

☒ 응용논문

속성지향추론법과 시뮬레이션을 이용한 지식기반형 Job Shop 스케줄러의 개발[†]

- Development of a Knowledge-Based Job Shop Scheduler Applying
the Attribute-Oriented Induction Method and Simulation -

한 성식*

Han, Seong Sik

신 현표**

Shin, Hyeon Pio

Abstract

The objective of this study is to develop a knowledge-based scheduler applying simulation and knowledge base. This study utilizes a machine induction to build knowledge base which enables knowledge acquisition without domain expert.

In this study, the best job dispatching rule for each order is selected according to the specifications of the order information. And these results are built to the fact base and knowledge base using the attribute-oriented induction method and simulation. When a new order enters in the developed system, the scheduler retrieves the knowledge base in order to find a matching record. If there is a matching record, the scheduling will be carried out by using the job dispatching rule saved in the knowledge base. Otherwise the best rule will be added to the knowledge base as a new record after scheduling to all the rules. When all these above steps finished the system will furnish a learning function.

1. 서론

지식베이스를 만들기 위한 지식 획득 방법에는 크게 지식 추출(knowledge elicitation)과 기계추론(machine induction)의 두 가지가 있다. 그런데 지식베이스의 근원을 살펴보면 대부분 인간 전문가와의 면담이나 전문서적과 전문잡지 등을 통한 지식 추출이 대부분인데 이러한 방법은 시간과 비용이 과다하게 소요되는 것은 물론 지식 획득 자체의 어려움 때문에 전문가 시스템의 개발에 있어서 주요 애로점이 되고 있다. 이러한 이유로 인해 인간 전문가를 필요로 하지 않고도 지식을 획득할 수 있는 방법인 기계추론 방법이 주목을 받으며 많은 연구가 진행되어 왔으며 본 연구에서도 그 방법을 이용하고자 한다 [1, 6, 10, 12, 15].

지식기반형 일정계획 전문가 시스템(knowledge-based scheduling expert system)을 구축하기 위해서는 먼저 시뮬레이션을 이용해서 기반 데이터를 생성해야 하는데 기존의 연구에서는 이 단계에서 시뮬레이션 전용 언어를 이용하였고 이렇게 얻어진 데이터를 저장하기 위해서 DBMS(Database Management System)를 사용하였다. 그런데 시뮬레이션 언어와 DBMS의 인터페이스가 용이하지 않기 때문에 중간에 인간의 인위적인 개입이 필요하게 된다. 그러나 본 연구에서는 시뮬레이션과

* 인하대학교 산업공학과 석사과정

** 인하대학교 산업공학과 교수

† 본 연구는 1997년도 인하대학교 산업과학기술연구소 연구비 지원에 의해 수행되었음

추론기관 구축에 범용 프로그래밍 언어인 Microsoft Visual Basic 5.0[®]을 사용하고 지식베이스에 관계형 DBMS인 Microsoft Access 97[®]을 사용하여 인터페이스 문제를 해결하였다. 두 프로그램은 동일하게 Jet Database Engine을 사용하기 때문에 저장할 때는 물론 데이터를 검색하고 가공하는데도 인간의 개입이 전혀 불필요한 시스템으로의 개발이 용이하다.

실제 연구에 있어서는 시뮬레이션을 이용하여 각 주문정보의 특성에 따른 최적의 작업할당규칙을 선택하고 이 결과를 속성지향 추론법을 이용하여 사실베이스(fact base)와 지식베이스(knowledge base)로 구축한다. 결국 새로운 주문정보가 입력되는 경우에는 스케줄러가 구축된 지식베이스를 검색해서 주문정보의 특성과 일치하는 레코드가 있으면 해당 작업할당규칙을 사용하여 일정계획을 수행하고 일치하는 레코드가 없으면 모든 규칙에 대해 일정계획을 수행한 후 최적의 규칙을 선택하여 기존의 지식베이스에 새로운 레코드로 추가하는 학습 기능을 갖추게 된다.

2. 속성지향 추론법

속성지향 추론법(attribute-oriented induction method)이란 관계형 데이터베이스에 저장되어 있는 각각의 레코드로부터 규칙 형태의 지식을 추출하는 방법으로서 각 레코드를 개념적 경향(conceptual bias)에 따라 정리한다. 즉 특정한 필드의 값이 여러 가지로 주어질 때 그 값들을 일정한 임계치(threshold value)의 개수로 줄이면서 해당 필드를 일반화시키는 것이다. 일종의 그룹화를 통해 각 그룹에 특정한 기호를 부여하는 것으로도 볼 수 있다 [3, 7, 9, 13].

