

■ 연구보고

FMS 일정계획 수립을 위한 지식기반시스템 개발에 대한 연구

최영민

서울시스템 주식회사, 부설 공학연구소

오병완 · 김진용

동국대학교 대학원 산업공학과

이진규

동국대학교 공과대학 산업공학과

Development of a Knowledge-Based System to Establish FMS Scheduling

Young-Min Choi

Seoul System Engineering Institute, Seoul System Co.

Byeong-Wan Oh · Jin-Yong Kim

Dept. of Industrial Engineering, Dongguk University Graduate School

Jin-Gyu Lee

Dept. of Industrial Engineering, Dongguk University

Abstract

FMS are being installed to improve productivity, manufacturing consistency and flexibility. However, FMS are quite expensive and efforts must be made to avoid the high investment risk.

The objective of this paper is to enable the real-time rescheduling under dynamic changes in FMS environment. For this purpose, a KBSS (Knowledge-Based Scheduling System) in FMS environment is developed.

This KBSS will meet various requirements of users, for example, to minimize makespan, average flow time, or to maximize machine utilization.

1. 서론

오늘날 과학기술의 급속한 발달과 사회·경제적 수준의 향상으로 제품에 대한 시장의 수요는 다양해지고 있고, 제품의 수명주기는 점차적으로 짧아지고 있다. 또한, 기업측으로는 생산수명이 그 만큼 짧아지게 되었고, 이러한 변화에 탄력적으로 대응할 수 있는 생산시스템의 필요성이 절실히 요구되고 있다. 이러한 요구를 충족시킬 수 있는 시스템이 바로 유연생산시스템(FMS: Flexible Manufacturing System)이다.

FMS는 전체 생산시스템이 컴퓨터의 통제하에 운영되며, 다양한 품목들을 필요한 양 만큼 적시에 생산할 수 있도록 설계된 고도로 진보된 생산시스템이다[최정상, 1989]. 그러나, FMS는 다양한 시장변화에 신속하고 융통적으로 대응함과 동시에 생산성과 설비의 효율을 동시에 높여주고 잠재적으로 비용을 절감하는 등 많은 잇점을 갖고 있는 반면에, 도입시의 막대한 투자비용으로 인한 높은 투자위험을 감소시키기 위해서는 많은 비용과 노력이 뒤따른다[Ranky, 1988].

FMS의 높은 투자비용에 대한 빠른 회수를 위해서는 설비의 이용율을 높여야 하고, 설비의 이용율을 높이기 위해서는 설치 및 운영문제를 해결해야만 한다. 이와 관련된 문제는 FMS의 설계, 생산계획, 일정계획, 제어문제 등 대개 4가지 영역으로 구분된다[장성용, 1991].

본 논문의 연구는 이러한 영역중에서 일정계획 수립을 위한 지식기반시스템을 이용한 FMS 일정계획시스템 개발에 관한 연구이다. 생산에서 일정계획의 목적은 가능한 짧은 시간내에 가공을 마치고, 납기지연을 최소화하는데 그 목적이 있다[2, 7, 9]. 특히, FMS에서의 일정계획은 높은 설비투자비용으로 인해 기계의 가동율이 강조되어야 할 뿐만 아니라[최정상, 1989], 제어되는 시스템이 공장의 동적인 상황변화에 대해 즉각적이고, 효율적인 의사결정을 내려줄 수 있어야 한다. 그러나, 현재 컴퓨터에 기반을 둔 일정계획시스템들은 공장의 동적인 상황변화에 대해 효율적인 의사결정을 내려주지 못하고 있을 뿐만 아니라, 실시간에 제어가 거의 불가능한 상태이다[12, 13, 14, 15, 16, 17].

이러한 FMS에서의 일정계획 문제점들을 해결하기 위해 여러가지 접근방법들이 연구되고 있는데, 크게 1) 수리계획모형(mathematical model), 2) 휴리스틱 알고리즘, 3) 인공지능 접근법(Artificial Intelligence approaches) 등의 3가지 방향에서 연구가 이루어져 왔다[장성용, 1991].

그러나, 이론적으로 FMS 일정계획은 NP-Complete 문제로 수학적으로 공식화하기에는 너무 복잡하고, 공식화를 하더라도 최적해를 제공해 주지 못하므로 수리계획모형으로는 쉽게 해결할 수 없다. 또한, 휴리스틱 알고리즘 기법은 특정 상황의 특정 수행도에 대해서는 우수한 결과를 제시하지만, 일반적으로 모든 환경에 적용할 수 없고 적용영역이 제한되어 있다는 단점을 가지고 있다. 반면에, 일정계획 전문가의 경험에 의한 명쾌한 의사결정능력과 컴퓨터에 의한 보다 우수한 수행도 계산능력을 결합한 인공지능 접근방법들은 효율적인 방법으로 연구의 관심 대상이 되고 있다.

본 연구에서 제시되는 지식기반 일정계획 시스템(KBSS: Knowledge Based Scheduling System)은 FMS 환경하에서 동적인 상황변화인 사용자의 요구조건, 부품의 수

작업의 수, 납기일의 변경 등과 같은 시스템의 상황변화에 즉시 대응하여 재일정이 가능하도록 시스템이 구축된다. 이 시스템의 우수성, 유연성, 효율성에 대한 평가는 생산 현장에서의 실측 자료를 획득하기 어려운 것이 현실이므로 시뮬레이션을 통해 분석되고 평가된다.

2. 지식기반 일정계획 시스템의 이론적 고찰

2.1 지식기반 일정계획 시스템의 구조

본 연구의 지식기반 일정계획시스템은 기존의 방법과는 다른 4가지 구성요소로 되어있다. 즉, 추론기관(Inference Engine), 지식베이스(Knowledge Base), 알고리즘베이스(Algorithm Base), 데이터베이스(Data Base) 등으로 구성되어 추론기관으로 나머지 3가지 구성요소의 자료가 입력되어 처리된다[6, 8, 10].

