Journal of the Korean Institute
of Industrial Engineers
Vol. 18, No.2, Dec., 1992

모의실험을 통한 전문가 시스템+

김 선 욱*

A Simulation-Based Expert System Paradigm+

Sun-Uk Kim*

Abstract

Both simulation and expert systems are popular ways to solve complex and hard problems. However, the results of the simulation, which include a large amount of valuable information as a good knowledge source, are not used efficiently. Furthermore, the development of the expert systems can fail because there is no expert or an expert is not available.

A new Simulation-Based Expert System(SIMBES) paradigm has been constructed to overcome these problems. It consists of simulator, feature extractor, machine learning system, performance evaluator and Knowledge-Based Expert System(KBES).

A SIMBES was implemented for an existing schedule-based MRP system in Smalltalk/V to show how this paradigm works and experimented for a large number of jobs. The KBES and the existing system produced better schedules for 72 percent and 28 percent of the jobs, respectively.

1. 서 론

많은 연구에서 쉽게 찾아 볼 수 있듯이 운용분석(OR)은 산업체에서 가장 많이 사용되는 문제 분석 방법 중의 하나로 널리 알려져 있다. 그 분야는 여러종류의 최적화 방법을 제시해 주는 것이 사실이나 생산, 자재 및 일정관리 분야처럼 문제의 규모가 크고 복잡한 경우 그와 같은 최적화 방법의 시도는 많은 한계점을 갖는다. 예를들면, Job Shop 일정계획 문제는 NP-Hard 문제로 규정되어 있다[16]. 다시말해서 가상적인 Job Shop이 아니라 실제공장에서 일어날 수 있는 규모의 문제를 풀려고 할때, 그 최적화 방법은 포기되지 않을 수 없다.

이러한 유형의 복잡한 문제를 풀기위해 모의실험(Simulation) 방법이 활발하게 이용되고 있으며 그 응용분야 또한 광범위하다. 그 중심이론은 주어

+이 논문은 1991년도 교육부지원 한국학술진흥재단의 자유공모과제 학술연구조성비에 의하여 연구되었음.

* 단국대학교 산업공학과

진 시스템을 모델화하여 컴퓨터에 표현한 것으로서, 그 표현된 모델을 통하여 많은 가능한 상황 또는 변수를 변화시켜 시스템의 행태(Behavior)를 예측하는 수단을 제공한다[11]. 수 많은 횟수로 모의된 상황과 그 결과들은 그 영역의 문제를 풀기 위한 중요한 지식원(Knowledge Source)임에 분명하다. 따라서 이러한 지식들을 축적 내지는 저장시키지 아니하고 일회용으로 끝내는 것은 커다란 손실임에 틀림없다. 특히 시간과 자원이 제약된 상황하에서는 이 주장이 더더욱 설득력을 갖는다. 이런 상황에서 우리는 당연히 어떻게하면 시간과 자원을 낭비하지 않으면서 복잡하고 난이도가 높은 문제일지라도 효율적으로 풀 수 있는가 하는 문제가 대두된다.

또한 문제의 규모가 크고 어려운 문제를 해결하기 위한 방법으로 최근 전문가 시스템(Knowledge-Based Expert System)이 널리 이용되어 많은 발전을 거듭하고 있다. 그러나 일반적으로 전문가 시스템을 개발하기 위해서는 인간 전문가가 존재하여야 하며, 이 전문가로부터 지식을 추출해내는 지식획득(Knowledge Acquisition) 과정은 전문가 시스템 개발시 가장 어려운 단계에 속한다[9].

더구나 산업체가 실제로 직면하는 생산, 자재, 일정계획 문제처럼 규모가 크고 복잡한 문제에서는 전문가가 존재하지 않는 경우도 존재한다. 여기에서 중요한 사실은 물론 각 분야에 전문적 지식(Expertise)이 존재한다는 점이다. 이 존재하는 전문적 지식을 유도하기 위한 방편으로도 시스템의 여러 가능상황에 대한 행태의 관찰 및 결과 수집을 위한 모의실험 개발이 필수적이다. 앞에서와 마찬가지로 이러한 가능상황과 그 행태가 중요한 지식원이 되며, 이로부터 전문가 시스템을 개발하려고 한다.