다음의 [표 2-2]는 [표 2-1]의 내용을 일반화시키는 방법을 보여준다. 즉 $\{A_1, \dots, A_k\} \subset B$ 라는 식에서 $1 \leq i \leq k$ 인 경우 B 는 A_i 의 일반화라는 뜻이다. 그런데 일반화를 시키기 전에 각 레코드에 초기값이 1인 vote라는 새로운 필드를 만들어 둔다. 그리고 실제로 일반화시키는 과정에 들어가서는 모든 필드의 값이 동일한 레코드가 두 개 이상 있을 경우에 하나의 레코드만 남겨두고 나머지는 모두 삭제하며 남겨진 레코드의 vote 필드에 삭제되는 레코드들의 vote 필드 값을 누적시켜 새로운 레코드로 받아들인다. 결국 이렇게 만들어진 레코드를 하나의 규칙으로 인식하는 것이다.

[표 2-1] A 대학의 데이터베이스

Name	Category	Major	Birth Place	GPA*
Anderson	M.A.	history	Los Angeles	3.5
Fraser	M.S.	physics	Dallas	3.9
Gupta	Ph.D.	math	Bombay	3.3
Liu	Ph.D.	biology	Shanghai	3.4
Monk	Ph.D.	computing	San Diego	3.8
Wang	M.S.	statistics	Nanjing	3.2

* GPA = Grade Point Average

[표 2-2] A 대학 데이터베이스의 Concept Hierarchy

$\{computing, math, biology, statistics, physics\} \subset science$
$\{history\} \subset art$
$\{Los Angeles, San Diego\} \subset California$
$\{Dallas\} \subset Texas$
$\{Bombay\} \subset India$
$\{Shanghai, Nanjing\} \subset China$
$\{China, India\} \subset foreign$
$\{California, Texas\} \subset USA$
$\{3.0 - 3.4\} \subset good$
$\{3.5 - 4.0\} \subset excellent$

[표 2-3] A 대학 데이터베이스의 일반화 I

Major	Birth Place	GPA	Vote
art	California	excellent	1
science	Texas	excellent	1
science	California	excellent	1
science	India	good	1
science	China	good	2

[표 2-4] A 대학 데이터베이스의 일반화 II

Major	Birth Place	GPA	Vote
art	USA	excellent	1
science	USA	excellent	2
science	Foreign	good	3

A 대학 데이터베이스의 특성을 알고 싶을 때 [표 2-1]의 데이터를 전공, 출생지, 평점 등의 3개 필드에 대해서 일반화시킨 결과는 [표 2-3]과 같다. 그리고 이 결과를 한번 더 일반화시킨 결과는 [표 2-4]와 같다. 최종 결과를 통해서 알 수 있는 사실은 먼저 미국 출신 학생은 인문계와 자연계 모두 성적이 뛰어나다는 것이다. 또한 외국 출신 자연계 학생들의 성적이 우수하다는 사실도 알 수 있다.

이상과 같은 방법으로 데이터베이스를 정리한 후에 데이터베이스를 갱신시키기 위해서는 정리된 데이터베이스를 대상으로 학습을 시킨다. 새로운 레코드가 발생하면 먼저 원래의 데이터베이스를 일반화시킨 기준과 동일한 방법으로 새 레코드를 일반화시킨다. 그리고 나서 일반화시킨 새 레코드의 각 필드 값을 데이터베이스에 정리되어있는 레코드들과 비교하여 vote를 제외한 모든 필드 값이 동일한 레코드가 있으면 기존 레코드의 vote 값을 1 증가시키며 새 레코드는 삭제한다. 만일 일치하는 레코드가 없으면 새 레코드의 vote 필드에 초기값 1을 부여하며 데이터베이스에 새로운 레코드로 추가시킨다.

