

절삭가공의 Neural Network 모델을 위한 ACO 및 PSO의 응용

오수철*#

*부경대학교 시스템경영공학부

Application of Ant Colony Optimization and Particle Swarm Optimization for Neural Network Model of Machining Process

Soo-Cheol Oh*#

*Department of Systems Management & Engineering, Pukyong National University, Korea

(Received 26 June 2019; received in revised form 9 July 2019; accepted 12 July 2019)

ABSTRACT

Turning, a main machining process, is a widespread process in metal cutting industries. Many researchers have investigated the effects of process parameters on the machining process. In the turning process, input variables including cutting speed, feed, and depth of cut are generally used. Surface roughness and electric current consumption are used as output variables in this study. We construct a simulation model for the turning process using a neural network, which predicts the output values based on input values. In the neural network, obtaining the appropriate set of weights, which is called training, is crucial. In general, back propagation (BP) is widely used for training. In this study, techniques such as ant colony optimization (ACO) and particle swarm optimization (PSO) as well as BP were used to obtain the weights in the neural network. Particularly, two combined techniques of ACO_BP and PSO_BP were utilized for training the neural network. Finally, the performances of the two techniques are compared with each other.

Key Words : Turning(선삭), Neural Network(신경망), ACO(개미집단최적화), PSO(입자군최적화)

1. 서 론

선삭은 기계가공 산업에서 가장 널리 사용되며 중요한 절삭가공 공정 중의 하나이다. 절삭가공에서 공정변수들의 영향을 조사하는데 목적을 둔 여러가지 연구가 진행되어 왔다. 또한, 선삭 가공에서 공정변수들이 가공의 결과에 미치는 영향을 체계적으로 조사하기 위해서 실험적 방법, 통계적 해석

도구, 혁신적 최적화 기법 등이 사용되어 왔다^[1]. 선삭공정의 공정변수들 가운데 입력 매개변수로서는 절삭속도, 이송, 절삭깊이를 주로 사용하며 출력 변수는 표면거칠기 등이 있다. 공정변수들이 결과에 미치는 효과를 나타내기 위하여 neural network(NN)과 같은 혁신적 최적화기법을 활용하여 선삭공정의 시뮬레이션 모델을 구축할 수 있다. 이 경우에 실제 현장에서 선삭가공을 할 때의 출력값과 NN을 이용한 선삭가공 공정의 시뮬레이션 모델을 이용한 출력값을 비교하면 차이가 있다. 이 차이를 최소화하기 위해서는 NN을 구성하는 가중치 값의 집합을 적절히 찾아야 하는데, NN을 이용한

Corresponding Author : scoh@pknu.ac.kr

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

모델에서는 최적의 가중치를 결정하는 것이 핵심이며 이것을 트레이닝(training)이라 한다. NN의 가중치를 찾는 트레이닝을 위해서 Back Propagation(BP)이 보편적으로 많이 이용되어 왔고 그 외 여러 가지 기법들이 사용되어져 왔다. Punuhsingon & Oh^[2]는 BP와 Particle Swarm Optimization(PSO)의 두 기법을 이용하여 트레이닝을 하고 두 기법을 비교하는 연구를 수행하였고, Oh^[3]는 Ant Colony Optimization(ACO) 기법을 이용한 트레이닝에 대한 연구를 수행하였다.

본 연구에서는 neural network(NN)을 이용한 선삭 가공 공정의 시뮬레이션 모델을 제안하고자 한다. 이 모델에서는 NN에 공정변수를 입력하면 절삭성능을 예측할 수 있다. 입력 매개변수로서는 절삭속도, 이송, 절삭깊이를 출력 변수는 표면거칠기, 전류소모량을 사용하기로 한다. 또한 NN을 구성하는 가중치 조합을 찾기 위하여 BP를 포함하여 ACO 및 PSO 기법을 활용하기로 한다. 나아가서는 ACO와 BP를 결합한 기법과 PSO와 BP를 결합한 두 가지 조합 기법을 고안하고 두 조합 기법의 우수성을 집중적으로 비교, 분석 및 검토하기로 한다.

