

ANN을 이용한 절삭성능의 예측과 ACO를 이용한 훈련

오수철*[#]

*부경대학교 시스템경영공학부

Prediction of Machining Performance using ANN and Training using ACO

Soo-Cheol Oh*[#]

*Department of Systems Management & Engineering, Pukyong National University, Busan, Korea
(Received 18 August 2017; received in revised form 13 September 2017; accepted 10 October 2017)

ABSTRACT

Generally, in machining operations, the required machining performance can be obtained by properly combining several machining parameters properly. In this research, we construct a simulation model, which that predicts the relationship between the input variables and output variables in the turning operation. Input variables necessary for the turning operation include cutting speed, feed, and depth of cut. Surface roughness and electrical current consumption are used as the output variables. To construct the simulation model, an Artificial Neural Network (ANN) is employed. With the ANN, training is necessary to find appropriate weights, and the Ant Colony Optimization (ACO) technique is used as a training tool. Especially in particular, for the continuous domain, ACOR is adopted and the related algorithm is developed. Finally, the effects of the algorithm on the results are identified and analyzed.

Key words : ACO(개미군 최적화), Neural Network(신경망), Surface Roughness(표면조도), Electric Current Consumption(전류소모량)

1. 서 론

공작기계인 선반에서 수행하는 선삭작업은 기계 가공 산업에서 널리 사용되며 중요한 절삭공정 중의 하나이다. 선삭공정의 입력 매개변수에는 절삭 속도, 이송, 절삭깊이, 냉각방법 등 여러 가지가 있다. 또한 출력 변수에는 표면거칠기, 공구수명, 진동, 전류 소모량 등이 있으며 이것들은 입력 매개변수에 의해 영향을 받는다.

선삭공정에서 입력변수와 출력 변수간의 관계를 모델링하는 연구는 여러 가지가 있는데 Ezilarasan

et al.^[4]은 절삭력, 플랭크마모, 표면거칠기를 예측하기 위하여 RSM(Response Surface Methodology) 방법을 이용한 경험적 모델을 제시하였다. Ahilan *et al.*^[1]은 신경망(neural network) 모델을 개발하였는데 Back Propagation(BP), 유전 알고리즘, Particle Swarm Optimization (PSO) 등의 방법을 이용하여 training을 하는 하이브리드 모델을 제시하였다. Punuhsingon & Oh^[2]에서는 BP와 PSO의 두 가지 방법을 이용하여 신경망을 training하는 모델을 제시하였다.

본 연구에서는 선삭가공에서 출력변수 값을 예측하기 위하여 Artificial Neural Network (ANN)을 이용한 시뮬레이션 모델을 제안하는데 모델 내에서 입력변수로서는 절삭속도, 이송, 절삭깊이가 사

Corresponding Author : scoh@pknu.ac.kr

Tel: +82-51-629-6483, Fax: +82-51-629-6478

용되고 출력변수로서는 표면거칠기와 전류소모량이 사용된다. ANN의 응용에서는 네트워크내의 적절한 가중치를 찾고 조정하는 것이 핵심인데 이 가중치를 조정하기 위하여 Ant Colony Optimization (ACO)의 방법을 이용한다. 원래 ACO는 이산 최적화 문제를 해결하기 위하여 고안되었는데^[3], 변수들이 연속적인 영역에 속하는 경우에는 직접적으로 응용하기는 곤란하다. Socha and Dorigo^[7], Socha and Blum^[8]에서는 변수들이 연속적인 경우에 적용할 수 있는 ACO 기법을 제안하고 이를 ACO_R 라 명명하였다. Pandian^[6]은 ACO_R 기법을 이용하여 신경망을 training하는데 기계가공이 아닌 식량생산 분야의 모델에 적용하고 있다. 본 연구에서는 선삭가공 공정을 대상으로 모델을 구축하는데 ANN 내의 수치 가중치를 최적화하기 위하여 ACO_R 기법을 응용한 알고리즘을 제시하고 ACO_R 내에서 변수들이 출력에 어떠한 영향을 끼치는지 검토하고 분석하였다.