3. 지식기반형 스케줄러

3.1 실험대상 시스템

실험 대상 시스템은 5개의 작업과 3대의 기계로 구성되는 job shop 환경이며 각 작업 별로 필요한 기계의 수는 1대에서 3대 사이로 각기 다르다. 그리고 각 작업마다 사전에 정해진 기계별 작업 순서를 따르며 그 순서마다 특정한 가공시간을 필요로 한다. 작업할당규칙에는 Chang 등의 연구에서 사용된 42가지 중에서 대표적인 12개를 사용하며 하나의 주문정보에 대해서 두 개 이상의 규칙이 tie를 이룰 때는 Chang 등의 실험 결과에 나온 우선 순위에 따라 tie-breaker로 사용된다. 그리고 일정계획을 평가할 수행 척도(performance criteria)로는 다음과 같은 여섯 가지를 사용한다 [2, 4, 5, 8, 14].

- (1) makespan or maximum completion time (C_{max})
- (2) mean completion time (\bar{C})
- (3) maximum waiting time (W_{max})
- (4) mean earliness (\bar{E})
- (5) mean tardiness (\bar{T})
- (6) number of tardy jobs (N_t)

[표 3-1] 작업할당규칙

순위	작업할당규칙	설명
1	SPT · TWK	Select the job with the smallest value of the processing time multiplied by the total work
2	SPT	Select the job with the shortest processing time
3	SPT/TWK	Select the job with the smallest ratio of the processing time to the total work
4	LTWK	Select the job with the least total work to be done
5	EDD	Select the job with the earliest due date
6	STSLACK/TWK	Select the job with the smallest ratio of its static slack to the total work
7	STSLACK	Select the job with the smallest static slack
8	STSLACK/OP	Select the job with the smallest ratio of its static slack to the number of total operations
9	MTWK	Select the job with the most total work to be done
10	LPT · TWK	Select the job with the largest value of the processing time multiplied by the total work
11	LPT	Select the job with the longest processing time
12	LPT/TWK	Select the job with the largest ratio of the processing time to the total work

이상과 같은 방법으로 발생시킨 주문정보를 사실베이스로 저장하려면 앞서 소개한 속성지향 추론법을 사용해야 한다. 즉 다양한 값으로 나타나는 각 인자의 값들을 일정한 임계치의 개수로 줄이면서 각 인자를 일반화시켜야 한다 [3]. 임계치의 개수를 정하기 위해서는 각 인자 별로 수행 척도에 미치는 영향을 분석해 볼 필요가 있다. 1,000개의 주문정보를 발생시킨 후 MINITAB Release 10.5 Xtra[®]를 사용하여 주문정보의 6가지 매개변수와 6가지 수행 척도의 상관관계를 분석한 결과는 [표 3-5]와 같다. 그리고 이러한 상관관계를 고려하여 [표 3-6]과 같이 상관관계가 높은 인자일수록 임계치의 개수를 많게, 상관관계가 낮은 인자일수록 적게 부여하는 것이다.

[표 3-5] 주문정보의 특징과 수행 척도 사이의 상관관계

	C_{max}	\bar{C}	W_{max}	\bar{E}	\bar{T}	N_i
Num	0.662	0.755	0.487	-0.383	0.387	0.331
NumSD	0.044	0.053	0.254	-0.025	-0.031	-0.032
Time	0.515	0.479	0.390	-0.184	0.239	0.171
TimeSD	-0.068	-0.134	-0.009	0.017	-0.044	-0.041
Date	0.004	-0.008	0.010	0.774	-0.617	-0.709
DateSD	-0.320	-0.365	-0.216	0.215	-0.102	-0.061

[표 3-6] 주문정보의 인자별 임계치의 개수

인자	수준	기호	인자	수준	기호
총가공수	$5 \leq \text{Num} < 7$	a1	평균가공시간의 표준편차	$0 \leq \text{TimeSD} < 0.45$	d1
	$7 \leq \text{Num} < 9$	a2		$0.45 \leq \text{TimeSD}$	d2
	$9 \leq \text{Num} < 12$	a3	평균납기	$1 \leq \text{Date} < 9$	e1
	$12 \leq \text{Num} < 14$	a4		$9 \leq \text{Date} < 13$	e2
	$14 \leq \text{Num}$	a5		$13 \leq \text{Date} < 17$	e3
가공수의 표준편차	$0 \leq \text{NumSD} < 0.7$	b1	평균납기의 표준편차	$17 \leq \text{Date}$	e4
	$0.7 \leq \text{NumSD} < 1.3$	b2		$0 \leq \text{DateSD} < 2.25$	f1
	$1.3 \leq \text{NumSD}$	b3		$2.25 \leq \text{DateSD} < 4.75$	f2
평균가공시간	$1 \leq \text{Time} < 1.75$	c1	$4.75 \leq \text{DateSD}$	f3	
	$1.75 \leq \text{Time} < 2.15$	c2			
	$2.15 \leq \text{Time} < 2.55$	c3			
	$2.55 \leq \text{Time}$	c4			

