International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 9201-9218

http://www.ripublication.com

Flowshop Scheduling Using Modified Improved Genetic Algorithm in Textile Industry

S.Jeyakkannan*

Department of Mechanical Engineering
Park College of Engineering and Technology
Coimbatore-641 659, Tamilnadu, India
E-mail:sjeyakkannan2006@gmail.com
** Corresponding Author*

T.Prabaharan

Department of Mechanical Engineering
Mepco Schlenk College of Engineering Sivakasi-626 005, Tamilnadu, India.
E-mail: tpraba@mepcoeng.ac.in

K.M.Mahaboob Sheriff

Department of Mechanical Engineering
SBM College of Engineering & Technology Dindigul-624 005, Tamilndu,India
E-mail:meha_sheriff@rediffmail.com

Abstract

Flowshop scheduling problem with processing times in textile yarn manufacturing industry are considered for this work. The objective is to minimize the makespan and meanflow time by using the exact sequencing method. In this paper heuristic algorithm is utilized to obtain high quality solutions in a reasonable amount of time. In this paper Modified Improved Genetic algorithm (MIGA) approach is used to obtain optimal and near optimal solution. The MIGA is also used in the random instances problem and benchmark problems. The performance of the heuristic is evaluated by comparing its solution with other heuristic methods. The MIGA provides better solution than any other heuristic methods

Key words: Flowshop scheduling, makespan, meanflow time, sequencing, Modified Improved Genetic Algorithm

Reference to this paper should be made as follows: Jeyakkannan.S., Prabaharan.T., and Mahaboob Sheriff.K.M., (2014) “Flowshop Scheduling By Using Modified Improved Genetic Algorithm in Textile Industry”,

Biographical notes: S.Jeyakkannan is working as an Associate Professor in Department of Mechanical Engineering, Park College of engineering and technology, Coimbatore, Tamilnadu, India. Currently, he is pursuing his Ph.D under the guidance of Dr. T. Prabaharan, Professor in the Department of Mechanical Engineering, Mepco Schlenk Engineering College, Sivakasi under Anna University, Chennai. He has presented and published a paper in the international journal. His fields of interest are manufacturing scheduling and meta-heuristics.

T. Prabaharan is a Professor in the Department of Mechanical Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India. He received his Ph.D in 2005 from Anna University, Chennai, Tamilnadu, India in the field of scheduling optimisation for automated manufacturing environments. He has published 40 technical papers in refereed journals and conferences. Currently, he is doing his research in scheduling, robotics and production planning.

K.M. Mahaboob Sheriff is currently working as a Professor in the Department of Mechanical Engineering, SBM College of Engineering and Technology, Dindigul. He has published over 10 refereed papers which include both journals and conferences. He has prepared project proposals and submitted to various funding agencies. He has also organized a conference and a number of workshops and training programmes. He is a life member of ORSI and ISTE. His current areas of interest are logistics and supply chain management, scheduling, operations management and meta-heuristics.

1. Introduction:

The flowshop scheduling problem is one of the major and significant types of scheduling problems that was introduced by Holland in 1975. In this kind of problems, a number of machines are located in sense, and jobs will be operated in a fixed order. In Flow shop problem there are some stages. In each stage there is only one machine. Production flow of jobs is from stage one to the last stage and it is possible to complete the task with different length of time. The flow shop with multiple processors (FSMP), has been extensively studied in the literature. An overview on FSMP research is given by linn and Zhang (1999). However, most of the considered problems are theoretical models. . The case we are dealing with presents similarities with some of the two-stage FSMP. With respective to the two-stage FSMP, Narasimhan and Pan-walkar (1984) considered a real-life FSMP with one machine at stage 1 and two machines at stage 2. Setup time represent the amount of time that is needed to prepare the machines for operations are added with processing time.

A Flow shop consists of series of production stages, each of which has several machines operating in parallel that can be identical or uniform. Some stages may have only one machine, but at least one stage must have multiples machines. The flow of jobs through the shop is unidirectional. Each job is processed by one machine in each stage and it must go through one or more stage. A job consists of several operations to be performed by one machine on each stage. In this problem all jobs and machines are available at time zero, machines at a given stage are identical, any machine can process only one operation at a time and any job can be processed by only one machine at a time.

2. Literature Review

FSMP in the literature. Gupta (1988a) showed that the two-stage FSMP problem is NP-hard, developed a heuristic in finding a minimum makespan schedule of a special case when there is only one machine at stage 2. Sriskandarajah and Sethi with the makespan performance. The algorithms were analyzed in the worst and average case performance contexts. Gupta and Tunc (1991) proposed two heuristics to find a minimum makespan schedule for the case when there is only one machine at stage 1. The global lower bounds on the makespan were also discussed. Deal and Hunsucker (1991) studied the FSMP with identical number of machines at the two stages. A lower bound calculation for the makespan was introduced and employed to evaluate the performance of three job sequencing manner.