2. 본 론

2.1 Artificial Neural Network

Artificial Neural Network(ANN)은 생체모방 신경망의 정보처리 시스템으로 단순하지만 상호 연결된 많은 처리단위들로 구성되고 인간 또는 유기체의 뇌의 신경계를 시뮬레이션 하는데 사용된다^[9]. ANN은 하나의 레이어(layer) 또는 다중레이어로 구성된다. 각 레이어는 다수의 뉴런(neuron) 들을 포함하며 하나의 레이어와 다른 레이어의 뉴런들 간의 연결에는 가중치가 연계되어 있다. ANN은 일련의 입력을 적용하면 원하는 일련의 출력을 산출할 수 있도록 학습을 시킬수 있는데 이것을 training 이라 한다. training은 입력 벡터를 순차적으로 적용하면 정해진 절차에 따라서 신경망의 가중치를 조절하는 식으로 수행된다. 일반적인 응용에서는 주로 다중레이어 신경망을 사용하는데 이것은 하나의 입력 레이어, 하나 이상의 hidden 레이어, 하나의 출력 레이어로 이루어져 있다. Back propagation(BP)은 다중레이어 신경망의 가중치를 반복적으로 해결하는 일반적인 방

법이다. BP 알고리즘은 목적함수를 최소화하도록 설계된 하나의 최적화 기법인데 공통적으로 사용되는 목적함수는 자승에러(squared error)이며 아래와 같다.

$$\varepsilon^2 = [T - \Phi]^2$$

여기서 T는 목표하는 출력값, Φ 는 뉴런의 출력, ε 는 에러 항목을 나타낸다. 본 연구에서는 선삭가공 모델을 구축하는데 ANN을 사용하며 ANN에 절삭변수를 입력하면 절삭성능을 예측할 수 있다. 실제 실험에 의한 관찰값과 ANN을 이용한 예측값의 차이를 최소화하기 위해서는 신경망을 구성하는 적절한 가중치 값을 찾아야 한다.

2.2 Ant Colony Optimization

Ant Colony Optimization(ACO)는 Dorigo et al.^[3]에 의해 1990년대 초에 이산(discrete) 최적화 문제에 응용하기 위해 도입된 최적화 기법이다. ACO의 기원은 사회적 곤충, 새의 무리들이 갖는 어떤 속성을 최적화 유형의 업무에 적용하는 것을 연구하는 집단지능(swarm intelligence)이라고 부르는 분야에 소속되어 있다. 실제로 개미집단의 최단 경로 발견 능력은 최적화 문제를 해결하기 위하여 인공적 개미집단(artificial ant colonies)이라는 분야에서 깊이 탐구되고 있다.

2.3 Continuous ACO

ACO 알고리즘은 초기에는 조합 최적화 문제(Combinatorial Optimization Problems: COPs)들을 해결하는데 도입되었다. COPs의 예는 스케줄링, 차량경로 선정, 시간표 작성 등이 있다. 한편, COPs의 유형이 아닌 중요한 문제 유형들이 있는데 연속변수들의 값을 선택하도록 요구하는 최적화 문제 집단이다. 이러한 문제들은 연속적인 허용값들의 영역을 유한 집합으로 변환하고 나서 조합 최적화 알고리즘을 적용하여 해결할 수 있는데 이것은 항상 편리한 것은 아니다. 특히 초기의 가능 영역이 방대하거나 해의 해상도가 매우 클 때는 그러하다. 이런 경우에는 연속변수들을 근원적으로 취급할 수 있는 알고리즘이 일반적으로 훨씬 잘 작동한다.

초창기의 ACO 기법을 연속 최적화 문제에 적용하기 위한 연구들이 있었으나, 대부분의 방법들은 원래의 ACO 골격을 따르지 않는다. 그 이후 ACO를 연속적인 영역에 적용할 수 있도록 확장한 ACO_R 이라는 방법이 Socha and Dorigo^[7], Socha and Blum^[8]에 의해 제안되었고 최근까지는 ACO의 기본 성격에 가장 근접된 형태의 변형으로 간주되고 있다. ACO_R의 기본적인 아이디어는 ACO가 이산적인 확률분포를 사용하는 것과 달리 연속적인 확률밀도 함수를 사용하는 것이다.

