

자치적 객체들간의 Bidding을 통한 FMS 생산현장통제에 관한 연구

°문장석*, 박진우**

* ㈜대우자동차, ** 서울대학교 산업공학과

A Study on the FMS Shop Floor Control by Bidding among Autonomous Objects

Jang Seok Moon*, Jin Woo Park**

* Daewoo Motors Co., ** Seoul National University, Dept. Industrial Eng.

본 연구에서는 객체지향 개념과 분산형 문제해결 방법의 장점을 혼합한 FMS 생산현장통제 시스템을 제안하였다. 제안된 시스템은 FMS의 모든 구성요소를 지능적인 객체로써 표현하고, 객체들간의 Bidding 과정을 통하여 생산현장의 상태변화에 대응해 나가는 구조를 가지고 있다. 실험 결과, 제안된 시스템은 기존의 Dispatching rule에 비하여, job의 시스템내 체류시간이 감소하며 기계부하가 평준화되는 현상을 나타내었다.

주요어 : 객체지향 개념, 분산형 문제해결 방법, 자치적 Bidding, 생산현장통제, 유연생산시스템(FMS)

1. 서론

대량생산체제는 현대의 소비자의 다양한 기호에 따른 제품 수명주기의 단축 및 다양화에 의해 그 장점이 상실되었다. 대량생산체제의 효율성과 Job shop의 유연성을 혼합하여 중간생산량(mid-volume)의 batch생산과, 중간 다양성(mid-variety)의 제품 생산에 적합한 FMS가 등장하였다.

FMS의 운영에 관련된 문제는 크게 생산계획문제, 생산일정계획문제, 분개문제로 나누어 볼 수 있다. FMS 생산계획문제에는 부품공류선택문제, 기계군 형성 문제, 생산제품비율문제, 자원할당문제 및 작업할당문제등이 있다. FMS 일정계획문제에는 부품투입순서결정문제, 작업순서결정문제, 기계선정문제, 부품우선순위문제, 물자이송장치선택문제, 교통통제문제, 작업자선정문제등이 있다. FMS 통제문제에는 NC 기계의 고장 및 수리문제, 수요변동이나 기계 고장에 따른 일정계획의 재조정 및 대체공정결정, 재공품 및 완제품의 검사 정책 결정, 공구관리 및 공구 교환 통제, 정보 수집 및 공정 통제, Feedback 등이 있다.

생산현장통제에 관한 많은 연구에도 불구하고 실질적으로 생산현장에서는 단순한 우선순위규칙에 의한 통제기법들이 사용되고 있다. 이는 제안된 기법들이 개념적으로 또한 구현하기에 매우 복잡하고, 효율성도 그다지 좋은 결과를 보이지 못하며, 시스템의 특징이 변함에 따라 그 수정이 매우 힘들다는 단점을 가지고 있기 때문이다.

Davis가 제안한 협상모형을 이용한 연구들은 비교적 현실에 잘 적용할 수 있도록 설계되어 있으나, 기존의 연구에서 협상에 참여하는 객체들은 기계에 한정되었다[Davis et al, 1983].

본 연구에서는 시스템내의 모든 구성요소를 객체로써 표현하고 이들이 서로 자치적으로 자신의 목적성취를 위해 행동하도록 하면서 Bidding mechanism을 통하여 시스템 전체적인 목적을 자연스럽게 성취할 수 있도록 하고자 한다. 객체지향기법을 사용함으로써 개념적으로 이해하기 쉬워지고 타 시스템에의 적용시에도 적은 노력으로 쉽게 이식이 가능해지도록 하였다.

2. 연구현황

생산현장 통제문제는 크게 전통적인 기법과 인공지능 응용기법으로 나누어 볼 수 있다. 이 중 전통적인 기법은 off-line 일정계획기법, on-line 일정계획 및 통제기법, multi-pass 일정계획 및 통제기법등으로 구분하였다.

