

## Evolutionary approaches for scheduling a flexible manufacturing system with automated guided vehicles and robots

Chandramouli Anandaraman<sup>\*</sup>, ArunVikram Madurai Sankar and Ramaraj Natarajan

*Department of Production Engineering, National Institute of Technology, Tiruchirappalli – 620 015, Tamil Nadu, India*

### ARTICLE INFO

#### Article history:

Received 25 January 2012

Accepted March, 5 2012

Available online

7 March 2012

#### Keywords:

*Sheep Flock Heredity Algorithm*

*Artificial Immune System*

*AGV*

*Robots*

*FMS scheduling*

*Multi objective optimization*

### ABSTRACT

This paper addresses the scheduling of machines, an Automated Guided Vehicle (AGV) and two robots in a Flexible Manufacturing System (FMS) formed in three loop layouts, with objectives to minimize the makespan, mean flow time and mean tardiness. The scheduling optimization is carried out using Sheep Flock Heredity Algorithm (SFHA) and Artificial Immune System (AIS) algorithm. AGV is used for carrying jobs between the Load/Unload station and the machines. The robots are used for loading and unloading the jobs in the machines, and also used for transferring jobs between the machines. The algorithms are applied for test problems taken from the literature and the results obtained using the two algorithms are compared. The results indicate that SFHA performs better than AIS for this problem.

© 2012 Growing Science Ltd. All rights reserved

## 1. Introduction

FMS is a highly automated manufacturing system well suited for the simultaneous production of a wide variety of part types in low to mid volume quantities at a low cost, while maintaining a high quality of the finished products. FMS has emerged as a viable alternative to conventional manufacturing system and existing FMS implementations have already demonstrated a number of benefits in terms of cost reductions, increased utilizations, reduced work-in-process levels, etc. Proper scheduling of an FMS is necessary for the efficient utilization of machines and other material handling equipment.

In FMS scheduling, decisions that need to be made include not only sequencing of jobs on machines, but also the routing of the jobs through the system. Apart from the machines, other resources in the system such as material handling devices e.g., AGVs, robots must be considered. Effective scheduling reduces the non-productive time spent by the job in the FMS such as travelling time between stations/machines and the waiting time of the jobs.

An AGV is a material handling system that uses independently operated, self-propelled vehicles that are guided along defined pathways in the floor. The vehicles are powered by means of on-board

<sup>\*</sup> Corresponding author. Tel: +919840557585  
E-mail: acmouli89@gmail.com (C. Anandaraman)

batteries that allow operation for several hours between recharging. The definition of pathways is generally accomplished using wires embedded in the floor or reflective paint on the floor surface. Guidance is achieved by sensors on the vehicles that can follow the guide wires or paint.

Robots are material handling systems that transport parts between machines, and also load and unload parts onto and from machines and AGVs. Industrial robots perform complex tasks in the minimum possible time in order to obtain high productivity and flexibility.

## 2. Literature Review

Stecke (1983) divided the FMS operation problem into two sub problems: preproduction setup and production operation. Buzacott and Yao (1986) presented a comprehensive review of the analytical models developed for the design and control of FMS. They strongly advocate the analytical methods as giving better insight into the system performance than the simulation models. Heragu and Kusiak(1988) illustrated the basic types of FMS layouts and presented two algorithms for solving the problem. El-Maraghy and Ravi (1992) developed modern tools for the design, modeling and evaluation of FMS. A model for simultaneous scheduling of machines and material handling system in an FMS for makespan minimization is presented by Bilge and Ulusoy (1995). The problem was formulated as a non-linear integer-programming model and was addressed using the sliding time window approach. Ulusoy et al. (1997) addressed the same problem using Genetic algorithms (GA). In this approach, the chromosome represents both the operation number and AGV assignment which requires the development of special genetic operators. Khoo et al. (2000) described the development of a prototype genetic algorithm-enhanced multi objective scheduler for a manufacturing system. Ronald and Uzsoy(2001) contributed to the development of heuristics, a major area within the field of operations research. Unlike exact algorithms, where time efficiency is the main measure of success, there are two burning issues in evaluating heuristics: how fast solutions can be obtained and how close do they come to being optimal.

Engin and Doyen (2004) explained a new approach to solve hybrid flowshop scheduling problems using AIS. Abdelmaguid et al. (2004) presented a new hybrid genetic algorithm for the simultaneous scheduling problem for minimization of makespan. The hybrid GA is composed of GA and heuristic. The GA is used to address the first part of the problem that is theoretically similar to the job shop scheduling problem and the vehicle assignment is handled by a heuristic called Vehicle Assignment Algorithm (VAA). Lacomme et al. (2005) addressed the simultaneous job input sequencing and vehicle dispatching for a single AGV system. They solved the problem using a heuristic branch-and-bound approach coupled with a discrete event simulation model. Gobal and Kasilingam (1991) developed a simulation model to determine the number of AGVs needed to meet the material handling requirements. That calculation was done based on idle time of the machines and the vehicles, and the waiting time of parts. Reddy and Rao(2006) addressed the simultaneous scheduling problem as a multi objective problem in scheduling with conflicting objectives which are more complex and combinatorial in nature. They solved the problem by using evolutionary algorithm and non-dominated sorting. Srinivas and Deb (1994) explained that many traditional methods scalarize the objective vector into a single objective in multi objective optimization problems.

Kats and Levner(1998) addressed cyclic scheduling of operations in an FMS handled by a single robot. They focused on periodic scheduling of parts in a robotic production system, functioning under a repetitive robot's route. Crama et al (2000) developed cyclic scheduling in robotic flowshops. They discussed the cyclic scheduling problems by using robots as material handling devices. Zacharia and Aspragathos (2005) dealt with optimal robot task scheduling based on GAs. Abdelmaguid and Nassef (2009) applied the scheduling of multiple load material handling equipment to the traditional job shop scheduling problem. Udhayakumar and Kumanan(2010) addressed the scheduling of job and tool in an FMS using Ant Colony Optimization (ACO) algorithm. Satish Kumar et al. (2011) addressed simultaneous scheduling of both machines and vehicles with alternative machines for the makespan

minimization objective. So far in the literature, the scheduling of material handling devices such as AGV and robots in FMS simultaneously with machines has not been considered. Also, the robots have been used only for loading and unloading of the jobs. The present work focuses on the scheduling of robots that can serve two machines and an AGV. Also, the robots considered here are multifunctional, i.e., they are used for loading/unloading as well as transfer of jobs between machines.

### 3. Problem Definition

Three loop layouts each consisting of four machines, two identical robots and one AGV have been generated. The layouts are taken as inputs for the scheduling of machines, AGV and the robots. In majority of literature, AGVs are the only material handling devices, while the loading & unloading of jobs is assumed to be done manually. In this work, one AGV is used for material handling, while two robots, each situated between two machines are used for both material handling and for loading/unloading of jobs. This is done to improve the flexibility of the FMS. Here, three different layouts and ten job sets each consisting of one to eight different jobs are considered. The objective is to minimize makespan, mean flow time and mean tardiness.

#### 3.1 FMS Description

The FMS layouts for this problem have been generated as loop layouts. A loop layout is a layout in which the machines in an FMS are arranged in a close ring-like network and the materials are transported around this network in only one direction. Minimal material handling and flexibility of access make the loop layout an attractive choice for FMS.

