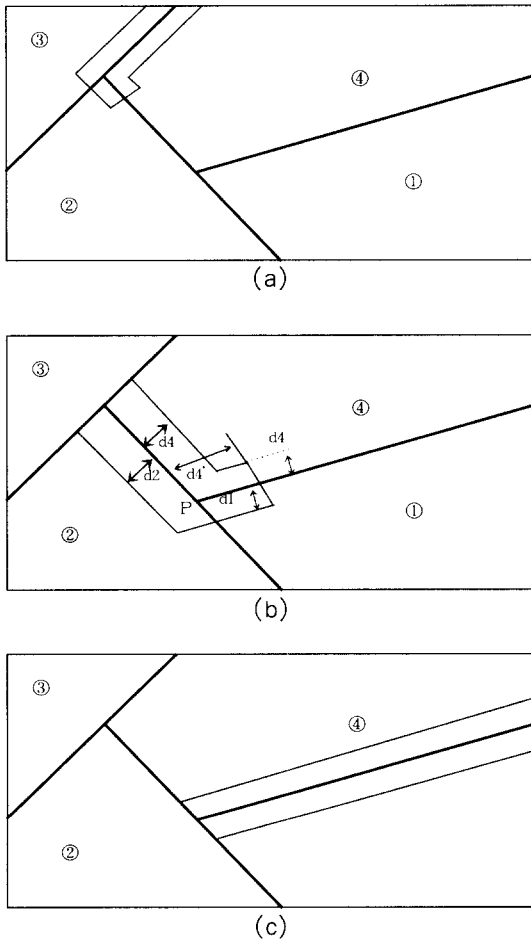
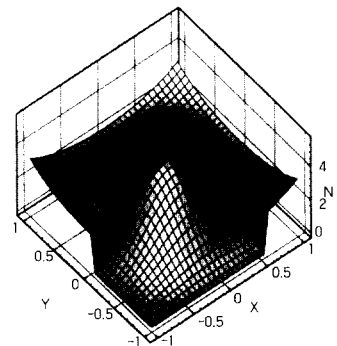


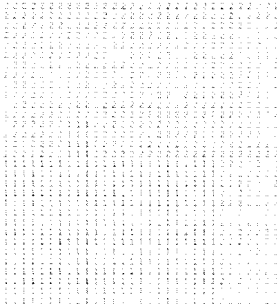
Fig.7 Three "Tile" Regions around a Region 4

5. 연방 에이전트 시스템의 검증 예

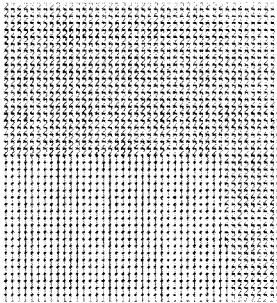
본 연구에서 개발된 연방 에이전트 시스템의 성능을 검증하기 위해 Fig.8(a)와 같이 불연속 패턴을 가진 서로 다른 함수의 합성함수에 대해 961개의 학습 데이터와 1849개의 테스트 데이터를 가지고 검증을 수행하였다. Fig.8(b)는 이 함수에 대한 두 개의 학습 데이터 세트를 나타내고 있으며, 예상과 같이 입력 영역을 하나의 신경망 모델(single neural network)로 학습시킨 경우는 Fig.8(d)와 같이 경계 부근에서 좋지 못한 결과를 보여주고 있다. Fig.8(c)는 테스트 데이터에 대해 분류된 영역을 나타내고 있으며, Fig.8(e)-(f)는 서로 다른 gating ratio에 대한 예측 결과를 보여주고 있다. 여기서 gating ratio가 0 이라는 것은 경계 영역을 두지 않은 경우로서 경계 부분에서 예측능력이 크게 떨어지게 된다. 그러나 적절한 gating ratio ($R=0.3$ $B=0.4$)에 대해 이 시스템은 전체적으로 우수한 결과를 보이고 있다. 이러한 gating ratio 값은 문제에 따라 달라질 수 있으며, 이들의 최적값을 찾게되면 매우 정확성 있는 예측 성능을 발휘할 수 있다.