Research work has also been done in the generalized FSMP case where the number of stages and the number of machines at each stage are not restricted. The logic on how the rules in these papers were developed will be of great value to us in developing good heuristic. Kochhar and Morris (1987) presented heuristics to minimize the mean flowtime for the scheduling problem, which consists of two sub-problems: entry point sequencing and dispatching. Various optimization techniques, including myopic and local search methods, and dispatching methods, trying to minimize the effects of setup time and blocking, were investigated for the two sub-problems, respectively. Wittrock (1985, 1988) proposed an algorithm primarily to minimize the makespan and secondarily to minimize the queue. The basic approach of the algorithm is to decompose the problem into three sub-problems, known as machine allocation, sequencing, and timing, each of which is solved by a heuristic. The following generalized FSMP studies all used the makespan as the criterion. Ding and kittichartphayak (1994) developed three heuristics, in which a job sequence is first determined for the first stage, and then the remaining stages follow the same sequence is first determined for the first stage, and then the remaining stages follow the same sequence. Rajkumar & Shahabudeen (2007) presented an effective Improved Genetic Algorithm (IGA) for flow shop scheduling, incorporating multi-crossover operators, multi-mutation operators and hyper mutation operators which gives a better solution when compared with the earlier reported results. Modified improved genetic algorithm (MIGA) is proposed and developed in this paper by improving improved genetic algorithm by incorporating some cross over and mutation operators. The objective of this paper is to develop a Modified improved genetic algorithm and hence to find the optimal solution using proposed modified genetic algorithm to minimize the make span and mean flow time in a flow shop scheduling problem and hence to find the optimal solution using this proposed MIGA. are discussed below.

3. Problem Definition

A spinning and weaving textile mill situated in South India has been considered in this paper. It produces top quality yarn meeting or exceeding the standards expected by the most quality-conscious buyers in the world. A substantial portion of the output is exported to countries in the EU and South-East Asia. Top domestic buyers have also been regular customers. Designed with flexibility in mind, the mill is able to supply a wide range of products, from the coarsest to the finest counts, from cotton, synthetics and blends.

The various steps involved in the yarn manufacturing process are depicted in the figure1. Different varieties of cotton together mixed to get homogeneous blend of cotton for uniform yarn quality and shade. Then, raw cotton is opened and cleaned in blow room for supply to next department (carding) in the form of lap or sheet (chute feed). Before pre-comber drawing process, it has been converted into sliver form by carding and collected in cans. After passing through pre-comber draw frame, delivered sliver is converted into a lap form by lap former machine. The comber machine separate longer fibres to form sliver in can and feed it to finisher.

Finisher-draw frame converts the collected fibre into a uniform linier density of sliver by doubling and drafting. Then simplex machine is used to convert the sliver to roving form by drafting, twisting and winding on simplex bobbin. To convert the roving to yarn, ring frame empties are used. Final yarn to market is made with the use of Auto coner that will wind the cop form (35 – 40 gm) yarn on cone of 1.0 to 3.0 kg by passing through electronic yarn clearer to clear yarn fault. Packing of individual cone of yarn is carried out in suitable mode (Bag, Carton or Pallet) by confirming its quality and weight as per customer requirement.