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

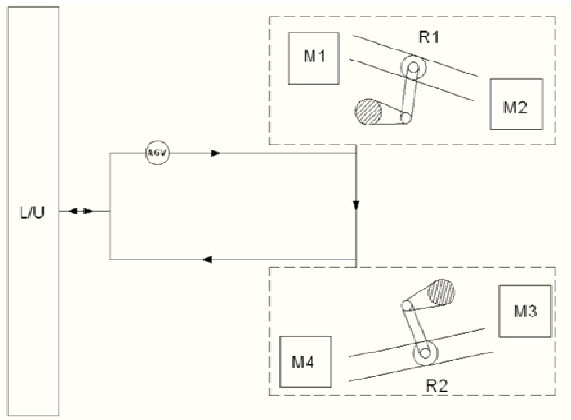
- (i) The types and number of machines are known. Operations are non-preemptive. There is sufficient input/output buffer space at each machine.
- (ii) Processing, set-up, loading and unloading times are available and are deterministic.
- (iii) Number of AGV(s) is known.
- (iv) Flow path layout is given and travel times on each segment of the path are known.
- (v) A Load/Unload (L/U) station serves as a distribution centre for parts not yet processed and as a collection centre for parts finished. All vehicles start from L/U station initially and return to thereafter accomplishing all their assignments. There is sufficient input/output buffer space at the (L/U) station.
- (vi) AGVs carry a single unit-load at a time. They move along predetermined shortest paths. Pre-emption of trips is not allowed. The trips are called loaded or deadheading (empty) trips depending on whether a part is carded or no part is carded during that trip, respectively. The durations for the deadheading trips are sequence dependent and are not known until the vehicle route is specified.
- (vii) Robots can handle only single unit-load at a time. They move along specially constructed tracks between the machines. Two kinds of motion are possible for the robots. They can move along the tracks to carry a job from one machine to another. Once the robot reaches a machine, it can transfer the job from the machine to AGV or vice versa or to itself using a combination of translatory and rotary motions. The duration of robot travel from one machine to other and the time taken for robot to transfer a job from/to machine and AGV are known.
- (viii) 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 and 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.

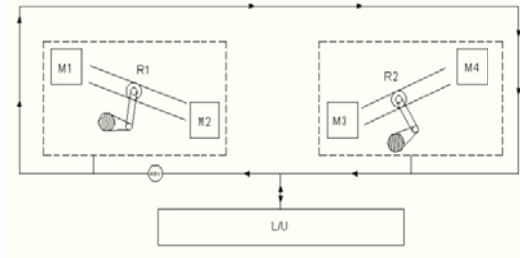
The set of part types to be produced during the planning period and the routing of each part type are available before making scheduling decisions. In other words, routing flexibility is not considered. The routing for a part type can be selected based on considerations of technological feasibility and processing efficiency, or by formulating the set-up phase problems in a manner that can also handle the routing problems.

- (ix) 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 only the remaining operations.
- (x) Issues of machine failure or downtime, scraps, rework, robot, AGV repair and maintenance are ignored and left as issues to be considered during real-time control.

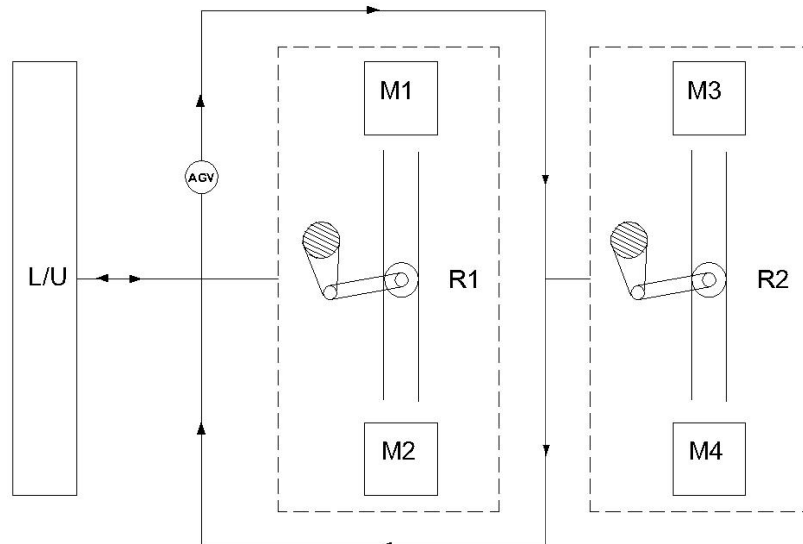
The loop layouts generated for this problem are given in Fig. 1, Fig. 2 and Fig. 3.



**Fig. 1.** Loop layout 1



**Fig. 2.** Loop layout 2



**Fig. 3.** Loop layout 3

### 3.2 Assumptions

The following assumptions are made:

- (i) All robots are identical in shape, configuration and capacity.
- (ii) All robots are capable of handling all the jobs in the job sets taken.
- (iii) Robot R1 caters to machines M1 and M2. Robot R2 caters to machines M3 and M4.
- (iv) The jobs are transferred through AGV when a direct transfer through robot is not possible.
- (v) The robot transfers the jobs to/from AGV at a common point for M1 and M2 as well as for M3 and M4.
- (vi) Transfer times between machines are known.
- (vii) There is no interference between AGV and robots. They operate independently of each other.
- (viii) The number of AGVs used is one and the number of robots is two.
- (ix) The speed of AGV is 20 m/min.
- (x) The speed of the robots is 20 m/min.
- (xi) The distances between the machines and distances between loading- unloading station and machines are known.

### 3.3 Input Data

The job sets for this problem are taken from Bilge and Ulusoy (1995). The time taken for a job to be transferred from one station to another through AGV and/or robot is expressed in the transfer time matrices given in Tables 1-3. Data for the job sets used in the example problems are given in Tables 4-6. Each of the ten job sets contains four to eight different jobs to be processed on three to five machines. The number after the letter 'M' denotes the machine number and the number within the parentheses is the processing time of the job on that machine.

**Table 1**

Transfer time matrix for Layout 1

		Destination				
		L/U	M1	M2	M3	M4
Source	L/U	0	6	6	8	8
	M1	8	0	2	5	5
	M2	8	2	0	5	5
	M3	6	10	10	0	2
	M4	6	10	10	2	0

**Table 2**

Transfer time matrix for Layout 2

		Destination				
		L/U	M1	M2	M3	M4
Source	L/U	0	3	3	13	13
	M1	13	0	2	13	13
	M2	13	2	0	13	13
	M3	3	5	5	0	2
	M4	3	5	5	2	0

**Table 3**

Transfer time matrix for Layout 3

		Destination				
		L/U	M1	M2	M3	M4
Source	L/U	0	3	3	7	7
	M1	10	0	2	7	7
	M2	10	2	0	7	7
	M3	7	7	7	0	2
	M4	7	7	7	2	0

**Table 4**

Data for Job sets 1 to 6

Job Set 1					
Job 1	M1(8)	M2(16)	M4(12)		
Job 2	M1(20)	M3(10)	M2(18)		
Job 3	M3(12)	M4(8)	M1(5)		
Job 4	M4(14)	M2(18)			
Job 5	M3(10)	M1(15)			
Job Set 2					
Job 1	M1(10)	M4(18)			
Job 2	M2(10)	M4(18)			
Job 3	M1(10)	M3(20)			
Job 4	M2(10)	M3(15)	M4(12)		
Job 5	M1(10)	M2(15)	M4(12)		
Job 6	M1(10)	M2(15)	M3(12)		
Job Set 3					
Job 1	M1(16)	M3(15)			
Job 2	M2(18)	M4(15)			
Job 3	M1(20)	M2(10)			
Job 4	M3(15)	M4(10)			
Job 5	M1(8)	M2(10)	M3(15)	M4(17)	
Job 6	M2(10)	M3(15)	M4(8)	M1(15)	
Job Set 4					
Job 1	M4(11)	M1(10)	M2(7)		
Job 2	M3(12)	M2(10)	M4(8)		
Job 3	M2(7)	M3(10)	M1(9)	M3(8)	
Job 4	M2(7)	M4(8)	M1(12)	M2(6)	
Job 5	M1(9)	M2(7)	M4(8)	M2(10)	M3(8)
Job Set 5					
Job 1	M1(6)	M2(12)	M4(9)		
Job 2	M1(18)	M3(6)	M2(15)		
Job 3	M3(9)	M4(3)	M1(12)		
Job 4	M4(6)	M2(15)			
Job 5	M3(3)	M1(9)			
Job Set 6					
Job 1	M1(9)	M2(11)	M4(7)		
Job 2	M1(19)	M2(20)	M4(13)		
Job 3	M2(14)	M3(20)	M4(9)		
Job 4	M2(14)	M3(20)	M4(9)		
Job 5	M1(11)	M3(16)	M4(8)		