따라서 본 연구에서는 이러한 필요성에 부응하여 두가지 목적을 갖고 연구를 진행한다. 첫째, 모의실험을 통하여 전문가 시스템을 구축하기 위한 일반적 방법론 또는 구조도(General Framework)를 개발한다. 둘째, Schedule-Based 자재계획 문제에 실제로 이 일반적 구조도를 적용하여 전문가 시스템을 실험적으로 개발하고자 한다.

2. 모의실험을 통한 전문가 시스템 (Simulation-Based Expert System)

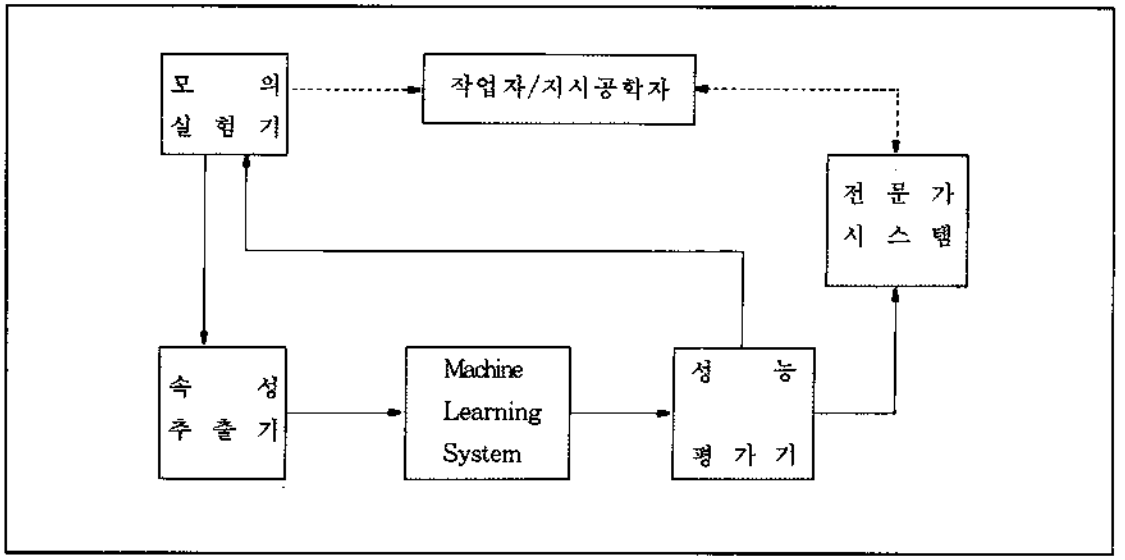
이 모의실험을 통한 전문가 시스템은 SIMBES로 약칭된다. 우선 SIMBES구조 및 각 구성 요소들의 상호 연관 관계를 자세하게 검토한다. 이어서 SIMBES구조의 중요한 구성요소 중의 하나인 Induction 소프트웨어를 소개한다.

2.1 SIMBES 구성

SIMBES는 [그림 1]에 보여지는 바와 같이 크게 두가지 부류로 나눌 수 있다. 첫번째 가능한 구조는 모의실험기(Simulator), 작업자 또는 지식공학자와 전문가 시스템으로 구성될 수 있다. 이 구조도는 작업자를 많은 모의실험 문제와 결과에 노출시켜 훈련시킴으로써 축적된 지식을 전문가 시스템의 지식베이스로 전이시킨다는 것을 주요 골격으로 한다. 그러나 이 구조도는 작업자의 학습 훈련을 통한 지식획득 방법으로서 비교적 덜 복잡한 문제에 사용가능하며, 그렇지 않고 난이도가 높은 문제에 대하여 전문가 시스템을 구축하기에는 적합하지 않다. 그 주된 이유는 작업자가 복잡한 문제와 모의실험 결과를 분석하여 지식획득을 수행하기에는 인간의 능력을 크게 벗어나기 때문이다. 예를들면, 어떤 객체를 구성하는 속성(Attribute)의 수가 대단히 많고 그 속성이 이산치가 아닌 연속치 값을 갖는다면, 그 속성들의 상호작용이 커짐으로 인해 연속치 값에 대한 명확한 구분을 정하여 객체들을 분류하는 일은 인간으로 하여금 불가능한 일이다.