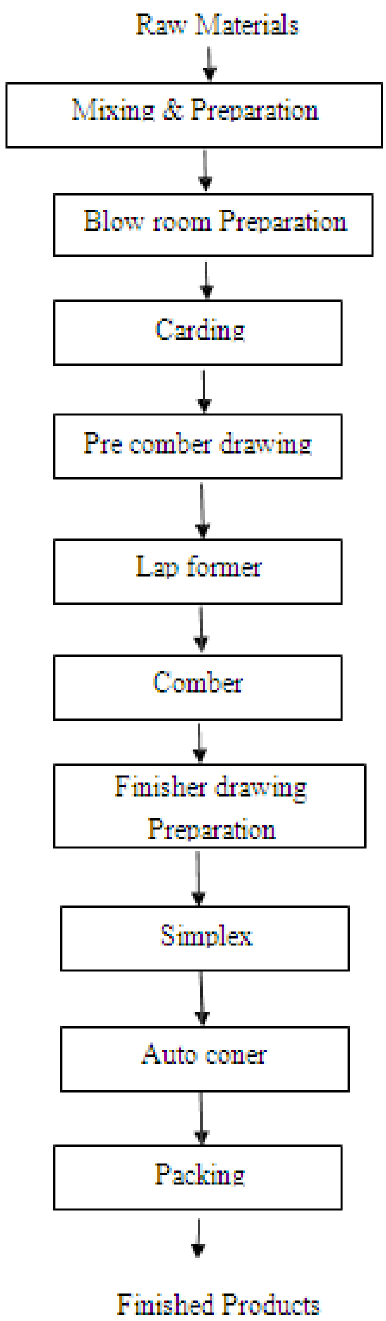

Figure 1 Flow sequence of yarn manufacturing process

Since the yarn manufacturing process involves the usage of number of machines and equipments at various stages, it is quite important to find out the optimal flow sequence of different form of cotton through these machines. The setup and arrangement of machines for yarn manufacturing process resembles the structure of hybrid flow shop scheduling. Hence the yarn manufacturing process is simulated here as a flow shop scheduling problem.

The flow shop scheduling problems can be formally described as follows (*Engin and Döyen, 2004*): Machines are arranged into *s* stages in series; in each stage *k* *(k=1,….,s)* there are m_k identical machines in parallel; job *j*, *j=1,….,n*, has to be processed on any machine at each stage and job *j* has finite processing times in each stage *(p1j, p2j,…., psj)*. It is assumed that all jobs and machines are always available during the scheduled period and the preemption is not allowed.
The objective is to find a schedule which minimizes the maximum completion time (makespan). FS problems are NP-Hard when the objective is to minimize the makespan (*Gupta, 1988*). A flow shop scheduling problem's mathematical model is described as a mix integer programming.

4. Solution Methodology

Modified improved genetic algorithm (MIGA) is proposed and developed in this paper for multi objective flow shop scheduling problem and hence to fine the optimal solution using the proposed modified improved genetic algorithm (MIGA).
Make span Tmax : the length of time required to finish processing all jobs, i.e., Tmax = max $\{p_1,p_2,p_3 \ldots p_n\}$ where c_T is the completion time of job p_n.
Mean flow time; the mean amount of time that all jobs spend in the production system i.e., where T_i is the completion time of job total idle time of machine. MFT = w x $f_1(t)$ + w x $f_2(t)$

J : The set of the jobs to be scheduled. [J] = n.

S : The set of the stage that all the jobs will be processed, [s] = l, J_S : Subscript representing stage / job j.
S_S : Subscript representing stage l.

M_S : Subscript representing machine i.

SM_n: The number of the machines at stage l. P : A large positive number,
JT_S : Starting time of job j at stage l.

JT_P : Processing time of job j at stage l,

W = Weightage factor = 0.5

In MIGA, the following changes are incorporated to avoid premature convergence: Elitist strategy: remove the worst chromosome from the current population and add the best chromosome into that population. Hyper mutation: when the number of generations without improving the best solution greater than a pre-specified constant (Gm), premature convergence can be assumed then increase the probability of mutation (Hyper mutation), and continue the search (Venkata Ranga Neppalli et.al, 1996). The purpose of increasing the probability is to verify the population of MIGA. Re –Assign (Insert) new randomly generated population if makespan is not converging or the generation G_p.

Table 1. shows the necessary parameters involved in the implementation of MA along with its value and symbols.

Table 1. Parameters used in MIGA

Sl. No	Parameter	Symbol	Value
1	Population size	S_p	**100**
2	Probability of crossover	C_p	**0.9**
3	Probability of mutation	M_p	**0.4**
4	Probability for roulette wheel selection	Ro_p	**0.45**
5	Probability for tournament selection	To_p	**0.3**
6	Probability for rank selection	Ra_p	**0.25**
7	Probability for two-point crossover	T_p	**0.5**
8	Probability for order crossover	Or_p	**0.2**
9	Probability for PMX crossover	P_p	**0.125**
10	Probability for SJOX crossover	SJ_p	**0.05**
11	Probability for LOX crossover	L_p	**0.125**
12	Probability for arbitrary two-job change mutation	P_{2j}	**0.05**
13	Probability for arbitrary three-job change mutation	P_{3j}	**0.15**
14	Probability for arbitrary four-job change mutation	P_{4j}	**0.25**
15	Probability for arbitrary five-job change mutation	P_{5j}	**0.35**
16	Probability for shift change mutation	SH_p	**0.2**
17	A constant mutation multiplier	Ψ	**1.2**
18	Maximum number of generations	G_{Max}	10000
19	Generation limit for population re-assignment	P_g	2500
20	Generation limit for changing mutation probability	M_g	**1500**

Steps in MIGA

Step 0: Given the parameter required, such as population size [S_P], cross over probability [C_p], mutation probability [M_p] set K=1 and randomly generate an initial population of size S_{P-1} plus NEH heuristic solution.