앞의 [표 3-3]의 주문정보 특징에 해당하는 6개 인자들을 일반화시킨 것과 시뮬레이션의 결과를 종합하면 다음의 [표 3-7]과 같다. 결과적으로 예제 1의 주문정보 하나에 대응해서 [표 3-4]와 [표 3-7]이 80개의 필드를 지닌 하나의 레코드로 저장된다. 본 연구에서는 50,000개의 다양한 주문정보를 생성하며 동일한 횟수만큼 시뮬레이션을 수행해서 나온 레코드 50,000개를 데이터베이스로 저장하여 사실베이스로 사용하였다. 참고적으로 133MHz Pentium CPU와 40MB RAM이 장착된 PC 환경에서 50,000개의 사실베이스 구축에 약 110시간이 소요되었다.

[표 3-7] 예제 1의 주문정보 특징 일반화

Num	NumSD	Time	TimeSD	Date	DateSD	BestRule	Vote
a4	b1	c2	d1	e2	f1	3	1

3.3 지식베이스의 구축

앞의 3.2절에서 언급한 방법으로 사실베이스를 구축했을 때 [그림 3-1]과 같이 동일한 주문정보의 특징("a1/b1/c1/d1/e3/f2")에 서로 다른 규칙이 선택되는 경우도 있다. 이러한 경우에는 먼저 8개의

필드 값 모두가 동일한 레코드들을 [그림 3-2]와 같이 하나의 레코드만 남기고 나머지는 모두 삭제하는데 남겨진 레코드의 vote 필드에 지워지는 레코드의 vote 필드 값을 누적시킨다.