2.3.1 ACO_R 내에서 페로몬의 표현

Socha and Dorigo^[7]는 ACO를 연속적인 영역에 적용할 수 있도록 확장한 ACO_R을 제안하였다. ACO_R에서는 유한개의 해(solution)들을 아카이브(archive) T 라는 명시적 메모리 내에 저장하고 유지한다. 즉, n 차원 문제의 각각의 해 s_j 에 대해서 n 개 변수들의 값과 목적함수 값 $f(s_j)$ 을 저장한다. j 번째 해의 i 번째 변수는 s_j^i 로 표기한다. 먼저, 해의 구성성분 또는 변수들을 사용하여 동적으로 확률밀도함수 (Probability Density Function: PDF)를 생성한다. 저장된 해의 집합에 의거하여 PDF를 생성하는 방법은 아래와 같다. 가우스 함수들에 근거한 PDF를 사용하는데 약간 개선된 형태인 가우스 커널 PDF를 사용한다. 가우스 커널 (gaussian kernel)은 여러 개의 일차원 가우스 함수 $g_j^i(x)$ 의 가중합으로 정의하며 이것을 $G^i(x)$ 로 명명한다.

$$G^i(x) = \sum_{j=1}^k \omega_j g_j^i(x) = \sum_{j=1}^k \omega_j \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x - \mu_j^i)^2}{2(\sigma_j^i)^2}} \quad (1)$$

여기서 ω 는 개별 가우스 함수들과 관련된 가중치 벡터, μ^i 는 평균 벡터, σ^i 는 표준편차 벡터이다. 이와 같이 가우스커널 PDF G^i 는 3개의 벡터 ω , μ^i , σ^i 의 매개변수들로 이루어지며 아카이브 내의 해들은 이 매개변수들의 값을 계산하는데 사용된다. 아카이브에 저장된 해의 개수는 k 로 설정하고, 이 매개변수는 PDF의 복잡도를 결정한다. k 개의 개별 가우스 함수들이 가우스커널 PDF를 구성한다,

각 차원 $i = 1, 2, \dots, n$ 에 대해서 각각 다른 가우스 커널 PDF G^i 가 정의된다. 각 G^i 에 있어서 k 개의 모든 해의 i 번째 변수값들은 벡터 μ^i 의 요소들이 된다.

$$\mu^i = \{\mu_1^i, \dots, \mu_k^i\} = \{s_1^i, \dots, s_k^i\} \quad (2)$$

가중치 벡터 ω 는 아래의 방식으로 만들어진다. 아카이브 T 에 추가되는 각 해는 목적함수 값을 계산하여 평가되고 서열화된다. 그리고 아카이브 내의 해들은 서열순으로 저장된다. 각 해는 해의 품질에 비례하는 관련된 가중치 ω 를 갖는다. 해 s_j 의 가중치 ω_j 는 아래 공식에 따라 계산된다.

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(j-1)^2}{2q^2k^2}} \quad (3)$$

여기서 가중치는 가우스 함수의 값으로 정의된다. 함수의 인수는 j , 평균은 1.0, 표준편차는 $q \cdot k$ 이다. q 는 알고리즘의 매개변수이다. q 가 작으면 서열이 높은 해들이 더 강력한 우선권을 갖는다. 각각의 가우스 커널 PDF G^i 의 최종 형태를 찾기 위해서 표준편차 벡터 σ^i 도 정의해야 한다. 단계 i 에서 표준편차 σ_j^i 를 결정하기 위해서 아래의 식이 사용된다.

$$\sigma_j^i = \xi \sum_{e=1}^k \frac{|s_e^i - s_j^i|}{k-1} \quad (4)$$

이때 매개변수 $\xi (>0)$ 는 ACO에서의 페로몬의 증발율과 유사한 효과를 갖는다. ξ 의 값이 클수록 알고리즘의 수렴속도는 떨어진다. ACO내에서는 페로몬의 증발율은 장기기억에 영향을 주지만 ACO_R에서는 ξ 는 장기 기억이 사용되는 방법에 영향을 미친다.

2.3.2 ACO_R의 골격

1) 아카이브 T 의 초기화

초기해의 생성은 검색 영역 (a,b)에 걸쳐 균일분포를 생성하는 방식으로 수행한다. 이것은 균일하

게 분포된 평균을 갖는 k 개의 정규분포를 수행함으로써 생성한다. 즉 아카이브 T 의 초기화는 균일 랜덤샘플링에 의해 k 개의 해를 생성하는 것이다.