$\{P(K) = \{p_1(k), p_2(k) \ldots\ldots P_{sp}(k)\}$

Step : 1 Evaluate makespan for all the chromosomes in and set the best two chromosomes Y* and Y** respectively.

Step : 2 set current best makespan as M_c.

Step : 3 if $M_c = M_{c-1}$ then set count mut = count mut + 1&

cntrndpop =cntrndpop + 1 else set count mut = 0 and cntrndpop = 0.

Step : 4 set r = 0.

Step 5 : Generate a random number r, if $r \leq To_p$, select two parents Y_1 & Y_2 from p(k) by tournament selection, else select the parents by roulette wheel selection.

5.1 perform any one of the following selections.

5.1.1 if $r \leq Ro_p$, select two parents Y_1 & Y_2 from p(k) by roulette wheel selection,

5.1.2 if $r \leq (Ro_p, + To_p)$ perform tournament selection, else select the parents by rank selection. Step 6: Generate r, if $r \leq C_p$, go to 7.1 else go to step 7.2,
6.1 perform any one of the following crossover operations.

6.1.1 generate r, if $r \leq T_p$, perform two point crossover .

6.1.2 if $T_p < r \leq (T_p + Or_p)$ perform order crossover.

6.1.3 if $(T_p + Or_p) < r \leq (T_p + Or_p + P_p)$ perform PMX crossover.

6.1.4 if$(T_p + Or_p + P_p) < r \leq (T_p + Or_p + P_p + SJ_P)$ perform similar job order crossover.

6.1.5. Else perform linear ordered crossover (LOX).

Let Y'_1 and Y'_2 be the offspring's of cross overing parents

Y_1 and Y_2 go to step 7.

Step 7: let $Y_1' = Y_1$ and $Y'_2 = Y_2$.

Step 8 : Generate r, if $r \leq M_p$, go to step 8.1 else go to step 8.2.

8.1 perform any one of the following mutation operations.

8.1.1 generate r, if $r \leq P_{2j}$, perform arbitrary two job change mutation.

8.1.2. if $P_{2j} < r \leq (P_{2j} + P_{3j})$, perform arbitrary three job change mutation.

8.1.3. if $(P_{2j} + P_{3j}) < r \leq (P_{2j} + P_{3j}+P_{4j})$, perform arbitrary four job change mutation.

8.1.4. if $(P_{2j} + P_{3j}+P_{4j}) < r \leq (P_{2j} + P_{3j}+P_{4j}+P_{5j})$, perform arbitrary five job change mutation.

8.1.5. else perform shift change mutation.

Mutate for, Y'_1 to generate chromosome Y''_1 and Y''_2 to generate chromosome Y''_1 and Y''_2 into P (k+1) and let r = r+1, go to step 9.

8.2 Let $Y''_1 = Y'_1$ and $Y''_2 = Y'_2$. then, put Y''_1 and Y''_2 into P(k+1) and let r = r+l;

Step 9 : if r < sp/2, then go to step 5 otherwise, go to step 10.

Step 10 : if count mut > G_m, then change the mutation probability $M_p = \Psi^* M_P$, else no change in M_P
value.

Step 11 : if cntrndpop > P_g , then generate new population of size 0.75 * Sp randomly and replace the current population.

Step 12 : update Y*, Y** and M_c in P(k).

Step 13 : Adopt Elitist strategy. Insert the two best chromosomes Y* and Y** into the current population by removing two worst chromosomes (having max. makespan).

Step 14 : if k > G_{max} go to step 14. Else set k=k+1 and go to step 3.

Step 15 : output makespan value and the corresponding sequence as the result.
stop.

5. Results and Discussion

To solve the FSS problem existing in textile mill environment, necessary data has been collected directly from the textile mill. The setup and sequence of machines for yarn manufacturing process are depicted in figure 1 and the numbers of machines available at each stage for different operations are shown in table 2. Though number of machines in each stage is more than one it is considered that all machines in each stage is considered as one single machine. Summation of machining time and setup time of each job is collectively assumed as job processing time on each machine. With the help of case data obtained from textile mill, real FSS problem is solved by the use of MIGA as per the steps explained in the previous section. The problem is

coded in Html, JavaScript with PHP backend and MySQL database on Intel dual core processor (80 GB HDD, 2 GB DDR2 RAM @ 2.5
GHz). After performing 120 iterations, in textile industry the solution obtained for makespan and meanflow time by MIGA and percentage of improvement is compared with the solution obtained by using GA and IGA for the same case as shown in table 3.The performance comparison clearly indicate that MIGA performs GA and IGA in terms of solution quality in outwell.

