

실시간 일정 계획을 위한 지식 획득용 Intelligent Simulator의 개발

김기태*, 이상복*, 박찬권**, 박진우*

* 서울대학교 산업공학과
** 영산 국제산업대학교 경영정보학과

Abstract

FMS를 포함한 다양한 형태의 생산 시스템에 있어서 일정계획문제는, 고가의 생산설비를 효율적으로 운영한다는 의미에서 점점 더 중요해지고 있다. 일정계획문제를 해결하기 위하여 많은 방법들이 제시되었고 지금 현재도 다양한 방법들이 시도되고 있다. 이런 많은 방법중에서 최적해를 구하는 수리계획법이 가장 우수한 대안을 제시하는 것으로 알려져 있지만, 최적해를 구하는 데 상당히 많은 시간이 필요한 것으로 알려져 있다. 다음으로 우선 순위 규칙을 제공하는 방법은 짧은 시간을 이용하여 어느 정도 효율적인 일정계획을 제시하지만, 생산 시스템의 구성과 상황에 따라서 가장 적합한 우선순위 규칙이 변한다고 알려져 있다. 이런 상황에 대응하여 시스템의 구성과 상황에 가장 적합한 규칙을 제안하는 전문가시스템들이 연구되고 있다. 본 연구는 일정계획을 실시간에 수립하는 스케줄러를 위한 지식을 획득하는 Intelligent Simulator개발에 관한 연구이다.

일정계획을 위한 인공지능 스케줄러를 구축하는 데 있어 가장 큰 난관은 지식을 어떻게 획득할 것인가 하는 점이다. 이런 지식획득의 방법으로 주로 사용되는 방법은 인간 전문가의 지식을 추출하는 방법이 있다. 그러나 인간 전문가가 없거나 인간전문가의 지식으로는 부족한 경우에는 기계학습을 통하여 지식을 추출한다. 기계학습을 위하여는 반드시 많은 case들을 통하여 지식을 획득하게 되는 데 이런 case의 축적을 위하여는 실제의 시스템을 운영해본다는 것은 많은 비용을 필요로 한다. 이때 필요한 경험의 생성을 위하여 시뮬레이션은 그 중요한 효용성이 있어서 많은 연구자들이 사용하였다. 그러나 경험을 다 생성한 후 그 경험으로부터 지식을 획득하다보니 서로 다른 방법의 사용에 의하여 동일한 경험으로부터 서로 다른 지식이 생성되는 가능성이 있다. 본 논문은 시뮬레이션이 계속 수행되는 상황에서 일정계획에 관련된 지식을 획득하여 축적할 수 있는 Intelligent Simulator의 개발에 관한 연구이다.

주요어 : 일정계획, 지식획득, 시뮬레이션, 시뮬레이터

1. 연구의 목적

일정 계획을 위한 인공지능 기반형 스케줄러를 구축하는 데 있어 가장 큰 난관은 지식을 어떻게 획득할 것인가 하는 점이다. 이런 지식획득을 위해 주로 사용되고 있는 방법은 첫째, 지식을 갖고 있는 인간 전문가들의 지식을 추출하여 규칙 베이스(RuleBase) 형태로 지식을 구축하는 방법이다. 그러나 인간 전문가가 존재하지 않거나 인간 전문가의 지식으로도 해결할 수 없는 환경에 대하여는 다른 방법이 필요하다. 즉 기계 학습(Machine Learning)이나 추론(Induction) 등의 방법을 이용한다. 일

정 계획의 문제에 대하여는 주로 Neural Net을 훈련시키는 방법이 많이 사용되고 있다. Neural Net을 사용하든지, 아니면 다른 방법을 이용하든지 기계 학습을 위하여는 반드시 실제의 경험과 같은 자료들이 필요하다. 즉 지식을 획득하고자 하는 환경에 대하여 과거의 운영 경험과 같은 다양한 경험 지식이 필요하다. 그러나 이런 경험을 축적하는 데에 많은 시간과 노력이 필요할 뿐만 아니라 자료 중에 오류가 포함될 가능성도 배제할 수 없다. 그래서 일정 계획을 위한 기존의 지식 기반형 스케줄러는 대부분 대상으로 하는 생산 시스템의 시뮬레이션 모

