İ.Ü. İşletme Fakültesi Dergisi, C: 25, S: 2/Kasım 1996, s.13-34

DESIGN OF EXPERT SYSTEMS FOR JOB SHOP SCHEDULING: A CONCEPTUAL FRAMEWORK

Güneş Gençyılmaz
İstanbul University
Faculty of Business Administration
Department of Production
Avcılar, Istanbul
Mehmet Fatih Taşgetiren
Sakarya University
Industrial Engineering Department
Esentepe Campus/ADAPAZARI

Abstract

This paper investigates the current expert scheduling approaches to determine the research areas for the encouragement of the use of expert systems in production scheduling. A framework is presented to clarify the state-of-the-art for the expert scheduling systems. The rule-based representation of the knowledge base, the rule generation scheme for it, and foward backward chaining procedures for the inference engine are explained to show how an expert system functions in a job-shop environment.

1. Introduction

Scheduling of jobs through machines in a job shop is one of the most complex problems, which is basically concerned with assigning jobs to machines and then sequencing them on machines accordingly, thus resulting in a schedule which is both compatible with the technological constraints imposing a route for each job on machines and optimal with respect to some performance criterion identifying objectives of scheduling. Scheling objectives such as minimisation of flow time, minimisation of work-in-process, minimisation of mean tardiness, ete. are generally conflicting each other and vary from one industry to another. It is therefore important to find a good solution that will in turn reduce the scheduling cost significantly.

Scheduling of n jobs through m machines results in (n!) possible schedules among which exist one or more optimal schedules according to a performance criterion employed (French, 1982). These optimal schedules can theoretically be found in a finite number of computations indicating direct search by complete enumeration which is impractical even with the use of computers. For example, a simple problem having 5 jobs and 5 machines requires (5!) possible alternatives to be evaluated. In addition, Rinnooy Kan (1976) reports that most of the scheduling problems are NP-Hard including n-job m-machine problems. These problems have exponential time complexity function meaning that the maximum number of operations required to solve the problem exhibit an exponential behaviour. For this reason, it is impractical to solve these problems by not only complete enumeration methods but also implicit enumeration methods such as branch and bound or dynamic programming reducing the number of alternatives to be searched.

Due to the fact that realistic size and complexity of scheduling problems have caused the efforts to fail in finding an optimal solution baved on traditional optimisation techniques, research in schduling theory has been directed towards using heuritic algorithms which produce near optimal schedules and reduce computational time, because they avoid searhing for all alternatives by applying priority rules for job assignment. In these approaches, priority rules are compared with each other under a performance criterion to measure the effectiveness of schedules in meeting a specified scheduling objective. However, these appoaches usually aim at optimizing a single goal function with a single parameter and find it difficult to achieve multiple objectives (Charalambous and Hindi, 1991, Noronho and Sarma, 1991). In addition, scheduling environment is generally surrounded by accumulated 'knowledge' that will affect scheduling decisions as to which job should be selected as the next action according to the current state of system(Cheng, 1985, Chow and Huang, 1990). Traditional optimisation tecniques and heuristic approaches deal with only a portion of knowledge accumulated and cannot respond to the actual state of system in dynamic schduling environment. Construction of a schedule based on all the relevant scheduling information can be achieved by the use of expert systems inferring solutions from structured knowledge.

2. Expert Scheduling Approaches

Through the use of knowledge gained by trial and error, human schedulers obtain relatively good results in developing schedules that effectively meet the objectives even though they are not optimal, but near optimal. In addition, schedulers respond to the unexpected factors such as machine failures, new orders, urgent jobs, ete. and modify the original schedule to cope with these issues without jeopardizing the acceptable performance criterion related with the production objecties. However, human experts find it difficult to construct an adequate schedule in the case of more complex siuations, and their ability is limited to a solution that will satisfy the overall objective. This brings about the attention of researchers to utilize artificial intelligence tecniques such as expert systems to solve the problems mentioned above.

Recently, a number of expert scheduling systems have been developed. These systems are generally classified into two categories (MacFarland and Grand, 1987):

1. Real- Time Scheduling Systems,
2. Job Assignment Systems.

Real-time scheduling systems are applications of artificial intelligence to scheduling problems. The idea behind these systems is that when a machine becomes free, an expert system is invoked to determine the next action to be taken depending on the system state defined by the status of the machines in the shop floor. The general structure of such a system is illustrated in Figure 1.