Table 2. Arrangement of Machines in Textile Mill

S.No	**Stages**	**Processing machines and operations**
1	S1	Mixing and separation
2	S2	Blow room
3	S3	Carding
4	S4	Pre comber drawing
5	S5	Lap former
6	S6	Comber
7	S7	Finished Drawing
8	S8	Simplex
9	S9	Ring frame
10	S10	Auto coner
11	S11	Packing

Table 3. Impact of Solution Methodologies on Textile Mill Problem for makespan value.

l	n	Makespan Value			Best Solution Method	Improve ment % over GA	Improvement % over IGA
		GA	IGA	MIGA			
11	12	659	598	598	MIGA	9.25	0

Also, the impact of solution methodology on each benchmark problem for makespan value and percentage of improvement is shown in Table4, which clearly indicates that MIGA is performing better than GA and IGA in terms of providing good solution to the FSS problem for makespan value.

Table 4. Impact of Solution Methodologies on Bench Mark Problem for makespan

Random problem Instances	l	n	Makespan Value			Best Solution Method	Improvement % over GA	Improvement% over IGA
			GA	IGA	MIGA			
Car1	5	11	8502	7968	7795	MIGA	8.31	2.17
Car2	4	13	8490	8316	8316	MIGA	2.04	0
Car 3	5	12	8449	8449	8449	MIGA	0	0
Car 4	4	14	8834	8834	8834	MIGA	0	0
Car 6	9	8	8994	8240	8226	MIGA	8.53	0.16
Car 8	8	8	8800	7991	7790	MIGA	11.47	2.51
Rec1	5	20	1444	1437	1431	MIGA	0.9	0.41
Rec7	10	20	1770	1691	1688	MIGA	4.63	0.17
Rec13	15	20	2207	2019	2019	MIGA	8.52	0
Rec19	10	30	2435	2277	2277	MIGA	6.48	0

Further , the impact of solution methodology on each random problem instance for makespan value and percentage of improvement is shown in Table5, which clearly indicates that MIGA is performing better than GA and IGA in terms of providing good solution to the FSS problem for makespan value.

Table 5. Impact of Solution Methodologies on Random Problem for makespan values over GA and IGA with MIGA

Random Problem Instances	No of Machines (l)	No of Jobs (n)	Makespan Value			Percentage of improvement over GA	Percentage of improvement over IGA
			GA	IGA	MIGA		
R1	6	7	169	155	155	8.28	8.28
R2	8	7	298	208	208	0	0
R3	4	8	199	199	199	0	0
R4	5	8	281	281	281	0	0
R5	6	8	242	241	237	2.06	1.65
R6	7	8	562	560	537	4.44	4.1
R7	8	8	239	246	239	0	2.84
R8	5	9	519	489	489	5.78	0
R9	9	9	599	602	580	3.17	3.65
R10	10	9	931	909	887	4.72	2.42
R11	7	10	824	801	797	8.27	0.049
R12	8	10	561	577	557	0.071	8.46
R13	9	10	757	683	683	9.77	0
R14	10	10	1647	1556	1524	7.4	2.05
R15	11	10	1339	1316	1298	3.061	1.36
R16	12	10	2139	2016	1980	7.43	1.78
R17	6	11	902	864	861	4.54	0.034
R18	7	11	1343	1299	1279	4.76	1.53
R19	8	11	1205	1172	1139	5.47	2.81
R20	9	11	930	890	873	6.12	1.91
R21	10	11	1040	1017	1003	3.55	1.37
R22	11	11	1168	1070	1070	8.39	0
R23	11	12	896	874	839	6.3	4

In textile industry the solution obtained for mean flow time and percentage of improvement by MIGA is compared with the solution obtained by using GA and IGA for the same case as shown in table6. The performance comparison clearly indicate that MIGA performs better than GA and equally good with IGA in terms of solution quality.

Table 6. Impact of Solution Methodologies on Textile Mill Problem for Mean flow time.

n	l	Mean flow time			Best Solution Method	Improvement % over GA	Improvement % over IGA
		GA	IGA	MIGA			
12	11	475.32	425.65	425.65	MIGA	10.44	0

Also, the impact of solution methodology on each benchmark problem for mean flow time and percentage of improvement is shown in Table7, which clearly indicates that MIGA is performing better than GA and IGA in terms of providing good solution to the FSS problem .