**Table 5**

Data for Job sets 7 to 9

Job Set 7					
Job 1	M1(6)	M4(6)			
Job 2	M2(11)	M4(9)			
Job 3	M2(9)	M4(7)			
Job 4	M3(16)	M4(7)			
Job 5	M1(9)	M3(18)			
Job 6	M2(13)	M3(19)	M4(6)		
Job 7	M1(10)	M2(9)	M3(13)		
Job 8	M1(11)	M2(9)	M4(8)		
Job Set 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	M1(10)	M2(14)	M3(18)	M4(9)	
Job 6	M1(10)	M2(14)	M3(18)	M4(9)	
Job Set 9					
Job 1	M3(9)	M1(12)	M2(9)	M4(6)	
Job 2	M3(16)	M2(11)	M4(9)		
Job 3	M1(21)	M2(18)	M4(7)		
Job 4	M2(20)	M3(22)	M4(11)		
Job 5	M3(14)	M1(16)	M2(13)	M4(9)	

**Table 6**  
Data for Job set 10

Job Set 10				
Job 1	M1(11)	M3(19)	M2(16)	M4(13)
Job 2	M2(21)	M3(16)	M4(14)	
Job 3	M3(8)	M2(10)	M1(14)	M4(9)
Job 4	M2(13)	M3(20)	M4(10)	
Job 5	M1(9)	M3(16)	M4(18)	
Job 6	M2(19)	M1(21)	M3(11)	M4(15)

### 3.4 Objectives of Scheduling

For the scheduling of machines, AGV and robots, the following objectives are considered namely makespan, mean flow time and mean tardiness. These are important parameters which characterize the performance of a manufacturing system. Makespan indicates the total time taken by the set of jobs for the completion of all the operations. Mean flow time is a measure of the total productive time spent by the jobs in the system, i.e. the time taken for processing. Tardiness is a quantity for measuring the lateness of the jobs, against pre-determined due times for completion of each job. The equations used for computing the job completion time, makespan, mean flow time and mean tardiness are given in Eqs. (1) - (6).

$$O_{ij} = T_{ij} + P_{ij} \quad (1)$$

$$\text{Job completion time} = C_i = \sum_{j=1}^n O_{ij} \quad (2)$$

$$\text{Makespan} = \text{Max} (C_1, C_2, C_3, \dots) \quad (3)$$

$$\text{Mean flow time} = \frac{1}{n} \sum_{i=1}^n C_i \quad (4)$$

$$\text{Tardiness} = Td_i = \text{Max} ((C_i - D_i), 0) \quad (5)$$

$$\text{Mean tardiness} = \frac{1}{n} \sum_{i=1}^n Td_i \quad (6)$$

where

$i$  : job number

$j$  : operation number

$n$  : number of jobs

$O_{ij}$  : Time taken for  $j^{\text{th}}$  operation of  $i^{\text{th}}$  job

$T_{ij}$  : Total travelling time for  $i^{\text{th}}$  job before  $j^{\text{th}}$  operation

$P_{ij}$  : Total processing time for  $i^{\text{th}}$  job and  $j^{\text{th}}$  operation

$D_i$  : Due time of  $i^{\text{th}}$  job

### 3.5 Scheduling of AGV and Robots

Machines are scheduled based on the operation sequence derived by the Sheep Flock Heredity Algorithm and Artificial Immune System algorithm. Initially AGV carries jobs from the L/U station to the respective workstations where the first operations are scheduled. Transfer of the job for the next operation takes place through either AGV or robot. If the machine where the next operation is scheduled is adjacent to the machine where the previous operation took place (as shown in the layouts Fig. 1, Fig. 2 and Fig. 3) the job is transferred by the robot alone. In all other cases, the AGV reaches the machine where the previous operation has been completed, picks up the job and carries it to the machine scheduled for the next operation, and the robot transfers the job from AGV to the machine. The robots are used for both loading/unloading of jobs and the transfer of jobs between machines, thus making it a fully automated manufacturing system. Also, while scheduling, it is checked whether the machine for the next operation and the AGV or robot required for transferring the job are available. For

example, Job set1 and Layout 1 are considered for scheduling. For scheduling the FMS and calculating the makespan, mean flow time and mean tardiness, initially continuous numbers are given for each of the operations as shown in Table 7. These numbers are used to generate the initial random sequences, while obeying the precedence relation as shown in Table 3.4, i.e., the operations of a particular job must be in increasing order.

**Table 7**

Numbering of operations in Job Set 1

Job No.	Job 1			Job 2			Job 3			Job 4		Job 5	
Machine	M1	M2	M4	M1	M3	M2	M3	M4	M1	M4	M2	M3	M1
Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13

### 3.5.1. Calculation of makespan, mean flow time and mean tardiness for a particular operation sequence

The following initial sequence is generated randomly, using the numbers assigned to the operations in Table 3.4.

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

The transfer times and machining times for the operations are taken from Tables 1-3 and 4-6 respectively. The makespan of the Job set when scheduled according to the above sequence is calculated as follows:

#### 7<sup>th</sup> Job

Time to transfer job from L/U to m/c 3 + Processing time =  $8+12 = 20$

#### 4<sup>th</sup> Job

Time travelled by AGV prior to job pick up + Time to transfer job from L/U to m/c 1 + Processing time =  $11+6+20 = 37$

#### 1<sup>st</sup> Job

Time travelled by AGV prior to job pick up + Time to transfer job from L/U to m/c 1 + Processing time =  $22+6+8 = 36$

#### 12<sup>th</sup> Job

Time travelled by AGV prior to job pick up + Time to transfer job from L/U to m/c 3 + Processing time =  $33+8+10 = 51$

#### 8<sup>th</sup> Job

Time of completion of previous operation for robot 2 + Transfer time from m/c 3 to m/c 4 + Processing time =  $41+2+8 = 51$

#### 9<sup>th</sup> Job

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 4 to m/c 1 + Processing time =  $51+10+5 = 66$

#### 5<sup>th</sup> Job

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 1 to m/c 3 + Processing time =  $61+5+10 = 76$

#### 2<sup>nd</sup> Job

Time of completion of previous operation for robot 1+ Time to transfer job from m/c 1 to m/c 2 + Processing time =  $64+2+16 = 82$



**13<sup>th</sup> Job**

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 3 to m/c 1 + Processing time =  $67.5+10+15 = 92.5$

**10<sup>th</sup> Job**

Time travelled by AGV prior to job pick up + Time to transfer job from L/U to m/c 3 + Processing time =  $81+8+14 = 103$

**3<sup>rd</sup> Job**

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 2 to m/c 4 + Processing time =  $93+5+12 = 110$

**11<sup>th</sup> Job**

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 4 to m/c 2 + Processing time =  $103+10+18 = 131$

**6<sup>th</sup> Job**

Time travelled by AGV prior to job pick up + Time to transfer job from m/c 3 to m/c 2 + Processing time =  $112+10+18 = 140$ .

Makespan = 140

Mean flow time =  $(20+37+36+51+51+66+76+82+92.5+103+110+131+140)/13 = 76.57$

Mean Tardiness = Tardiness/No. of Jobs

Calculation of Tardiness for the Job Set 1 is shown in Table 8.