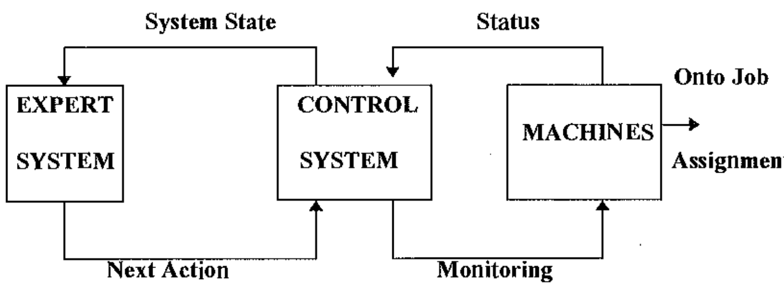

FIGURE 1. THE STRUCTURE OF REAL TIME EXPERT SCHEDULING SYSTEMS.

As shown in Figure 1, the control system monitors the machines and gathers information to determine their status. Then it conveys these information to the expert system as the system state, and based on the system state, the expert system executes a set rules to determine the next action to be taken. The next action will probably be a selection of next job to be processed on the free machine.

There are some advantages of this kind of systems. Because of the dynamic nature of the system, adjustments for new orders, urgent jobs and machine failures, ete. are dynamically performed at the shop floor. Multiple objectives can be achieved since the scheduling decision is made based on the system state implying which performance criterion and priority rule would be suitable for the current status of the machine. However, the main disadvantage of these systems is due to the fact that they basically rely on the machine status which is only a portion of the knowlledge accumulated in the scheduling environment, thus dealing with only a portion of the production objectives too. Furthermore, performance measure cannot be computed beforehand for a specific priority rule in order to compare the shedules with one another because of the use of a different priority rule in each step of job assignment according to the system state implied by the control system.

In job assignment systems, jobs are assigned to machines at certain times. The adaptation of artifical intelligence technology as expert systems to scheduling problems in the context of job assignment models created new schemes for the implementation of traditional algorithms and heuristic approaches in such a way that jobs are evaluated in a prioritized order and assigned to machines so as to from a schedule according to the priority index determined by the knowledge of the expert system. These approaches are not new and have been applied to scheduling problems without the use of expert systems. However, the use of expert systems have led to more understanable and flexible heuristic models by means of the presentation of sophisticated knowledge representation schemes for building the knowledge base. The general structure of a job assignment model employing an expert system is illustrated in Figure 2.

The expert system in this approach functions by executing a set of rules to put the jobs in a prioritized order for shop loading where the jobs are allocated to some specific machines in a queue. It also selects a job from the queue to be assigned to a specific machine. So, both shop loading and sequencing can be conducted by the expert systems.

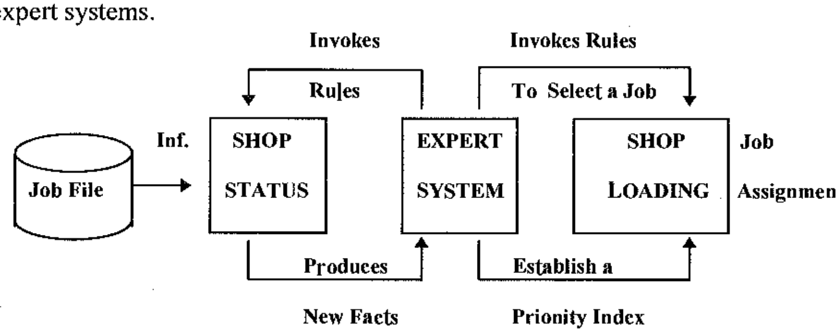

FIGURE 2. JOB ASSIGNMENT SYSTEMS EMPLOYING AN EXPERT SYSTEM.

This approach to scheduling problem has an advantage of producing a schedule that can be predicted beforehand. This feature provides the schedulers with the comparison of alternative schedules for different priority rules under a given performance criterion in order to decide on which schedule is dominating. However, this approach can only be applied to some specific classes of problems, and the scheduling decisions are sequential which may lead to a bad schedule due to an unwise job assignment early in the beginning of the schduling period. In addition, it is almost the same as employing a heuristic approach except for the existence of the rule set embedded in the knowledge base.