2. NN과 트레이닝

NN은 두뇌의 학습 메카니즘에 의해 영향을 받고, 레이어 형태로 조직화되고 레이어 내에 정렬된 상호연결된 뉴런들의 집합이다. NN은 하나 또는 다중 레이어로 구성된다. 하나의 레이어와 다른 레이어의 뉴런들 간의 연결에는 가중치가 할당되어 있다. NN의 일반적인 응용에서는 주로 다중 레이어를 사용하는데 이것은 하나의 입력 레이어, 하나 이상의 히든 레이어, 하나의 출력 레이어로 이루어져 있다. NN은 일련의 입력을 적용하면 원하는 일련의 출력을 산출할 수 있도록 학습을 시킬 수 있는데 이것을 트레이닝이라 한다. 트레이닝은 일종의 최적화 문제이며 목적은 에러(error)가 최소화되도록 최적의 가중치의 집합을 찾는 것이다. 트레이닝에는 두 가지 유형이 있다^[4]. 첫 번째는 에러 함수의 음의 기울기를 유지하며 가중치를 조절함으로써 에러를 최소화한다. 이 유형에는 Back Propagation(BP)이 있는데 이것은 NN을 트레이닝할 때 가장 많이

사용하는 방법이다. 문제점으로는 국지적(local) 최소화에 쉽게 빠질 가능성이 있으며 알고리즘의 수렴속도가 너무 느리며 수렴 거동은 가중치의 초기값, 학습률 등의 변수값의 선택에 따른 의존도가 강하다. 두 번째 방법은 후보해들의 집단, 즉 각각 다른 복수의 가중치 벡터를 이용하여 작업을 한다. 후보해들은 서로 상호작용을 하며 보다 적은 에러가 달성될 수 있도록 가중치 영역을 향해 나아간다. 이 유형의 방법에는 Ant Colony Optimization(ACO)과 Particle Swarm Optimization(PSO) 등이 있다.

본 연구에서는 BP의 단점을 보완하기 위하여 ACO와 BP를 결합하고 PSO와 BP를 결합한다. 첫 번째는 ACO를 이용하여 전체적인(global) 최적해를 찾고난 후에 BP를 결합하여 국지적인 최적해를 찾아 나가는 형태이다. 두 번째는 PSO를 이용하여 전체적인 최적해를 찾고난 후에 BP를 결합하여 국지적인 최적해를 찾아 나가는 형태이다.

2.1 ACO와 BP

ACO는 1990년대 초에 이산(discrete) 최적화 문제에 응용하기 위해 도입된 최적화 기법이다. ACO 알고리즘은 초기에는 조합 최적화 문제들을 해결하는데 도입되었고 그 이후 ACO를 연속적인 영역에 적용할 수 있도록 확장한 ACO_R 기법이 Socha and Dorigo^[5]에 의해 제안되었다. ACO에 포함되는 매개변수는 다섯 종류인데 k는 아카이브를 구성하는 초기해의 개수, nos는 신규해 집단의 크기, q는 가중치 계산에 사용되는 변수, ξ 는 표준편차의 계산에 사용되는 변수, N은 신규해의 집단을 반복적으로 생성하는 회수를 나타낸다. ACO 기법을 NN에 도입하자면, ACO의 각 해의 변수값은 NN을 구성하는 가중치를 나타내며, 각 해의 목적함수는 NN의 최종 출력에 따르는 MSE(Mean squared error) 값을 나타낸다. 본 연구에서는 ACO와 BP를 결합하여 ACO_BP라 명명하는데, 먼저 ACO를 수행한 후 결과적으로 NN의 가중치 집합을 산출하고 이것을 이어받아 또 다시 BP를 수행하고 최종적인 가중치 집합을 도출한다. ACO_BP 알고리즘은 Oh^[3]의 알고리즘을 기본으로 일부 확장하였으며 구체적인 절차는 다음과 같다.

- (단계 1) 아카이브를 구성하기 위하여 k개의 초기해들을 생성한다.
- (단계 2) 생성된 각 해의 목적함수를 계산한다.
- (단계 3) k개의 해들을 목적함수 값에 따라 오름차순으로 정렬한다.
- (단계 4) 매개변수 q 값을 지정하고 이를 이용하여 k개 해들의 가중치를 계산한다.
- (단계 5) 신규해의 개수와 개선해의 집단을 반복적으로 생성하는 횟수를 지정한다.
- (단계 6) ij=1로 지정한다
- (단계 7) i=1로 지정한다.
- (단계 8) k 개의 해 가운데 하나의 해를 roulette wheel 방법⁶⁾을 이용하여 선택한다.
- (단계 9) 선택된 해를 이용하여 샘플링을 함으로서 새로운 해를 생성한다. i<nos 이면 i=i+1 로 지정하고 단계 8로 간다. 아니면 다음 단계로 간다.
- (단계 10) 새로운 해들의 목적함수 값을 계산한다.
- (단계 11) 새로 생성된 해들을 서열에 맞추어 아카이브에 추가한다. 생성된 해가 아카이브 내에 있는 기존해보다 우수하면 아카이브에 투입하고 아니면 투입하지 않는다. 투입된 수와 같은 수의 최악의 해들을 아카이브에서 제거한다. ij<N 이면 ij=ij+1 로 지정하고 단계 7로 간다. 아니면 다음 단계로 간다.
- (단계 12) k개의 해 중에서 MSE 값이 최소인 해를 찾는다.
- (단계 13) 최소값에 해당하는 변수값들을 NN에 투입하고 BP 알고리즘을 이용하여 MSE 값이 최소가 되는 가중치 값을 찾는다.