Off-line 일정계획기법은 월, 일 혹은 시간 단위로 설정된 계획기간동안에 대하여 기간초에 일정계획을 세우고 생산현장에서는 그 계획에 따라 작업을 수행해 나가는 방법이다. 따라서 이 방법은 실시간 반응에 대한 요구가 없기 때문에 수리적 방법 등을 사용하여 계획기간 전체를 고려한 정확한 해를 구해낼 수 있으나, 실제 진행과 계획간에 큰 차이가 생기거나 계획에 포함되지 않은 이상상황(기계의 고장, 자원 부족 등)의 발생시에는 일정계획대로 작업을 수행할 수가 없게 된다. 이러한 상황에서는 재일정계획이 필수적이며 이에 소요되는 시간이 매우 크게 되어 그 동안 비효율적인 작업수행이 이루어지는 단점을 가지고 있다.

On-line 일정계획 및 통제 기법은 Dispatching rule이나 Heuristic을 이용하여 생산현장에서 실시간으로 의사결정을 이루어 나가는 방법이다. On-line 일정계획 및 통제 기법에 관한 연구는 새로운 rule의 개발, 기존의 rule을 다양한 상황에 시험해보는 Simulation 연구, 그리고 현장의 상태변화에 따라 적절히 rule을 변경시켜 주는 연구등이 있다.

Blackstone은 Job shop에서의 Dispatching Rule 34개를 소개하고 다음과 같이 4가지로 분류하였다[Blackstone, 1982].

- ① Processing time을 포함한 Rule
- ② Due date를 포함한 Rule
- ③ 위의 어느 것도 포함하지 않은 Rule
- ④ 혼합 Rule

이러한 다양한 Dispatching Rule(이하 약칭 DR)에 관한 simulation연구가 많은 학자들에 의해 수행되었고, 이로부터 다음과 같은 결론을 얻을 수 있다.

- ①. Performance measure나 시스템 구성의 변화에 따라 DR의 Performance가 변화하며 어떤 rule도 항상 최상의 해를 제공하지는 못한다.
- ②. SPT를 기반으로 하는 Rule들은 평균대기시간을 최소화하고 LPT를 기반으로 하는 Rule들은 기계 Utilization을 최대화한다.
- ③. Due date를 기반으로 하는 rule들은 Processing time을 기반으로 하는 rule보다 job lateness의 분산이 작다.

④. Due date를 정하지 않거나, 매우 tight하게 due date를 설정하는 경우(총 가공시간의 6배 이하), 또는 due date는 loose하나(총 가공시간의 7배 이상) 시스템의 복잡도가 클 경우(기계 utilization이 95%에 가까울 때)는 SPT가 due date를 기반으로 하는 rule보다 좋다.

Heuristic은 DR보다 더 다양한 정보를 이용하여 일정계획을 수행하기 위한 것으로 일반적으로 단순한 DR보다 좋은 결과를 보이거나 계산시간이 길어지고 구현상에 어려움을 가져온다.

이러한 On-line 일정계획기법은 Off-line 일정계획기법에 비하여 생산현장의 동적인 변화에 잘 대응하는 반면, Off-line 일정계획기법과 같은 look-ahead특성을 가지지 못하고 myopic한 해를 제공하는 단점을 가지고 있다.

Multi-pass 일정계획 기법은 일정계획 기간동안 선택된 하나의 DR에 의해 일정계획 및 통제를 행하지 않고 현장의 상태에 최적의 rule을 선택하여 통제를 수행하고 상태 변화에 따라 통제 rule을 변화시켜 주는 기법을 말한다. MPECS는 짧은 시간마다 deterministic simulation을 통하여 현재에 가장 적절한 DR을 선택하여 현장을 통제한다. 이 시스템의 핵심 요소는 다음과 같다[Wu, 1987].

- ①. 실시간 현장 정보와 일정계획 지식을 기반으로 일정계획 대안들을 제시하는 Expert 시스템
- ②. 제안된 대안들의 performance를 비교하는 Simulation model
- ③. Simulation의 결과를 분석하고 최적의 rule을 선정하는 Decision structure
- ④. 선정된 rule을 현장에 적용하는 Controller

Chandra는 DR의 단점을 다음과 같이 들었다[Chandra et al, 1991].

