

O. Bilkay · O. Anlagan · S. E. Kilic

Job shop scheduling using fuzzy logic

Received: 4 August 2002 / Accepted: 3 March 2003 / Published online: 14 February 2004
© Springer-Verlag London Limited 2004

Abstract In a flexible manufacturing system (FMS) scheduling problems become extremely complex, even for simple breakdowns, when dynamic uncertainties such as machine breakdowns and the uncertain arrival of jobs are taken into consideration. In the first stage of this study, a fuzzy logic-based algorithm for assigning priorities to part types that are to be machined is proposed. In the second stage, an operation-machine allocation and scheduling algorithm is presented. A criteria contribution equalizer is used in decision-making. The proposed algorithm can re-generate the schedule in case of a machine breakdown, and therefore can be used as an on-line controller. The system architecture and linguistic variables are presented and results showed that the proposed algorithm improves the system efficiency.

Keywords Part type priority · Scheduling · FMS · Fuzzy logic · Dispatching · Fuzzy scheduling

1 Introduction

In dynamic production environments, i.e. in flexible manufacturing systems (FMSs) that are subject to limited resources, random machine failures or multiple production criteria, the problem of controlling and scheduling the production process is best tackled by a synergy of the computer's scheduling algorithms and the production planner's effective internal heuristics. In this "interactive scheduling", the production planner remains in control and is able to control the scheduling process by using experience and intuition via computer support. In other words, the scheduling system should act as a decision support system for the production

planner. However, references in the literature to practical systems where this interaction has been realised are rare, and the models tend to be oversimplified [1].

According to Saygin and Kilic [2], the existing software tools are typically too slow and cannot react to the changing shop floor conditions, are based on simplistic formulations of reality that ignore important constraints, are based on a single objective function or simplistic trade-offs and are difficult to install and integrate into pre-existing commercial shop floor systems.

Much of the research in manufacturing systems has focused on the development of algorithms for scheduling and control. The emphasis is on the development of algorithms that can lead to the full automation of the FMS scheduling and control operation. However, fully automated FMS control is a highly complex task due to the dynamic structure of an FMS and due to changing production goals. The automatic control system has to cope with unanticipated events such as machine breakdowns or late parts. Current and proposed automatic control systems lack adaptability and intelligence to cope with such deficiencies. Human-supervised scheduling and control of FMSs appears to be a necessary and desirable alternative. Supervisory control of a simulated FMS was studied experimentally by Dunkler et al. [3]. Their experimental results showed that with human supervision both due date and inventory performance of the FMS can be improved.

Traditional scheduling tools for continuous process industries have not been developed with batch processing in mind. Efficient batch processing requires intelligent scheduling tools. Dessouky et al. [4] studied the development of an architecture which combines intelligent tools with a virtual plant for performing management tasks. The scheduling algorithm uses a database of current customer orders and a knowledge base that contains rules about customer priority, equipment cleaning, process preference, equipment history, batch sequencing rules, cost history, cost estimates and other knowledge to aid the user in the selection of the appropriate schedule. The uniqueness of this approach is the integration of a

O. Bilkay (✉) · O. Anlagan · S. E. Kilic
Department of Mechanical Engineering,
IMTRG, METU, 06531 Ankara, Turkey
E-mail: bilkay@encon.com.tr
Tel.: +90-312-4477122
Fax: +90-312-4476988

virtual plant, which can be viewed as a static plant simulation model, with a rule-based scheduling approach.

For the efficient operation and implementation of an FMS, varieties of decision problems need to be addressed. These decision problems are broadly classified into pre-release decision problems and post-release decision problems. Pre-release decisions refer to the FMS planning problem, which deals with the pre-arrangement of part types and tools before it begins to process; whereas, post-release decisions refers to the FMS scheduling problem, which considers the sequencing and routing of part type at the time the system is in operation.

Many job shop scheduling approaches have been reported in the literature, but most of these approaches are non-analytical. The reason for this is that the scheduling problem becomes extremely complex, even for simple breakdowns, when dynamic uncertainties such as machine breakdowns and the uncertain arrival of jobs are taken into consideration [5].

Part type selection is one of the FMS planning problems, which partitions the production requirements into batches of part types for simultaneous manufacturing, where part input sequencing is one of the FMS scheduling problems, which determines the sequence and schedule of the release of the part into the system.

A part type selection decision involves a number of objectives and constraints that are imprecise and conflicting in nature. Although machine capacity might be sufficient as overloading of the machines on the system is permitted to a small extent, it may not be possible to process all the part type orders required in a particular planning period due to a limited number of tool slots and available processing time. Thus, a subset of part type orders has to be selected and the required tools allocated to the machines before the orders are to be processed. A part type has generally more than one operation and each operation can be performed on one or more alternative machines.

The part type selection and machine allocation decisions are connected with making certain decisions based on the input data. The input data are often imprecise or uncertain and in some situations are the result of the decision-maker's subjective assessment. On the other hand, the systems considered are often so complex that it is impossible to determine all the relations that exist within them; thus, it is impossible to determine a good model of a real system. Analysing a great number of variables, under the condition that when the variables are defined imprecise data and when functional relations among them cannot be determined, becomes very complicated and practically impossible using classical mathematical models. Making such a decision requires a high computational time, and it is meaningless to calculate all possibilities. Instead, part types should be selected by a proper decision-making process. One of the methods suitable for the consideration of uncertainty and subjectivity is based on fuzzy logic. Models based on fuzzy logic consist of "IF-THEN" rules. The main idea is to develop a model, which simulates the decision-making

process without determining the exact functional relations between input and output variables [6].

Fuzzy systems have been applied to a wide variety of fields ranging from control, signal processing, communications, integrated circuit manufacturing and expert systems to business, medicine, psychology, etc. However, most significant applications have concentrated on control problems.

The type and number of inputs to the fuzzy scheduler affect the quality of the scheduling significantly. The types of input should be selected based on their ability to describe the prevailing conditions in the job shop. On the other hand, the number of inputs suitable for fuzzy scheduling determines the complexity of the inference engine.

2 Proposed Algorithm

2.1 Part Type Selection

In this study, an attempt was made to develop a fuzzy logic-based decision-making algorithm to determine priorities for part types that are to be machined, prior to scheduling [7]. The proposed system assigns priorities to part types taking the batch size, due date, total processing time and tool slots needed into account. The reasons for the selection of these input parameters are as follows. First of all, batch size and processing time of the part types are selected as inputs to the fuzzy controller as these two parameters directly affect the utilization rate of the machine tools. In order to maximize throughput (no. of part types produced/assigned to a particular machine tool) the part types utilizing less machining times should have higher priorities. Secondly, the number of tool slots that a particular part type requires on a certain machine is another important parameter as tool slot capacities of machine tools are limited.

