International Journal of Advanced Science and Technology
Vol. 29, No. 6, (2020), pp. 6983-7001

`

# Implementation of Heuristic Algorithms to Simultaneous Scheduling of Machines and AGVs in FMS

G. Durga Prasad[1], V. Mohanmanoj[2], Nageswara Rao Medikondu [3] , Varakumari.S [4]
*[1,2]Department of Mechanical Engineering NRIIT, Nunna, AP, India*
*[3]Department of Mechanical Engineering KLEF University, Guntur, AP, India*
*[4]Department of ECE, KLEF University, Guntur, AP, India*

***Abstract***

*Flexible Manufacturing System (FMS) is a complex system consisting of elements like workstations, automated storage and retrieval systems, and material handling devices such as robots and AGVs. In this paper, an attempt is made to consider simultaneously the machine and vehicle scheduling aspects in an FMS for minimization of the makespan. Scheduling is concerned with the allocation of limited resources to tasks over time. It is a decision making process. It links the operations, time, cost and overall objectives of the company. In this work, RAPID ACCESS (RA) heuristic algorithm is adopted to solve the scheduling problems in FMS. Eighty, two problems and their existing solutions with different approaches are examined. The NEH heuristic algorithm provides better solutions with less computational time.*

***Keywords:*** *Flexible Manufacturing System, Heuristic Algorithms, makespan, AGVs*

## I. INTRODUCTION

Scheduling is an important decision-making process being used on a regular basis in various manufacturing and service industries. It deals with allocation of resources to tasks for the specified time periods optimizing one or more objectives. The resources and tasks in industries are of different forms. The resources are the machines in a workshop, runways at an airport, crew at a construction site, processing units in a computing environment, and so on. The tasks are the operations in a production process, take-offs and landings at an airport, stages in a construction project, execution of computer programs, and so on. Each task will have certain priority level, starting time and due date. One of the objectives is the minimization of tasks completion with respect to due dates.

## II. LITERATURE SURVEY

Johnson (1954) has first presented an algorithm that can find optimum sequence for an n-job and 2-machine problem. Palmer (1965) has used a single iteration method for minimizing the makespan through sequencing of jobs based on the slope index and sorting in decreasing order useful tom-machine and n-job flow shop scheduling problems. The simple Johnson algorithm extended by Campbell et al. (1970) uses a number of iterations before reaching the final result, which is widely used and commonly known as Campbell, Dudek and Smith heuristics (CDS). Gupta (1971) has suggested a distinct algorithm for minimizing the completion time and the make span. He has adopted a different technique to obtain the slope index, based on which the sequencing of jobs is done in the flow shop environment. Nawaz et al. (1983) have developed a

`

model on the total processing time of the individual jobs. The priority of the jobs in the schedule is based on the concept that the job with the highest in process time has the greatest lead over the others. They have utilized this concept and suggested a heuristics for minimizing make span, which is commonly known as NEH heuristics algorithm. Nagar et al. (1996) have developed a distinct method for the minimization of mean flow time along with the minimization of make span in a flow shop environment. They have used two different methods viz., branch and bound method and the. Genetic algorithm to reach the required objective.Nowicki&Smutnicki (1996) have implemented tabu search while solving the flow-shop scheduling problem. Neppalli et al. (1996) have used the basic evolutionary algorithm for solving the two machine problem by minimizing the make span. Jungwattanakit et al. (2005) have evaluated the sequencing of heuristics for flexible flow shop scheduling problems. Biskup& Herrmann (2008) have generated a model using due dates as constraints applicable only to single machine problems which are past sequence dependent models. The motive behind this model is to minimize the penalties which are incurred where the demand is not fulfilled within the due date. He &Hui (2008) have used the evolutionary algorithm for the scheduling of a single-stage and multi-product batch plants along with parallel units. They have considered a large size problem and solved using genetic algorithm and finally proposed the heuristics approach. Eren&Guner, (2008) have created a model for a two machine flow shop problem and approach the concept of learning to minimize the total completion time and the makespan. Tseng & Liao, (2008) have considered m-machine and n-job flow shop scheduling problem to minimize the total earliness and tardiness. They have used particle swarm optimization technique minimizing the weighted earliness and weighted tardiness as per the company requirements. Wu & Zhou, (2008) have considered a stochastic scheduling approach to get the required schedule of jobs. They have used a stochastic environment along with a single machine to carry out the target of minimizing the lateness for the completion of the job. Mosheiov&Sarig (2009) have considered various cost factors (viz., earliness, lateness, tardiness, number of tardy jobs and the latest due date demand) and scheduled a sequence of jobs through minimization of due dates, lateness, tardiness, and other cost factors. Cheng & Lin (2009) have considered various methods including Johnson's rules, the concept of relocation and the introduction of composite jobs for solving the flow shop scheduling problem. Artificially jobs are created with the same idle time as that of the actual working machine which helps in minimizing the make span of the flow shop problem. Wu & Lee (2009) have minimized the total completion time in various flow shop introducing the concept of learning. Li et.al, (2009) have suggested a method for achieving the optimal solution through minimization of the total flow time. This method involves composite heuristics models while solving for total flow time in permutation flow shop environment. The jobs are to be processed in a particular sequence. The heuristics approach developed by Li et al (2009) has worked very well in minimizing the total flow time. Modrak et al. (2013) have made comparison between the various heuristics algorithms from the makespan output. The NEH algorithm requires maximum iterations, where as the Palmer algorithm needs only one iteration to reach to the definite results. Through Palmer algorithm is very fast it lacks accuracy.