형을 구축하고, 이 시뮬레이션의 결과를 경험으로 하여 지식을 획득하였다. 그러나 시뮬레이션이 다 수행된 후, 그 결과를 분석하기 때문에 시뮬레이션 수행 중에 발생하는 복잡다양한 상태를 반영하지 못하는 한계가 있었다. 본 논문은 시뮬레이션이 계속 수행되는 상황하에서 일정 계획에 관련된 지식을 지속적으로 획득하여 축적할 수 있는 Intelligent Simulator의 개발이 그 목적이다.

기존의 시뮬레이션 방법들은 다양한 환경에 대하여 시뮬레이션 프로그램을 작성하여 각 대안별로 생산시스템의 성능을 비교하였다. 생산시스템의 환경 즉 다양한 공장의 구성 형태에 대한 대안이 제시되고 비교되었다. 그러나 한 공장의 구성형태라고 하더라도 작업 주문의 특성 - 생산해야할 제품 및 그 제품의 공정계획의 특성 - 에 따라서 그 수행도에 차이가 발생한다. 생산해야할 제품은 MRP 환경에서 MPS로 계획되고, 제품의 공정계획은 BOM에 의하여 결정된다. 그러므로 생산시스템의 보다 유용한 시뮬레이션을 위해서는 MRP환경의 데이터베이스 정보를 공유할 필요가 있다. 기존의 시뮬레이터들도 다양한 형태의 데이터 파일 혹은 데이터베이스 파일과 정보를 공유하지만 보다 긴밀하고 보다 용이하게 정보를 공유하는 시뮬레이터가 필요하다. 본 논문은 계속 변화하고 있는 생산 시스템의 다양한 자료에 대하여 즉시적으로 시뮬레이션을 실행하면서 해당 시스템의 상태에 가장 적응적으로 운영되도록 하는 지식을 제공하는 Intelligent Simulator를 개발에 관한 연구이다.

2. 현황 분석

2.1 일정 계획 문제

일정 계획에 대하여 전문가 시스템 혹은 Neural Net 등의 인공지능방법을 적용하고자 한 연구는 다양한 방향으로 접근되어 오고 있다. Kusiak은 생산시스템의 일정계획에 대한 전문가 시스템의 응용에 대하여 20여종류의 접근 방법에 대하여 비교 분석하였다[KUSI, 1988]. 그리고, 일정 계획 문제의 해결을 위한 전문가시스템에 필요한 Knowledge Base를 획득하기 위하여 시뮬레이션 분석이 부가적으로 수행되어진 연구도 대두되었다.

Pierravel은 FMS일정 계획 문제를 풀기 위하여 주어진 FMS에 대한 시뮬레이션 수행을 통하여 Knowledge Base를 구축하였다. 동일한 상황에 대하여 다양한 Priority Rule을 적용하여 그 결과들을 상호 비교하여 가장 적합하게 맞아떨어지는 Priority Rule을 제안하도록 하는 Knowledge Base를 구축하였다.

Job Shop Scheduling문제를 해결하기 위해, Grabot은 시뮬레이션을 이용하여 Dispatching Rule 별로 시뮬레이션 결과를 비교하여 가장 적합한 Rule을 다수 결정하여 저장한다. 실제 적용 시에는, 주어진 수행도 기준과 시스템의 특징에 대하여 Fuzzy Approach를 이용하여 가장 우수한 Dispatching Rule을 찾아내어 적용하였다[GRAB, 1994].

2.2 시뮬레이션 시스템의 현황

컴퓨터 시뮬레이션을 위한 시뮬레이션 시스템의 개발은 많은 발전이 이루어지고 있다. 컴퓨터 시뮬레이션 시스템을 특징에 따라 Linguistic Modelling, Parametrical Modelling, Object-Oriented Modelling으로 분류되어 질 수 있으며, 현재는 Object-Oriented Modelling의 방법을 도입하려는 경향이 있다.