2.2 PSO와 BP

PSO는 하늘을 나는 새(bird) 집단의 이동을 모의 실험하여 최상의 해를 탐색하는 알고리즘의 하나이다⁷⁾. PSO에서는 각각의 새는 입자라 부르는데 이 입자들은 임의의 속도로 날아서 몇 번의 반복 후에 글로벌 최상의 위치를 찾는다. 매번 반복 시에 개별 입자는 모멘텀과 자신의 최상의 위치, 그리고 자기 이웃들의 최상의 위치에 근거하여 자신의 속도 벡터를 조정하고, 자기가 날아갈 새로운 위치를 계산한다. PSO에서는 탐색 공간을 가정하고 입자의 전체적인 숫자는 n으로, i 번째 입자의 위치는 X_i 로

표현할 수 있다. i 번째 입자가 지금까지 찾아낸 최상의 위치는 P_{ib} 로, 전체 입자들의 무리가 지금까지 찾아낸 최상의 위치는 P_g 로, i 번째 입자의 속도는 V_i 로 나타낸다. 모든 입자는 목적함수에 의해 결정되는 Fitness value를 가진다. 입자의 위치와 속도는 아래의 공식으로 주어진다.

$$v_i(t+1) = v_i(t) + c1 * rand() * (p_i(t) - x_i(t)) + c2 * rand() * (p_g(t) - x_i(t)) \quad \dots\dots\dots (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad \dots\dots\dots (2)$$

여기서 t는 시간, c1, c2는 양의 값을 갖는 가속도 상수, rand()는 0과 1 사이의 난수이다. PSO의 개념을 NN에 대입하자면 PSO에서 입자의 위치는 NN의 가중치 값에 해당하고, PSO에서 fitness value는 NN의 MSE 값이 된다. 본 연구에서는 PSO와 BP를 결합하여 PSO_BP라 명명하는데, 먼저 PSO를 수행한 후 결과적으로 NN의 가중치 집합을 산출하고 이것을 이어받아 또 다시 BP를 수행하여 최종적인 가중치 집합을 도출한다. PSO_BP 알고리즘의 구체적인 절차는 다음과 같다.

- (단계 1) 그룹 내의 입자수와 세대수를 결정한다.
- (단계 2) 하나의 입자의 위치와 속도를 랜덤하게 결정한다.
- (단계 3) 입자의 위치에 의거해서 NN을 구성한다.
- (단계 4) 트레이닝 데이터 집합을 이용해서 현재 입자의 Fitness value를 평가한다. 현재 입자의 P_{ib} 를 결정한다.
- (단계 5) 단계 2, 3, 4를 입자수 n만큼 반복한다. 전체 입자들 중에서 P_g 를 결정한다.
- (단계 6) 최대 반복세대수에 도달했으면 단계 10으로, 아니면 다음 단계로 간다.
- (단계 7) 현재 입자들 중에서 최상의 입자는 저장한다. 모든 입자들의 위치, 속도를 갱신하여 새로운 입자 그룹을 생성한다.
- (단계 8) 새로운 입자들의 Fitness value를 평가한다. 최악의 입자를 이전 단계의 최상 입자로 교체한다. 각 입자의 P_{ib} 와 전체 입자들의 P_g 를 결정한다.
- (단계 9) 열 세대 동안 P_g 가 변하지 않으면 다음 단계로, 아니면 단계 6으로 간다.

(단계 10) P_g 를 출력한다.

(단계 11) BP 알고리즘을 이용하여 주어진 사이클 동안 P_g 주변을 탐색한다. 만약 탐색결과가 P_g 보다 좋으면 탐색 결과를 출력한다. 아니면 P_g 를 출력한다.