a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	1

[그림 3-1] 동일한 조건에 다른 규칙이 선택되는 경우

a1	b1	c1	d1	e3	f2	1
a1	b1	c1	d1	e3	f2	3
a1	b1	c1	d1	e3	f2	5

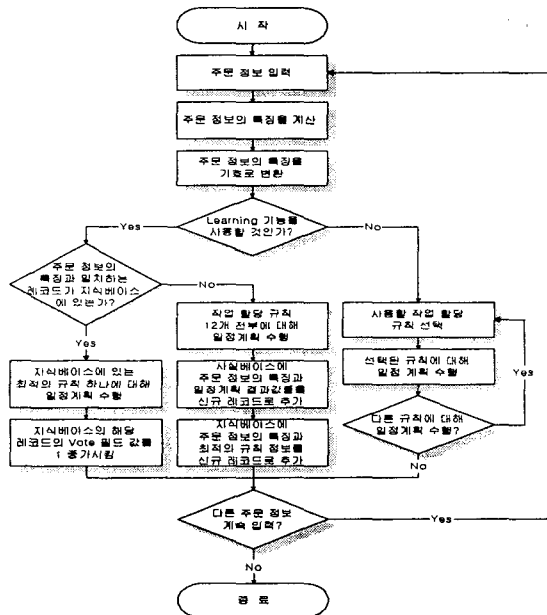
[그림 3-2] 규칙별로 vote를 누적시킨 결과

a1	b1	c1	d1	e1	f1	5
a1	b1	c1	d1	e2	f1	1
a1	b1	c1	d1	e2	f3	3

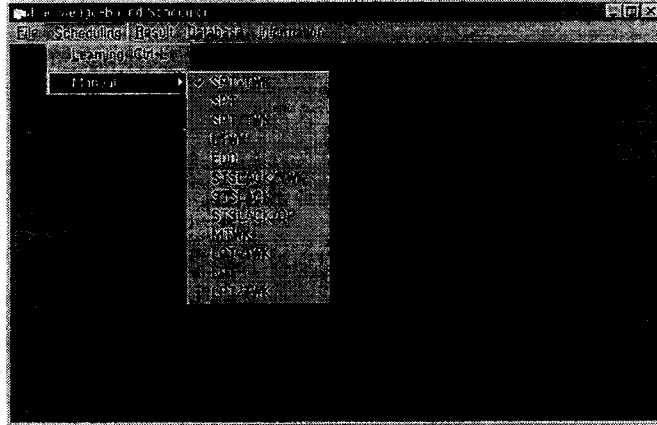
[그림 3-3] 단일 조건에 단일 규칙만 선택되는 최종 결과

[그림 3-2]와 같이 주문정보의 특징에 해당하는 6개 필드 값들은 동일하고 BestRule은 다른 레코드가 있는 경우에는 누적된 vote 필드 값을 비교하여 가장 큰 레코드만 남기고 나머지는 모두 삭제한다. 만일 이 과정에서 가장 큰 값에 tie가 발생하는 경우에는 [표 3-1]의 우선 순위가 tie-breaker로 사용된다. 결과적으로 하나의 주문정보의 특징("a1/b1/c1/d1/e3/f2")에 대응하는 단일한 규칙이 다음 그림-마지막 행-과 같이 선택된다. 결국 이와 같은 과정을 거치면서 만들어진 [그림 3-3]과 같은 데이터베이스가 최종 지식베이스로 저장되어 일정계획을 할 때 검색되게 된다. 그런데 위와 같은 수준으로 주문정보의 특징을 표현할 때 가능한 총 조합의 수는 1,440가지(= 5×3×4×2×4×3)가 된다. 참고적으로 실제 실험을 해본 결과 10,000개의 레코드를 만들었을 때는 616가지의 조합이 발생했고 50,000개의 레코드를 만들었을 때는 819가지의 조합이 발생했다.

3.4 지식기반형 스케줄러의 개발



[그림 3-4] 스케줄러의 주요 흐름



[그림 3-5] 지식기반형 스케줄러의 주 화면

1) 지식베이스에 일치하는 레코드가 있는 경우

[그림 3-6]과 같은 주문정보를 입력시킨 예제 2를 살펴보자. 이 상태에서 Learning 메뉴를 누르면 [그림 3-7]과 같은 대화상자가 나타난다. 즉 입력한 주문정보의 특징과 일치하는 레코드 ("a4/b1/c3/d1/e3/f1")가 기존의 지식베이스에 있기에 그곳에 저장되어 있는 SPT*TWK를 사용하여 일정계획을 수행한 것이다.

Job	Priority	Setup	Time	Weight
1	3	3	5	
1	2	1	5	
2	1	3	12	
3	2	2	21	
3	3	9	21	
3	1	1	21	
4	3	2	9	
4	2	1	9	
4	1	3	9	
5	1	3	22	
5	3	3	22	
5	2	3	22	
0	0	0	0	
0	0	0	0	
0	0	0	0	

[그림 3-6] 예제 2의 주문정보



[그림 3-7] 예제 2의 실행결과 알림 대화상자

Job ID	a4	Earliness	14	
Job ID	b1	Mean Completion Time	8	
Job ID	c3	Maximum Waiting Time	5	
Job ID	d1	Mean Tardiness	6	
Job ID	e3	Mean Tardiness	0	
Job ID	f1	Number of Tardy Jobs	0	
Rule	T (SPT-TWK)		Vote	72

[그림 3-8] 예제 2의 실행 결과

위의 [그림 3-8]을 보면 예제 2를 실행시킨 결과가 종합적으로 나와 있다. 예제 2에 관한 주문정보의 특징은 "a4/b1/c3/d1/e3/f1"이며 사용된 작업할당규칙은 SPT*TWK이다. 이 규칙을 사용하여 일정계획을 수행한 결과 makespan은 14분, mean completion time은 8분, maximum waiting time은 5분, mean earliness는 6분, mean tardiness는 0분 그리고 number of tardy jobs는 0개로 나타났다. 또한 Gantt Chart는 [그림 3-9]와 같다. 예제 2는 기존의 지식베이스를 이용하였기에 사실베이스의 레코드 개수는 50,000개로 변화가 없다. 그러나 지식베이스는 [그림 3-10]과 [그림 3-11]의 첫째 행에 나타난 것처럼 BestVote 필드 값이 71에서 하나 증가한 72로 바뀌게 된다.

위의 [그림 3-14]를 보면 예제 3을 실행시킨 결과가 종합적으로 나타나 있다. 예제 3에 관한 주문 정보의 특징은 “a5/b1/c4/d1/e4/f1”이며 12개의 작업할당규칙을 모두 사용하여 일정계획을 수행한 결과 SPT*TWK 규칙이 가장 우수한 것으로 나타났다. 이 규칙을 사용하여 일정계획을 수행한 결과 makespan은 21분, mean completion time은 15분, maximum waiting time은 12분, mean earliness는 5.1분, mean tardiness는 0.2분 그리고 number of tardy jobs는 1개로 나타났다. 이 예제에 관한 일정계획 결과 역시 Gantt Chart로 볼 수 있다.