2.1.1 지식베이스(Knowledge Base)

지식기반 일정계획시스템에서의 지식은 일정계획 관련 문헌 및 일정계획 전문가로부터 획득된다. 여기서 획득된 지식은 다음과 같은 지식표현 방법에 따라 일정계획 문제의 유형, 부품, 작업, 일정계획 형성방법 등을 나타낸다.

(1) First-order Logic

First-order Logic은 사실이나 거짓 등의 개별구문이 아니라 가정과 결론 사이의 상호관계에 기초하여 지식을 표현하는 방법이다.

$$\begin{aligned} \text{예) } \text{DONE}(\text{TASK}, t) \leftarrow & \text{COMPLETE}(P_1, t), \\ & \text{COMPLETE}(P_2, t), \\ & \text{COMPLETE}(P_3, t) \end{aligned}$$

(부품 P_1, P_2, P_3 가 각각 시간 t 에서 완성되었다면, TASK는 현재 시간 t 에서 완료된다; TASK는 부품 P_1, P_2, P_3 로 구성된다.)

(2) Horn clauses

First-order logic의 한 부분집합인 Horn clauses는 복수의 조건이나 무조건에 대해 하나의 결론만을 갖는 형태로 지식을 표현한다. Horn clauses는 First-order logic의 완전한 형태보다 프로그래밍 언어에 보다 가깝게 지식을 표현할 수 있는 장점이 있다.

(3) 생성규칙(Production Rule)

“IF ~ THEN”으로 지식을 표현하는 형식으로 그 예는 다음과 같다.

예) IF 부품 P_i 가 다른 부품 P_j 를 가공중인 기계 M_a 에 할당된다면, THEN 대안기계 M_b 의 가용성은 100%이다.

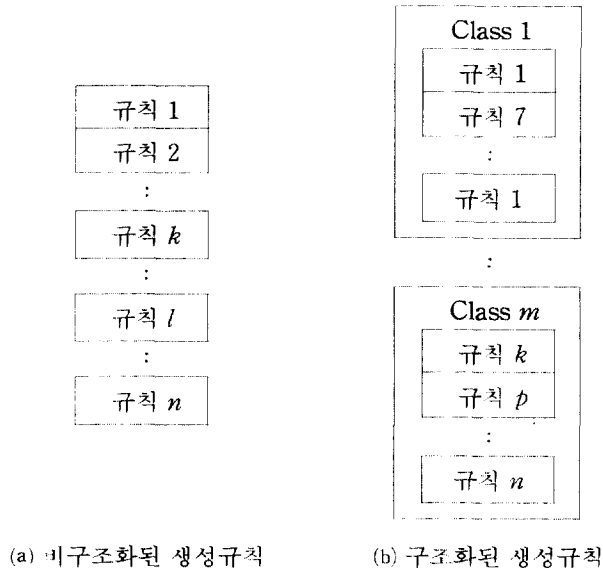
(4) 메타 규칙(Meta Rule)

다른 규칙에 작용하는 생성규칙을 메타 규칙이라 하는데 그 예는 다음과 같다.

예) IF 시간 T 가 갱신(update)되었다면, THEN 생성규칙 R_i 를 활용하라.

(5) 구조화된 생성규칙 (Structured Production Rules)

일정계획시스템의 효율성을 증진시키기 위하여 구조화된 생성규칙으로 지식을 표현한다. 비구조화된 생성규칙과 구조화된 생성규칙의 비교는 다음 (그림 1)과 같다.



< 그림 1 > 생성규칙의 두 가지 유형

(6) 프레임 (Frames)

실질적으로 생성규칙만으로 사실, 상태, 규칙 등을 표현하기에는 부족하다. 여기에는 프레임이 특히 효율적이다. 일반적으로, 대부분의 지식베이스 시스템이 프레임과 생성규칙을 결합하여 지식을 표현하고 있다.

예) (PART 1

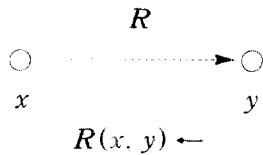
```
(AKO ($VALUE (SUBTASK)))
(SUBTASK -OF ($VALUE (TASK 1 5)))
(HAS -ACTIVITY ($VALUE (LOADING 1)
(O1 2)
(O2 3)
(UNLOADING 4)))
(DUE -DATE ($VALUE (10 (UNIT : DAY))))
(PRIORITY ($VALUE (URGENT))))
```

이 프레임은 PART 1이라는 부품을 묘사하고 있는데, 첫번째 슬롯은 PART 1이 TASK 1과 같은 큰 명령의 부분집합이라는 것을 의미하며, 실제로 두번째 슬롯은 PART 1이 주문량 5를 가진 TASK 1의 부분작업임을 의미한다. HAS -ACTIVITY는 작업순

서를 나타내는데 LOADIND이 PART1이 수행하여야할 첫번째 활동이고, 작업 1, 작업 2, UNLOADING 순으로 수행됨을 의미한다. 네번째, 다섯번째 슬롯은 그 부품의 납기일과 그 부품의 유형과 관련된 우선순위를 나타낸다.

(7) 시멘틱 네트워크(Semantic Network)

시멘틱 네트워크는 선언질의 2진 선언형의 도식적인 표현으로 그 표현법은 다음과 같다.



예) IDLE(M_1, t) ←

: 기계 M_1 은 시간 t 에서 가용할 때만하다.

MACH-OP(M_1, O_1) ←

: 기계 M_1 은 작업 O_1 을 가공처리한다.

PART-FIRST-OP(P_1, O_1) ←

: 부품 P_1 은 작업 O_1 을 첫번째로 처리한다.

2.1.2 알고리즘 베이스(Algorithm Base)

추가되는 알고리즘들은 베이스에 쉽게 결합될 수 있어야 한다. 생산규모나 자동화 수준에 따라 일정계획 문제를 해결하는 서로 다른 알고리즘들은 알고리즘베이스로 구축되어야만 한다.

