International Journal of Physical Sciences Vol. 4(2), pp. 139-148, March, 2009
Available online at http://www.academicjournals.org/IJPS
ISSN 1992 - 1950 © 2009 Academic Journals

Full Length Research Paper

Scheduling of AGVs and machines in FMS with makespan criteria using sheep flock heredity algorithm

K. V. Subbaiah[1], M. Nageswara Rao[2]* and K. Narayana Rao[3]

[1]Department of Mechanical Engineering, Andhra University of Engineering, Visakhapatnam.
[2]Department of Mechanical Engineering, Andhra University of Engg., Visakhapatnam.
[3]Department of Mechanical Engineering, Government, Polytechnic, Visakhapatnam.

Accepted February 2, 2009

This paper addresses the problem of simultaneous scheduling of machines and two identical automated guided vehicles (AGVs) in a flexible manufacturing system (FMS) so as to minimize makespan and mean tardiness. For solving this problem, a sheep flock heredity algorithm is proposed. An increase in the performance of the FMS under consideration would be expected as a result of making the scheduling of AGVs an integral part of the overall scheduling activity. For this particular problem, coding has been developed, which gives optimum sequence with makespan value and AGV'S schedule for ten job sets and four layouts. Most of the time, results of sheep flock algorithm are better than other algorithm and traditional methods.

Key words: Scheduling, AGVs, FMS.

INTRODUCTION

A flexible manufacturing system (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.

The increased demand for manufactured goods has increased the pressure on the manufacturing system, which in turn has motivated management to find new ways to increase productivity, considering the scarce available resources.

A flexible manufacturing system (FMS) has emerged as a viable alternative to conventional manufacturing system. Existing FMS implementations have already demonstrated a number of benefits in terms of cost reductions, increased utilizations, reduced work-in-process levels, e.t.c. However, there are a number of problems faced during the life cycle of an FMS. These problems are classified into: design, planning, scheduling and control. In particular, the scheduling task and control problem during the operation is of importance owing to the dynamic nature of FMS such as flexible parts, tools, AGV routings and AS/RS storage assignments. These are primarily concerned with scheduling problems of FMS.

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, e.g. material-handling devices like AGVs and AS/RS must be considered.

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 batteries that allow operation for several hours between recharging. The definition of the 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.

Mean tardiness is useful when the objective function of the company includes a penalty per unit of time if a job completion is delayed a specified due date.

LITERATURE REVIEW

Sabuncuoglu and Hommertzheim (1992) addressed the simultaneous scheduling problem using a dynamic programming approach. They tested different machines and AGV scheduling rules in FMS against the mean flow time criterion. Another off-line model for simultaneous sche-

*Corresponding author. E-mail: rao_medikondu@yahoo.co.in.

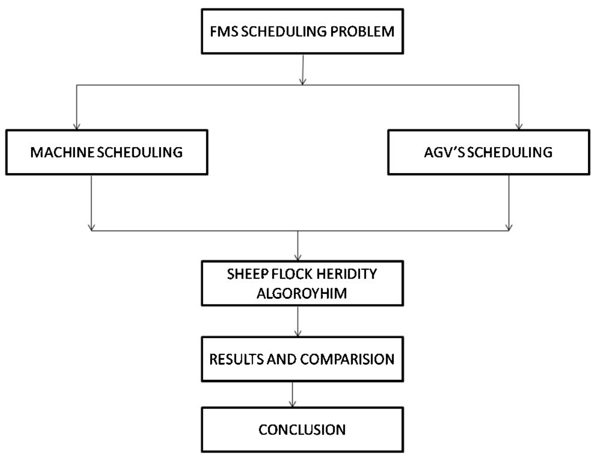

Figure 1. Basic model for this particular problem.