**Table 8**

Calculation of Tardiness of jobs

Job	1	2	3	4	5
Due time	42	84	126	168	210
Completion time	110	140	66	131	92.5
Tardiness = Max((Completion time - Due time), 0)	68	56	0	0	0

Total tardiness =  $68+56+0+0+0 = 124$

Mean tardiness =  $124 / 13 = 24.8$

**4. Proposed Algorithms**

The non-traditional approaches used for optimization are Sheep Flock Heredity Algorithm (SFHA) (Nara & Kim, 1996; Nara et al., 1999) and Artificial Immune System (AIS) algorithm (De Castro & Von Zuhlen, 1999; Huang, 1999; Costa et al., 2002; Attux, 2003; Zheng et al., 2004). A non-dominated sorting technique is included in both the optimization algorithms to compare the solutions with respect to multiple objectives. Every solution in the population is compared with the other solutions in the population for all the objectives and a solution is declared as a non-dominated solution if it is superior in at least one objective over the other solutions. The fitness of a solution is assigned based on the number of solutions it dominates.

**4.1 Sheep Flock Heredity Algorithm**

Normally, sheep in each flock are living within their own flock under the control of shepherds. So, the genetic inheritance only occurs within the flock. Some special characteristics in one flock develop only within the flock by heredity, and the sheep with high fitness characteristics to their environment breed in the flock. Assume there are two flocks occasionally mixed with the other flocks. The characteristics of the sheep in neighbouring flocks can be inherent to the sheep in this flock. In the field, the flock of the sheep, which has better characteristics to the field environment, breeds most. The natural evolution

phenomenon of flocks can be corresponded to the genetic operations of this type of string. For this kind of string, we can define two kinds of operations. (i) Normal genetic operations between strings. (ii) Genetic operations between sub-strings within one string. In SFHA, special string structure and hierarchical genetic operations (crossover and mutation) are introduced. They are (i) sub-chromosome level genetic operation and (ii) chromosome (global) level genetic operation. This hierarchical operation is referred to as “multi-stage-genetic operation”.

#### 4.2 Flow Chart of Sheep Flock Heredity Algorithm

Fig.4 shows the flow chart of the steps involved in the implementation of Sheep Flock Heredity Algorithm.

#### 4.3 Steps in Sheep Flock Heredity Algorithm

Begin

    Initialize the population.

##### **Stage 1:**

Select the parent

Sub chromosome level crossover

Set sub chromosome level crossover probability

    If population is less than or equal to sub chromosome level probability

        Perform sub chromosome level crossover

    Else retain the old sequences

Sub chromosome level mutation

Set sub chromosome mutation probability

    If population probability is less than or equal to sub chromosome mutation probability

        Perform sub chromosome level mutation

    Else retain the same sequences

##### **Stage 2:**

Select the best sequence from population

Chromosome level crossover

Set crossover probability

    If population probability is less than or equal to crossover probability

        Perform chromosome level crossover

Else retain the same sequences

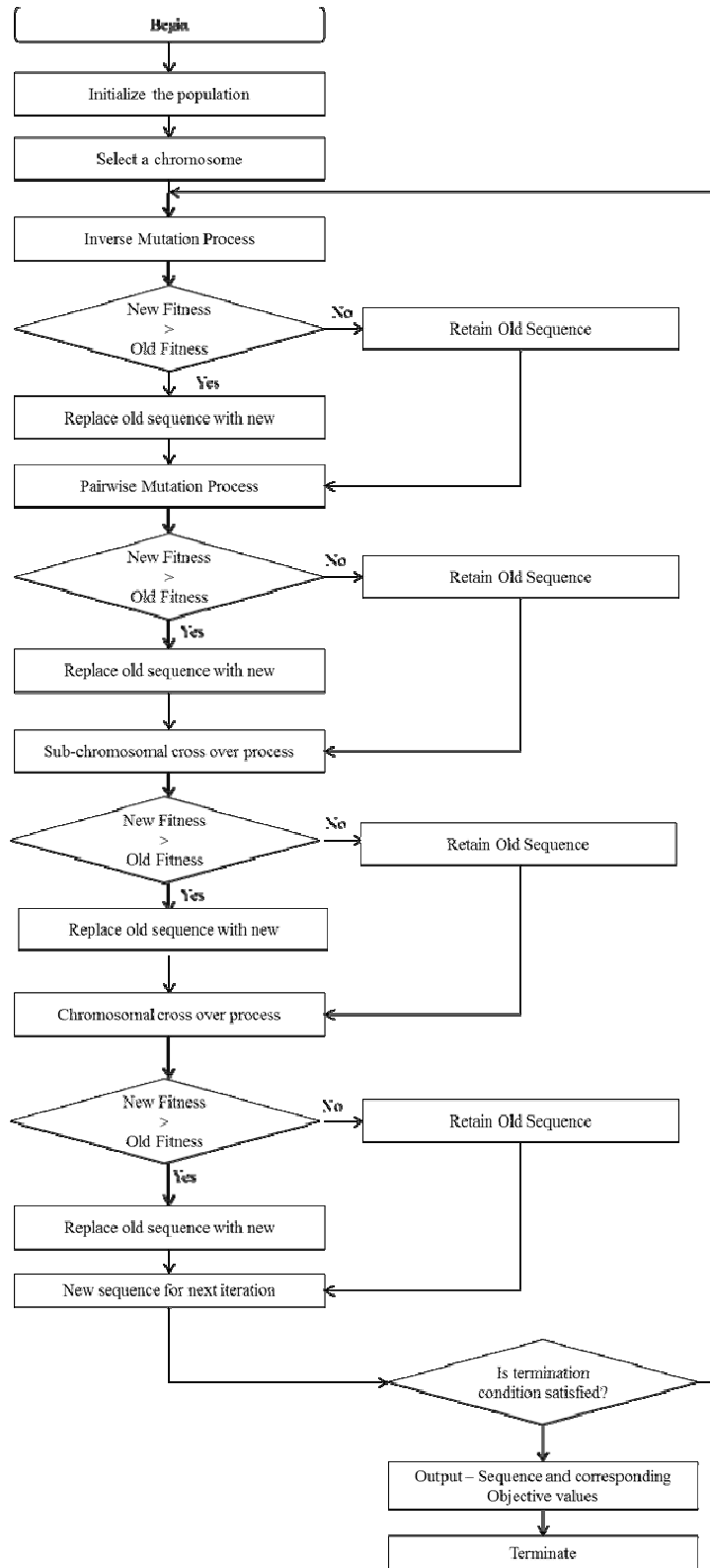
Chromosome level mutation

    If population probability is less than or equal to mutation probability

        Perform chromosome level mutation

    Else retain the same sequences

End if terminal condition satisfied



**Fig. 4.** Flow chart for SFHA

#### 4.3 Artificial Immune System Algorithm

AIS is built on the following principles of the immune system.

- a) Clonal selection principle
- b) Affinity maturation principle

#### a) Clonal Selection Principle

The clonal selection principle is the algorithm used by the immune system to describe the basis feature of an immune response to an antigenic stimulus. Each schedule is assigned a fitness value, which is taken as its affinity. The cloning of antibodies is done directly proportional to their fitness function values. Therefore, there will be more clones of antibodies that have higher fitness values than those of with lower fitness values in the new generated clone population.

#### b) Affinity Maturation Principle

It consists of two methods namely Mutation and Receptor editing.

##### 1. Mutation:

A two phased mutation procedure were used for the generated clones.

##### i) Inverse Mutation:

For a sequence  $s$ , let  $i$  and  $j$  be randomly selected two positions in the sequences. A neighbor of  $s$  is obtained by inverting the sequence of jobs between  $i$  and  $j$  positions. If the fitness value of the mutated sequence is higher than that of the original sequence, then the mutated one is stored in the place of the original one. Otherwise, the sequence will be mutated again with random pair wise interchange mutation.

