

Machine Layout Decision Algorithm for Cell Formation Problem Using Self-Organizing Map

Yong-Deok Jeon[†]

Department of Liberal Arts, Dongyang University

자기조직화 신경망을 이용한 셀 형성 문제의 기계 배치순서 결정 알고리즘

전 용 덕[†]

동양대학교 교양학부

Self Organizing Map (SOM) is a neural network that is effective in classifying patterns that form the feature map by extracting characteristics of the input data. In this study, we propose an algorithm to determine the cell formation and the machine layout within the cell for the cell formation problem with operation sequence using the SOM. In the proposed algorithm, the output layer of the SOM is a one-dimensional structure, and the SOM is applied to the parts and the machine in two steps. The initial cell is formed when the formed clusters is grouped largely by the utilization of the machine within the cell. At this stage, machine cell are formed. The next step is to create a flow matrix of the all machine that calculates the frequency of consecutive forward movement for the machine. The machine layout order in each machine cell is determined based on this flow matrix so that the machine operation sequence is most reflected. The final step is to optimize the overall machine and parts to increase machine layout efficiency. As a result, the final cell is formed and the machine layout within the cell is determined. The proposed algorithm was tested on well-known cell formation problems with operation sequence shown in previous papers. The proposed algorithm has better performance than the other algorithms.

Keywords : Cell Formation, Machine Layout, Self-Organizing Map, Flow Matrix

1. 서 론

다품종 소량생산방식은 제품표준화에 의한 대량생산방식에 비하여 비효율적이며 비경제적이다. 이와 같은 다품종 소량생산방식의 어려운 문제점을 해결하고 대량생산방식의 장점을 살리기 위하여 그룹 테크놀로지(Group Technology : GT)개념이 도입되었다. 셀 생산방식(cellular manufactur-

ing)은 그룹 테크놀로지의 개념을 제조시스템에 적용한 것이다. 기업들은 셀 생산 방식을 적용함으로써 준비시간의 단축, 재공품재고 통제, 자재취급비용 감소, 준비비용과 직/간접 노동비용의 감소, 품질 개선, 자재흐름 유연화, 기계활용도 제고, 공간활용의 개선, 종업원의 사기 앙양 등 다양한 장점들을 누릴 수 있다[3]. 셀 생산방식 시스템의 개발과 시행을 위한 기본 단계로서 셀 형성 과정은 각각의 기계 셀에 할당된 부품군을 처리하는 상호 독립적인 여러 개의 셀들을 구성하는 것이다. 셀을 형성하기 위해서는 셀 형성 문제(cell formation problem, CFP)를 해결해야 한다[9].

Received 1 June 2019; Finally Revised 11 June 2019;
Accepted 12 June 2019

[†] Corresponding Author : ydjeon@dyu.ac.kr

셀 형성 문제는 m 대의 기계와 n 개의 부품으로 구성된 생산체계에서, $a_{ij}=0, 1$ 로 특정 부품이 특정 기계를 사용하는지 여부를 명시하는 경우(이를 일반적으로 CFP라 한다), $a_{ij}=1, 2, \dots, m$ 으로 기계사용 순서를 명시한 경우, $a_{ij}=t$ 로 특정 기계로 작업을 수행하는 시간을 명시하는 경우 셀 능력을 만족시키도록 형성하는 CaCFP(Capacitated CFP)로 분류된다. 기계사용 순서를 명시한 $a_{ij}=1, 2, \dots, m$ 문제는 다시 부품 이동 비용(cost)을 최소화시키도록 셀(군)을 형성하는 CoCFP(Cost CFP)와 기계 배치순서 LCFP(layout CFP)를 결정하는 문제로 분류된다. LCFP와 CoCFP의 a_{ij} 는 작업순서(sequence)로 값이 주어진다. 이 문제에 대해 셀 내부의 기계 배치순서는 고려하지 않고, 단지 셀 내부의 불연속적인 기계간과 다른 셀들로의 부품 이동비용만을 최소화시키도록 셀을 형성하는 문제를 CoCFP라 하며, 단순히 셀 내부의 기계 배치순서만을 고려하는 경우를 LCFP라 한다[10]. 본 연구에서는 LCFP에 초점을 맞춘다.

셀 형성 문제를 풀기위해 많은 연구가 진행되었다. 대표적인 방법으로는 군집분석(cluster analysis), 그래프분할법(graph partitioning approaches)과 메타휴리스틱 방법(metaheuristic approaches)이 있다[2]. 셀 형성 문제는 NP-난제(NP-Hard)인 조합 최적화 문제로, 휴리스틱의 다항 시간 알고리즘이 존재하지 않고 있어 부득이 메타휴리스틱 방법을 적용하고 있다[1, 16, 13]. 대표적인 메타 휴리스틱 방법으로는 유전자 알고리즘(genetic algorithm, GA), 진화 알고리즘(evolutionary algorithm, EA), 군집최적화(particle swarm optimization, PSO), GRASP(greedy randomized adaptive search procedure) 등을 적용하고 있음에도 불구하고 아직까지 다양한 모든 문제들에 대해 최적 해를 구하는 알고리즘이 존재하지 않고 있다. CoCFP는 Teymourian et al.[18]이 있으며, LCFP에 대해서는 Nair와 Narendran [15]의 군집분석 기법의 CASE, Mutingi와 Onwubolu[14]의 군 유전자 알고리즘(group genetic algorithm, GGA), Mahadavi와 Mahadevan[11]의 CLASS, Maddavi et al.[12]의 흐름 행렬 기반 휴리스틱 알고리즘, Lee[10]의 흐름행렬 기반 역-구성 알고리즘(RCA) 등이 있다.