2.1.3 데이터베이스(Data Base)

데이터베이스는 알고리즘들에 의해 이용되는 일정계획 모형의 모수들을 모아놓은 것이다.

2.1.4 추론기관(Inference Engine)

지식기반 일정계획시스템에서 추론기관은 지식베이스에서 규칙과 알고리즘의 일정 생성절차를 찾는 과정을 통제하는 역할을 한다. 지식기반 일정계획시스템에서의 추론기관은 “backward chaining” 또는 “forward chaining” 통제전략을 따르는데, 추론기관은 채택된 규칙들의 리스트를 간직하여 사용자가 요구할 경우 설명해 주는 기능도 한다.

2.2 지식기반 일정계획 시스템의 의의

컴퓨터 기술이 발전해감에 따라, 제조분야를 비롯한 여러 영역에서는 발생하는 문제들을 해결하는데 있어 컴퓨터가 인간의 사고방식을 흉내내어 마치 인간이 의사결정을 내려 문제를 해결하는 것처럼 작동하는 인공지능 기법이 널리 응용되고 있다. 이 기법중에서

현재 실질적으로 가장 많이 적용되고 있는 것이 바로 지식기반시스템으로써, 전문가 또는 관련서적·논문 등과 같은 기타 다른 정보원을 통해 획득된 지식을 데이터베이스화하고 그 지식을 사용자와의 접촉을 통해 사용자의 요구조건에 따라 추론기관을 거쳐 정제된 지식을 비전문가인 사용자에게 제공해주는 시스템이다[6, 8, 10]. 따라서, 지식기반시스템은 기존의 프로그램과는 상이한 것이고, FMS 환경하에서의 일정계획과 같이 단순한 수리적 프로그래밍만으로는 쉽게 해결할 수 없는 어려운 문제들을 해결하는데 이용될 수 있다. 이 방법을 적용하면, 공장에서 발생하는 동적 상황을 점검하여 지식베이스에 구축된 지식과 추론을 통하여 발생한 문제를 해결해 나감으로써 마치 생산시스템을 일정계획 전문가가 운영하는 것처럼 효율적이고 융통성있게 통제할 수 있을 것이다.

2.3. FMS 일정계획 문제의 선행연구

지식기반시스템은 많은 제조영역에서 응용되고 있는데, 특히 FMS와 같은 자동화시스템 영역에서 막대한 자본투자와 시스템의 많은 변수요인으로 그 중요성이 점차 확대되고 있다[Nasr and Elsayed, 1990].

이러한 상황에 대응하기 위해 많은 연구가 이루어졌고, 주목할만한 몇가지 일정계획시스템들에 관한 연구를 살펴보면 다음과 같다[Tabe and Salvendy, 1988].

1) ISIS(Bourne and Fox, Fox 1983, Fox et al., 1983): AI기법을 이용하여 작업일정계획 기능을 자동화하려는 최초의 시도로 카네기-멜런 대학에서 개발된 계층적 조건-직접탐색법으로, 노스캐롤라이나 윈스턴-살렘의 웨스팅하우스 터빈부품 공장을 위해 설계된 시스템이다.

2) NUDGE(Goldstein and Roberts, 1977): 관리자를 위한 약속일정을 기록하기 위해 개발된 시스템으로, 비형식적인 약속사항을 받아들일 수 있고, 작업자의 목표달성 진행사항을 감시하며, 다가오는 납기일을 관리자가 변경할 수 있도록 개발하였다.

3) ISA(Ow and Smith, 1986): 카네기-멜런 대학과 DEC(Digital Equipment Corporation)에 의해 공동개발된 고수준의 일정계획 시스템으로 재고량과 공장의 생산능력을 총합하여 회사의 영업사원이 고객이 요구한 납기일에 맞출 수 있는 일정계획을 제공한 것으로 일정계획 전문가의 경험을 표현할 수 있는 생성규칙을 기반으로 구축하였다.

4) HTS(Miller, 1987): 반자동 제조환경에서의 일정계획 자동생성을 위한 휴리스틱기법 기반 시스템으로, 시스템에서 이용되는 해결방법은 가능한 일정들의 트리생성에 있다.

5) CALLISTO(Fox et al., 1984, Sathi and Fox, 1985): SRL을 이용한 지식기반 시스템으로 프로젝트 관리 문제를 해결하기 위해 개발하였다.

6) OPIS(Ow and Smith, 1986): ISIS의 단점을 해결하기 위해 카네기-멜런 대학에서 개발된 시스템으로, 시스템 구조는 문제를 분해하여 문제의 상태에 따른 해결방법을 휴리스틱 기법을 이용하여 일정계획을 선택하였다. 지식 구조는 문제해결 활동의 통제 및 조정 사례가 담긴 지식 사원의 계층적 조직화에 있으며, 상호작용 일정계획시스템이 아니기 때문에 실시간 문제를 다루지 못하였다.

이러한 선행연구를 고찰해 보면, 시스템들이 대부분 예측 기능을 가진 일정계획시스템이라는 단점을 갖고 있다. 그러나, 제조환경에서 일반적으로 발생하는 문제들은 예측 불

가능한 동적인 것이기 때문에 예측에 의한 일정계획시스템은 생산제조 환경에서 유발될 수 있는 문제에 적절히 대응하기가 어렵다[Tabe and Salvendy, 1988]. 따라서, 본 논문에서는 제조환경에 있어서 다양한 변화가 있는 일정계획 문제를 실시간에 해결할 수 있는 방법을 제시하고자 하는 것이다.

3. 시스템의 설계와 개발

3.1 시스템의 설계

본 연구에서 다루게 될 FMS 일정계획은 동적일정계획 문제로, 작업대상물이 시스템에 도착했다더라도 시스템내에서 한 대 이상의 기계가 가용할 때만 비로소 작업순서를 결정하게 된다. 동적일정계획은 공장이나 시스템내에서 발생하는 동적 상황에 따라 일정계획 또는 재일정계획이 이루어지게 되는데, 고려될 수 있는 동적상황들을 살펴보면, 1) 기계의 고장, 2) 작업의 새로운 배척, 3) 기계에서 한 부품의 작업완료, 4) 운반장비의 고장 등으로 인한 운반경로의 변경, 5) 납기일 변경 등이 있다. 본 연구에서 개발하고자 하는 일정계획시스템은 이러한 동적일정계획 문제를 해결할 수 있도록 설계되었다.