두번째 가능한 구조도는 모의실험기, 속성 추출

기(Feature Extractor), Machine Learning System, 성능 평가기(Performance Evaluator), 전문가 시스템으로 이루어진다. 이 구조도는 전자가 갖는 한계성을 극복하여 복잡하고 어려운 문제나 인간 전문가를 확보하기가 어려울때 전문가 시스템의 구축을 가능하게 하여 준다. 따라서 본 연구에서 추구하는 주요 구조는 더욱 일반성을 갖는 후자의 경우로 귀착된다.



[그림 1] 모의실험을 통한 전문가 시스템 구조도

우선 모의실험기는 Task Generator, Task Executer와 Evaluator로 구성된다. 가령 일정계획의 예를들면 많은 경우의 작업을 발생시키는 Job Generator, 이 작업들을 일정계획하는 Job Scheduler와, 그 유도된 일정계획을 특정기준에 의거 평가하는 Schedule Evaluator가 필요하게 된다. 이러한 기능들이 가능하다면 어떤 모의실험 소프트웨어라도 크게 문제되지 않는다.

일단 지정된 입력 파라메터에 의거하여 얻어진 모의실험 결과는 속성 추출기에 의해 속성치(Attribute Value)들이 추출된다. 물론 이에 우선하여 주요 속성들이 정의되어져야 한다. 이러한 속성을 결정하는 방법은 많은 모의실험을 통해 수동으로 행해지는 작업으로 많은 시간과 노력을 필요로 한다. 그러나 일단 이 과정이 끝나면 그 이후의 과정은 거의 자동적으로 행하여 진다는 점에서 속성의 정의 과정을 애로공정(Bottleneck)으로 볼 수 있다. 이렇게 하여 Machine Learning System에 필요한 많은 수의 훈련용 예제(Training Set)가 확보될 수 있으며, 이것들은 전문가 시스템의 지식획득을 위한 주요한 지식원이 된다.

인간 두뇌에서 행해지는 배움 현상을 여하히 컴퓨터 프로그램으로 재현시킬 수 있는가를 기본 목표로 하는 Machine Learning에는 성능 향상과 지식획득의 두 관점에서 설명되어 진다. 전자의 관점으로는 SOAR[7]나 LEX[2] 등을, 후자의 관점으로는 Induction[2]과 Neural Net[8] 등을 대표적인 것으로 볼 수 있다. 특히 Induction은 Neural Net와는 달리 적은 갯수의 훈련예제와 짧은 훈련기간이 소요된다는 장점때문에 본 연구의 대상이 되었다. Induction에 대한 연구는 초창기 CLS[5]에서 출발하여 최근 C4[15]에 이르기까지

많은 연구가 진행되어 왔다. Kim[6]은 이에 대한 설명을 자세하게 하고 있으므로 본 연구에서는 C4의 개정판인 C4.5에 대해서만 다음 절에서 설명하기로 한다.

결국 Induction시스템 중 하나인 C4.5는 유도된 훈련용 예제를 통하여 일련의 규칙을 출력시킨다. 이 출력된 규칙을 미 학습자료(Test Data)에 적용하여 분별해 내는 그 규칙들의 성능을 관측한다. 이때 미 학습자료는 모의실험기에 의해 제공된다. 이러한 성능의 관측과 평가는 수동으로 행하여지거나 비교과정을 프로그램하여 자동화 시킬 수 있다. 훈련용 예제의 갯수가 증가함에 따라 출력된 규칙의 성능이 변화하기 때문에 평가관 또는 시스템 개발자는 만족스러운 수준에 도달할 때까지 예제 발생과 규칙의 출력 과정을 반복적으로 수행하여야 한다. 이렇게하여 최종적으로 결정된 규칙은 전문가 시스템의 지식베이스를 형성한다.