Finally, due date is an important parameter for the generation of a proper schedule as due dates should be strictly met in order to eliminate delay penalties. These parameters were used in similar studies in literature, especially those which concentrated on the maximization of throughput and minimization of work-in-process [8, 9, 10, 11, 12, 13]. The generated output is a priority percentage, similar to a study on a real time dispatcher for flexible manufacturing systems/cells using fuzzy logic [14]. In the proposed system, the normalized contribution functions for the batch size, due date, total processing time and tool slots needed are calculated [7]. The four inputs to the system are:

1. Normalized contribution function of the batch size of part type i : defined as the ratio of the difference of the batch size of part type i and minimum batch size to the difference of maximum and minimum batch size.

$$n(b_i) = \frac{[b_i - (b_i)_{\min}]}{[(b_i)_{\max} - (b_i)_{\min}]}$$

where:

- b_i : batch size of part i
- $(b_i)_{\min}$: minimum batch size
- $(b_i)_{\max}$: maximum batch size

2. Normalized contribution function of the due date of part type i : defined as the ratio of the difference of the due date of part type i and minimum due date to the difference of maximum and minimum due date.

$$n(d_i) = \frac{[d_i - (d_i)_{\min}]}{[(d_i)_{\max} - (d_i)_{\min}]}$$

where:

- d_i : due date of part i
- $(d_i)_{\min}$: minimum due date
- $(d_i)_{\max}$: maximum due date

3. Normalized contribution function of the total processing time of part type i : defined as the ratio of the difference of the total processing time of part type i and minimum total processing time to the difference of maximum and minimum total processing time.

$$n(t_i) = \frac{[t_i - (t_i)_{\min}]}{[(t_i)_{\max} - (t_i)_{\min}]}$$

where:

- t_i : total processing time of part i
- $(t_i)_{\min}$: minimum total processing time
- $(t_i)_{\max}$: maximum total processing time

4. Normalized contribution function of the tool slots that part type i needs: defined as the ratio of the difference of the tool slots that part type i needs and minimum tool slots needed to the difference of maximum and minimum tool slots needed.

$$n(s_i) = \frac{[s_i - (s_i)_{\min}]}{[(s_i)_{\max} - (s_i)_{\min}]}$$

where:

- s_i : tool slots that part type i needs
- $(s_i)_{\min}$: minimum tool slots needed
- $(s_i)_{\max}$: maximum tool slots needed

All of the four input variables are associated with identical linguistic values: small (S), medium (M), and big (B). The generated output is a priority percentage: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%.

There are $3 \times 3 \times 3 \times 3 = 81$ rules for the system. The membership functions for the normalized batch size, normalized due date, normalized total processing time and normalized tool slots needed are similar and presented in Fig. 1. Triangular shaped membership functions were selected, as it is the most common type of membership used in similar studies [6, 14, 15, 16, 17, 18]. At the initial stages of this study different forms of membership functions were studied but triangular

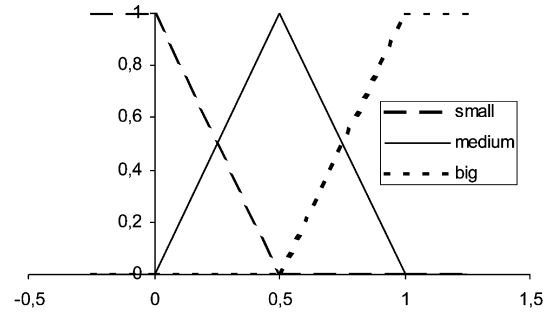


Fig. 1 Membership function for the input variables for stage 1

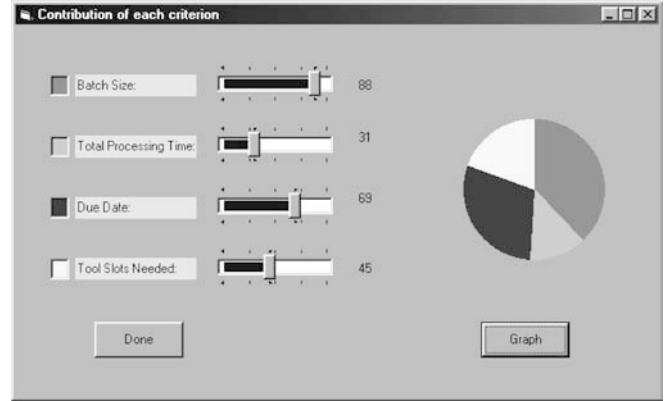


Fig. 2 Criteria contribution modifier window for stage 1

membership functions showed the best performance both in computing time and performance.

A criteria contribution modifier is used similar to the one in a study carried out by Smed et al. [1], by which the contribution of each criterion (batch size, due date, total processing time and tool slots needed) to the final solution can be modified by the user. Each criterion associated with the problem can be represented as a fuzzy set by defining a membership function, which corresponds to the intuitive rule behind the criterion. The relative importance of the values given by the user assigns weights to the fuzzy sets. These weights ensure that the more important criteria have a greater effect on the objective function than the less important ones. This feature is particularly useful because it provides the user with a way to alter the goals of the optimisation dynamically.

The criteria contribution modifier window that is used in the program is shown in Fig. 2. With the help of the sliders, the user can change the relative importance of each criterion (batch size, due date, total processing time and tool slots needed). It should be noted that 12-12-12-12, 50-50-50-50 and 76-76-76-76 for batch size, due date, total processing time and tool slots needed, respectively, give the same result, because these 4 values show the relative importance of these criteria. That is, in the case of 12-12-12-12 for batch size, due date, total processing time and tool slots needed, respectively, it means that all these four criteria have equal importance, therefore equal contribution to the objective function.

Finally, some sample rules for the fuzzy decision system are as follows:

- *IF* batch is *S AND* time is *S AND* date is *S AND* slot is *S THEN* decision is 0.9
- *IF* batch is *S AND* time is *M AND* date is *M AND* slot is *M THEN* decision is 0.6
- *IF* batch is *M AND* time is *S AND* date is *S AND* slot is *B THEN* decision is 0.5
- *IF* batch is *M AND* time is *M AND* date is *M AND* slot is *M THEN* decision is 0.5
- *IF* batch is *B AND* time is *M AND* date is *S AND* slot is *M THEN* decision is 0.4
- *IF* batch is *B AND* time is *B AND* date is *B AND* slot is *S THEN* decision is 0.2

The logic behind the rule creation process is that the more critical it is that a parameter meet the objectives, the more urgently it should be processed to avoid the bottlenecks on machine tools or delays in the makespan. For example, if due date is tight then the part type should be processed as soon as possible to eliminate/minimize the penalty cost occurring as a consequence of this delay.