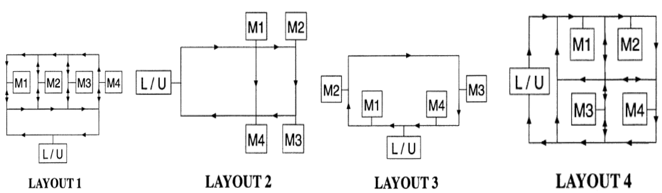

Figure 2. The layout configurations used in generating example problems.

duling of machines and material handling system in an FMS for the makespan minimization is presented by Bilge and Ulusoy (1995). The problem was formulated as a non-linear mixed integer-programming model and was addressed using the sliding time window approach. Ulusoy et al. (1997) has addressed the same problem using genetic algorithms. In their approach, the chromosome represents both the operation number and AGV assignment which requires the development of special genetic operators.

Rao and Reddy (2006), addresses the simultaneous scheduling problem as a multi-object problem in scheduling as scheduling with conflicting objectives are more complex and combinatorial in nature. He solved the problem by non-dominating sorting evolutionary algorithm. Wu and Wysk (1988) described some scheduling algorithm which employs discrete simulation in combination with straight forward part dispatching rules in a dynamic fashion.

METHODOLOGY

In this study, a flexible manufacturing system (FMS) in which material transfer between machines is performed by a number of identical automated guided vehicles (AGVs) is considered, and the problem of simultaneous scheduling of machines and AGVs is addressed. We have considered 4 different layouts and 10 job sets consisting of 1 - 8 different job sets and operations on machines to be performed. The problem is formulated as a nonlinear mixed integer programming model. Its objective is makespan minimization. The formulation consists of constraint sets of a machine scheduling sub problem and a vehicle scheduling sub problem which interact through a set of sheep flock heredity algorithm constraints for the material handling trip starting times. An iterative procedure is developed where, at each iteration, a new machine schedule is generated by a sheep flock heredity algorithm procedure. The Basic model for this particular problem is explained in Figure 1.

Algorithm used

The algorithm applied for the present study is the sheep flock heredity algorithm. It is found that the proposed algorithm referred to as the multi-stage genetic operation can find better solutions than those of the simple genetic algorithm through thermal generator maintenance scheduling examples. For example, each sub-chromosome represents the operational schedule of one machine for several consecutive years, and a whole chromosome presents the operational schedule of multiple machines for multiple years. So as to cope with this kind of special string structure, hierarchical genetic operations (crossover and mutation) are introduced. They are; (1) sub-chromosome level genetic operation and; (2) chro-mosome (global) level genetic operation.
Sheep algorithm is used because of the following;

- It is a multi-stage genetic operation, can find better solutions than those of the simple genetic algorithm.
- Algorithm shows reasonable combination of local and global search.
- The method is effectively applied to planning problems for multiple years, and the method is tested by the real scale generator maintenance scheduling problem.

FMS description

The FMS considered in this work has the configuration as shown in Figure 2. There are four machines having computer numerical machines (CNCs), each with an independent and self sufficient tool magazine, one automatic tool changer (ATC) and one automatic pallet changer (APC).

Assumptions

The types and number of machines are known, there is sufficient input/output buffer space for each machine's machine loading allocation of tools to machine assignment of operation to machine are made pallet and other necessary equipment are allocated. The speed of AGV (40 m/min), the distance between the two machines and the distance between loading/ unloading machines are known.

Input data

The input data that is, traveling time matrix from Table 1 and job sets for the problem is taken from Bilge and Ulusoy (1995). Data given in Table 1 gives the distances from load/unload stations to machines and distances between machines in metres for all the 4 layouts .The 10 job sets given each containing four to eight different job sets, machines in each job set to be processed and numbers within the parenthesis is the processing time of a particular job on a specified machine. The load/unload (LIU) station serves as a distribution center for parts not yet processed and as a collection center for parts finished. All vehicles start from the LIU station initially.