2) 해의 구축

의사결정변수 X_i ($i=1,2,\dots,n$)가 주어지면 n 개의 단계를 수행하여 하나의 해를 구축한다. 먼저 단계 i 에서 변수 X_i 값을 하나 선택한다. 앞에서 서술했듯이 가우스커널 PDF는 여러 개의 정규 가우스 함수들로 구성된다. 함수들의 개수는 아카이브의 크기 k 와 동일하다. 단계 i 에서는 오로지 i 번째 차원에 대한 정보만 사용되며, 각 단계마다 최종 가우스커널 PDF G^i 는 구분되어 진다.

식 (1)에 따라 PDF G^i 를 정의하기 위해 벡터 ω , μ^i , σ^i 의 값들이 정의되어야 한다. 실제 샘플링 과정은 다음과 같이 진행된다. 먼저, 가중치 벡터 ω 의 요소들을 식 (3)에 따라 계산한다. 그 다음, 샘플링은 두 단계로 이루어진다. 첫 단계는 가우스 커널을 구성하는 가우스 함수들 중의 하나를 선택하는 것이다. 즉 아카이브 내에서 대응하는 가중치에 따라 하나의 해를 선택하는 것과 동일하다. j 번째 가우스 함수를 선택하는 확률 p_j 는 다음과 같다.

$$p_j = \frac{\omega_j}{\sum_{r=1}^k \omega_r} \quad (5)$$

두 번째 단계는 선택된 가우스 함수를 샘플링하는 것이다. 이것은 매개변수화 정규분포에 따라 난수를 생성할 수 있는 난수생성기를 사용하거나 균일 난수생성기를 사용하여 수행할 수 있다. 예컨대 첫 단계에서 선택된 해의 첫 번째 변수 s_j^i 을 중심으로 하는 가우스 함수를 샘플링하여 난수를 생성하고 할당한다. 이 단계를 점진적으로 구축하여 하나의 해를 생성한다. 이러한 두 단계 샘플링은 식 (1)에 정의된 가우스커널 PDF G^i 를 샘플링하는 것과 동일하다. 단계 i 에서는 선택된 단일 가우스 함수 $g^i(x)$ 를 위해 표준편차도 주어져야 한다. 이때 표준편차 벡터 σ^i 전체를 계산할 필요는 없고 σ_j^i 만 알면 된다. 이러한 전반적인 과정이 각 차원

i ($1,\dots,n$)에 대해서 반복되고, 그 때마다 표준편차 σ_j^i 가 단일 차원 i 의 사용과 함께 계산된다.

3) 페로몬 갱신

앞서 언급했듯이 ACO_R의 경우에는 페로몬 정보가 하나의 아카이브로써 저장된다. 이것은 페로몬 갱신 절차는 이 아카이브 상에 어떤 갱신 작업을 해야 하는 것을 의미한다. 아카이브 T 의 크기 k 는 알고리즘의 매개변수이다. 그러나 k 는 해결할 문제의 차원의 수 보다 적지 않을 수도 있다. 페로몬 갱신은 새로 생성된 해들을 아카이브 T 에 추가하고 동일한 수의 최악의 해들을 제거함으로써 수행된다. 따라서 아카이브의 원래 크기는 변하지 않는다. 이러한 과정을 거침으로써 아카이브 내에는 최상의 해들만 유지된다.

2.4 ACO_R 알고리즘의 구조

ACO를 연속 최적화 문제에 효과적으로 응용할 수 있는 방법을 제안한다. 즉, ANN의 적정 가중치를 찾기 위하여 training 과정 동안에 연속적인 가중치 변수를 조절하여 나간다. 본 연구에서는 최적의 가중치를 찾기 위하여 BP 대신에 ACO_R을 사용하며, 신경망 모델의 목적함수는 Mean Squared Error(MSE)를 사용하며 ε^2/n 으로 나타낸다. 여기서 n 은 해(solution)벡터 내의 해의 개수를 나타낸다.

ANN을 training 하는데 사용하는 ACO_R 알고리즘의 구체적인 절차는 다음과 같다.

(단계 1) 아카이브 T 를 구성하기 위하여 k 개의 초기해를 생성한다. 각 해의 변수값은 ANN을 구성하는 가중치를 나타낸다.

(단계 2) 생성된 각 해의 목적함수를 계산한다. 해의 목적함수는 ANN의 최종 출력에 의해 결정되는 MSE 값을 나타낸다.

(단계 3) k 개의 해들을 목적함수 값에 따라 오름차순으로 정렬한다.