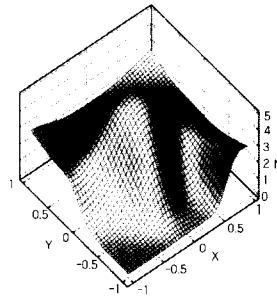
(a) original function



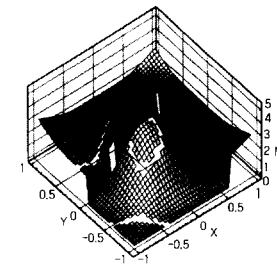
(b) 2 learning sets



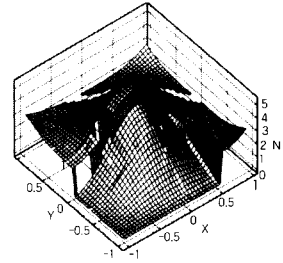
(c) test data classified by decision tree



(d) single neural network solution



(e) Federated agents solution with $R=0.0$ and $B=0.0$



(f) Federated agents solution with $R=0.3$ and $B=0.4$

Figure.8 Solutions of the federated agents

6. 공학설계에의 적용 : 선박 중앙단면 형상 설계

전통적으로 선박의 중앙 평행부는 선박 전체 길이의 70%이상을 차지하는 부분으로 이 단면의 형상은 선박의 구조나 강도, 선가(船價) 등에 결정적인 영향을 미치게 된다. 따라서 중앙 단면의 형상설계는 설계 단계에 있어서 매우 중요한 단계라고 할 수 있다. 이러한 중앙단면의 형상이 결정되면 설계 규정집에 따라 보강재의 배치나 철판의 두께 등을 결정하는 scantling 작업을 수행하게 된다. 이는 선박이 설계될 때 강도나 안전 측면에서 선급 회사로부터 검사를 받게 되는데 설계시 이러한 선급 규정에 따라 설계를 수행한다. 선급 규정은 고려해야 할 항목이 매우 많고 이러한 항목들이 서로 복잡하게 연결되어 있어 지식 기반 시스템에 의한 접근이 최근 들어 많이 이루어지고 있다(Lee, et al. 1996; Yeun & Yang, 1997). 그러나 중앙단면 형상 결정에 대한 연구(양영순 & 연윤석, 1996; Yeun, et al. 1997)는 극히 찾아보기 힘들다.

본 논문에서는 개발된 연방 에이전트 시스템의 공학적 범용성을 검증하기 위해 산적화물선(bulk carrier)의 중앙단면 형상 결정에 적용하였다. 그러나 불행하게도 각 조선소별로 과거의 실적선은 각 사 고유의 설계 노하우를 포함하고 있기 때문에 대외비로 취급되고 있어서 충분한 학습 데이터를 마련하는 것이 사실상 거의 불가능한 일이다. 따라서 50개의 입수된 실적선 데이터 이외에 나머지 부족한 학습 데이터는 선박 개념설계 시스템(BASCON-III)을 20 번 수행하고 그 결과들을 이용하여 방법을 모색하였다(Lee, et al 1994). 총 70개의 데이터 중 45개는 학습 데이터로 사용하고 나머지는 테스트 데이터로 사용하였다. 문제의 복잡성을 줄이기 위해서 입력 파라미터(Table 2)를 8개로 제한하였으며, 출력은 하나로 하여 필요한 설계변수 각각에 대해서 학습을 수행하였다. 즉 $R^8 \rightarrow R^1$ 매핑을 통한 학습을 시도하였다. 그러나 학습 데이터의 수와 비교하여 입력 데이터의 차원이 아직도 큰데, 이러한 경우 신경망의 견실한 예측 성능을 기대하기 어렵다. 이 문제점의 해결 방안으로 지역변수(local variable) 선정 기법이 있다. 이것은 전체 입력 공간의 차원보다 적고 최적의 신경망 성능을 줄 수 있는 몇 개의 입력 변수를 선정하는 것인데, 잘 알려진 분류 시스템인 C4.5도 이 기법을 채용하고 있다(Quinlan, 1993). 본 연구에서는 많은 실험을 통하여 지역 에이전트와 경계 에이전트의 성능 극대화하는 최적의 입력 변수들을 선정하였다. 생성된 의사결정 트리의 크기를 줄이는 과정(pruning)을 수행하기 위하여 20% 학습 데이터가 임의로 선정되어 사용되었다. Fig.9는 전형적인 산적화물선의 중앙단면을 나타내는 형상이고, 이들 각 설계 항목에 대해서 연방 에이전트 시스템을 이용하여 얻은 결과를 단독 신경망에 의한 결과와 비교하여 Table 2에 나타내었다.