Mahadavi와 Mahadevan[11]의 CLASS 알고리즘과 Mahadavi et al.[12]의 휴리스틱 알고리즘은 모두 기계 i 에서 j 로의 연속적인 순방향 제품 이동 횟수를 계산한 흐름 행렬(flow matrix)에 기반을 두고 있으며, 각각 9단계와 15단계의 과정을 수행한다. 이들 방법은 셀을 형성하는데 있어 체계적인 방법 대신 휴리스틱 방법을 적용하여 기계 대수가 많아 질수록 셀을 형성하는데 어려움이 있다. Lee[10]의 흐름행렬 기반 역-구성 알고리즘(RCA)은 체계적인 방법으로 셀을 형성한다.

최근 생산 현장의 광범위한 문제에 인공지능 기법을 적용하고 있다. 인공지능 기법은 생산 현장의 복잡한 문

제를 해결하는 데 있어서 효과적인 수단으로 알려져 있다. 인공지능에 기초한 방법 중 신경망(neural networks)은 학습 능력을 통하여 실제 생산 현장의 문제에 대하여 유연하고 적응력 있는 해를 제공해 왔다. 셀 형성 문제에도 신경망의 이러한 성질을 이용하여 훌륭한 효과를 보았다 [6, 7]. 자기조직화 신경망(Self Organizing Map : SOM)은 입력데이터의 특징을 추출하여 지도를 형성하는 패턴 분류에 효과적인 신경망이다. 본 연구에서는 SOM의 이러한 성질을 이용하여 LCFP에 대해 셀 형성과 셀 내부의 기계 배치순서를 결정하는 알고리즘을 제안한다.

2. LCFP의 효율성 평가 척도

특정 부품이 특정 기계로 작업하는 순서가 결정된 셀 형성 문제는 $n \times m (i=1, 2, \dots, n, j=1, 2, \dots, m)$ 의 부품 P_i 와 기계 M_j 행렬에 대해 s_{ij} 는 부품 i 가 기계 j 로 작업을 수행하는 순서로 주어진다. LCFP는 셀을 형성함과 더불어 셀 내부의 기계 배치순서도 결정하는 문제이다. 이 문제에 대해 다음과 같이 정의한다.

- 부품 : $P_i, i=1, 2, \dots, n, n$: 부품 수
- 기계 : $M_j, j=1, 2, \dots, m, m$: 기계 수
- 셀 : $C_k, k=1, 2, \dots, c (c < m)$
- P_k : k 번째 셀에 속한 부품 수
- M_k : 셀 k 에 속한 기계 수
- M_{sk} : 셀 k 의 크기(size) = $P_k \times M_k$
- s_{ij} : i 번째 제품이 j 번째 기계에서 작업(operation)이 수행되는 순서
- f_{ij} : i 번째 기계에서 j 번째 기계로 연속적인 순방향 이동 수(number of consecutive forward movement)
- v_k : 셀 k 의 빈칸(void) 수
- N_{mk} : 셀 k 내부 부품들의 연속적인 순방향 이동 수
- N_{tk} : 셀 k 내부 부품들의 총 이동 수
= $(n_k \times m_k) - v_k - n_k$
- N_{opk} : 셀 k 내부의 작업 (operation) 수
- N_{ops} : 총 작업 수 = $\sum_{i=1}^n \max_j s_{ij}$
- N_{mov} : 총 작업 이동 수 = $N_{ops} - n$

LCFP의 기계 배치 효율성은 Mahadavi와 Mahadevan[11]이 제안한 식 (1)의 ACMI(average cell movement index)와 식 (2)의 OMI(overall movement index)로 평가한다.

$$ACMI = \frac{\sum_{k=1}^c P_k \times CMI_k}{n}, \text{ where } CMI_k = \frac{N_{mk}}{N_{tk}} \quad (1)$$

$$OMI = \frac{\sum_{k=1}^c N_{mk}}{N_{mov}} \quad (2)$$

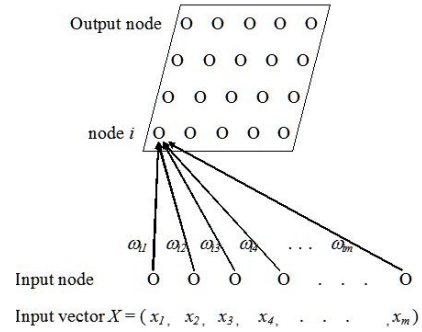
ACMI와 OMI는 모두 연속적인 순방향 작업 순서 효율성에만 초점을 맞추고 있다. 만약, N_{mk} 를 극대화하도록 1개의 셀로 구성한다면 ACMI와 OMI를 증가시킬 수 있다. 그러나 척도만을 강조하면 셀 내부의 기계 활용도 측면은 저하된다. 이러한 단점을 보완하고자, Mahadavi et al.[12]은 셀 내부의 기계 활용도에 대한 셀 형성 효율성 평가 척도로 식 (3)의 ACUI(average cell utilization index)를 제안하였다.