(단계 4) 가중치 계산에 필요한 매개변수 q 값을 지정하고 이를 이용하여 k 개 해들의 가중치를 계산한다.

(단계 5) 신규해의 개수 nos 와 개선해의 집단을 반복적으로 생성하는 횟수 N 을 지정한다.

(단계 6) $ij=1$ 로 지정한다

(단계 7) $i=1$ 로 지정한다.

(단계 8) k 개의 해 가운데 하나의 해를 선택한다. 해를 선택할 때 roulette wheel 방법^[5]을 이용한다.

(단계 9) 선택된 해를 이용하여 샘플링을 함으로서 새로운 해를 생성한다.

$i < \text{nos}$ 이면 $i=i+1$ 로 지정하고 단계 8로 간다. 아니면 다음 단계로 간다.

(단계 10) 생성된 nos 개의 신규해들의 목적함수 값을 계산한다.

(단계 11) 새로 생성된 해들을 서열에 맞추어 아카이브 T에 추가한다.

새로 생성된 해가 아카이브 내에 있는 기존해보다 우수하면 아카이브에 투입하고 아니면 투입하지 않는다. 그리고 나서 투입된 수와 같은 수의 최악의 해들을 아카이브에서 제거한다. 결국 아카이브의 크기 T는 변하지 않으며 최상의 해들만 아카이브 내에 유지된다. $ij < N$ 이면 $ij=ij+1$ 로 지정하고 단계 7로 간다. 아니면 다음 단계로 간다.

(단계 12) k 개 해들의 MSE 값 중에서 최소값을 찾고 종료한다.

3. 실험 및 분석

Table 1 Observation data

No	Input			SR	ECC
	V(m/min)	f(mm/rev)	d(mm)		
1	13	0.17	0.9	7.671	4.4
2	13	0.1125	0.6	5.252	4.4
3	70	0.055	0.9	1.491	4.57
4	13	0.055	0.3	1.66	4.4
5	41.5	0.1125	0.3	3.06	4.77
6	41.5	0.1125	0.6	3.274	4.83
7	70	0.17	0.9	7.44	4.7
8	70	0.1125	0.6	2.067	4.53
9	41.5	0.1125	0.6	3.168	4.8
10	70	0.17	0.3	2.926	4.67
11	41.5	0.1125	0.6	2.107	4.93
12	41.5	0.17	0.6	1.963	4.9
13	41.5	0.1125	0.6	2.772	4.87
14	70	0.055	0.3	2.99	4.67
15	41.5	0.055	0.6	2.517	4.93
16	41.5	0.1125	0.6	3.232	4.97
17	41.5	0.1125	0.9	3.232	5.07
18	41.5	0.1125	0.6	3.447	5.2
19	13	0.055	0.9	3.379	4.77

3.1 실험 계획

본 연구에서는 선삭공정을 시뮬레이션하는 모델을 구축하기 위하여 ANN을 사용한다.

시뮬레이션 모델의 구체적인 내용은 2.4절의 알고리즘을 이용하여 구축하고 이를 위한 컴퓨터 프로그램용 code는 MATLAB7 언어를 사용하여 작성하였다.

선삭공정에서 사용하는 입력변수는 절삭속도(cutting speed:V), 이송(feed:f), 절삭깊이(depth of cut:d)의 세 종류를 사용하며 출력변수는 표면조도(Surface roughness:SR)와 전류소모량(Electric current consumption: ECC)을 사용한다.

Punushingon and Oh[2]에서는 선삭공정을 대상으로 세 종류의 입력변수를 사용하여 실험조건에 맞추어 실제적으로 선삭가공을 하고 결과값을 측정하였다. 실험조건은 절삭속도, 이송, 절삭깊이의 세 변수 각각에 대하여 세 종류의 레벨값을 사용하였다. 그리고 각 변수의 레벨값들을 조합하여 19개의 입력 조건을 만들고 각 입력 조건에 대응하는 표면조도와 전류소모량의 출력값 즉, 실험값을 얻었고 그 결과는 Table 1에 주어져 있다. 본 연구에서는 Table 1의 입.출력 데이터를 이용하여 시뮬레이션 모델을 구축한다. 모델을 구축하는데 사용하는 ANN은 다중 레이어 신경망을 사용하는 데 이것은 한 개의 입력 레이어, 두 개의 hidden 레이어, 한 개의 출력 레이어로 이루어져 있다. 각 레이어의 뉴런의 개수는 입력 레이어는 3개, 첫 번째 hidden 레이어는 12개, 두 번째 hidden 레이어는 6개, 출력 레이어는 2개로 이루어져 있다. 이와 같은 구조를 갖는 ANN에 Table 1에 주어진 입력 데이터를 적용하였다. ANN을 training하기 위하여 ACO_R 알고리즘을 적용하고 ACO_R이 ANN의 training에 어떤 영향을 미치는지를 검토 및 분석하였다.