## III. SIMULTANEOUS SCHEDULING THROUGH HEURISTIC ALGORITHM

The job shop and flow shop scheduling problems are examined for increasing the efficiency of the machines and obtaining the optimal processing data. Several methods have been considered in the present study. Johnsons (1954) two-machine problem with the objective of minimizing makespan, basically divides the jobs into two categories and sequences them from left to right and right to left respectively. Palmer (1965) ranking of jobs is based on a slope index computed from the processing times there by giving preference to jobs that tend to progress from low to high processing times. Campbell (1970) simple algorithm will be useful to obtain solution of large sequence problems without computers. It provides approximate solutions to the n job, m machine sequencing problems it considers no passing. The criterion is minimum elapsed time up to m-1 sequences. Gupta (1971) has modified the Palmer's slope index sorting n items based on a heuristic. Dannenbring (1977) has developed rapid access procedure to combine the advantages of Palmers slope index and the Campbell methods. Nawaz (1983) have developed heuristic for the permutation flow shop scheduling problem with the makespan minimization criterion for m machines and n jobs. Ronconi (2004) has developed a Min Max (MM) algorithm addressing flow shop makespan minimization problem with no buffers. In this work RA Heuristic algorithm is modified to solve simultaneous scheduling problems in FMS environment.

### 3.1 Rapid Access Heuristic (RAH) Algorithm

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

Step 1:- Mark the Least Processing Time for each job

- If the both the jobs processing times are same take the first one as priority

Step 2:- Divide the jobs into two groups based on their processing time namely U, V

- U group as follows (Ti A<Ti B)
- V group as follows (Ti A>Ti B)

Step 3:- Sortthe jobs "U" as follows which considering least processing times and these processing times are arranged with minimum process time to maximum process time.

Step 4:- Sort the jobs "V" as follows with consideration of least processing time on each job and arranging these processing times from maximum processing time to minimum processing time.

Step 5:- During the assignment period in U group if the least processing times are same (or) Tie select the job with least processing time as first priority. In V group if the least processing time are Tie (or) equal select the job with max. Total processing time as first priority

According to Johnson algorithm the optimal sequence is 'UV'

Step 6:- Continue the process till all jobs are completed.

`

## IV. FMS DESCRIPTION

The FMS in this study has the configurations shown in Fig. 1 There are four Machines having Computer Numerical Controlled Machines (CNCs) each equipped with an independent and self sufficient tool magazine, one Automatic Tool Changer (ATC) and one Automatic Pallet Changer (APC).