3.1.1 기본가정

본 연구에서 제시하게 될 일정계획시스템을 구축하고 수행하는데 있어서, 기본가정은 다음과 같다.

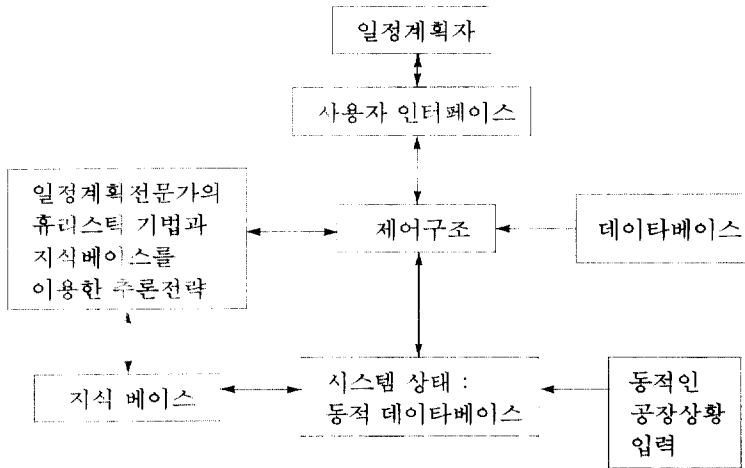
- 1) 각 작업의 가공시간은 사전에 알려진 값으로 확정적이다.
- 2) 한 기계에서 한번 시작된 작업이 완성되기 전까지는 중단되지 않는다.
- 3) 각 기계들은 여러 작업을 동시에 수행할 수 없다.
- 4) 작업은 요구된 가공순서에 따라 구성되며, 사용자가 임의로 변경할 수 없다.
- 5) 대기중인 작업은 워크 센터(Work Center)에 포함된 재공품 저장공간이나 별도의 재공품 저장소에 대기하게 된다.
- 6) 부품 또는 자재는 팰릿의 여유가 생길 때마다 시스템에 동적으로 들어온다.
- 7) 작업순서는 시스템에서 한 대 이상의 기계가 가용할 때 결정된다.
- 8) 가공 및 운반장비는 네트워크 통신을 통해 시스템제어기에 의해 통제되고, 모든 관련 데이터는 일정계획시스템을 이용하는 사용자에게 의해 언제든지 검색이 가능하도록 데이터베이스에 저장된다.

3.2 시스템 구조

본 연구에서 제시되는 일정계획시스템의 기본구조는 <그림 2>와 같다.

여기서, 제어구조는 '사용자 인터페이스'로 부터 받아들여진 사용자인 일정계획자의 요구에 대해 시스템의 상태를 점검하여 지식베이스의 문제해결 구조를 통해 최적의 일정계획을 수립하거나 재일정계획을 가능케 하는 기능을 갖는다.

시스템의 입력요소는 데이터베이스에 기록된 내용을 검색하여 이용하게 되는데, 각 기



< 그림 2 > 지식기반 FMS 일정계획시스템의 구조

계에 할당되는 작업순서, 각 작업을 처리할 수 있는 대안 기계의 리스트와 같은 사전일정 데이터와 아울러 작업번호, 처리시간, 납기일, 작업유형(정상작업 또는 긴급작업) 등과 같은 각 작업의 상세한 데이터를 필요로 한다. 또한, 공장에서 발생하는 동적인 상황은 근거리통신망(LAN: Local Area Network)을 통해 시스템 상태를 기록하는 동적 데이터베이스에 저장된다.

동적상황이 발생되면 이것은 시스템 상태를 기록하는 동적 데이터베이스에 기록되고 다시 제어구조를 통해 사용자 인터페이스에 전달되어 새로운 휴리스틱 기법이나 추론전략을 통해 재일정계획이 가능하게 되는 것이다.

발생된 동적상황에 대한 문제해결 구조로 이루어진 지식베이스에 기록되는 지식들의 표현방법은 프레임 구조를 이용한다. 프레임은 응용 영역에서 객체들과 객체들에 대한 클래스의 구조화된 표현을 제공해 주고, 클래스 표현으로부터 개별적인 표현에 이르기까지 표현이동을 알려주는 구조(상속성)를 제공해 준다. 또한, 계산표현을 위한 절차의 기술을 가능케 해주고 특정 지식이 없는 경우 표현된 것 중 하나를 선택할 수 있도록 하는데, 이러한 것들은 프레임으로 구조화된 지식이 부분문제들로 나누어져 해결될 수 있는 분석적 모형을 수립하는데 이용될 수 있기 때문이다.

사용자 인터페이스는 사용자의 요구사항을 시스템의 제어구조에 정확히 전달할 수 있도록 사용자가 사용하기 쉽고 알아보기 편리하도록 설계되어 있으며, 보조입력장치인 마우스(mouse)를 이용하여 풀다운(pull-down) 또는 팝업(pop-up) 메뉴방식으로 나타나는 사항들을 선택할 수 있도록 하였다.

이 시스템은 IBM-PC AT, 386, 486 호환기종 환경에서 운용되도록 설계되었으며, 개발에 사용된 언어는 객체지향적 프로그래밍 언어(Object Oriented Programming Language)인 미국 볼랜드(Borland)사의 C++이다.

3.3. 시스템의 개발

본 연구의 지식기반 일정계획 시스템은 사용자가 일정계획시스템을 사용할 때, 사용자 인터페이스는 사용자에게 현재의 공장 및 시스템 상황을 보여주고, 사용자는 부품의 새로운 배치를 입력하거나 새로운 요구사항을 선택한다. 이러한 사용자의 요구에 따라 일정계획시스템은 시스템 상태를 저장하고 있는 동적 데이터베이스와 작업에 관한 일반적인 데이터를 저장하고 있는 데이터베이스를 검색해서 검색된 데이터를 지식베이스를 이용한 휴리스틱 기법이나 추론전략을 통하여 재일정계획을 수행하도록 한다.