2.2 C4.5

C4.5는 ID3[12]를 확장한 시스템으로 의사결정 나무구조 발생기(Decision Tree Generator)와 Interpreter, 규칙 발생기(Rule Generator)와 Interpreter로 구성되어 있다. 의사결정 나무구조 발생기는 특정 갯수의 훈련용 예제를 입력으로 받아 의사결정 나무구조를 발생시킨다. 이 나무구조를 이용하여 규칙 발생기는 일련의 규칙을 만들어 낸다. 나머지 두개의 Interpreter는 사용자로 하여금 일회적인 질의응답을 받을 수 있도록 지원되었다.

우선 의사결정 나무구조 발생기는 훈련용 예제에서 일부의 예제를 임의로 선택하여, 그 예제들을 잘 분별할 수 있는 나무구조를 작성한다. 이 임시 나무구조는 훈련용 예제의 선택되지 않은 나머지 예제들에 대하여 계속 분별검사를 시도한다. 이때 분별을 제대로 하지 못한 경우 그 예제를 추가하여 임시 나무구조를 재작성한다. 훈련용 예제를 모두 분별할 때까지 이러한 과정이 반복적으로 실시된다.

나무구조를 구축할때 어려운 일 중의 하나는 나무구조의 각 노드(Node)에 어떠한 속성을 배치할 것인가를 결정하는 문제이다. 이렇게 결정된 노드는 훈련용 예제를 분류하는데 대단히 중요한 역할을 한다. Quinlan[13]은 각 노드에 배치할 속성을 결정하기 위한 여러가지 선택 기준에 대한 연구를 수행하였다. 이들 기준 중 C4.5는 정보이론에 입각해서 최대 Gain을 얻은 속성을 선택하여 배치하는 방법을 이용하였다. 일단 속성들을 노드에 배치함으로써 나무구조가 유도되면, 그 나무구조의 효용을 높이기 위한 차원에서 특정가지의 대체작업과 일반화(Generalization)과정이 이루어진다[14]. 이렇게 함으로써 충분한 분별력을 갖고 최소한의 갯수로 이루어진 일련의 규칙들을 유도할 수 있다.

이 C4.5는 ID3와 크게 세 가지 면에서 구별될 수 있다. 첫째, ID3가 제공하는 나무구조 보다 한 단계 더 나아가 규칙으로 지식을 표현해 줌으로써 편리함과 이해도를 증진시켜 준다. 둘째, 복잡하고 어려운 문제에서 거의 필연적으로 발생할 수 있는 Noise 데이타의 처리 문제에 효과적으로 대처할 수 있다. 세째, 실제 문제에서 속성이 이산형이 아닌 연속적인 값을 갖는 경우가 흔히 존재한다. 이렇게 연속적인 속성치 값을 취하는 것은 Induction 사용의 큰 제약이 되어 왔으나 C4.5는 이러한 문제도 효과적으로 해결할 수 있다.

3. 적용 사례

본 장에서는 앞에서 개발된 구조도의 사용 방법을 실제로 보여 주기 위해 한 문제를 선정하였다. 우선 그 선정된 시스템과 가지고 있는 문제점을 개략적으로 서술하고, 이어서 SIMBES 구축을 위한 모델과 가정사항을 제시한다.

3.1 기존 시스템 개요

MRP(Material Requirements Planning) 시스템은 자재소요를 계획하고 추적하는 독보적인 능력을 갖고 있으나 생산 일정계획 수립에 도움을 주지 못한다. 반면에 생산 일정 계획시스템은 비교적 좋은 일정계획 수립을 가능케는 하나 세부 자재 계획 능력을 갖지 못한다. 따라서 바람직한 생산계획 수립을 위해서는 실제로 이 두 시스템은 통합되어져야 한다. 이러한 논리에서 Hastings를 중심으로 한 연구진은 Schedule-Based Material Requirements Planning(SBMRP) 시스템을 개발하였다[4].

