A MODIFIED HEURISTIC JOB SHOP SCHEDULING ALGORITHM IN AUTOMATED MANUFACTURING SYSTEMS

Prof. Dr. Güneş GENÇYILMAZ
Arş.Grv. Şakir ESNAF
Üretim Anabilim Dalı

1. INTRODUCTION

The scheduling problem has been a challenge to researchers and manufacturers for several decades. With advances in technology, the associated difficulties tend to be more sophisticated. This creates an increasing need for improved usage of the costly machinery. Hence the methodologies for modelling a scheduling problem, i.e., representation and manipulation of scheduling information, gain a new importance (Turksen et al. 1992). The scheduling function plays an important role in automated manufacturing systems (AMS) and especially in Flexible Manufacturing Systems (FMSs). However, AMS scheduling is tremendously complex due to combinatorial explosion, technological constraints and goals to be achieved (Alptekin and Rabelo 1992). There have been numerous studies by operation research and artificial intelligence researchers on the scheduling problem in manufacturing systems over ten years. Research in scheduling has focused on understanding the variety of scheduling environments that exist, and constructing scheduling models specific to these particular cases. Four types of scheduling problem are distinguished in the literature:single machine-single operation, parallel machines-single operation, flowshop series of machines-multiple operations, job shop network of machines-multiple operations (Turksen et al. 1992). Job shop scheduling problems in AMS can be solved by following three types of methods: a) exhaustive methods (0-1 algoritlım etc.), b) heuristic criteria (list scheduling, shifting bottleneck), c) natural algorithms (i.e simulated annealing, genetic algorithms) (Alfano ct al. 1994). To date numerous papers have been pubiished on the job shop scheduling in AMS using heuristic knowledge-based systems. Erschler and Esquirol (1986) presented a job-shop scheduling system, MASCOT, which uses a constraint-based analysis. An expert scheduling system to the preecding was presented in Bensana et al. (1986). The job shop scheduling system, OPAL, integrates the constraint-based analysis module with the rule-based decision support module. The control strategy of the decision-support module is based on the fuzzy set methodology. Subramanyam and Askin (1986) discussed an approach for scheduling an FMS on a daily basis for two shifts to meel the weckly produetion requircmcnt Shaw and Winston (1985) studied the planning and control problem in a cellular flexible manufacturing system as a general job shop scheduling. ISIS, deveioped by Fox (1983) at thc Carnegie Mellon University, is a well-known expert scheduling system for large-scale job shops. KBSS is deveioped and presented by Kusiak (1990) for job shop scheduling in AMS environment (Kusiak 1990). Modified algoritlım in this paper is taken KBSS as a 'skeleton' but instead of using inference engine and individual rules LRA rule and new rule combinations are used.

2. SCHEDULING RULES

Two different types of scheduling rules are used in modified algorithm. One of them is LRA rule and the others are based on combinating of two scheduling rules. First combination is called as LSO-SAO, second is LUA-S(LDR), third is SPT-BP, fourth and last one is RAN. These rules are fired in the sequence of LRA, LSO-SAO, LUA-S(LDR), SPT-BP and RAN . If more than one operation is sclected by a rule, thc next rules perform thc furthcr selection. Details of the rules are given below.

2.1.LRA (Largest Relative Advantage) rule.

This rule was suggestcd by Roll el al. for using in automated manufacturing systems, especially in FMS . The rule proposes which operation should be pcrformcd by a machine, each time it becomes available. Whcn a machine completes an operation, the rule searehers for an available operation on item which (i) can be processed on the machine, and (ii) can be processed cfficiently on that machine with the entirc system in mind.

The LRA rule is best explained by a short example. Table 1 displays machines, operations, and the respeetive processing times (bold faced). For example, the processing time of operation '03' on machine 'M2' is 46 time units. An emply celi indicates that an operation cannot be performed on that machine. LRA scheduling rule is as follows:

(1) Down every column (operation), selcet each non-empty celi, in turn. Subtract its processing time from the minimal aiternative processing time in that column. For example, thc best alternative to processing operation '04' on machine 'M3' is to proccss it on machine 'M1'; with a time difference of 3 units (regular print). A positive difference means that no better aiternative exists, while negative difference means that selceling an alternative machine would entail an inerease in processing time.

Table 1. LRA rule, example matrix.

