Management Science Letters 8 (2018) 543–554

Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

Implementation of heuristic algorithms to synchronized planning of machines and AGVs in FMS

K. Prakash Babu[a,b], V. Vijaya Babu[a] and Nageswara Rao Medikondu[c*]

[a]*Mechanical Department, College of Engineering, Andhra University, Visakhapatnam - 530 003, India*
[b]*Mechanical Department, VRSEC, Vijayawada - 520007, India*
[c]*Mechanical Department, KLEF, Guntur - 522502, India*

C H R O N I C L E

Article history:
Received: January 11, 2018
Received in revised format: March 3, 2018
Accepted: May 4, 2018
Available online: May 4, 2018

Keywords:
Flexible manufacturing system
Heuristic algorithms
Makespan
AGVs

A B S T R A C T

Flexible Manufacturing System (FMS) is a compound system containing essentials like workplaces, computerized storing and recovery systems, and material control devices such as automatons and automated guided vehicles (AGVs). In this paper, an attempt is made to study concurrently the machine and vehicle planning features in an FMS for minimization of the makespan. Planning is concerned with the distribution of partial resources to tasks over time and it is a resolution making procedure. It associates the processes, time, cost and overall purposes of the company. In this work, Nawaz-Enscore-Ham (NEH) heuristic algorithm is implemented to solve the scheduling problems in FMS. Eighty two problems and their existing solutions with different approaches are examined. The preliminary results indicate that the NEH heuristic algorithm provides better solutions with less computational time.

© 2018 by the authors; licensee Growing Science, Canada

1. Introduction

Scheduling is an essential decision creating process used on a systematic basis in different manufacturing and service industries. It deals with distribution of resources to jobs for the quantified time periods improving one or more objectives. The possessions and tasks in industries are in different forms. The possessions are the machines in a workshop, handling units in a computing environment, etc. The jobs are the operations in a production process, implementation of computer programs, etc. Each job will have definite precedence level, preliminary time and due date. One of the objectives is to minimize the makespan with respect to due dates. Johnson (1954) first introduced an algorithm that can find the best sequence for an n-job and 2-machine problem. Palmer (1965) developed an algorithm for minimizing the makespan based on the slope index and sorting in non-increasing order used in m-machine and n-job flow shop scheduling problems. The simple Johnson algorithm drawn-out by Campbell et al. (1970) uses a number of iterations before reaching the final result, which is broadly used and generally known as Campbell, Dudek and Smith (CDS) heuristics. Gupta (1971) recommended a distinct algorithm for reducing the make span. He implemented an altered method to obtain the slope index, based on which the sequencing of jobs is performed in the flow shop environment. Nawaz et al. (1983) established an algorithm based on the concept that the job with the maximum process time has the highest lead over the

* Corresponding author. Tel: +917658940234
E-mail address: medikondu1979@gmail.com (N. Rao Medikondu)

© 2018 by the authors; licensee Growing Science, Canada
doi: 10.5267/j.msl.2018.5.001

others for reducing make span, which is generally known as NEH heuristics algorithm. Nagar et al. (1996) proposed a different method for the minimization of mean flow time along with the minimization of make span in a flow shop environment by using of two altered approaches viz., branch and bound method and the genetic algorithm (GA) to reach the required objective. Nowicki and Smutnicki (1996) executed tabu search for solving the flow-shop scheduling problem. Neppalli et al. (1996) used the basic evolutionary procedure for solving the two machine and *n* jobs problems by minimizing the make span. Jungwattanakit et al. (2005) estimated the sequencing of heuristics for FMS scheduling problems. Biskup and Herrmann (2008) spawned a model using due dates as restraints valid only to single machine problems which are earlier sequence dependent models. The purpose behind this model is to reduce the penalties sustained where the demand is not satisfied within the due date. He and Hui (2008) used a metaheuristic algorithm for the scheduling of consignment plants along with similar units. They deliberated a large size problem and solved using GA method. Eren and Guner (2008) produced a model for a two machine flow shop problem and approach the idea of learning to minimize the total completion time and the makespan. Tseng and Liao (2008) measured *m*-machine and *n*-job flow shop scheduling problem to reduce the total earliness and lateness. They used particle swarm optimization method reducing the biased earliness and biased lateness as per the enterprise necessities. Wu and Zhou (2008) measured a stochastic scheduling method to get the needed schedule of jobs. They used a stochastic environment along with a single machine to bring out the target of minimizing the tardiness for the accomplishment of the job. Mosheiov and Sarig (2009) restrained several cost aspects (viz., earliness, delay, lateness, number of slow jobs and the modern due date demand) and scheduled a series of jobs through minimization of due dates, tardiness and additional cost factors. Cheng and Lin (2009) measured several methods including Johnson's rules, the concept of relocation and the introduction of composite jobs for solving the flow shop scheduling problem. Preciously jobs are produced with the equal idle time as that of the actual working machine which aids in minimizing the make span. Wu and Lee (2009) reduced the total completion time in various flow shop scheduling problems with presenting the concept of learning. Li et al. (2009) recommended a method for attaining the best solution through minimization of the total flow time. This method contains hybrid heuristic models while solving for total flow time. Modrak et al. (2013) presented some evaluation between the several heuristics algorithms from the makespan output. The NEH algorithm requires maximum iterations, whereas the Palmer algorithm requires only one iteration to reach to the definite results. Through Palmer algorithm is very fast it lacks accuracy.

