韓國經營科學會誌
第30권 第3號
2005年 9月

대단위 부품-기계 군집 문제를 위한 Look-ahead 휴리스틱 알고리듬

백종관* · 백준걸** · 김창욱***

A Look-ahead Heuristic Algorithm for Large-scale Part-Machine Grouping Problems

Jong-Kwan Baek* · Jun-Geol Baek** · Chang Ouk Kim***

■ Abstract ■

In this paper, we consider a multi-objective machine cell formation problem. This problem is characterized as determining part route families and machine cells such that total sum of inter-cell part movements and maximum machine workload imbalance are simultaneously minimized. Together with the objective function, alternative part routes and the machine sequences of part routes are considered in grouping part route families. Due to the complexity of the problem, a two-phase heuristic algorithm is proposed. And we developed an *n*-stage look-ahead heuristic algorithm that generalizes the roll-out algorithm. Computational experiments were conducted to verify the performance of the algorithm.

Keyword : Part-Machine Grouping Problem, Look-ahead Heuristic, Multi-Objective Function, Alternative Part Routes, Operation Sequence, Roll-Out Algorithm

1. 서 론

오늘날 고객의 요구는 빠르게 변화하고 있으며 이러한 변화를 만족시키기 위해 제조 현장은 다양한 종류의 제품을 동시에 생산할 수 있도록 구성되어지고 있다. 특히 job shop 형태의 제조 현장은 많

논문접수일 : 2004년 10월 25일 논문게재확정일 : 2005년 5월 2일
* 서일대학 산업시스템경영과
** 인덕대학 산업시스템경영과
*** 연세대학교 정보산업공학과

은 기계들로 이루어져 있으며 다양한 제품을 동시에 생산하기 때문에 매우 복잡한 생산 흐름을 가진다. 따라서 빠르게 변화하는 고객의 요구를 만족시키기 위해서는 복잡한 생산 흐름을 효과적으로 관리하는 것이 매우 중요하다.

일반적으로 job shop 환경에서의 복잡한 생산 흐름을 효과적으로 관리하기 위해서 셀형 제조(cellular manufacturing)가 적용되어진다. 셀형 제조는 job shop 형태의 시스템에서 부품의 흐름을 단순화하기 위해 제안된 방법으로, 유사한 특성을 갖는 부품(part)들을 군(group)으로 묶고, 이 부품을 가공하는 기계(machine)들을 셀(cell)로 묶음으로써 효율적인 생산이 가능하도록 제안된 방법이다. 셀형 제조를 위한 부품-기계 군집 알고리듬(part-machine grouping algorithm)은 현장에서 기계 셀의 위치와 부품의 흐름을 결정하며 부품 군과 이에 따른 기계 셀을 얼마나 효과적으로 구성하느냐에 따라 부품의 이동, 재공품 수, 인건비, 그리고 총 생산시간 등의 지표를 좌우한다[3].

현재까지 부품-기계 군집 문제를 해결하기 위해 다양한 형태의 방법이 제시되었는데, 대표적인 연구로는 Kusiak *et al.*[10]이 제시한 행렬이론 방법, Kusiak[9]이 제시한 수리계획 방법, Lee and Garcia-Diaz[11]이 제시한 그래프 이론 방법, Venugopal and Narendran[13]이 제시한 인공신경망 방법 등이 있다. 그러나 앞에서 언급한 대부분의 연구들은 하나의 부품이 하나의 부품가공경로(part route)를 갖는다는 조건 하에서 제시된 방법으로서, 오늘날의 생산 현장에서와 같이 하나의 부품이 다수의 대체 부품가공경로(alternative part route)를 지니고 있는 경우에 대해 적용되기 힘들다는 문제점을 지닌다.

최근 들어 Caux *et al.*[4], Hwang and Ree[7]와 Yin and Yasuda[16] 등에 의해 대체가공경로를 고려한 부품-기계 군집 알고리듬에 대한 연구가 진행되었는데, 이 연구들에서는 부품의 기계 셀간 총 이동거리(total inter-cell part movements)를 최소화시키는 단일 목적을 지니고 있다. 그러나 실제 제조현장에서는 부품의 이동거리를 최소화시키는 것도 중요하지만 각각의 기계에 할당되는 부하를 평준화시키는 것도 매우 중요한 목적이 된다. 또한 이들의 연구는 부하 평준화에 대해 고려하지 않았으며, 부품 군과 기계 셀의 수를 미리 정해진 값으로 제한함으로써 최적의 부품 군과 기계 셀을 형성하기 힘들다는 문제점을 지닌다.

이러한 문제를 해결하기 위해서 Zhao and Wu[17]는 대체 부품가공경로를 갖는 부품-기계 군집 문제에 대해 기계 부하의 불균형(machine workload imbalances)과 기계 셀간 이동거리(inter-cell part movements)를 최소화하기 위한 변형된 유전자 알고리듬을 제시하였다. Zhao and Wu의 연구는 대체 부품가공경로가 존재하는 상황에서 여러 목적함수를 동시에 고려할 수 있다는 장점을 지니고 있으나 기계 셀의 수와 하나의 셀에 포함될 수 있는 기계의 수를 미리 정해진 값으로 제한하여 최적의 부품 군과 기계 셀을 형성할 수 없다는 단점을 지니고 있으며 유전자 알고리듬의 특성상 해를 도출하기 위해서는 긴 계산시간을 요구한다는 단점을 지닌다.

Kim *et al.*[8]은 대체 부품가공경로를 갖는 부품-기계 군집 문제를 위한 일반화된 군집 알고리듬을 통해 부품의 기계 셀간 이동과 기계 부하의 평준화를 동시에 고려할 수 있는 방법을 제시하였다. 그러나 이 연구에서 제시한 알고리듬은 대체가공경로는 고려하고 있으나 부품의 가공순서(operation sequence)를 고려하지 않고 단지 부품가공경로 상에 존재하는 기계 정보만을 고려하였기 때문에 부품 가공순서가 부품의 셀간 이동에 미치는 영향을 반영하지 못했으며, 각각의 부품이 특정 기계에서 가공될 때 소요되는 가공시간(processing time)을 고려하지 않았다는 문제점을 지닌다.

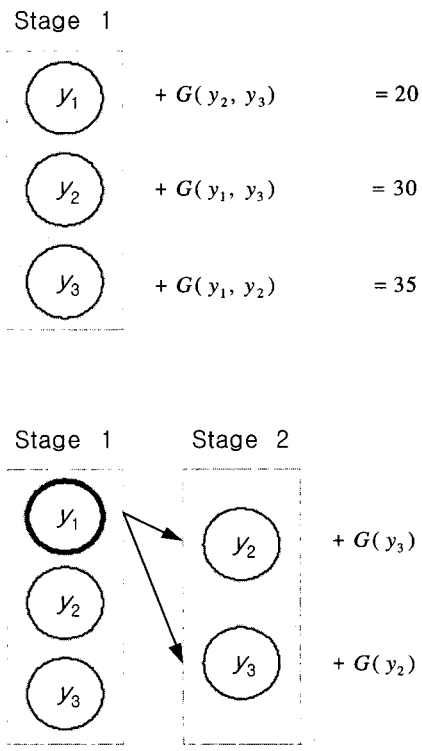
위에서 언급한 여러 가지 문제점들을 해결하기 위해서 Baek *et al.*[1]은 대체가공경로와 가공순서를 갖는 부품 정보를 알고 있을 때, 부품별 대체 부품가공경로와 가공순서를 고려하여 부품의 기계 셀간 이동을 최소화시키고 동시에 기계 부하를 평준화시킬 수 있는 방법을 제시하였다. 또한 Zhao and

Wu[17]의 연구와는 달리 기계 셀의 수와 셀에 포함될 수 있는 기계의 수를 미리 정해진 값으로 제한하지 않음으로써 최적의 부품 군과 기계 셀을 형성할 수 있도록 하였다.