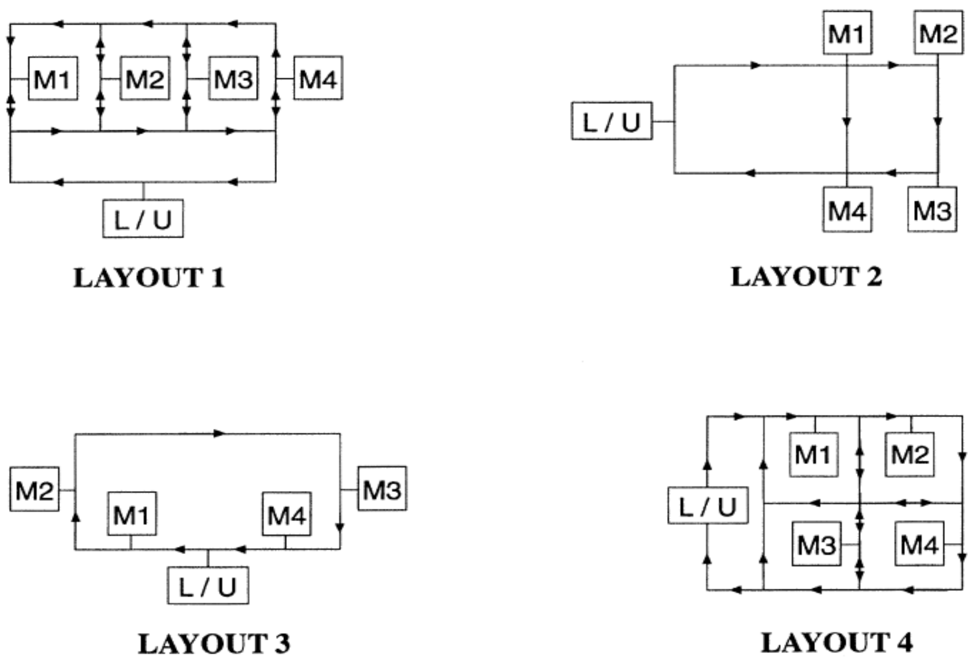



**Fig. 1 Layout configurations in example problems**

### 4.1 FMS Environment

The environment within which the FMS operates can be described as follows:

1. The types and number of machines are known. Operations are no preemptive. There is sufficient input/output buffer space at each machine
2. Processing, set-up, loading and unloading times are available and are deterministic.
3. Number of AGVs is given and the AGVs are all identical in the sense that they have the same speed and load carrying characteristics.
4. Flow path layout is given and travel times on each segment of the path are known.
5. A load/unload (L/U) station serves as a distribution center for parts not yet processed and it serves as a collection center for parts finished. All vehicles start from the L/U station initially and return after accomplishing all assignments. There is sufficient input/output buffer space at the (L/U) station.

`

6. AGVs carry a single unit-load at a time. They move along predetermined shortest paths, without delay due to congestion. Preemption of trips is not allowed. The trips are called loaded or deadheading (empty) trips depending on whether a part is carried or no part is carried during that trip, respectively. The durations for the deadheading trips are sequence dependent and are not known until the vehicle route is specified.

7. It is assumed that all the design and set-up issues within the hierarchy of OR/MS problems in an FMS as suggested by Stecke, & Solberg (1981) have already been resolved. Machine loading (i.e., the allocation of tools to machines and the assignment of operations to machines) is made. Pallets and other necessary equipment are allocated to parts. The set of part types to be produced during the planning period and the routing of each part type are available before making scheduling decision. In other words, routing flexibility is not considered. The routing for a part type can be selected based on considerations of technological feasibility and processing efficiency, or by formulating the set-up phase problems in a manner that can also handle the routing decisions.

8. Ready-times of all jobs are known. Initially, partially processed parts might be available at machines waiting for further processing, and they can be treated as jobs having zero ready times and their routing consists of the remaining operations.

9. Such issues as traffic control, congestion, machine failure or downtime, scraps, rework, and vehicle dispatches for battery change are ignored and left as issues to be considered during real-time control.

### 4.2 Assumptions

a. The types and no. of machines are known

b. There is sufficient input/output buffer space for each machine

c. Machine loading

    i. Allocation of tools to machine has been done

    ii. Assignment of operations to machine are made

d. Pallet and other necessary equipment are allocated

e. The speed of AGV kept at 40 m/min

f. The distance between the two machines and distance between loading/ unloading machines are known

### 4.3 Objective Function

The objective is to minimize the makespan and the formulae used are given below:

Operation completion time=Oij=Tij+Pij (1)

`

Job completion time

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

Makespan = max (C1, C2, C3….. Cn) (3)