Table 7. Impact of Solution Methodologies on Bench Mark Problem for mean flow time

Random problem Instances	l	n	Mean flow time			Best Solution Method	Improvement % over GA	Improvement% over IGA
			GA	IGA	MIGA			
Car1	5	11	6414.6	6231.4	6131.8	MIGA	4.4	1.59
Car2	4	13	7014.5	6983.25	6983.25	MIGA	4.37	0
Car 3	5	12	7017.4	7017.4	7017.4	MIGA	0	0
Car 4	4	14	7902.25	7868.25	7868.25	MIGA	0.43	0
Car 6	9	8	6913.66	6475.33	6149.44	MIGA	11.05	5.03
Car 8	8	8	6850.62	6243.13	6419.75	IGA	2.46	-2.82
Rec1	5	20	1284.8	1265.2	1258.6	MIGA	2.02	0.52
Rec7	10	20	1465.9	1406.5	1432.4	IGA	2.28	-1.84
Rec13	15	20	1633.1	1599.47	1599.47	MIGA	2.05	0
Rec19	10	30	2143.6	1981.1	1981.1	MIGA	7.58	0

Further , the impact of solution methodology on each random problem instance for mean flow time and percentage of improvement is shown in Table8, which clearly indicates that MIGA is performing better than GA and IGA in terms of providing good solution to the FS problem .

Table 8. Impact of Solution Methodologies on Random Problem for mean flow time values and over GA and IGA with MIGA

Random Problem Instances	No of Machines (l)	No of Jobs (n)	Mean Flow Time Value			Percentage of improvement over GA	Percentage of improvement over IGA
			GA	IGA	MIGA		
R1	6	7	136.29	115.29	115.29	15.4	0
R2	8	7	154.86	152.71	152.71	1.38	0
R3	4	8	137.13	132.38	132.37	3.471	0
R4	5	8	218.63	204.63	204.63	6.403	0
R5	6	8	176.38	185	182.25	-3.32	1.486
R6	7	8	436	411.88	440.38	-1	-6.91
R7	8	8	176	173.12	173.12	1.63	0
R8	5	9	358.55	350.55	350.555	2.23	0
R9	9	9	459.77	451.44	419.11	8.84	7.16
R10	10	9	718.89	708.62	699	4.74	3.36
R11	7	10	618.1	590.1	561.6	9.14	4.82
R12	8	10	426.1	428.6	431.5	-1.26	-0.676
R13	9	10	571.7	492.7	492.7	13.8	0
R14	10	10	1251.3	1162.3	1173.7	6.2	-0.98
R15	11	10	1002.8	998.1	993.8	0.897	0.43
R16	12	10	1764.2	1576.1	1591.1	9.81	-0.951
R17	6	11	627.36	609.09	600.36	4.3	1.43
R18	7	11	980	887.45	923	5.81	-4
R19	8	11	872.72	867.45	823.27	5.66	5.093
R20	9	11	712	638.45	686.91	3.52	-7.59
R21	10	11	823.82	808.09	802.73	2.56	0.663
R22	11	11	872.82	860.54	860.54	1.406	0
R23	11	12	698.83	686.92	680.92	2.562	0.873

The cause behind this improvement is mainly due to the small changes incorporated in the GA which is explained as follows. Getting initial solution by using NEH heuristic, SPT rule and LPT rule simplifies the computational effort of finding initial best solution by GA. Due to the application of set of crossover operator each with the given probabilities, the diversity can be enhanced and the search region can be extended by using MIGA. In addition to that, selection of a set of mutation operators in MIGA will enrich the search template so that the exploration and exploitation abilities can be simultaneously enhanced on the basis of the advantage of combining several different search mechanisms.

6. Conclusion

This paper studied the real production problem existing in a textile industry. Since the setup and sequence involved in the yarn manufacturing process resembles the FSS problem, it has been treated as a real FSS model with makespan and mean flow time as objective. Since the problem is NP-hard, it has been approached by metaheuristic to get the solution. In this paper, the modified improved genetic algorithm (MIGA) is considered to obtain solution. In MIGA, to increase the algorithm performance and also the quality of the solutions, the parameters of the GA and IGA are tuned by the improvement of the other parameters and it is used.

Initially MIGA has been used to solve the real FSS problem by using case data collected from the textile mill and the result is shown in table 3 and table 6. Similarly, the effectiveness of MIGA is also tested in the benchmark problems which are listed in table 4 and table 7. Similarly the effectiveness of MIGA is also tested in the random problems which are listed in table 5 and table 8. Computational results show high performance of this MIGA over GA and IGA for the textile problem, benchmark and random instances problems in a reasonable time.