본 연구에서는 일정계획전문가의 지식을 쉽게 획득하기 어렵기 때문에, 휴리스틱 알고리즘을 개발하여 일정계획을 수행하도록 한다. 또한, 이 알고리즘에서는 사용자가 요구하는 수행도의 평가 척도에 따라 최적 할당규칙을 적용하게 되는데, 이에 따른 최적 할당규칙의 선택은 시뮬레이션을 통해 얻어진 지식을 기초로 한다.

3.3.1 작업순서 결정을 위한 휴리스틱 알고리즘

(1) 기호 설명

- OP_i : 작업 i ($i = 1, \dots, n$), P_i : 부품 i ($i = 1, \dots, l$),
- M_i : 기계 i ($i = 1, \dots, m$), t : 현재시간
- S_0 : 현재 t 시간에서 일정계획수립이 불가능한 작업들의 집합,
- S_1 : 현재 t 시간에서 일정계획수립이 가능한 작업들의 집합
- S_2 : 현재 t 시간에서 처리중인 작업들의 집합
- S_3 : 현재 t 시간에서 처리가 완료된 작업들의 집합
- P : 현재 처리중인 부품들의 집합
- M : 현재 이용중인 기계들의 집합
- R_i : 잔여처리시간(현재시간 + OP_i 의 처리시간)
- T : 잔여처리시간들의 집합

(2) 알고리즘의 단계

본 논문에서 개발한 KBSS의 알고리즘의 각 단계는 다음과 같다.

- 【단계 1】** 시스템 상태를 점검하여 현재시간 t 와 모든 기계와 부품의 가용할 때만 상태를 알아내고, 현재 처리중인 작업들의 집합 S_2 를 구성한다.
- 【단계 2】** t 시간 현재 일정계획이 가능한 작업들의 집합 S_1 과 일정계획이 불가능한 작업들의 집합 S_0 를 구성한다. 집합 $S_1 = \emptyset$ 이면, 단계 5로 간다(일정계획이 가능한 작업이 되기 위해 서는 현재 그 작업을 처리하는데 이용되는 기계가 가용할 때만한 상태이어야 하며, 현재 그 작업이 속한 부품이 가공되고 있지 않아야 하며, 선행 작업이 없거나 선행작업이 있더라도 현재 t 시간에서 완료되었어야 한다).
- 【단계 3】** 사용자가 요구한 조건에 따라(예 : 총가공시간을 최소화하는 일정계획, 평균흐름시간을 최소화하는 일정계획, 최대흐름시간을 최소화하는 일정계획, 기계이용율을 최대화하거나 적정수준 이하로 조정하는 일정계획 등), 시뮬레이션을 통해

언어진 최저 할당규칙을 적용해서 집합 S_1 중에서 하나의 작업 OP_i 를 선택한다.

【단계 4】 처리중인 작업들의 집합 S_2 에 OP_i 를 추가한다.

일정계획이 가능한 작업들의 집합 S_1 에서 OP_i 를 삭제한다.

잔여처리시간 R_i 를 계산한다.

여기서 구한 R_i 를 잔여처리시간의 집합 T 에 추가한다.

집합 S_1 에서 OP_i 가 제거됨에 따른 부품과 기계와 같은 가용자원의 상태변화에 따라 단계 2에서와 같이 집합 S_0 와 S_1 을 재구성한다. 집합 $S_1 \neq \emptyset$ 이면, 단계 3으로 간다. 만약 $S_1 \cup S_2 = \emptyset$ 이면, 알고리즘을 끝낸다.

【단계 5】 잔여처리시간의 집합 T 중에서 현재시간 t 이후로 가장 작은 잔여처리시간을 갖는 작업 OP_i 를 선택한다. OP_i 를 완료된 작업의 집합 S_3 에 추가한다. 처리중인 작업의 집합 S_2 에서 OP_i 를 삭제한다. 같은 잔여처리시간을 갖는 작업이 있을 경우 반복한다.

현재시간 t 를 OP_i 의 완료시간으로 갱신한다. 단계 2로 간다.

3.3.2 지식베이스의 형성

본 논문에서 제시된 지식기반 일정계획시스템의 지식베이스는 기본적으로 4가지 유형의 프레임으로 구성되는데, 그 구조를 살펴보면 다음과 같다.

프레임 1

프레임명 "PROBLEMS"

```
{ 문제번호 : [ number ]
  { 문제특징
    { 부품의 수 : [ number ],
      작업의 수 : [ number ] } }
  { 사용자 요구조건 : [ 총가공시간의 최소화/
    평균흐름시간의 최소화/
    기계이용율의 최대화 ] } }
```

프레임 2

프레임명 "PARTS"

```
{ 부품번호 : [ number ]
  { 부품속성
    { 남기시간 : [ number(hours) ],
      작업의 수 : [ number ],
      처리계획의 수 : [ number ] } }
  { 처리계획 1 { 작업번호 : [ number ],
    기계번호 : [ number ],
    처리시간 : [ number(minute) ] } }
  :
  }
```

```
{ 처리계획  $n$  { 작업번호 : [ number ],
               기계번호 : [ number ],
               처리시간 : [ number(minute) ] } } }
```

프레임 3

프레임명 "OPERATIONS"

```
{ 작업번호 : [ number ]
  { 작업속성
    { 관련 부품번호 : [ number ] }
    { 작업의 긴급도 : [ 긴급/정상 ] }
    { 선행작업의 수 : [ number ] }
    { 후속작업의 수 : [ number ] } }
  { 선행작업 1 { 작업번호 : [ number ] } }
    :
  { 선행작업  $a$  { 작업번호 : [ number ] } }
  { 후속작업 1 { 작업번호 : [ number ] } }
    :
  { 후속작업  $b$  { 작업번호 : [ number ] } } }
```