- ①. DR은 한가지 목적함수를 가진다.
- ②. Shop-status의 변동에 따라 여러 목적함수들간의 상대적인 중요도의 변동을 고려하지 못한다.
- ③. DR은 인간의 추론과정과 다르다.
- ④. Alternative routing이 고려된 상황에서는 DR은 적합하지 못하다.

이와 같은 DR의 단점을 극복하기 위하여 Opportunistic reasoning 방법을 사용하여 현장 상태 변화에 따라 Work progress rate의 최대화와 Tardy job 수의 최소화의 두가지 목적을 Adaptive하게 바꿔주는 시스템을 제안하였다.

Nakasuka는 LADS(Learning-aided dynamic scheduler)라는 시스템을 구축하였는데 이는 simulation을 통한 learning으로 Binary decision tree를 만들어서 현장 상황에 적합한 rule의 선택을 가능하게 하였다[Nakasuka et

al, 1992].

이러한 Multi-pass 알고리즘들은 모두 Single-pass 알고리즘보다 향상된 performance를 나타냄을 보이고 있다.

앞서 서술한 전통적 기법들은 시스템의 내·외부적인 원인에 의한 이상상황의 발생등 생산현장의 동적인 변화에 효과적으로 대응하지 못하는 단점을 가지고 있기 때문에, 이러한 문제를 해결하기 위하여 인공지능기법의 활용에 관한 연구가 활발히 진행되었다.

ISIS는 Job shop을 [표 2-1]과 같이 5가지의 constraint들이 모여 있는 것으로 보고, 이러한 constraint들을 만족시키며 생산일정계획을 수립하기 위하여 hierarchical constraint-directed search방법을 사용하였다 [Fox, 1984].

[표 2-1] 일정계획 제약조건

Organizational goals	Due date Work-in-process Shop stability Shifts Cost Productivity goals Quality
Physical constraints	Machine physical constraints Set-up times Processing time Quality
Causal restrictions	Operation alternatives Machine alternatives Tool requirements Personnel requirements Inter-operation transfer times
Availability constraints	Resource reservations Machine down time Shifts Maintenance
Preference constraints	Operation preferences Machine preferences Sequence preferences

OPIS는 shop의 상태를 다양한 관점에서 파악하여 그 상태에서 가장 적절한 생산일정계획 방법을 사용하도록 하기 위하여 multiple problem decomposition방법을 사용하였다[Ow, 1986].

Sarin은 ISIS나 OPIS는 Predictive 시스템이므로 동적인 상황의 대응에 구조적 취약점을 가지고 있음을 지적하고, 문제를 유형별로 분류하는 방법(Problem Categorization)을 이용한 Reactive 시스템을 제안하였다 [Sarin et al, 1990].

생산 시스템의 지리적 분산, Bottom-up 방식으로 진행되어 온 개별자동화에 의한 자동화의 섬에 의한 풍부한

computing resource, 분산 시스템의 robustness등이 분산형 생산일정계획의 개발을 유도하였다.

YAMS는 객체지향형 시스템으로 공장을 상위 level의 Workcell과 하위 level의 Workstation으로 계층적으로 모형화하였다. 각 Workstation 또는 Workcell간에는 Davis의 Contract Net 모형을 통한 협상(Negotiation)을 이용한 Message 전달을 통하여 생산일정계획이 이루어진다 [Parunak, 1986].

Shaw는 Cellular FMS에서 신뢰성, 확장성, 성능(Performance), 비용효율(Cost-effectiveness) 등의 측면에서 분산형 생산일정계획이 집중형 생산일정계획보다 유리함을 서술하고 Bidding mechanism을 이용한 Cooperative 생산일정계획 시스템을 제안하였다([Shaw, 1988], [Shaw, 1989]).

이러한 분산형 생산일정계획 기법들은 여러 Agent들간의 Bidding에 의해 문제를 해결함으로써 보다 유연하게 환경에 적용할 수 있게 되었으나 이들은 모두 기계들에게만 독립적으로 지능을 부여하였을 뿐 다른 객체들은 이에 부속되어 기계의 결정에 따라 움직일 수 밖에 없게 되어 있다. 따라서 이상상황에 보다 능동적으로 적용하기 위해서는 시스템내의 각종 객체들에게 지능을 부여하여 이들이 모두 의사결정에 참여하게 하여야 한다.