$$ACUI = \frac{\sum_{k=1}^c CUI_k}{C}, \text{ where } CUI = \frac{N_{opk}}{N_{sk}} \quad (3)$$

3. 자기 조직화 신경망

SOM은 비지도 학습(unsupervised learning)을 사용하는 신경망 알고리즘의 일종으로 Kohonen[8]에 의하여 제시된 경쟁학습(competitive learning) 모형이다. 경쟁학습은 입력 패턴에 대해서 경쟁층의 노드 중에서 승자 노드를 결정하여 승자 노드의 연결강도만을 입력 패턴에 대응하여 조절하는 학습방법이다. SOM은 일반 경쟁학습 모델과는 달리 학습과정에서 이웃(neighborhood)의 개념을 사용한다. SOM에서는 승자 노드뿐만 아니라 승자 노드의 이웃 노드들의 연결강도도 함께 조절된다. 그 결과로 출력층에 입력 패턴의 유사성을 반영하는 지도를 스스로 형성한다. 즉, SOM은 입력 패턴들을 출력 노드들에 의한 위상학적인 공간으로 매핑(mapping)할 수 있는 능력을 가진다.

SOM의 일반적인 구조는 <Figure 1>과 같다. SOM은 입력층(input layer)과 출력층(output layer)으로 두 개의 층으로 구성되어 있다. 입력층은 유클리디안 공간상의 m 차원 벡터를 입력벡터로 가질 수 있으며 일차원 또는 이차원의 격자(lattice) 출력층과 연결되어 있다. 입력벡터는 모든 출력노드와 연결되어 있고 연결선은 연결강도(weight)를 가진다. 승자노드의 결정은 입력벡터와 연결강도벡터의 거리를 계산하여 거리가 가장 작은 노드를 선택한다. 여기서 말하는 거리는 유클리디안 공간상의 벡터 거리(euclidean distance)를 의미한다.



<Figure 1> General Structure of SOM

승자 노드를 결정하고 난 후에는 코호넨의 학습규칙에 따라 노드의 연결강도를 조정해야 하는데 이 규칙은 식 (4)로 표현된다.

$$w(new)_{ij} = w(old)_{ij} + \alpha(x_i - w_{ij}(old)) \quad (4)$$

여기서

$w(new)_{ij}$: 신경세포 i, j 사이의 조절된 후 연결강도

$w(old)_{ij}$: 신경세포 i, j 사이의 조절되기 전 연결강도

α : 학습률($0 < \alpha \leq 1$)

x_i : 신경세포 i 의 활성화값

SOM에서 신경망을 학습시키는 과정은 다음과 같다.

절차 1: 연결강도를 초기화한다. 학습률과 이웃의 범위를 정한다.

절차 2: 입력층에 입력패턴(입력벡터)을 제시한다.

절차 3: 각 출력노드의 연결강도벡터와 입력패턴의 거리를 구해서 가장 짧은 거리의 승자노드(승자 신경세포)를 구한다.

절차 4: 승자노드와 이웃노드의 연결강도를 조절한다.

절차 5: 절차 2-4 과정을 모든 입력벡터에 대해 반복한다.

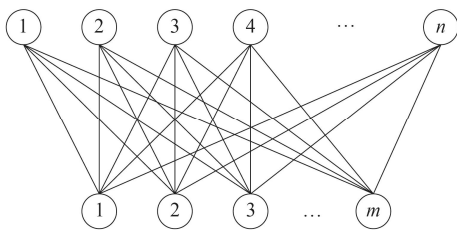
절차 6: 이웃의 범위와 학습률을 감소하면서 이웃의 범위가 자기 자신이 될 때까지 절차 5 과정을 반복한다.

학습률은 신경망 학습 결과의 성능저하가 그리 크지 않는 범위 내에서 적당한 값을 정한다. 이웃의 범위는 처음에는 출력층의 모든 노드를 포함하도록 하였다가, 점차 이웃의 범위를 줄여나가 마지막에는 승자노드만 포함하도록 한다.

4. 제안하는 알고리즘

본 연구에서는 SOM의 출력층을 <Figure 2>와 같이 1차원 구조로 하고, 부품 및 기계에 대하여 2단계로 SOM을 적용하는 알고리즘을 제안한다. 학습률은 사전 실험을

통하여 정한다. SOM의 출력층이 1차원구조일 경우 출력 노드의 수를 충분히 크게 만들면(예를 들어 입력벡터의 2배) 입력데이터(부품)를 입력벡터(가공 기계)가 유사한 순서로 1차원 출력노드에 펼쳐놓을 수 있다[5]. 이 상태에서 기계를 입력벡터로 하여 같은 방법을 적용하면 부품과 기계가 모여 군집을 형성하게 된다. 형성된 군집중 ACUI가 가장 높게 형성되는 것부터 기계-부품 그룹으로 묶어 나가며, 이 과정을 모든 기계가 기계-부품 그룹에 포함될 때까지 진행한다. 이렇게 하여 형성된 것이 초기 기계-부품 그룹(셀)이다. 이 단계에서 기계 그룹이 형성된다.



<Figure 2> SOM with One-Dimensional Output Layer

각 기계 그룹에서 다음의 과정을 통하여 기계 배치순서를 결정한다.