2.2 Numerical example

For the part type priority phase of the algorithm the following sample runs were performed. Input data for the scheduler is given in Table 1. There are 15 part types with a total of 45 operations that are to be performed on 6 machines.

2.2.1 Sample run 1

Contribution of criteria:

- batch size: 50%
- due date: 50%
- total processing time: 50%
- tool slots needed: 50%

The above contribution values mean that all these criteria must have equal contribution to the final decision (priority).

The generated output for the ranking of the part types is presented in Table 2.

A higher priority value means that this particular part type should be processed as soon as possible; whereas, a low priority value means just the opposite. As it can be seen from Table 2, maximum priority is assigned to part type 1 and minimum to part type 5. When the input data is analysed, part type 1 should obviously have the highest priority as its batch size and unit processing time is small (it can be processed in less amount of time compared to others), its due date is tight (it should be processed as soon as possible) and its tool slot requirements is small (it occupies less tool slots in the tool magazine compared to others). On the

Table 1 Input data for priority calculation

Part no	Operation no	Batch size	Due date	Unit processing time	Tool slots needed
1	1	8	698	18	1
1	2	8	698	25	1
2	1	9	2822	24	1
2	2	9	2822	22	1
2	3	9	2822	26	2
3	1	13	2202	11	3
3	2	13	2202	14	1
3	3	13	2202	19	1
3	4	13	2202	22	2
4	1	9	2083	25	1
4	2	9	2083	16	1
4	3	9	2083	7	1
4	4	9	2083	21	1
4	5	9	2083	19	1
5	1	12	3087	13	1
5	2	12	3087	23	3
5	3	12	3087	25	1
5	4	12	3087	7	1
5	5	12	3087	24	3
6	1	9	2822	18	1
6	2	9	2822	25	1
6	3	9	2822	24	1
6	4	9	2822	22	1
7	1	13	2202	26	2
7	2	13	2202	11	3
7	3	13	2202	14	1
7	4	13	2202	19	1
7	5	13	2202	22	2
8	1	10	2156	25	1
8	2	10	2156	16	1
8	3	10	2156	7	1
9	1	7	2089	21	1
9	2	7	2089	18	2
10	1	9	2163	23	1
10	2	9	2163	17	2
11	1	12	2265	18	1
11	2	12	2265	25	2
11	3	12	2265	21	1
12	1	8	1975	23	2
12	2	8	1975	14	1
13	1	10	2015	25	1
13	2	10	2015	18	2
14	1	9	2114	28	1
15	1	14	1875	22	1
15	2	14	1875	19	1

Table 2 Priority values for sample run 1

Part type	Priority
1	0.774
2	0.453
3	0.216
4	0.400
5	0.113
6	0.367
7	0.118
8	0.541
9	0.635
10	0.574
11	0.317
12	0.625
13	0.572
14	0.633
15	0.332

contrary, part type 5 has the lowest priority as its batch size and unit processing times are above the average, its tool slot requirements are high and its due date is far away.

2.2.2 Sample run 2

The input data is the same as the above case except the contribution of criteria, which are given as follows:

- batch size: 1%
- due date: 1%
- total processing time: 100%
- tool slots needed: 1%

The above contribution values mean that total processing time is the most important parameter to be taken into account in the final decision (priority); whereas, the others have relatively equal importance. Also it should be mentioned that, with the above weights the final decision is based mainly on total processing time to the extent that the algorithm enables.

The generated output for the ranking of the part types is presented in Table 3.

As it can be seen from Table 3, the priority of part type 1 decreased when compared to sample run 1. This is because of the fact that both the batch size and the unit processing time for part type 1 are not the minimum values; therefore, when total processing time is the most important parameter in the final decision, its priority obviously decreased.

3 Scheduling

The scheduling problems stated in literature are taken into account when designing the proposed system. First of all, as stated by Pinedo [19] most theoretical research has focused on models with a single objective, but in the real world, there are usually a number of objectives. Therefore, the scheduling system proposed in the scope

Table 3 Priority values for sample run 2

Part type	Priority
1	0.762
2	0.480
3	0.230
4	0.384
5	0.108
6	0.428
7	0.125
8	0.520
9	0.613
10	0.548
11	0.336
12	0.563
13	0.532
14	0.624
15	0.339

of this study deals with several objectives, namely minimizing makespan, minimizing penalty costs, balancing the workloads on machine tools and meeting the due dates of the part types in the work order.

Secondly, mathematical models often do not take preferences into account. In a model, a job either can or cannot be processed on a given machine. But in the methodology proposed, alternative machines can be defined for a particular operation in the input file. The algorithm searches for the most suitable machine tool, for which details are given in the following paragraphs.

Generally in the mathematical models, the weights (priorities) of the jobs are assumed to be fixed, i.e. they do not change over time. Since this could not be the case in real world situations, the system calculates the priorities of the part types prior to schedule generation.

It is obvious that the best schedule generated with certain initial conditions does not work when one or more conditions in the shop floor or part type parameters change. Therefore, the system should be capable of modifying the schedule accordingly in order to meet the objectives as closely as possible. The system proposed has a very flexible re-scheduling algorithm, by which the schedule can be modified/generated with the new initial conditions. The available time of machine tools, due dates and arrival times of part types as well as batch sizes and periods for that particular machine tool can be defined by the user prior to the re-scheduling process. Also the user has the flexibility to define different processing and setup times for the operations on alternative machines defined. In this way the optimum schedule can be generated taking current conditions into account.

As last but not least, machine availabilities can be defined by the user when preparing the input file if a scheduled maintenance exists. This overcomes the disadvantage of most scheduling systems, as most assume that all the machine tools are available for the scheduling period Pinedo [19]. Also, at every intermediate stage of the schedule generation process, instantaneous machine utilization rates are calculated and operation-machine assignment decisions are made based on this data.

In this respect, in the second stage of the proposed algorithm, the operation-machine allocation decisions are made. The schedule for the system is generated for a certain time period with the help of fuzzy decision-making. The parameters that are taken into account in the decision-making process are described in the following paragraphs.

First of all, with the given batch size, arrival time, due date, unit processing time, setup time, delay cost and alternative machines, the instantaneous priorities of the machine tools are calculated. That is, the available machine tools are ranked in descending order with taking the instantaneous machine utilization and available processing time into account. Instantaneous machine utilization is defined as the total processing time for the workpieces that were assigned to that particular machine to the total time interval.

$$\text{instantaneous machine utilization} = \frac{\sum \text{processing time of operations}}{\text{total time interval}}$$

where the total processing time of operations includes the corresponding setup times for a certain operation on a certain machine, and total time interval is the arrival time of the operation that is to be assigned to a certain machine tool.