시뮬레이션 시스템의 구축에 있어서는 시뮬레이션의 내부를 구축하는 과정에 지식 표현을 위한 특정한 모듈을 시뮬레이션 모형에 첨가하여 지식기반형 시뮬레이션 시스템을 구축하여 오고 있다[ZEIG, 1990]. 이런 현상은 시뮬레이션에 관련된 객체들의 활동 양태를 지식의 형태로 시뮬레이션 프로그램에 첨가해 놓은 형태로 구현되어 오는 경향이 있다. 이런 지식은 시뮬레이션의 수행을 위해서 추가된 것이다.

한편 시뮬레이션 시스템을 단지 시뮬레이션 전용으로만 사용하는 것이 아니라 다른 용도로 전환하여 사용하면서, 시뮬레이션 기능에 특수한 목적의 기능을 추가적으로 가지는 특수 목적 시스템도 제안되고 있다. Smith는 FMS의 제어 시스템을 구축하기 위하여 시뮬레이션 모형을 이용하였다. 시뮬레이션을 수행하는 과정과 실제 시스템의 운영을 대응시키면서, 의사 결정이 필요한 경우마다 시뮬레이션을 직접 수행하여 그 결과에 따라 제어 시스템을 운영하는 방안을 제시하였다. 즉 시뮬레이션 결과에 맞추어서 시스템을 운영하다가 시뮬레이션에서 예견된 상황과 다른 상황이 발생하였을 경우에는 새롭게 시뮬레이션을 적용하고 그 시뮬레이션의 결과를 이용하여 의사결정을 하였다. 그리고 그 결정에 따라서 실제 시스템에 대하여 지시를 내려 운영하였다[SMIT, 1992]. 즉 FMS의 제어 시스템을 시뮬레이션과 제어 시스템의 통합 시스템으로 구축하였다.

2.3 지식 획득 및 지식 표현

지식을 획득하고 또한 그 지식을 표현하는 방법에는 많은 방법이 있다. 이들중 어떤 방법이 시뮬

레이션을 이용한 지식획득 및 지식표현에 가장 적합할 것인 지에 대하여 많은 연구가 이루어졌다.

Yih는 Trace-Driven Knowledge Acquisition for Expert Scheduling System을 구축하여, 시뮬레이션 결과를 이용하여 Markov Chain을 훈련시켰다. 훈련된 Markov Chain을 이용하여 가장 적합한 Dispatching Rule을 실시간으로 제시하도록 하여 실시간으로 일정 계획을 수행하도록 하였다[YIH, 1992]. 그런데 이때 고려하는 시스템 상태를 나타내는 Attribute와 각 Attribute간의 상태전이 확률값인 Transition Probability는 인간 전문가가 선택하여 입력하도록 하였다.

조현보는 일정 계획을 위한 Dispatching Rule을 선택하기 위해 Neural Net을 사용하였다. 다양한 조건에 대하여 시뮬레이션을 수행한 결과를 이용하여 Neural Net을 훈련시킨 후, 훈련된 Neural Net에 실제 시스템의 상황을 입력하여 가장 적합한 Dispatching Rule을 제안하도록 하였다[CHO, 1993]. 이때 Neural Net에서 사용된 Input Layer에서 사용되는 Attribute는 조현보의 판단에 의하여 임의로 필요하다고 판단되는 것을 사용하였다. 실제 상황을 고려해 보면 이렇게 사용된 Input Layer상에 적용된 Attribute가 전체 시스템의 성능 및 상황을 반영하는 요소로서 충분한 지는 계속 검토하여야 할 문제이다.