이 시스템에서 이용하는 주요 방법은 MPS(Master Production Schedule)내의 작업을 주어진 자원의 용량내에서 전진 일정계획(Forward Scheduling) 방법으로 로딩(Loading)시킨다. 이때 물론 작업의 선후관계가 지켜질 수 있도록 작업의 일정계획은 진행되며, 작업로딩 순서는 암시적으로 FIFO(First In First Out)를 이용하였다. 이렇게 함으로써 최적안은 아닐지라도 실행가능한(Feasible) 일정계획안을 대단히 용이하게 도출할 수 있다. 또한 자재 소요계획은 유도된 작업 일정계획에 따라 자동적으로 시간별 계획이 수립된다. 여기에서 중요한 개념은 일정계획의 실행가능성이 이미 검증되었기 때문에 이 계획안을 기본으로 유도된 시간별 자재 소요 계획도 물론 실행가능하다는 점이다. 이와같이 SBMRP 시스템은 실행가능한 생산 일정계획 뿐만 아니라 실행가능한 자재 소요계획을 빠르게 유도할 수 있다는 점에서 대단히 유용하다.

그러나 전진 일정계획 방법을 이용하는 SBMRP 시스템은 많은 재고비용을 수반하는 단점이 있다. 이러한 문제점을 해결하기 위해 White[7]는 후진 일정계획(Backward Scheduling)의 개념을 도입하였다. 이 후진 일정계획 방법은 전진 일정계획 방법과 유사하나 납기를 기준으로 하여 마지막 공정을 우선적으로 일정계획을 하고 선후관계를 지키면서 후속작업을 일정계획하는 점이 다르다. 그러나 이 방법은 재고비용을 감소시킬 수 있는 반면 마감일을 지키지 못함으로 인한 지연비용을 증가시킬 수 있는 문제점이 있다. 본 논문에서는 SIMBES 개발의 예시를 보여주기 위해 두개의 기존 시스템 중 Hastings 연구진이 개발한 SBMRP에 대해서만 비교와 평가가 이루어 진다.

3.2 문제의 설명 및 가정

SBMRP 문제를 위한 하나의 Job Shop모형은 편의상 5대의 기계로 구성되며, 각 제품의 BOM은 최대 6개의 제조 부품과 다수의 구매 부품으로 구성된다. 그 제조부품들은 각각 작업순서, 작업시간, 납기등을 지정받는다.

본 시스템에 대한 가정은 다음과 같다.

(1) 일정계획 목표는 지연비용과 재고비용의 합을 최소로 한다.

(2) 각 기계는 한번에 한 작업만을 수행한다.

(3) Preemption은 허용하지 않는다.

(4) 작업시간은 이동시간과 준비시간을 포함한다.

(5) 작업의 납기는 비확률적이다.

(6) 지연 단위 비용과 재고 단위 비용은 동일하다.

(7) 각 작업의 작업시간은 비확률적이다.

4. SIMBES의 개발 및 평가

SIMBES를 개발하기 전에 우선적으로 Induction의 기초가 되는 예제를 정의하였다. 일단 정의를 완료한 후 앞에서 개발한 구조도에 따라 하나의 SIMBES가 Smalltalk언어를 사용해 구현되었다. 그러나 본 장에서는 실제 프로그램 코드 보다는 그

개발 절차에 촛점을 두어 구체적으로 설명한다. 마지막으로, 개발된 시스템과 기존시스템 간의 성능을 비교 평가한다.

4.1 예제의 정의

Induction의 출발점인 예제는 여러 개의 속성(Attribute)과 범주(Class)로 이루어져 있다. 따라서 예제의 정의를 위해서는 속성과 속성치가 취할 수 있는 값의 형태와 범위, 각 예제가 속하는 범주를 구체적으로 지정해야 한다. 본 연구에서 여러가지 일정계획 방법을 범주로, 연속적인 수치를 갖는 여러가지 작업의 속성들을 예제의 속성으로 간주하였다.