3. 실험계획

본 연구에서는 선삭공정을 시뮬레이션하는 NN 모델을 구축한다. 먼저 NN을 트레이닝하고 가중치 조합을 결정하기 위하여 BP, ACO와 PSO의 세 기법을 조합하여 ACO_BP와 PSO_BP라는 두 알고리즘을 만들고 이들을 적용하여 NN을 가동한다. NN 모델의 구축을 위한 컴퓨터 프로그램은 MATLAB 언어를 사용하여 코딩하였다.

3.1 입력 데이터

선삭공정의 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)의 두 종류를 사용한다. Punuhsingon and Oh^[2]에서는 선삭공정을 대상으로 실험조건에 맞추어 실제적인 선삭가공을 하고 결과값을 측정하였다. 실험조건은 절삭속도, 이송, 절삭깊이의 세 가지 변수마다 세 종류의 값을 사용하고 각 변수의 값들을 조합하여 19개의 입력 조건을 만들고 각 입력 조건에 대응하는 표면조도와 전류소모량의 출력값을 얻었다. 본 연구에서는 그 중의

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

일부인 10개의 데이터 집합을 사용하며 Table 1에 주어졌다.

3.2 NN의 구조와 BP

NN은 입력 레이어, 히든 레이어, 출력 레이어로 이루어진다. 본 연구에서는 입력변수가 세 개, 출력변수가 두 개이므로 입력 레이어의 뉴런의 수는 3, 출력 레이어의 뉴런의 수는 2를 사용한다. 히든 레이어는 두 가지 형태를 사용한다. 첫째는 뉴런의 수가 6인 한 개의 히든 레이어를 사용한다. 둘째는 각각 12개, 6개의 뉴런을 갖는 두 개의 히든 레이어를 사용한다. 즉, NN에 BP를 적용하여 시뮬레이션을 수행하는데 히든 레이어의 수는 1과 2의 두 가지, 입력 데이터 집합의 수는 5, 7, 10의 세 가지를 조합하여 6개의 경우를 구성하였다. 각 경우에 대하여 NN에 BP를 적용하여 MSE 값을 산출하였고 그 결과는 Fig. 1 ~ Fig. 6에 주어졌다. BP의 cycle 수는 히든 레이어가 한 개인 경우에는 500회, 히든 레이어가 두 개인 경우에는 1000회를 사용하였다. 히든 레이어가 한 개인 경우의 결과는 Fig. 1 ~ Fig. 3에 주어지는데 세 개의 그래프 모두 BP의 cycle 횟수가 증가할수록 MSE 값이 감소하는 것을 알 수 있다. 히든 레이어가 두 개인 경우의 결과는 Fig. 4 ~ Fig. 6에 주어지는데, Fig. 4와 Fig. 5에서는 BP의 cycle 횟수가 증가할수록 MSE 값이 점점 감소하지만 Fig. 6에서는 MSE 값이 과동 형태를 보여주고 있다.

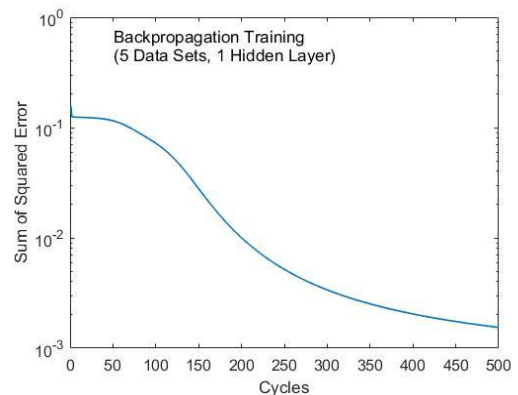


Fig. 1 Values of the MSE for BP (Hidden layer=1, data set=5)

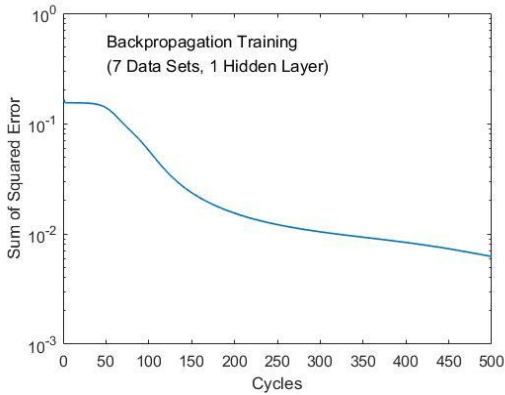


Fig. 2 Values of the MSE for BP (Hidden layer=1, data set=7)

