

유한용량 Flexible Manufacturing Systems(FMS) 스케줄링 문제에 대한 효율적인 탐색 알고리즘 연구

김황호 · 최진영[†]

아주대학교 산업공학과

An Efficient Search Algorithm for Flexible Manufacturing Systems (FMS) Scheduling Problem with Finite Capacity

Hwang Ho Kim · Jin Young Choi

Department of Industrial Engineering, Ajou University

In this paper, we propose an efficient search algorithm for finding an optimal schedule to minimize makespan, while avoiding deadlock situation in Flexible Manufacturing Systems (FMS) with finite capacity, in which each job needs to be processed in several job stages for completion. The proposed algorithm uses a modeling and control method based on Petri-net. Especially, we improve the efficiency of the search algorithm by using a priority rule and an efficient bounding function during the search procedure. The performance of the proposed algorithm is evaluated through a numerical experiment, showing that it holds considerable promise for providing an optimal solution efficiently comparing to past work.

Keyword: A* algorithm, deadlock avoidance policy, flexible manufacturing system scheduling, petri-nets

1. 서론

현대에 이르러 제조 현장의 측면에서는 급변하는 제조 기술의 발전과 공장의 자동화, 또한 고객의 측면에서는 제품에 대한 요구의 다양성과 짧은 제품 주기 등의 이유에 의하여 제조 업계에서는 효율성과 유연성을 동시에 갖추고 있는 Flexible Manufacturing System (FMS)의 도입이 급격히 늘어나고 있는 추세이다. 따라서 이러한 FMS에서 얼마나 효율적인 스케줄링을 하는가도 관심의 대상이 되고 있으며 지금까지 많은 연구가 많이 진행되어 오고 있다. 하지만 유한용량의 자원을 고려한 FMS에 대한 논리적 제어를 다룬 연구는 무한용량을 고려한 경우에 비해 많이 수행되지 않았으며 최근에서야 다양한 연구가 조금씩 진행되고 있는 추세이다.

본 연구에서는 <Figure 1>과 같이 구성된 유한용량 FMS를 고려한다. 이는 전형적인 FMS의 레이아웃(layout)이며, 여러 단계의 프로세싱이 필요한 작업들이 이 기계 1, 2, 3을 이용하여 연속적으로 또한 동시에 수행된다. 이 때, 처리되는 작업들에 대해 교착(Deadlock) 상태를 방지하면서 동시에 총 작업수행시간(Makespan)을 최소화하는 방안이 필요하게 된다. 본 논문에서는 이러한 유한 용량 FMS에서 총 작업소요시간을 최소로 하는 효율적인 스케줄링 알고리즘을 제시하고자 한다.

<Figure 1>의 FMS 모델은 또한 일반적인 자원할당 시스템의 분류 중 하나인 S⁴R(Systems of Sequential Systems with Shared Resources) 모델로 분류할 수 있다. S⁴R은 공유된 자원들을 사용하여 연속적으로 일어나는 공정을 갖는 작업들의 모델을 나타내며 좀 더 자세한 이론은 Abdallah *et al.*(1998)에서 확인할 수 있

[†]연락처 : 최진영 교수, 443-749 경기도 수원시 영통구 원천동 산 5번지 아주대학교 산업정보시스템공학부, Fax : 031-219-1610,
E-mail : choijy@ajou.ac.kr

투고일(2009년 01월 13일), 심사일(1차 : 2009년 01월 13일), 게재확정일(2009년 01월 21일).

다. 본 논문의 실험 대상이 되는 동일한 S⁴R 모델은 Ramaswamy *et al.*(1996)에서 처음 제기되었고 이 모델에 대해 다양한 스케줄링 알고리즘을 사용한 논문이 발표되었다. Abdallah *et al.*(1998)에서는 분기 한정법 원칙(Branch and Bound Principle)과 깊이 우선 탐색(Depth First Search) 알고리즘과 우선순위 규칙을 적용하고 구조적 제어 방법으로 Siphon을 이용하여 스케줄링 하는 방법을 제시하였다. Xiong *et al.*(1997)에서는 최고 우선 탐색(Best First Search)과 조절 가능 역추적 탐색(Controllable Backtracking Search)을 이용한 스케줄링 알고리즘을 제시하였다. Xu *et al.*(2002)에서는 유전자 알고리즘(Genetic Algorithm)과 구조적 제어 방법으로 Siphon이 사용되고 또한, He *et al.*(2006)에서도 구조적 제어 방법으로 Siphon이 사용되었으며 최고 우선 탐색(Best First Search)의 한 예인 A* 알고리즘을 사용하여 스케줄링 하였다. 그러나 이러한 기존 연구 결과는 한계 함수 및 세부적인 탐색 규칙 등에 대한 개선을 통해 탐색 알고리즘의 효율성을 높이는 것이 필요하며, 궁극적으로 FMS 구성의 확장성을 고려한 효율적인 탐색 알고리즘에 대한 연구가 필요하다.