2. Simultaneous scheduling through heuristic algorithm

The job shop and flow shop scheduling problems are inspected for amassed the efficiency of the machines and attaining the optimal processing data. Some methods have been measured in the current study. Johnsons (1954) two-machine problem with the objective of reducing makespan, essentially divides the jobs into two groups and orders them from left to right and right to left correspondingly. Palmer (1965) grade of jobs is created on a slope index calculated from the processing times there by giving preference to jobs that tend to progress from small to great processing times. Campbell (1970) simple algorithm will be suitable to attain solution of enormous sequence problems without computers. It affords inexact solutions to the n job, m machine sequencing problems it reflects no transient. The principle is least elapsed time up to m-1 sequences. Gupta (1971) improved the Palmer's slope index arranging n items based on a heuristic. Dannenbring (1977) established rapid access technique to syndicate the advantages of Palmers slope index and the Campbell methods. Nawaz (1983) established heuristic for the scheduling problem with the makespan minimization criterion for m machines and n jobs. Ronconi (2004) established a Min Max (MM) algorithm addressing flow shop makespan minimization problem with no barriers. In this work NEH algorithm is altered to solve simultaneous scheduling problems in FMS environment.

2.1 Nawaz Enscore Ham (NEH) Algorithm

This algorithm is identified as insertion algorithm for make span minimization, whose procedure is highlighted below.

Step 1:- Compute the total summation of processing time for each job.
Step 2:- Sort the jobs in the descending order of processing times.
Step 3:- Take the first two jobs in the arranged sequence and formulate altered combinations.
Step 4:- Compute the make span for each of the combination.
Step 5:- Select the combination with smallest make span.
Step 6:- Introduce the next job from the sequence found in Step 2.
Step 7:- Bring out all possible combination of the other jobs now.
Step 8:- Reprise Step 4 followed by Step 5 followed by Step 6.
Step 9:- Endure the process till all jobs are finished.

FMS Description

The FMS in this study has the arrangements shown in Fig. 1 There are four machines having Computer Numerical Controlled Machines (CNCs) each furnished with a sovereign and self-adequate tool magazine, one Automatic Tool Changer (ATC) and one Automatic Pallet Changer (APC).

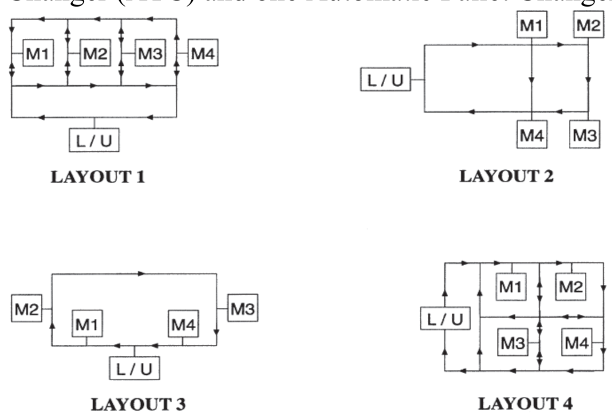

Fig. 1. Layout arrangements in example problems

Objective Function

The objective is to reduce the makespan and the formulae used are given as follows,
Process completion time

$$O_{ij} = T_{ij} + P_{ij}, \tag{1}$$

Job completion time

$$(C_i) = \sum_{i=1}^{n} O_{ij}, \tag{2}$$

$$\text{Makespan} = \max(C_1, \cdots, C_n), \tag{3}$$

where j=operation, i=job, T_{ij}=travelling time and P_{ij}=operation processing time.