Where j=operation, i=job, Tij=travelling time, Pij=operation processing time

### 4.4 Input Data

The input data (i.e. travelling time matrix) of Table 1 and Job sets of Bilge &Ulusoy (1995) are considered. Data in Table 1 gives the distances from load/unload stations to machines and distances between machines in metres for all the four layouts .The Ten job sets in Table 2 each having four to eight different job sets, machines in each job set and numbers with in the parenthesis is the processing time of particular job on specified machine. The load/unload (L/U) station serves as a distribution center for parts not yet processed and as a collection center for parts finished. All vehicles start initially from the L/U station. Trips follow the shortest path between two points either between two machines or between a machine and the L/U station. Preemption of trips is not allowed. 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 |

| Layout-4 | | | | | |
|---|---|---|---|---|---|
| From/To | L/U | M1 | M2 | M3 | M4 |
| L/U | 0 | 4 | 8 | 10 | 14 |

`

| | | | | | |
|---|---|---|---|---|---|
| M1 | 12 | 0 | 2 | 8 | 10 |
| M2 | 10 | 12 | 0 | 6 | 8 |
| M3 | 4 | 6 | 8 | 0 | 2 |
| M4 | 2 | 4 | 6 | 12 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| 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**

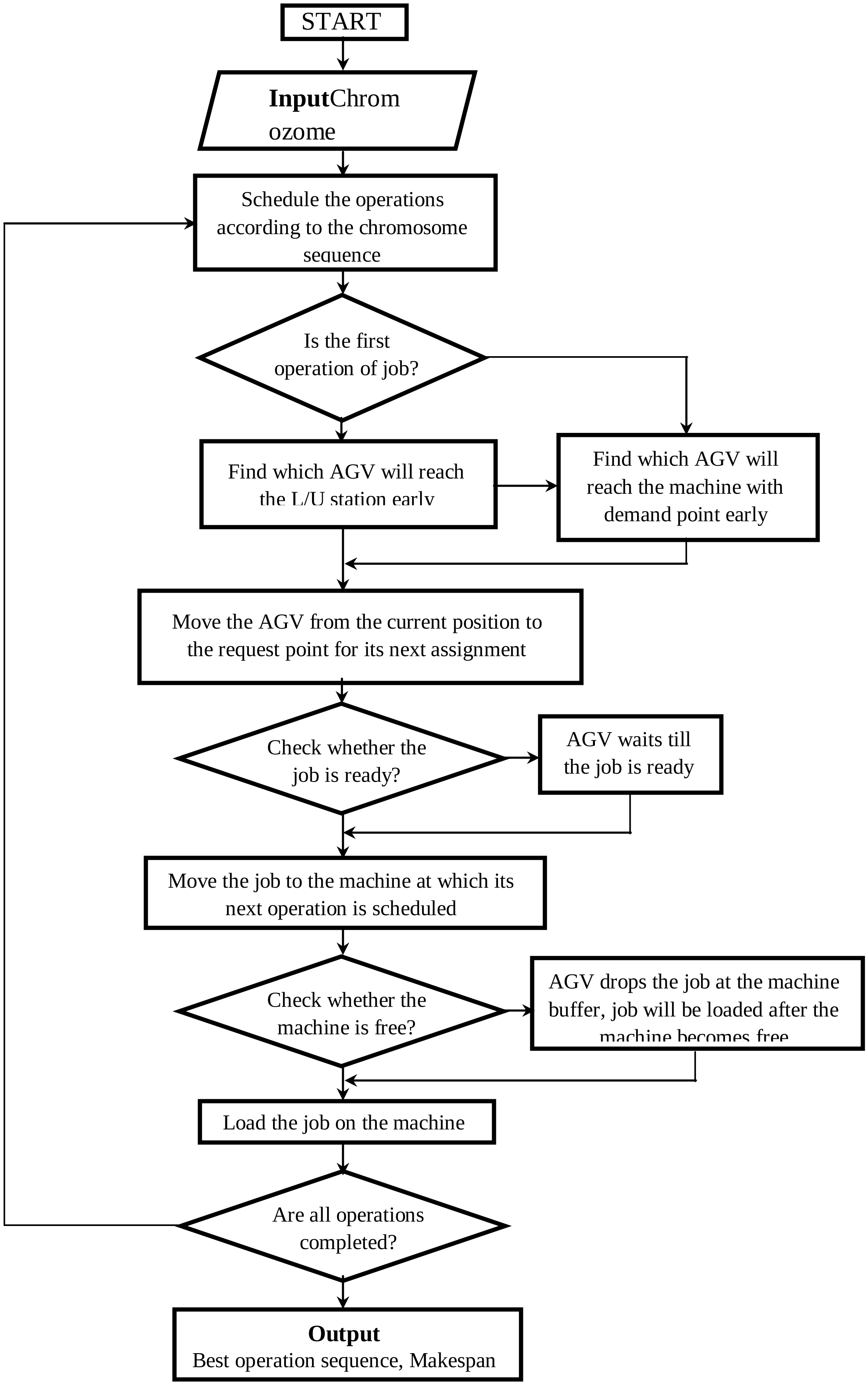
| | |
|---|---|
| JobSet-1<br>Job 1: Ml(8); M2(16); M4(12)<br>Job 2: Ml(20); M3(10); M2(18)<br>Job 3: M3(12); M4(8); Ml(15)<br>Job 4: M4(14); M2(18)<br>Job 5: M3(10); Ml(15) | JobSet-2<br>Job 1: Ml(10); M4(18)<br>Job 2: M2(10); M4(18)<br>Job 3: Ml(10); M3(20);<br>Job 4: M2(10); M3(15); M4(12)<br>Job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