그러나 Baek *et al.*[1]이 제시한 알고리듬은 부품별로 하나의 부품가공경로를 선정하여 목적 함수를 가장 많이 줄일 수 있는 기계 셀에 할당하는 탐욕적 탐색 알고리듬(greedy search algorithm) 방법을 사용하는데, 탐욕적 탐색 알고리듬은 한번에 한 단계씩 가장 좋은 방향으로 전진하는 근시안적 알고리듬(myopic algorithm)이기 때문에 부분 최적해(local optimal solution)에 빠질 위험이 있다.

일반적인 문제의 탐욕적 탐색 알고리듬에서는 두 개의 근접한 대안 중에서 하나의 대안이 선택되면 다른 대안은 다음 순서로 선택되는 경우가 대부분이다. 예를 들어, SPT(Shortest Processing Time first)나 EDD(Earliest Due Date first)와 같은 규칙이 적용되는 경우에 가장 좋은 값을 갖는 부품이 선택되고 나면 두 번째로 좋은 값을 갖는 부품이 다음으로 선택되거나 최소한 매우 높은 우선순위(priority)를 갖게 된다. 그러나 본 연구의 대상이 되는 부품-기계 군집 문제는 먼저 할당되어 있는 부품가공경로에 의해 다음에 선택될 부품가공경로가 결정되기 때문에 어떤 부품가공경로가 먼저 선택되느냐에 따라 전혀 다른 부품가공경로가 결정된다. 결국, 기계 셀에 먼저 할당된 부품가공경로가 다른 부품가공경로의 선택에 많은 영향을 미치기 때문에 일반적인 문제에서의 탐욕적 탐색 알고리듬보다 초기 할당이 매우 중요한 역할을 한다. 따라서 부품가공경로를 기계 셀에 할당하기 전에, 부품가공경로가 할당된 다음, 몇 단계 후의 목적 값을 미리 알아보는 것이 효과적이다.

본 연구에서는 위에서 언급한 문제점을 해결하기 위해 roll-out 알고리듬을 응용한 부품-기계 군집 알고리듬을 제시하고자 한다. roll-out 알고리듬이란 DP(Dynamic Programming) 해법을 기본으로 여러 가지 형태의 조합 최적화(combinatorial optimization) 문제에 대한 개략적인 해를 제공할 수 있는 알고리듬이다[2]. DP 해법은 최적해(optimal solution)를 찾을 수 있지만 해를 구하기 위해서는 많은 시간과 큰 메모리 공간이 소요되는 문제점을 지닌다. 특히 문제의 크기가 커질수록 소요되는 계산 시간이 기하급수적으로 커지기 때문에 복잡한 형태를 지니며 빠른 계산 시간을 요구하는 현실 문제에는 적용될 수 없다는 단점을 지닌다. 따라서 유효한 계산 시간 안에 적절한 해를 구하기 위해서 이런 DP 해법을 응용한 roll-out 알고리듬이 사용될 수 있다. roll-out 알고리듬은 DP 해법과는 달리 각 단계(stage)에서 경험적(heuristic) 알고리듬을 통해 가장 좋은 해를 제공할 가능성이 있는 하나의 상태(state)만을 선택한 후 다른 상태들은 버리고 다음 단계로 전진한다. 따라서 방대한 계산 시간을 요구함으로써 유효 시간 안에 해를 찾을 수 없는 DP 해법에 비해 roll-out 알고리듬은 유효한 계산 시간 안에 적절한 해를 구할 수 있다는 장점을 지닌다.

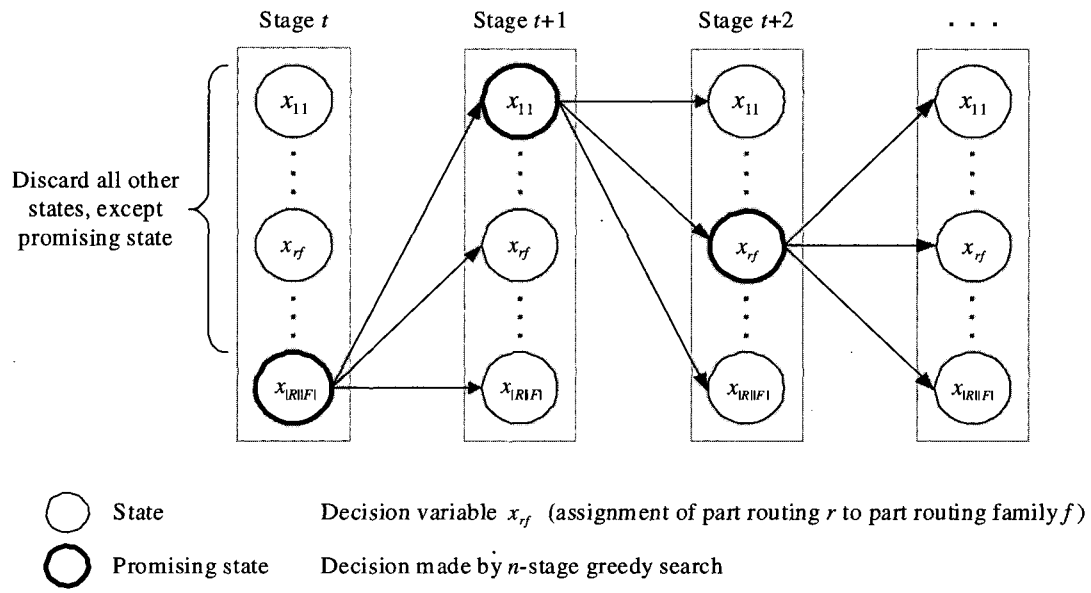


[Figure 1] roll-out algorithm

예를 들어 [Figure 1]과 같이, y_1, y_2, y_3라는 3개의 변수를 순서대로 결정해야 하는 문제에서 첫 번째 단계(stage)에서는 3개의 변수 y_1, y_2, y_3이 가능 대안이 될 수 있다. roll-out 알고리듬은 각각의 대안이 선택되어졌다고 가정하고 최종 해에 대한 목적함수의 값을 계산한다. 먼저 y_1이 결정되었다고 가정하고 남아있는 두개의 변수(y_2, y_3)를 대상으로 경험적 알고리듬을 이용하여 해를 구한다. 마찬가지 방법으로 y_2, y_3이 결정되었다고 가정한 후 목적함수의 값을 계산한다. 각각의 목적함수의 값이 20, 30, 35가 나왔다면 가장 좋은 목적함수 값을 가진 변수 y_1을 첫 번째 단계의 해로 결정한다. 같은 방법으로 y_1이 결정된 상태에서 두 번째 단계의 해를 결정하기 위해 같은 방법으로 진행한다. 즉, 첫 번째 단계에서 y_1이 결정된 상태에서 두 번째 단계에서의 해를 결정하기 위해 두 번째 단계에서 y_2, y_3이 결정되었다고 가정한 후에 각각의 최종 목적함수 값을 계산한 후 작은 목적함수 값을 만드는 해를 결정한다. 이와 같은 방법으로 각 단계에서의 모든 대안에 대해 최종 목적함수의 값을 예측한 후에 가장 좋은 대안을 결정하고 다음 단계로 진행한다.