Yih와 조현보는 Neural Net을 이용하여 시스템 상황에 가장 적합하다고 판단되는 Dispatching Rule을 찾아서 실시간에 일정 계획을 만드는 스케줄러를 제안하였다. 그러나 두 사람은 시스템 상황으로 고려하는 요소를 각각 [표 1]과 같이 선택하였다. 두 사람이 선택한 Attribute는 서로 비슷한 부분도 있지만 서로 다른 부분이 많다. 어떤 요소가 보다 더 중요한 지 혹은 추가적으로 고려되어야 하는 Attribute에는 무엇이 있는 지 현재의 연구 현황으로서는 보다 더 깊은 연구가 필요하다. 무엇보다도 여기서 선택되어진 Attribute들은 Yih와 조현보의 판단에 의하여 선택되어진 것이다. 그래서 이 판단은 사람마다 다를 수 있으며, 시스템의 특징에 따라서 다를 수도 있다.

[표1] Neural Net에서 선택된 Input Layer의 Attribute

Yih	조현보
1. NO. of Jobs in the Queue	1. Routing Complexity
2. NO. of tardiness Jobs	2. Performance Criteria
3. Maximum Tardiness	3. Travel Time
4. Mean Tardiness	4. Processing Time
5. Variance of processing time	5. System Congestion
6. Variance of Setup Time	6. Machine Utilization
	7. Job Late Factor
	8. Queue Status Factor

3.4 시스템 상태 벡터를 이용한 시스템 상태예측
 시스템의 상태를 다양한 Attribute로 이루어진 Tuple로 표현할 때, 어떤 Attribute가 사용되는 것이 가장 좋을 지는 누구도 알 수 없다. 모든 Attribute값의 변화 추이를 관찰해 나가는 과정 중에 중요한 Attribute를 찾아내고 그 Attribute를 중심으로 하여 시스템의 상태를 판단한다. 여기서 판단되는 시스템의 상태는 분석자가 요구하는 다양한 수행도를 기준으로 판단하는 부분이다. 그러나 시스템의 수행이 다 종료되지 않은 상태에서 시스템의 수행도를 결정할 수는 없다. 그 동안에 이루어진 결과들을 이용하여 그 우열을 결정할 수는 있을 것이다. 현재의 상황에 대하여 일정 기간 이후의 미래를 예측하는 방법으로 DPN(Dynamic Probability Network)의 방법이 이용되었다. 주어진 상태에 대한 전이 확률을 이용하여 현재 상황에 대하여 일정 기간 이후의 시스템 상태를 예측하는 방법이 제시되었다.

Forbes는 PPN(Dynamic Probability Network)을 이용하여 현재 상태의 다음 상태를 예측하고 Temporally Invariant Networks와 Stochastic Sampling을 이용하여 실시간 정보를 효과적으로 획득하여 다양한 상황에 대하여 효과적으로 자동차를 운전하는 알고리즘을 제안하였다[FORB, 1995]. 그러나 여기서의 각 상태들은 정성적인 상태정보였다. 이에 대응하여 정량적인 Attribute의 값을 정성적인 값으로 변환시키기 위해 Classification방법이 제시되었다. Attribute의 값들을 기준으로 시스템의 상태를 추정하여 몇 가지의 상태로 나누는 방법으로 다양한 방법들이 연구되었다.

Shaw는 주어진 시스템의 상태를 나타내는 다양한 parameter의 조합을 pattern으로 설정하고 이 pattern의 변화에 대응하여 일정 계획의 실제 수행 형태를 모니터링하는 Pattern-Directed Scheduling방법을 제시하였다. 이 때 Pattern에서 고려되는 요인으로는 System Configuration(버퍼의 크기, 기계 댓수, 가공단계수), Machine Parameter(MTBF), Job Characteristic(Setup Time)주어진 수행도 등이었다[SHAW, 1987].

3. 개념 설계

Intelligent Simulator를 구축을 위한 Simulator의 개념 설계는 두 단계로 구분되어질 수 있다. 먼저 시뮬레이션 프로그램을 구축하기 위한 프로그래밍 환경과 실제의 시뮬레이션 프로그램이 수행되는 환경에 대한 시뮬레이션 환경으로 나뉘어 질 수 있다.