예제 3의 주문정보 특징은 기존의 지식베이스에 일치하는 레코드가 없기에 [그림 3-15]와 같이 사실베이스에 레코드를 하나 추가하여 레코드 개수가 50,000개에서 하나 증가한 50,001개로 된다. 또한 지식베이스의 레코드 개수 역시 [그림 3-16]과 같이 819개에서 하나 증가한 820개로 된다.

Job ID	Job Type	Param 1	Param 2	Param 3	Param 4	Param 5	Param 6	Param 7	Param 8	Param 9	Param 10	Param 11	Param 12	Param 13	Param 14	Param 15	Param 16	Param 17	Param 18	Param 19	Param 20	
49982	a3	b3	c1	d1	e3	f2	2	1	12	5.5												
49983	a4	b1	c3	d1	e2	f1	3	1	16	9.1												
49984	a2	b3	c2	d2	e1	f2	5	1	11	6.1												
49985	a3	b1	c1	d1	e2	f1	1	1	7	4.1												
49986	a3	b2	c2	d2	e3	f1	1	1	9	5.1												
49987	a3	b1	c2	d2	e2	f1	1	1	9	5.1												
49988	a4	b1	c2	d2	e2	f1	6	1	16	2.5												
49989	a2	b2	c2	d2	e2	f1	8	1	11	5.1												
49990	a3	b1	c2	d1	e3	f2	3	1	9	4.4												
49991	a3	b1	c1	d1	e3	f1	1	1	9	5.1												
49992	a3	b2	c2	d1	e3	f1	9	1	14	5.1												
49993	a2	b1	c2	d1	e2	f1	1	1	7	4.4												
49994	a4	b3	c3	d1	e2	f1	5	1	16	9.6												
49995	a3	b1	c3	d1	e1	f1	3	1	13	7.1												
49996	a3	b2	c2	d1	e3	f1	3	1	11	6.1												
49997	a2	b2	c2	d1	e3	f1	3	1	9	5.1												
49998	a4	b1	c2	d1	e3	f1	4	1	14	9												
49999	a1	b2	c1	d2	e3	f2	1	1	5	2.5												
50000	a3	b3	c3	d2	e3	f2	5	1	13	8.3												
50001	a5	b1	c4	d1	e4	f1	1	1	21	15												

[그림 3-15] 예제 3의 사실베이스

Job ID	Job Type	Param 1	Param 2	Param 3	Param 4	Param 5	Param 6	Param 7	Param 8	Param 9	Param 10	Param 11	Param 12	Param 13	Param 14	Param 15	Param 16	Param 17	Param 18	Param 19	Param 20	
a5	b1	c1	d2	e2	f1	5	1	5	2													
a5	b1	c1	d2	e4	f1	1	1	1	1													
a5	b1	c2	d1	e1	f1	1	1	1	1													
a5	b1	c2	d1	e2	f1	1	1	1	1													
a5	b1	c2	d1	e3	f1	1	1	1	1													
a5	b1	c2	d1	e4	f1	1	1	1	1													
a5	b1	c2	d2	e1	f1	1	1	1	1													
a5	b1	c2	d2	e2	f1	1	1	1	1													
a5	b1	c2	d2	e3	f1	1	1	1	1													
a5	b1	c2	d2	e4	f1	1	1	1	1													
a5	b1	c3	d1	e1	f1	1	1	1	1													
a5	b1	c3	d1	e2	f1	1	1	1	1													
a5	b1	c3	d1	e3	f1	1	1	1	1													
a5	b1	c3	d1	e4	f1	1	1	1	1													
a5	b1	c3	d2	e1	f1	1	1	1	1													
a5	b1	c3	d2	e2	f1	1	1	1	1													
a5	b1	c3	d2	e3	f1	1	1	1	1													
a5	b1	c3	d2	e4	f1	1	1	1	1													
a5	b1	c4	d1	e2	f1	1	1	1	1													
a5	b1	c4	d1	e3	f1	1	1	1	1													
a5	b1	c4	d1	e4	f1	1	1	1	1													