Mach.\ Op.	01	02	03	04	05
M1	70 -45 IV	79 ∞ I		31 -3 III	39 6 II
M2	25 45 I		46 28 II	35 -7 IV	45 -6 III
M3	75 -50 III		74 -28 II	28 3 I	

(2) Along each row (machine) set a priority ranking $(U_{iq})^l$ (Roman numerals in Table 1) in descending order. For example, the operations that can be performed by machine 'M2' are ranked in the order: '01' (45), '03' (28), '05' (-6), '04' (-7) (Arzi and Roil 1993).

In this paper LRA rule matrix is used for each resource to select the operation which has the best priority ranking.

2.2. LSO-SAO rule

This rule is the combination of two rules. LSO rule selects an operation with the largest number of succesive operations. SAO selects an operation belonging to a part with the minimum number of schedulable operations (according to updated S_1). If an operation succeeds LSO and SAO rules at the same time is selected.

2.3. LUO-S(LDR) rule

This rule is the combination of two rules. LUO rule selects an operation belonging to a part wilh the largest number of unprocessed operations (Kusiak 1990). S(LDR), proposed as a new rule in this paper, selects the smallest value of largest differences between basic processing time and each of the alternative processing time of operation i which is the member of updated S_1. If an operation succeeds LUO and S(LDR) rules at the same time is selected.

2.4. SPT-BP rule

This rule combination has a little difference. SPT rule is the well-known scheduling rule that selects an operation with the shortest processing time. Instead of selecing minimum or maximum of values of operations, BP rule selects the operation which is using basic process plan. Alike above combinations, operation which succeeds two rules at the same time is selected.

2.5 RAN rule

Algorithm needs this rule to avoid of unexpecled bottlenecks. If more than one operation passes above ali, RAN rule randomly selects an operation.

3. MODIFIED HEURISTIC RULE-BASED ALGORITHM

A process plan specifies the operations belonging to the part, processing times of all operations, and the resources required such as machines, lools, or pallets / fixtures.

In many manufacturing systems, one associates with each part a basic process plan and or more alternative basic plans.

3.1 Notation

Before the modified heuristic algorithm is presented, notation and definitions are given below:

I = set of ali operations

K = set of ali parts

IP_k = set of operations belonging to part $P_k, k \in K$

L = set of all resource types

Q_l = set of resources of type $l, l \in L$

d_k = due date of part $P_k, k \in K$

f_i = completion time of operation $i, i \in I$

rt_i = remaining processing time of operation $i, i \in I$.

$(U_{iq})^l$ = priority ranking of operation i which is processed using resource of type l, $i \in I, q \in Q_l, l \in L$

ns_{ik} = number of successive operations of operation i in part $P_k, i \in I, k \in K$

np_{ik} = number of unprocessed operations in part P_k corresponding to operation

$i, i \in I, k \in K$

t = current scheduling time

r_{lq} = resource q of type l, $q \in Q_l, l \in L$

In particular, the following for type of resources are used:

- Machine ($l = 1$)
- Tool ($l = 2$)
- Pallet / fixture ($l = 3$)
- Material handling carrier ($l = 4$)

Resource r_{lq} is available if it can be used without any delay, $q \in Q_l, l \in L$. The status sr_{lq} of such resource equals 1; otherwise $sr_{lq} = 0$.

A process plan $PP_k^{(v)}$ of a part P_k is a vector of triplets, each containing operation number, processing time, and set of resources to process the operation. It is denoted as follows:

$$PP_k^{(v)} = \left[(a, t_a^{(v)}, R_a^{(v)}), \ldots, (i, t_i^{(v)}, R_i^{(v)}), \ldots, (b, t_b^{(v)}, R_b^{(v)})\right],$$

where $R_i^{(v)} = (r_{lq}^{(v)}, r_{lq}^{(v)}, \ldots, r_{lq}^{(v)}), q \in Q_l, l \in L$

a,.....,i,.....,b denotes operation numbers,

v = 0 denotes the basic process plan,

v = 1,2,....., denotes an alternative process plan, and

$t_i^{(v)}$ = processing time of operation i using process plan v.

Without loss of generality, it is assumed that $t_i^{(0)} \leq t_i^{(v)}, i \in I$ which holds in practice.

A process plan $PP_k^{(v)}$ for part P_k and the corresponding operations is available, If each element in $PP_k^{(v)}$ has been specified.