그러나 roll-out 알고리듬도 DP 해법에 기초를 두고 있기 때문에 대단위 문제에 대해 알고리듬을 그대로 사용할 경우 많은 계산 시간이 요구되며, 합리적인 시간 안에 해를 구하기 힘들다는 문제점을 지닌다. 따라서 본 연구에서는 탐욕적 탐색 알고리듬의 근시안적인 문제를 해결하면서 roll-out 알고리듬의 시간적인 단점을 해결하기 위해 roll-out 알고리듬을 응용한 look-ahead 알고리듬을 제시하고자 한다.

look-ahead 알고리듬은 [Figure 2]과 같은 형태로 표현될 수 있는데, t 단계에서 고려해야 하는 변수가 k개 있는 경우 각각의 변수가 선택되어졌다는 가정 하에 경험적 알고리듬을 통해 $(t+n)$ 단계까지의 해를 계산한 후, $(t+n)$ 단계에서 가장 좋은 해를 갖게 하는 t 단계에서의 변수를 결정하고 나머지 변수들은 버린다. 따라서 t 단계에서의 변수를 결정하기 위해서는 n 단계 후의 해를 계산해야 하기 때문에 n-stage look-ahead 알고리듬으로 부르기도 한다.



[Figure 2] n-stage look-ahead algorithm

본 연구에서 제시하는 look-ahead 알고리듬은 한 단계를 전진할 때 경험적 알고리듬을 통해 몇 단계 후의 해를 미리 예측한 후 현 단계의 해를 확정하기 때문에 현 단계에서의 상태만을 고려하는 탐욕적 탐색 알고리듬의 근시안적인 단점을 해결할 뿐만 아니라 몇 단계 후의 해를 기준으로 하느냐에 따라서 계산 시간이 달라지기 때문에 roll-out 알고리듬의 시간적 단점도 해결할 수 있다. 특히 본 연구의 대상이 되는 부품-기계 군집 문제의 경우 초기 할당이 다음 단계의 할당에 막대한 영향을 미치기 때문에 몇 단계 후의 값을 미리 예측한 후 할당하는 look-ahead 알고리듬은 매우 효과적이다.

본 연구에서는 위에서 제시한 n-stage look-ahead 알고리듬을 이용하여 대체 부품가공경로와 가공 순서가 존재하는 부품-기계 군집 문제에서 기계 셀간 이동과 기계간 부하의 차이를 줄일 수 있는 해를 도출하는 방법을 제시하고자 한다.

2. 부품-기계 군집 문제

2.1 문제 정의

본 연구에서는 실제 제조현장의 상황을 표현하기 위해 대체 부품가공경로와 가공 순서를 갖는 부품 정보를 알고 있을 때 효율적으로 부품 군과 기계 셀을 형성하기 위한 부품-기계 군집 알고리듬을 제시하고자 한다. 지금까지의 일반적인 부품-기계 군집 문제에서는 가공순서가 고려되지 않거나 대체 부품가공경로가 존재하지 않는 문제가 대부분이였다. 그러나 실제 제조현장에서는 각각의 부품에 대해 대체 부품가공경로가 존재하고 각 기계에서의 가공 순서 및 가공 시간이 존재한다. 따라서 본 연구에서는 각 부품의 가공경로 정보가 <Table 1>과 같이 주어진다고 가정한다.

<Table 1>에서 각 부품은 연간 소요량을 가지고 있으며 두개의 대체 부품가공경로를 지니고 있다. 각각의 대체 부품가공경로는 가공순서를 지니며 표시된 숫자는 각 기계에서의 가공 순서를 나타낸다. 특정 기계에서 '0'으로 표시된 부분은 가공경로에 해당 기계가 포함되지 않는다는 것을 의미하며 괄호 안의 숫자는 각 기계에서 소요되는 가공 시간을 나타낸다[1].

예를 들어, 부품 1의 가공경로 2는 기계 1과 기계 2에서 순차적으로 가공되며, 이 때 소요되는 가공시간은 각각 3과 2가 된다. 또한 본 연구에서는 각 부품에 대한 연간 소요량이 미리 주어져 있으며, 각각의 부품은 최종적으로 선택된 하나의 가공경로에 의해 가공되어진다고 가정한다. 그리고 각각의 기계는 해당 기계별 능력 한계(capacity limit)를 지니고 있으며 부품-기계 군집 문제는 이러한 기계의 능력 한계를 벗어나지 않는 범위에서 해답을 제시할 수 있어야 한다.

<Table 1> A machine-cell problem example.

Part (Annual demand)	Part route	Machine (Processing time)			
		1	2	3	4
1 (100)	1	1 (2)	3 (3)	2 (1)	0 (0)
	2	1 (3)	2 (2)	0 (0)	0 (0)
2 (80)	3	2 (3)	1 (1)	0 (0)	0 (0)
	4	0 (0)	0 (0)	1 (2)	2 (4)
3 (50)	5	0 (0)	2 (1)	3 (2)	1 (4)
	6	1 (3)	2 (2)	3 (1)	0 (0)
4 (120)	7	0 (0)	0 (0)	1 (2)	2 (2)
	8	0 (0)	3 (1)	1 (2)	2 (1)
Capacity limit		450	450	450	450

2.2 기호 정의

R : 부품가공경로 집합
r : 부품가공경로 번호 ($r=1,2,\cdots,|R|$)
P : 부품 집합
p : 부품 번호 ($p=1,2,\cdots,|P|$)
M : 기계 집합
m : 기계 번호 ($m=1,2,\cdots,|M|$)
F : 부품 군 집합
f : 부품 군 번호 ($f=1,2,\cdots,|F|$)

r_f : 부품 군 f의 대표 부품가공경로 번호

R_f : 부품 군 f에 속한 부품들의 가공경로 집합($r_f \in R_f$)

R_p : 부품 p의 대체 부품가공경로 집합 ($\bigcup_{p \in P} R_p = R$)

d_r : 부품가공경로 r에 대한 연간 소요량 ($d_r = d_{r'}$ if $r, r' \in R_p$ for some p)

$\Psi(r)$: 부품가공경로 r이 거치는 기계의 수

t_{rm} : 부품가공경로 r에서 기계 m의 가공시간

L_m : 기계 m에 할당되는 부하

CL_m : 기계 m의 능력 한계(capacity limit)

2.3 수리적 모델

본 연구의 대상이 되는 대체 부품가공경로와 가공 순서를 고려한 부품-기계 군집 문제는 다음과 같은 수리적 모델을 통해 표현될 수 있다.

$$Min \quad J = \alpha \sum_{r \in R} \sum_{f \in F} d(r, r_f)\, x_{rf} + \beta \max_{m_1, m_2 \in M} \frac{|L_{m_1} - L_{m_2}|}{CL_{\max}} \qquad (1)$$

식 (1)은 부품의 셀간 이동과 기계간 부하의 차이를 최소화하기 위해 사용된 목적함수이다. 식 (1)에서 x_{rf}는 부품가공경로 r이 부품 군 f에 포함되는지를 나타내기 위해 0과 1의 값을 가지는 결정변수이다. 또한 부품 군 f의 대표 부품가공경로 r_f와 부품군의 수 $|F|$도 결정 변수이다. 따라서 목적함수의 첫 번째 식은 대표 부품가공경로(r_f)와 해당 대표 부품가공경로로 구성된 부품 군에 할당되는 부품가공경로(r)와의 차이를 최소화함으로써 유사성이 높은 부품들을 하나의 군으로 형성하여 부품의 기계 셀간 이동을 최소화하기 위해 사용되었고, 목적함수의 두 번째 식은 기계간 부하의 차이를 최소화하기 위해 사용되었다.