3. Rule Generation

As metioned before, decision as to which priority rule should be employed for job assignment is vital to the success of the system developed. Proper decision for varying state of system in each step of job assignment improves the performance of schedules. This is the main advantage of expert system considering the actual state of system which covers the entire knowledge related with the scheduling environment. Furthermore, this feature leads to the satisfaction of the satisfaction of multiple objectives by adapively changing the scheduling goals according to the current state of the system.

How these advantages can be utilized for a schduling application depends on how to generate the proper production rules for the knowledge base. It is possible to generate the production rules by a sound definition of the system state for the shop floor. As shown in Figure 3, system state is usully described in terms of the status of three objects(Kusiak and Chen, 1988):

- Job Status,
- Shop Loading Statuus,
- Machine Status,

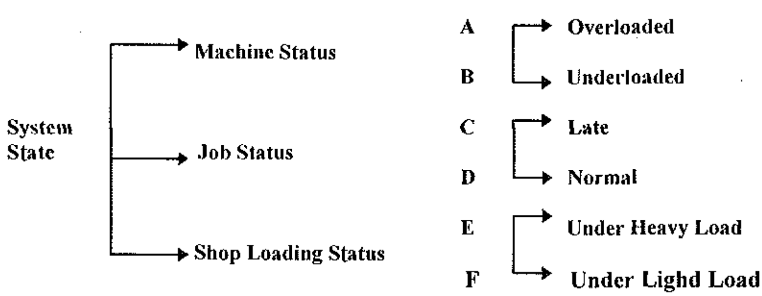

FIGURE 3. DEFINITION OF SYSTEM STATE.

The machine status can be characterized by the following two attributes: overloaded or underloaded. Job status can be defined as late or normal. Finally, shop loading status can be defined in two attributes: under heavy load or under light load.

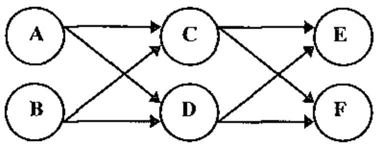

FIGURE 4. DECISION TREE FOR SYSTEM STATE.

As shown in Figure 4, establishing a decision tree by the combinations of attributes for each status will result in the generation of production rules. Each path in the decision tree impiies a production rule which would be partitioned into a set of rule base in order to enrich the explanation capabihty ofi expert systems. The question of which priority rule should be used for each path in the decision tree is the crucial point of generating production rules. For example, if the path is ACE, in other words, if the machine is overloaded and/or the job is late and/or the shop is under heavy load, then which priority rule should be used for job assignment ? Once an answers, say the SPT rule, is found to this question, it is placed into the THEN clause of the production rule.This procedure can be conducted for each path in the decision tree to set up the production rules for the knowledge base. An example is given in Figure 5, which is implemented in the work by Gençyıhnaz and Taşgetiren (1996).

	ACE		**SPT**
	ACF		**SPT**
	ADE		**SPT**
	ADF		**SPT**
I		**THEN**	
	BCE		**COVERT**
	BCF		**COVERT**
	BDE		**S/OPN**
	BDF		**S/OPN**

FIGURE 5. RULE GENERATİON SCHEME FOR KNOWLEDGE BASE

4. Franmework For Expert Scheduling Systems

A typical expert scheduling system is illustrated in Figure 6. There are four basic elements of an expert scheduling system: knowledge base, inference engine or reasoning mechanism, data manipulation, and scheduler.

The knowledge base contains the knowledge about the rules, facts and other information regarding the job shop problem. In a rule based representation, the facts are simply the statements which will be used by the rules to manipulate and derive some other facts reiated to the problem domain. For example,

F- Average **Slack** time of **Job =10**

is a fact implying the status of the job. The rules are the conditional sistements of knowledge in the form of **IF.....THEN** clouses. For example,

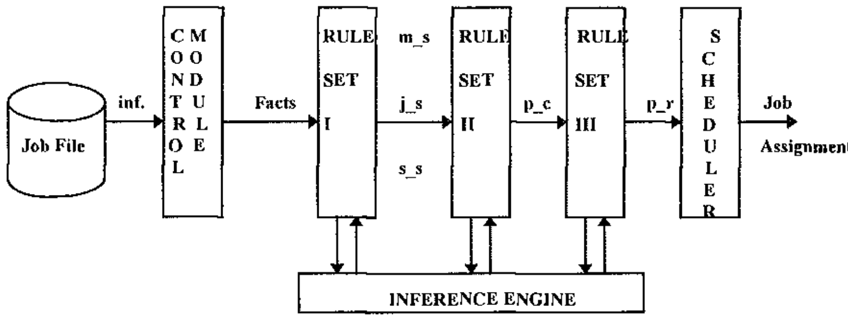

FIGURE 6. STRUCTURE OF EXPERT SCHEDULING SYSTEM

R- IF Average **Slack Time of Job <=20**
THEN Job Status=Late.