With the calculated relative ranks for machine tools that are available for that operation, the algorithm searches for the most appropriate machine tool for the operation in question and decides whether to assign the operation to that machine tool, reject it or divide the batch size. The batch size division decision is made with the help of fuzzy decision-making based on several factors. The inputs to the fuzzy system are based on the following concerns:

Is the operation in question the first operation of a certain part type and has the batch been divided before?

Meeting the due dates of the work order is one of the main objectives of our study; therefore, first operations of the part types have higher priorities in order to initiate the machining process of a certain part type as soon as possible. Similarly, if the batch size has been divided then the operation has a higher priority in order to make it available for the next operation. Values between 0 and 1 are assigned to each case as given below, so that they can be compatible with the normalized values used as other inputs to the controller.

- 1: If it is the first operation and batch has been divided before
- 0.5: If it is the first operation and batch has not been divided before
- 0.5: If it is not the first operation and batch has been divided before
- 0.25: If it is not the first operation and batch has not been divided before

Will any other operation need the same machine tool until the operation in question is completed?

The algorithm searches for the operations, which must be processed in that machine tool (i.e. there is no alternative machine for that operation), that are awaiting or that will arrive the system before the operation that is to be assigned is completed. If there is another operation that will wait as a consequence of this assignment then the entire batch of the operation is either assigned to that machine tool or the batch size is divided or the operation is rejected for the time being. The values for these are:

- 0.75: If no other operation needs that machine tool
- 0.25: If another operation needs that machine tool

Normalized delay cost

One of the points that distinguishes this study from previous ones in the literature is the introduction of

the delay cost concept during the decision-making for schedule generation. The delay cost or penalty for the part type that this operation belongs to is taken into account in order to minimize the total penalty for the delay of the completion of the operations of a certain part type. It is defined as the ratio of the difference of the delay cost of part type i and minimum delay cost to the difference of maximum and minimum delay cost.

$$n(c_i) = \frac{[c_i - (c_i)_{\min}]}{[(c_i)_{\max} - (c_i)_{\min}]}$$

where:

- c_i : delay cost of part i
- $(c_i)_{\min}$: minimum delay cost
- $(c_i)_{\max}$: maximum delay cost

Priority

The priorities of the part types that were calculated in the first stage of the scheduling system are also taken into account.

The above-mentioned inputs to the fuzzy system are associated with the linguistic values small (S), medium (M) and big (B) and are identical for four of the inputs.

The generated output is an availability percentage: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%. The assignment and batch size division decisions are made based on these availability percentages. There are $3 \times 3 \times 3 = 81$ rules for the system. The membership functions for the inputs are given in Fig. 3. It should be mentioned that the fuzzy controller is run at every intermediate stage during the schedule generation process. That is, in order to generate a schedule for 75 operations, the fuzzy controller runs at least a couple of hundred times.

A criteria contribution modifier window, shown in Fig. 4, is used similar to the one in the first stage of the scheduling system, by which the contribution of each criterion to the final solution can be modified by the user. The relative importance of the values given by the user assigns weights to the fuzzy sets. It is here worth mentioning that the contribution modifier does not cause major changes in the generated schedule; instead, it helps the fuzzy controller to generate outputs taking into account the user preferences about the effect of inputs.

Therefore, some sample rules for the fuzzy decision system are:

- IF inp1 is S AND inp2 is S AND inp3 is S AND inp4 is S THEN decision is 0.1
- IF inp1 is S AND inp2 is M AND inp3 is M AND inp4 is M THEN decision is 0.4
- IF inp1 is M AND inp2 is S AND inp3 is S AND inp4 is B THEN decision is 0.5
- IF inp1 is M AND inp2 is M AND inp3 is M AND inp4 is M THEN decision is 0.5

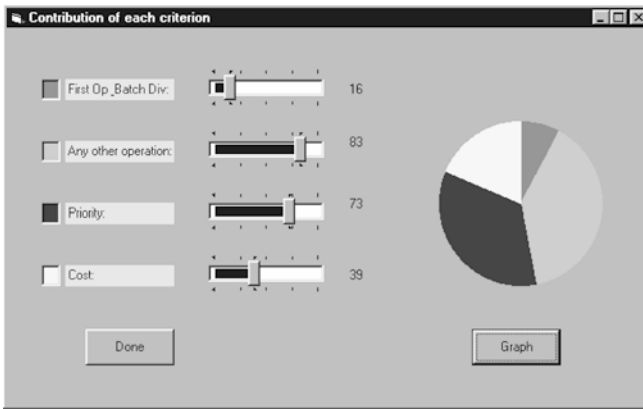


Fig. 3 Membership function for all input variables for stage 2

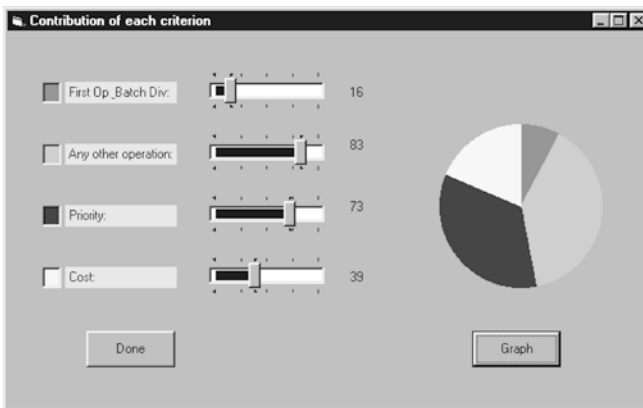


Fig. 4 Criteria contribution modifier window for stage 2

- IF inp1 is B AND inp2 is M AND inp3 is S AND inp4 is M THEN decision is 0.6
- IF inp1 is B AND inp2 is B AND inp3 is B AND inp4 is S THEN decision is 0.8

Finally, the re-scheduling algorithm for machine failures should be mentioned. Machine failure is one of the greatest problems for the efficient operation of a pre-determined schedule. In the case of a machine failure, the whole schedule shifts if proper measures are not taken as soon as possible. The operations on the failed machine have to wait; therefore, it affects the completion times of other operations, as some of them are the prerequisites to others. In the proposed algorithm, in the case of a machine failure, the algorithm modifies the schedule accordingly. That is, the operations assigned to that particular machine tool are dispatched to other alternative machines, if possible. Therefore, the algorithm can be used as an on-line controller in the system to modify the schedule in case of a failure, and as a result the most appropriate schedules with the recent initial conditions are generated in a couple of seconds. An example of such a failure is given below.