3. 분산형 인공지능 시스템

분산형 인공지능 시스템은 문제의 이론적 구조와 Domain의 실제적인 구조의 두 이유로 발생한다.

첫째, 이론적으로 볼 때, 어떤 문제들은 순차적 알고리즘보다는 분산형 framework으로 파악할 때, 보다 쉽게 표현된다.

둘째, 실제적으로 볼 때는, 통신상의 문제로 인하여 문제를 해결하기 위하여 필요한 모든 정보를 한 곳으로 모으는 것이 불가능할 수 있다.

분산형 인공지능 시스템에 대한 동기는 다음과 같은 곳에서 찾을 수 있다.

- ①. 생산 설비의 지리적 분산
- ②. 분산형 설계에 의하여 한 부분의 고장으로 인하여 전체 시스템이 down되는 것을 막을 수 있다.
- ③. 점차적으로 진행되어 온 bottom-up방식의 자동화로 인한 자동화의 섬으로 인하여 분산형 시스템을 구축하기에 충분한 computing resource가 존재한다.

DIA(Distributed Intelligent Agents)는 개별적으로 지능적인 node들간의 상호작용에 의하여 시스템이 구성되나 그 결과로 시스템이 지능적이 되는지 여부는 알 수가 없

다. 반면 CIA(Composite Intelligent Agents)는 node들의 지능성 여부에는 관계없이 그들의 상호작용에 의한 전체 시스템이 지능적으로 행동하도록 하는데 중점을 두고 있다.

대부분의 분산형 인공지능 시스템의 구축에는 이러한 DIA와 CIA의 상호작용을 포함하고 있으며, 이로 인하여 하나의 지능형 agent에 의한 상태보다 각기 지능적인 agent에 의한 전체적인 지능형 시스템 구성을 시도하고 있다.

본 논문에서는 FMS의 모든 구성요소를 객체로 표현하고, 이들이 FMS의 운영을 위한 각종 의사결정과정에 참여하게 함으로써 이들 각기 분산형 인공지능 시스템의 agent로서의 역할을 수행하게 하고자 하며, 이들 각 agent들간의 의사결정수단으로써 Bidding mechanism을 사용하였다.

4. Bidding Mechanism

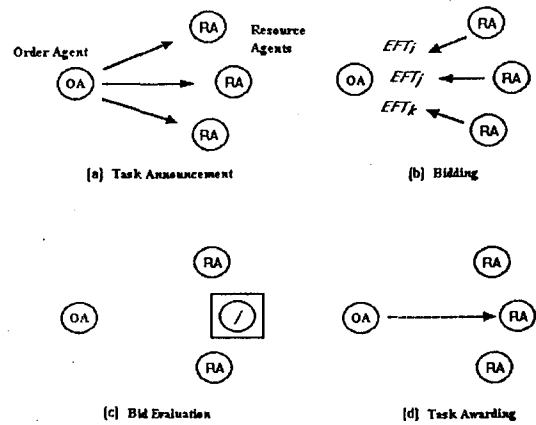
Bidding Mechanism은 Davis[Davis et al, 1983]의 협상 모형에서 최초로 제시된 후 여러 학자들에 의하여 응용되어 왔다. 특히 이 기법은 FMS와 같이 다양한 Resource 들간의 Interaction에 의하여 작업이 수행되는 환경에서의 의사결정을 위한 도구로서 매우 적합하다. Bidding Mechanism은 크게 다음과 같은 3단계로 이루어진다([그림 4-1] 참조).

- ①. Task Announcement : 새로운 job이 발생하거나 어떤 job이 operation을 마쳤을 때, order agent(이하 OA)는 resource agent(이하 RA)에게 task를 announce한다.
- ②. Bidding : Task announce를 받은 RA는 그 task를 평가하고 그 때의 자신의 state에 따라 OA에게 bid를 낸다. Bid에는 (1) estimated processing time, (2) estimated waiting time, (3) estimated travel time 등이 포함된다.
- ③. Bid Evaluation and Task Awarding : OA는 여러 RA에게서 받은 bid를 특정한 목적에 따라 평가한 후 task 수행에 가장 적합하다고 평가된 bid를 제시한 RA(예를 들면, 가장 빨리 task를 수행할 수 있는 RA)에게 task awarding을 한다.