Table 1. Travel time matrix for this particular problem

Layout 1		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		**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	10	2
	M4	4	8	10	12	0
Layout 3		**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		**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

Data for the job sets used in example problems

Job Set 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)

Job Set 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); M4(17)
Job 6: Ml(10); M2(15); M3(12)

Job Set 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);
Job 6: M2(10); M3(15); M4(8);

Job Set 4

Job 1: M4(11); Ml(10); M2(7)
Job 2: M3(12); M2(10); M4(8)
Job 3: M2(7); M3(10); Ml(9); M3(8)
Job 4: M2(7); M4(8); Ml(12); M2(6)
Job 5: Ml(9); M2(7); M4(8); M2(10); M3(8)

Job Set 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)

Job Set 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)

Job Set 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(l1); M2(9); M4(8)

Job Set 8

Job 1: M2(12); M3(21); M4(11); M4(6)

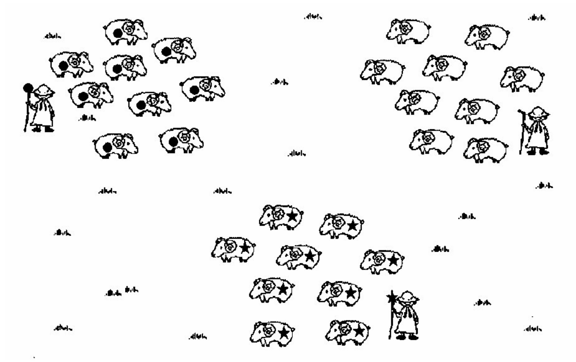

Figure 3. Flocks of sheep in a field.

Figure 4. Mix of two flocks of sheep.

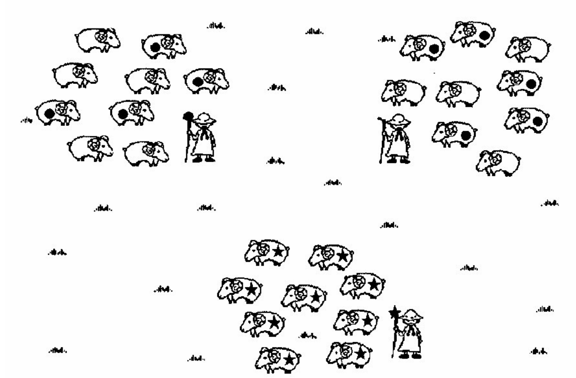

Figure 5. New flock of sheep in field.

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)

Job Set 9

Job 1: M3(9); Ml(12); M2(9);
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)

Job Set 10

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

Objective function

Operation completion time= Oij= Tij + Pij; where j= operation' i= job, Tij= traveling time, Pij= operation processing time.

Mean tardiness= $\frac{1}{n}\sum_{i=1}^{n} T_i$; where n = number of jobs; Ti= tardiness.

Optimization parameters considered

Population size= 10, iterations completed= 1000.

SHEEP FLOCK HEREDITY ALGORITHM

Introduction

Sheep flock algorithm was developed by Hyunchul and Byungchul (2001). Consider the several separated flocks of sheep in a field as shown in Figure 3. 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. In other words, 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

In such a world, let us assume that two sheep flocks were occasionally mixed in a moment when shepherds looked aside as shown in Figure 4.Then, shepherd of the corresponding flocks run into the mixed flock, and separate the sheep as before. However, shepherds can not distinguish their originally owned sheep because the appearance of any sheep is the same. Therefore, several sheep of one flock are inevitably mixed with the other flocks as shown in Figure 5, namely, the characteristics of the sheep in the neighboring flocks can be inherent to the sheep in other flocks in this occasion. Then, in the field, the flock of the sheep which has better fitness characteristics to the field environment breeds most.

The above natural evolution phenomenon of sheep flocks can be corresponded to the genetic operations of this type of string. For this kind of string, we can define the following two kinds of genetic operations:

- Normal genetic operations between strings as shown in **Figure** 7,
- Genetic operations between sub-strings within one string as shown in **Figure** 8.

We will refer to this type of genetic operation to “multi-stage genetic operation”.