Input Data

The input data (i.e. travelling time matrix) of Table 1 and Job sets of Bilge and Ulusoy (1995) are deliberated in Table 2. Data in Table 1 gives the distances from load/unload stations to machines and distances among machines in meters for all the four layouts .The Ten job sets in Table 2 each having four to eight altered job sets, machines in each job set and numbers with in the aside is the processing time of specific job on quantified machine. The load/unload (L/U) station serves as a distribution center for parts not yet processed and as a gathering center for parts finished. All vehicles start originally from the L/U station. Trips follow the shortest path among two points either between two machines or among a machine and the L/U station. Prevention of trips is not endorsed. The trips are called loaded or deadheading (empty) trips. The durations for the deadheading trips are sequence dependent and are unknown until the vehicle route is specified

Table 1
Travel time matrix for this particular problem

Layout-1					
From/To	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

Layout-2					
From/To	L/U	M1	M2	M3	M4
L/U	0	4	6	8	6
M1	6	0	2	4	2
M2	8	12	0	2	4
M3	6	10	12	0	2
M4	4	8	10	12	0

Layout-3					
From/To	L/U	M1	M2	M3	M4
L/U	0	2	4	10	12
M1	12	0	2	8	10
M2	10	12	0	6	8
M3	4	6	8	0	2
M4	2	4	6	12	0

Layout-4					
From/To	L/U	M1	M2	M3	M4
L/U	0	4	8	10	14
M1	18	0	4	6	10
M2	20	14	0	8	6
M3	12	8	6	0	6
M4	14	14	12	6	0

Table 2
Data for the Job Sets Used in Example Problems

JobSet-1 Job 1: Ml(8); M2(16); M4(12) Job 2: Ml(20); M3(10); M2(18) Job 3: M3(12); M4(8); Ml(15) Job 4: M4(14); M2(18) Job 5: M3(10); Ml(15)	JobSet-2 Job 1: Ml(10); M4(18) Job 2: M2(10); M4(18) Job 3: Ml(10); M3(20); Job 4: M2(10); M3(15); M4(12) Job 5: Ml(10); M2(15); M4(12) Job 6: M1(10); M2(15); M3(12)
JobSet-3 Job 1:Ml(16); M3(15) Job 2:M2(18); M4(15) Job 3:Ml(20); M2(10) Job 4:M3(15); M4(10) Job 5:Ml(8);M2(10);M3(15);M4(17) Job 6: M2(10);M3(15);M4(8);Ml(15	JobSet-4 Job1: M4(11); Ml(10); M2(7) Job2: M3(12); M2(10); M4(8) Job3: M2(7); M3(10); Ml(9); M3(8) Job4: M2(7); M4(8); Ml(12);M2(6) Job5:Ml(9);M2(7);M4(8);M2(10);M3(8)
JobSet-5 Job 1: Ml(6);M2(12);M4(9) Job 2: Ml(18);M3(6); M2(15) Job 3: M3(9);M4(3);Ml(12) Job 4: M4(6);M2(15) Job 5: M3(3);Ml(9)	JobSet-6 Job 1: Ml(9); M2(11); M4(7) Job 2: Ml(19); M2(20); M4(13) Job 3: M2(14); M3(20); M4(9) Job 4: M2(14); M3(20); M4(9) Job 5: Ml(11); M3(16); M4(8) Job 6: Ml(10); M3(12); M4(10)
JobSet-7 Job 1: Ml(6); M4(6) Job 2: M2(11); M4(9) Job 3: M2(9); M4(7) Job 4: M3(16); M4(7) Job 5: Ml(9); M3(18) Job 6: M2(13); M3(19); M4(6) Job 7: Ml(10); M2(9); M3(13) Job 8: Ml(11); M2(9); M4(8)	JobSet-8 Job 1: M2(12); M3(21);M4(11) Job 2: M2(12); M3(21);M4(11) Job 3: M2(12); M3(21);M4(11) Job 4: M2(12); M3(21);M4(11) Job 5: Ml(10); M2(14);M3(18);M4(9) Job 6: Ml(10);M2(14); M3(18);M4(9)
JobSet-9 Job 1: M3(9);Ml(12);M2(9);M4(6) Job 2: M3(16);M2(11); M4(9) Job 3: Ml(21); M2(18); M4(7) Job 4: M2(20); M3(22); M4(11) Job 5:M3(14);Ml(16);M2(13); M4(9)	JobSet-10 Job1:Ml(11);M3(19);M2(16);M4(13) Job2: M2(21);M3(16); M4(14) Job3:M3(8); M2(10); Ml(14); M4(9) Job4: M2(13); M3(20); M4(10) Job5: Ml(9); M3(16); M4(18) ; Job6:M2(19);Ml(21); M3(11);M4(15)