3.2 실험결과 및 분석

시뮬레이션에 사용되는 매개변수는 k , nos , q , ξ , N 의 다섯 종류이다. 매개변수 k 는 초기해의 집단인 아카이브의 크기 즉, 초기해의 개수를 나타낸다. 변수 nos 는 개선해 집단의 크기 즉, 아카이브

에 추가되는 해들의 최대 개수를 나타낸다. q는 가중치 계산에 사용되며 ξ 는 식 (4)에서 표준편차의 계산에 사용된다. q가 작으면 최고 순위의 해들이 강한 우선권을 가지며 q가 크면 해를 선택하는 확률은 더욱 균일하게 되고, 한편 ξ 는 그 값이 클수록 알고리즘의 수렴속도가 늦어지며, 즉 하위등급의 해가 빨리 잊혀진다^[6]. N은 개선해의 집단을 반복적으로 생성하는 횟수를 나타낸다.

위에 주어진 매개변수들을 투입하여 시뮬레이션 프로그램을 가동시키고 최종해가 되는 MSE 값을 산출하였다. 우선 매개변수 k의 값은 {15, 20, 25, 30, 40}의 5종으로 설정하여 프로그램을 가동시켰다. nos의 값은 10을 사용하였다. 사전에 nos의 값을 몇 가지 바꾸어 결과를 분석하였으나 별다른 차이가 없었기 때문에 nos가 10인 경우의 결과만 제시하였다. ξ 는 Pandian^[6]의 연구와 동일한 조건인 {0.75, 0.8, 0.85, 0.9}의 네 종류의 값을, q는 {0.06, 0.07, 0.08, 0.09}의 값을 사용하였다. ξ 와 q의 값을 조합하면 16가지 경우의 수가 생기는데 각각의 k 값에 대하여 ξ 와 q를 조합하여 16회의 시뮬레이션을 실시하였다. N의 값은 200과 500을 사용하였다. Table 2에서는 nos=10, N=200을 사용하여 시뮬레이션을 수행한 결과를 보여주고 있다. Table 2에서는 각각의 k값에 대하여 ξ 와 q를 조합하여 투입했을 때 산출되는 16개의 결과값 중에서 상위 세 개의 값을 선택하여 굵은 활자로 표시하였고 특히 k값의 각 열 중에서 최소값은 (*) 표시를 하여 나타내었다.

k는 다섯 개의 값을 사용하는데, 다섯 개의 각 열에서 상위값 세 개씩을 선택하여 비교해 보면 k가 15보다는 20~40인 경우에 상대적으로 좋은 결과

를 보여주고 있다. k가 20, 25, 30, 40인 경우를 서로 비교하면 이들 간에는 의미있는 차이가 나타나지 않는다. Table 2 내에서 전체적인 최소값은 k가 20인 경우에 0.0407임을 알 수 있다. ξ 는 네 가지 값을 사용하는데, 다섯 개의 각 열마다 굵은 활자로 표시한 상위값 세 개를 비교해 보면 ξ 가 0.85와 0.9에 이 값들이 집중되어 있다. 이로부터 ξ 는 0.75와 0.8보다는 0.85와 0.9인 경우에 더욱 좋은 결과를 나타내고 있음을 알 수 있다. q의 경우는 Table 4에서 살펴보기로 한다.