GA string can be divided into several sub-strings, and a length of each sub-string is the same. Then, we have the string structure as shown in Figure 6. Figure 6a shows the string structure when it is expanded and Figure 6b shows the same string when it is folded up

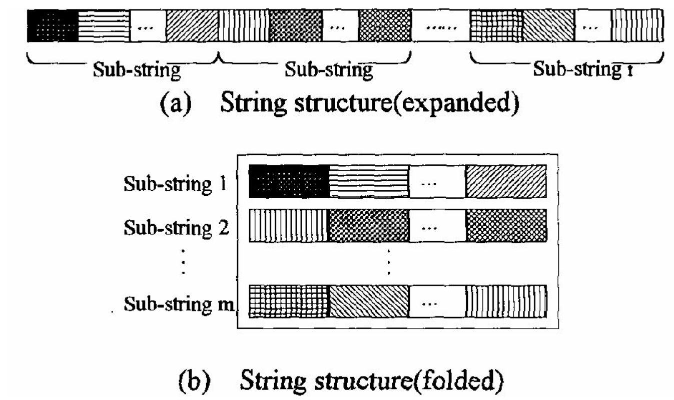

Figure 6. String structure.

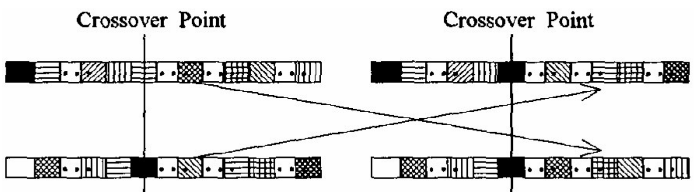

Figure 7. Genetic operations between strings.

in sub-string by sub-string.

Figure 7 shows the genetic operations between strings and Figure 8 shows genetic operations between sub-strings. Let us consider the one to one correspondence of the elements of both actions as shown in Table 2. Then, the inherence within one flock of sheep can be considered as the sub-chromosome level crossover, and the mixing and separating flocks can be corresponded to the chromosome level crossover of the multi-stage genetic operation.

Steps in sheep flock heredity algorithm

- Initial population is generated randomly.
- For each chromosome, evaluate the desired optimization fitness function.
- Do the sub chromosome level crossover and mutation.
- After selecting the best chromosome from the population do the chromosome level crossover and mutatation.
- The fitness function is calculated for each chromosome in the population. Then do the sorting function. After sorting the strings, the new population is cut down to the size of the old population and this completes one generation of genetic process.
- Loop to step (2) until a termination criterion is met, usually a sufficiently good fitness or a specified number of generations

Implementation of sheep flock heredity algorithm

The sheep flock heredity algorithm is implemented for optimizing the sequences of parts into the machines, the AGVs sequence for the problem. A new evolutionary computation algorithm based on Sheep flock heredity is proposed. The algorithm simulates heredity of sheep flocks in a prairie. Algorithm is developed for solving a large scale scheduling problem for a period of several successive years. The multi operative mechanisms of sheep flock are very efficient from a computational standpoint. The sheep flock algorithm

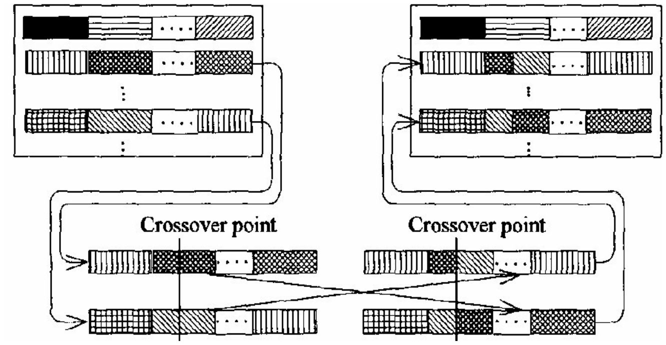

Figure 8. Genetic operations between sub-strings.

was built on the following principles;

- Sub chromosomal crossover
- First stage mutation
- Chromosomal crossover
- Second stage mutation

For implementation of sheep flock algorithm, I have considered Job set 5 and Layout 1.

Job set 5

Job 1			Job 2			Job 3			Job 4		Job 5	
M_1	M_2	M_4	M_1	M_3	M_2	M_3	M_4	M_1	M_4	M_2	M_3	M_1
1	2	3	4	5	6	7	8	9	10	11	12	13

In sheep flock algorithm, first continuous numbers are marked initially for the operations in a job set then random sequence of population ten is generated by following precedence relation that is, operation of the same job set must be in increasing order but anywhere in the sequence.