2.2 Vehicle scheduling methodology

Jobs are prepared based on the operation sequence derivative by the NEH heuristic algorithm. Mainly AGVs (Chawla et al., 2018) assignment jobs from the load/unload station to the specific workstations where the first processes are scheduled. AGVs make two types of trips, a loaded trip where it transports a load and a deadheading trip where the vehicle moves to pick up a load.

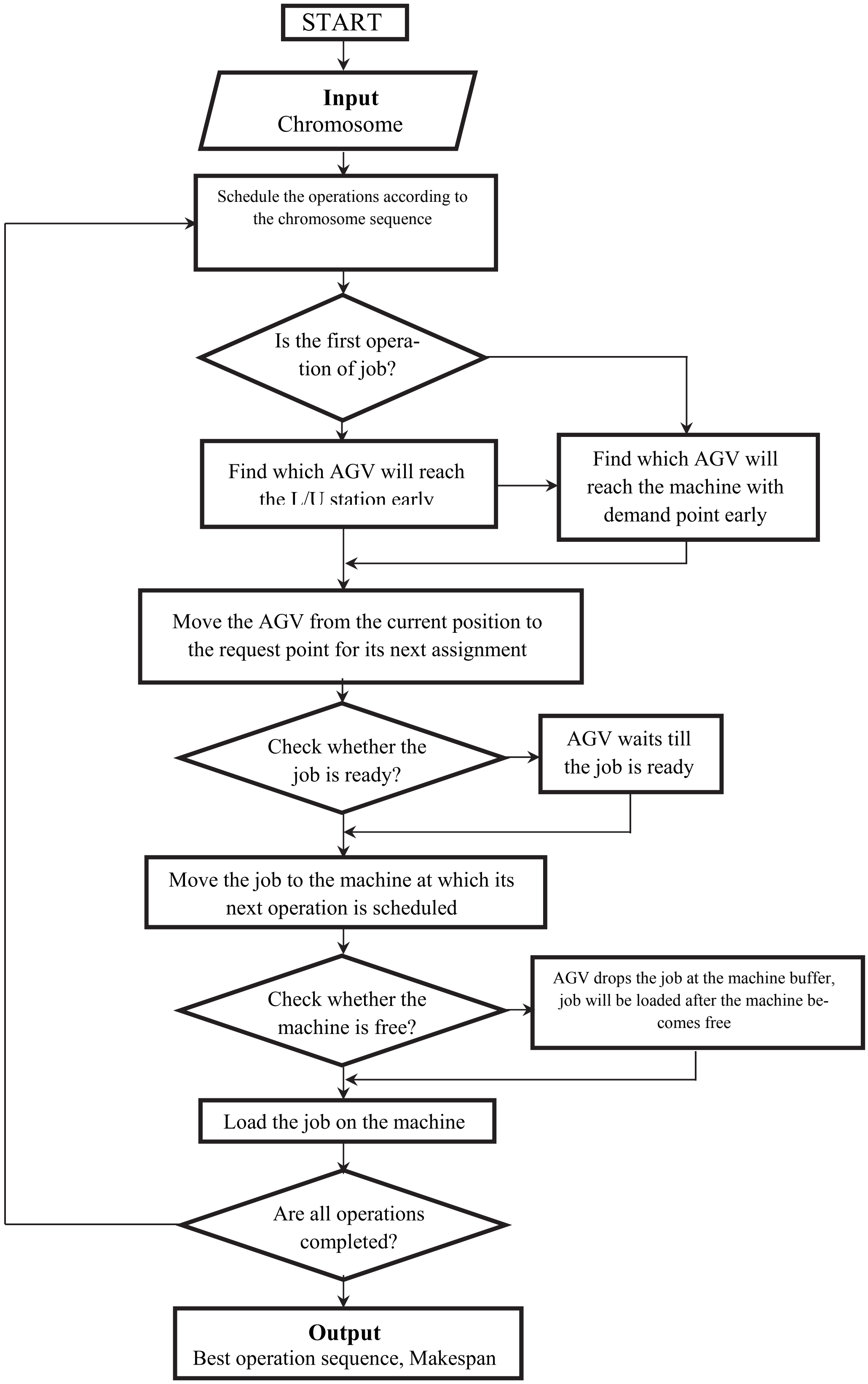

Fig. 2. Flow Chart for Synchronized scheduling of machines and AGVs

Deadheading trip can start instantaneously after the delivery and vehicle demand at different workstations are leisurely and the succeeding tasks are made. If both AGVs are accessible task is allocated to the initial vacant vehicle. If no vehicle is vacant, the earliest available times of the AGVs are calculated and the assignment is made. If the vehicle is idle and no job is complete, allocate the operation that is going to be accomplished early and is recognized the vehicle is stimulated to pick up that job. This kind of vehicle scheduling procedure helps in falling the waiting times and thus helps in refining the resource utilization and the throughput. The flow chart of the vehicle assignment procedure is given in Fig.2

2.3. Synchronized scheduling of machines and AGVs through heuristic algorithm

The NEH Heuristic Algorithm is implemented to the simultaneous scheduling problems. The basic input utilized to study this aspect is explained earlier.

Synchronized Scheduling - NEH Heuristic Algorithm

For the implementation of NEH Heuristic Algorithm, job set 1 and layout 1 are considered as an example. NEH Heuristic Algorithm constructs jobs sequence in an iterative manner. The iterations are sustained till all jobs from the gratified list are placed in the limited sequence. The NEH Heuristic is explained in the following steps for the job set 1:

Step 1: Considering the job set

Job Set No	Layout	No of Jobs	No of Operations	Sequence of Machines	Sequence of Operations
1	1	5	13	Job 1: 1-2-4 Job 2: 1-3-2 Job 3: 3-4-1 Job 4: 4-2 Job 5: 3-1	Job 1: 1-2-3 Job 2: 4-5-6 Job 3: 7-8-9 Job 4: 10-11 Job 5: 12-13

Step 2: The total sum of processing time for each job are calculated.

Job.No	Total Processing time
1	36
2	48
3	35
4	32
5	25

Step 3: The jobs are sorted in the decreasing order of processing times.

Job.No	Total Processing time
2	48
1	36
3	35
4	32
5	25

Step 4: The last two jobs are taken in the sorted sequence and different combinations are formulated.

Combination 1: Job 5 and Job 4, Combination 2: Job 4 and Job 5 it means operation numbers 10, 11, 12 and 13.

Step 5: The make span for each combination is calculated.

Combination 1: Job 5 and Job 4

O.No	M.No	V.No	VPL	POMN	VRT	POCT	VET	Max(7,8)	VLT	MRT	Max(10,11)	Process Time	Make Span
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
12	3	1	L/U	L/U	0	0	0	0	10	0	10	10	20
13	1	2	L/U	3	0	20	10	20	28	0	28	15	43
10	4	1	3	L/U	10	0	18	18	30	0	30	14	44
11	2	1	4	4	30	44	30	44	52	0	52	18	70

Makespan: 70

Combination 2: Job 4 and Job 5

O.No	M.No	V.No	VPL	POMN	VRT	POCT	VET	Max(7,8)	VLT	MRT	Max(10,11)	Process Time	Make Span
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
10	4	1	0	0	0	0	0	0	12	0	12	14	26
11	2	2	0	4	0	26	12	26	34	0	34	18	52
12	3	1	4	0	12	0	18	18	28	0	28	10	38
13	1	1	3	3	28	38	28	38	46	0	46	15	61

Makespan: 61

It is found that the minimum makespan is with combination '2'

Step 6: Now job 3 has been added to this combination and the same procedure has been reputed for obtaining makespans for each other combinations obtained by changing the sequences.