3.1 프로그래밍 환경

Intelligent Simulator 환경은 기존의 모든 시물레이션 프로그램이 가지고 있는 시물레이션 프로그램을 구축하는 Modeller와 MRP 환경의 다양한 Database의 정보를 공유하기 위한 DB Modeller와 시물레이션을 통하여 생성되는 지식을 저장하기 위한 KB Modeller로 구성되어 있다.

Modeller는 생산시스템의 환경을 있는 그대로 모사하는 것으로 생산시스템에 포함되어 있는 기계장비의 관점에서 시물레이션 프로그램을 작성한다. DB Modeller는 생산시스템에서 생산되는 제품 혹은 관리되는 원자재에 대한 정보를 저장하는 Database에 대한 구조를 입력한다. 또한 완제품과 원자재 이외의 재공품에 대한 정보와 생산제품에 대한 MPS 정보와 BOM 정보를 관리하기 위한 기능도 포함된다. KB Modeller는 시물레이션 수행결과 생성되는 지식을 저장하기 위한 형태를 정의하는 기능을 포함한다. 뿐만 아니라 비용 최소화 혹은 납기일 만족 등의 생산시스템의 수행 목적을 명시하고, 분석자가 생각하는 중요한 Parameter를 선택하여 입력시키는 기능을 포함한다.

위의 세가지 Modeller는 시물레이션의 수행을 위한 프로그램과 데이터에 대한 정의를 위한 것이다. 이들은 각 개별 파일로 존재하여서 하나의 프로젝트로 통합되어 진다.

3.2 시물레이션 환경

시물레이션 프로그램이 완성되고 난 후, 실행되기 위해서는 Modeller가 구축하는 시물레이션 환경을 구성하는 여러 가지의 클래스의 멤버들이 각자가 수행해야 할 기능을 원활하게 수행하는 것이 필요하고 이를 위해서 Simulator는 다양한 멤버함수들의 기능이 원활하게 수행되어야 한다. 이를 위하여 다음과 같은 클래스 정의가 이루어졌다.

3.2.1 시물레이션 수행을 위한 클래스

```
class CResource
    CMachine
    CLabor
class Entity
    COrder
    CProduct
    CWIP
    CMaterial
```

CResource class는 시물레이션 시스템 내에서 시간 지연과 관련된 클래스로 각종 기계 장비와 작업자와 같은 Object를 포함한다. 동시에 각 장비와 연관되어지는 대기 행렬에 대한 부분도 포함된다.

CEntity class는 시물레이션 시스템 내에서 계속 옮겨 다니면서 상태가 바뀌면서 Resource에 의하여 작업이 수행되어지는 실체를 표현한다. Entity에는 원자재, WIP과 완제품 뿐만 아니라 제품생산에 대한 생산 주문도 포함된다.

3.2.2 지식 획득을 위한 클래스

```
class Instance
class Categorizer
class Inducer
class CKB
```

class Instance는 시물레이션되고 있는 상황을 계속 추적하면서 변화되는 상황에 대하여 고려해야 할 Attribute를 판단하고 Attribute의 값을 몇 개의 구간으로 나누어 상태정의를 한다.

class Categorizer는 고려되어야 한다고 판단된 Attribute의 값들에 준하여 시스템 내의 상태를 설정하고 판단한다.

class Inducer는 시스템의 상태를 판단하는 다양한 방법을 가지고 있으며, 상태에 대하여 가장 적합하다고 판단되는 대안을 선택한다.

class CKB는 개별 기계장비와 작업자와 같은 class CResource와 class CEntity들의 Object에 대응하여 각 Object가 가지는 지식을 저장하고 관리한다.