7 4 5 1 18 9 6 10 12 11 13 2 3

Job	Machine	AGV	travel time	job reach	job ready	job completion
3,1	3	1	0.00	10.00	10.00	19
2,1	1	2	0.00	6.00	6.00	24
2,2	3	2	24.00	8.00	32.00	38
1,1	1	1	18.00	6.00	24.00	30
3,2	4	1	32.00	6.00	38.00	41
3,3,	1	1	41.00	10.00	51.00	63
2,3	2	2	38.00	6.00	44.00	59
4,1	4	2	54.00	12.00	66.00	72
5,1	3	1	63.00	10.00	73.00	76
4,2	2	2	72.00	8.00	80.00	95
5,2	1	1	76.00	8.00	84.00	93
1,2	2	2	86.00	6.00	95.00	07
1,3	4	2	104.00	8.00	112.00	121

Maximum job completion time: 121

placed at different fix points. Then the COF for the mutated sequences is found out, if the COF values are lower than the initial string, then the new string is replaced in place of the initial one, else the initial chromosome is retained.

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

1 2 3 4

Table 2. Correspondence of the elements.

Natural evolution	Multi stage genetic operation
Flock	String
Sheep	Sub-string
Mixed and separated	Chromosome level crossover
Inheritance within flock	Sub-chromosome level crossover

$\underline{1\ 4\ 7\ 10}\ \underline{12\ 2\ 5}\ \underline{8\ 11\ 13}\ \underline{3\ 6\ 9}$

1 2 3 4

subchromosomal crossover

$\underline{12\ 2\ 5}\ \underline{1\ 4\ 7\ 10}\ \underline{8\ 11\ 13}\ \underline{3\ 6\ 9}$

2 1 3 4

Repair function

A repair function is developed that validates chromosomes with any precedence violations. Although some problem specific heuristics are incorporated, the repair function is not designed to be too smart to prevent overly good repairs that lead to high performing children from poorly performing parents. When repairing, care is taken not to create other infeasibilities. Repair is used only to validate offspring generated by operation swap mutation.

Find positions of the operations which violate the precedence relations; If the distance in-between is smaller than half the chromosome length then swap violating operations else choose one of the operations randomly; take it out and reinsert it right before/ after the other one depending m the precedence relations.

First stage mutation

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 inversing the sequence of jobs between i and j positions. If the COF value of the mutated sequence (after inverse mutation) is smaller than that of the original sequence (a generated clone from an antibody), then the mutated one is stored in place of the original one. Otherwise, the sequence will be mutated again with random pair wise interchange mutation.

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

START =6 END =10

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

After repair:

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

Pair wise 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 COF value of the mutated sequence (after pair wise interchange mutation) is smaller than that of the original sequence, then store the mutated one in place of the original one. In the case where the algorithm could not find a better sequence after the two-mutation procedure, then it stores the original sequence (generated clone).

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

First Pos= 5 Second Pos= 9

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

After repair:

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

Chromosomal crossover: After the sub chromosomal crossovers and mutations, the obtained chromosomes are crossovered again by means of chromosomal crossover in which the best five chromosomes which have got the best COF values were chosen and ten new population is generated by means of crossing the chromosomes with the randomly chosen chromosome.

Width= 5

$\underline{4\ 12\ 10\ 1\ 2}\ \underline{11\ 7\ 13\ 5\ 6}\ \underline{3\ 8\ 9}$ 277

1 2 3

$\underline{4\ 12\ 10\ 1\ 2}\ \underline{3\ 8\ 9}\ \underline{11\ 7\ 13\ 5\ 6}$

1 3 2

After repair

4 12 10 1 2 3 7 8 11 9 13 5 6 275

Second stage mutation

Then again, the obtained chromosomes are muted with inverse and pair wise interchange mutations chromosomes after inverse and pairwise mutations thus;

Inverse mutation:

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

START= 3 END =8

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

After repair

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

Pair wise mutation:

First Pos =2 Second Pos =9

RESULTS AND DISCUSSION

10 different job sets with different processing sequences, and process times are generated and presented. Different combinations of these 10 job sets and 4 layouts are used to generate 82 example problems. In all these problems there are 2 vehicles. Table 3 consists of problems whose t_i/p_i ratios are greater than 0.25 while Table 4 consists of problems whose t_i/p_i ratios are lesser than 0.25.