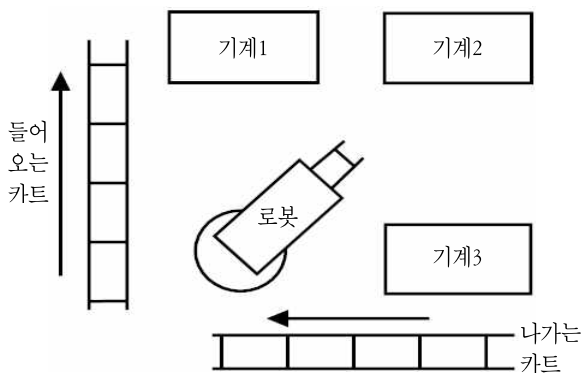


Figure 1. The layout of FMS

이를 바탕으로, 본 논문에서는 <Figure 1>에서 제안된 유한용량의 FMS에서 여러 단계의 프로세싱이 필요한 작업들에 대해 교착현상을 방지하면서 동시에 총 작업수행시간을 최소화하는 효율적인 탐색 알고리즘을 제안한다. 제안된 알고리즘은 페트리 넷(Petri net) 기반의 모델링 및 구조제어(Structural Control) 방법을 이용하며 특히, 최적 해의 탐색과정에서 프로세싱 작업을 선정할 때 시스템 상태를 고려한 우선순위 적용 및 효율적인 한계 함수를 사용함으로써 탐색 알고리즘의 효율성을 개선하였다. 특히, 기존의 A* 알고리즘을 이용한 He *et al.*(2006)보다 최대 43.7% 이상의 노드 탐색 수 감소를 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 제 2장에서는 고려된 FMS에 대한 페트리 넷 모델링 및 스케줄링 고려사항을 다루고 제 3장에서는 효율성이 개선된 알고리즘을 소개한다. 제 4장에서는 실험을 통해 제안된 알고리즘의 성능을 검증하며 마지막으로 제 5장에서 향후 연구 방향을 제안하여 논문을 마무리 하고자 한다.

2. 페트리 넷 모델링 및 스케줄링 고려사항

2.1 페트리 넷 모델링

FMS와 같이 동시 발생적이면서 비동기적인 시스템의 제어 구조를 모델링하는데 효과적인 도구로써 페트리 넷(Petri net)가 있으며 그래픽 모델을 통한 표현으로 설계자로 하여금 보다 쉽고 직관적으로 이해시키고 설계할 수 있다는 장점이 있다. 페트리 넷은 5개 요소(tuple) $PN = (P, T, F, W, M_0)$ 로 구성되어 있으며 각각의 정의는 <Table 1>과 같다.

Table 1. Formal Definition of a Petri net

요소 정의	
$P = \{p_1, p_2, \dots, p_m\}$	플레이스들의 유한집합,
$T = \{t_1, t_2, \dots, t_n\}$	트랜지션들의 유한집합,
$F \subseteq (P \times T) \cup (T \times P)$	아크들의 집합,
$W : f \rightarrow \{1, 2, 3, \dots\}$	가중 함수(weight function),
$M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$	초기 마킹,
$P \cap T = \emptyset$ 그리고 $P \cup T \neq \emptyset$.	
초기 마킹이 없는 페트리 넷 구조 $N = (P, T, F, W)$ 는 N 으로 표기한다.	
초기 마킹을 포함하는 페트리 넷은 (N, M_0) 로 표기한다.	

<Figure 2>는 <Figure 1>의 레이아웃에서 4개의 작업(job type)을 갖는 FMS 모델을 페트리 넷으로 표현한 그래프이다. 플레이스는 자원의 상태나 작업을 나타내고, 사건이나 작업 공정의 시작과 종료는 트랜지션에 의해서 표현된다. 정보의 흐름을 나타내는 토큰이 자원 플레이스에 위치하고 있다면 그 자원이 가용 상태임을 나타내며, 위치하고 있지 않다면 그 자원이 불가용 상태임을 의미한다. 또한, 작업 플레이스에 토큰이 위치하고 있다면 그 작업이 실행되고 있음을 의미하며 작업 플레이스에 토큰이 존재하지 않다는 것은 작업이 실행되지 않고 있음을 의미한다. 이러한 페트리 넷을 이용한 모델링을 통해 모델의 도달가능성, 교착상태 유무 등을 확인할 수 있으며 이를 기반으로 스케줄링 문제에 보다 쉬운 접근이 가능하다.