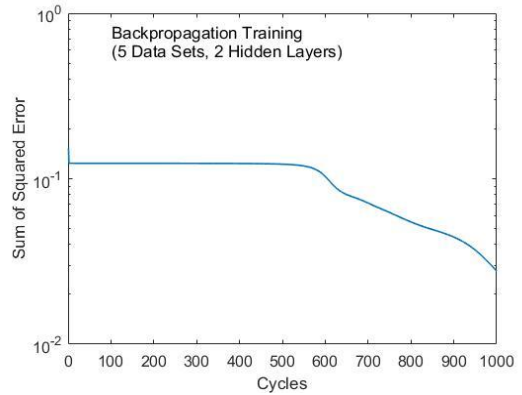


Fig. 4 Values of the MSE for BP (Hidden layer=2, data set=5)

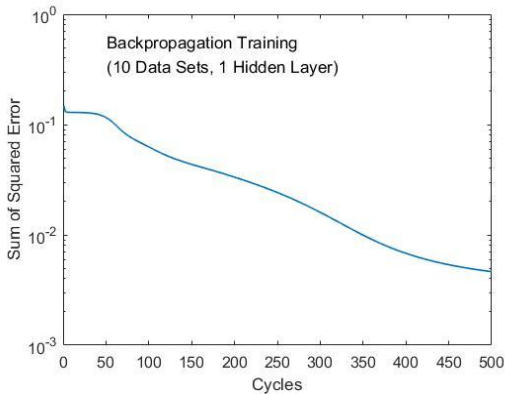


Fig. 3 Values of the MSE for BP (Hidden layer=1, data set=10)

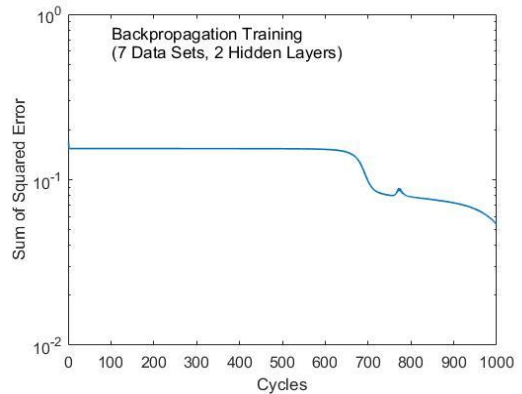


Fig. 5 Values of the MSE for BP (Hidden layer=2, data set=7)

MSE 값을 요약 정리해 놓은 Table 2를 살펴보면 히든 레이어가 한 개인 경우에는 데이터 집합의 수가 5일 때 MSE 값이 가장 우수하고, 히든 레이어가 두 개인 경우에도 데이터 집합의 수가 5일 때 MSE 값이 가장 우수하다. 표에서 히든 레이어의 개수를 기준으로 비교하면, 데이터 집합의 개수에 관계없이 레이어의 개수가 한 개인 경우가 cycle 횟수가 적음에도 불구하고 두 개인 경우보다 항상 우수하다. 요약하면 NN에 BP를 적용한 결과는 히든 레이어 개수가 많다고 좋은 것은 아니며 데이터 집합의 개수가 많다고 좋아지는 것도 아니다. 따라서

본 연구의 NN에서는 ACO, ACO_BP, PSO, PSO_BP를 적용하기 위하여 히든 레이어의 수는 1, 데이터 집합의 수는 5를 사용하기로 한다.

3.3 ACO_BP 및 PSO_BP의 구성

ACO와 BP 기법 둘 다 동일한 NN 구조를 사용해야 하기 때문에, ACO와 BP를 결합한 ACO_BP에서는 BP의 경우와 동일한 NN 구조를 적용한다. 즉, 입력 레이어는 세 개의 뉴런, 출력 레이어는 두 개의 뉴런, 히든 레이어의 수는 1을 사용하며 뉴런의 수는 6을 사용한다. 또한 데이터 집합은 Table 1에

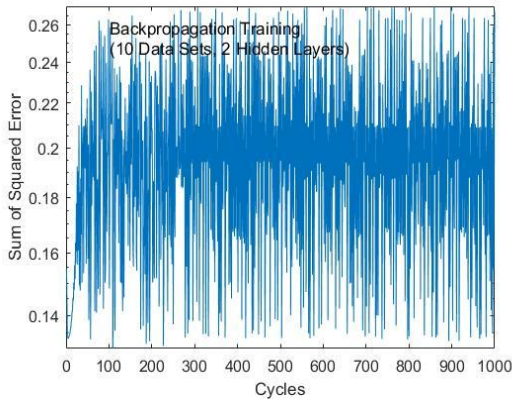


Fig. 6 Values of the MSE for BP (Hidden layer=2, data set=10)