##### ii) Pairwise interchange mutation:

Given a sequence  $s$ , let  $i$  and  $j$  be randomly selected two positions in the sequence  $s$ . A neighbor of  $s$  is obtained by interchanging the jobs in positions  $i$  and  $j$ . If the fitness value of the mutated sequence is higher than that of the original sequence, then replace the original one with the mutated one. In the case where the algorithm could not find a better sequence after the two-mutation procedure, then it stores the original sequence.

##### 2. Receptor Editing

After cloning and mutation processes, a percentage of the antibodies, in the antibody population are eliminated and randomly created antibodies are replaced with them. This mechanism allows finding new schedules that correspond to new search regions in the total search space.

#### 4.5 Flow Chart of Artificial Immune System Algorithm

Fig.5 shows the flow chart of the steps involved in the implementation of AIS algorithm.

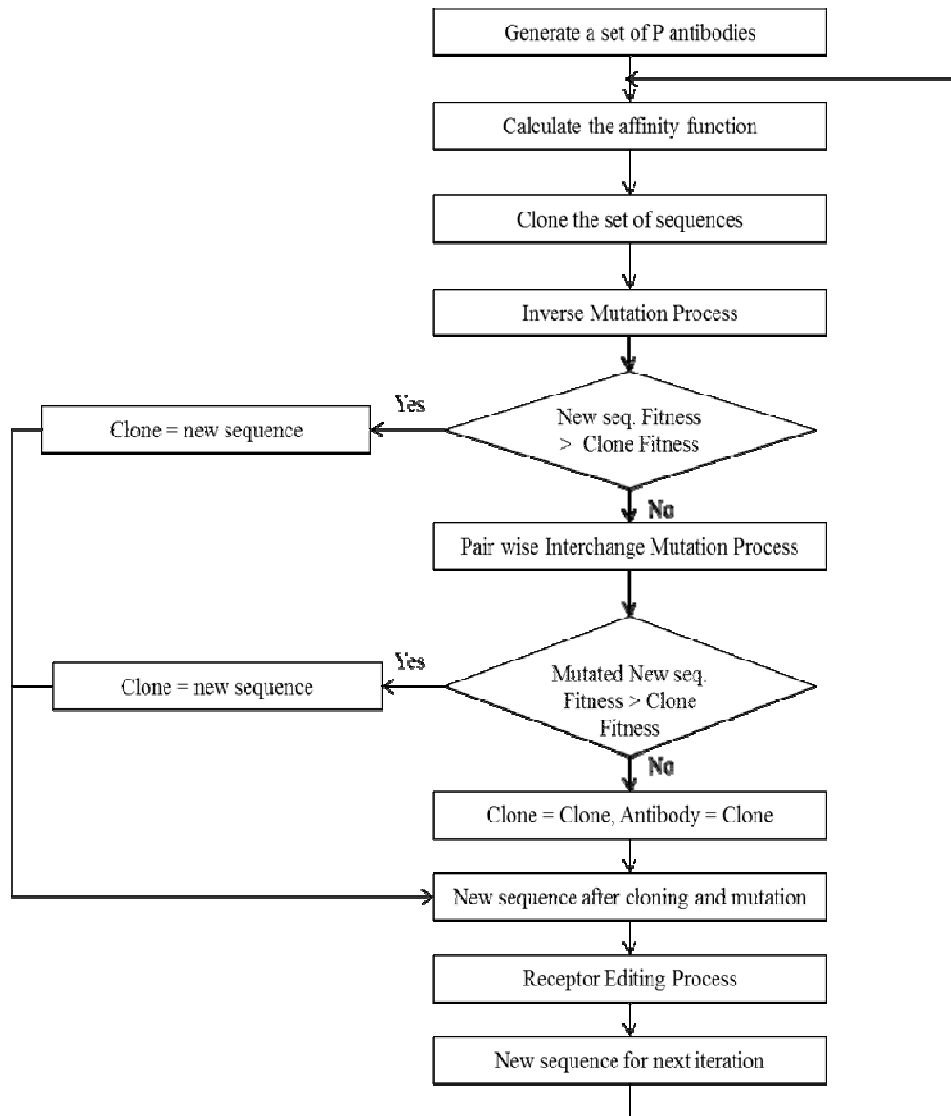
#### 4.6 Steps in Artificial Immune System Algorithm

- (i) Randomly choose the antibodies within some ranges and present it to antigen.
- (ii) Population loop for each antigen, do:
  - a. Determine the affinity values of all antibodies.
  - b. Select the ' $n$ ' highest affinity antibodies from the list whose affinity is greater than the average affinity of all antibodies.
  - c. The ' $n$ ' selected antibodies will be cloned independently and proportionally to their antigenic affinities, to generate a new antibody population of the same size.
  - d. Clone the generated antibodies.
  - e. Evaluate the affinity of matured clones.
  - f. Replace the low affinity antibodies with newly created antibodies.

(iii) Cycle: repeat Step (ii) until a given convergence criterion is met.

(iv) Reselect the highest affinity antibodies.

End.



**Fig. 5.** Flow chart for AIS algorithm

## 5. Implementation of algorithms

### 5.1 Implementation of SFHA for scheduling of AGV and Robots in FMS

The Sheep Flock Heredity Algorithm is based on the following operations

- i. Sub Chromosomal Crossover
- ii. First Stage Mutation
- iii. Chromosomal Crossover
- iv. Second Stage Mutation

For an instance of the implementation of Sheep flock heredity algorithm, Job set 1 and Layout 1 is considered.

### 5.1.1 Sub Chromosomal Crossover

The initial chromosomes are modified by sub chromosomal crossover. The entire string length is split into sub chromosomes of equal length and these sub chromosomes are moved randomly to form a new chromosome. If the fitness value corresponding to this new chromosome is higher than the fitness value for the parent chromosome, the parent chromosome is replaced by the new chromosome. Otherwise, the parent chromosome is retained.

Parent Chromosome

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

Sub chromosome length = 3

After Sub Chromosomal Crossover

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

After repair

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

### 5.1.2 First Stage Mutation

a) Inverse mutation

In a sequence, two positions  $i$  and  $j$  are randomly selected. The portion of the sequence between these two positions is inverted to get a new mutated sequence. The new sequence represents the sequence of operations after mutation. If the fitness value of the mutated sequence is higher than the fitness of the original sequence, the old sequence is replaced by the new sequence.

Mutated sequence

4	<b>6</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>13</b>	<b>11</b>	<b>2</b>	<b>10</b>	<b>12</b>	<b>1</b>	<b>5</b>	3
---	----------	----------	----------	----------	-----------	-----------	----------	-----------	-----------	----------	----------	---

Mutation between positions 2 and 12

After repair

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

b) Pair wise interchange mutation

Two positions are  $i$  and  $j$  randomly selected in the sequence. The operations in these positions are simply interchanged to obtain the mutated sequence. The fitness of the new sequence is compared with the fitness of the parent sequence. The sequence having the higher fitness value is stored and used for next stage operation.

Mutated sequence

4	<b>5</b>	7	8	9	12	10	1	<b>11</b>	13	2	6	3
---	----------	---	---	---	----	----	---	-----------	----	---	---	---

Mutation at positions 2 and 9

After repair

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

### 5.1.3 Chromosomal crossover

The chromosome obtained from the previous mutation process is crossed over again by chromosomal crossover. In this, the chromosome having the best fitness is crossed randomly to generate a new chromosome.

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

After crossover

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

After repair

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

### 5.1.4 Second stage mutation

The chromosome is mutated again using inverse and pair wise interchange mutations.

Inverse mutation

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

After mutations between positions 7 and 13 and repair

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

Pair wise interchange mutation

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

After mutation at positions 2 and 11 and repair

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

The chromosome obtained after the above operations is taken and the corresponding objective values are taken as the objectives for that generation. The chromosome is again subjected to the operations till the required number of generations after which the convergence criterion is met.

## 5.2 Implementation of Artificial Immune System Algorithm for Scheduling in FMS

For an instance of implementation of AIS, Job set 1 and Layout 1 is considered. Ten initial sequences are randomly generated, given in Table 9.