It is important to realize that the above scheduling rule, R, matches the fact, F, and produces another fact from the THEN clause of the production rule of R as follows:

FF- Job Status=Late.

This process is called inference engine or reasoning mechanism that any expert system should possess. So, the reasoning mechanism examines the facts, seans and searehes the knowledge base and fires the appropriate rules employed in the knowledge base according to the procedure established in advauce. Pham and Pham (1988) explain that the reasoning process can be managed in different ways depending on the control procedure that the expert system has. One strategy is to start with a given set of facts or given data and search for the **'IF'** portion of the rule which matches the fact. When such a rule is fired to generate a new fact which in turn causes other rules to be fired. The reasoning process stops when no more rules to be executed. This kind of process is called 'forward chaininge'. An example of forward chaining is given in Appendix A. Another strategy is to begin with a goal to be tested or proved by the rules whether or not it is satisfied with a given fact. The test is conducted by matehing the **'THEN'** portion of the rule with the desired goal stack to produce sub-goals until a sub-goal ssatisfied by a given fact is found. The reasoning prosoning process ends when the original goal is satisfied. This process is called 'backward chaining' which reduces the search tree if the knowledge base is enriched by the additional rules for the same problem domain. An examplle of the backward cchaining in the scheduling context is given in Appendbx B. Note that in Appendix B, the fact, **'Queue Time of Jobs=Very high'**, is not produced because it is irrelevant to the desired goal, **'Priority Rule=SPT'**, Backward chaining is preferred when the number of rules is large and forward chaining would cause a combinatorial explosion(Pham and Pham, 1988).

The data manipulation module monitors the job file to obtain necessary information or initial facts to be conveyed to the reasoning module inferencing the knowledge base. It can simply viewed as data collection module that derives the facts or information to be stored in the knowledge base. Another function of this module is that when a job assignment is made by the scheduler module, the feedback obtained from the scheduler is used to update the job file. Suppose that it gathered the fact, F1, from the job file as follows:

F1- Number of Job in The Queue of The Machine=7.

As shown in Figure 6, the knowledge base contains three rule sets: the state identification rules, the objective determination rules, and the decision making rules. The state identification rules determine the system state by interacting the reasoning module. An example of a rule which is incorporated in the state identification rules is:-

R1- IF Number of Job in The Queue of The Machine≥5
THEN Machine Status=Overloaded.

The reasoning module interacts with the first rule set and infers a new fact, F2, by matching the fact, F1, with the 'IF' portion of the rule, R1. So, the new fact, F2, from the 'THEN' portion of the rule, R1, is produced as follows:

F2- Machine Status=Overloaded.

Objective determination rules direct the scheduling decision toward the objective implied by the system state. An example of the objective determination rules is:-

R2- IF Machine Status=Overloaded
THEN Performance Criterion=Mean Flow Time.

Again, the reasoning module matches the fach, F2, with the 'IF' portion of the rule, R2 and produces another fact, F3, as follows:

F3- Performance Criterione=Mean Flow Time.

Finaliy, the decision making rules judge on the priority rule that will be used by the scheduler module for job assignment. An example of the decision making rules is:-

R3- IF Performance Criterion=Mean Flow Time
THEN Priority Rule=SPT.

As explained before, the reasoning mechanism matches the fact, F3, with the 'IF' portion of the rule, R3, and derives the fact, F4, as follows:

F4- Priority Rule=SPT.

This final fact indicates the priority rule to be employed for job assignment. The scheduler module considers this recommendation and implements the decision by scheduling the job with the shortest processing time.

The other parts of the system state, that is; the job status and the shop loading status can be determined in a similar way by considering the average slack time of each job and the number of operations scheduled respectively. How necessary conchısions are derived from each status of system state is illustrated in Appendix C and D.