3.1 Numerical Example

For the scheduling phase of the algorithm the following sample run was performed. Input data for the scheduler is given in Tables 4 and 5. There are 30 part types with a total of 75 operations that are to be performed on 10 machines (Machine A to Machine K). The “de”, for example, in the ‘alternative machines’ column for the 5th operation means that this operation can be performed both on Machine D and Machine E.

Table 4 Input data for schedule generation example

Operation id	Part no	Operation no	Batch size	Arrival time	Due date	Alternative machines	Tool slot requirements	Priority	Delay cost
1	1	1	8	122	698	h	1	0.8424	20
2	1	2	8	9999	698	af	1	0.8424	20
3	2	1	9	258	2822	j	1	0.3017	18
4	2	2	9	9999	2822	bg	1	0.3017	18
5	2	3	9	9999	2822	de	2	0.3017	18
6	3	1	13	361	2202	kg	3	0.1684	45
7	3	2	13	9999	2202	ah	1	0.1684	45
8	3	3	13	9999	2202	gk	1	0.1684	45
9	3	4	13	9999	2202	bg	2	0.1684	45
10	4	1	9	385	2083	gj	1	0.5105	33
11	4	2	9	9999	2083	g	1	0.5105	33
12	4	3	9	9999	2083	efg	1	0.5105	33
13	4	4	9	9999	2083	bak	1	0.5105	33
14	4	5	9	9999	2083	beg	1	0.5105	33
15	5	1	12	396	3087	bha	1	0.1572	62
16	5	2	12	9999	3087	dj	3	0.1572	62
17	5	3	12	9999	3087	aef	1	0.1572	62
18	5	4	12	9999	3087	ag	1	0.1572	62
19	5	5	12	9999	3087	j	3	0.1572	62

Table 4 (Contd.)

Operation id	Part no	Operation no	Batch size	Arrival time	Due date	Alternative machines	Tool slot requirements	Priority	Delay cost
20	6	1	9	158	2822	eh	1	0.8424	18
21	6	2	9	9999	2822	ajk	1	0.8424	18
22	6	3	9	9999	2822	dg	1	0.8424	18
23	6	4	9	9999	2822	bhj	1	0.8424	18
24	7	1	13	278	2202	df	2	0.1684	45
25	7	2	13	9999	2202	eg	3	0.1684	45
26	7	3	13	9999	2202	ce	1	0.1684	45
27	7	4	13	9999	2202	dfh	1	0.1684	45
28	7	5	13	9999	2202	bh	2	0.1684	45
29	8	1	10	642	2156	fk	1	0.4277	7
30	8	2	10	9999	2156	gh	1	0.4277	7
31	8	3	10	9999	2156	dbh	1	0.4277	7
32	9	1	7	586	2089	bf	1	0.3254	16
33	9	2	7	9999	2089	cdj	2	0.3254	16
34	10	1	9	603	2163	j	1	0.6547	28
35	10	2	9	9999	2163	cg	2	0.6547	28
36	11	1	12	652	2265	fag	1	0.2458	39
37	11	2	12	9999	2265	cb	2	0.2458	39
38	11	3	12	9999	2265	aek	1	0.2458	39
39	12	1	8	486	1975	efh	2	0.5463	42
40	12	2	8	9999	1975	aj	1	0.5463	42
41	13	1	10	503	2015	efg	1	0.4269	35
42	13	2	10	9999	2015	acj	2	0.4269	35
43	14	1	9	554	2114	de	1	0.2578	42
44	15	1	14	406	1875	ce	1	0.6874	54
45	15	2	14	9999	1875	ek	1	0.6874	54
46	16	1	9	523	2054	hg	2	0.5271	24
47	16	2	9	9999	2054	ah	3	0.5271	24
48	17	1	15	429	1598	bch	2	0.4234	64
49	18	1	8	875	1986	chk	2	0.2158	25
50	18	2	8	9999	1986	df	1	0.2158	25
51	19	1	12	657	1874	fgj	2	0.5236	45
52	19	2	12	9999	1874	abh	1	0.5236	45
53	19	3	12	9999	1874	dfh	2	0.5236	45
54	20	1	9	576	2351	aef	2	0.4231	14
55	20	2	9	9999	2351	gj	1	0.4231	14
56	20	3	9	9999	2351	hj	2	0.4231	14
57	21	1	13	957	2044	aej	1	0.5148	59
58	21	2	13	9999	2044	gh	2	0.5148	59
59	22	1	7	1125	2158	aej	1	0.2687	61
60	22	2	7	9999	2158	ek	2	0.2687	61
61	23	1	10	952	2367	eja	2	0.5127	34
62	23	2	10	9999	2367	ahb	3	0.5127	34
63	24	1	9	853	2048	ehc	1	0.4231	59
64	25	1	12	743	1986	hd	2	0.7536	12
65	25	2	12	9999	1986	j	1	0.7536	12
66	26	1	7	862	2157	fj	2	0.3452	48
67	26	2	7	9999	2157	ga	1	0.3452	48
68	26	3	7	9999	2157	f	2	0.3452	48
69	27	1	13	1135	2563	hc	1	0.5469	67
70	27	2	13	9999	2563	jd	1	0.5469	67
71	28	1	6	994	1198	eh	2	0.4321	29
72	29	1	8	1325	1896	dg	1	0.6324	56
73	29	2	8	9999	1896	cf	2	0.6324	56
74	30	1	14	983	1947	ej	2	0.4258	69
75	30	2	14	9999	1947	fh	1	0.4258	69

The output generated by the proposed algorithm using the input data given in Tables 4 and 5 is presented in Fig. 5. The percentages on the right side represent the corresponding machine utilization rates.