Operation i is schedulable at time t, if

1. No other operation that belongs to the same part is being processed at time t,

2. All operations preceeding operation i have been completed before time t, and

3. All resources required by the basic process plan to process operation i are available at time t.

Based on above definitions, further notation is introduced:

Operation status $s_i = \begin{cases} 0 & \text{operation i is not schedulable} \\ 1 & \text{operation i is schedulable} \\ 2 & \text{operation i is being processed (scheduled)} \\ 3 & \text{operation i has been completed} \end{cases}$

Resource status $sr_{lq} = \begin{cases} 1 & \text{resource } r_{lq} \text{ is available} \\ 0 & \text{otherwise} \end{cases}$

S_j = set of operations with $s_i = j, j = 0,1,2,3,4, i \in I$

st_k = slack time of part $P_k, st_k = (d_k - t - \sum t_i^{(0)})$ for $k \in K, i \in S_0 \cup S_1$

no_{ik} = number of schedulable operations in $S_1 \cap IP_k, i \in I, k \in K$

(Kusiak 1990).

Modified algorithm uses Kusiak's algorithm structure but it has two significant differences. LRA table which is determined for each resource is substituted for the inference engine. The second difference, instead of using Kusiak's individual rules, is combinated scheduling rules which are given below.

Rule 1: LRA rule

Rule 2: LSO-SAO rule

Rule 3: LUO-S(LDR) rule

Rule 4: SPT-BP rule

Rule 5: RAN rule

3.2 Algorithm

Step 0. Set current time t = 0 and resource status $sr_{lq} = 1$, $q \in Q_l, l \in L$. and construct LRA table like for each resource.

Step 1. Construct the following two sets:

- Set S_0 of nonschedulable operations (s_i=0)
- Set S_1 of schedulable operations (s_i=1)

Step 2. In the set S_1, select an operation i* based on the following scheduling rules:

LRA P1: $i^* = \{ i \mid min\, \{(U_{iq})^{l=1}\} \wedge min\, \{(U_{iq})^{l=2}\} \wedge, ...,$
$min\, \{(U_{iq})^{l=n}\}\}, \quad i \in S_1, q \in Q_l, l \in L.$

LSO - SAO P2: $i^* = \{ i \mid max\, \{ns_{ik}\} \wedge min\, \{no_{ik}\}\}, k \in K, i \in S_1$

LUO-S(LDR) P3: $i^* = \{ i \mid max\, \{np_{ik}\} \wedge min$
$\{\max_i\{\left|(t_i^{(0)} - t_i^{(v)})\right|, v = 1, ..., V\}\}\}, k \in K, i \in S_1$

SPT-BP P4: $i^* = \{ i \mid min\, \{t_i^{(0)}\}, i \in S$

RAN P5: break a tie randomly.

Step 3. Set

- Operation $s_{i^*} = 2$ for operation i^* selected in step 2.
- Operation status $s_i = 0$ for all unprocessed operations of the part corresponding to operation i^*

Delete operation i^* from S_1. If $S_1 \cup S_0 = \varnothing$, stop; otherwise, set

- Remaining processing time $ri_{i^*} = t_i^{(v)}$
- Resource status $sr_{lq} = 0$, for

$r_{lq} \in R_i^{(v)}, q \in Q_l, l \in L.$

Update S_1 and S_0. If $S_1 \neq \varnothing$, goto step 2. If $S_1 = \varnothing$, go to step 4.

Step 4. Construct set S_2 , and

- Calculate completion time $f_i = rt_i + t, i \in S_2$.
- Set current time $t = f_{i^\circ} = \min\{f_i\}, i \in S_2$.
- Sel operation status $s_{i^\circ} = 3$.
- Delete operation i° from S_2.
- Set resource status

$sr_{lq} = 1, r_{lq} \in R_i^{(v)}, q \in Q_l, l \in L$

- Set remaining time $rt_i = f_i - t, i \in S_2$.

Update S_1 and S_2.

Step 5. If $S_1 \cup S_0 = \varnothing$, stop otherwise go to step 6.

Step 6. If $S_1 \neq \varnothing$, go to step 2. If $S_1 = \varnothing$ go to step 4.

Flowchart of the above given algorithm is illustrated in figure 1. During scheduling procedure, due dates of each part aren't imposed as Kusiak's knowledge base algorithm because of comparing two aigorithms objectively. Using predelermined due dates two algorithms are compared in the last section.