식 (1)에서 $d(r, r_f)$는 두 개의 부품가공경로 r과 r_f가 얼마나 유사하지 않는 경로를 지녔는지를 나타내는 척도(dissimilarity measure)로서 다음과 같이 정의될 수 있다.

$$d(r_1, r_2) = 1 - \frac{N(r_1, r_2)}{N(r_1) + N(r_2) - N(r_1, r_2)} \qquad (2)$$

식 (2)에서 $N(r_i)$, $(i = 1, 2)$는 부품가공경로 r_i에서 만들 수 있는 기계 부분쌍(machine subsequence)의 수를 나타내며 $N(r_1, r_2)$는 부품가공경로 r_1, r_2의 기계 부분쌍에 대한 공통 기계 부분쌍의 수를 나타낸다. 따라서 식 (2)는 두 개의 부품가공경로 사이에 공통 기계 부분쌍이 적을수록 유사성이 더 낮다는 것을 의미하며, 만약 두 개의 부품가공경로가 공통 기계 부분쌍을 전혀 갖지 않는 경우에는 $N(r_1, r_2) = 0$이 되고 결국 $d(r_1, r_2) = 1$이 된다. 반대로 같은 부품가공경로를 갖는 경우에는 $d(r_1, r_2) = 0$이 된다. 따라서 $d(r_1, r_2)$는 0에서 1사이의 값을 갖으며 목적함수를 최소화하는 것은 유사성이 높은 부품을 같은 부품 군으로 선택한다는 것을 의미한다.

예를 들어, 세 개의 부품가공경로 r_1, r_2, r_3가 각각 (1, 3, 2, 5, 4), (1, 3, 2, 4, 5), (1, 2, 4, 5)의 기계 가공순서를 지녔다고 가정하면, r_1는 {(1, 3), (3, 2), (2, 5), (5, 4)}의 기계 부분쌍을 지녔고 $N(r_1) = 4$이다. 같은 방법으로 부품가공경로 r_2는 {(1, 3), (3, 2), (2, 4), (4, 5)}의 기계 부분쌍을 지녔고 $N(r_2) = 4$이며, 부품가공경로 r_3는 {(1, 2), (2, 4), (4, 5)}의 기계 부품쌍을 지녔고 $N(r_3) = 3$이 된다. 또한 $N(r_1, r_2) = 2$, $N(r_1, r_3) = 0$ 그리고 $N(r_2, r_3) = 2$이 되고, 이 값에 따라 $d(r_1, r_2) = 2/3$, $d(r_1, r_3) = 1$, $d(r_2, r_3) = 3/5$과 같이 계산될 수 있다.

식 (1)에서 L_m은 기계 m에 할당되는 부하를 나타내는 값으로써 다음과 같이 정의된다.

$$L_m = \sum_{f \in F} \sum_{\{r \in R \mid x_{rf} = 1\}} d_r \cdot t_{rm} \qquad (3)$$

식 (1)에서 CL_{max}는 모든 기계의 부하 능력 중에서 최대 값을 갖는 기계의 부하 능력을 의미하며 정규화(normalization)를 위해 사용되었다. 또한 α와 β는 목적 함수의 두 수식 중에서 사용자가 더 중요하게 생각하는 부분을 나타내기 위해 사용되는 중요도(weight)를 의미한다.

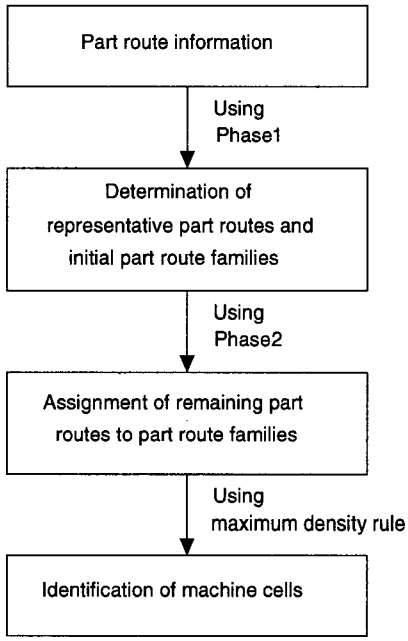
본 연구에서 제시하는 부품-기계 군집 문제를 풀기 위해서는 목적 함수에는 다음과 같은 제약조건이 고려되어야 한다.

(1) 제약조건 1 : 하나의 부품에 대해 하나의 부품가공경로만이 선택될 수 있으며 선택된 부품가공경로는 하나의 부품 군에만 할당될 수 있다.
(2) 제약조건 2 : 결정된 부품 군의 수는 부품의 수보다 더 클 수 없다.
(3) 제약조건 3 : 각 기계에 할당되는 부하는 해당 기계의 최대 부하 능력보다 클 수 없다.

3. 부품-기계 군집 알고리듬

본 연구에서는 대단위 부품-기계 군집 문제를 해결하기 위해 [Figure 3]와 같은 2단계 경험적(Heuristic) 해법을 제시하고자 한다. 단계 1에서는 부품가공경로의 유사성을 고려하여 대표 부품가공경로와 초기 부품 군을 결정한다. 부품가공경로의 유사성을 측정하기 위해서 2.3절에서 제시한 식 (2)를 척도로 사용한다.

단계 2에서는 roll-out 알고리듬을 응용한 n-stage look-ahead 알고리듬을 이용하여 나머지 부품에 대해 목적함수를 고려하여 부품 군에 할당한다. 할당되지 않은 부품 경로 중의 하나를 부품 군에 할당하기 위해 어떤 부품가공경로를 하나의 부품 군에 할당했을 경우에 대한 결과를 경험적 알고리듬을 이용하여 추정하고 가장 좋은 결과를 만들 수 있도록 부품가공경로를 부품 군에 할당한다. n-stage look-ahead 알고리듬에서 결과를 추정하기 위한 비용 함수로 Baek *et al.*[1]이 제시한 경험적 알고리듬을 이용한다. 따라서 제안한 알고리듬을 이용하여 가장 좋은 결과를 산출할 것으로 예상되는 부품가공경로를 부품 군에 할당해 나간다.