Combination	Sequence	Value of Makespan
1: Job 3 - Job 4 - Job 5	7 -8-9-10-11-12-13	101
2: Job 4 - Job 3 - Job 5	10-11-7-8-9-12-13	95
3: Job 4 - Job 5 - Job 3	10-11-12-13-7-8-9	105

It is found that the minimum makespan is with combination '2'

Step 7: Now job 1 has been added to this combination and the same procedure has been reputed for obtaining makespan for each other combinations obtained by changing the sequences

Combination	Sequence	Value of Makespan
1: Job 1- Job 4 - Job 3 - Job 5	1-2-3-10-11-7-8-9-12-13	131
2: Job 4- Job 1 - Job 3 - Job 5	10-11-1-2-3-7-8-9-12-13	136
3: Job 4- Job 3 - Job 1 - Job 5	10-11-7-8-9-1-2-3-12-13	146
4: Job 4- Job 3 - Job 5 - Job 1	10-11-7-8-9-12-13-1-2-3	145

It is found that the minimum makespan is with combination '1'

Step 8: Now job 2 has been added to this combination and the same procedure has been reputed for obtaining makespan for each other combinations obtained by changing the sequences

Combination	Sequence	Value of Makespan
1: Job 2- Job 1- Job 4 - Job 3 - Job 5	4-5-6-1-2-3-10-11-7-8-9-12-13	179
2: Job 1- Job 2- Job 4 - Job 3 - Job 5	1-2-3-4-5-6-10-11-7-8-9-12-13	167
3: Job 1- Job 4 - Job 2- Job 3 - Job 5	1-2-3-10-11-4-5-6-7-8-9-12-13	165
4: Job 1- Job 4 - Job 3 - Job 2- Job 5	1-2-3-10-11-7-8-9-4-5-6-12-13	190
5: Job 1- Job 4 - Job 3 - Job 5- Job 2	1-2-3-10-11-7-8-9-12-13-4-5-6	193

It is found that the minimum makespan is with combination '3' and for this combination the makespan calculation are shown below

O.No	M.No	V.No	VPL	POMN	VRT	POCT	VET =VRT+TRT1 (4 to 5)	Max(7,8)	VLT =VET+TRT2 (5 to 2)	MRT	Max(10,11)	Process Time	Make Span
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
1	1	1	0	0	0	0	0	0	6	0	6	8	14
2	2	2	0	1	0	14	6	14	20	0	20	16	36
3	4	1	1	2	6	36	12	36	44	0	44	12	56
10	4	2	2	0	20	0	30	30	42	56	56	14	70
11	2	2	4	4	42	70	42	70	78	36	78	18	96
4	1	1	4	0	44	0	50	50	56	14	56	20	76
5	3	1	1	1	56	76	56	76	84	0	84	10	94
6	2	1	3	3	84	94	84	94	100	96	100	18	118
7	3	2	2	0	78	0	88	88	98	94	98	12	110
8	4	2	3	3	98	110	98	110	116	70	116	8	124
9	1	1	2	4	100	124	108	124	134	76	134	15	149
12	3	2	4	0	116	0	122	122	132	110	132	10	142
13	1	2	3	3	132	142	132	142	150	149	150	15	165

O.No: Operations Number **M.No**: Machine Number **V.No**: Vehicle Number
VPL: Vehicle Previous Location **POMN**: Previous Operations Machine Number
VRT: Vehicle Ready Time **POCT**: Previous Opeartion Completion Time
VET: Vehicle Empty Trip **VLT**: Vehicle Loaded Trip **MRT**: Machine Ready Time

The above table shows the operation scheduling for the implementation of the NEH heuristic algorithm for job set 1 layout 1. From the table it is observed that operation 1 on machine 1 is completed by 14 min. Hence 2nd operation will start after completion of 1st operation on machine 1. In case of job set 1 and layout 1 operation 2 on machine 2 is completed by 36 min. Hence 3rd operation on machine 4 will start after the completion of 2nd operation on machine 2. Similarly no operation on the particular machine will start until the operation on the machine is completed. From the vehicle heuristic algorithm for the first two operations, AGVs are selected randomly and in case of the third operation, AGV '1' is selected based on the availability of AGV with minimum travel time and this constraint is checked in the proposed algorithm. For job set 2 and layout 2 the operational completion time (makespan) is 165.

3. Results and discussion

Ten dissimilar job sets with dissimilar processing orders, and process times are produced and accessible. Different amalgamations of these ten job sets and four layouts have been used to produce 82 example problems. In all these problems, there are two vehicles. Table 3 contains of problems whose t_i/p_i ratios are larger than 0.25, and Table 4 shows the results whose t_i/pi ratios are lesser than 0.25. A code has been used to entitle the example problems which are given in the first column. The numbers that follow 1.1 specify the job set and the layout. In Table 4 additional digit is attached to the code. Having 0 or 1 as the last digit indicates that the process times are doubled or tripled where in both cases travel times are condensed to half.