A code is used to designate the example problems which are given in the first column. The digits that follow EX indicate the job set and the layout. In Table 4 another digit is appended to the code. Here, having a 0 or 1 as the last digit implies that the process times are doubled or tripled, respectively, where in both cases travel times are halved.

The problems in Tables 3 and 4 are sorted according to their layouts. Looking closer in Table 3, one can observe

Table 3. Results comparison for t/p ratio >0.25.

Prob No	STW	UGA	AGA	PGA	SFHA
	[6]	[44]	[1]	[34]	
EX11	96	96	96	96	90
EX21	105	104	102	100	96
EX31	105	105	99	99	105
EX41	118	116	112	112	119
EX51	89	87	87	87	87
EX61	120	121	118	118	118
EX71	119	118	115	111	128
EX81	161	152	161	161	137
EX91	120	117	118	116	111
EX101	153	150	147	147	148
EX12	82	82	82	82	80
EX22	80	76	76	76	76
EX32	88	85	85	85	74
EX42	93	88	88	67	96
EX52	69	69	69	69	72
EX62	100	98	98	98	86
EX72	90	85	79	79	87
EX82	151	142	151	151	128
EX92	104	102	104	102	93
EX102	139	137	136	135	130
EX13	84	84	84	84	80
EX23	86	86	86	86	80
EX33	86	86	86	86	79
EX43	95	91	89	89	92
EX53	76	75	74	74	73
EX63	104	104	104	103	86
EX73	91	88	86	83	94
EX83	153	143	153	153	130
EX93	110	105	106	105	94
EX103	143	143	141	139	127
EX14	108	103	103	103	101
EX24	116	113	108	108	113
EX34	116	113	111	111	115
EX 44	126	126	126	126	130
EX 54	99	97	96	96	96
EX 64	120	123	120	120	125
EX 74	136	128	127	126	145
EX 84	163	163	163	163	146
EX 94	125	123	122	122	126
EX 104	171	164	159	158	173

STM- Sliding time window; AGA– Abdelmaguid genetic algorithm; UGA- Ulusoy genetic algorithm; PGA-Proposed genetic algorithm; t- travelling time; p- processing time.

that, while high improvements are achieved on layouts 2 and 3, improvements obtained on layouts 1 and 4 are less. The following steps are used to calculate the mean tardiness of the sheep flock heredity algorithm:

Repair Function: $^{13}/_{2}$= 6.5
12 1 4 2 5 7 10 8 11 13 3 6 9

Step 1: Calculating the average value of makespan by using the relation

$$(C_1 + C_2 + \ldots\ldots\ldots + C_n) / n$$

where; C_1, C_2, …..C_n = first layout make span values of n job sets

Table 4. Results comparison for t/p ratio <0.25.

Prob No	STW	UGA	AGA	PGA	SFHA
	[6]	[44]	[1]	[34]	
EX110	126	126	126	126	119
EX210	148	148	148	148	128
EX310	150	148	150	150	128
EX410	121	119	119	119	112
EX510	102	102	102	102	100
EX610	186	186	186	186	143
EX710	137	137	137	137	137
EX810	292	271	292	292	247
EX910	176	176	176	176	185
EX1010	238	236	238	238	123
EX120	123	123	123	123	132
EX220	143	143	143	114	111
EX320	148	145	145	100	97
EX420	116	114	114	114	140
EX520	100	100	100	100	136
EX620	183	181	181	181	244
EX720	136	136	136	136	155
EX820	287	268	287	287	184
EX920	174	173	173	173	118
EX1020	236	238	236	236	126
EX130	122	122	122	122	136
EX230	146	146	146	146	110
EX330	149	146	146	146	93
EX430	116	114	99	99	110
EX530	99	99	182	182	93
EX630	184	182	137	137	142
EX730	137	137	288	288	137
EX830	288	270	174	174	245
EX930	176	174	237	237	161
EX1030	237	241	124	124	185
EX140	124	124	217	217	118
EX241	217	217	151	151	187
EX340	151	151	221	221	136
EX 341	222	221	172	172	185
EX 441	179	172	148	148	166
EX 541	154	148	184	184	137
EX 640	185	184	137	137	161
EX 740	138	137	203	203	137
EX 741	203	203	293	293	203
EX 840	293	273	175	175	268
EX940	177	175	240	240	146
EX1040	240	244	175	175	185