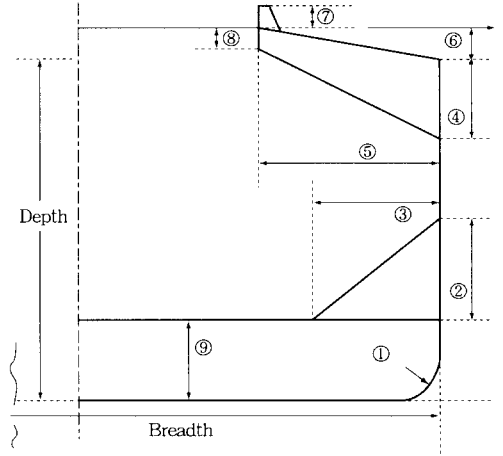


Fig.9 A Midship Section of Bulk Carrier Ship

일반적으로 연방 에이전트 시스템의 결과는 신경망의 은닉노드 개수, 가중치의 초기값, GP 트리의 세대수와 정확도 등에 많은 영향을 받는다. 그러나 이 실험에서 대부분의 경우 연방 에이전트 시스템의 결과가 단독 신경망의 결과보다 우수함을 알 수 있었다. 그러나 Table 2에 나타난 결과가 실제의 설계 지식과는 다소 차이가 있음을 알 수 있다. 예를 들어, 선박의 재화중량(dead weight : DWT)은 대부분의 설계변수에 큰 영향을 미치지만 여기서는 몇 개의 설계변수만이 영향을 받는 것으로 나타났다. 이러한 경향은 학습 데이터가 몇 개의 조선소 그리고 설계 프로그램 등 다양한 출처로부터 나온 것에서 기인 할 수 있다. 즉 조선소마다 고유의 노하우가 있어, 실적선 자료가 조선소마다 차이 있음을 확인할 수 있다. 다양한 출처에서부터 온 학습 데이터로 학습된 신경망은 이러한 사실을 반영할 수 없다. 학습 데이터 중 몇몇은 재화중량은 거의 동일하나 용적(volume)이 다른 경우가 있어, 이것 또한 시스템의 거동에 영향을 미칠 가능성이 있다. 그럼에도 불구하고 본 논문에서

제안된 시스템이 은닉층의 개수, weight의 초기 값 등의 인자에 따라 결과에 영향을 많이 받는 단독 신경망에 비해 예측 오류가 작아 일관성 있는 좋은 결과를 보여주고 있음을 확인할 수 있다.

Table 2. Solutions of single and the federated agents with $R=0.3$, and $B=0.5$. Note that the same gating ratios are used for all regions of divided input space for convenience.