[그림 3-16] 예제 3의 지식베이스

4. 결론

본 연구에서는 시뮬레이션과 속성지향 추론법을 통하여 사실베이스와 지식베이스를 구축한 후 전문가 시스템을 이용하여 구축된 지식베이스에서 가장 적합한 작업할당규칙을 찾아내어 적용하는 지식기반형 스케줄러를 개발하였다. 본 연구에서 개발한 시스템에 새로운 주문정보가 입력되는 경우에는 구축된 지식베이스를 검색해서 주문정보의 특성과 일치하는 레코드가 있으면 해당 작업할당규칙만을 사용하여 일정계획을 수행하고 일치하는 레코드가 없으면 모든 규칙에 대해 일정계획을 수행한 결과를 기존의 사실베이스와 지식베이스에 새로운 레코드로 추가하는 학습 기능이 있다. 또한 시뮬레이션과 추론기관 그리고 지식베이스 사이의 인터페이스 문제를 해결하였기에 새로운 일정계획을 수립하는 데는 데이터베이스를 검색하는 시간만큼만 소요되므로 결국 실시간 일정계획이 가능하다.

본 연구의 추후 연구 과제는 먼저 주문정보 특징을 일반화시키는 방법을 개선하는데 있다. 본 연구에서는 그 과정에 상관분석을 사용하였는데 그 외에도 분산분석이나 회귀분석과 같은 다른 기법의 사용도 고려해 볼만하다. 또한 사실베이스를 지식베이스로 전환하는 과정에 사용된 vote 필드에 관한 추가적인 연구도 필요하다. vote 필드의 누적 값이 일정 수준 이하인 조합들에 대해서는 그 결과의 신뢰성을 보완하기 위한 연구가 필요하다. 그리고 본 연구에서 사용한 방법과는 반대로 발생 가능한 모든 조합별로 일정횟수 이상의 실험을 반복하여 모든 조합에 대한 최적의 규칙을 완전하게 만들어 두는 방법도 연구할 가치가 있을 것으로 생각된다.

참고 문헌

- [1] Badiru, A. B., *Expert Systems Applications in Engineering and Manufacturing*, Prentice Hall 1992.
- [2] Baker, K. R., *Introduction to Sequencing and Scheduling*, John Wiley & Sons, 1974.
- [3] Cai, Y., Cercone, N. and Han, J., "Attribute-Oriented Induction in Relational Databases," *Knowledge Discovery in Databases*, Piatetsky-Shapiro, G. and Frawley, W. J., eds., AAAI/MIT Press, 1991, pp. 213-228.
- [4] Chang, Y., Sueyoshi, T. and Sullivan, R. S., "Ranking Dispatching Rules by Data Envelopment Analysis in a Job shop Environment," *IIE Transactions*, Vol. 28, 1996, pp. 631-642.
- [5] Conway, R. W., Maxwell, W. L. and Miller, L. W., *Theory of Scheduling*, Addison-Wesley, 1967.
- [6] Dagli, C. H. and Poshyanonda, T., "Knowledge-Based Scheduling and Control," *People and Product Management in Manufacturing*, Edosomwan, J. A., ed., Elsevier, 1990, pp. 211-239.
- [7] Frawley, W. J., Piatetsky-Shapiro, G. and Matheus, C. J., "Knowledge Discovery in Databases: An Overview," *Knowledge Discovery in Databases*, Piatetsky-Shapiro, G. and Frawley, W. J., eds., AAAI/MIT Press, 1991, pp. 1-27.
- [8] French, S., *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*, Ellis Horwood, 1982.
- [9] Han, J., Huang, Y., Cercone, N. and Fu, Y., "Intelligent Query Answering by Knowledge Discovery Techniques," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 3, 1996, pp. 373-390.
- [10] Johannsen, G. and Alty, J. L., "Knowledge Engineering for Industrial Expert Systems," *Automatica*, Vol. 27, No. 1, 1991, pp. 97-114.
- [11] Kim, K. T., "A Study on Development of a Knowledge Based Scheduler and Knowledge Acquisition through Simulation," *New Direction in Simulation for Manufacturing and Communications*, 1994, pp. 282-286.
- [12] Mallach, E. G., *Understanding Decision Support Systems and Expert Systems*, Irwin, 1994.
- [13] Stonebraker, M., "The Integration of Rule Systems and Database Systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 5, 1992, pp. 415-423.
- [14] Sule, D. R., *Industrial Scheduling*, PWS, 1997.
- [15] Turban, E., *Decision Support and Expert Systems: Management Support Systems*, Prentice Hall. 4th ed.. 1995.