5. 자치적 객체들간의 Bidding

먼저 본 연구에서 사용된 기호를 정리하면 [표 5-1]과 같고, Message를 정리하면 [표 5-2]와 같다.

기존의 Bidding 전략을 이용한 scheduling 및 control scheme들은 모두 한 방향에서 의사결정이 수행되기 때문



[그림 4-1] 일반적 Bidding 과정

에 다른 한 쪽은 단지 정보 처리를 수행하는 역할에 머물러 있다. 따라서 기존의 DR들에 비하여 많은 통신 overhead를 가지면서도 의사결정의 질은 특별히 좋아지지 않는 양상을 나타내고 있다.

[표 5-1] 기호 정리

기호	해설
i	Machine Index
j	Part Index
CT	Current Time
l(j)	Location of Part j
EFT	Expected Finishing time
ETT	Expected Travelling Time
PT	Processing Time
APT	Average Processing Time
CPT	Cumulative Processing Time
Dj	Due Date of Part j
RPTj	Remaining Processing Time of Part j
a_j	Priority of Part j

[표 5-2] Message 요약

Message Type	Information
Task Request Message	MA Id, Status, EFTi
Task Announce Message	PA Id, Status, EFTj
Machine List	(MA Id, Status, EFTi, Utilization), ..
Part List	(PA Id, Status, EFTj, Criticality), ..
Machine Bid	(MA Id, Part_PR _{ij}), ..
Part Bid	(PA Id, Mc_PR _{ij}), ..

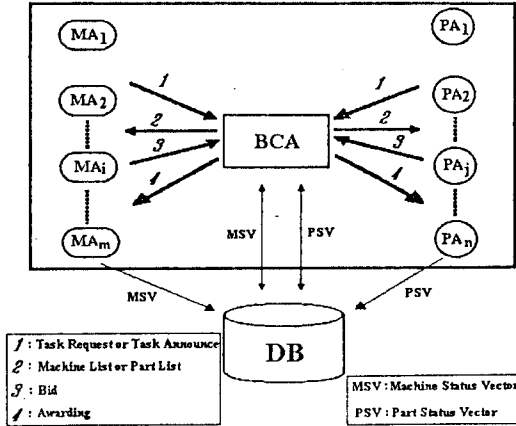
이러한 단점을 극복하기 위하여는 두가지 agent(RA, OA)들이 모두 자신의 state변화에 따라 능동적으로 시스템의 각종 정보를 이용하여 의사결정을 함으로써 통신 overhead에 의한 부담을 극복할 수 있는 보다 좋은 performance를 제공할 수 있어야 한다.

본 연구에서는 다음과 같은 것들을 가정하고 있다.

- 생산계획이 되어 있다.

- Routing sequence를 고려한다.
- Alternative machine을 고려한다.
- Alternative operation을 고려한다.

본 연구에 구현된 핵심 agent들은 Machine Agent(이하 MA), Part Agent(이하 PA)와 Bid Control Agent(이하 BCA)로써 이들간의 Bidding과정을 통하여 FMS 생산현장통제가 이루어진다((그림 5-1] 참조).



[그림 5-1] 자치적 객체간의 Bidding 과정

5-1. Machine Agents(MA)

시스템내에는 각 machine들마다 MA가 하나씩 존재하게 된다. MA는 operation을 수행하면서 데이터베이스에 자신의 status 변화를 기록하며, 자신의 utilization을 극대화하는 것을 목적으로 한다. 각 MA는 자신에게 할당된 job이 없으면 BCA에 Task Request Message를 보낸다. BCA로부터 Part List가 돌아오면 이 정보를 근거로 하여 각 part들에 대하여 다음과 같이 선호도를 계산하여 BCA에게 Bid를 낸다.