Symbol										Single neural network		Federated agents		
	D W T	V O L	Vs	L	B	D	Cb	T	Learning Error	Test Error	No. of Regions	Learning Error	Test Error	
①BRDS		⊙			⊙				2.07E-2	2.29E-2	2	1.25E-2	1.45E-2	
②HTKH		⊙		⊙	⊙				7.33E-2	1.13E-1	3	2.95E-2	9.96E-2	
③HTKB		⊙		⊙	⊙				5.23E-2	7.73E-2	3	2.88E-2	5.50E-2	
④UWTH	⊙				⊙				1.37E-1	1.71E-1	3	6.79E-2	1.23E-1	
⑤UWTB		⊙		⊙					4.39E-1	2.68E-1	3	1.18E-1	1.75E-1	
⑥CMBH		⊙	⊙	⊙	⊙				1.85E-3	8.82E-3	2	3.78E-3	4.12E-3	
⑦HCMH	⊙								6.69E-2	6.50E-2	2	2.73E-2	4.55E-2	
⑧HUGH		⊙	⊙		⊙	⊙			2.79E-3	4.26E-3	2	5.98E-4	3.33E-3	
⑨DBH	⊙	⊙		⊙					5.29E-3	5.80E-3	4	8.64E-4	3.88E-3	

where

Learning and Test Error: Mean Square Error
 DWT : Dead Weight(ton)
 VOL : Volume(ton)
 Vs : Service Speed(knot)
 L : Length between perpendiculars
 T : Draft(m)
 B : Breadth(m)
 D : Depth of Ship(m)
 Cb : Block Coefficient

BRDS : Bilge Radius(m)
 UWTB : Upper Wing Tank Breadth(m)
 UWTH : Upper Wing Tank Height(m)
 HTKB : Hopper Tank Breadth(m)
 HTKH : Hopper Tank Height(m)
 HCMH : Hatch Coaming Height(m)
 CMBH : Camber Height(m)
 HUGH : Hatch Under Girder Height(m)
 DBH : Double Bottom Height(m)

7. 결 론

본 논문에서는 의사결정 트리와 연계된 연방 에이전트 시스템에 대해 언급하였다. 즉 전형적인 다중 인공 신경망 시스템인 지역 에이전트와 불연속 영역이나 몇 개의 패턴이 동시에 존재하는 영역에서 우수한 성능을 보일 수 있는 경계 에이전트인 GP트리를 도입하였다. 또한 의사결

정 트리에 의해 주어진 정보를 바탕으로 적절한 지역 에이전트나 경계 에이전트를 선택해 주는 기능을 퍼실리테이터를 두어 담당하게 하였다. 여기서 제안된 연방 에이전트 시스템의 장점은 대부분의 분할된 입력 영역에서는 신경망에 의한 지역 에이전트를 이용하고 제한된 경계 영역에서 예측 성능이 뛰어난 GP트리를 이용함으로써 전체적인 시스템 성능이 향상되었다는 것이다. 신경망에 비해 예측성능은 우수하지만 최적

의 세대 수, 함수 집합, 터미널 집합을 찾는데 많은 시간과 노력을 요하는 GP의 사용을 좁은 경계 영역으로 제한하였다. 그러나 이 시스템의 성능은 gating ratio에 상당히 의존적이며 이에 따른 적절한 gating ratio를 어떻게 결정하느냐 하는 것이 매우 중요하다. 연방 에이전트 시스템을 구축하는 기본적인 전략은 대부분의 분리된 입력 공간에 신경망 모델을 사용하여 전체적인 시스템의 성능 향상을 꾀하고, 동시에 세분화된 영역의 경계 부분은 GP트리를 사용하는 것이다. 따라서 GP트리의 성능이 중요한 변수인데, 최소한 단독 신경망의 성능과 비교하여 그 성능이 떨어지지 않으면 만족한 결과를 얻을 수 있다. 그러나 GP트리의 성능이 단독 신경망의 성능보다 떨어진다고 해도 그 효과는 다른 서브네트웍에 파급되지 않기 때문에 적절한 gating ratio를 사용하여 구축된 연방 에이전트 시스템의 전체적인 성능을 크게 저하시키지는 않는다.