A second run was also performed in which, Machine C failed at time $t = 750$ and was unavailable until $t = 1,400$ (Fig. 6). In such a case, the algorithm modified the sche-

dule and dispatched some operations to other available machines in order to minimize the makespan. The makespan increased from 2,214 to only 2,288 (~3% increase) as a result of this re-scheduling process. Otherwise operation 63 would wait until the machine is in operation again (i.e. $t = 1,400$) and some other operations would be late as a consequence of this failure, which means that the

Table 5 Machining times for schedule generation example

Part no	Operation no	Machine	Unit processing time	Setup time
1	1	h	18	3
1	2	a	25	4
1	2	f	22	3
2	1	j	24	2
2	2	b	22	5
2	2	g	24	3
2	3	d	26	4
2	3	e	24	3
3	1	k	11	3
3	1	g	9	3
3	2	a	14	4
3	2	h	12	2
3	3	g	19	5
3	3	k	17	3
3	4	b	22	4
3	4	g	24	3
4	1	g	25	3
4	1	j	22	3
4	2	g	16	4
4	3	e	7	3
4	3	f	6	2
4	3	g	7	4
4	4	b	21	4
4	4	a	18	3
4	4	k	23	4
4	5	b	19	4
4	5	e	17	5
4	5	g	15	3
5	1	b	13	3
5	1	h	12	3
5	1	a	16	3
5	2	d	23	5
5	2	j	24	3
5	3	a	25	5
5	3	e	23	4
5	3	f	21	3
5	4	a	7	3
5	4	g	6	4
5	5	j	24	3
6	1	e	18	3
6	1	h	16	4
6	2	a	25	4
6	2	j	24	3
6	2	k	22	3
6	3	d	24	2
6	3	g	22	3
6	4	b	22	5
6	4	h	22	3
6	4	j	21	3
7	1	d	26	4
7	1	f	24	3
7	2	e	11	3
7	2	g	9	2
7	3	c	14	4
7	3	e	12	3
7	4	d	19	5
7	4	f	16	4
7	4	h	17	2
7	5	b	22	4
7	5	h	21	3
8	1	f	25	3
8	1	k	22	3
8	2	g	16	4
8	2	h	14	3
8	3	d	7	3
8	3	b	6	3

Table 5 (Contd.)

Part no	Operation no	Machine	Unit processing time	Setup time
8	3	h	8	2
9	1	b	21	4
9	1	f	18	3
9	2	c	18	3
9	2	d	15	3
9	2	j	13	3
10	1	j	23	2
10	2	c	17	5
10	2	g	22	3
11	1	f	18	4
11	1	a	13	2
11	1	g	16	3
11	2	c	25	3
11	2	b	23	2
11	3	a	21	5
11	3	e	17	4
11	3	k	22	3
12	1	e	23	4
12	1	f	24	3
12	1	h	21	3
12	2	a	14	5
12	2	j	12	3
13	1	e	25	4
13	1	f	22	3
13	1	g	21	3
13	2	a	18	5
13	2	c	16	3
13	2	j	15	2
14	1	d	28	4
14	1	e	25	3
15	1	c	22	5
15	1	e	23	2
15	2	e	19	4
15	2	k	17	5
16	1	h	23	5
16	1	g	22	2
16	2	a	27	4
16	2	h	24	3
17	1	b	24	5
17	1	c	22	3
17	1	h	21	3
18	1	c	21	4
18	1	h	18	3
18	1	k	17	4
18	2	d	19	5
18	2	f	17	4
19	1	f	23	4
19	1	g	21	3
19	1	j	19	4
19	2	a	18	5
19	2	b	16	4
19	2	h	18	3
19	3	d	21	3
19	3	f	19	2
19	3	h	17	3
20	1	a	24	6
20	1	e	22	3
20	1	f	24	3
20	2	g	18	4
20	2	j	16	3
20	3	h	21	5
20	3	j	18	3
21	1	a	26	4
21	1	e	24	3
21	1	j	22	3
21	2	g	21	3

Table 5 (Contd.)

Part no	Operation no	Machine	Unit processing time	Setup time
21	2	h	18	3
22	1	a	18	2
22	1	e	16	3
22	1	j	17	3
22	2	e	21	3
22	2	k	19	3
23	1	e	23	4
23	1	j	21	2
23	1	a	19	2
23	2	a	18	5
23	2	h	18	2
23	2	b	16	3
24	1	e	26	4
24	1	h	24	3
24	1	c	21	3
25	1	h	18	5
25	1	d	16	3
25	2	j	23	4
26	1	f	18	3
26	1	j	15	3
26	2	g	23	5
26	2	a	21	3
26	3	f	17	4
27	1	h	23	5
27	1	c	21	3
27	2	j	18	3
27	2	d	16	3
28	1	e	26	4
28	1	h	24	3
29	1	d	17	5
29	1	g	16	3
29	2	c	24	3
29	2	f	21	3
30	1	e	18	4
30	1	j	16	3
30	2	f	23	5
30	2	h	21	3

makespan would be 2,799, which is much higher than 2,288. This case is illustrated in Fig. 7.

In order to illustrate the effect of criteria contribution, a run was performed with same input data, except the contribution of inputs are as follows:

First operation and batch 1%
division:
Any other operation needs 1%
the same machine tool:
Delay cost: 100%
Priority: 1%

which means that the controller will take only the delay cost as the major objective to the extent that the algorithm enables. The Gantt chart of this run is given in Fig. 8. As it can be seen from this chart, makespan increased slightly as the only objective was to minimize delay cost, but in that case the delay penalties to be paid are at the minimum level.

Another run set was performed with different input data and the cases with no machine tool failure, failure without re-scheduling and failure with re-scheduling are presented in Figs. 9, 10 and 11, for which machine E was out of order from $t = 750$ to $t = 1,400$. As it can be seen from these charts makespan increased from 2,259 to 2,422, instead of 2,927 with the introduction of the re-scheduling algorithm. Therefore, the proposed re-scheduling algorithm compensated for a 650 min. machine breakdown with an increase of only 163 min.

4 Discussion

Scheduling is one of the most important issues in the planning and operation of manufacturing systems, but the generation of consistently good schedules has proven to be extremely difficult. The problem is that optimal scheduling solutions involve costly and impractical