5. Conclusion

In this paper, the structure of expert systems for job shop scheduling is outlined to demonstrate the feasibility of building it. The advantage of expert systems is due to the cosideration of the actual system state which covers the endire knowledge related with the scheduling environment. This feature leads to the satisfaction of multiple objectives by adaptively changing the scheduling goais and employing different prioity rules for job assignment.

Another advantage is that the modification of the knowledge base is easily managed by simply adding, deleting or changing the rules in the knowledge base without destroying other modules of the system. This feature provides us with the capability of adopting the system to the different job shops.

The effectiveness of the schedule can only be measured under a performance criterion for the entire system without testing the effects of the priority rules individually. Since the job assignment is made according to the current state of the system, the comparison of the priority rules cannot be made to determine which priority rule is good or bad as the scheduling process goes on.

Another disadvantage is that the knowledge base is dependent on the system developed. In other words, the parameters of the state identification rules are determined for the specific job shop developed. For other shops, these parameters should be modified for application.

APPENDIX A
FORWARD CHAINING

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1- IF Number of Jobs In The Queue of The Machine>10
THEN Machine Status=Overloaded.
R2- IF Machine Status=Overloaded
THEN Performance Criterion=Mean Flow Time.
R3- IF Machine Status=Overloaded
THEN Queue Time of Jobs=Very High.
R4- IF Performance Criterion=Mean Flow Time
THEN Priority Rule= SPT.

⇓

F1 and R1 match and F2 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status=Overloaded.

Rules:

R1- IF Number of Jobs In The Queue of The Machine>10
THEN Machine Status=Overloaded.
R2- IF Machine Status=Overloaded
THEN Performance Criterion=Mean Flow Time.
R3- IF Machine Status=Overloaded
THEN Queue Time of Jobs=Very High.
R4- IF Performance Criterion=Mean Flow Time
THEN Priority Rule=SPT.

⇓

F2 and R2 match and F3 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=12
F2- Machine Status=Overloaded.
F3- Performance Criterion=Mean Flow Time.

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT

⇓

F2 and R3 match and F4 is derived.

<u>Facts</u>:

F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status= Overloaded.
F3- Performance Criterion=Mean Flow Time.
F4- Queue Time of Jobs=Very High.

<u>Rules</u>:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

⇓

F3 and R4 match and F5 is derived.

<u>Facts</u>:

F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status=Overloaded.
F3- Performance Criterion=Mean Flow Time.
F4- Queue Time of Jobs=Very High.
F5- Priority Rule=SPT.

<u>Rules</u>:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Mahchine Status=Overloaded
	THEN	Queue Time of Jobs= Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

APPENDIX B
BACKWARD CHAINING

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Ruie=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No

⇓

G1 and R4 match and G2 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	

⇓

G2 and R2 match and G3 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded

	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	No
G3- Machine Status=Overloaded.	No

⇓

G3 and R1 match and G4 is devrived.

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1-	**IF**	Number of Jobs in The Queue of The Machine>10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	No
G3- Machine Status=Overloaded.	No
G4- Number of Jobs In The Queue of The Machine>10	No

⇓

G4 and F1 match and G4 is satisfied.

Facts:

F1- Number of Jobs In The Queue of The Machine=12

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine >10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean Flow Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.

R4- IF Performance Criterion=Mean Flow Time
THEN Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	No
G3- Machine Status=Overloaded.	No
G4- Number of Jobs In The Queue of The Machine>10	Yes

⇓

F1 and R1 match and F2 si derived.

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status=Overloaded.
<u>Rules</u>:
R1- IF Number of Jobs In The Queue of The Machine>10
THEN Machine Status=Overloaded.
R2- IF Machine Status=Overloaded
THEN Performance Criterion=Mean Flow Time.
R3- IF Machine Status=Overloaded
THEN Queue Time of Jobs=Very High.
R4- IF Performance Criterion=Mean Flow Time
THEN Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	No
G3- Machine Status=Overloaded.	Yes

⇓

F2 and R2 match and F3 is derived.