[Figure 3] two-phase heuristic algorithm

3.1 단계 1 알고리듬(phase 1 algorithm)

단계 1에서는 부품가공경로의 유사성을 고려하여 부품 군의 수($|F|$)와 각 부품 군에 대한 대표 부품가공경로를 결정한다. 이를 위해 본 연구에서는 다음과 같은 각각의 부품가공경로별 포텐셜(potential) 함수를 정의한다.

$$N_r = |\eta_\theta(r)|, \quad \forall r \in R$$

단, $\eta_\theta(r) = \{r' \mid d(r, r') \le \theta,\ r' \neq r\}$로 정의되며 이는 부품가공경로 r과의 거리가 θ보다 작은 부품가공경로들의 집합을 의미한다. 따라서 부품가공경로 r과 $r' \in \eta_\theta(r)$사이의 포텐셜 변화(difference)는 $\nabla(r, r') = N_{r'} - N_r$로 정의될 수 있고, $\nabla(r)_{max} = max_{\{r' \in \eta_\theta(r)\}} \nabla(r, r')$는 부품가공경로 r과 $r' \in \eta_\theta(r)$에 대한 포텐셜 변화의 최대값을 의미한

다. 따라서 만약 $N_r=0$인 경우, $\nabla(r)_{max}=0$이 되고 이 때의 부품가공경로 r은 특이점(outlier)이 된다. 또한 $N_r\neq0$이고 $\nabla(r)_{max}\leq0$인 경우, 이 때의 부품가공경로 r은 모드(mode)가 되고 하나의 부품군을 대표하는 가공경로의 후보가 될 수 있다.

이와 같은 알고리듬은 θ 값에 따라 부품 군의 수가 달라질 수 있으며, θ 값을 매우 적게 지정($\theta\approx0$)하면 부품 군의 수는 부품 수와 같아지고 θ 값을 매우 크게 지정($\theta\approx1$)하면 오직 한 개의 부품 군만 형성된다. 따라서 사용자는 θ 값을 적절히 조절해 가면서 최적의 부품 군을 형성해야 하는데, 본 연구에서는 이를 위해 θ 값을 0에서부터 시작하여 0.05씩 증가시키면서 최적의 부품 군을 형성할 수 있도록 하였다.

본 연구에서 제안하는 대표 부품가공경로 결정 알고리듬은 다음과 같다.

Step 1.1 $F=\emptyset$, $f=0$
Step 1.2
IF(there exist $p\in P$ such that $\eta_\theta(r)=\emptyset$ for all $r\in R_p$)
THEN
$f=f+1$, $F\leftarrow F\cup\{f\}$,
$r^*=\underset{r\in R_p}{argmin}\ \Psi(r)$,
$R_f=\{r^*\}$, $R\leftarrow R-\{r^*\}$,
$R\leftarrow R-\{r\mid r^*\in R_p \text{ and } r\in R_p\}$,
$L_m=d_{r^*}\cdot t_{r^*m}$ *for all* $m\in M$, $x_{r^*f}=1$,
and go to Step 1.4.
Step 1.3
IF(there exist r^* such that $\eta_\theta(r^*)\neq\emptyset$ and $\nabla(r^*)_{max}\leq0$)
THEN
$f=f+1$, $F\leftarrow F\cup\{f\}$,
$R_f=\{r^*\}$, $x_{r^*f}=1$,
$R\leftarrow R-\{r^*\}$,
$R\leftarrow R-\{r\mid r^*\in R_p \text{ and } r\in R_p\}$,
$R\leftarrow R-\{R_{p'}\mid r'\in\eta_\theta(r^*) \text{ and } r'\in R_{p'}$ *for some* $p'\}$,
$L_m=d_{r^*}\cdot t_{r^*m}$ *for all* $m\in M$, $x_{r^*f}=1$.
Step 1.4 IF($R=\emptyset$) **THEN** stop
ELSE go to Step 1.2

위의 알고리듬 Step 1.2에서 임의의 부품 p가 특이점(outlier)이면 부품 p는 홀로 독립적인 부품 군을 형성하며, 따라서 다른 부품 군과의 상관관계를 줄이기 위해 대체가공경로 중에서 가장 적은 기계를 거치는 부품가공경로를 대표 부품가공경로로 선택한다. 2.3절에서 제시한 제약조건 1과 제약조건 2를 만족시키기 위해 $R_f=\{r^*\}$식과 $R\leftarrow R-\{r\mid r^*\in R_p \text{ and } r\in R_p\}$식에 의해 특이점이 되는 부품($p$)의 모든 부품가공경로가 부품가공경로 집합($R$)에서 제거되어 더 이상 고려되지 않도록 한다. 그리고 선택된 부품(p)에 의한 기계 부하량이 계산되어진다.

Step 1.3은 포텐셜 함수에 의해 모드(mode)가 되는 대표 부품가공경로를 찾는 과정으로서, 하나의 부품가공경로가 대표 부품가공경로로 선정되면 다음과 같은 세 가지 연산(operation)이 수행되어진다. 처음 두 가지 연산인 $R\leftarrow R-\{r^*\}$, $R\leftarrow R-\{r\mid r^*\in R_p \text{ and } r\in R_p\}$는 제약조건 1과 제약조건 2를 만족시키기 위해 수행하는 작업이다. 또한 세 번째 연산인 $R\leftarrow R-\{R_{p'}\mid r'\in\eta_\theta(r^*) \text{ and } r'\in R_{p'} \text{ for some } p'\}$는 대표 부품가공경로($r^*$)와 거리상 θ경계 내에 존재하는 인접한 모든 부품가공경로와 해당 부품가공경로가 속한 부품의 다른 모든 가공경로가 부품가공경로 집합(R)에서 제거되어 추후 다른 대표 가공경로를 결정할 때 고려되지 않아야 한다는 것을 나타낸다. Step 1.4는 알고리듬의 종료 조건으로 모든 부품가공경로가 고려되었을 때 알고리듬이 종료된다는 것을 의미한다.

3.2 단계 2 알고리듬(phase 2 algorithm)

단계 2는 n-stage look-ahead 알고리듬을 이용하여 나머지 부품에 대해 목적함수를 고려하여 단계 1에서 결정된 부품 군에 할당하는 절차를 나타낸다. 본 연구에서 제안하는 부품 군과 기계 셀 형성 알고리듬은 다음과 같다. 단, 알고리듬에서 $\Delta_t J(x_{rf})$는 단계 t에서 $x_{rf}=0$을 $x_{rf}=1$로 변환

했을 때 목적함수 값의 차이를 의미한다. 또한 $\mathcal{T}$는 탐욕적 탐색 알고리듬(greedy search algorithm)을 이용한 추정된 목적함수 값을 나타내며, $\Delta_{t+1,t+n}\mathcal{T}$은 주어진 알고리듬에 의해 계산된 단계 $t+1$에서 $t+n$까지의 목적함수 값의 차이를 의미한다.

Step 2.1 $t=0$,
$P(t) \leftarrow P-\{p \mid r \in R_p \text{ and } r \in R_f \text{ of any } f \in F\}$,
$R_p(t) = R_p \text{ for all } p \in P(t)$,
$L_m(t) = L_m \text{ for all } m \in M$.