n = number of job sets

Step 2: Find out the due date (Di).
Here average makespan values are considered as Di.

Step 3: Calculating the lateness value (Li)

Li = makespan – due date

Step 4: Finding out tardiness value (Ti)
Ti = Max (Li, 0)

Step 5: Calculating the mean tardiness value ($\overline{T}$)

Table 5. Mean makespan and tardiness comparison for t/p ratio>0.25.

Layout	STW		UGA		AGA		PGA		SFHA	
	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness
1	118.6	8.3	116.6	7.8	115	8.4	114	8.8	113	8.5
2	99.6	9.8	96.4	9.5	96	10.5	96	9.6	92	7.9
3	102.8	10.2	100.5	9.5	100	10.4	100	10	93	7.3
4	128	8.6	125.3	8.1	123	8.3	123	8.1	127	8.6

Table 6. Mean makespan, Mean tardiness comparison for t/p ratio<0.25..

Layout	STW		UGA		AGA		PGA		SFHA	
	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness	Mean make span	Mean tardiness
1	167	22.4	164	21.3	167	22.4	167	22.4	145	15.4
2	164	22.4	162	21.2	163	22.5	163	22.5	144	15.1
3	165	22.5	163	21.5	164	22.5	164	22.5	145	15.6
4	190	18.75	187	18.58	188	19.5	188	19.5	169	15.25

$$\overline{T} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

Conclusion and Recommendations

In this paper the optimal sequences of Machines and AGVs are determined. The iterative algorithm created anticipates the complete set of flow requirements for a given machine schedule and makes vehicle assignments accordingly, as opposed to a real-time dispatching scheme that uses no information other than the move request queue. The iterative algorithm promises improvement in scheduling especially in environments where cycle times are short and travel times are comparable, or where the layout and the process routes do not suit each other. Most of the times results of Sheep flock algorithm are better than other algorithm and traditional methods. Out of 40 problems 22 problems give better results using SFHA when compared with other four algorithm and same results for 3 problems. It can also be observed from Table 4 that out of 42 problems, 38 problems give better results using SFHA when compared with other four algorithm and same results for the remaining 4 problems (Table 3). From Table 3 it can be observed that out of 40 problems, 29 give better results using SFHA when compared to the sliding time window (Abdelmaguid et al., 2004) and same results for 1 problem; compared with the Ulusoy genetic algorithm (UGA) (Ulusoy et al., 1997) 25 problems give better results and same results for 3 problems and compared with Abdelmaguid genetic algorithm (AGA) (Abdelmaguid et al., 2004), 20 problems give better results and same results for 4 problems.

Optimal and better solutions can be determined within fewer iterations of sheep flock algorithm, when compared with other algorithm in the Tables 3 and 4 respectively. From Tables 5 and 6 we conclude that mean makespan and tardiness values of 4 layouts are better in SFHA (Figures 9 and 10) when compared to other four algorithms. As per Table 5 we conclude that mean makespan and tardiness values of out of 4 layouts, 2 lay outs give better results in SFHA and the other 2 have some variations as compared to the other 4 algorithms. The computational results have indicated that sheep flock algorithm is very effective in generating optimal solutions for FMS.