**Table 9**  
Initial sequences

Sequence													Fitness value (Affinity)
1	2	7	4	12	3	8	5	10	6	11	9	13	1
7	1	10	12	8	4	2	13	3	11	5	9	6	5
7	1	10	2	11	12	4	8	3	13	5	9	6	0
4	5	10	11	12	1	2	7	8	9	13	6	3	1
7	4	1	12	8	9	2	10	5	13	6	11	3	4
4	7	8	5	10	6	11	1	2	9	12	3	13	1
10	7	12	1	2	4	13	8	5	3	11	9	6	0
4	5	7	6	8	1	10	2	9	3	11	12	13	9
7	1	4	12	2	8	10	9	5	13	6	11	3	1
4	5	1	6	12	13	7	10	11	2	3	8	9	0

The affinity values for the sequences are calculated using non dominated sorting.

### 5.2.1 Clonal Selection

The average value of affinity for the set of ten sequences is calculated. The sequences having affinity values above the average are selected for cloning, as shown in Table 10.

Average affinity = 2.2

**Table 10**

Sequences selected for Cloning

Sequence													Fitness value
1	2	7	4	12	3	8	5	10	6	11	9	13	1
7	1	10	12	8	4	2	13	3	11	5	9	6	5
4	5	10	11	12	1	2	7	8	9	13	6	3	1
7	4	1	12	8	9	2	10	5	13	6	11	3	4
4	7	8	5	10	6	11	1	2	9	12	3	13	1
4	5	7	6	8	1	10	2	9	3	11	12	13	9
7	1	4	12	2	8	10	9	5	13	6	11	3	1

The cloning process is nothing but generating a copy of the sequence. The number of clones generated for a particular sequence is proportional to its affinity. The sequences with higher affinity will have more clones than the sequences with lower affinity. The cloned sequences are shown in Table 11.

**Table 11**

Cloned sequences

Sequence													Fitness value
1	2	7	4	12	3	8	5	10	6	11	9	13	1
7	1	10	12	8	4	2	13	3	11	5	9	6	5
4	5	10	11	12	1	2	7	8	9	13	6	3	1
7	4	1	12	8	9	2	10	5	13	6	11	3	4
4	7	8	5	10	6	11	1	2	9	12	3	13	1
4	5	7	6	8	1	10	2	9	3	11	12	13	9
7	1	4	12	2	8	10	9	5	13	6	11	3	1
4	5	7	6	8	1	10	2	9	3	11	12	13	9
7	1	10	12	8	4	2	13	3	11	5	9	6	5
7	4	1	12	8	9	2	10	5	13	6	11	3	4

### 5.2.2 Mutation

The following mutation operations are carried out on each sequence. At the end of each operation, if the fitness of the new sequence is higher than the fitness of the original one, the original is replaced with the mutated sequence. Otherwise, the original sequence is retained.

1. Inverse mutation
2. Pair wise interchange mutation

### 5.2.3 Receptor editing

After the cloning and mutations, a proportion of the sequences are eliminated and are replaced by new sequences generated randomly. This procedure allows for finding new schedules that are in new search regions in the total search space.

## 6. Results and Discussion

Schedules for each problem are generated using SFHA and AIS algorithm. The coding for the optimization of scheduling has been developed using MATLAB. The generation of example problems



is based to a large extent on Bilge and Ulusoy (1995). Ten different job sets with different processing sequences and process times are generated and presented in Tables 4-6. Different combinations of these job sets and three layouts are used to generate 90 example problems. A series of trial experiments are used to estimate the optimal parameters for the optimization algorithms. For SFHA, the probabilities of sub chromosomal level crossover, mutation and chromosomal level cross over to take place are found to be 0.01, 0.05 and 0.001 respectively. The optimal population size is found to be 5. For AIS, the optimal probability of mutation and the population size is found to be 0.01 and 10 respectively. The number of low affinity antibodies to be replaced is fixed as 4. Every instance of a problem is executed for 10 runs using both the algorithms. Any alteration of the above parameters led to convergence at higher objective values than while using the optimal parameters. Table 12 shows some of the Pareto front solutions obtained. Solutions are seen to converge within 70-75 generations while using SFHA and within 170-180 generations in the case of AIS.

**Table 12**  
Pareto Optimal Solutions

Problem Number	Makespan	Mean Flow time	Mean tardiness
EX11	146.5	103.92	11.01
	148	102.76	9.34
	140	76.57	10.04
EX81	198	97.93	10.9
	196	98.41	4.25
	194	96.27	8.14
	195	97.36	3.77
EX12	181.5	102.28	13.92
	185	103.01	15.10
	190	101.31	12.19
	183	103.27	17.09
EX82	218	92.71	8.35
	215	99.55	9.01
	220	100.12	10.09
	219	91.40	7.42
	216	96.87	7.88
EX13	211.5	114.06	16.33
	214	115.53	17.31
	218	118.71	18.92
	219	115.15	14.56
EX83	239	97.52	5.44
	241	93.90	5.69
	243	98.04	4.91

The best makespan values taken from the Pareto optimal sets and the computational times taken by the two algorithms for each problem are given in Tables 13-15. A code is used to designate the problems which are given in the first column of Tables 13-15. The digits that follow EX indicate the job set and layout. The results obtained by doubling and tripling the processing times are given in Table 14 and Table 15 respectively, where in both cases the travel times have been halved. The above results have been graphically compared in Fig. 6a, Fig. 6b and Fig. 6c. From Table 13 it is observed that for 26 problems out of 30, SFHA gives a better makespan when compared with AIS. AIS gives a better makespan when compared to SFHA for 4 problems. From Table 14 it is observed that for all 30 problems, SFHA gives a better makespan when compared with AIS. From Table 15 it is observed that for 28 problems out of 30, SFHA gives a better makespan when compared with AIS. AIS gives a better makespan when compared to SFHA for 2 problems.

**Table 13**Makespan values for SFHA and AIS for  $t/p > 0.25$ 

Problem No.	SFHA	Computational Time (s)	AIS	Computational Time (s)
EX11	140.0	23.86	181.5	31.49
EX21	164.0	17.82	174.0	20.63
EX31	157.0	17.60	165.0	23.53
EX41	197.5	8.72	212.5	28.41
EX51	160.5	8.60	171.5	21.00
EX61	154.5	21.47	152.5	30.40
EX71	202.0	22.44	205.0	26.53
EX81	194.0	15.27	209.0	14.19
EX91	188.0	16.10	201.0	18.87
EX101	232.0	21.02	240.0	20.40
EX12	181.5	5.77	184.5	15.05
EX22	174.0	12.53	179.0	28.75
EX32	182.0	11.38	178.0	20.92
EX42	233.0	9.85	243.0	17.00
EX52	175.5	13.87	179.5	26.33
EX62	161.5	22.87	169.5	34.26
EX72	237.0	22.29	231.0	22.76
EX82	215.0	13.33	225.0	26.61
EX92	204.0	17.19	218.0	18.36
EX102	255.0	14.39	264.0	22.89
EX13	211.5	13.43	220.5	17.77
EX23	198.0	18.85	208.0	16.15
EX33	193.0	8.69	205.0	22.24
EX43	264.5	6.49	259.5	20.87
EX53	205.5	22.08	210.5	24.28
EX63	177.5	6.72	182.0	20.77
EX73	250.0	5.01	253.0	26.71
EX83	239.0	15.93	244.5	17.71
EX93	234.0	17.66	240.5	33.37
EX103	284.0	16.86	298.5	15.28