In the next section, modified heuristic algorithm is illustrated with Kusiak's numerical example.

4. SOLVING NUMERICAL EXAMPLE.

Example

Schedule 12 operations of three parts shown in figure 1 on three machines. It is assumed that

- Three different tools are available to process the operation

(This feature shows that production system is nol FMS).

- Ali other resources are unlimited, and

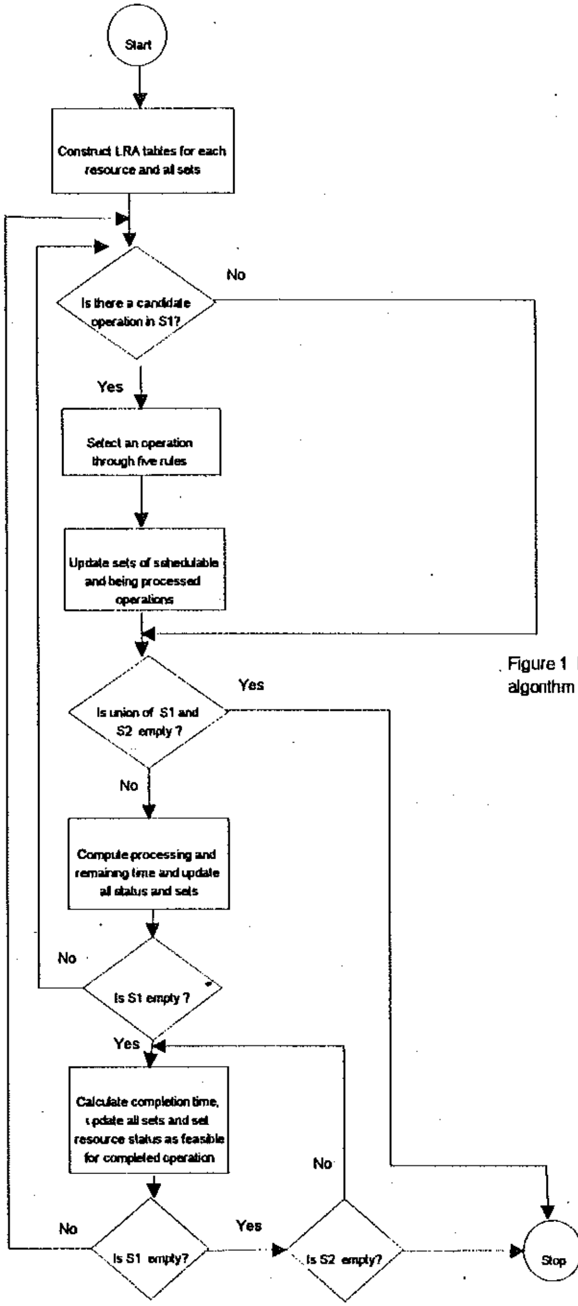

Figure 1 Flowchart of the proposed algorithm

- Due dates are not imposed.

The following notation is used for resources:

r_{1q} = machine (resource type 1), q=1,2,3

r_{2q} = tool (resource type 2), q=1,2,3

The machine and tool status are represented as follows:

sr_{1q} = machine status, q=1,2,3

sr_{2q} = lool status, q=1,2,3

t_i = the processing time $t_i^{(0)}$ of operation i in the basic process plan.

The basic process plans of the four parts are as follows:

$PP_1^{(0)}$= [(1,4,2,2),(2,5,1,3),(3,2,3,2)]

$PP_2^{(0)}$= [(4,6,1,3),(5,3,2,2),(6,3,3,1)]

$PP_3^{(0)}$= [(7,3,3,1),(8,3,1,2),(9,6,3,1),(10,2,1,3)]

$PP_4(0)$= [(11,4,3,2),(12,3,2,3)]

Note that for any triplet in the basic process plans, the first elements denotes operation number, the second denotes processing time and the third pair denotes the required machine number (resource type 1) and tool number (resource type 2),

The alternative process plans for the four parts are:

$PP_1^{(1)}$= [(1,6,3,1),(2,6,2,2),(3,4,1,1)]

$PP_1^{(2)}$= [(1,7,1,3),(2,7,1,2),(3,5,1,3)]

$PP_2^{(1)}$= [(4,6,1,3),(5,3,2,2),(6,3,3,1)]

$PP_2^{(2)}$= [(4,8,3,1),(5,8,1,3),(6,5,2,3)]