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status=Overloaded.
F3- Performance Criterion=Mean Flow Time.
<u>Rules</u>:
R1- IF Number of Jobs In The Queue of The Machine >10
THEN Machine Status=Overloaded.
R2- IF Machine Status=Overloaded
THEN Performance Criterion=Mean Flow Time.
R3- IF Machine Status=Overloaded

	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion= Mean Flow Time
	THEN	Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	No
G2- Performance Criterion=Mean Flow Time.	Yes

⇓

F3 and **R4** match and **F4** is derived.

<u>Facts</u>:

F1- Number of Jobs In The Queue of The Machine=12.
F2- Machine Status=Overloaded.
F3- Performance Criterion=Mean Flow Time.
F4- Priority Rule=SPT.

<u>Rules</u>:

R1-	**IF**	Number of Jobs In The Queue of The Machine >10
	THEN	Machine Status=Overloaded.
R2-	**IF**	Machine Status=Overloaded
	THEN	Performance Criterion=Mean **Flow** Time.
R3-	**IF**	Machine Status=Overloaded
	THEN	Queue Time of Jobs=Very High.
R4-	**IF**	Performance Criterion=Mean Flow Time
	THEN	Priority Rule=SPT.

Goal Stacks:	Satisfied
G1- Priority Rule=SPT.	Yes

APPENDIX C
FORWARD CHAINING

<u>Facts</u>:

F1- Number of Jobs In The Queue of The Machine=10
F2- Average Slack Time=15
F3- Number of Operations Scheduled=170

<u>Rules</u>:

R1-	**IF**	Number of In Jobs The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time <20
	THEN	Job Status=Late.
R3-	**IF**	Number of Operations Scheduled ≥100 **OR**

Number of Operations Scheduled ≤800
THEN Shop Loading Status=Under Heavy Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Late **AND**
Shop Loading Status=Under Heavy Load
THEN Performance Criterion=Mean Tardiness.
R5- IF Performance Criterion=Mean Tardiness
THEN Priority Rule=COVERT.

⇓

F1 and R1 match and F4 is derived.

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=15.
F3- Number of Operations Scheduled=170.
F4- Machine Status=Underloaded.

<u>Rules</u>:
R1- IF Number of Jobs In The Queue of The Machine ≤10
THEN Machine Status=Underloaded.
R2- IF Average Slack Time<20
THEN Job Status=Late.
R3- IF Number of Operations Scheduled ≥100 **OR**
Number of Operations Scheduled ≤800
THEN Shop Loading Status=Under Heavy Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Late **AND**
Shop Loading Status=Under Heavy Load
THEN Performance Criterion=Mean Tardiness.
R5- IF Performance Criterion=Mean Tardiness
THEN Priority Rule=COVERT.

⇓

F2 and R2 match and F5 is derived.

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=15.
F3- Number of Operations Scheduled=170.
F4- Machine Status=Underloaded.
F5- Job Status=Late.

<u>Rules</u>:
R1- IF Number of Jobs In The Queue of The Machine ≤10
THEN Machine Status=Underloaded.
R2- IF Average Slack Time <20
THEN Job Status=Late.
R3- IF Number of Operations Schduled ≥100 **OR**

		Number of Operations Schduled ≤800
	THEN	Shop Loading Status=Under Heavy Load.
R4-	**IF**	Machine Status=Underloaded **AND**
		Job Status=Late **AND**
		Shop Loading Status=Under Heavy Load
	THEN	Performance Criterion=Mean Tardiness.
R5-	**IF**	Performance Criterion=Mean Tardiness
	THEN	Priority Rule=COVERT.

⇓

F3 and R3 match and **F6** isderived.

Facts:

F1- Number of Jobs In The Queue of The Machine=8

F2- Average Slack Time=15

F3- Number of Operations Scheduled=170

F4- Machine Status=Underloaded.

F5- Job Status=Late.

F6- Shop Loading Status=Under Heavy Load.

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time <20
	THEN	Job Status=Late.
R3-	**IF**	Number of Operations Scheduled ≥100 **OR**
		Number of Operations Scheduled ≤800
	THEN	Shop Loading Status=Under Heavy Load.
R4-	**IF**	Machine Status=Under **AND**
		Job Status=Late **AND**
		Shop Loading Status=Under Heavy Load
	THEN	Performance Criterion=Mean Tardiness.
R5-	**IF**	Performance Criterion=Mean Tardiness
	THEN	Priority Rule=COVERT.

⇓

F4, F5 and **F6** match with R4 and F7 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=8.

F2- Average Slack Time=15.