먼저 기계별 연속적인 순방향 이동 빈도수를 계산한 전체 기계의 흐름행렬(flow matrix)을 작성한다. 각 기계 그룹에서 기계 배치순서를 이 흐름행렬을 기반으로 하여 다음과 같이 결정한다.

각 기계 그룹 내에 있는 기계만의 흐름행렬을 전체 기계의 흐름행렬에서 추출한다. 이 흐름행렬을 고려하여 각 기계 그룹의 기계 배치순서를 다음과 같이 결정한다.

기계 그룹의 기계가 2개일 경우는 다음과 같이 기계 배치순서를 결정한다. 기계 그룹 내의 기계 흐름행렬에서 큰 값을 선택한다. 해당 기계순서의 셀을 형성한다. 기계 흐름행렬에서 값이 동일할 경우 임의로 선택한다.

기계 그룹의 기계가 3개 이상일 경우는 다음과 같이 기계 배치순서를 결정한다. 기계 그룹 내의 기계 흐름행렬에서 가장 큰 값을 선택한다. 해당 기계순서의 묶음을 형성한다. 가장 큰 값이 복수일 경우 각각 기계순서의 묶음을 형성한다. 각각의 묶음이 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 기계그룹 내의 기계 흐름행렬에서 그 다음으로 큰 값을 선택한다. 해당 기계순서의 묶음을 형성한다. 형성된 묶음이 기존 묶음(앞 단계에서 구한 묶음)에 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 선택한 값이 복수일 경우는 각각 기계순서의 묶음을 형성한다. 형성된 묶음이 기존 묶음에 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 각각의 묶음이 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 지금까지의 묶음 형성 결과에 맞지 않는 묶음은 삭제한다. 이와 같은 과정을 기계 그룹

내의 모든 기계의 배치순서가 결정될 때까지 반복한다. 최종적으로 형성된 묶음의 기계순서로 셀을 형성한다.

마지막 단계로 전체 기계와 부품에 대해 보다 기계 배치 효율성을 증대시키는 최적화를 수행한다.

이렇게 하여 최종 셀이 형성되고 기계 배치순서가 결정된다.

본 연구에서 제안하는 알고리즘의 절차를 요약하면 다음과 같다.

단계 1: 부품을 입력벡터로 하는 SOM을 구성한다.

단계 2: 부품에 대하여 SOM을 학습시킨다.

단계 3: 단계 2의 결과에서 기계를 입력벡터로 하는 SOM을 구성한다.

단계 4: 기계에 대하여 SOM을 학습시킨다.

단계 5: 초기 기계-부품 그룹(셀)의 형성

절차 1: 부품과 기계가 모여 이루는 군집 중 ACUI가 가장 높게 형성되는 것부터 기계-부품 그룹으로 묶어 나간다.

절차 2: 모든 기계가 기계-부품 그룹에 포함되었을 때 멈춘다.

절차 3: 이렇게 하여 초기 기계-부품 그룹(셀)을 형성한다. 이 단계에서 기계 그룹이 형성된다.

단계 6: 기계의 흐름행렬(flow matrix) 작성

단계 7: 각 기계 그룹에서 기계 배치순서 결정

(1) 기계 그룹의 기계가 2개일 경우

절차 1: 기계 그룹 내에 있는 기계만의 흐름행렬을 전체 기계의 흐름행렬에서 추출한다.

절차 2: 기계 그룹 내의 기계 흐름행렬에서 큰 값을 선택한다. 값이 동일할 경우 임의로 선택한다.

절차 3: 해당 기계순서의 셀을 형성한다.

(2) 기계 그룹의 기계가 3개 이상일 경우

절차 1: 기계 그룹 내에 있는 기계만의 흐름행렬을 전체 기계의 흐름행렬에서 추출한다.

절차 2: 기계 그룹 내의 기계 흐름행렬에서 가장 큰 값을 선택한다.

절차 3: 해당 기계순서의 묶음을 형성한다. 가장 큰 값이 복수일 경우 각각 기계순서의 묶음을 형성한다.

절차 4: 각각의 묶음이 좌우로 연결될 경우 연결하여 한 묶음으로 합친다.

절차 5: 기계 그룹 내의 기계 흐름행렬에서 그 다음으로 큰 값을 선택한다.

절차 6: 해당 기계순서의 묶음을 형성한다. 형성된 묶음이 기존 묶음(앞 절차에서 구한 묶음)에 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 선택한 값이 복수일 경우는 각각 기계순서의 묶음을 형성한다.

- 절차 7 : 형성된 묶음이 기존 묶음에 좌우로 연결될 경우 연결하여 한 묶음으로 합친다. 각각의 묶음이 좌우로 연결될 경우 연결하여 한 묶음으로 합친다.
- 절차 8 : 지금까지의 묶음 형성 결과에 맞지 않는 묶음은 삭제한다.
- 절차 9 : 이와 같은 과정을 기계 그룹 내의 모든 기계의 배치순서가 결정될 때까지 반복한다.
- 절차 10 : 최종적으로 형성된 묶음의 기계순서로 셀을 형성한다.