`

| | |
|---|---|
| JobSet-7<br>Job 1: Ml(6); M4(6)<br>Job 2: M2(11); M4(9)<br>Job 3: M2(9); M4(7)<br>Job 4: M3(16); M4(7)<br>Job 5: Ml(9); M3(18)<br>Job 6: M2(13); M3(19); M4(6)<br>Job 7: Ml(10); M2(9); M3(13)<br>Job 8: Ml(l1); M2(9); M4(8) | JobSet-8<br>Job 1: M2(12); M3(21);M4(11)<br>Job 2: M2(12); M3(21);M4(11)<br>Job 3: M2(12); M3(21);M4(11)<br>Job 4: M2(12); M3(21);M4(11)<br>Job 5: Ml(10); M2(14);M3(18);M4(9)<br>Job 6: Ml(10);M2(14); M3(18);M4(9) |
| JobSet-9<br>Job 1: M3(9);Ml(12);M2(9);M4(6)<br>Job 2: M3(16);M2(11); M4(9)<br>Job 3: Ml(21); M2(18); M4(7)<br>Job 4: M2(20); M3(22); M4(11)<br>Job 5:M3(14);Ml(16);M2(13); M4(9) | JobSet-10<br>Job1:Ml(11);M3(19);M2(16);M4(13)<br>Job2: M2(21);M3(16); M4(14)<br>Job3:M3(8); M2(10); Ml(14); M4(9)<br>Job4: M2(13); M3(20); M4(10)<br>Job5: Ml(9); M3(16); M4(18) ;<br>Job6:M2(19);Ml(21); M3(11);M4(15) |

## V. VEHICLE SCHEDULING METHODOLOGY

Jobs are scheduled based on the operation sequence derived by the NEH heuristic algorithm. Initially AGVs carry jobs from the load/unload station to the respective workstations where the first operations are scheduled. AGVs make two types of trips, a loaded trip where it carries a load and a deadheading trip where the vehicle moves to pick up a load. Deadheading trip can start immediately after the delivery and vehicle demand at different workstations are considered and the subsequent assignments are made. If both AGVs are available task is assigned to the earliest available vehicle. If no vehicle is available, the earliest available times of the AGVs are computed and the assignment is made. If the vehicle is idle and no job is ready, assign the operation that is going to be completed early and is identified the vehicle is moved to pick up that job. This type of vehicle scheduling methodology helps in reducing the waiting times and thus helps in improving the resource utilization and the throughput. The flow chart of the vehicle assignment methodology is given in Fig.2


START
InputChromozome
Schedule the operations according to the chromosome sequence
Is the first operation of job?
Find which AGV will reach the L/U station early
Find which AGV will reach the machine with demand point early
Move the AGV from the current position to the request point for its next assignment
Check whether the job is ready?
AGV waits till the job is ready
Move the job to the machine at which its next operation is scheduled
Check whether the machine is free?
AGV drops the job at the machine buffer, job will be loaded after the machine becomes free
Load the job on the machine
Are all operations completed?
Output
Best operation sequence, Makespan

`

**Fig. 2 Flow Chart for Simultaneous scheduling of machines and AGVs**

## VI. SIMULTANEOUS SCHEDULING OF MACHINES AND AGVs THROUGH HEURISTIC ALGORITHMS

The RA Heuristic Algorithm is implemented to the simultaneous scheduling problems. The basic input utilized to study this aspect is taken from section 4.4.