2.2 스케줄링 고려사항

페트리 넷 모델링에 기반을 둔 효율적인 FMS 스케줄링 구현을 위해서는 다음과 같은 사항들이 고려되는 것이 필요하다. 페트리 넷 기반의 FMS 스케줄링을 위해서는 도달가능 그래프(Reachability Graph)가 주로 사용된다. 도달가능 그래프는 페트리 넷에 의해 구현된 모델 상에서 플레이스에 위치한 토큰들의 분포를 마킹 값으로 갖는 노드(Node)들이 FMS 모델의 작업 준비 상태를 의미하는 초기 마킹(M_0)부터 작업의 완료 상태를 의미하는 마지막 마킹(M_f)까지의 모든 진행 가능한 경로

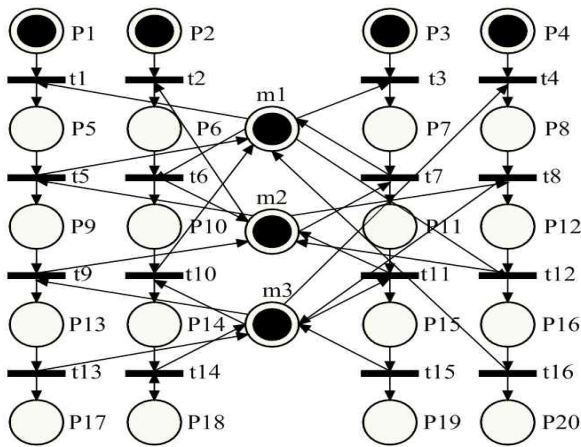


Figure 2. The Petri net model of the system

를 따라 서로 연결된 그래프로써, FMS의 가능한 모든 공정흐름을 확인할 수 있고 총 작업수행시간을 최소로 갖는 스케줄을 찾을 수 있다. 하지만 도달가능 그래프는 아무리 작은 규모의 FMS라 할지라도 많은 분석과 노드들의 수가 지수(exponential)적으로 증가하기 때문에 도달가능 그래프 전체를 구현하는 것은 많은 어려움이 있다. 따라서 최적의 스케줄을 정하기 위해 최소의 노드만을 탐색하는 방법이 요구된다.

또한 유한자원의 FMS 스케줄링에 있어서 중요한 문제 중 하나가 바로 교착상태(Deadlock)이다. 교착상태란 자원을 차지하고 있는 두 개 이상의 작업이 각각 서로의 자원을 기다리기 때문에 계속해서 대기하고 있는 상태를 말하며 최적의 스케줄링을 위해서는 순환대기(Circular Wait) 상태를 방지하여 교착 상태를 제어하는 방법이 요구된다.

3. 제안된 알고리즘

본 장에서는 앞에서 고려된 사항을 기반으로 FMS 스케줄링을 위한 효율적인 알고리즘을 제안하고자 한다. 제안된 알고리즘은 도달가능 그래프의 노드 탐색에 있어서 효율성을 높이고 불필요한 노드 탐색을 줄이기 위한 한계함수, 우선순위 규칙과 선형 부등식을 이용한 교착제어 방안(DAP) 등을 적용한다.

3.1 새로운 한계 함수의 정의

도달가능 그래프에서 총 작업소요시간을 최소로 하는 스케줄을 찾기 위해서 각각의 노드에 대한 비교작업을 필요로 한다. 이러한 고유한 마킹(M)을 갖는 노드들 간의 비교를 위한 함수로 $f(M)$ 을 정의한다. $f(M)$ 은 Cost 함수로써 각 마킹 M에 대응하는 노드를 탐색하는 과정에서 탐색 방향을 정하는 역할을 하며 정의는 다음과 같다.

$$f(M) = g(M) + h(M) \quad (1)$$

이 때 식 $g(M)$ 은 초기 마킹(M_0)부터 현재 마킹(M)까지의 걸리는 작업소요 시간을 의미하며, $g(M) = \max_i \{P_i(M)\}$ ($i = 1, 2, \dots$, 총 기계수)으로 정의된다. P_i 는 기계 i 에서 초기마킹부터 현재 마킹까지 소요되는 시간이다. 또한, 휴리스틱 함수인 $h(M)$ 은 현재 마킹(M) 이후부터 마지막 마킹(M_f)까지 남아있는 작업소요 예측 시간을 나타내며, $h(M) = \max_i \{T_i(M)\}$ ($i = 1, 2, \dots$, 총 기계수)으로 표현될 수 있다. T_i 는 기계 i 에서 현재 마킹(M) 이후부터 마지막 마킹(M_f)까지 남아있는 공정의 소요 예측 시간으로 정의되며, (기계 i 에서 남아있는 모든 공정의 작업시간의 합) - (식 (2)의 값)에 의해 추정될 수 있다.