4. Conclusion

In this paper, the finest sequence of machines and AGVs were resolute by using NEH heuristic algorithm. It has been detected from Table 3 that 29 problems out of 40 have given superior results when compared with FCFS. 35 problems have given superior results when compared with SPT (Nageswara rao et al., 2017) and 33 problems have shown superior results when compared with LPT. It has detected from Table 4 that out of 42 problems, 30 ones have given superior results using NEH when compared with FCFS, 42 problems have given enhanced results when compared with SPT and 42 problems have given improved results when compared with LPT.

Table 3
Performance assessment for t/p>0.25

Job. No	t/p	FCFS	SPT	LPT	NEH
1.1	0.59	173	193	177	165
2.1	0.61	158	158	177	169
3.1	0.59	202	224	198	195
4.1	0.91	263	267	264	260
5.1	0.85	148	164	148	147
6.1	0.78	231	240	227	225
7.1	0.78	195	210	201	173
8.1	0.58	261	261	266	261
9.1	0.61	270	277	268	259
10.1	0.55	308	308	310	305
1.2	0.47	143	173	165	147
2.2	0.49	124	124	130	116
3.2	0.47	162	188	160	154
4.2	0.73	217	223	224	215
5.2	0.68	118	144	131	117
6.2	0.54	180	169	165	158
7.2	0.62	149	160	149	136
8.2	0.46	181	181	198	181
9.2	0.49	250	249	244	205
10.2	0.44	290	288	287	274
1.3	0.52	145	175	167	145
2.3	0.54	130	130	136	122
3.3	0.51	160	190	162	158
4.3	0.8	233	237	230	226
5.3	0.74	120	146	133	117
6.3	0.54	182	171	167	160
7.3	0.68	155	166	151	138
8.3	0.5	183	183	200	183
9.3	0.53	252	251	246	207
10.3	0.49	293	294	293	280
1.4	0.74	189	207	189	189
2.4	0.77	174	174	174	169
3.4	0.74	220	250	212	213
4.4	1.14	301	301	298	298
5.4	1.06	171	189	171	171
6.4	0.78	249	252	237	234
7.4	0.97	217	242	151	192
8.4	0.72	285	285	200	285
9.4	0.76	292	311	290	285
10.4	0.69	350	350	345	345

Table 4
Performance assessment for t/p<0.25

Job.No	t/p	FCFS	SPT	LPT	NEH
1.10	0.15	207	248	252	207
2.10	0.15	217	217	225	185
3.10	0.15	257	327	282	255
4.10	0.15	303	328	317	277
5.10	0.21	152	190	187	154
6.10	0.16	304	281	297	272
7.10	0.19	231	240	264	213
8.10	0.14	338	338	347	332
9.10	0.15	390	367	359	324
10.10	0.14	452	429	444	398
1.20	0.12	194	238	246	197
2.20	0.12	194	194	206	167
3.20	0.12	241	311	270	241
4.20	0.12	285	312	298	248
5.20	0.17	142	180	184	143
6.20	0.12	292	260	284	251
7.20	0.15	212	218	249	188
8.20	0.11	306	319	334	306
9.20	0.12	380	355	347	309
10.20	0.11	445	423	439	388
1.30	0.13	195	239	247	196
2.30	0.13	197	197	209	170
3.30	0.13	240	312	271	240
4.30	0.13	292	317	301	255
5.30	0.18	141	181	183	143
6.30	0.24	296	261	285	252
7.30	0.17	215	221	250	191
8.30	0.13	307	320	335	307
9.30	0.13	381	356	348	310
10.30	0.12	448	426	442	391
1.40	0.18	213	255	254	213
2.41	0.13	307	307	319	267
3.40	0.18	261	330	282	258
3.41	0.12	370	476	411	310
4.41	0.19	434	471	451	393
5.41	0.18	218	269	270	222
6.40	0.19	310	288	299	275
7.40	0.24	239	251	270	221
7.41	0.16	329	344	385	224
8.40	0.18	343	343	349	339
9.40	0.19	396	379	370	325
10.40	0.17	466	445	455	415

Acknowledgment

The authors greatly acknowledge the financial support from DST-SERB, Govt.of India (Sanction No: SB/EMEQ-501/2014) for carrying out This R & D activity.