먼저, 자신의 Utilization Factor(UF_i)와 Part Preference(Part_PR_{ij})를 계산한다.

$$UF(i) = \frac{\text{Utilization of Machine } i}{\text{Average Utilization of Machines}}$$

$$\text{Part_PR}_{ij} = (\text{CPT}_{ij} - (\text{CT} - \text{EFT}_{i(j)} - \text{ETT}_{i(j)})) / UF_i$$

여기서, Part_PR_{ij}는 part j의 기계 i에 대한 선호도, CPT_{ij}는 part j의 기계 i에서의 누적 가공시간, CT는 현재 시간, EFT_{i(j)}는 기대 완료 시간, ETT_{i(j)}는 기대 운송 시간을 의미한다.

만약 EFT나 ETT가 크면 그 job을 수행하기 위하여 많은 시간을 기다려야 함으로 utilization을 감소시킴을 의미하며, CPT가 크면 대상 job의 processing 시간이 길어

서 utilization이 커지게 된다. 또한 UF_i로 나누어 줌으로 인하여 다른 machine보다 utilization이 낮은 machine이 상대적으로 큰 선호도를 갖게 되며, 다른 machine보다 utilization이 높은 machine은 상대적으로 작은 선호도를 갖게 된다.

이 때, Part_PR가 음수이면 이 part는 machine에 할당 되었을 때, 실질적인 operation 시간 보다 대기시간이 더 큼으로 이러한 part들은 low preference set에 넣어주게 되며 기본적으로 high preference part들 중에서 선택하게 된다. 그러나 만약 high preference set이 공집합이면 이 때는 low preference set에 있는 part들 중에서 선택하게 된다.

5-2 Part Agents(PA)

시스템내에는 각 job들마다 PA가 하나씩 존재하게 된다. 새로운 job이 발생하면 그에 따라 PA가 증가하고, job이 끝나면 소멸하게 된다. PA들은 operation이 수행됨에 따라 데이터베이스에 자신의 status 변화를 기록하며, due date를 지키면서 가능한 한 빠른 시간내에 공정을 끝마치고자 한다.

PA도 역시 idle 상태가 되면 BCA에 Task Announce Message를 보낸다. BCA로부터 Machine List가 돌아오면 이 정보를 근거로 하여 각 machine들에 대하여 다음과 같이 선호도를 계산하여 BCA에게 Bid를 낸다.

먼저, 자신의 Delayness(DE_j)를 계산한다.

$$DE_j = \frac{\text{Delay of Part } j}{\text{Average Delay of Parts}}$$

여기서 Delay란 각 part가 다음 operation의 수행을 위하여 얼마나 계속하여 Queue에서 대기하고 있었는가를 나타내는 것이며, DE_j는 상대적인 Delay정도를 의미한다.

한편, 각 part들은 다음과 같이 Criticality(CR_j)를 계산한다.

$$CR_j = ((\text{RPT}_j \times \alpha_j) - (D_j - \text{CT})) * DE_j$$

여기서, RPT_j는 part j의 잔여 가공 시간, α_j는 part j의 우선순위, D_j는 part j의 due date, CT는 현재 시간을 의미한다.

만약 CR > 0 이면 그 part는 due date에 가까워진 것(Critical Part)을 나타내므로 이러한 part들에 대하여는 보다 많은 관심을 가져야 하며, CR < 0 이면 아직 여유가 있는 part(Non-Critical Part)임을 나타낸다. 여기서 α는, part의 Criticality를 결정하는 중요한 변수로 작용한다. 정의된 Criticality를 바탕으로 Part의 각 machine에 대한 선호도가 계산된다. 여기서, MC_PR_{ij}는 기계 i의 part j에 대한 선호도이다.

$$MC_PR_{ij} = ((CT-EFT_i-ETT_{(i,j)}) + (APT_j/PT_{ij})) \cdot CR_j$$

윗 식의 해석은 Part_PR과 마찬가지로이고, 여기서 APT_j는 job을 수행가능한 모든 기계에서의 평균 가공 시간으로써 이를 대상기계에서의 가공시간으로 나누어 줌으로써 다른 기계들에 대한 상대적 선호도를 크게 혹은 작게 해 줄 수 있게 된다.