- 단계 8 : 전체 기계와 부품에 대해 보다 기계 배치 효율성을 증대시키는 최적화 수행
- 단계 9 : 최종 셀의 형성 및 기계 배치순서 결정

5. 수치예제

<Table 1>의 수치예제는 Nair와 Narendran[15]이 제시한 작업순서가 주어진 8대의 기계와 20개의 부품이 있는 셀 형성 문제이다. <Table 1>의 수치예제에 대하여 본 연구에서 제안하는 알고리즘을 적용하면 다음과 같다.

<Table 1> 8×20 Problem

Machine	Part																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1	2					1	1		3		1	1		1	3		1	
2			1	1		1	4						2					2		2
3		2						2	3		2		2	3		2	1			2
4			5	2		2	2			2								1		1
5	2				2	5				3		1			1	2				
6	1				1			2	1		3			2						3
7			3	3		3	3				1	2						4		4
8			4	4		4	1											3		5

SOM에 적용시키기 위해서 <Table 2>와 같이 기계 작업순서를 제외한 문제로 변환한다.

<Table 2> Machine-Part Matrix without Operation Sequence

Machine	Part																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1	1					1	1		1		1	1		1	1		1	
2			1	1		1	1							1				1	1	1
3		1						1	1		1		1	1		1	1		1	1
4			1	1		1	1			1								1		1
5	1				1	1				1		1			1	1				
6	1				1			1	1		1			1						1
7			1	1		1	1			1	1							1		1
8			1	1		1	1											1	1	1

- 단계 1 : 부품을 입력벡터로 하는 <Figure 2>와 같은 구조의 SOM을 구성한다. 이 경우 <Figure 2>에서 m , n 은 각각 20, 40이다.

- 단계 2 : 부품에 대하여 SOM을 학습시킨 결과, 출력노드에 <Table 3>과 같이 입력벡터(가공기계)가 유사한 순서로 펼쳐진다.

- 단계 3 : <Table 3>의 행렬에서 기계를 입력벡터로 하는 <Figure 2>와 같은 구조의 SOM을 구성한다. 이 경우 <Figure 2>에서 m , n 은 각각 8, 16이다.

- 단계 4 : 기계에 대하여 SOM을 학습시킨 결과는 <Table 4>와 같다.

<Table 3> Output of SOM Learning for Part

Machine	Part																			
	11	14	2	8	13	16	19	9	17	1	5	15	10	12	6	20	4	7	18	3
1	1	1	1	1	1	1	1	1	1											1
2			1													1	1	1	1	1
3	1	1	1	1	1	1	1	1	1											
4													1		1	1	1	1	1	1
5									1	1	1	1	1	1	1					
6									1	1	1	1	1	1	1					
7	1															1	1	1	1	1
8																1	1	1	1	1

<Table 4> Output of SOM Learning for Machine

Machine	Part																			
	11	14	2	8	13	16	19	9	17	1	5	15	10	12	6	20	4	7	18	3
5									1	1	1	1	1	1	1					
6									1	1	1	1	1	1		1				
1	1	1	1	1	1	1	1	1	1											1
3	1	1	1	1	1	1	1	1	1											
2			1													1	1	1	1	1
7	1														1	1	1	1	1	1
4													1		1	1	1	1	1	1
8															1	1	1	1	1	1

- 단계 5 : <Table 4>에서 보는 바와 같이 ‘1’이 군집을 이루는 것을 볼 수 있다. 부품과 기계가 모여 이루는 군집중 ACUI가 가장 높게 형성되는 것부터 기계-부품 그룹으로 묶어 나간다. 모든 기계가 기계-부품 그룹에 포함되었을 때 멈춘다. 이렇게 하여 <Table 5>에 나타낸 바와 같이 기계-부품 그룹(셀)을 형성한다. 이 단계에서 기계 그룹이 형성된다.

<Table 5> Cell without Operation Sequence

Machine	Part																			
	11	14	2	8	13	16	19	9	17	1	5	15	10	12	6	20	4	7	18	3
5									1	1	1	1	1	1	1					
6									1	1	1	1	1	1		1				
1	1	1	1	1	1	1	1	1	1	1										1
3	1	1	1	1	1	1	1	1	1	1										
2			1													1	1	1	1	1
7	1														1	1	1	1	1	1
4													1		1	1	1	1	1	1
8															1	1	1	1	1	1

- <Table 5>를 원 문제의 기계 작업순서를 포함시켜 나타내면 <Table 6>과 같다. 형성된 기계 그룹은 (5, 6), (1, 3), (2, 7, 4, 8)이다.

단계 6 : <Table 1>에서 기계별 연속적인 순방향 이동 빈도수를 계산한 전체 기계의 흐름행렬(flow matrix)을 작성하면 <Table 7>과 같다.

<Table 6> Initial Cell with Operation Sequence

Machine	Part																				
	11	14	2	8	13	16	19	9	17	1	5	15	10	12	6	20	4	7	18	3	
5									2	2	2	1	3	1	5						
6								2		1	1	2	1	3		3					
1	3	1	1	1	1	1	1	1	3											2	
3	2	3	2	2	2	2	2	3	1												
2		2														1	2	1	4	2	1
7	1													2	3	4	3	3	4	3	
4												2			2	1	2	2	1	5	
8															4	5	4	1	3	4	

<Table 7> Flow Matrix for 8×20 Problem

	1	2	3	4	5	6	7	8
1		1	5			1	1	
2	1		1	2		1		1
3	1				1			
4		2			1		3	
5	1					1	1	
6			1	1	2		1	
7		1	1			1		4
8				2	1		1	

단계 7 : 각 기계 그룹에서 기계 배치순서를 이 흐름행렬을 기반으로 하여 다음과 같이 결정한다.