Step 2.2
$(r^*, f^*, p^*) =$
$\underset{p \in P(t),\ r \in R_p(t),\ f \in F}{argmin} \{\Delta_t J(x_{rf}) + \Delta_{t+1,t+n}\mathcal{T}\}$.

Step 2.3
IF($L_m(t) + d_{r^*} \cdot t_{r^*m} \le CL_{\max}$, *for all* $m \in M$)
THEN go to Step 2.4
ELSE $R_{p^*}(t) = R_{p^*}(t) - \{r^*\}$ and
go to Step 2.2

Step 2.4 $t=t+1$,
$R_{f^*} \leftarrow R_{f^*} \cup \{r^*\}$, $P(t) \leftarrow P(t-1) - \{p^*\}$,
$X(t) = X(t-1) + I_{r^*f^*}$,
$L_m(t) = L_m(t-1) + d_{r^*} \cdot t_{r^*m}, \forall m \in M$.
IF($P(t) = \varnothing$) **THEN** stop
ELSE go to Step 2.2

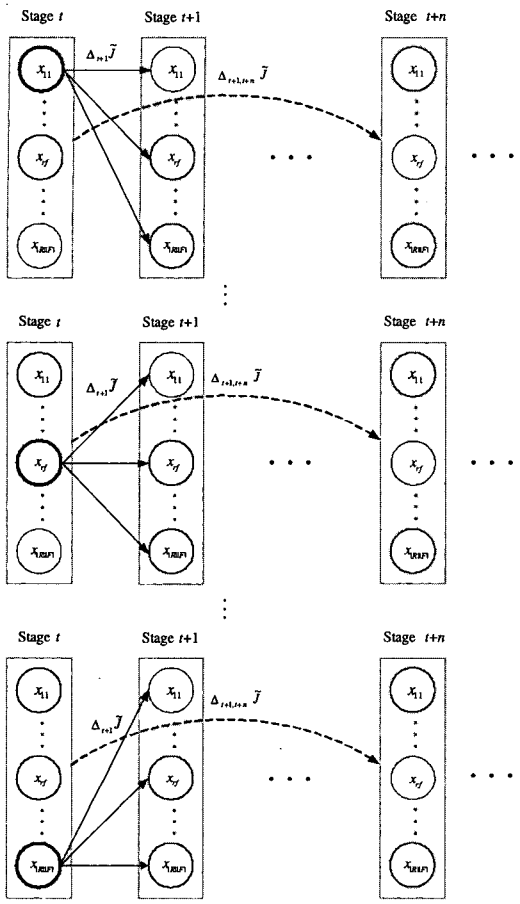
위의 알고리듬에서 Step 2.1은 단계 1 알고리듬에 의해 선정된 대표 부품가공경로를 갖는 부품들을 부품 군 형성 대상에서 제외시키는 절차를 나타낸 것이다. 즉, 단계 1에 의해 선정된 대표 부품가공경로를 갖는 부품들은 이미 부품 군에 할당되었기 때문에 더 이상 부품 군 할당을 위해 고려될 필요가 없고, 이에 따라 해당 부품의 모든 대체 부품가공경로가 가공경로 집합(R)에서 제거된다.

Step 2.2는 단계 2 알고리듬의 핵심부분으로서, n-stage look-ahead 알고리듬에 의해 부품 군(f^*)에 할당될 부품(p^*)과 해당 부품의 부품가공경로(r^*)를 결정한다. 즉, 단계 t에서 $\Delta_t J(x_{rf}) + \Delta_{t+1,t+n}\mathcal{T}$를 가장 작게 만드는 부품 군, 부품, 부품가공경로를 결정한다.

여기서 $\Delta_{t+1,t+n}\mathcal{T}$은, 탐욕적 탐색 알고리듬에 의해 계산된, 단계 $t+1$에서 $t+n$까지의 목적함수 값의 차이를 의미한다. 즉, 단계 $t+1$에서부터 단계 $t+n$까지 탐욕적 탐색 알고리듬에 의해 각 단계에서 목적함수 값의 차이($\Delta_{t+i}\mathcal{T}$ ($i=1,2,\cdots,n$))를 가장 작게 만드는 해를 선택하여 계산된 값으로 $\Delta_{t+1,t+n}\mathcal{T}$은 각 단계에서의 목적함수 값의 차이인 $\Delta_{t+i}\mathcal{T}$ ($i=1,2,\cdots,n$)의 최소값의 합이 된다.

만약 [Figure 4]에서처럼 단계 t에서의 부품 군과 부품, 부품가공경로를 결정하기 위해 단계 t에서 선택할 수 있는 대안들이 있을 경우, 각 대안을 단계 t에서 선택하였다고 가정하고 탐욕적 탐색 알고리듬에 의해 단계 $t+1$에서부터 단계 $t+n$까지 $\Delta_{t+i}\mathcal{T}$ ($i=1,2,\cdots,n$)가 가장 작은 값을 찾는다. 따라서 단계 t에서 선택할 수 있는 모든 대안에 대해 $\Delta_t J(x_{rf}) + \Delta_{t+1,t+n}\mathcal{T}$의 값을 계산한 후에 가장 작은 값을 갖는 대안을 선택하게 된다. 이때 빠른 시간 안에 $\Delta_{t+1,t+n}\mathcal{T}$을 계산하기 위해 탐욕적 탐색 알고리듬이 사용된다. 따라서 탐욕적 탐색 알고리듬의 성능도 중요한 요소가 되며 본 연구의 n-stage look-ahead 알고리듬에서 각 단계별 탐욕적 탐색 알고리듬은 Baek *et al.*[1]이 제안한 알고리듬을 사용한다.

Step 2.3은 Step 2.2에서 선택된 부품가공경로에 의해 발생하는 각 기계에서의 부하가 해당 기계의 능력한계(CL_m)를 벗어나지 않는지 검사하는 단계로써, 조건을 만족시키는 경우 선택된 부품가공경로를 Step 2.4를 통해 부품 군에 할당하고 만족시키지 못한 경우 선택된 부품가공경로를 부품 군 할당 대상에서 제외하고 Step 2.2로 되돌아간다. 단계 2 알고리듬은 모든 부품이 부품 군에 할당되었을 때 종료된다.



[Figure 4] *n*-stage look-ahead algorithm example

단계 1 알고리듬과 단계 2 알고리듬에 의해 유사한 특성을 갖는 부품들에 대한 부품 군이 결정되면, 해당 부품 군에 대한 효율적인 가공을 수행하기 위해 기계 셀을 형성해야 한다. 본 연구에서는 기계 셀을 형성하기 위해 최대밀도규칙(maximum density rule)[15]을 사용한다. 최대밀도규칙은 각각의 기계에 대한 작업이 가장 많이 수행되는 부품 군에 기계를 할당하는 방법으로 형성된 부품 군에 속한 부품들의 가공경로에서 0이 아닌 값이 가장 많은 기계를 해당 부품 군의 기계 셀에 할당한다.