$$\max\{g(M) - EST_i, 0\} \quad (2)$$

이 때, EST _{i} 는 각각의 기계 i 에 대하여 $g(M)$ 의 시점보다 먼저 실행될 수 있는 남아있는 공정들의 시작점이다. 즉, 각각의 기계에 대하여 $g(M)$ 의 시점보다 선행될 수 있는 남아있는 공정의 시작점을 탐색하여 만약 여러 개의 공정이 선행 될 수 있다면, 그 중에서 가장 먼저 선행되는 공정의 시작 시간을 $T_i(M)$ 에 반영하게 된다. 만약 남아있는 공정들이 $g(M)$ 의 시점보다 선행되는 작업이 없다면 식 (2)의 값은 0이 된다.

이러한 $g(M)$ 과 $h(M)$ 의 합으로 정의된 $f(M)$ 을 이용하여 도달가능 그래프의 탐색 과정에서 마지막 마킹(M_f)에 도달하기 전에 여러 개의 탐색 대상이 되는 노드들 중에서 가장 작은 $f(M)$ 을 선택 한다면 알고리즘의 완벽한(exhaustive) 탐색을 보장할 수 있다. 또한, 이렇게 정의된 $h(M)$ 은 식 (3)의 조건을 만족하며, 따라서 모든 마킹(M)에 대하여 본 연구에서 제안된 A* 알고리즘은 최소 총 작업소요시간의 산출을 보장한다(Pearl, 1981).

$$h(M) \leq h^*(M) \quad (3)$$

이 때, $h^*(M)$ 은 현재 마킹(M)에서 마지막 마킹(M_f)까지의 최적 작업소요시간을 나타낸다.

3.2 우선순위 규칙의 적용

도달 가능 그래프에서 노드를 탐색하는 과정에서 최적의 총 작업소요시간을 찾기 위해 한계 함수 $f(M)$ 을 이용한다는 것은 앞서 언급을 하였다. 하지만 $f(M)$ 이 동일한 노드가 두 개 이상 존재할 때 어떠한 노드를 선택해야 하는지의 문제가 발생한다. 선택에 따라 결과적으로 노드 탐색 수에 영향을 미칠 수 있기 때문이다. 우선순위규칙은 이와 같은 문제를 해결하기 위해 적용된다. 다시 말해 우선순위 규칙은 $f(M)$ 이 동일한 여러 개의 노드들 중에서 목적에 부합하는 노드가 선택되도록 하는 역할을 수행한다. 그동안 많은 우선 순위 규칙이 소개되었으나(Moser et al., 1992) 본 논문에서는 우선순위 규칙으로써 LTWK(Least Total Work)를 제안한다. LTWK는 $f(M)$ 이 같은 노드들 중에서 남아있는 작업소요시간이 가장 적은 노드가 선택되는 규칙이다.

3.3 선형부등식을 이용한 교착제어 방안

효율적인 탐색을 위해서는 도달 가능 그래프에서 교착 상태의 노드를 미리 판단하고 절단(Truncation)함으로써 교착상태를 제어하고 또한 불필요한 노드 탐색을 줄일 수 있다.

Siphon은 페트리 넷트 상에서 이러한 교착상태를 감지하기 위해 도입된 구조적인 특성으로써 Abdallah *et al.*(1998), Xu *et al.*(2002), He *et al.*(2006) 등 에서 스케줄링 과정에 있어서 탐색 노드 수를 줄이는 절단방법으로 사용되었다. 예를 들어 <Figure 2>의 페트리 넷트에서 Boer *et al.*(1994)에서 제안된 알고리즘에 의해 생성된 Minimal Siphon 집합 $\{S_1, S_2, S_3, S_4, S_5, S_6\}$ 은 다음과 같다.

- $S_1 = \{P_5, P_7, P_{10}, P_{16}, m_1\}$,
- $S_2 = \{P_6, P_9, P_{11}, P_{12}, m_2\}$,
- $S_3 = \{P_6, P_{12}, P_{13}, P_{14}, P_{15}, m_2, m_3\}$,
- $S_4 = \{P_8, P_{13}, P_{14}, P_{15}, m_3\}$,
- $S_5 = \{P_9, P_{10}, P_{11}, P_{16}, m_1, m_2\}$,
- $S_6 = \{P_{13}, P_{14}, P_{15}, P_{16}, m_1, m_2, m_3\}$

만약 어느 노드의 마킹이 Siphon 집합에 토큰이 없는 마킹이라면 이 노드에서 진행된 노드들은 교착 상태로 판단되어 탐색전에 절단된다(Abdallah *et al.*, 1998). 하지만 이러한 Siphon을 이용하는 방법은 FMS의 규모가 커질수록 Minimal Siphon 집합을 도출하는 과정이 매우 복잡해진다 단점이 있기 때문에 본 논문에서는 대상 문제의 확장성(scalability)을 고려하여 선형부등식을 이용한 효율적인 교착 제어 방법의 적용을 제안한다.