5-3 Bid Control Agent(BCA)

BCA는 각 agent들에게서 Task Request 혹은 Task Announce Message를 받으면 그에 해당하는 Part List나 Machine List를 agent에게 돌려주는 역할을 한다. 그 후 각 agent로부터 bid를 받아서 Matching Process를 통하여 적절한 MA와 PA의 짝을 선정하는 일을 한다.

먼저, 각 MA와 PA들간의 상호 Preference(PR_{ij})를 계산한다.

$$PR_{ij} = \beta \times Part_PR_{ij} + (1 - \beta) \times Mc_PR_{ij} \quad \forall i, j$$

여기서, PR_{ij}는 기계 i와 part j간의 상호 선호도, β는 두 선호도에 대한 가중 상수으로써, 이를 조정함에 따라서 두 가지 측면에서의 선호도가 전체 선호도에 미치는 중요도를 조절할 수 있다.

따라서, 전체 시스템의 목적인 상호 선호도가 큰 MA와 PA의 짝을 선정하는 문제는 다음과 같은 수리계획문제로 모형화할 수 있다.

$$\begin{aligned} \text{Max} \quad & \sum_i \sum_j X_{ij} \cdot PR_{ij} \\ \text{s.t.} \quad & \sum_j X_{ij} \leq 1 \quad \forall i \\ & \sum_i X_{ij} = 1 \quad \forall j \\ & X_{ij} = 0, 1 \quad \forall i, j \end{aligned}$$

여기서 X_{ij}가 1 이면 기계 i에 part j가 할당됨을 의미한다.

위와 같이 표현된 배정문제를 풀어서 그에 따라 선택된 job을 machine에 할당하여 주게 되며, machine utilization이 낮은 MA나 criticality가 큰 PA가 존재하고 그들이 일차 할당에서 job을 할당받지 못한 경우에는 그들을 할당받아 줄 수 있는 MA나 PA가 있는지를 알아보고 어떤 MA나 PA가 동의하면(즉, 사전에 할당된 job 대신에 새로운 job을 수행하고자 하는 경우) 그들을 다시 재할당해 준다. 이와 같은 과정을 통하여 다음 job이 할당된 MA와

PA에는 Assigned 신호를 보내고, 그렇지 못한 agent들에게는 Rejected 신호를 보낸다. 그리고 이 과정을 모든 MA에 다음 job이 할당될 때 까지 계속한다.

6. 실험결과

제안된 FMS 생산현장통제 기법의 효율성 검증을 위하여 다음과 같은 FMS에 대하여 시뮬레이션을 수행하였다 (30회씩의 실험을 수행하였다.).

- ① 6대의 동일한 NC 공작기계
- ② 4가지 종류의 Parts
- ③ Alternative routing을 허용
- ④ Shop에서 수행가능한 operation의 종류 : 20 operation
- ⑤ Part의 operation 순서 유지
- ⑥ 기계 i에서의 part type j의 operation k의 가공시간 : Uniform
- ⑦ 각 part의 도착시간 : Exponential
- ⑧ 각 part의 due date : Uniform

실험결과는 [표 6-1], [표 6-2], [표 6-3]에 요약되어 있다.

[표 6-1] Flow time 비교

part rule	1	2	3	4
SPT	279.29	505.29	357.50	432.78
EDD	1420.74	2775.86	5888.45	5278.50
Bidding	251.29	386.04	339.02	402.80

[표 6-2] Earliness 비교

part rule	1	2	3	4
SPT	1214.52	918.52	1913.76	1923.73
EDD	*	*	*	*
Bidding	1319.53	1130.18	1971.96	2015.15

*는 발생없음

위의 결과를 보면, 제안된 Bidding 기법을 사용하였을 때, 모든 part에 대하여 시스템내 체류시간이 가장 짧고, slack 시간은 SPT보다 약간 크며, 기계의 utilization은 고르게 분산됨을 알 수 있다.