Table 3에서는 Table 2의 결과를 보다 세밀하게 분석하기 위하여 다섯 가지 매개변수의 조합을 조정하고 시뮬레이션한 결과를 보여주고 있다. nos는 Table 2와 동일하게 10을 사용하고 N을 500으로 증가시키고 ξ 는 0.85와 0.9를, k값은 20, 25, 30, 40의 값을 사용하고 q는 Table 2와 동일한 조건을 사용하여 시뮬레이션을 실시하였다. Table 3에서 각각의 k값에 대하여 산출되는 여덟 개의 결과값 중에서 상위 세 개의 값을 선택하여 굵은 활자로 표시하였고 특히 k값의 열 중에서 최소값은 (*) 표시를 하여 나타내었다. N의 경우는 Table 2와 3을 비교해 보면 그 값이 커질수록

MSE 값이 감소하여 결과가 좋아지는 것을 알 수 있으며, 반면 값이 커질수록 시뮬레이션 운용 시간은 증가한다. Table 3에서 ξ 를 기준으로 검토하면 전체적인 결과값 중에서 상위값 세 개의 분

Table 2 Simulation result 1

No.	ξ	q	k				
			15	20	25	30	40
			nos=10, N=200				
1	0.75	0.06	0.001	0.0986	0.0993	0.0926	0.083
2	0.75	0.07	0.1009	0.086	0.1005	0.0843	0.0828
3	0.75	0.08	0.0973	0.0992	0.0859	0.0844	0.0843
4	0.75	0.09	0.0875	0.0844	0.0894	0.0844	0.0837
5	0.8	0.06	0.1008	0.0844	0.098	0.0737	0.0561
6	0.8	0.07	0.0987	0.0848	0.0816	0.0722	0.0535
7	0.8	0.08	0.0874	0.0844	0.0908	0.0843	0.051
8	0.8	0.09	0.0863	0.0844	0.0606	0.0781	0.0623
9	0.85	0.06	0.1007	0.0844	0.0808	0.0837	0.0442*
10	0.85	0.07	0.0844	0.0842	0.084	0.055	0.0471
11	0.85	0.08	0.0995	0.0813	0.0551	0.0571	0.049
12	0.85	0.09	0.0853	0.0505	0.056	0.0568	0.0637
13	0.9	0.06	0.096	0.0565	0.0432*	0.0519	0.0536
14	0.9	0.07	0.0989	0.0513	0.0564	0.0534	0.0517
15	0.9	0.08	0.0805	0.0407*	0.0444	0.0618	0.0511
16	0.9	0.09	0.0757*	0.0539	0.0542	0.0476*	0.0465

Table 3 Simulation result 2

No.	ξ	q	k			
			20	25	30	40
			nos=10, N=500			
1	0.85	0.06	0.0761	0.028	0.0307	0.0355
2	0.85	0.07	0.0833	0.0155*	0.0328	0.0383
3	0.85	0.08	0.026*	0.0417	0.0427	0.0422
4	0.85	0.09	0.0316	0.0298	0.0352	0.0363
5	0.9	0.06	0.0287	0.043	0.0304	0.0269*
6	0.9	0.07	0.0356	0.0357	0.0252*	0.0436
7	0.9	0.08	0.0291	0.0403	0.0356	0.0481
8	0.9	0.09	0.0276	0.0367	0.035	0.0438

또는 ξ 가 0.85인 경우는 7회, 0.9인 경우는 5회로 비슷한 형태를 보여주고 있다. 각 열에서 최상의 값의 분포만 보더라도 0.85인 경우는 2회, 0.9인 경우는 2회로 비슷하다. 따라서 ξ 값 0.85와 0.9 사이에는 결과에 의미있는 차이가 없음을 알 수 있다. k 가 20, 25, 30, 40인 경우를 비교하기 위하여 각 열에서의 상위값 세 개를 살펴보면 Table 2와 마찬가지로 네 개의 열 사이에는 의미있는 차이가 나타나지 않는다. 네 개의 열 중에서 최소값은 k 가 25일 때 0.155를 보인다.

Table 4는 Table 3을 대상으로 하여 q 값을 기준으로 다시 정렬한 것이다. Table 4에서 상위값 세 개와 최상의 값을 기준으로 살펴보면 q 값의 변화에 따른 뚜렷한 특징을 찾아볼 수가 없다. 따라서 q 값의 변화는 시뮬레이션의 결과에 별다른 영향을 미치지 않는다고 판단할 수 있다.