4. 실험 결과

본 연구에서 제시한 알고리듬의 성능을 평가하기 위해 <Table 3>에서 제시한 바와 같은 4가지 유형의 문제에 대한 실험을 통해 Kim *et al.*[8]과 Baek *et al.*[1]이 제시한 알고리듬과의 비교 평가를 수행하였다. Baek *et al.*[1]이 제시한 알고리듬은 본 연구와 일치하는 실험 조건을 가지고 있기 때문에 직접적인 비교 대상이 된다. Kim *et al.*[8]이 제시한 알고리듬은 가공 순서와 가공 시간을 고려하지는 못하지만 대체 가공 경로를 고려하고 부품 군과 기계 셀의 수를 제한하지 않는 등의 본 연구와 유사한 조건을 갖고 있을 뿐만 아니라 비교적 좋은 해를 계산하기 때문에 비교 대상인 두 알고리듬의 기준점으로 두고 비교 평가하였다.

〈Table 3〉 part-machine grouping problem types

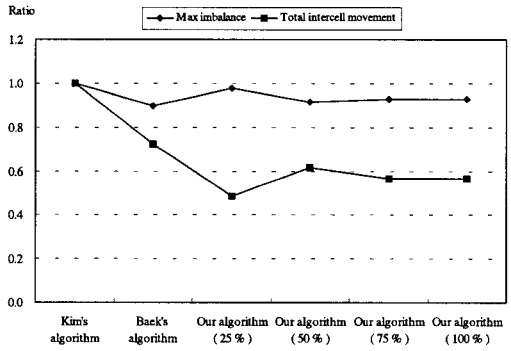
	Number of parts	Number of part routes	Number of machines
Problem type 1	10	25	10
Problem type 2	20	50	20
Problem type 3	30	75	30
Problem type 4	40	100	40

본 실험에서는 4가지 문제 유형에 대해 난수(random number) 발생을 이용하여 20개의 문제를 생성하고, 이들 문제에 대해 본 연구에서 제시한 알고리듬과 Kim *et al.*[8]과 Baek *et al.*[1]이 제시한 알고리듬을 이용하여 생성된 부품-기계 군집 결과를 비교하였다.

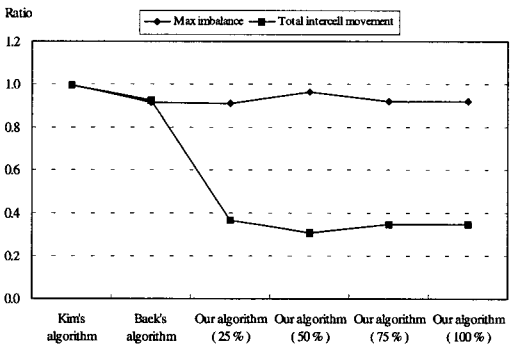
주어진 문제를 풀기 위해 본 알고리듬을 위한 파라미터(parameter)를 다음과 같이 설정하였다.

(1) 본 연구에서 제안하는 n-stage look-ahead 알고리듬을 적용하기 위해 n의 크기는 부품 수의 25%, 50%, 75%, 100%로 설정하였다.
(2) 본 연구에서 제안하는 알고리듬과 Kim *et al.*[8]과 Baek *et al.*[1]이 제안한 알고리듬을 비교하기 위해 각 알고리듬의 α, β는 $\alpha = \beta = 0.5$

로 설정하였다. 또한 θ에 대한 설정은, Kim *et al.*[8]이 제안한 알고리듬에서는 θ를 $\{0, 1, \sqrt{2}, \sqrt{3}, \cdots, \sqrt{|M|}\}$으로 설정하였고, Baek *et al.*[1]이 제안한 알고리듬과 본 연구에서 제안한 알고리듬에서는 θ를 0에서 1까지 0.05씩 증가시킨 값으로 설정하였다.

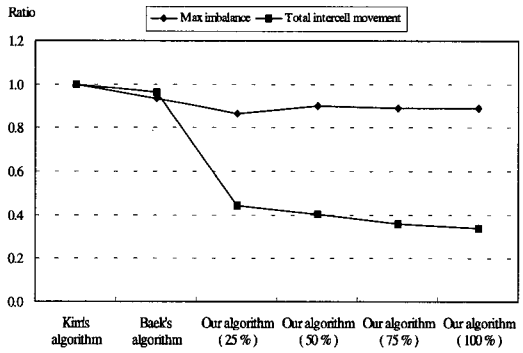


[Figure 5] Comparison result (problem type 1)

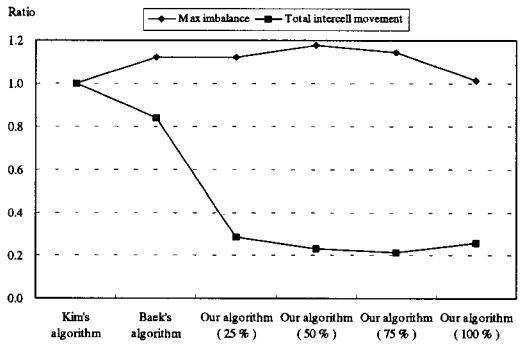


[Figure 6] Comparison result (problem type 2)

[Figure 5]~[Figure 8]은 각각의 문제에 대한 실험 결과를 나타낸 그래프로써 Y 축은 Kim *et al.*[8]이 제안한 알고리듬의 결과와 나머지 알고리듬에 의한 결과를 비율로 나타낸 것이다. 실험 결과에서 본 연구에서 제안한 알고리듬이 Kim *et al.*[8]이 제안한 알고리듬과 Baek *et al.*[1]이 제안한 알고리듬에 비해 효과적인 부품-기계 군집 결과를 도출함을 알 수 있다. 특히. 셀간 이동에 대한 개선에 매우 효과적임을 알 수 있는데, 이는 기계간 부하의 차이는 어떤 대체 부품가공경로를 선택하느냐에 따라 결정되며 기계 셀의 구성과는 무관하기 때문이다. 즉, 기계간 부하 차이는 사용되는 알고리듬에 의해 개선될 수 있는 한계가 존재하는 반면, 부품의 셀간 이동은 어떤 알고리듬에 의해 기계 셀을 어떻게 구성하느냐에 따라 크게 향상될 수 있기 때문이다.

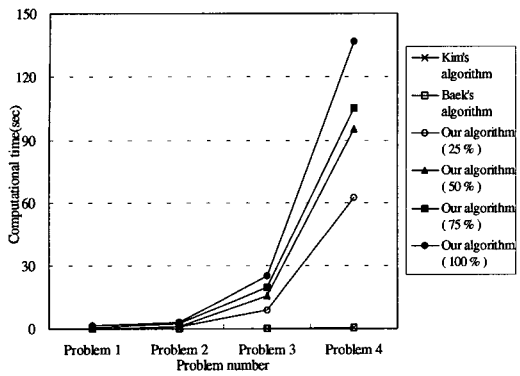


[Figure 7] Comparison result (problem type 3)



[Figure 8] Comparison result (problem type 4)

[Figure 5]와 [Figure 8]을 보면 본 연구에서 제안한 n-stage look-ahead 알고리듬에서 look-ahead의 비율이 높다고 해서 반드시 더 좋은 해를 보장하진 않는다는 것을 알 수 있는데, 이는 n-stage look-ahead 알고리듬이 DP와는 달리 최종 단계까지 진행한 후 해를 결정하지 않고 n 단계까지 진행한 후 해를 결정하기 때문에 최적해가 아닌 해를 도출할 가능성이 있기 때문이다.