F3- Number of Operations Scheduied=170.

F4- Machine Status=Underloaded.

F5- Job Status=Late.

F6- Shop Loading Status=Under Heavy Load.

F7- Performance Criterion=Mean Tardiness.

Rules:

R1-	**IF**	Number of Jobs In The Machine ≤10

THEN Machine Statuse=Underloaded.
R2- IF Average Slack Time <20
THEN Job Status=Late.
R3- IF Number of Operations Scheduled ≥100 **OR**
Number of Operations Scheduled ≤800
THEN Shop Loading Status=Under Heavy Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Late **AND**
Shop loading Status=Under Heavy Load
THEN Performance Criterion=Mean Tardiness.
R5- IF Performance Criterion=Mean Tardiness
THEN Priority Rule=COVERT.

⇓

F7 and R5 match and **F8** is derived.

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=15
F3- Number of Operations Scheduled=170.
F4- Machine Status=Underloaded.
F5- Job Status=Late.
F6- Shop Loading Status=Under Heavy Load.
F7- Performance Criterion=Mean Tardiness.
F8- Priority Rule=COVERT.

<u>Rules</u>:
R1- IF Number of Jobs In The Queue of The Machine ≤10
THEN Machine Status=Underloaded.
R2- IF Average Slack Time <20
THEN Job Status=Late.
R3- IF Number of Operations Scheduled ≥100 **OR**
Number of Operations Scheduled ≤800
THEN Shop Loading Status=Under Heavy Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Late **AND**
Shop Loading Status=Under Heavy Load
THEN Performance Criterion=Mean Tardiness.
R5- IF Performance Criterion=Mean Tardiness
THEN Priority Rule=COVERT.

APPENDIX D
FORWARD CHAINING

<u>Facts</u>:
F1- Number of Jobs In The Queue of The Machine=8.

F2- Average Slack Time=25
F3- Number of Operations Scheduled=65

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time ≥20
	THEN	Job Status=Normal.
R3-	**IF**	Number of Operations Scheduled <100 **OR** Number of Operations Scheduled >800
	THEN	Shop Loading Status=Under Light Load.
R4-	**IF**	Machine Status=Underloaded AND Job Status=Normal AND Shop Loading Status=Under Light Load
	THEN	Performance Criterion=Machine Utilization.
R5-	**IF**	Performance Criterion=Machine Utilization
	THEN	Priority Rule=S/OPN.

⇓

F1 and R1 match and **F4** is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=25
F3- Number of Operations Scheduled=65.
F4- Machine Status=Underloaded.

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time ≥20
	THEN	Job Status=Normal.
R3-	**IF**	Number of Operations Scheduled <100 **OR** Number of Operations Scheduled >800
	THEN	Shop Loading Status=Under Light Load
R4-	**IF**	Machine Status=Underloaded AND Job Status=Normal AND Jop Loading Status=Under Light Load
	THEN	Performance Criterion=Machine Utilization
R5-	**IF**	Performance Criterion=Machine Utilization
	THEN	Priority Rule=S/OPN.

⇓

F2 and R2 match and F5 is derived.

Facts:

F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=25
F3- Number of Operations Scheduled=65.

F4- Machine Status=Underloaded.
F5- Job Status=Normal.

<u>**Rules:**</u>

R1- IF Number of Jobs In The Queue of The Machine ≤10
THEN Machine Status=Underloaded.
R2- IF Average Slack Time ≥20
THEN Job Status=Normal.
R3- IF Number of Operations Scheduled <100 **OR**
Number of Operations Scheduled >800
THEN Shop Loading Status=Under Light Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Normal **AND**
Shop loading Status=Under Light Load
THEN Performance Criterion=Machine Utilization.
R5- IF Performance Criterion=Machine Utilization
THEN Priority Rule=S/OPN.

⇓

F3 and R3 match and F6 is derived.

<u>Facts:</u>

F1- Number of Jobs In The Queue of The Machine=8.
F2- Average Slack Time=25
F3- Number of Operations Scheduled=65.
F4- Machine Status=Underloaded.
F5- Job Status=Normal.
F6- Shop Loading Status=Under Light Load.