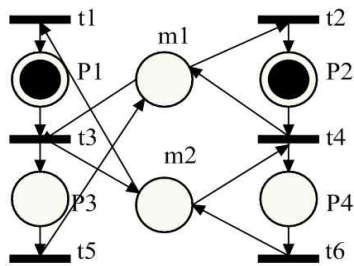


Figure 3. Deadlock situation

<Figure 3>은 작업 1의 공정이 기계 2에 위치한 상태에서 기계 1을 기다리고, 작업 2의 공정이 기계 1에 위치한 상태에서 기계 2를 기다리는 교착상태를 나타내는 페트리 넷트이다. 위와 같은 교착상태를 방지하기 위해서는 P1과 P2가 동시에 진행되는 것을 피해야 한다. 즉, P1과 P2 두 개의 플레이스에 한 개 이하의 토큰이 존재할 때 교착상태를 피할 수 있으며 이를 식으로 표현하면 다음과 같다.

$$J_{11} + J_{21} \leq 1 \tag{4}$$

식 (2)에서 좌변의 J_{ij} 는 i 번째 작업(Job Type)의 j 번째 공정

(Operation)에 있는 작업의 수를 의미 한다. 이를 <Figure 1>의 전체 시스템을 고려해서 확장시키면 최적의 교착제어 방법은 다음과 같은 8개의 선형부등식으로 표현할 수 있다(Reveliotis *et al.*, 2007).

$$\begin{aligned} J_{11} + J_{21} &\leq 1, \\ J_{11} + J_{42} &\leq 1, \\ J_{31} + J_{21} &\leq 1, \\ J_{31} + J_{42} &\leq 1, \\ J_{12} + J_{41} &\leq 1, \\ J_{32} + J_{21} &\leq 1, \\ J_{21} + J_{22} + J_{41} &\leq 2, \\ J_{22} + J_{41} + J_{42} &\leq 2 \end{aligned} \tag{5}$$

식 (5)는 기계 2대에서 순환 대기 상태(Circular Wait Condition) 일 경우와 기계 3대 전체에서 순환 대기 상태인 경우를 고려한 최적의 교착 회피정책을 나타낸 선형 부등식이다.

- STEP 1 :** OPEN 리스트에 초기 마킹(M_0)을 갖는 노드를 삽입한다.
- STEP 2 :** 만약 OPEN 리스트에 어느 노드도 존재하지 않는다면 알고리즘은 실패로 끝난다.
- STEP 3 :** OPEN 리스트에 있는 값들 중에 $f(M)$ 값을 최소로 갖는 노드를 선택하여 CLOSED 리스트에 넣는다. 이때, 동일한 $f(M)$ 값을 갖는 마킹이 두 개 이상 존재할 경우, 앞에서 제시한 LTWK 우선순위 규칙을 적용하여 남아있는 작업시간이 가장 적은 노드를 먼저 선택한다.
- STEP 4 :** 만약 OPEN 리스트에 새로이 추가된 노드가 마지막 마킹(M_f)을 갖는다면 탐색은 종료되고 이때 $f(M_f)$ 을 총 작업 소요시간으로 산출한다.
- STEP 5 :** CLOSED 리스트에 존재하는 마킹(M)에서 진행 가능한 노드들(M')을 구한다.
- STEP 6 :** 새로 구한 노드(M')에 대한 CHECK(M')이 참인 경우 STEP 5로 이동한다.
- STEP 7 :** 구한 노드들의 $f(M')$ 값을 구하고 노드들의 포인터가 부모 노드(M)를 가리키게 한다.
- STEP 7.1 :** 만약 새로이 구한 노드(M')가 이미 OPEN 리스트나 CLOSED 리스트에 존재해 있고 $f(M')$ 이 크다면 새로이 구한 노드(M')는 무시된다. 또한 새로이 구한 노드(M')가 이미 OPEN 리스트나 CLOSED 리스트에 존재해 있고 $f(M')$ 이 작다면 새로운 $f(M')$ 값으로 갱신이 되고 이때 노드(M')가 CLOSED 리스트에 존재했다면 OPEN 리스트로 이동시킨다.
- STEP 7.2 :** 구한 노드(M')가 OPEN 리스트나 CLOSED 리스트에 포함되지 않았다면 OPEN 리스트에 삽입하고 $f(M')$ 값을 할당한다.
- STEP 8 :** STEP 2로 이동한다.