프레임 4

프레임명 "SCHEDULING"

```
{ 문제번호 : [ number ]
  { 할당규칙번호 : [ number ] }
  { 기계유희시간
    { 기계 1 : [ number(minute) ] }
      :
    { 기계  $m$  : [ number(minute) ] } }
  { 부품가공완료시간
    { 부품 1 : [ number(minute) ] }
      :
    { 부품  $p$  : [ number(minute) ] } }
  { 기계이용율 : [ number ] } }
```

3.3.3 수행도 평가 척도에 따른 할당규칙의 시뮬레이션

본 논문에서 제시된 휴리스틱 알고리즘은 사용자가 요구하는 조건에 따라 작업을 선택하는 기준이 달라진다. 따라서, 사용자의 요구조건에 따른 최적 할당규칙의 적용을 위해 시뮬레이션을 통한 결과가 지식기반 일정계획시스템의 하나의 지식으로 제공된다.

일정계획에 관련된 사용자의 요구조건(수행도 평가 척도)은 크게 분류하여 첫번째, 총가공시간(makespan)을 최소화하는 일정계획, 두번째, 평균흐름시간(average flowtime)을 최소화하는 일정계획, 세번째, 기계이용율(machine utilization)을 최대화하는 일정계획 등이 있다. 이러한 요구조건을 최적으로 만족시키는 일정계획을 위해 다음과 같은 12

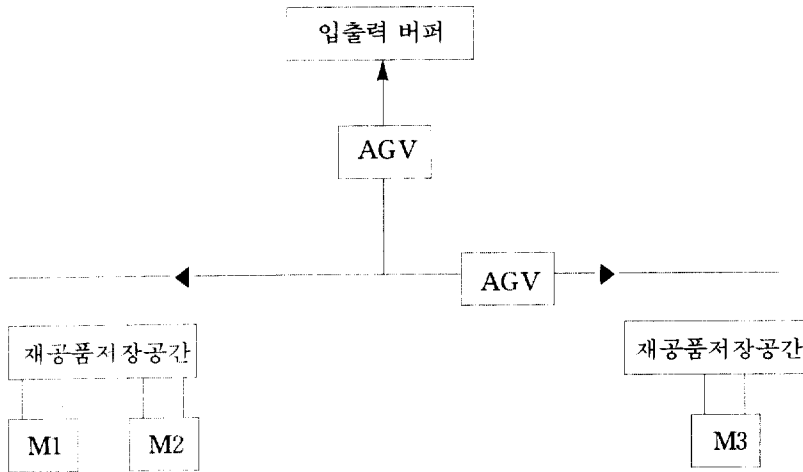
개의 서로 다른 할당규칙들을 사용자의 요구에 따라 테스트하게 되는데, 그 규칙들은 다음과 같다.

1. SIO(shortest imminent operation time) : 가장 짧은 처리시간을 갖는 긴급작업을 선택한다.
2. SPT(shortest processing time) : 가장 짧은 처리시간을 갖는 작업을 선택한다.
3. SRPT(shortest remaining processing time) : 가장 짧은 잔여처리시간을 갖는 작업을 선택한다.
4. SMT(smallest value obtained by multiplying the processing time of the imminent operation by the total processing time for the part) : 긴급작업의 처리시간에 총처리시간을 곱해서 최소값을 갖는 작업을 선택한다.
5. SDT(smallest ratio obtained by dividing the processing time of the imminent operation by the total processing time for the part) : 긴급작업의 처리시간을 총처리시간으로 나누어 그 비율이 가장 작은 작업을 선택한다.
6. LIO(longest imminent operation time) : 가장 긴 처리시간을 갖는 긴급작업을 선택한다.
7. LPT(longest processing time) : 가장 긴 처리시간을 갖는 작업을 선택한다.
8. LRPT(longest remaining processing time) : 가장 긴 잔여처리시간을 갖는 작업을 선택한다.
9. LMT(largest value obtained by multiplying the processing time of the imminent operation by the total processing time for the part) : 긴급작업의 처리시간에 그 부품의 총처리시간을 곱해서 가장 큰 값을 갖는 작업을 선택한다.
10. LDT(largest ratio obtained by dividing the processing time of the imminent operation by the total processing time for the part) : 긴급작업의 처리시간을 총처리시간으로 나누어 그 비율이 가장 큰 작업을 선택한다.
11. MRO(largest number of remaining operations) : 후속작업의 수가 가장 많은 작업을 선택한다.
12. FRO(fewest number of remaining operations) : 후속작업의 수가 가장 적은 작업을 선택한다.

4. 시뮬레이션의 실행과 결과분석

4.1 모형의 설정

본 연구에서 시뮬레이션의 실행과 결과를 분석하고 고찰하기 위한 모형은 〈그림 3〉과 같다. 사용자의 요구에 따른 최적 할당규칙을 지식으로 채택하기 위하여 각 각 1대의 머시닝센터를 보유한 3대의 W/S과 2대의 AGV로 구성된 시스템으로 10개의 서로 다른 부품을 생산할 수 있는 FMS 제조환경을 예로 든다.



< 그림 3 > FMS 제조회장의 설비배치

4.2 시뮬레이션의 수행 목적과 방법

본 연구에서 시뮬레이션을 수행하는 주된 목적은 연구된 사용자의 요구조건에 따른 변화, 작업의 수, 납기일의 변경 등과 같은 시스템 상태의 변화에 즉시 대응하여 재일정이 가능하도록 작업순서의 결정, 할당규칙에 따른 수행도의 분석 및 알고리즘의 효율성을 보이는 데 있다.

