



GENETIC ALGORITHM OPTIMIZATION OF FLOW SHOP SCHEDULING PROBLEM WITH SEQUENCE DEPENDENT SETUP TIME AND LOT SPLITTING

Gaurav Kumar¹, Shailja Singhal²

Mechanical Engineering¹, Computer Engineering²

¹JMIT Radaur, ²DIET Karnal

House No 1394 Sector-7 Urban Estate Karnal, 132001
India

Abstract: The Flow-shop scheduling is a schedule planning for large volume systems with very less variations in requirements. In flow-shop scheduling problem (FSSP) environment, the objective of this paper is to find an optimal schedule ordering of M machines for the N jobs for Flow shop problem with sequence dependent setup time and lot splitting using genetic algorithm approach (GA). Present work considers two case studies. One case study represents the conventional flow shop whereas other case study represents the general flow shop. In conventional flow shop, all jobs require every machine in a shop whereas in general flow shop a job may not require every machine in a shop. For each case study, five simulation runs are performed for each combination of crossover and mutation probabilities in order to optimize makespan and to find job sequence. Results indicate that optimized value of makespan can be achieved through various job sequence instead of one and job splitting assist the scheduler in reducing makespan.

Keywords: Genetic Algorithm (GA), Flow Shop Scheduling (FSS), Flow Shop Scheduling Problem (FSSP), Sequence dependent setup time (SDST), constructive genetic algorithm (CGA), total weighted completion time (WCT).

I. Introduction

A flow shop is characterized by a unidirectional flow of work, i.e. all jobs have the same processing order through the machines. The flow-shop scheduling problem consists of ' M ' machines and ' N ' jobs. The scheduler's objective is to find an optimal ordering of ' M ' machines for the ' N ' jobs. Flow shop production has been widely used in many industrial applications, such as manufacturing of variety of printers in computer industry.

Sequence Dependent setup (SDST) scheduling problems are frequently encountered in various production systems. When a machine switches from processing one part type to another, setup time and cost may be incurred because of changing the accessory and/or machine settings. The setup time and cost may depend on both the step just processed and the next step to be processed. In Flow Shop, the jobs are processed in lots and a lot consists of several discrete and identical items are to be processed on several machines configured as a flow shop. Instead of transferring the entire lot after all of its items have been processed on a machine, the transferring of items of the lot in smaller batches is called sublotting. This technique of splitting a lot into sublots (also termed transfer lots), and processing different sublots simultaneously over different machines, albeit still maintaining their movement over the machines in accordance with their flow shop configuration, is called lot streaming/lot splitting.

A large number of optimization algorithms have developed to implement the various optimization techniques in recent past. An optimization algorithm may be defined as a procedure, which is executed iteratively by comparing various solutions till the optimum or a satisfactory solution, is found. There are two categories of optimization algorithms (Deb K, 2003). One category includes those algorithms that are deterministic with specific rules for moving from one solution to another (for example lagreingian, branch and bound etc). Another category includes those algorithms that are stochastic in nature and with probabilistic transition rules (genetic algorithm, simulated annealing and tabu search etc). These algorithms are called "Metaheuristics". These are new and gaining popularity due to certain properties which the deterministic algorithms do not have (Deb K, 2003) As SDST Flow shop scheduling with lot splitting involves several parameters, thus it leads to a combinational flow shop problem. In the present work, an attempt is made to optimize SDST Flow shop scheduling problem with consideration of lot splitting using Genetic Algorithm (GA) approach.

II. Literature Review

This research includes literally hundreds of papers in exact techniques and also heuristic and meta heuristic algorithm for flow shop scheduling problems and some of its variants. However, the reality of the production system is more complicated and there is noticeable gap between the theory and application of the existing