Table 2 Values of the MSE for BP

		No. of data sets		
		5	7	10
No. of hidden layers	1	0.00159	0.00423	0.00366
	2	0.12273	0.15398	0.23965

서 위에서 순차적으로 다섯 개를 사용한다. ACO에 포함되는 매개변수는 다섯 종류이며 nos , N , k , ξ , q 이다. 여기서 nos 는 10, N 은 500을 사용하며 k 는 네 종류, ξ 는 세 종류, q 는 네 종류의 값을 사용하였다. 그룹 내의 입자수는 100, 세대수는 500으로 설정하였다. PSO와 BP를 결합한 PSO_BP에서는 ACO_BP와 동일한 NN 구조를 사용하기로 한다. PSO 기법에 사용되는 매개변수로는 가속상수 c_1 , c_2 와 난수 r 등이 있다. 이 연구에서는 c_1 , c_2 의 값을 각각 1, 1.5, 2로 구성하고 이들 아홉 개의 조합에 대하여 PSO 프로그램을 가동하였다.

4. 실험결과 및 분석

4.1 ACO와 ACO_BP, PSO와 PSO_BP의 비교

Table 3에서 Table 6까지는 히든 레이어가 1개인 neural network을 대상으로 ACO, ACO_BP, PSO,

PSO_BP의 네 개의 알고리즘을 이용하여 시뮬레이션을 수행하고 MSE 값을 산출하였다.

Table 3에는 ACO 알고리즘을 이용하여 시뮬레이션을 수행하고 MSE 값을 도출하였다. k 는 20, 25, 30, 40의 네 개의 값을, ξ 는 0.85, 0.9, 0.95의 세 종류의 값을, q 는 0.06, 0.07, 0.08, 0.09의 네 개의 값을 사용하였다. 각각의 k 값에 대하여 ξ 와 q 를 조합하여 12회의 시뮬레이션을 실시하였고 각 열마다 최소값 세 개는 (*)표로 표현하였다. Table 3에서 k 가 20인 경우에 25, 30, 40보다 미세하게 우수한 결과를 보여주고 있고 k 가 25, 30, 40인 경우에는 이들 간에 의미있는 차이를 찾기가 어렵다. ξ 는 0.85, 0.9일 때의 결과가 0.95보다 우수하다. q 는 0.06, 0.07, 0.08, 0.09의 4개 값의 변화에 따른 유의미한 결과를 찾기는 어렵다. Table 4에서는 ACO_BP 알고리즘을 이용하여 시뮬레이션을 수행하고 MSE 값을 도출하였다. Table 3과 동일한 k , ξ , q 값과 동일한 조합을 사용하였다. Table 4에서 k 가 20, 25, 30, 40인 경우의 결과를 비교하면 이들 간에 의미있는 차이를 찾기는 어렵다. ξ 는 0.85, 0.9, 0.95인 경우의 결과를 비교하면 이들 간에 의미있는 차이를 찾기는 어렵다. q 는 0.06, 0.07, 0.08, 0.09의 값들의 변화에 따른 유의미한 결과를 찾기는 어렵다. Table 3과 Table 4의 결과를 종합하면 ACO_BP 알고리즘을 적용할 시에는 네 개의 k 값, 세 개의 ξ 값, 네 개의 q 값을 이용하여 조합을 구성하고 시뮬레이션을 수행한 후에 전체 결과 값들 중에서 최소값을 찾으려면 된다. Table 3과 Table 4에 주어진 MSE 값을 기준으로 ACO와 ACO_BP 알고리즘을 비교하면, ACO보다 ACO_BP인 경우에 MSE 값이 62% 감소한다. 즉 ACO나 BP를 단독으로 사용하는 것보다는 두 기법을 조합한 ACO_BP를 사용하면 훨씬 우수한 결과를 얻을 수 있다. Table 4를 보면 평균값 열의 최소값이 0.0002이다. 이 수치의 의미는 실제 선삭공정에서 주어지는 결과값과 시뮬레이션 모델의 결과값이 거의 차이가 없음을 나타낸다.