3.2.3 기타

```
class CDB
    CBomDB
    CInventoryDB
    CMPSDB
class CMathCal
```

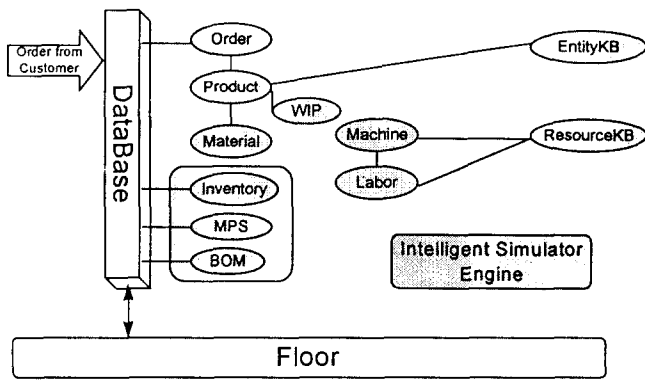
class CDB는 시물레이터가 관리해야 하는 데이터 베이스의 정보를 관리하기 위한 클래스로서, 생산 주문에 대한 MPS 정보와 원자재, 재공품 및 완제품 정보와 개별 생산해야 할 제품의 공정계획을 가지는 BOM 정보를 포함한다.

class CMathCal은 시물레이터가 필요로 하는 각종의 계산을 위한 함수를 포함한다.

위와 같은 클래스들로 구성되어지는 시물레이션 시스템을 그림으로 보면 [그림1]과 같다.

4. 결론

기존의 많은 시물레이션 시스템의 단일 목적성에서 벗어나서 시물레이션을 수행하면 시물레이션한



[그림 1] 클래스 개요도

시스템의 수행도만 결과로 나타나는 것이 아니라, 시뮬레이션한 시스템의 운영상의 지식이 자동적으로 생성되게 된다. 시스템의 수행도 분석을 단순히 수치적으로 판단하게 되는 수준을 넘어서, 보다 우수한 시스템을 만들기 위한 대안의 수립이 시뮬레이션 종료 후에 가능하도록 할 수 있게 한다.

5. 참고 문헌

- [CHO, 1992] Cho, Hyuenbo and Richard A. Wysk, A Robust Adaptive Scheduler for an Intelligent Workstation Controller, Working Paper
- [FORB, 1995] Forbes, Jeff, Tim Huang, Keiji Kanazawa and Stuart Russell, The BATmachine: Towards a Bayesian Automated Taxi, the 14th International Joint Conference on Artificial Intelligence 1995, VOL.2
- [GRAB, 1994] Grabot, B. and L. Geneste, Dispatching rules in Scheduling : a fuzzy approach, International Journal of Production Research, VOL. 32, NO. 4, 1994, pp. 903-915
- [KUSI, 1988] Kusiak, Andrew and Mingyuan Chen, Expert Systems for Planning and scheduling Manufacturing systems, European Journal of Operational Research, VOL. 34, 1988, pp 113-130
- [MOSC, 1996] Mosca, P, P. Giribone and A. G. Bruzzone, Study of the behaviour of flexible production systems using neural nets, Production Planning and Control, VOL.7, NO.5, 1996, pp 462-470
- [PIER, 1992] Pierreval, Henri , Expert System for Selecting Priority Rules in Flexible Manufacturing Systems, Expert Systems with Applications, VOL. 5, pp. 51-57, 1992
- [SMIT] Smith, Jeffrey S. , Design and Implementation of FMS Control S/W, Working Paper

[YIH, 1992] YIH, YUEHWERN, Learning real-time scheduling rules from optimal policy of semi-Markov decision process, International Journal of Computer Integrated Manufacturing , VOL. 5, NO.3, 1992, pp 171-181

[YIH, 1994] Yih, Yuehwern, L.C. Rabelo and A. Jones, A Hybrid AI/Simulation framework for FMS Scheduling, In Proceedings of New Directions in Simulation for Manufacturing and Communication(SIM94), Tokyo, Japan, 1-2 August 1994, pp 204-210

[ZEIG, 1990] Zeigler, Bernard P. , Object-Oriented Simulation with Hierarchical, Modular Models, Academic Press, 1990