Figure 4. A* algorithm

3.4 제안된 A* 알고리즘

알고리즘은 초기마킹(M_0)으로부터 시작한다. 각 탐색 단계에서 현재 마킹(M)에서 점화(firing)가능한 새로운 마킹(M')들

중에서 가장 $f(M)$ 이 작은 마킹이 선택된다. 만약 마킹이 교착 상태에 이르게 된다면 불필요한 노드탐색을 막기 위해 절단하게 된다. 마지막 마킹(M_f)까지 도달하게 되면 알고리즘은 더 이상의 탐색을 멈추고 $f(M_f)$ 이 총 작업소요시간이 된다. 탐색 과정을 통해 새롭게 형성된 마킹(M)들은 하위 탐색을 위해서 리스트에 첨가된다. 이때 이미 탐색을 마친 마킹들과 아직 탐색을 하지 않은 마킹을 구별하기 위해 각각 CLOSED 리스트와 OPEN 리스트라는 서로 다른 리스트들을 사용한다. CHECK(M) 함수는 새로 생성된 마킹 (M)이 교착상태에 빠지는지의 여부를 알기 위해 사용된다. CHECK(M) 함수가 참(TRUE)이 이라면 교착상태를 의미하고 마킹은 불필요한 확장을 피하기 위해 절단된다. 아닐 경우는 거짓(FALSE)이 된다. 이 때, CHECK(M) 함수는 앞에서 제시한 선형부등식 (5)를 적용한다. 구체적인 알고리즘에 대한 설명은 <Figure 4>와 같다.

Table 2. Job sequence and Processing time

작업	기계#: 공정 시간		
	공정 1	공정 2	공정 3
작업 1	M1 : 40	M2 : 100	M3 : 36
작업 2	M2 : 45	M1 : 65	M3 : 98
작업 3	M1 : 212	M2 : 73	M3 : 32
작업 4	M3 : 55	M2 : 65	M1 : 35

4. 실험을 통한 성능 검증

4.1 실험 설계

실험을 통해 본 논문에서 제시한 알고리즘의 효율성을 입증하였다. 비교를 위하여 <Figure 1>에서 고려한 3개의 기계를 갖는 FMS에서 서로 다른 로트 사이즈를 갖는 모델에 대해서 실험을 실시하였으며, He *et al.*(2006)에서의 결과와 비교하였다. 고려된 FMS에 대한 페트리 네트는 <Figure 2>와 같으며 각 작업에 대응하는 기계 상의 공정 시간은 <Table 2>와 같다. 예

를 들어 작업 1은 기계 1, 기계 2, 기계 3 순으로 공정이 진행되며 각각의 공정은 40초, 100초, 36초의 시간이 소요함을 의미한다. 하지만 공정 시간 이외에 로딩(loading)과 언로딩(unloading) 시간은 고려하지 않는다. 실험은 3.4GHz 펜티엄D를 사용하였고 Visual Studio 2005에서 C언어로 모델을 구현하였으며 로트 사이즈가 [1 1 1 1]인 경우, 초기 마킹(M_0)과 마지막 마킹(M_f)은

$$M_0 = \{1111000000000000000000000000111\},$$

$$M_f = \{00000000000000000000000011111111\}$$

으로 정의하였다. 앞의 20자리는 페트리 네트에서 작업 플레이스에서의 토큰의 유무를 의미하며 숫자를 통해 로트 사이즈를 설명한다. 뒤의 3자리는 자원 플레이스에서 토큰의 유무를 뜻한다. 또한 동일한 페트리 네트 모델을 기반으로 로트 사이즈가 [2 2 0 2]와 [3 4 1 3]인 경우 초기 마킹(M_0)과 마지막 마킹(M_f)은 다음과 같다.