This MIGA was able to find optimal (or) near optimal solution for all the problems which highlights the robustness of this algorithm. The experimental results reveal that MIGA has the capability to solve large problems with a reasonable accuracy.

7. References

1. Alaykyran K., Engin O., and Doyen A., 2007 "Using ant colony optimization to solve hybrid flow shop scheduling problems", The International Journal of Advanced Manufacturing Technology, Vol. 35, No. 5-6, pp. 541–550.
2. Arthanary T.S., and Ramaswamy K.G., 1971 "An extension of two machines sequencing problem", OPSEARCH, Journal of the Operational Research Society of India, Vol. 8, No. 4 pp. 10–22,.
3. Brah S.A., and Loo L.L., 1999 " Heuristics for scheduling in a flow shop with multiple processors", European Journal of Operational Research, Vol. 113, No.1,pp 113-122,.
4. Chung-Yee Lee. and George. L., Vairaktarakis, 1994 "Minimizing makespan in hybrid flowshops", Operations Research Letters, Vol. 16, No. 3, pp. 149–158,.
5. Ding F.Y., and Kitticharphayak D., 1994 " Heuristics for scheduling flexible flow lines", Computers and Industrial Engineering, Vol.26, No.1, pp.27-34.
6. Engin O., and Doyen A., 2004 "A new approach to solve hybrid flow shop scheduling problems by artificial immune system", Future Generation Computer Systems, Vol. 20, No. 6, pp. 1083–1095,.
7. Engin. O., and Döyen. A., 2004 "A New approach to solve hybrid flow shop scheduling problems by artificial immune system", *Future generation computer systems*. **20** 1083-1095.

8. Engin. O., Ceran. G., Yilmaz. M.K., 2011 “An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems”, Applied Soft Computing, 11 3056–3065.
9. Figielska E., 2008 “ A new heuristic for scheduling the two-stage flowshop with additional resources”, Computers and Industrial Engineering. Vol.54, No.4, pp.750-763,
10. Gupta J.N.D., Hariri .A.M.A., and Potts.C.N., 1997 “Scheduling a two-stage hybrid flow shop with parallel machines at the first stage”. *Journal of operations research*.69,.171-191.
11. Gupta. J. N. D., 1988 “Two-stage, hybrid flow shop scheduling problem”. Journal of the Operational Research Society, 39(4):359–364, 1988.
12. Gupta. J. N. D., Sexton. R. S., and Tunc. E. A., 2000 “Selecting scheduling heuristics using neural networks”. Int. Journal on Computing, 12(2):150–162.
13. Hong Wang. M. A., 1998 *“A new model in designing neural network in optimization”:* (Business Administration Graduate Program, The Ohio State University, Thesis,).
14. Hung-Tso Lin., Ching-lioa., 2001 “A case study in a two-stage hybrid flow shop with setup time and dedicated machines”
15. Iravani. S.M.R., Teo. C.P., 2005 “Asymptotically optimal schedules for single-server flow shop problems with setup costs and times”, Operations Research, Letters 33 421-430.
16. Jenabi. M., Fatemi Ghomi. S. M. T., Torabi. S. A., and Karimi. B. 2007. “Two hybrid meta-heuristics for the finite horizon elsp in flexible flow lines with unrelated parallel machines”. Applied Mathematics and Computation, 186(1):230–245,.
17. Jin. Z. H., Ohno. K., Ito. T., and Elmaghraby. S. E., 2002. “Scheduling hybrid flowshops in printed circuit board assembly lines”. Production and Operations Management, 11(2):216–230,. Jungwattanakita. J., Reodechaa. M., Chaovalitwongsea. P., Werner. F., 2009 “A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times and dual criteria”, Computers & Operations Research, 36 358 – 378.
18. Low C., 2005 “Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines”, Computers and Operations Research, Vol. 32, No. 8, pp. 2013–2025,
19. Nasser aejaber, chewn- Lung chen, venkateshwaran vempathi. S., 1993 ” An application of genetic algorithms in Flow shop scheduling problems” European Journal of operations Research , Vol.80, Page 389-396.
20. Niu Q., Zhou T.,and Ma S., 2009 “A quantum-inspired immune algorithm for hybrid flow shop with makespan criterion”, Journal of Universal Computer Science, Vol. 15, No. 4, pp.765–785.
21. Oguz. C., and Ercan. M., 2005 “A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks”. Journal of Scheduling, 8(4):323–351,.