시뮬레이션 수행 방법은 생산현장의 실측자료를 구하기 어려운 것이 현실이므로 동적 제조회장에 관한 각종 데이터는 통계분포를 이용하여 사용하였다. 즉, 일양분포 U(1,10)과 U(2,5)로 생성한 부품번호, 주문량, 납기일을 갖는 <표 1>의 주문 데이터를 기준으로 사용자의 요구에 따른 수행도를 평가한다. 각 부품에 관한 작업의 수는 U(3,5), 작업 처리시간은 U(5,10)의 일양분포 난수로 발생시킨 데이터를 이용하여 30회씩 실시하였다. KBSS는 객체 지향적 프로그래밍 언어인 볼랜드사의 C++를 사용하여 개발하였고, IBM 호환기종 386DX/33 컴퓨터로 실행하였다.

4.3 시뮬레이션의 결과 및 분석

4.3.1 부품의 수, 작업의 수가 고정되었을 경우

사용자가 요구하는 총가공시간을 최소화하는 경우, 평균흐름 시간을 최소화하는 경우, 기계이용율을 최대화하는 경우로 나누어 시뮬레이션을 수행한 결과, 12가지 할당 규칙의 수행도 평가 척도별 효율성의 우선순위는 다음과 같았다 <표 2 참조>.

- 1) 총가공시간을 최소화하는 경우 : SPT > SIO > MRO > LMT > LDT > SDT > LRPT > LPT > LIO > FRO > SMT > SRPT
- 2) 평균흐름시간을 최소화하는 경우 : MRO > SIO > SPT > LDT > LRPT > SDT > LMT > LIO > LPT > SMT > FRO > SRPT

3) 기계이용율을 최대화하는 경우 : MRO > SIO > SPT > LRPT > LDT > SDT > LIO > LPT > LMT > SMT > FRO > SRPT

〈 표 1 〉 주문 데이터

주문번호	부품번호	주문량	납기일
0001	003	7	5
0002	001	10	4
0003	005	4	4
0004	009	5	2
0005	004	3	3
0006	002	1	3
0007	008	10	5
0008	010	5	2
0009	007	6	5
0010	005	2	4

〈 표 2 〉 할당규칙 선택을 위한 시뮬레이션 결과

할당규칙	총가공시간	평균호류시간	기계이용률
SIO	675.23	563.49	0.941
SPT	673.73	567.79	0.933
SRPT	720.17	624.18	0.863
SMT	718.90	603.37	0.893
SDT	684.63	577.63	0.918
LIO	690.03	582.54	0.915
LPT	686.90	583.72	0.913
LRPT	686.53	576.38	0.922
LMT	679.53	583.29	0.908
LDT	684.20	575.64	0.921
MRO	675.97	537.64	0.984
FRO	696.13	611.32	0.870

4.3.2 부품의 수, 작업의 수가 변화될 경우

본 절에서는 동적인 제조환경, 즉 부품의 수와 작업의 수가 변화하는 상황하에서 사용자가 원하는 수행도 척도에 따라 최적 할당규칙이 선택될 수 있다는 것을 보이기 위해 몇가지 결과를 제시한다.

〈 표 3 〉 부품수 = 3, 작업수 = 12 경우의 수행 결과

부품	작업	할당규칙	총가공시간	평균흐름시간	기계이용율	cpu time
3	12	SIO	355.00	267.33	0.721	1.14
		SPT	354.00	280.33	0.714	1.09
		SRPT	330.00	276.67	0.669	3.30
		SMT	355.00	276.33	0.721	1.17
		SDT	337.00	270.33	0.728	1.19
		LIO	337.00	270.33	0.728	1.18
		LPT	330.00	276.67	0.669	1.11
		LRPT	354.00	280.33	0.714	2.89
		LMT	337.00	270.33	0.728	1.18
		LDT	337.00	270.33	0.728	1.19
		MRO	337.00	270.33	0.728	1.27
		FRO	346.00	296.00	0.624	1.12

총가공시간의 최소화 : SRPT, LPT

평균흐름시간의 최소화 : SDT, LIO, LMT, LDT, MRO

기계이용율의 최대화 : SDT, LIO, LMT, LDT, MRO

〈 표 4 〉 부품수 = 5, 작업수 = 17 경우의 수행 결과

부품	작업	할당규칙	총가공시간	평균흐름시간	기계이용율	cpu time
5	17	SIO	337.00	263.33	0.842	1.76
		SPT	313.00	264.67	0.828	1.53
		SRPT	319.00	286.00	0.776	5.93
		SMT	330.00	275.33	0.798	1.82
		SDT	310.00	281.00	0.787	1.87
		LIO	310.00	281.00	0.787	1.86
		LPT	314.00	271.00	0.826	1.78
		LRPT	321.00	275.00	0.794	5.14
		LMT	310.00	281.00	0.787	1.86
		LDT	310.00	281.00	0.787	1.85
		MRO	311.00	256.00	0.857	1.92
		FRO	310.00	283.00	0.783	1.88

총가공시간의 최소화 : SDT, LIO, LMT, LDT, FRO

평균흐름시간의 최소화 : MRO

기계이용율의 최대화 : MRO

〈 표 5 〉 부품수 = 7, 작업수 = 23 경우의 수행 결과

부품	작업	할당규칙	총가공시간	평균흐름시간	기계이용율	cpu time
7	13	SIO	366.00	238.00	0.919	1.58
		SPT	366.00	253.00	0.884	11.26
		SRPT	416.00	286.33	0.809	2.12
		SMT	366.00	235.33	0.898	2.25
		SDT	409.00	283.00	0.835	2.17
		LIO	409.00	284.67	0.813	2.06
		LPT	416.00	307.33	0.714	11.96
		LRPT	366.00	215.67	1.000	2.12
		LMT	366.00	235.33	0.898	2.25
		LDT	366.00	235.33	0.898	2.58
		MRO	391.00	282.67	0.828	2.23
		FRO	367.00	237.07	0.924	1.98

총가공시간의 최소화 : SIO, SPT, SMT, LRPT, LMT, LDT

평균흐름시간의 최소화 : LRPT

기계이용율의 최대화 : LRPT

위의 결과를 분석, 고찰하여 보면 부품의 수와 작업의 수가 변화하는 상황에서 본 연구에서 개발한 KBSS를 이용했을 때 수행도 평가 척도에 따라 자동적으로 최적의 할당규칙을 제시하여 준다는 것을 알 수 있다. 또한, 부품의 수와 작업의 수가 많을 수록 CPU time은 오래 걸린다는 것을 알 수 있다. 그러나, 이러한 문제점들은 지식베이스에 동일하거나 유사한 경우의 일정계획 결과가 수록되어 있다면, 지식베이스를 검색하고 추론하는 시간만으로 신속한 일정계획이 가능할 것이다.