References

Bilge, U., & Ulusoy, G. (1995). A time window approach to simultaneous scheduling of machines and material handling system in a FMS. *Operations Research, 43*(6), 1058–1070.

Biskup, D., & Herrmann, J. (2008). Single-machine scheduling against due dates with past-sequence dependent setup times. *European Journal of Operational Research*, *191*(2), 587–592.

Campbell, H.G, Dudek, R.A., & Smith M.L. (1970). A heuristic algorithm for the n job m machine sequencing problem. *Management Science*, *16*, B630–B637.

Chawla, V., Chanda, A., & Angra, S. (2018). Automatic guided vehicles fleet size optimization for flexible manufacturing system by grey wolf optimization algorithm. *Management Science Letters*, *8*(2), 79-90.

Cheng, T. C. E., & Lin, B. M. T. (2009). Johnson's rule, composite jobs and the relocation problem. *European Journal of Operational Research, 192*, 1008–10013.

Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science*, *23*(11), 1174-1182.

Eren, T., & Güner, E. (2008). A bicriterion flowshop scheduling with a learning effect. *Applied Mathematical Modeling*, *32*, 1719–1733.

Gupta, J. N. D. (1971). A functional heuristic algorithm for the f1owshop scheduling problem. *Operational Research*, *22*, 39-47.

He, Y., & Hui, C. W. (2008). A rule-based genetic algorithm for the scheduling of single-stage multiproduct batch plants with parallel units. *Computers and Chemical Engineering, 32*, 3067–3083.

Johnson, S.M. (1954). Optimal two-and-three-stage production schedules with set-up times included. *Naval Research Logistic, 1*, 61–68.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2005). An evaluation of sequencing heuristics for flexible flowshop scheduling problems with unrelated parallel machines and dual criteria. *Otto-von-Guericke-Universitat Magdeburg, 28*(05), 1–23.

Li, X., Wang, Q., & Wu, C. (2009). Efficient composite heuristics for total flow time minimization in permutation flow shops. *OMEGA, the International Journal of Management Science, 37*(1), 155–164.

Modrak, V., Semanco, P., & Kulpa, W. (2013). Performance Measurement of Selected Heuristic Algorithms for Solving Scheduling Problems. In: 11th International Symposium on Applied Machine Intelligence and Informatics, 205–209.

Mosheiov, G., & Sarig, A. (2009). Due-Date Assignment on Uniform Machines. *European Journal of Operational Research*, 193(1), 49-58.

Nagar, A., Heragu, S. S., & Haddock, J. (1996). A combined branch and bound and genetic algorithm based approach for a flow shop-scheduling problem. *Annals Operation Research*, *63*, 397–414.

Nageswararao, M., Narayanarao, K., & Rangajanardhana, G. (2017). Integrated scheduling of machines and agvs in fms by using dispatching rules. *Journal of Production Engineering*, *20*(1), 75-84.

Nawaz, M., Enscore Jr. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *OMEGA*, *11*(1), 91–95.

Neppalli, V. R., Chen, C. L., & Gupta, J. N. D. (1996). Genetic algorithms for the two stage criteria flowshop problem. *European Journal of Operational Research*, *95*(2), 356–373.

Nowicki, E., & Smutnicki, C. (1996). A fast tabu search algorithm for the permutation flowshop problem. *European Journal of Operational Research*, *91*(1), 160–175.

Palmer, D.S. (1965). Sequencing jobs through a multistage process in the minimum total time: A quick method of obtaining a near-optimum. *Operational Research*, *16*, 101–107.

Ronconi, D. P. (2004). A note on constructive heuristics for the flowshop problem with blocking. *International Journal of Production Economics, 87*(1), 39–48.

Stecke, K.E., & Solberg, J.J. (1981). Loading and control policies for a flexible manufacturing system. *International Journal of Production Research*, *19*(5), 481 – 490.

Tseng, C. T., & Liao, C. J. (2008). A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. *European Journal of Operational Research*, *191*(2), 360-373.

Wu, C.C., & Lee, W.C. (2009). A note on the total completion time problem in a permutation flow shop with a learning effect. *European Journal of Operational Research, 192*, 343–347.
Wu, X., & Zhou, X. (2008). Stochastic scheduling to minimize expected maximum lateness. *European Journal of Operational Research*, *190*(1), 103–115.

© 2018 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).