기계 그룹 (5, 6)의 경우는 기계 그룹의 기계가 2개 일 경우이다. 기계 그룹 내에 있는 기계만의 흐름행렬을 전체 기계의 흐름행렬에서 추출하면 <Table 8>과 같다. 기계 그룹 내의 기계 흐름행렬에서 큰 값을 선택하면 2이다. 해당 기계순서(6, 5)의 셀을 형성한다.

기계 그룹 (1, 3)의 경우도 기계 그룹의 기계가 2개 일 경우이다. 같은 방법으로 해당 기계순서(1, 3)의 셀을 형성한다.

기계 그룹 (2, 7, 4, 8)의 경우는 기계 그룹의 기계가 3개 이상일 경우이다. 기계 그룹 내에 있는 기계만의 흐름행렬을 전체 기계의 흐름행렬에서 추출하면 <Table 9>와 같다. 기계 그룹 내의 기계 흐름행렬에서 가장 큰 값을 선택하면 4이다. 해당 기계순서의 묶음을 형성하면 (7, 8)이다. 기계 그룹 내의 기계 흐름행렬에서 그 다음으로 큰 값을 선택하면 3이다. 해당 기계순서의 묶음을 형성하면 (4, 7)이다. 형성된 묶음이 기존 묶음 (7, 8)에 좌로 연결된다. 연결하여 한 묶음 (4, 7, 8)으로 합친다. 기계 그룹 내의 기계 흐름행렬에서 그 다음으로 큰 값을 선택하면 2이다. 선택한 값이 복수이므로 각각 기계순서의 묶음 (2, 4), (4, 2), (8,

4)을 형성한다. 묶음 (4, 2), (8, 4)가 지금까지의 묶음 형성 결과에 맞지 않는 묶음이므로 삭제한다. 묶음 (2, 4)가 기존 묶음 (4, 7, 8)에 좌로 연결된다. 연결하여 한 묶음 (2, 4, 7, 8)으로 합친다. 기계 그룹 내의 모든 기계의 배치순서가 결정되었으므로 종료한다. 최종적으로 형성된 묶음은 (2, 4, 7, 8)이다. 최종적으로 형성된 묶음의 기계순서(2, 4, 7, 8)로 셀을 형성한다.

<Table 8> Flow Matrix for Machine Group(5, 6)

	5	6
5		1
6	2	

<Table 9> Flow Matrix for Machine Group(2, 4, 7, 8)

	2	4	7	8
2		2		1
4	2		3	
7	1			4
8		2	1	

모든 셀에서 기계의 배치순서가 결정되었다. 형성된 결과를 블록대각구조로 나타내면 <Table 10>과 같다.

<Table 10> Final Cell with Operation Sequence

Machine	Part																				
	1	5	15	10	12	11	14	2	8	13	16	19	9	17	6	20	4	7	18	3	
6	1	1	2	1	3								2			3					
5	2	2	1	3	1									2	5						
1						3	1	1	1	1	1	1	1	1	3					2	
3						2	3	2	2	2	2	2	2	3	1						
2						2										1	2	1	4	2	1
4					2											2	1	2	2	1	5
7						2	1									3	4	3	3	4	3
8																4	5	4	1	3	4

단계 9 : 최종 셀의 형성 및 기계 배치순서를 결정한 결과는 <Table 10>이다.

6. 실험 결과와 분석

본 연구에서 제안하는 알고리즘의 성능을 평가하기 위해 Tam[17]의 12×19, Nair와 Narendran[15]의 8×20과 25×40, Harhalakis et al.[4]의 20×20 문제에 대해 적용하여 본다. Nair와 Narendran[15]의 8×20 문제는 수치예제에서 해보았다. 나머지 문제들은 <Table 11>에 제시되어 있다. 이 데이터들이 실험 데이터로 선정된 이유는 기존 알고리즘으로 얻은 해가 알려져 있기 때문에 기존 알고리즘과 본 연구에서 제안하는 알고리즘의 결과를 비교하여 제안하는 알고리즘의 성능을 평가할 수 있는 벤치마킹 데이터로 적합하기 때문이다. 25×40 및 12×19, 20×20 문제에 대해서는 결과만을 <Table 12>에 제시하였다.

(b) 12×19 problem

Part	Machine												
	3	5	1	2	6	4	8	9	11	10	7	12	
1			1			2	3						
2			1	2	3	4	5				6		
3			1	2		3	5	6			4		
8	1	2		3	4	5	6	7					
10	1				2	3	4						
9	1	2			3	4	5	6					
7					1	2	3	4					
4			1			2		4			3		
6					1		4	5		2	3		
5			1		2			5		3	4		
12					1					3	2		
11					1							2	
13										1	2		
16										1	3	2	
17										1	2		
15										1	2	3	4
14										1		2	3
18										1			2
19										1		2	3

(c) 20×20 problem

Part	Machine																			
	3	11	2	13	14	16	17	4	15	6	7	8	19	20	10	1	12	9	18	5
2	3	1	2																	
4	1	2	3																	
19	1	3	2																	
11	3			2									1							
6		1			2	3	4													5
7					2	3														1
15				1	2	3	4													
5								1	2	3	4									
8				2				5	1	3									4	
13							4		3	1	2									
16									1	3	2									
3												1	3	2						
10												3	1	2						
18												1	2	3	4					
14			4									1			2	3				
20															2	3	4			1
17															2	3	4			1
9			3												4	5	2	1		
12																5	4	1	2	3
1														5		2	1	3	4	

8×20 문제와 25×40 문제에 대한 제안하는 알고리즘의 결과를 CASE, CLASS, CCA, RCA와 비교한 결과는 각각 <Table 13>과 <Table 14>에 제시되어 있다. 제안하는 알고리즘은 기존의 여러 알고리즘과 비교하여 성능이 우

수함을 알 수 있다.