It is possible to adapt the sheep flock heredity algorithm approach to performance criteria other than the makespan, such as the minimization of maximum tardiness and mean flow time. In machine schedule generation, on the other hand, due date related priority rules can be employed in conjunction with active or non delay schedule generators. Problems can be implemented as real time scheduling problem and with necessary additions; traffic control and safety can be incorporated for automated guided vehicle. The number of AGV'S can be increased. Robots and automated storage retrieval system (AS/RS) can be incorporated to this problem.

LIMITATIONS OF THE WORK

In this particular problem we consider the entire automated guided vehicle as an identical one but in practice that may not be possible. Traffic control, congestion, machine

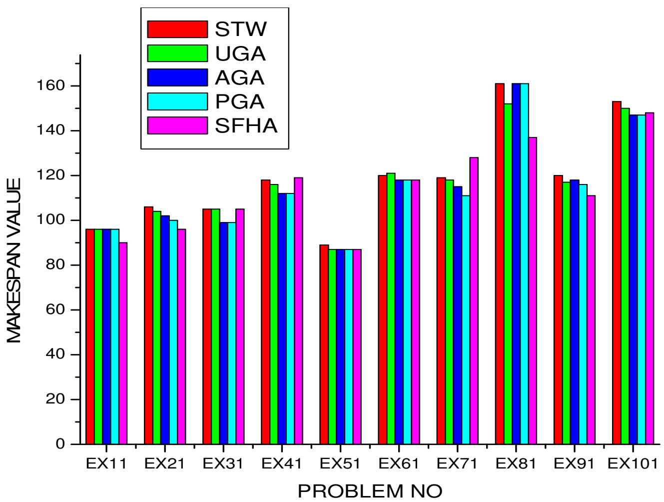

Figure 9. Comparison chart for t/p ratio >0.25.

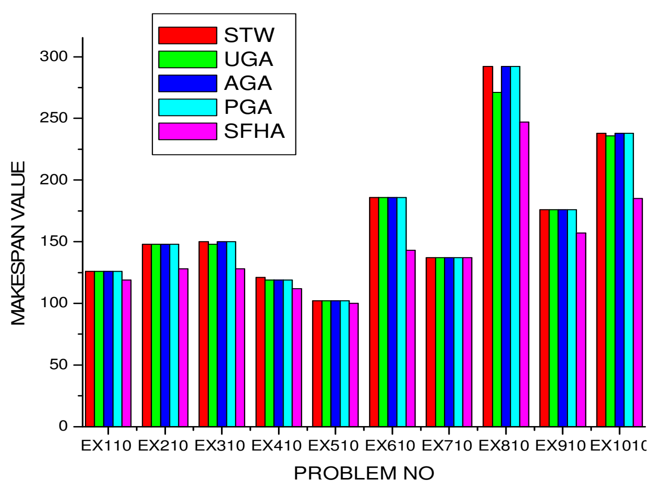

Figure 10. Comparison chart for t/p ratio <0.25.

failure or downtime, scraps, rework and vehicle dispatches for battery changer are ignored here and left as issues to be considered during real-time control. Number of machine considered are 4 which may vary in real time problems and the layout is a simple one.

REFERENCES

Sabuncuoglu I, Hommertzheim DL (1992). "Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system." Int. J. Prod. Res. 30: 1059–1079.

Bilge U, Ulusoy G (1995). A time window approach to simultaneous scheduling of machines and material handling system in an FMS." Oper. Res. 43(6): 1058–1070.

Ulusoy G, Sivrikayaserifoglus F, Bilge U (1997). "A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles." Computer Operations Res 24(4): 335–351.

Reddy BSP, Rao CSP (2006). A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS." Int. J. Adv. Manuf. Technol. 31: 602-613.

Wu SYD, Wysk RA (1988). "Multi-pass expert control system –a control/scheduling structure for flexible manufacturing cells." J. Manuf. Syst. 7(2): 107–120.

Hyunchul K, Byungchul A (2001). "A New Evolutionary Algorithm based On Sheep Flocks Heredity Model." IEEE.

Abdelmaguid TF, Nassef AO, Kamal BA, Hassan MF (2004). A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles." Int. J. Prod. Res. 42: 267–281.