8. 추후 연구과제 및 토의

본 연구들 통하여 앞으로의 연구과제나 나타난 문제점을 요약하면 다음과 같다.

연합 신경망의 성능은 gating ratio에 큰 영향을 받기 때문에 gating ratio를 체계적으로 결정할 수 있는 방안이 모색되어야 한다. 또한 본 연구의 접근 방법이 유효한 경우는 분리된 입력 공간이 의사결정 트리 형태로 잘 표현될 수 있어야 하며, 의사결정 트리의 예측 성능이 우수해야 된다. 만일 분리된 공간의 수가 많으면 tile 네트웍의 증가에 따른 많은 GP 트리가 필요하다는 단점이 있다. GP트리의 생성시 요구되는

대부분의 계산량은 언덕 오르기(hill-climbing)를 사용한 가중치 산정에 소요된다. 따라서 계산량을 줄이기 위해서는 보다 빠른 가중치 산정 방법의 도입이 요구된다. 현재는 가중치 산정을 선형 연상기억(linear associative memory)을 사용하려고 시도하고 있는데, 언덕 오르기(hill-climbing) 탐색과 비교하여 매우 우수한 계산 효율을 보이고 있다.

중앙단면 데이터는 여러 출처를 통하여 얻은 것인데, 이는 곧 데이터에 노이즈(noise)가 존재하는 것과 같은 효과를 낼 수 있다. 더욱이 학습 데이터의 양이 적고 노이즈가 있는 경우 overfitting이 문제로 대두되기 때문에, 본 연구에서 많은 시행 착오와 실험을 반복하여 어느 정도 만족한 결과를 얻고자 하였다. 따라서 신경망과 GP트리의 예측성능을 개선하기 위해서 보다 체계적으로 유용한 방법론을 적용할 필요가 있는데, 이와 관련하여 앞으로의 연구과제를 요약하면 다음과 같다.

첫째, 인공신경망 영역에서 일반화 성능을 높이기 위해서 실용적으로 사용될 수 있는 방법이 stopped training(Sarle, 1995)이 있는데, 이러한 방법들은 어느 정도 충분한 학습데이터를 확보해야 하기 때문에 중앙단면인 경우는 적용하기가 어렵다. 그러나 신경망의 일반화 성능을 향상시키기 위해서 cross-validation(Plutowsk at el, 1994)이나 bootstrapping(Efron & Tibshirani, 1993)등의 방법은 적용이 가능하리라고 판단된다.

둘째, GP는 최근에 연구가 활성화되고 있는 시점이기 때문에 예측 성능 향상에 관련된 연구를 찾아보기 어렵다. 그러나 일반적으로 크기가 작은 GP트리가 그렇지 않은 트리보다 일반화 성능이 우수하다고 할 수 있기 때문에(Zhang & Muehlenbein, 1995) 적합도에 트리의 크기를

계측하는 함수를 포함하는 것이 유리하다고 판단된다. 신경망에 쓰이는 cross-validation 이나 bootstrapping이 GP에 유용하게 적용될 수 있는지 검토해 보는 것도 앞으로 수행할 흥미 있는 연구과제라 할 수 있다. 또한 본 논문에서 overfitting을 방지하기 위해 제시된 방법보다 신뢰성 있는 대안이 필요한데, 현재는 선형보간법이나 nearest neighbor를 기반으로 한 방법을 시도하고 있다.