Fig. 5 Gantt chart for run 1 when there is no machine failure

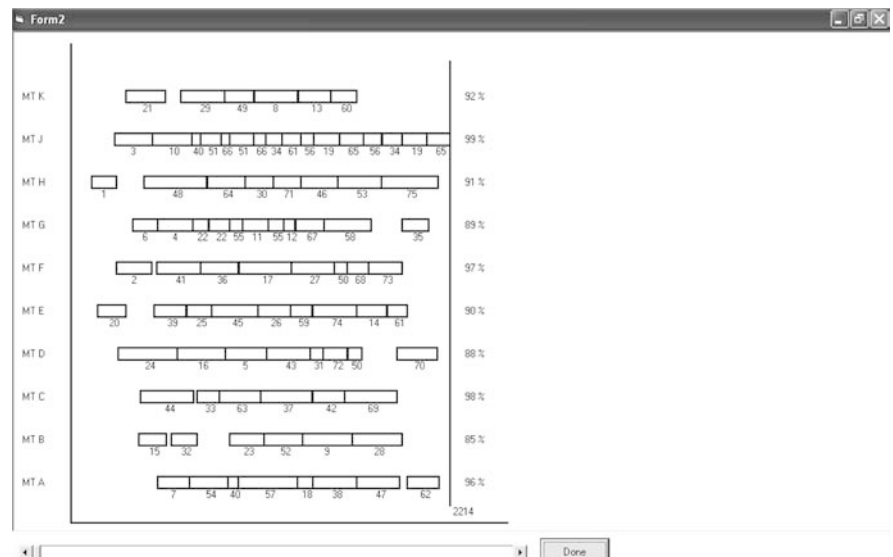


Fig. 9 Gantt chart for run 2 when there is no machine failure

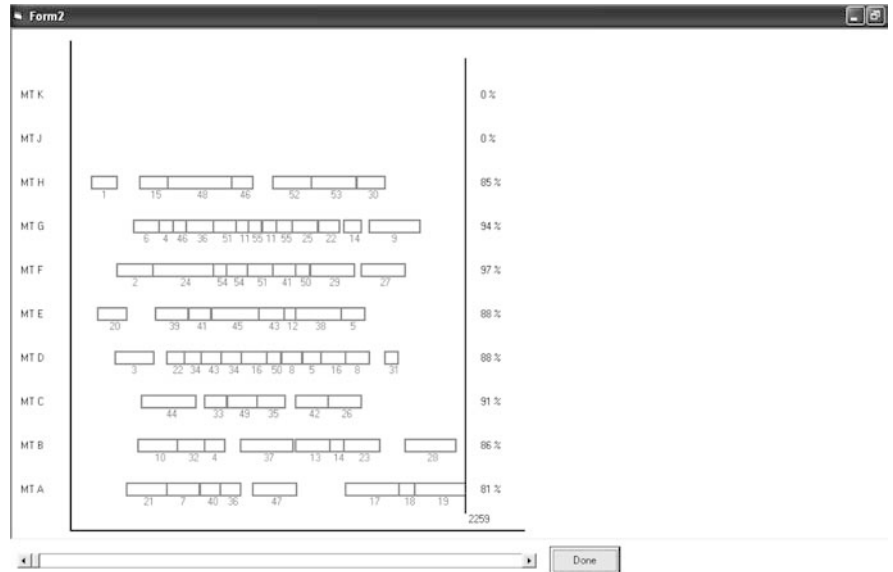


Fig. 10 Gantt chart for run 2 when there is machine failure (with re-scheduling)

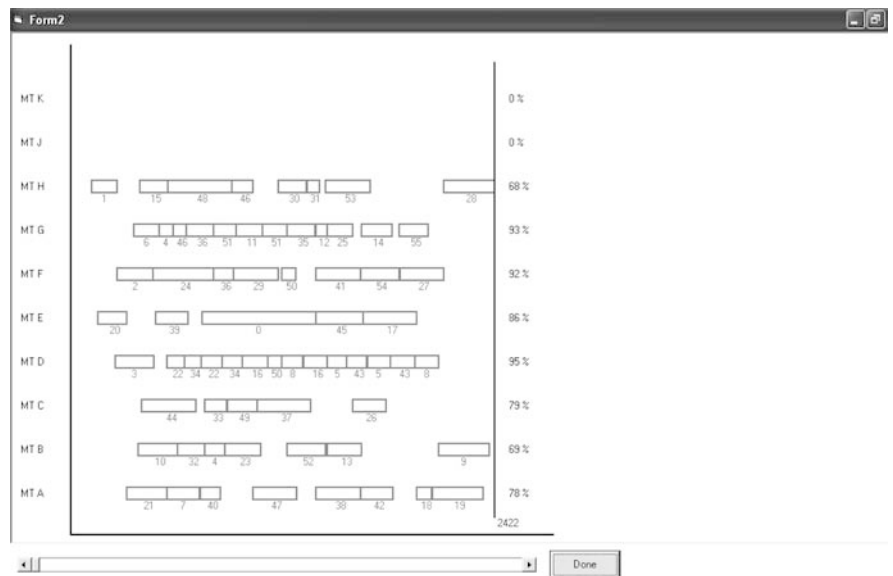
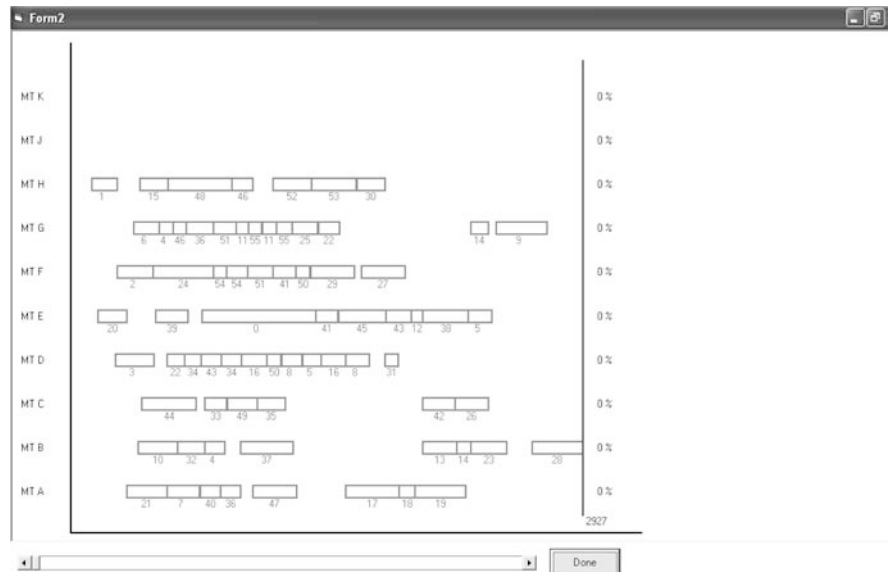


Fig. 11 Gantt chart for run 2 when there is machine failure (without re-scheduling)



enumeration procedures, while the performance of most heuristic techniques is difficult to estimate and varies considerably from one problem to the next.

On the other hand, fuzzy control has proven to be successful in problems for which exact mathematical modelling is hard or impossible, but an experienced human operator can control the process. In particular, fuzzy logic decision-making is suited for problems which have many and often conflicting objectives with information given in linguistic terms [15].

In the first part of this study, using fuzzy arithmetic, all the necessary calculations were performed relating to the batch size, total processing time, tool slots needed and due date similar to a previously proposed methodology for a dynamic Dial-A-Ride problem [18]. The values calculated by this approach represent the input data for developed approximate reasoning algorithms.