5. 결론

본 연구는 FMS 상황에서 기존 연구의 문제점으로 나타나고 있는 동적상황에서의 실시간 재일정계획 문제를 해결할 수 있는 시스템을 구축하고자 하였다. 이 시스템의 특징은 동적인 상황변화인 사용자의 요구조건, 부품의 수, 작업의 수, 납기일의 변경 등과 같은 시스템의 상황변화에 즉시 대응하여 재일정이 가능하도록 시스템을 구축하였다.

본 논문에서는 이러한 동적인 상황변화에 대응해서 실시간에 재일정계획이 가능하도록 지식베이스에 구축된 동일한 상황에 대한 지식을 검색하고 추론하는 시간만으로 재일정계획이 이루어질 수 있도록 개발하였다. 따라서, 본 논문에서 개발한 FMS 일정계획을 위한 지식기반시스템은 다양한 상황변화에 대해 실시간에 일정계획자에게 의사결정능력을 제공해 준다.

그러나, 본 논문에서 개발된 일정계획시스템을 직접 현장에서 적용하는데는 많은 제약 조건이 따른다. 우선 생산시스템이 완전히 컴퓨터에 의해 제어가 가능한 시스템이어야 한다. 또한, 공구의 교환 및 보정, 자동유도운반차의 최적 운반경로, 생산설비의 확장 및 교체에 따른 제조환경의 변화 등 수많은 변수들이 고려되어야 한다.

앞으로의 연구는 일정계획 문제 하나만을 갖고 FMS 제조환경에서의 문제점을 해결해 나가기 보다는 설계문제, 생산계획 문제, 일정계획 문제, 제어문제 등 모든 관련 문제들을 종합적으로 연구해 나가는 방향이 되어야 할 것이다.

아울러 앞으로의 지식기반 일정계획시스템은 일정계획자의 의사결정 과정의 분석을 통해 새로운 지식으로 학습하는 과정이 포함된 Tandem 시스템으로 개발되어야 할 것이다. 그러한 시스템을 완성하기 위해서는 일정계획자의 의사결정 과정을 분석하여 추론할 수 있도록, 프로토콜 분석과 신경망 구조시스템에 관한 연구들이 병행되어야 할 것이다.

참고문헌

- [1] 장성용 (1991), 「FMS의 생산계획 및 일정계획을 위한 통합적인 의사결정지원 시스템」, 서울대학교 대학원 박사학위 논문.
- [2] 최정상 (1989), “자동생산시스템(유연생산시스템)에 관한 연구개관,” *Journal of the KSQC*, Vol. 17, No. 2, pp. 170-176.
- [3] Chan, D. Y. and Bedworth, D. D. (1990), “Design of a scheduling system for flexible manufacturing cells,” *INT. J. PROD. RES.*, Vol. 28, No. 11, pp. 2037-2049.
- [4] Firebaugh, M. W. (1988), *Artificial Intelligence: A Knowledge-Based Approach*. PWS-KENT Publishing Co.
- [5] Hu, D. (1988), *C/C++ for Expert Systems*, MIS Inc.
- [6] Kanet, J. J. and Adelsberger, H. H. (1987), “Expert systems in production scheduling,” *European Journal of Operational Research*, Vol. 29, pp. 51-59.
- [7] Kusiak, A. and Chen, M. (1988), “Expert systems for planning and scheduling manufacturing systems,” *European Journal of Operational Research*, Vol. 34, pp. 113-130.
- [8] Kusiak, A. (1990), *Intelligent Manufacturing Systems*, Prentice-Hall, Inc.
- [9] Montazeri, M. and Van Wassenhove (1990), L. N., “Analysis of scheduling rules for an FMS,” *INT. J. PROD. RES.*, Vol. 28, No. 4, pp. 785-802.
- [10] Nasr, N. and Elsayed, E. A. (1990), “Job shop scheduling with alternative machines,” *INT. J. PROD. RES.*, Vol. 28, No. 9, pp. 1595-1609.
- [11] Ranky, P. G. (1988), “A real-time, rule-based FMS operation control strategy in CIM environment - Part I,” *INT. J. CIM*, Vol. 1, No. 1, pp. 55-72.
- [12] Sarin, S. C. and Salgame, R. R. (1990), “Development of a knowledge-based

- system for dynamic scheduling," *INT. J. PROD. RES.*, Vol. 28, No. 8, pp. 1499–1512.
- [13] Shaw, M. J. and Winston, A. B. (1989), "An Artificial Intelligence Approach to the Scheduling of Flexible Manufacturing Systems," *IIE Transactions*, Vol. 21, No. 2, pp. 170–183.
- [14] Tabe, T. and Salvendy, G. (1988), "Toward a hybrid intelligent system for scheduling and rescheduling of FMS," *INT. J. CIM*, Vol. 1, No. 3, pp. 154–164.
- [15] Shaw, M. J. and Winston, A. B. (1989), "An Artificial Intelligence Approach to the Scheduling of Flexible Manufacturing Systems," *IIE Transactions*, Vol. 21, No. 2, pp. 170–183.
- [16] Shaw, M. J. (1988), "Knowledge-based Scheduling in flexible manufacturing systems: An integration of pattern-directed inference and heuristic search," *INT. J. PROD. RES.*, Vol. 26, No. 5, pp. 821–844.
- [17] Tabe, T. and Salvendy, G. (1988), "Toward a hybrid intelligent system for scheduling and rescheduling of FMS," *INT. J. CIM*, Vol. 1, No. 3, pp. 154–164.