Table 5에서는 PSO 알고리즘을 이용하여 시뮬레이션을 수행하고 MSE 값을 도출하였다. c_1 , c_2 의 값을 각각 1, 1.5, 2로 부여하고 c_1 , c_2 를 조합하여 아홉 개 경우에 대하여 NN을 가동하고 결과를 표

**Table 3 Values of the MSE for ACO
(hidden layer=1, data set=5)**

ξ	q	k				Ave.
		20	25	30	40	
0.85	0.06	0.0008	0.0004	0.0001*	0.0003*	0.0004*
	0.07	0.0084	0.0066	0.0003	0.0005	0.0040
	0.08	0.0005	0.0002*	0.0003*	0.0013	0.0006*
	0.09	0.0007	0.0002*	0.0003	0.0003*	0.0004*
0.9	0.06	0.0006	0.0012	0.0002*	0.0005*	0.0006*
	0.07	0.0001*	0.0068	0.0013	0.0012	0.0023
	0.08	0.0002*	0.0006	0.0005	0.0071	0.0021
	0.09	0.0001*	0.0016	0.0010	0.0038	0.0016
0.95	0.06	0.0008	0.0009	0.0013	0.0012	0.0010
	0.07	0.0002	0.0003	0.0020	0.0041	0.0017
	0.08	0.0007	0.0003	0.0013	0.0008	0.0008
	0.09	0.0002	0.0003*	0.0076	0.0006	0.0022

**Table 4 Values of the MSE for ACO_BP
(hidden layer=1, data set=5)**

ξ	q	k				Ave.
		20	25	30	40	
0.85	0.06	0.0002	0.0002	0.0001*	0.0002	0.0002*
	0.07	0.0067	0.0065	0.0001*	0.0002	0.0034
	0.08	0.0002	0.0001	0.0001	0.0001	0.0002*
	0.09	0.0002	0.0001*	0.0002	0.0002	0.0002*
0.9	0.06	0.0002	0.0002	0.0001*	0.0001*	0.0002*
	0.07	0.0001*	0.0066	0.0002	0.0002	0.0018
	0.08	0.0001	0.0002	0.0002	0.0004	0.0002*
	0.09	0.0001*	0.0003	0.0003	0.0002	0.0002*
0.95	0.06	0.0002	0.0002	0.0006	0.0001*	0.0003
	0.07	0.0001	0.0001	0.0004	0.0002	0.0002*
	0.08	0.0002	0.0001*	0.0002	0.0001*	0.0002*
	0.09	0.0001*	0.0001*	0.0066	0.0001	0.0017

에 명시하였다. 또한 c1, c2의 각 조합에 대하여 5 회씩 가동을 하고 평균값을 구하였고 각 열마다 세 개의 최소값을 (*)표로 표현하였다. Table 5에서 평균값 열을 살펴보면 c1이 1.5와 2, c2가 1.5와 2일 때의 조합이 다른 조합보다 미세하게 우수한 결과를 보여주고 있음을 알 수 있다. Table 6에서는 PSO_BP 알고리즘을 이용하여 시뮬레이션을 수행하

**Table 5 Values of the MSE for PSO
(hidden layer=1, data set=5)**

c1	c2	1st.	2nd.	3rd.	4th.	5th.	Ave.
1	1	0.1476	0.1493	0.1485	0.1472	0.1484	0.1482
	1.5	0.1488	0.1443	0.1504	0.1477	0.1411	0.1464
	2	0.1493	0.1453	0.1474*	0.1494	0.1492	0.1481
1.5	1	0.1488	0.1483	0.1488	0.1470*	0.1472	0.1480
	1.5	0.1459*	0.1436*	0.1481*	0.1503	0.1336*	0.1443*
	2	0.1467*	0.1434*	0.1489	0.1485	0.1397*	0.1455*
2	1	0.1487	0.1470	0.1480*	0.1492	0.1497	0.1485
	1.5	0.1488	0.1477	0.1485	0.1388*	0.1439	0.1455*
	2	0.1459*	0.1352*	0.1485	0.1341*	0.1367*	0.1401*

**Table 6 Values of the MSE for PSO_BP
(hidden layer=1, data set=5)**

c1	c2	1st.	2nd.	3rd.	4th.	5th.	Ave.
1	1	0.0035*	0.0077	0.0076	0.0066*	0.0075	0.0066
	1.5	0.0065*	0.0078	0.0072*	0.0076	0.0068*	0.0072
	2	0.0073	0.0074	0.0074	0.0076	0.0055*	0.0071
1.5	1	0.0073	0.0077	0.0077	0.0078	0.0076	0.0076
	1.5	0.0070	0.0077	0.0075	0.0075	0.0074	0.0074
	2	0.0076	0.0033*	0.0072*	0.0075	0.0061*	0.0063*
2	1	0.0077	0.0064*	0.0071*	0.0071	0.0077	0.0072
	1.5	0.0077	0.0025*	0.0073	0.0058*	0.0073	0.0061*
	2	0.0018*	0.0075	0.0076	0.0068*	0.0075	0.0062*

고 MSE 값을 도출하였는데, Table 5와 동일한 c1, c2의 조합을 사용하였다. Table 6에서 평균값 열을 살펴보면 c1이 1.5와 2, c2가 1.5와 2일 때의 조합이 다른 조합보다 미세하게 우수한 결과를 보여주고 있음을 알 수 있다.