The developed algorithm is used prior to the scheduling algorithm for a job shop. The scheduling algorithm is also based on fuzzy logic decision-making for assigning the operations of the part types to suitable machines, with objectives being the makespan, relative machine loads and cost. In a previous study, solution for the complex problem of scheduling under uncertainty using a fuzzy expert system approach with a fuzzy rule base and fuzzy inferencing was proposed [20].

Similar to our study, Chan and Chan [11] presented a simulation model of a flexible manufacturing system, which minimizes three performance criteria simultaneously, i.e. mean-flow time, mean tardiness and mean earliness. As a result, they concluded that the proposed algorithm is found to be useful even when the performance criteria used are biased on purpose. This indicates that the decision-maker can vary the weighing factor for each performance criterion according to their preference.

For job shop scheduling, using conventional heuristic algorithms (priority rules) only, it is almost impossible to achieve an optimal solution. Research has been carried out to improve the heuristic algorithms to give a near-optimal solution by Roy and Zhang [21]. They proposed a fuzzy logic-based, dynamic scheduling algorithm aimed at achieving this goal. The concept of new membership functions is discussed in the algorithm as a link to connect several priority rules. The constraints to determine the membership function of jobs for a particular priority rule are established, and three membership functions are developed. In order to decide the weight vector of priority rules, an aggregate performance measure is suggested.

A study which Smed et al. [1] proposed includes a visual presentation of the overall state of the production, a possibility to edit the schedule manually (the system checks the capacity constraints automatically) or improve it algorithmically, information about products, jobs, components, simulated times, etc. and a possibility to employ feeder and printing order optimisers selectively.

A criteria equalizer option, similar to the one used in our study, is present in the program, by which the

contribution of each criterion (widths, orphans, setup, urgencies, oven, groups and total setup) to the final solution can be modified by the user. Each criterion associated with the problem can be represented as a fuzzy set by defining a membership function, which corresponds to the intuitive rule behind the criterion. The relative importance of the values given by the user form a prioritisation of the criteria and assigns weights to the fuzzy sets. These weights ensure that the more important criteria have a greater effect on the objective function than the less important ones.

5 Conclusion

In the first stage of this study, a fuzzy logic-based algorithm for assigning priorities to part types that are to be machined is proposed. The decision is made taking the batch size, due date, total processing time and tool slots needed into account. A criteria contribution equalizer is used in decision-making. The generated output is a priority percentage. The proposed algorithm is used prior to operation machine allocation and scheduling process to assign machining priorities for part types.

In the second stage, an operation-machine allocation and scheduling algorithm is presented. A criteria contribution equalizer is used in decision-making. Our results revealed that, when the user modifies the weighing factors of each criterion, the results change by increasing the overall efficiency of the fuzzy scheduler for the desired performance criterion. The proposed algorithm regenerates the schedule in the case of a machine breakdown, without increasing the makespan by much. The system architecture and linguistic variables are presented and results showed that the proposed algorithm improves the system efficiency.

As a result it can be concluded that, the proposed system combines the part type priority, batch size division, instantaneous machine utilization rate, compensation of machine breakdowns or other changes in the shop floor conditions and the delay cost concepts successfully and can be used both off-line and on-line where necessary in small to medium size job shops.

References

1. Smed J, Johtela T, Johnsson M, Puranen M, Nevalainen O (2000) An interactive system for scheduling jobs in electronic assembly. *Int J Adv Manuf Technol* 16:450–459
2. Saygin C, Kilic SE (1999) Integrating flexible process plans with scheduling in flexible manufacturing systems. *Int J Adv Manuf Technol* 15:268–280
3. Dunkler O, Mitchell CM, Govindaraj T, Ammons JC (1988) The effectiveness of supervisory control strategies in scheduling flexible manufacturing systems. *IEEE Trans Syst Man Cybern*, March–April 18:223–237
4. Dessouky Y, Beaumariage T, Roberts C, Ogle M (1993) An intelligent system for batch scheduling for the process industry. *Comput Ind* 22:233–247

5. Chrystolouris G (1992) *Manufacturing systems: theory and practice*. Springer, Berlin Heidelberg New York
6. Babic O, Krstic T (2000) Airspace daily operational sectorization by fuzzy logic. *Fuzzy Sets Syst* 116:49–64
7. Bilkay O, Anlagan O, Kilic SE (2002) Part type selection using fuzzy logic. In: Tenth Int Conf Mach Design Prod, Cappadocia, Turkey, September
8. Tetzlaff UAW, Pesch E (1999) Optimal workload allocation between a job shop and an FMS. *IEEE Trans Robot Autom* 15:20–32
9. Kannan VR, Ghosh S (1993) An evaluation of the interaction between dispatching rules and truncation procedures in job-shop scheduling. *Int J Prod Res* 31:1637–1654
10. Choi J, Hitomi K, Econ P (1994) A method of flexible scheduling for flexible manufacturing systems. *Int J Prod Econ* 33:247–255
11. Chan FTS, Chan HK (2001) Dynamic scheduling for a flexible manufacturing system—the pre-emptive approach. *Int J Adv Manuf Technol* 17:760–768
12. Barman S (1997) Simple priority rule combinations: an approach to improve both flow time and tardiness. *Int J Prod Res* 35:2857–2870
13. Vidyarthi NK, Tiwari MK (2001) Machine loading problem of FMS: a fuzzy-based heuristic approach. *Int J Prod Res* 39:953–979
14. Unver O, Anlagan O, Leblebicioglu K (1998) A real-time dispatcher for flexible manufacturing systems/cells using fuzzy logic. In: Proc 2nd Int Conf Appl Fuzzy Syst Soft Comput ICAFS'96, Siegen, Germany
15. Niittymaki J, Pursula M (2000) Signal control using fuzzy logic. *Fuzzy Sets Syst* 116:11–22
16. Raber R (1994) Fuzzy in control. *Sensor Review* 14:26–28
17. Arieh DB, Lee SE (1990) Fuzzy logic controller for part routing. Dept Ind Eng, Manhattan, Kansas
18. Teodorovic D, Radivojevic G (2000) A fuzzy logic approach to dynamic Dial-A-Ride problem. *Fuzzy Sets Syst* 116:23–33
19. Pinedo M (1995) *Scheduling: theory, algorithms and systems*. Prentice Hall, Englewood Cliffs
20. Erkmen AM, Erbudak M, Anlagan O, Unver O (1997) Genetically-tuned fuzzy scheduling for flexible manufacturing systems. *IEEE Int Conf Robot Autom* 2:951–956
21. Roy U, Zhang X (1996) Heuristic approach to n/m job shop scheduling: fuzzy dynamic scheduling algorithms. *Prod Plan Control*, May–June, 7:299–311