Table 2, 3, 4에 주어진 결과들을 요약하면 다음과 같이 정리할 수 있다. k 는 초기해의 개수를 나타내는데 20~40의 영역에서 좋은 결과를 나타내며 구체적으로 k 가 20, 25, 30, 40인 경우에 이 값들 간에는 시뮬레이션의 결과에 큰 차이가 나타나지 않는다. 매개변수 nos 는 개선해 집단의 크기를 나타내는데 기본적으로 10을 사용하며 그 이상의 값을 사용해도 결과에 별다른 영향을 미치지 않는다. ξ 는 0.75와 0.8보다는 0.85와 0.9인 경우에 좋은 결과를 나타내고 있는데 이는 ξ 값이 커질수록 결과가 좋아지는 것을 의미한다. q 는 가중치 계산에 사용되는데 0.06, 0.07, 0.08, 0.09의 4종의 값을 사용하였는데 값의 변화에 따른 결과값의 차이가 뚜렷하게 나타나지 않는다. N 은 개선해의

집단을 반복적으로 생성하는 횟수를 나타내는데 N 의 값이 커질수록 MSE 값이 감소하며 결과가 좋아진다. 종합적으로 정리하면 시뮬레이션 결과에 가장 큰 영향을 미치는 매개변수는 ξ 와 N 이며 나머지 매개변수들은 본 연구에서 제시한 값을 사용하면 될 것으로 판단된다.

4. 결론

본 연구에서는 선삭공정의 가공 모델을 구축하기 위하여 ANN을 사용하여 시뮬레이션모델을 구축하는 방법을 제시하였다. ANN의 핵심적인 구성요소인 가중치를 찾기 위한 training 방법은 다양한데 본 연구에서는 ACO_R 에 의한 training 방법과 이를 구현하는 구체적인 알고리즘을 제시하였다. 나아가서는 ACO_R 을 구성하는 여러 가지 매개변수가 ACO_R 의 성능에 미치는 영향을 조사하기 위하여 매개변수들의 조합을 검토하고 분석하였다. 결론적으로 k 는 20~40의 값을, nos 는 10의 값을, ξ 는 0.85~0.9의 값을, q 는 0.06~0.09의 값을, N 은 500 이상의 값을 선택하면 되는 것으로 정리할 수 있다.

후 기

‘이 논문은 부경대학교 자율창의학술연구비(2016년)에 의하여 연구되었음’

REFERENCES

1. Ahilan, et al., “Modeling and prediction of machining quality in CNC turning process using intelligent hybrid decision making tools”, Applied Soft Computing, Vol. 13, No. 3, 1543-1551, 2013.
2. Charles, S. C., Punuhsingon, and Oh, S-C., “Prediction of Surface Roughness and Electric Current Consumption in Turning Operation using Neural Network with Back Propagation and Particle Swarm Optimization”, Journal of the

Table 4 Simulation result 3

No.	ξ	q	k			
			20	25	30	40
1	0.85	0.06	0.0761	0.028	0.0307	0.0355
5	0.9	0.06	0.0287	0.043	0.0304	0.0269*
2	0.85	0.07	0.0833	0.0155*	0.0328	0.0383
6	0.9	0.07	0.0356	0.0357	0.0252*	0.0436
3	0.85	0.08	0.026*	0.0417	0.0427	0.0422
7	0.9	0.08	0.0291	0.0403	0.0356	0.0481
4	0.85	0.09	0.0316	0.0298	0.0352	0.0363
8	0.9	0.09	0.0276	0.0367	0.035	0.0438

Korean Society of Manufacturing Process Engineers, Vol. 14, No. 3, 65-73, 2015.

3. Dorigo, M., Maniezzo, V. and Colomi, A., "Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics, Part B(Cybernetics), Vol. 26, No. 1, 29-41, 1996.
4. Ezilarasan et al., "An experimental analysis and measurement of process performances in machining of nimonic C-263 super alloy", Measurement, Vol. 46, No. 1, 185-199, 2013.
5. Gen, M. and Cheng, R., Genetic algorithms & Engineering Design, John Wiley & Sons, 1997.
6. Pandian, A., "Training neural networks with ant colony optimization," Master thesis, California State University, spring 2013.
7. Socha, K. and Dorigo, M., "Ant colony optimization for continuous domains", European journal of operational research, Vol. 185, No. 3, 1155-1173, 2008.
8. Socha, K. and Blum, C., "An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training", Neural Computing and Applications, Vol. 16, No. 3, 235-247, 2007.
9. Wasserman, P. D., Neural Computing: Theory and Practice, Van Nostrand Reinhold, 1989.