기존의 방법들이 암시적으로 FIFO를 사용하였다는 점에 착안하여 많은 작업선택 규칙(Dispatching Rules)중 전형적으로 이용되는 다섯개를 선정하였다. 이들은 FIFO, EDD(Earliest Due Date), SPT(Shortest Processing Time), LPT(Longest Processing Time)와 , LS(Least Slack)로 Panwalker와 Iskander[10]는 이외에도 많은 작업선택 규칙들을 상세하게 설명하고 있다. FIFO는 작업 도착의 빠른 순서에 따라 작업을 선택하며 EDD는 가장 빠른 납기를 갖는 작업을 우선적으로 선택하는 규칙이다. 또한, SPT와 LPT는 작업시간의 크기에 따라 작업의 선택우선권을 달리하는 것으로 전자는 가장 작은 작업시간을, 후자는 가장 긴 작업시간을 갖는 작업이 우선적으로 선택된다. 마지막으로 LS는 납기에서 작업시간을 제한 여유시간이 최소로 되는 작업을 우선적으로 선택하는 규칙이다.

이 다섯가지 작업선택 규칙에 전진 및 후진 일정계획 방법을 조합하면 총 10개의 조정된 규칙을 얻을 수 있다. 예를 들면, Forward EDD는 전진 일정계획에 의거 작업 일정계획을 수립하되, 작업선택 규칙은 EDD로 납기일이 가장 이른 작업을 우선적으로 일정계획함을 의미한다. 따라서 본 문제가 다루는 범주는 10개의 요소로 구성된다.

작업의 속성을 정의하는 과정은 대단히 어려운 단계로 초기에는 중요한 하나 또는 두개의 속성만으로 모의실험을 실시한다. 그와같은 실험이 반복됨에 따라 여타 속성들이 추가되는 과정을 거친다. 본 문제에서 작업의 여유시간, 납기일, 작업시간등의 정보는 만족스러운 일정계획을 유도하는데 대단히 중요하다[1][3]. 이와 같은 사실을 바탕으로 많은 실험적인 과정을 거쳐 여덟개의 속성들이 정의되었다. 대표적인 속성들로는 총 가능시간과 총 작업 처리시간의 비율(taOVERtp), 납기의 분산(varOfDues), 여유시간의 분산(varOfSlacks), 작업갯수의 분산(varOfNoOfOperations), 처리시간의 분산(varOfProcTimes), 공장 부하의 분산(varOfLoad) 등이 있다.

4.2 SIMBES의 개발

SIMBES의 출발점인 모의실험기는 본 문제에 적용했을 때 Job Generator, Job Scheduler와, Evaluator로 크게 구분된다. 이 모의실험의 Job Generator에 필요한 파라메터들을 살펴보면, 작업시간 분포는 Exponential(2)와 Exponential(3)을, 납기일 분포는 Uniform(0,5), Uniform(0,10), Uniform(0,15)와 Uniform(0,20)을 갖도록 정의하였다. 이외에 수집할 총 작업수로 480개의 작업이 입력되면 Job Generator는 각각의 경우에 60개의 작업을 발생한다. 이와같이 발생된 각 작업들에 Job Scheduler는 앞서 정의한 10개의 조정된 규칙들을 적용하여 일정계획을 수행한다. 이때 Evaluator는 유도된 10개의 일정계획을 총 비용의 관점에서 평가하여 최선의 조정된 규칙을 선정한다.

이 최선의 규칙으로 귀착된 작업에 대하여 속성 추출기는 앞서 정의된 여덟개 속성의 속성치들을 분석하여 추출해 낸다. 이와같은 과정으로 수집할 총 작업수 만큼 반복하면 총 480개의 훈련용 예제

를 얻게 된다. 이 훈련용 예제의 좌편은 정의된 속성의 값들로 우편에는 최선으로 판정받은 조정된 규칙으로 이루어져 있다. C4.5는 이 훈련용 예제를 입력으로 하여 규칙으로 표현된 지식을 자동적으로 발생시켜 준다. 이 C4.5는 Smalltalk으로 구현되지 않은 유일한 요소로서 UNIX에서 운영되는 소프트웨어이다.