### 6.1 Simultaneous Scheduling - RA Heuristic Algorithm

For implementation of RA Heuristic Algorithm, job set 1 and layout 4 are considered as an example. RA Heuristic Algorithm constructs jobs sequence in an iterative manner. The iterations are continued till all jobs from the content list are placed in the partial sequence.

The RA Heuristic is explained in the following steps for the job set 1:

Ex: **Johnson Algorithm for Jobset-1 and Layout-4**

| **Job** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Ti1** | 8 | 20 | 15 | 0 | 15 |
| **Ti2** | 16 | 18 | 0 | 18 | 0 |
| **Ti3** | 0 | 10 | 12 | 0 | 10 |
| **Ti4** | 12 | 0 | 8 | 14 | 0 |

**For job-1**

**TiA=(m*Ti1)+((m-1)*Ti2)+((m-2)*Ti3)+((m-3)*Ti4)**

Here **m=**no. of machines=4

**=** (4*8)+(3*16)+(2*0)+(1*12) = **92**

**TiB=(1*Ti1)+(2*Ti2)+(3*Ti3)+(4*Ti4)**

**=** (1*8)+(2*16)+(3*0)+(4*12) = **88**

**For job-2**

**TiA=(m*Ti1)+((m-1)*Ti2)+((m-2)*Ti3)+((m-3)*Ti4)**

`

Here **m=**no. of machines=4

**=**(4*20)+(3*18)+(2*10)+(1*0) = **154**

**TiB = (1*Ti1)+(2*Ti2)+(3*Ti3)+(4*Ti4)**

**=** (1*20)+(2*18)+(3*10)+(4*0) = **86**

**For job-3**

**TiA=(m*Ti1)+((m-1)*Ti2)+((m-2)*Ti3)+((m-3)*Ti4)**

Here **m=**no. of machines=4

**=**(4*15)+(3*0)+(2*12)+(1*8) = **92**

**TiB = (1*Ti1)+(2*Ti2)+(3*Ti3)+(4*Ti4)**

**=** (1*15)+(2*0)+(3*12)+(4*8) = **83**

**For job-4**

**TiA=(m*Ti1)+((m-1)*Ti2)+((m-2)*Ti3)+((m-3)*Ti4)**

Here **m=**no. of machines=4

**=**(4*0)+(3*18)+(2*0)+(1*14) = **68**

**TiB = (1*Ti1)+(2*Ti2)+(3*Ti3)+(4*Ti4)**

**=** (1*0)+(2*18)+(3*0)+(4*14) = **92**

**For job-5**

**TiA=(m*Ti1)+((m-1)*Ti2)+((m-2)*Ti3)+((m-3)*Ti4)**

Here **m=**no. of machines=4

**=**(4*15)+(3*0)+(2*10)+(1*0) = **80**

**TiB = (1*Ti1)+(2*Ti2)+(3*Ti3)+(4*Ti4)**

**=** (1*15)+(2*0)+(3*10)+(4*0) = **45**

`

| **Job** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Ti A** | **92** | **154** | **92** | **68** | **80** |
| **Ti B** | **88** | **86** | **83** | **92** | **45** |

Now Performing the **Johnsons Algorithm**

- Mark the Least Processing Time for each job

**TiA 92 154 92 68 80**

**TiB 88 86 83 92 45**

- Divide the jobs into two groups based on their processing time namely U,V

U group as follows (Ti A<Ti B) = 4

V group as follows (Ti A>Ti B) = 1, 2 , 3 ,5

- Sortthe jobs "U" as follows which considering least processing times and these processing times are arranged with minimum process time to maximum process time

U = 4

Process time = 68

- Sort the jobs "V" as follows with consideration of least processing time on each job and arranging these processing times from maximum processing time to minimum processing time

V= 1 2 3 5

Process time =88 86 83 45

Arrangement = 88 86 83 45

V=1 2 3 5

- According to Johnson algorithm the

**Job** : 4 1 2 3 5

- optimal sequence is 'UV'