$PP_3^{(1)}$= [(7,4,3,2),(8,5,3,3),(9,7,2,1),(10,2,3,2)]

$PP_3^{(2)}$= [(7,4,2,2),(8,5,2,1),(9,9,1,3),(10,4,1,2)]

$pp_4(1)$= [(11,4,1,3),(12,5,1,2)]

$pp_4(2)$= [(11,4,3,1),(12,6,3,3)]

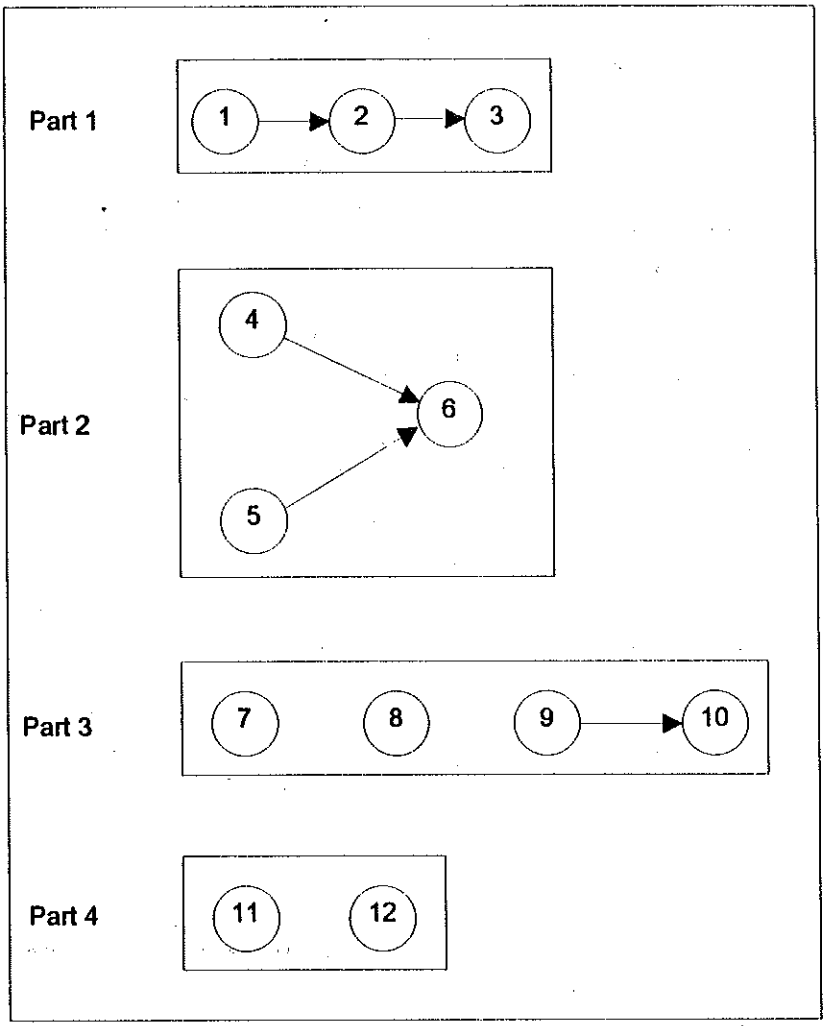

Figure 2. Parts with operations and precedence constraints.

Solution procedure

Step 0. Set current time t=0 and $sr_{lq}=1$, l=1,2, q=1,2,3. and construct the LRA table for each resource (l=1,2) as follows:

Table 2. LRA table for machines.

	01	02	03	04	05	06	07	08	09	10	11	12
M 1	7 -3 VI	5 1 III 7 -2 V	4 -2 V 5 -3 VI	6 2 1	8 - 5 VII	5 -2 V		3 2 1	9 -3 VII	2 2 1 4 -2 V	4 0 IV	5 -2 V
M 2	4 2 1	6 - 1 III		6 2 1	3 1 II	5 -2 IV	4 - 1 III	5 -2 IV	7 -1 III			3 2 1
M 3	6 -2 V		2 2 1	8 -2 V	4 - 1 IV	3 2 1	3 1 II 4 - 1 IV	5 -2 V	6 1 II	2 2 1	4 0 III 4 0 III	6 -3 VI

Table 3. LRA table for tools.