4개의 실험 데이터에 제안하는 알고리즘과 다른 알고리즘 결과를 비교한 결과는 <Table 15>에 제시되어 있다. <Table 15>에서 ACMI, OMI와 ACUI의 3가지 성능평가

<Table 13> Comparison of Proposed Algorithm with Other Algorithms for 8×20 Problem

Cell No.	CASE		CLASS, CCA, RCA		Proposed Algorithm	
	Machine	Part	Machine	Part	Machine	Part
1	(3,1)	{2,8,9,11,13,14,16,17,19}	(1,3)	{2,8,9,11,12,14,16,17,19}	(6,5)	{1,5,10,12,15}
2	(4,7,8,2)	{3,4,6,7,18,20}	(2,4,7,8)	{3,4,6,7,18,20}	(1,3)	{2,8,9,11,12,14,16,17,19}
3	(5,6)	{1,5,10,12,15}	(6,5)	{1,5,10,12,15}	(2,4,7,8)	{3,4,6,7,18,20}
ACMI(%)	21.7%		50.0%		50.0%	
OMI(%)	22.0%		39.0%		39.0%	
ACUI(%)	100.0%		100.0%		100.0%	

<Table 14> Comparison of Proposed Algorithm with Other Algorithms for 25×40 Problem

Cell No.	CASE		CLASS		CCA, RCA		Proposed Algorithm	
	Machine	Part	Machine	Part	Machine	Part	Machine	Part
1	14,13,15,22	18,32	9,8,10,18,7,4,16,1,2,17	1,2,5,7,10,15,16,17,19,21,22,28,30,36,38	8,10,9	10,19,21,22,28,38	8,10,9	10,19,21,22,28,38
2	18,4,7,16	1,5,7,16,17,30	24,20,3,11,25	3,9,12,13,14,33	18,7,16,4	1,5,7,16,17,30	18,7,16,4	1,7,16,17,30
3	19,5	8,15,23,24,31	21,6,5,19	8,11,23,24,25,29,31,35,40	21,6	11,25,27,29,35,40	21,6	11,25,27,29,35,40
4	20,3,11,25	3,9,13,14,33	23,12,22,15,14,13	4,6,18,20,26,27,32,34,37,39	25,1,2,17	2,12,36	25,1,2,17	2,12,36
5	21,6	11,25,27,29,35,40	-	-	20,3,11	3,9,13,14,33	20,3,11	3,9,13,14,33
6	23,12	4,6,20,26,34,37,39	-	-	5,19	8,15,23,24,31	5,19	8,15,23,24,31
7	1,2,17,24	2,12,36	-	-	22,15,14,13	18,32	22,15,14,13	18,32
8	9,8,10	10,19,21,22,28,38	-	-	23,12,24	4,6,20,26,34,37,39	23,12,24	4,5,6,20,26,34,37,39
ACMI(%)	46.5%		48.8%		52.7%		54.0%	
OMI(%)	26.3%		30.5%		32.6%		33.3%	
ACUI(%)	83.5%		22.4%		83.3%		83.6%	

<Table 15> Comparison of Algorithms

Pro-blem	CASE				CLASS				CCA				RCA				Proposed Algorithm			
	No. of cells	ACMI	OMI	ACUI	No. of cells	ACMI	OMI	ACUI	No. of cells	ACMI	OMI	ACUI	No. of cells	ACMI	OMI	ACUI	No. of cells	ACMI	OMI	ACUI
8×20	3	21%	22%	100.0%	3	50.0%	39.0%	100.0%	3	50.0%	39.0%	100.0%	3	50.0%	39.0%	100.0%	3	50.0%	39.0%	100.0%
12×19	-	-	-	-	2	65%	50%	-	-	-	-	-	2	65.2%	52.7%	57.1%	2	66.7%	56.4%	58.5%
20×20	-	-	-	-	4	65%	41%	-	-	-	-	-	6	69.5%	40.7%	85.5%	6	71.4%	40.7%	83.0%
25×40	8	46.5%	26.3%	83.5%	4	48.8%	30.5%	22.4%	8	52.7%	32.6%	83.3%	8	52.7%	32.6%	83.3%	8	54.0%	33.3%	83.6%

척도를 살펴보면 제안하는 알고리즘은 4개 데이터 모두에서 ACMI, OMI와 ACUI의 3개 성능척도 거의 모두에 대해 기존의 알고리즘보다 더 좋은 해를 얻어 성능이 우수한 알고리즘이라 할 수 있다.