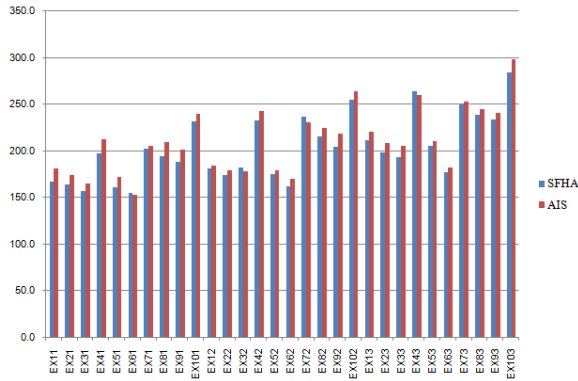
**Table 14**Makespan values for SFHA and AIS for  $t/p < 0.25$ 

Problem No.	SFHA	Computational Time (s)	AIS	Computational Time (s)
EX110	127.5	13.38	136.5	22.19
EX210	127.0	24.71	133.0	31.41
EX310	133.5	14.55	147.5	33.89
EX410	143.5	11.74	146.0	17.09
EX510	109.0	15.03	120.0	29.56
EX610	128.0	11.78	136.5	28.70
EX710	141.5	22.98	149.5	25.48
EX810	164.5	11.52	165.8	14.46
EX910	146.0	10.86	159.5	16.27
EX1010	189.5	5.24	202.0	18.13
EX120	130.5	10.87	144.0	26.47
EX220	134.0	19.36	149.5	18.64
EX320	133.5	6.67	142.5	14.55
EX420	153.0	8.64	166.5	14.02
EX520	117.5	15.02	125.0	24.30
EX620	134.0	9.80	139.0	27.33
EX720	151.5	16.00	161.2	18.53
EX820	176.5	25.10	179.0	23.87
EX920	164.5	20.90	177.5	26.89
EX1020	190.5	24.82	194.0	30.83
EX130	142.5	8.92	153.0	31.80
EX230	147.0	14.28	154.0	21.84
EX330	143.0	23.06	151.0	26.74
EX430	170.0	15.69	171.5	16.92
EX530	135.0	24.61	149.5	31.72
EX630	140.5	11.53	154.5	14.36
EX730	167.5	5.28	171.5	30.10
EX830	188.0	5.93	198.0	21.88
EX930	160.0	13.96	165.5	15.18
EX1030	209.0	22.16	220.5	29.43

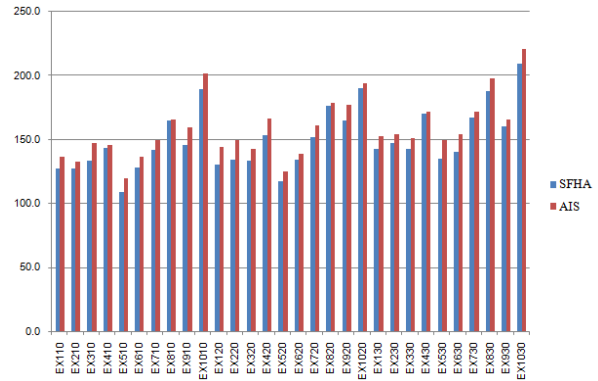
**Table 15**

Makespan values for SFHA and AIS for  $t/p < 0.25$

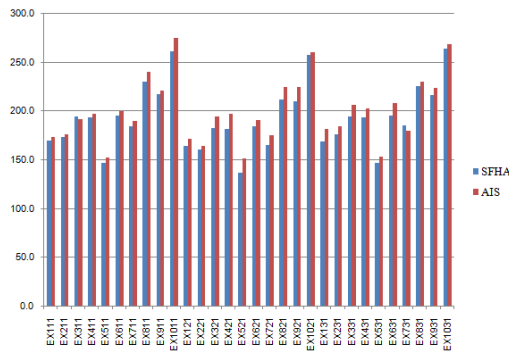
Problem No	SFHA	Computational Time (s)	AIS	Computational Time (s)
EX111	170.0	18.63	173.5	22.66
EX211	173.0	13.08	176.0	23.68
EX311	194.5	7.93	192.0	13.60
EX411	193.5	11.22	197.0	17.62
EX511	146.5	24.64	152.5	33.90
EX611	195.0	10.46	200.0	18.12
EX711	184.5	14.56	190.0	22.33
EX811	230.5	13.91	240.0	19.51
EX911	217.0	17.85	221.5	19.94
EX1011	261.5	9.52	275.5	17.01
EX121	164.0	19.73	172.0	30.73
EX221	161.0	17.93	164.5	27.47
EX321	183.0	5.05	194.5	8.21
EX421	182.0	9.34	197.5	10.94
EX521	137.0	11.69	151.0	21.86
EX621	184.0	6.09	190.5	11.15
EX721	165.0	9.10	175.5	17.25
EX821	211.5	22.87	225.0	28.14
EX921	210.5	10.81	224.5	17.55
EX1021	258.0	10.13	260.5	17.00
EX131	168.5	14.94	182.0	17.84
EX231	176.0	24.54	184.0	28.37
EX331	194.5	12.40	206.0	22.33
EX431	193.5	20.89	202.7	23.14
EX531	146.5	18.63	152.8	22.66
EX631	195.0	13.08	208.7	23.68
EX731	185.5	7.93	179.4	13.60
EX831	225.5	11.22	230.3	17.62
EX931	216.5	24.64	224.1	33.90
EX1031	264	10.46	268.5	18.12