Table 5와 Table 6을 종합하면 PSO_BP 알고리즘을 적용할 시에는 c1은 1.5와 2, c2는 1.5와 2의 네 개의 조합에 대하여 각각 프로그램을 가동한 후 평균값을 구하고 그 중에서 최소값을 찾으면 된다. Table 5와 Table 6에 주어진 MSE 값을 기준으로 PSO와 PSO_BP 알고리즘을 비교하면, PSO보다는 PSO_BP인 경우가 MSE 값이 95% 감소한다. 즉 PSO나 BP를 단독으로 사용하는 것보다는 두 기법을 조합한 PSO_BP를 사용하면 훨씬 우수한 결과를 얻을 수 있다.

4.2 ACO_BP와 PSO_BP의 비교

Table 3부터 Table 6까지를 살펴보면 ACO보다는 ACO_BP가 우수하고 PSO보다는 PSO_BP가 우수함을 알 수 있다. 더 나아가 ACO_BP와 PSO_BP를 비교해 보면 Table 4의 평균값 열의 최소값 0.0002와 Table 6에서 평균값 열의 최소값 0.0061을 비교할 때 ACO_BP의 결과가 PSO_BP의 결과보다 훨씬 우수함을 알 수 있다. 따라서 BP, ACO, PSO 기법들을 개별적으로 사용하는 것보다는 조합하여 사용하는 것이 더 좋은 결과를 주며, 그 중에서도 ACO_BP 기법을 사용하는 것이 가장 좋은 결과를 주는 것을 알 수 있다.

5. 결 론

본 연구에서는 NN을 이용한 선삭가공 공정의 시뮬레이션 모델을 제안하였다. NN을 구성하는 가중치 조합을 찾기 위하여 BP를 포함하여 ACO 및 PSO 기법을 활용하였다. 분석에 의하면 PSO_BP 보다 ACO_BP 기법을 사용하는 것이 가장 좋은 결과를 제공한다. 즉, 본 연구에서 제안하는 NN에서는 트레이닝 기법으로 ACO_BP를 사용하면 최적의 가중치 조합을 얻을 수 있다. 본 연구의 NN을 이용하면 실제적인 선삭가공 공정에 거의 유사한 결과를 제공하는 성능이 우수한 시뮬레이션 모델을 구축할 수 있다.

후 기

“이 논문은 부경대학교 자율창의학술연구비(2018년)에 의하여 연구되었음”

REFERENCES

1. Ciurana, J., Arias, G., Ozel, T., “Neural Network Modeling and Particle Swarm Optimization (PSO) of Process Parameters in Pulsed Laser Micromachining of Hardened AISI H13 Steel,” *Materials and Manufacturing Processes*, Vol. 24, No. 3, pp. 358-368, 2009.
2. Punuhsingon, C. S. C., 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 Korean Society of Manufacturing Process Engineers*, Vol. 14, No. 3, pp. 65-73, 2015.
3. Oh, S-C., “Prediction of machining performance using ANN and Training by using ACO,” *Journal of the Korean Society of Manufacturing Process Engineers*, Vol. 16, No. 6, pp. 125-132, 2017.
4. Rakitianskaia, A. and Engelbrecht, A., “Weight Regularisation in Particle Swarm Optimisation Neural Network Training,” *IEEE Symposium on Swarm Intelligence*, 2014.
5. Socha, K. and Dorigo, M., “Ant colony optimization for continuous domains,” *European Journal of Operations Research*, Vol. 185, No. 3, pp. 1155-1173, 2008.
6. Gen, M. and Cheng, R., “Genetic algorithms & Engineering Design,” John Wiley & Sons, Inc., 1997.
7. Zhang, J.-R., Zhang, J., Lok, T.-M., Lyu, M. R., “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network Training,” *Applied Mathematics and Computation*, Vol. 185, No. 2, pp. 1026-1037, 2007.