`

10-11-1-2-3-4-5-6-7-8-9-12-13

| 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) |
| 10 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 14 | 14 | 28 |
| 11 | 2 | 2 | 0 | 4 | 0 | 28 | 14 | 28 | 40 | 0 | 40 | 18 | 58 |
| 1 | 1 | 1 | 4 | 0 | 14 | 0 | 28 | 28 | 32 | 0 | 32 | 8 | 40 |
| 2 | 2 | 1 | 1 | 1 | 32 | 40 | 32 | 40 | 44 | 58 | 58 | 16 | 74 |
| 3 | 4 | 2 | 2 | 2 | 40 | 74 | 40 | 74 | 80 | 28 | 80 | 12 | 92 |
| 4 | 1 | 1 | 2 | 0 | 44 | 0 | 64 | 64 | 68 | 40 | 68 | 20 | 88 |
| 5 | 3 | 1 | 1 | 1 | 68 | 88 | 68 | 88 | 94 | 0 | 94 | 10 | 104 |
| 6 | 2 | 2 | 4 | 3 | 80 | 104 | 86 | 104 | 110 | 74 | 110 | 18 | 128 |
| 7 | 3 | 1 | 3 | 0 | 94 | 0 | 106 | 106 | 116 | 104 | 116 | 12 | 128 |
| 8 | 4 | 1 | 3 | 3 | 116 | 128 | 116 | 128 | 134 | 92 | 134 | 8 | 142 |
| 9 | 1 | 2 | 2 | 4 | 110 | 142 | 116 | 142 | 156 | 88 | 156 | 15 | 171 |
| 12 | 3 | 1 | 4 | 0 | 134 | 0 | 148 | 148 | 158 | 128 | 158 | 10 | 168 |
| 13 | 1 | 1 | 3 | 3 | 158 | 168 | 158 | 168 | 176 | 171 | 176 | 15 | 191 |

**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 operation scheduling of through RAPID ACCESS heuristic algorithm for job set 1 layout 1. From the table it is observed that operation 10 on machine 1 is completed by 28 min. Hence 11$^{th}$ operation will start after completion of 10$^{st}$ operation on machine 1. In case of job set 1 and layout 4 operation 11 on machine 2 is completed by 58 min. Hence 1$^{st}$ operation on machine 4 will start after completion of 11$^{th}$ operation on machine 2. Similarly no

ISSN: 2005-4238 IJAST 

operation on the particular machine will start until the operation on the machine is completed.

From the vehicle heuristic algorithm for first two operations AGVs are selected randomly in case of third operation AGV '1' is selected basing on the availability of AGV with minimum travel time this constraint will be taking care in the algorithm. For job set 1 and layout 4 the operational completion time (makespan) is 191.

## VII. RESULTS AND DISCUSSION

Ten different job sets with different processing sequences, and process times are generated and presented. Different combinations of these ten job sets and four layouts are used to generate 82 example problems. In all these problems there are two vehicles. Table 3 consists of problems whose $t_i/p_i$ ratios are greater than 0.25, and whose $t_i$/pi ratios are lesser than 0.25 ratios are represented in Table 4. A code is used to designate the example problems which are given in the first column. The digits that follow 1.1 indicate the job set and the layout. In Table 4 another digit is appended to the code. Having 0 or 1 as the last digit implies that the process times are doubled or tripled where in both cases travel times are reduced to half.

**Table 3 Performance evaluation for t/p>0.25**

| Job. No | t/p | FCFS | SPT | LPT | RA |
|---|---|---|---|---|---|
| 1.1 | 0.59 | 173 | 193 | 177 | 159 |
| 2.1 | 0.61 | 158 | 158 | 177 | 172 |
| 3.1 | 0.59 | 202 | 224 | 198 | 211 |
| 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 | 194 |
| 8.1 | 0.58 | 261 | 261 | 266 | 261 |
| 9.1 | 0.61 | 270 | 277 | 268 | 263 |
| 10.1 | 0.55 | 308 | 308 | 310 | 312 |
| 1.2 | 0.47 | 143 | 173 | 165 | 141 |
| 2.2 | 0.49 | 124 | 124 | 130 | 128 |