22. Oguz. C., Zinder. Y., Do. V., Janiak. A., and, Lichtenstein. M., 2004 “flow-shop scheduling problems with multiprocessor task systems”. European Journal of Operational Research, 152(1):115–131,.
23. Quan-Ke Pan, Ling Wang, Jun-Qing Li, Jun-Hua Duan.A., *2014* “novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimization”, *Omega*, *45*, *42-56*,.
24. Raiz Rube.N., Matoto concepcio n., 2005 “ A genetic algorithm for hybrid flowship, with sequence dependent set upsire and machine eligibility , “ European journal of operations Research vol.196, page 781-800
25. Rajasekaran.S., Vijay Lakshmi G.A., 2004 “ Neural networks Fuzzy Logic and Genetic algorithms synthesis and applications”, prentice hall , New Delhi,
26. Rajkumar. R., and Shahabudeen. P., 2007 “An improved genetic algorithm for the flowshop scheduling problem” International Journal of Production Research, 1-17.
27. Rashidi, M.Jahandar M and Zandieh M., 2010 “ An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines”, The International Journal of Advanced Manufacturing Technology, Vol. 49, No. 9-12, pp. 1129–1139,.
28. Ruiz .R., Rodriguez. J.A., 2010, “The hybrid flow shop scheduling problem”, European Journal of Operational Research, 205, 1–18.
29. Ruiz Rube n., Nago Marcelo seido, Luiz Amtonic Naguerira Lorera 2008 “ A Constructive genetic algorithm for permutation flowshop scheduling ‘ computers & Industrial Engineering , Vol.55 , Page 195-207.
30. Ruiz. R., and Maroto C., 2006 “A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility”. European Journal of Operational Research, 169(3):781–800,. Seidgar. H., Ezzati. M., Kiani. M., Tavakkoli-Moghaddam. R., 2013 “An Efficient Genetic algorithm for two stage Hybrid flow shop scheduling with preemption and sequence dependent setup time” J.Math. computer sci.6[2013] 251-259.
31. Serifoglu. F. S., and Ulusoy. G., 2004 “Multiprocessor task scheduling in multistage hybrid flow- shops”: A genetic algorithm approach. Journal of the Operational Research Society, 55(5):504–512. Shokrollahpour. E., Zandieh M., Dorri B., 2010 “A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem”, International Journal of Production Research, 1–17.
32. Singh M.R., and Mahapatra S.S., 2012 “A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks”, The International Journal of Advanced Manufacturing Technology, Vol. 62, No. 1-4, pp 267-277.
33. Soewnadi. H., and Elmaghraby S.E., 2001 “Sequencing three stage flexible flowshops with identical machines to minimize makespan”, IIE Transactions, Vol.33, No.11, pp 985-994.
34. Sriskandarajah. C., and Sethi S.P., 1989 “ Scheduling algorithms for flexible flowshops: worst and average case. Performance “, European journal of Operational Research,Vol.43, No.2, pp.143-160.

35. Tavakkoli-Moghaddam. R., Safaei N., and Sassani. F., 2009. "A memetic algorithm for the flexible flow line scheduling problem with processor blocking". Computers & Operations Research, 36(2):402–414,.
36. Tavakkoli-Moghaddam. R., Taheri. F., Bazzazi. M., Izadi. M., Sassani. F., 2009 "Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints", Computers & Operations Research, 36 3224-3230.
37. Venkata Ranga Neppalli, C.-L.C. and Jatinder, Gupta. N.D., 1996, "Genetic algorithms for the two stage bicriteria flowshop problem". European Journal of Operational Research., 95, 356–373.
38. Wang H.M., Chou F.D., and Wu F.C., 2011 "A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan", The International Journal of Advanced Manufacturing Technology, Vol. 53, No. 5-8, pp761–776,.
39. Wang. H., Jacob. V., and Rolland. E., 2003 "Design of efficient hybrid neural networks for flexible flow shop scheduling". Expert Systems, 20(4):208–231,.
40. Wu. Y., Liu. M., and Wu. C., 2003. "A genetic algorithm for solving flow shop scheduling problems with parallel machines and special procedure constraints". In Proceedings of the Second International Conference on Machine Learning and Cybernetics, volume 3, pages 1774–1779. IEEE press.
41. Xiao W., Hao P., Zhang S., and Xu X.. 2000 "Hybrid flow shop scheduling using genetic algorithms. In Proceedings of the 3rd World Congress on Intelligent Control and Automation", pages 537–541. IEEE press,.
42. Yaurima. V., Burtseva. L., and Tchernykh. A., 2009. "Hybrid flowshop with unrelated machines, sequence dependent setup time, availability constraints and limited buffers". In press at Computers & Industrial Engineering,.