7. 결 론

본 연구에서는 셀의 기계 배치순서를 결정하는 문제에 대해 인공지능 기법을 적용한 알고리즘을 제안하였다. 인공지능에 기초한 방법 중 신경망은 학습 능력을 통하여 실제 생산 현장의 문제에 대하여 유연하고 적응력 있는 해를 제공한다. 자기조직화 신경망은 입력데이터의 특징을 추출하여 지도를 형성하는 패턴 분류에 효과적인 신경망이다. 본 연구에서는 SOM의 이러한 성질을 이용하여 LCFP에 대해 셀 형성과 셀 내부의 기계 배치순서를 결정하는 알고리즘을 제안하였다.

본 연구에서 제안하는 알고리즘은 4개 데이터 모두에 대해 ACMI, OMI와 ACUI의 3개 성능척도 거의 모두에 대해 기존의 알고리즘보다 더 좋은 해를 얻어 성능이 우수한 알고리즘이라 할 수 있다.

References

- [1] Diaz, J.A., Luna, D., and Luna, R., A GRASP Heuristic for the Manufacturing Cell Formation Problem, *Trabajos de Investigación Operativa*, 2012, Vol. 20, Issue 3, pp. 679-706.
- [2] Goncalves, J.F. and Resende, M.G.C., An Evolutionary Algorithm for Manufacturing Cell Formation, *Computers & Industrial Engineering*, 2004, Vol. 47, Issue 2-3, pp. 247-273.
- [3] Ham, I., Hitomi, K., and Yoshida, T., *Group Technology : Production Methods in Manufacture*, Kluwer-Nijhoff, Boston, MA, 1985.
- [4] Harhalakis, G., Nagi, R., and Proth, J.M., An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications, *International Journal of Production Research*, 1990, Vol. 28, No. 1, pp. 185-198.
- [5] Jeon, Y.D. and Kang, M.K., A Self-Organizing Neural Networks Approach to Machine-Part Grouping in Cellular Manufacturing Systems, *Journal of Society of Korea Industrial and Systems Engineering*, 1998, Vol. 21, No. 48, pp. 123-132.
- [6] Jeon, Y.D. and Kang, M.K., Grouping of parts reflecting changes of manufacturing conditions : An algorithm based on the self-organizing neural networks, *Journal of the Korean Institute of Plant Engineering*, 1998, Vol. 3, No. 2, pp. 241-251.
- [7] Jeon, Y.D., Machine-Part Grouping with Alternative Process Plan : An algorithm based on the self-organizing neural networks, *Journal of Society of Korea Industrial and Systems Engineering*, 2016, Vol. 39, No. 3, pp. 83-89.
- [8] Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 1982, Vol. 43, Issue 1, pp. 59-69.
- [9] Kumar, C.S. and Chandrasekharan, M.P., Grouping efficacy : A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, *International Journal of Production Research*, 1990, Vol. 28, No. 2, pp. 233-243.
- [10] Lee, S.U., Machine Layout Decision Algorithm for Cellular Formation Problem, *Journal of The Korea Society of Computer and Information*, 2016, Vol. 21, No. 4, pp. 47-54.
- [11] Mahadavi, I. and Mahadevan, B., CLASS : An Algorithm for Cellular Manufacturing System and Layout Design Using Sequence Data, *Robotics and Computer-Integrated Manufacturing*, 2008, Vol. 24, Issue 3, pp. 488-497.
- [12] Mahadavi, I., Shirazi, B., and Paydar, M.M., A Flow Matrix-based Heuristic Algorithm for Cell Formation and Layout Design in Cellular Manufacturing System, *International Journal of Advanced Manufacturing Technology*, 2008, Vol. 39, Issue 9-10, pp. 943-953.
- [13] Murugan, M. and Selladurai, V., Formation of Machine Cells/Part Families in Cellular Manufacturing Systems Using an ART-Modified Single Linkage Clustering Approach-A Comparative Study, *Jordan Journal of Mechanical and Industrial Engineering*, 2011, Vol. 5, No. 3, pp. 199-212.
- [14] Mutingi, M. and Onwubolu, G.C., *Manufacturing System, Chapter 10. Integrated Cellular Manufacturing System Design and Layout Using Group Genetic Algorithms*, Interopen.com, 2012, pp. 205-222.
- [15] Nair, G.J. and Narendran, T.T., CASE : A Clustering Algorithm for Cell Formation with Sequence Data, *International Journal of Production Research*, 1998, Vol. 36, No. 1, pp. 157-179.
- [16] Nouri, H., Tang, S.H., Tuah, B.T.H., Ariffin, M.K.A., and Samin, R., Metaheuristic Techniques on Cell For-

mation in Cellular Manufacturing System, *Journal of Automation and Control Engineering*, 2013, Vol. 1, No. 1, pp. 49-54.

- [17] Tam, K.Y., An Operation Sequence Based Similarity Coefficient for Part Family Formations, *Journal of Manufacturing Systems*, 1988, Vol. 9, Issue 1, pp. 55-68.
- [18] Teymourian, E., Mahadavi, I., and Kayvanfar, V., A New

Cell Formation Model Using Sequence Data and Handling Cost Factors, *Proceedings of International Conference on Industrial Engineering and Operations Management*, 2011, Kuala Lumpur, Malaysia, Jan, 22-24.

ORCID

Yong-Deok Jeon | <http://orcid.org/0000-0002-9530-7050>