`

| | | | | | |
|---|---|---|---|---|---|
| 3.2 | 0.47 | 162 | 188 | 160 | 175 |
| 4.2 | 0.73 | 217 | 223 | 224 | 216 |
| 5.2 | 0.68 | 118 | 144 | 131 | 112 |
| 6.2 | 0.54 | 180 | 169 | 165 | 154 |
| 7.2 | 0.62 | 149 | 160 | 149 | 144 |
| 8.2 | 0.46 | 181 | 181 | 198 | 181 |
| 9.2 | 0.49 | 250 | 249 | 244 | 239 |
| 10.2 | 0.44 | 290 | 288 | 287 | 273 |
| 1.3 | 0.52 | 145 | 175 | 167 | 143 |
| 2.3 | 0.54 | 130 | 130 | 136 | 130 |
| 3.3 | 0.51 | 160 | 190 | 162 | 173 |
| 4.3 | 0.8 | 233 | 237 | 230 | 226 |
| 5.3 | 0.74 | 120 | 146 | 133 | 114 |
| 6.3 | 0.54 | 182 | 171 | 167 | 156 |
| 7.3 | 0.68 | 155 | 166 | 151 | 150 |
| 8.3 | 0.5 | 183 | 183 | 200 | 183 |
| 9.3 | 0.53 | 252 | 251 | 246 | 241 |
| 10.3 | 0.49 | 293 | 294 | 293 | 279 |
| 1.4 | 0.74 | 189 | 207 | 189 | 191 |
| 2.4 | 0.77 | 174 | 174 | 174 | 172 |
| 3.4 | 0.74 | 220 | 250 | 212 | 225 |
| 4.4 | 1.14 | 301 | 301 | 298 | 298 |
| 5.4 | 1.06 | 171 | 189 | 171 | 171 |
| 6.4 | 0.78 | 249 | 252 | 237 | 237 |

| | | | | | |
|---|---|---|---|---|---|
| 7.4 | 0.97 | 217 | 242 | 151 | 210 |
| 8.4 | 0.72 | 285 | 285 | 200 | 285 |
| 9.4 | 0.76 | 292 | 311 | 290 | 285 |
| 10.4 | 0.69 | 350 | 350 | 345 | 348 |

**Table 4 Performance evaluation for t/p<0.25**

| Job.No | t/p | FCFS | SPT | LPT | RA |
|---|---|---|---|---|---|
| 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 |

## VIII. CONCLUSIONS

In this paper the optimal sequence of machines and AGVs are determined by using NEH heuristic algorithm. It is observed from Table 3 that 29 problems out of 40 give better results when compared with FCFS.35 problems give better results when compared with SPT (Nageswararao et al. 2017). and 33 problems give better results when compared with LPT. It is observed from Table 4 that can be observed that from table 1.4 out of 42 problem 30 problems gives better results using NEH when compared with FCFS, 42 problems gives better results when compared with SPT and 42 problems gives better results when compared with LPT.

## ACKNOWLEDGMENTS

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

1. Bilge, U., &Ulusoy, G. (1995). A time window approach to simultaneous scheduling of
2. machines and material handling system in a FMS. Operations Research, 43 (6), 1058–1070.
3. 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.
4. 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.
5. 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.
6. Dannenbring, D. G. (1977).An evaluation of flow shop sequencing heuristics. Management Science, 23(11), 1174-1182.
7. Eren, T., &Güner, E. (2008).A bicriterionflowshop scheduling with a learning effect.Applied Mathematical Modeling, 32, 1719–1733.
8. Gupta, J. N. D. (1971). A functional heuristic algorithm for the f1owshop scheduling problem. Operational Research, 22, 39-47.
9. He, Y., &Hui, C. W. (2008). A rule-based genetic algorithm for the scheduling of single-stage multi-product batch plants with parallel units. Computers and Chemical Engineering, 32, 3067–3083.
10. Johnson, S.M. (1954). Optimal two-and-three-stage production schedules with set-up times
11. included. Naval Research Logistic, 1, 61–68.
12. 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.
13. 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.
14. 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.

`

15. Mosheiov, G., &Sarig, A. (2009).Due-Date Assignment on Uniform Machines. European Journal of Operational Research, 193(1), 49-58.
16. 
17. 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.
18. 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.
19. 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.
20. 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.
21. Nowicki, E., &Smutnicki, C. (1996).A fast tabu search algorithm for the permutation flowshop problem. European Journal of Operational Research, 91(1), 160–175.
22. 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.
23. Ronconi, D. P. (2004).A note on constructive heuristics for the flowshop problem with blocking. International Journal of Production Economics, 87(1), 39–48.
24. 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.
25. 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.
26. 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.
27. Wu, X., & Zhou, X. (2008). Stochastic scheduling to minimize expected maximum lateness. European Journal of Operational Research, 190 (1), 103–115