〈표 1〉 Smalltalk로 표현된 지식의 일부 예제

```
SBMRPExpert add : (Rule
        number : 1
        condition : [
                (#taOVERtp gt : 1.33)
          & (#taOVERtp lt : 1.45)
          & (#varOfDues gt : 4.34)
          & (#vorOfSlacKs gt : 0.98)
          & (#varOfSlacks lt : 23.24)
          & (#varOfLoad lt : 8.38)
         action : [#forwardLS]).

SBMRPExpert add : (Rule
        number : 2
        condition : [
                (#taOVERtp lt : 1.27)
          & (#varOfDues lt : 46.27
          & (#varOfProcTimes gt : 12.2)
          & (#varOfProcTimes lt : 48.24
          & (#varOfNoOfOperations lt : 0.2)]
        action : [#forwardEDD]).
```

C4.5에 특정 갯수의 훈련용 예제를 사용하여 일련의 지식을 얻었다 할지라도 최종적으로 제공하여야 할 예제의 갯수를 결정하는 문제는 대단히 어려운 문제이다. 그것은 일반적으로 훈련용 예제가 많으면 많을수록 유도된 지식의 질(Quality)은 나아지기 때문이다. 이 문제를 풀기위한 하나의 방편으로 규칙 수의 증감과 그 규칙의 분별력을 관찰하여 만족스러운 수준을 결정하였다. 이 단계까지 완료되면 확정된 최종규칙을 얻을 수 있으며, 이 규칙들은 KBES로 약칭되는 전문가 시스템(Knowledge-Based Expert System)의 지식베이스를 형성한다.

이 지식의 일부가 Smalltalk언어로 〈표 1〉에 제시되고 있다. 예를들면, 첫번째 규칙은 어떤 일정계획 작업의 taOVERtp가 1.33보다 크고 1.45보다 작으며, varOfDues가 4.34보다 크며, VarOfSlacks이 0.98보다 크고 23.24보다 작고, varOfLoad가 8.38보다 작은 경우에는 forwardLS 선택규칙을 이용하라는 식으로 읽혀진다.

4.3 시스템의 성능 평가

상기와 같은 구조도에 따라 총 31개의 규칙이 도출되어 지식베이스를 구성하고 여기에 Inference Engine이 추가되어 하나의 KBES가 구축되었다. SIMBES의 최종제품으로 나타나는 이 KBES는 마지막으로 평가단계를 거쳐야 한다. 이를 위해 비교의 대상으로 Hastings의 연구진이 개발했던 기존 SBMRP시스템을 선정하였으며 작업들의 일정계획 능력을 두가지 기준 위에서 평가하였다.

이의 구체적인 내용을 살펴보면 우선 Job Generator를 통해 480개의 작업을 임의로 발생시킨 뒤 이 작업들에 기존 시스템과 KBES로 일정계획을 수립하도록 만들었다. 이와같이 유도된 일정계획들에 대하여 어느 시스템이 상대적으로 보다 나은 일정계획을 많이 만들어냈는가와 총비용의 두가지 기준에서 평가하였다. 이와같은 평가를 10회 반복하여 실시하였으나 두 시스템의 차이는 통계적으로 안정된 결과로 나타났다. 이에 대한 상세한 결과가 〈표 2〉에 요약되어 있다. KBES는 평균적으로 전체 작업의 72%에 대해, 기존 시스템은 단지 28%에 대해 우월하게 일정계획을 수립하였다. 이러한 결과는 SIMBES를 이용하여 구축된 KBES의 성능이 기존 시스템을 크게 능가하고 있음을 보여주고 있다.

〈표 2〉 기존시스템과 KBES의 성능비교

횟 수	기존 시스템		KBES	
	우월한 일정 계획의 수	총 비 용	우월한 일정 계획의 수	총 비 용
1	150	21344	357	13941
2	137	21370	369	13837
3	146	21088	354	13754
4	138	21302	367	13788
5	146	21432	357	14074
6	146	20628	371	13139
7	147	20717	367	13154
8	147	21262	369	14090
9	141	21364	364	13863
10	144	20531	363	13026
평 균	144	21104	364	13667

5. 결 론

최적화 방법의 한계로 나타나는 복잡하고 어려운 실제적인 문제를 풀려는 노력은 자연스럽게 모의실험으로 이어지곤 한다. 그러나 이러한 모의실험으로 도출되는 결과들을 일회용으로 끝내지 아니하고 시간과 자원의 제약된 상황에서 효과적으로 활용할 수 있는 방법론이 필요하였다. 이러한 필요성에 의거하여 본 연구에서는 모의실험을 통한 전문가 시스템을 구축하기 위한 구조도를 개발하였다.