인공 신경망이나 GP는 소위 전역적 모델(global model)이라고 할 수 있는 반면 nearest neighbor 기반의 방법은 지역적 모델(local model) 혹은 lazy learning(Atkeson et al, 1996)의 일종으로 서로 상반되는 관점을 보이고 있다고 할 수 있다. 그러나 지역적 모델도 함수근사에 유용한 대안을 제시할 수 있다. 즉 본 논문에 사용된 중앙단면 데이터를 사용하여 단독 신경망을 학습하고 그 결과를 nearest neighbor 기반의 방법과 비교했는데, 둘 다 비슷한 예측 성능을 보였다. 여기서 nearest neighbor 기반의 방법이란 새로운 입력 데이터가 주어졌을 때 학습 집합으로부터 유사한 몇 개의 데이터를 추출하고, 이들로부터 가우스 커널기반의 가중평균을 구해 사용하는 방식을 의미한다. 이러한 지역적 모델은 전역적 모델을 사용하여 만족한 결과를 얻지 못할 때 매우 유용한 대안이 될 수 있고 전역적 모델에서와 같은 학습과정이 필요 없다는 장점이 있다. 지역적 모델(local model)은 데이터가 많을 수록 보다 우수한 성능을 발휘할 수 있음을 쉽게 짐작할 수 있다. 만일 노이즈를 극복할 수 있는 방법과 함께 지역적 모델을 중앙단면 설계에서 적용한다면 우수한 성능을 발휘할 수 있을 것으로 판단된다.

본 논문의 방법론을 적용하기 위해서는 입력 공간을 분리할 수 있어야 하는데, 중앙단면의

경우는 주로 경험과 직관적인 방법에 의존하였다. 즉 중앙단면을 구성하는 부재에 영향을 미칠 수 있는 변수들을 과거 설계경험을 통하여 쉽게 파악될 수 있는데, 통상 가장 중요한 변수는 재화중량이다. 어떤 부재의 수치를 재화중량과 영향력이 비교적 크다고 판단되는 변수를 선정하여 가시화하면 2-3 정도의 집단이 이루어짐을 쉽게 파악할 수 있다. 만일 또 다른 중요한 변수가 있다면 재화중량과 이 변수에 대한 데이터를 가시화 하여 그 패턴을 살펴볼 수 있다. 이때에도 앞의 경우와 유사한 형태의 집단을 이루는데, 그것은 재화중량 이외에 선정된 변수들이 실제로 독립변수가 아니며, 재화 중량과 밀접한 관계를 맺고 있기 때문이다. 일반적으로 개념설계 단계에서 선박의 길이 높이 폭 등은 재화중량을 사용하여 추정된다는 것을 상기하면 된다. 재화 중량만을 사용하여 중앙단면 부재를 추정하지 않는 이유는 부재가 높이나 높이에 더 많은 영향을 받고 있을 가능성이 많기 때문이다. 만일 중앙단면과 달리 입력공간을 경험이나 직관에 의존하여 분리할 수 없다면 군집화(clustering)에 사용되는 다양한 기법들을 이용하여 공간을 나누고 이것을 의사결정 트리로 표현하는 방법을 고려해 볼 수 있다.

후 기

본 논문이 완성되기까지 많은 조언을 해 주신 서울대학교 양영순 교수님과 시스템 구축에 도움을 준 서울대학교 대학원의 한상민 군에게 깊은 감사를 드린다. 아울러 본 논문을 매우 세심히 검토해 주신 심사 위원들께 깊은 사의를 표한다.