<u>**Rules:**</u>

R1- IF Number of Jobs In The Queue of The Machine ≤10
THEN Machine Status=Underloaded.
R2- IF Average Slack Time ≥20
THEN Job Status=Normal.
R3- IF Number of Operations Scheduled <100 **OR**
Number of Operations Scheduled >800
THEN Shop Loading Status=Under Light Load.
R4- IF Machine Status=Underloaded **AND**
Job Status=Normal **AND**
Shop Loading Status=Under Light Load
THEN Performance Criterion=Machine Utilization.
R5- IF Performance Criterion=Machine Utilization
THEN Priority Rule=S/OPN.

⇓

F4, F5 and F6 match with R4 and F7 is derived.

<u>Facts:</u>

F1- Number of Jobs In The Queue of The Machine=8.

F2- Average Slack Time=25
F3- Number of Operations Scheduled=**65**.
F4- Machine Status=Underloaded.
F5- Job Status=Normal.
F6- Shop Loading Status=Under Light Load.
F7- Performance Criterion=Machine Utilization.

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time ≥20
	THEN	Job Status=Normal.
R3-	**IF**	Number of Operations Scheduled <100 **OR**
		Number of Operations Scheduled >800
	THEN	Shop Loading Status=Under Light Load.
R4-	**IF**	Machine Status=Underloaded **AND**
		Job Status=Normal **AND**
		Shop Loading Status=Under Light Load
	THEN	Performance Criterion=Machine Utilization.
R5-	**IF**	Performance Criterion=Machine Utilization
	THEN	Priority Rule=S/OPN.

⇓

F7 and R5 match and **F8** is derived.

Facts:

F1- Number Jobs In The Queue of The Machine=8.
F2- Average Slack Time=25
F3- Number of Operations Scheduled=65.
F4- Machine Status=Underloaded.
F5- Job Status=Normal.
F6- Shop Loading Status=Under Light load.
F7- Performance Criterion=Machine Utilization.
F8- Priority Rule=S/OPN.

Rules:

R1-	**IF**	Number of Jobs In The Queue of The Machine ≤10
	THEN	Machine Status=Underloaded.
R2-	**IF**	Average Slack Time ≥20
	THEN	Job Status=Normal.
R3-	**IF**	Number of Operations Scheduled <100 **OR**
		Number of Operations Scheduled >800
	THEN	Shop Loading Status=Under Light Load.
R4-	**IF**	Machine Status=Underloaded **AND**
		Job Status=Normal **AND**
		Shop Loading Status=Uunder Light Load
	THEN	Performance Criterion=Machine Utilization.

R5- IF Performance Criterion=Machine Utilization
THEN Priority Rule=S/OPN.

References

A. Kusiak and M. Chen, 'Expert Systems for Plannig and Scheduling Manufacturing Systems', European Jornal of Operational Research, North-Holland, 113-130, 34(1988).

D. G. MacFarland, F. H. Grant, 'Artificial Intelligence In Sequencing and Scheduling', IIE Integrated Systems Conference Proceedings, 1987.

D. T. Pham and P. T. N. Pham, 'Expert System in Mechanical and Manufacturing Engineering', The International Journal of Manufacturing Technolohy, IFC Publications, 3(3), 3-21, 1988.

Feng-Chang Chang, 'A Knowled-Based Real Time Decision Support System For Job Shop Scheduling At the Shop Floor Level', Ph.D Dissertation, The Ohio State University, 1985.

G. Gençyılmaz and M. F. Taşgetiren, 'Proceedings of The First Turkish Symposium On Intelligenl Mamufacturing Systems', 30-31 May 1996: 75-84, Sakarya Universty.

L. R. Chow and H. S. Huang, The Design of An Intelligent Job Scheduler', Information and Managemct Sciences, Vol. 1, No. 1, 1990.

O. Charalambous and K. S. Hindi, 'A Rewiew of Artificial Intellihence-Based Job Shop Scheduling Systems', Informmation on Decision Technologies, North-Holland, 189-202, 17(1991).

Rinnooy Kan, 'Machine Scheduling Problems: Classifications, Complexity and Computations', Martinus Nijihoff, The Hauge, 1976.

S. French, 'Sequencing and Scheduling: An Introduction to Mathematics of The Job Shops', Ellis Harwood Series, West Sussex, London, 1982.

S. J. Noronha and S. Sarma,'Knowledge-Based Approaches for Scheduling Problems: A Survey', IEEE Transaction on Knowledge and Data Engineering, Vol3, No. 2, 1991.