[표 6-3] 기계 가동률 비교

part rule	1	2	3	4	5	6
SPT	.38	.56	.44	.49	.54	.21
EDD	1.0	1.0	1.0	1.0	1.0	1.0
Bidding	.39	.46	.45	.45	.44	.37

7. 결론 및 추후연구방향

앞에서 본 바와 같이, FMS 생산현장통제를 위하여 자치적 객체들간의 Bidding을 적용할 때, 개념적으로 이해하기 쉬우며, 시스템의 구조가 변하거나 심지어는 전혀 다른 시스템에 적용할 때에도 그 구현이 매우 쉽다는 장점을 가지고 있다. 또한 제안된 기법을 적용할 때에는 시스템의 동적인 변화에 대하여 각 객체들이 능동적이면서 적절한 판단으로 전체적인 시스템의 효율을 증대시키는 양태를 보인다.

반면, 제안된 시스템에서는 대부분의 정보를 각 객체들이 가지고 있기 때문에, 전체 시스템 수준에서 필요한 정보를 얻기 위하여는 그 정보를 가지고 있는 객체와의 통신이 필요하게 된다. 즉 통신에 의한 시간지연이 발생할 가능성을 가지고 있다. 이러한 시간지연은 제안된 시스템의 특성인 시의적절한 의사결정이라는 특징을 감소시킬 가능성을 의미한다. 그러나 급속한 하드웨어의 발전을 고려할 때, 이러한 통신상의 시간지연은 제안된 시스템의 장점과 비교할 때, 간과할 수 있을 정도의 것이라고 판단된다.

본 연구에서 제안된 자치적 객체들간의 Bidding 기법에서는 독자적으로 움직이는 각 객체들이 자신의 상태변화와 타 객체들의 상태변화를 감지하도록 하면서 다른 객체들의 움직임에는 간섭을 하지 않도록 되어야 한다. 따라서 추후에 Multi-tasking이 가능한 운영체제하에서 시스템의 특성을 충분히 표현하여 시뮬레이션을 하는 것이 필요로 되고 있으며, 이 때에는 현재의 실험결과에 비하여 보다 정확한 특성치 분석이 가능하게 될 수 있으며, 제안된 시스템의 장점이 명확히 나타날 수 있으리라고 기대된다.

8. 참고문헌

Blackstone, J. H., Phillips, D. T., and Hogg, G., 1982, A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations, *International Journal of Production Research*, Vol. 20, No. 1, 27-45.

Chandra, J., and Talavage, J., 1991, Intelligent Dispatching for Flexible Manufacturing, *International Journal of Production Research*, Vol. 29, No. 11, 2259-2278.

Davis, R., and Smith, R. G., 1983, Negotiation as a Metaphor for Distributed Problem Solving, *Artificial Intelligence*, Vol. 20, No. 1, 63-109.

Fox, M. J., 1984, ISIS - A Knowledge-based System for Factory Scheduling, Vol. 1, No. 1, 25-49.

Nakusaka, S., and Yoshida, T., 1992, Dynamic Scheduling System Utilizing Machine Learning as a Knowledge Acquisition Tool, *International Journal of Production Research*, Vol. 30, No. 2, 411-431

Ow, P. S., Smith, S. F., and Fox, M. S., 1986, Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-based Factory Scheduling Systems, *AI magazine*, FALL.

Parunak, H. V. D., 1986, Distributed Artificial Intelligence Systems, in: A. Kusiak (ed.), *Artificial Intelligence: Implications for CIM*, IFS (Publications) and Springer-Verlag, 225-251.

Sarin, S. C., and Salgame, R. R., 1990, Development of a Knowledge-based System for Dynamic Scheduling, *International Journal of Production Research*, Vol. 28, No. 8.

Shaw, M. J., 1988, Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making, *Journal of Manufacturing Systems*, Vol. 7, No. 2, 83-94.

Shaw, M. J., 1989, FMS Scheduling as Cooperative Problem Solving, *Annals of Operations Research*, Vol. 17, 323-346.

Wu, S-Y. D., 1987, An Expert System Approach for the Control and Scheduling of Flexible Manufacturing Cells, *Ph. D. Thesis*, The Pennsylvania State University.