참고문헌

- 양영순, 연운석, "유전적 프로그래밍을 이용한 중앙단면 설계 지식의 추출", *대한조선학회 '96 추계 학술대회 발표집*, (1996), 86-89.
- 양영순, 한상민, "지능형 선상 가열 시스템 개발", *서울대학교 조선해양공학과 기술보고서*, 1996.
- 이경호, 연운석, 김남준, "초기설계 단계에서 Decision Tree를 이용한 선박의 설계정보 추출", *한국전문가시스템학회 '96 추계 학술대회 발표집* (1996), 189-195.
- 이경호, 연운석, "다중 인공신경망의 Federated Architecture와 그 응용-선박 중앙단면 형상 설계를 중심으로", *한국 CAD/CAM학회 논문집*, 제2권 제2호(1997), 77-84.
- Adeli, H. & S. L. Hung, "*Machine Learning*", John Wiley and Sons, New York, 1995.
- Atkeson, C. G., A. W. Moore and S. Schaal, "Locally Weighted Learning", *AI Review*, Vol.11 (1996).
- Efron, B. and R. J. Tibshirani, "*An Introduction to the Bootstrap*", Chapman & Hall, London, 1993.
- Happel, B. and J. M. Murre, "Design and Evolution of Modular Neural Network Architecture", *Neural Networks*, Vol. 7, No.1 (1994), 985-1004.
- Jacobs, R. A. and M. I. Jordan, "Adaptive Mixtures of Local Experts", *Neural Computation*, Vol.3, No.1 (1991).
- Jacobs, R. A. and M. I. Jordan, "Learning piecewise control strategies in a modular neural network architecture", *IEEE Transactions on System, Man, and Cybernetics*, Vol.23, No.2 (1993), 337-345.
- Koza, J. R., *Genetic Programming : On the Programming of Computers by Means of Natural Selection*, The MIT Press, 1992.
- Koza, J. R., *Genetic Programming II : Automatic Discovery of Reusable Programs*, The MIT Press, 1994.
- Lee, D., K. H. Lee and S. H. Han, "Development of the Adaptable User-oriented Conceptual Ship Design System", *Proc. of 9th International Conference on Computer Applications in Shipbuilding* (1994), 11.61-11.71.
- Lee, K. H. and D. Lee, "Object-Oriented Approach to a Knowledge-Based Structural Design System", *Expert Systems with Applications*, Vol.10, No.2 (1996), 223-231.
- Murthy, S. K., S. Kasif and S. Salzberg, "A System for Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research*, Vol.2 (1994), 1-32.
- Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
- Plutowski, M., S. Sakata and H. White, "Cross-validation estimates IMSE", *Advances in Neural Information Processing Systems 6*, San Mateo, CA: Morgan Kaufman, (1994) 391-398.
- Sarle, W. S., "Stopped Training and Other Remedies for Overfitting," *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics*, (1995), 5352-5360.
- Sharman, K. C., A. I. Esparcia-Alcazar and Y. Li, "Evolving Signal Processing Algorithms by Genetic Programming", *Proc. First IEEE Int. Conf. on GA in Eng. Syst.: Innovations and Appl., Sheffield, Sept.*, (1995), 473-480. Also available

at <http://www.mech.gla.ac.uk/~yunli/csc95012.htm>.

- Smieja, F. J., "The Pandemonium System of Reflective Agents", *IEEE Transactions on Neural Networks*, Vol.7, No.1 (1996).
- Waibel, A., "Modular Construction of Time-Delay Neural Networks for Speech Recognition", *Neural Computation*, Vol.1 (1989), 39-46.
- Watson, A.H. and I. C. Parmee, "Identification of Fluid Systems using Genetic Programming", *Proceedings of Fourth European Congress on Intelligent Technique and Soft Computing*, (1996). Also available at <http://techweb.see.plym.ac.uk/soc/research/edc/publicat/papers/wec96.zip>.
- Yeun, Y. S., K. H. Lee, S. M. Han and Y. S. Yang, "Multiple Neural Networks Coupled with Oblique Decision Trees: a Case Study on Configuration Design of Midship Structure", *Proc. of Intelligent Information Systems* 97, Dec., (1997), 161-167.
- Yeun, Y. S. and Y. S. Yang, "Design Knowledge Representation and Control for the Ship Structural Design", *Knowledge-Based Systems*, Vol.10, No.2 (1997), 121-132.
- Zhang, X., J. P. Mesirov and D. I. Waltz, "Hybrid System for Protein Secondary Structure Prediction", *Journal of Molecular Biology*, (1992), 1049-1063.
- Zhang, B. T. and H. Muehlenbein, "Balancing Accuracy and Parsimony in Genetic Program-ming", *Evolutionary Computation*, Vol.3, No. 1 (1995), 17-38.