- 로트 사이즈 [2 2 0 2]인 경우,

$$M_0 = \{2202000000000000000000000000111\},$$

$$M_f = \{000000000000000000000002202111\}$$

- 로트 사이즈 [3 4 1 2]인 경우,

$$M_0 = \{3412000000000000000000000000111\},$$

$$M_f = \{000000000000000000000003412111\}$$

4.2 실험 결과 및 분석

실험의 결과는 <Table 3>에 나타나 있다. 본 논문에서 제안한 알고리즘을 통하여 최소 총 작업소요 시간을 갖는 스케줄을 산출하였다. 동일한 A* 알고리즘을 사용한 He *et al.*(2006)과 노드 탐색 수를 비교하면 로트 사이즈가 [1 1 1 1]인 경우 기존의 He *et al.*(2006)의 알고리즘에 비해 노드 탐색 수가 조금 증가하였으나 로트 사이즈가 커질수록, 노드 탐색 수의 개선 효과가 점점 커졌음을 확인하였다. 특히 로트 사이즈가 [3 4 1 2]인

Table 3. Experimental results for different lot sizes

로트 사이즈 (J1 J2 J3 J4)	최소 총 작업 소요시간	경우 1			경우 2			경우 3
		(1)의 노드 탐색 수	(2)의 노드 탐색 수	(2)에 대한 (1)의 개선율	(1)의 노드 탐색 수	(2)의 노드 탐색 수	(2)에 대한 (1)의 개선율	
1 1 1 1	512	115	101	+13.9%	142	133	+6.8%	-19.0%
2 2 0 2	603	326	372	-12.4%	398	458	-13.1%	-18.1%
3 4 1 2	916	1308	2325	-43.7%	1781	3069	-42.0%	-26.6%

주) 경우 1: 교착제어정책에 의한 절단을 고려한 경우.

경우 2: 교착제어정책에 의한 절단을 고려안한 경우.

경우 3: (1)에서 교착제어정책에 의한 개선율.

(1): 본 논문에서 제안한 알고리즘을 사용한 경우.

(2): He *et al.*(2006) 논문에서 제안한 알고리즘을 사용한 경우.

경우 43.7%의 노드 탐색 수 감소를 통해 보다 높은 효율을 입증하였다.

이를 통해 로트 사이즈가 커질수록 제안된 알고리즘이 보다 높은 효율을 가짐을 보였다. 또한 제시된 교착제어정책에 의한 절단 기법을 통해 노드 탐색 수가 최대 26.6%까지 감소한 것을 확인하였다. 각각의 로트 사이즈에서 스케줄링 알고리즘에 의해 정해진 스케줄의 간트 차트는 <Figure 5>와 같다.

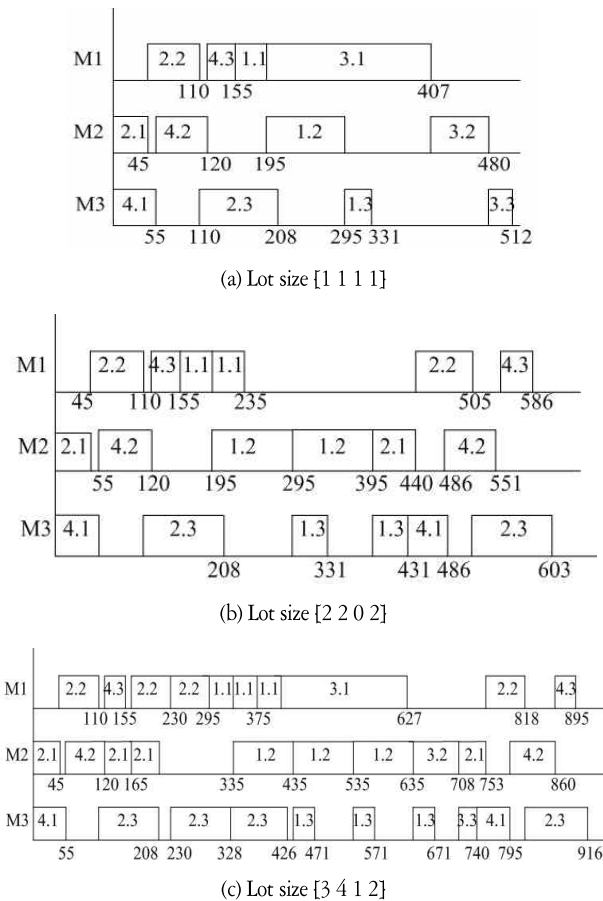


Figure 5. The Gantt chart of different lot sizes

4.3 대규모 FMS 시스템 적용 방안

본 논문에서 고려한 FMS 시스템은 기계 3대로 이루어져 있다. 하지만 실제 산업 현장에서 사용되는 FMS는 보다 많은 기계 및 자원을 사용한다. 기계의 수가 많아지고 동시에 작업의 공정 수가 늘어난다면 페트리 넷 모델로 구현되는 플레이스 와 트랜지션의 수 또한 증가하게 된다. 이러한 모델의 상태를 나타내는 마킹 상태 벡터(Marking State Vector)의 표현은 큰 문제가 되지 않지만, 도달가능 그래프 안에서의 전체 노드 수가 급격하게 증가하게 된다. 그러므로 최적의 스케줄을 산출하는 것도 중요하지만 규모가 큰 시스템의 에 대해 최적에 가까운 솔루션을 얼마나 효율적으로 도출할 수 있는지 여부도 큰 관심사이다. 이러한 문제를 해결하기 위한 방안으로 탐색 노드

수를 줄이기 위한 효율적인 bounding 함수의 개발, Resource Upstream Neighborhood (RUN) 알고리즘 또는 Banker's 알고리즘과 같은 Polynomial 시간 교착제어 알고리즘(Reveliotis, 2005)의 적용 및 노드 탐색의 효율성을 높이기 위한 새로운 우선순위 정책의 적용 등이 고려될 수 있다.