**Fig. 6a.** Results comparison

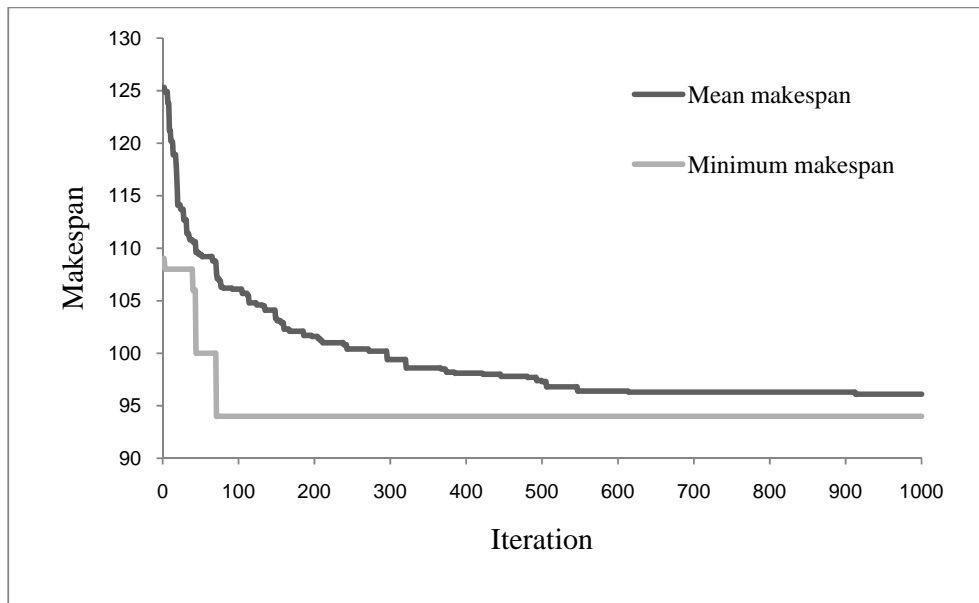


**Fig. 6b.** Results comparison

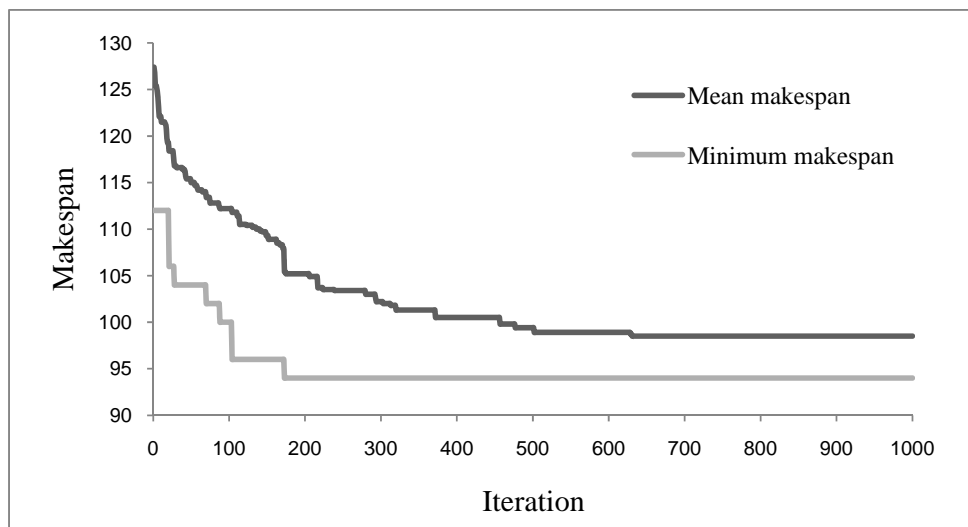


**Fig. 6c.** Results comparison

Fig.7 and Fig.8 present graphs showing the minimum makespan and mean makespan of the population after each iteration obtained using SFHA and AIS for a sample problem. The solution converges at a faster rate and to a lower value for SFHA than AIS. This can be explained by the nature of the operations in both the algorithms. The mutation operations in SFHA are useful in local search, while the crossover operations help discover new solution spaces. In AIS, the mutation operations perform a similar role as in SFHA. But due to the cloning operation, some of the fit solutions are lost. Even though new solutions are discovered using receptor editing, these do not make up for the lost solutions. Hence SFHA is able to maintain a wide range of candidate solutions and find optimal solutions at a faster rate than AIS. In Fig.7 and Fig.8 the mean makespan for both SFHA and AIS reach about the same value at the end of 1000 generations, even as the minimum makespan values differ considerably. This indicates that SFHA provides more diverse solutions compared to AIS. This is found to hold true when the ratio of average travelling time to average processing time is changed.



**Fig. 7.** Convergence of SFHA



**Fig. 8.** Convergence of AIS

## 7. Conclusion and future research

The purpose of this study is to make AGV and robots scheduling an integral part of the scheduling activity. The scheduling of machines, AGV and robots in FMS are addressed using Sheep Flock Heredity Algorithm and Artificial Immune System for the minimization of makespan, mean flow time and mean tardiness. The observed results reveal that SFHA gives better results when compared to AIS algorithm for this particular problem. The algorithms present a good number of diversified solutions for the set of problems considered.

The future research includes modelling of the problem as a real time scheduling problem and with necessary additions. Traffic control and safety can be incorporated for the automated guided vehicles. The number of AGVs and robots can be increased. An automated storage retrieval system (AS/RS) can be incorporated to this problem. The problem can also be extended to more number of machines.

## References

- Abdelmaguid, T. F., Nassef, A. O., Kamal, B. A. & Hassan, M. F. (2004). A hybrid GA/heuristic approaches to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, 42, 267-281.
- Abdelmaguid, T. F. & Nassef, O. A. (2009). A constructive heuristic for the integrated scheduling of machines and multiple-load material handling equipment in job shops. *International Journal of Advanced Manufacturing Technology*, 46(9-12), 1239-1251.
- Attux, E. (2003) A Paradigm for blind IIR equalization using the constant modulus criterion and an artificial immune network. *IEEE XIII Workshop on Neural Networks for Signal Processing, Campinas, Brazil*, 839-848.
- Bilge, U. & Ulusoy, G. (1995) A time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, 43(6), 1058-1070.
- Buzacott, J. A. & Yao, D. D. (1986). Flexible manufacturing systems: a review of analytical models. *Management Science*, 32(7), 890-905.
- Costa, A. M., Vargas, P. A., Von Zuben, F. J. & Franca, P. M. (2002). Makespan Minimization on parallel processors: an immune based approach. *Proceedings of the Special Sessions on Artificial Immune Systems, IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, USA*, 920-925.
- Crama, Y., Kats, V., van de Klundert, J. & Levner, E. (2000). Cyclic scheduling in robotic flowshops. *Annals of Operations Research*, 96, 97-124.
- De Castro, L. N. & Von Zuhlen, F. J. (1999). Artificial Immune systems, Part I. Basic theory and applications, Technical Report, TR-DCA 01/99, 3-14.
- El-Maraghy, H. A. & Ravi, T. (1992). Modern tools for the design, modeling and evaluation of flexible manufacturing systems. *International Journal of Robotics and Computer-Integrated Manufacturing*, 9(4), 335-340.
- Engin, O. & Doyen, A. (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, 20, 1083-1095.
- Gobal, S. L. & Kasilingam, R. G. (1991). A Simulation Model for Estimating Vehicle Requirements in Automated Guided Vehicle Systems. *Computers and Industrial Engineering*, 21, 623-627.
- Heragu, S. S. & Kusiak, A. (1988). Machine layout problem in flexible manufacturing systems. *Operations Research*, 36(2), 258-268.
- Huang, S. (1999). Enhancement of thermal unit commitment using immune algorithms based optimization approaches. *Electrical Power and Energy Systems*, 21(4), 245-252.
- Khoo, L. P., Lee, S. G. & Yin, X. F. (2000). A prototype genetic algorithm-enhanced multiobjective scheduler for manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 16, 131-138.
- Kats, V., & Levner, E. (1998). Cyclic Scheduling of Operations for a Part Type in an FMS Handled by a Single Robot: A Parametric Critical-Path Approach. *International Journal of Flexible Manufacturing Systems*, 10, 129-138.

- Koichi Nara, Tomomi Takeyama & Hyunchul Kim (1999). A New Evolutionary Algorithm Based on Sheep Flocks Heredity Model and Its Application to Scheduling Problem. *IEEE Transactions*, VI, 503-508.
- Lacomme, P., Moukrim, A. & Tchernev, N. (2005). Simultaneous job input sequencing and vehicle dispatching in a single-vehicle automated guided vehicle system: a heuristic branch-and-bound approach coupled with a discrete events simulation model. *International Journal of Production Research*, 43(9), 1911-1942.
- Nara, K. & Kim, H. (1996). Evolutionary Computing by Multi-stage Genetic Operation and its Application to Optimal Scheduling problem. *Proceedings of ALIFE V, Nara, Japan*, 407-412.
- Reddy, B. S. P. & Rao, C. S. P. (2006). A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. *International Journal of Advanced Manufacturing Technology*, 31, 602-613.
- Ronald, L. & Rardin Reha Uzsoy (2001). Experimental Evaluation of Heuristic Optimization Algorithms, School of Industrial Engineering, Purdue University, West Lafayette, IN 47907-1287, USA.
- Satish Kumar, M. V., & Ranga Janardhana Rao, C. S. P. (2011) Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing. *International Journal of Advanced Manufacturing Technology*, 53(1-4), 339-351.
- Srinivas, N. & Deb, K. (1994). Multiobjective optimization using non dominated sorting in genetic algorithms. *Evolutionary Computations*, 2(3), 221-248.
- Stecke, K. E. (1983). Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems. *Management Science*, 29(3), 273-288.
- 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.
- Udhayakumar, P., & Kumanan, S. (2010). Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm. *International Journal of Advanced Manufacturing Technology*, 50(9-12), 1075-1084.
- Ulusoy, G., Sivrikaya-Serifoglu, F. & Bilge, U. (1997). A genetic algorithm approaches to the simultaneous scheduling of machines and automated guided vehicles. *Computers and Operations Research*, 24(4), 335-351.
- Zachariah Aspragathos, N. A. (2005). Optimal robot task scheduling based on genetic algorithms. *Robotics and Computer-Integrated Manufacturing*, 21, 67-79.
- Zheng, H., Zhang, J. & Nahavandi, S. (2004). Learning to detect texture objects by artificial immune approaches. *Future Generation Computer Systems*, 20(7), 1197-1208.