더구나 최근 전문가 시스템의 발전과 더불어 지식획득의 중요성은 더욱 강조되고 있다. 그러나 전문가가 없는 경우나 전문가를 극히 구하기 어려울 때 지식획득은 불가능하게 된다. 이러한 경우에도 그 문제의 모델을 구축하여 모의실험을 행함으로써 인공적인 지식을 자동적으로 추출하여 문제의 해결 내지는 성능제고를 도모할 수 있다는 점은 SIMBES의 구조도에 매우 큰 중요성을 부여한다.

이에 대한 구체적인 사례를 보여주기 위해 SBMRP문제가 선정되었으며, 이를 근거로 SIMBES가 실험적으로 개발되었다. 개발된 시스템을 평가하기 위해 총 4800개의 일정계획작업이 이용되었다. KBES는 이들 중 약 72%에 대해 기존 시스템 보다 우월한 일정계획을 수립하였다. 이 사실은 SIMBES의 성능을 단적으로 보여줌과 아울러 SIMEBS구조도의 필요성을 보여주고 있다. 따라서 상황이나 가정이 변한다해도 본 연구에서 정의한 그 구조도에 따라 간단하게 일련의 지식을 추출하여 전문가 시스템을 용이하게 구축할 수 있다.

참 고 문 헌

[1] Conway, R. W., "Priority Dispa-tching and Job Lateness in a Job Shop," J. Ind. Eng., 16, 123, 1965.

[2] Cohen, P. R., A and Feigenbaum, E. A., The Handbook of Artificial Intelligence, Vol. 3,

William Kaufmann Inc., 1982.

[3] Elvers, D. A., "Job Shop Dispatching Rules Using Various Delivery Date Setting Criteria," Prod. Inv. Mgmt., 14,62, 1973.

[4] Hastings, N. A. J., Marshall, P., and Willis, R. J., "Schedule-Based MRP : An Integrated Approach to Production Scheduling and Material Requirements Planning," J. Opl. Res. Soc., Vol. 33, 1021-1029, 1982.

[5] Hunt, E. B., Marin, J., and Stone, P. J., Experiments in Induction, New York : Academic Press, 1966.

[6] Kim, S. U., "The Role of Machine Learning to Acquire Knowledge for Expert Systems," Journal of Computer Application Research Group, Vol. 4, No. 1, 1991.

[7] Laird, J. E., Rosenbloom, P. S., and Newwell, A., "Towards Chunking as a General Learning Mechanism," Proceedings of AAAI-84, Austin, Texas, 1984.

[8] Lippmann, R. P., "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, April, 1987.

[9] Olson, J. R., and Rueter, J. H., "Extracting Expertise from Experts : Methods for Knowledge Acquisition," Expert Systems, August, Vol. 4, No. 3, 1987.

[10] Panwalker, S. S., and Iskander, W., "A Survey of Scheduling Rules," Operations Research, Vol. 25, No. 1, Jan-Feb., 45-61, 1977.

[11] Pritsker, A. Alan B., Introduction to Simulation and SLAMII, John Wiley & Sons, New York, 1984.

[12] Quinlan, J. R., "Discovering Rules by Induction from Large Collection of Examples," In D. Michie(ed), Expert Systems in the Micro Electronic Age, Edinburgh Univ. Press, 1979.

[13] _____, "Induction of Decision Trees," Machine Learning 1, 81-106, 1986.

[14] _____, "Simplifying Decision Trees," International Journal of Man-Machine Studies 27, 221-234, 1987.

[15] Quinlan, J. R., Compton, P. J., Horn, K. A., and Lazarus, L., "Inductive Knowledge Acquisition : A Case Study," Proceedings of the Second Australian Conference on Applications of Expert Systems, Sydney, 1986.

[16] Rinnooykan, A. H. G., Machine Scheduling Problems:Classification, Complexity, and Computations, Nijhoff, The Hague, Holland, 1976.

[17] White, C., Modelling and Design of Flexible Manufacturing Systems, edited by Kusiak, A., Elsevier Science, Amsterdam, The Netherlands, 1986.