5. 결론

본 논문에서는 유한용량의 FMS에서 여러 단계의 프로세싱이 필요한 작업들에 대해 교착 현상을 방지하면서 동시에 총 작업수행시간을 최소화하는 효율적인 탐색 알고리즘을 제안하였다. 제안된 알고리즘은 페트리 넷 기반의 모델링 및 구조 제어 방법을 이용하며 최적 해의 탐색과정에서 프로세싱 작업을 선정할 때, 시스템 상태를 고려한 우선순위 적용 및 효율적인 Cost 함수를 사용함으로써 탐색 알고리즘의 효율성을 개선하였다. 특히, 기존의 A* 알고리즘을 이용한 He et al.(2006) 보다 최대 43.7% 이상의 노드 탐색 수 감소를 확인할 수 있었다.

향후 연구 과제로서 효율적인 휴리스틱 함수 h(M)의 제안을 통한 알고리즘의 개선방안 및 FMS의 확장성(Scalability)을 고려한 효율적인 교착제어정책을 적용하는 방안 등이 고려되고 있다.

참고문헌

Abdallah, I. B., El Maraghy, H. and El Mekkawy, T. (1998), An efficient search algorithm for deadlock-free scheduling in FMS using Petri nets, *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1793-1798.

Boer, E. R. and Murata, T. (1994), Generating Basis Siphons and Traps of Petri Nets Using the Sign Incidence Matrix, *IEEE Transaction on Circuits and Systems I: Fundamental Theory and Application*, **41**(4), 266-271.

Fanti, M. P. and Zhou, M. C. (2004), Deadlock control methods in automated manufacturing systems, *IEEE Transaction on Systems, Man, and Cybernetics*, **34**(1), 5-22.

He, Y. L. and Wang, G. N. (2006), Petri nets based deadlock-free scheduling for flexible manufacturing systems, *Proceedings of 2006 IEEE on Control, Automation, Robotics and Vision*, 1-5.

Lee, D. Y. and DiCesare, F. (1994), Scheduling FMS Using Petri Nets and Heuristic Search, *IEEE Transaction on Robotics and Automation*, **10**(2), 123-132.

Murata, T. (1989), Petri Nets : Properties, Analysis, and Applications, *Proceedings of the IEEE*, **77**(4), 541-580.

Pearl, J. (1981), Heuristic Search Theory : Survey of Recent Results, *Proceeding of 7th International Joint Conference Artificial Intelligence*, Vancouver, Canada, 554-562.

Ramaswamy, S. E. and Joshi, S. B. (1996), Deadlock-free Schedules for Automated Manufacturing Workstations, *IEEE Transaction Robotics Automation*, **12**(3), 391-400.

Reveliotis, S. A. (2005), Real-time management of resource allocation systems.

Reveliotis, S. A., Roszkowska, E. and Choi, J. Y. (2007), Generalized Algebraic Deadlock Avoidance Policies for Sequential Resource Allocation Systems, *IEEE Transaction on Automatic Control*, **52**(12), 2345-2350.

Xiong, H. H. and Zhou, M. (1997), Deadlock-free scheduling of an automated manufacturing system based on Petri nets, *Proceedings of the International Conference Robotics and Automation Albuquerque, NM*, 945-950.

Xu, G. and Wu, Z. (2002), Deadlock-free scheduling method using Petri net model analysis and GA search, *Proceedings of the International Conference Contr. Applications*, 2, 1153-1158.



김 황 호

아주대학교 산업정보시스템공학부 학사
 현재: 아주대학교 산업공학과 석사 과정
 관심분야: 유연 생산 시스템 모델링 및 분석,
 이산사건시스템



최 진 영

한양대학교 산업공학과 학사
 한국과학 기술원 석사
 미국 Georgia Tech 박사
 한국전자통신연구원 선임연구원
 삼성 네트워크 부장
 현재: 아주대학교 산업정보시스템공학부
 조교수
 관심분야: 이산사건시스템 이론 및 응용, SCM,
 생산 시스템 최적화, 데이터마이닝