	01	02	03	04	05	06	07	08	09	10	11	12
M 1	6 -2 V		4 -2 V	8 -2 V	4 -1 IV	3 2 I	3 1 II	5 -2 V	6 1 II 7 -1 IV		4 0 III	
M 2	4 2 I	6 -1 IV 7 -2 V	2 2 I	6 2 I	3 1 II	5 -2 V	4 -1 IV 4 -1 IV	3 2 I		2 2 I 4 -2 IV	4 0 III	5 -2 V
M 3	7 -3 V	5 1 II	5 -3 V	6 2 I	8 -5 VI	5 -2 IV		5 -2 IV	9 -3 V	2 2 I	4 0 III	3 2 I 6 -3 VI

Step 1. Construct the following two sets:

$$S_0 = \{2,3,6,10\}$$

$$S_1 = \{1,4,5,7,8,9,11,12\}$$

Step 2. Using scheduling rule 1, operations, 1,4,7,8,9,12, are selected and rule 2 is triggered. According to the second rule, operation 1 is selected.

Step 3. Set:

$$s_1=2$$

$$s_2=0,\ s_3=0$$

Since $S_0 \cup S_1 = \{2,3,4,5,6,7,8,9,10,11,12\} \neq \varnothing$, set:

$$rt_1 = t_1 = 4$$

$$sr_{12} = 0,\ sr_{22} = 0$$

Set of schedulable operations $S_1 = \{4,7,8,9\} \neq \varnothing$. Go to step 2.

Step 2. Using scheduling rules 1 and 2, operation 4 is seleeted.

Step 3. Set:

$$s_4=2$$

$$s_5=0,\ s_6=0$$

Since $S_0 \cup S_1 = \{2,3,5,6,7,8,9,10,11,12\} \neq \varnothing$, set:

$$rt_4 = t_4 = 6$$

$$sr_{11} = 0,\ sr_{23} = 0$$

Set of sehedulable operations $S_1 = \{7, 9, 11\} \neq \varnothing$. Go to step 2.

Step 2. Using rules 1 and 2, operation 9 is seleeted.

Step 3. Set:

$$s_9=2$$

$$s_7=0,\ s_8=0,\ s_{10}=0$$

Since $S_0 \cup S_1 = \{2,3,5,6,7,8,10,11,12\} \neq \varnothing$, set:

$$rt_9 = t_9 = 6$$

$$sr_{13} = 0,\ sr_{21} = 0$$

Since ali resources aren't idle set of sehedulable operations $S_1 = \varnothing$. Go to step 4.

Step 4. Construct $S_2 = \{1,4,9\}$.

- Calculate completion time $f_1 = 4$, $f_4 = 6$, $f_9 = 6$.
- Sel current lime $t = f_1 = \min\{4,6,6\} = 4$.
- Set $s_1 = 3$.
- Dclete operation 1 from S_2.
- Set $sr_{12} = 1$, $sr_{22} = 1$.
- Set remaining time $rt_4 = 6 - 4 = 2$, $rt_9 = 6 - 4 = 2$.

Update $S_1 = \{2\}$.

Step 5. Since $S_1 \cup S_0 = \{2,3,5,6,7,8,10,11,12\} \neq \varnothing$, go to step 6.

Step 6. $S_1 = \{2\} \neq \varnothing$, go to step 2.

Step 2. Using rule 1, operation 2 is seleeted.

After seven iteralions, details are given in table 4, the gantt chart of the final schedule obtained and is shown in figure 3.