[Figure 9] Comparison of computational time

[Figure 9]는 Kim *et al.*[8]이 제시한 알고리듬, Baek *et al.*[1]이 제시한 알고리듬, 그리고 본 연구에서 제시한 알고리듬을 이용하여 각각의 유형의 문제에 대한 해를 도출하기 위해 소요된 계산 시간(Computational time)을 나타낸 그래프이다. 본 연구에서는 Pentium Ⅳ 컴퓨터에서 C++로 작성된 프로그램을 이용하여 실험하였는데, 예상한 대로 look-ahead의 비율이 높아질수록 계산 시간이 길어짐을 알 수 있다.

〈Table 4〉 Variation of inter-cell part movements.

	Problem type 1	Problem type 2	Problem type 3	Problem type 4
$\alpha=1$, $\beta=0$,	1.00	1.00	1.00	1.00
$\alpha=0.75$, $\beta=0.25$	1.20	1.59	1.88	1.82
$\alpha=0.5$, $\beta=0.5$	1.26	2.30	2.19	2.05
$\alpha=0.25$, $\beta=0.75$	1.63	2.51	2.35	2.60
$\alpha=0$, $\beta=1$	2.75	4.00	3.89	4.06

마지막으로 본 연구에서 제안한 알고리듬에서 α와 β가 결과에 미치는 영향을 분석하기 위해 α와 β의 값을 변화시켜가면서 수행한 실험결과를 비교하였다. <Table 4>와 <Table 5>는 α와 β 값의 변화에 따른 셀간 이동과 기계간 부하의 차이를 비교한 결과로서 α = 1, β = 0일 때의 값에 대한 비율로 결과를 표현하였다.

〈Table 5〉 Variation of maximum workload imbalance

	Problem type 1	Problem type 2	Problem type 3	Problem type 4
$\alpha=1$, $\beta=0$,	1.00	1.00	1.00	1.00
$\alpha=0.75$, $\beta=0.25$	0.84	0.82	0.75	0.78
$\alpha=0.5$, $\beta=0.5$	0.66	0.63	0.59	0.60
$\alpha=0.25$, $\beta=0.75$	0.62	0.61	0.58	0.57
$\alpha=0$, $\beta=1$	0.56	0.59	0.53	0.51

<Table 4>와 <Table 5>에서 보듯이 β의 값이 커짐에 따라 기계간 부하의 차이는 작아지고, 부품의 셀간 이동은 커짐을 알 수 있다. 또한, 기계간 부하의 차이가 작아지는 비율보다 부품의 셀간 이동이 커지는 비율이 높다는 것을 알 수 있다. 이는 앞에서 언급한 바와 같이 기계간 부하의 차이보다 부품의 셀간 이동이 사용되는 알고리듬에 더 큰 영향을 받기 때문이다.

5. 결 론

본 연구에서는 가공순서와 대체 부품가공경로를 갖는 부품 정보를 알고 있을 때, 대단위 부품-기계 군집 문제를 해결하기 위한 look-ahead 휴리스틱 알고리듬을 제시하였다.

본 연구에서 제시한 n-stage look-ahead 알고리듬은 부품의 셀간 이동과 기계간 부하의 차이를 동시에 고려할 수 있고, 기존의 Kim *et al.*[8]과 Baek *et al.*[1]이 제시한 알고리듬보다 효율적인 결과를 도출할 수 있다. 또한 DP와는 달리 look-ahead의 크기를 조절할 수 있기 때문에 대단위 부품-기계

군집 문제에 대해 유효한 계산 시간 안에 효율적인 해를 도출할 수 있다.

추후 연구 과제로는 하나의 부품에 대해 하나 이상의 부품가공경로가 선택될 수 있는 경우에 대한 효율적인 부품-기계 군집 알고리듬의 개발이 필요하며, 기계간 부하의 차이에 대해 좀더 효과적인 알고리듬의 개발이 필요하다.

참 고 문 헌

[1] Baek, J.G., J.K. Baek, and C.O. Kim, "A Part-Machine Grouping Algorithm Considering Alternative Part Routings and Operation Sequences," *Journal of the Korean Institute of Industrial Engineers*, Vol.29, No.3(2003), pp.213-221.

[2] Bertsekas, D.P., J.N. Tsitsiklis, and C. Wu, "Rollout algorithm for combinatorial optimization," *Journal of Heuristics*, Vol.3(1997), pp.245-262.

[3] Burbidge, J.L., *Production Flow Analysis for Planning Group Technology*, Clarendon Press, 1997.

[4] Caux, C., R. Bruniaux, and H. Pierreval, "Cell Formation with Alternative Process Plan and Machine Capacity Constraints : A New Combined Approach," *International Journal of Production Economics*, Vol.64, No.1-3(2000), pp.279-284.

[5] Chu, C.H., "Recent Advances in Mathematical Programming for Cell Formation," *Planning, Design, and Analysis of Cellular Manufacturing Systems*, Edited by Kamrani, A.K., Parsaei, H.R., and Liles, D.H., Elsevier, (1995), pp.1-45.

[6] Harhalakis, G., R. Nagi, and J.M. Proth, "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications," *International Journal of Production Research*, Vol.28, No.1(1990), pp.185-198.

[7] Hwang, H., P. Ree, "Routes Selection for the Cell Formation Problem with Alternative Part Process Plans," *Computers and Industrial Engineering*, Vol.30, No.3(1996), pp. 423-431.

[8] Kim, C.O., Y. Park, and J. Jun, "Generalized Clustering Algorithm for Part-Machine Grouping with Alternative Process Plans," *Journal of the Korean Institute of Industrial Engineers*, Vol.27, No.3(2001), pp.281-288.

[9] Kusiak, A., *Intelligent Manufacturing Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[10] Kusiak, A., A. Vannelli, and K.R. Kumar, "Clustering Analysis : Models and Algorithms," *Control and Cybernetics*, Vol.15, No.2(1986), pp.139-154.

[11] Lee, H. and A. Garcia-Diaz, "Network Flow Procedures for the Analysis of Cellular Manufacturing Systems," *IIE Transactions*, Vol.28, No.4(1996), 333-345.

[12] Selim, H.M., R.G. Askin, and A.J. Vakharia, "Cell Formation in Group Technology : Review, Evaluation and Directions for Future Research," *Computers and Industrial Engineering*, Vol.34, No.1(1998), pp.3-20.

[13] Venugopal, V. and T.T. Narendran, "Machine-cell Formation through Neural Network Models," *International Journal of Production Research*, Vol.32, No.9(1994), pp.2105-2115.

[14] Wicks, E.M. and R.J. Reasor, "Designing Cellular Manufacturing Systems with Dynamic Part Populations," *IIE Transactions*, Vol.31, No.1(1999), pp.11-20.

[15] Won, Y., "New p-median Approach to

Cell Formation with Alternative Process Plans," *International Journal of Production Research*, Vol.38, No.1(2000), pp.229-240.

[16] Yin, Y. and K. Yasuda, "Manufacturing Cell's Design in Consideration of Various Production Factors," *International Journal of Production Research*, Vol.40, No.4(2002), pp.885-906.

[17] Zhao, C. and Z. Wu, "A Genetic Algorithm for Manufacturing Cell Formation with Multiple Routes and Multiple Objectives," *International Journal of Production Research*, Vol.38, No.2(2000), pp.385-395.