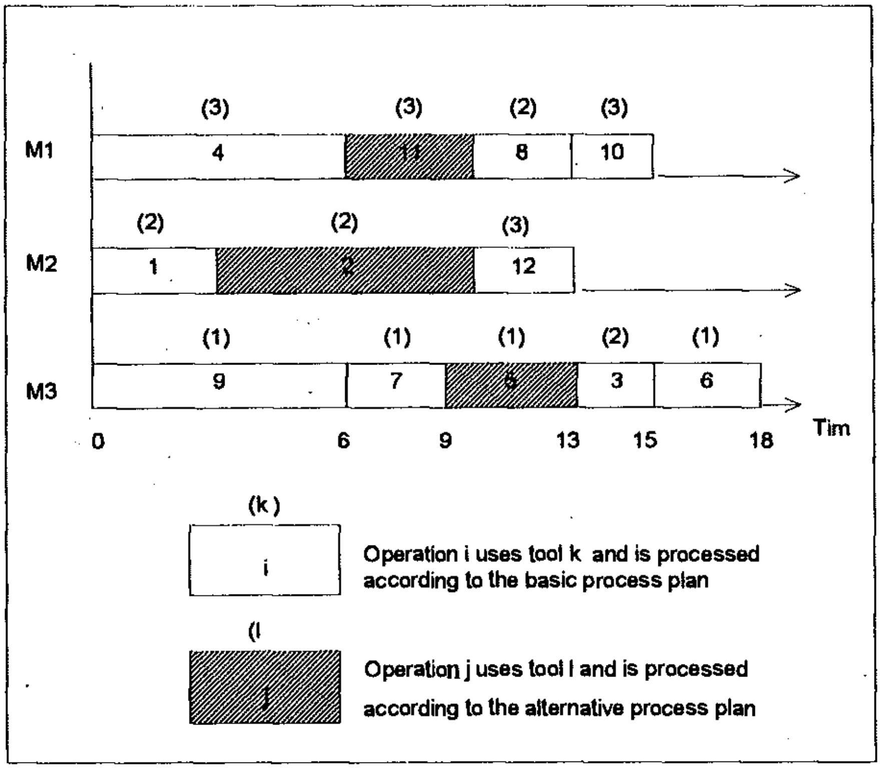

Figure 3. The final schedule

Table 4. Details of the iteralions

iteration number	1	2	3	4	5	6	7
selected operations	1,4,9	2	7,11	5	12, 8	10, 3	6

5. CONCLUSION

In order to cvaluale thc quality of solutions generated by Kusiak's KBSS and modified algorithm, seven measures of performance were used:

- Maximum flow time (F_{max})

$$F_{max} = \max_{1q}\{F_{1q}\}$$

- Average flow time (F)

$$F = \sum_{1q=1}^{N} F_{1q} / N$$

- Latencss of thc parts (L_k)

$$L_k = C_k - d_k$$

C_k = *Completion time of part k.*

- Tardiness (T_k)

$$T_k = max\{ C_k - d_k, 0 \}$$

- Machine utilization (U_m)

$$U_m = \sum_{1q=1}^{N} U_{1q} / N$$

Where

$$U_{1q} = \sum_{1q \in M_{(1q)}} t_i^{(v)} / F_{1q}$$

$M_{(1q)}$ = the set of operations processed on machine 1q.

• Basic and alternative process plan utilization (U_{bp}, U_{ap})

$$U_{bp} = n_{bp} / (n_{bp} + n_{ap})$$

$$U_{bp} = n_{ap} / (n_{bp} + n_{ap})$$

where n_{bp}= number of processed operations using basic process plan.

n_{ap}= number of processed operations using alternative process plan.

Due dates are assumed of four parts as 17,17,19 and 10 . Computational results of the two algorithm for numerical example are presented in Table 5. Modified algorithm gives us a better results than Kusiak's KBSS but it has to be tested on real and complex job shop problems. The real application of modified algorithm will be presented in near future.

Table 5. Computational results of the two algorithms

Measures of Performance	Kusiak's KBSS	Modified algorithm
F_{max}	21	18
F	18.6	15.334
L_1	-5	-2
L_2	0	1
L_3	2	-4
L_4	8	3
T_k	8	3
U_m	.968	1
U_{bp}	.417	.75
U_{ap}	.583	.25

References

Alfano, M.; Genco A.; Lopes S.; Prestigiacomo, A., Scheduling Simulation on a Parallel Virtual Machine, Proceedings of the European Simulation Symposium 94', 1994, Boğaziçi University, İstanbul.

Alptekin, S.; Rabelo L.C., Integrating simulation with artificial neural networks for flexible manufacturing system scheduling, Proceedings of the Advances in simulation '92 Symposium: Ali Rıza Kaylan, Tuncer Ören (eds),1992, Boğaziçi University, İstanbul.

Arzi,Y.; Roll, Y., Real-time production control of an FMS in a produce to order environment, International Journal of Production Research, 1993, Vol.:31, No:9, 2204-2205.

Kusiak, A., Intelligent Manufacturing Systems, New Jersey: Prentice-Hall, Inc., 1990.

Turksen, I.B.; Ulguray, D.; Wang, Q., Hierarchical scheduling based on approximate reasoning - A comparison with ISIS, Fuzzy Sets and Systems, 1992,Vol.:46, 349-371.