methods. Graves (1981) pointed out this problem and proposed several research directions in this area. Some other studies by Cox (1977) and Ford *et al.* (1987) showed that there is very little scope of application of the existing scheduling methods. Nasser *et al.* (1993) studied an application of GA for flow shop problems with makespan as the criterion. Man *et al.* (1996) introduced GA as a complete entity, in which knowledge of this emerging technology can be integrated together to form the framework of a design tool for industrial engineers. Yamada *et al.* (1997) proposed the multi-step crossover fusion (MSXF) as a unified operator of a local search method and a recombination operator in genetic local search. Sun (1998) studied a shop scheduling problem in which set of n jobs to be processed on two machines for getting the objective to minimize makespan. Reeves Colin. R (1998) modified genetic algorithm by taking into account the features of the landscape generated by the operators used for solving the n -job, m -machine permutation flow shop sequencing problem (PFSP) for minimizing makespan. Subodha kumar *et al.* (2000) studied m -machine no-wait flowshops with multiple products requiring lot streaming with makespan minimization as objective. Varela Ramiro *et al.* (2001) presented a strategy to hybridize a standard GA by introducing heuristic knowledge in a fraction of individuals of the initial population. Ponnambalam *et al.* (2001) collected the different GAs from the literature and an attempt was made to evaluate them. JeongIn-Jae *et al.* (2003) solved a flow shop scheduling problem with no-wait flexible lot streaming. Chingjonglaio *et al.* (2003) discussed a scheduling problem taken from a label sticker manufacturing company. The production system was a two-stage hybrid flow shop with the characteristics of sequence-dependent setup time at stage 1, dedicated machines at stage 2, and two due dates. Zalinda Othman *et al.* (2004) explored the use of Genetic Algorithms (GA) in solving the problem associated with the integrated production scheduling. Ruiz (2005) studied an approach to solve hybrid flow shop with sequence dependent setup times using GA approach. Marimuthua *et al.* (2006) addresses the problem of making sequencing and scheduling decisions for n jobs– m -machines flow shops under lot sizing environment. Cohn Amy (2007) addresses problem of how to sequence products in a blocking flow shop so as to minimize makespan. Yong *et al.* (2008) proposed a novel genetic chromosome-encoding approach. In this encoding method, the operation of crossover and mutation was done in three-dimensional coded space. Cheng Shu-Chen *et al.* (2008) presented an effective hybrid approach based on constructive genetic algorithm (CGA) for PFFS scheduling with the criterion to minimize the total weighted completion time (WCT).

2.1 Problem Formulation

Literature review reveals that there is a little attempt that has been made to solve flow scheduling with the consideration of SDST and lot splitting. The present work is an attempt in this direction. The problem can be defined as “There is a given set of ‘ N ’ jobs that are to be processed on a set of ‘ M ’ Machines. The operation processing time of every operation of each job on a machine is known in advance. The set up time of every operation on a machine for a job is also known. The objective is to find an optimal schedule for makespan performance measure using Genetic Algorithm approach.

III. Adopted Methodology

The methodology adopted to optimise the SDST flow shop scheduling problem with lot splitting is described in following section and sub sections.

3.1 Genetic Algorithm Procedure

The skeleton or framework of the GA methodology is discussed below.

3.1.1 Initialization

The performance of GA is found better with a random start than from a preselected starting population (Anderson, Ferris, 1994). Thus here also, initial population is generated randomly.

3.1.2 Fitness Function

The objective of schedule generation is to minimize the makespan performance measure. As the scheduling problem is of minimization type, thus following function is used to compute the value of fitness function.

$$F(x) = 1/(1 + f(x))$$

Where $f(x)$ is the makespan.

3.1.3 Selection

In the present study, Tournament Selection is applied. In Tournament Selection a group of ‘ q ’ individuals is randomly chosen from the population. These elements take part in the tournament. i.e a winning individual is determined depending upon minimum makespan or maximum fitness value. However, tournament can be generalized to an arbitrary group size ‘ q ’ called tournament size or selection pressure. The selection pressure can

be taken as 2 or 3 or higher depending upon problem. Too small Selection pressure leads to chances of premature convergence. Thus in this present study, selection pressure is taken as 2.

3.1.4 Crossover

In this study, two point crossover is used. For a crossover, two strings are to be selected randomly to make a pair for crossover. As population size is 20 so there are 20 strings after selection, thus it will give rise to 10 pair of strings. For each pair, before crossing over, a check is carried out whether crossover is desired or not with the crossover probability (p_c). Moreover crossing position is selected randomly from between 1 to 7.

For Example

String 1: AB|CDE|FGH
String 2: BC|AEF|DHG

Let crossover positions selected are after 2 and after 5. The elements between these are exchanged in parent strings, keeping other bits unchanged. So off springs produced are

New String 1 after crossover: AB **AEF** FGH

New String 2 after crossover: BC **CDE** DHG

Due to above crossover methodology, some illegal off springs may generate. Thus, repairing is done to resolve the illegitimacy of off springs after mutation.

3.1.5 Mutation

In the present work, Position Based mutation is used. Similar to crossover in mutation also, a check is performed for each string whether mutation is desired or not, with mutation probability (p_m). The mutation position in a string is also determined randomly.

Example:

String before mutation: AB**H**CDGEF

Let the randomly selected job is H and position after mutation is 5. Thus the string after mutation is

So string after mutation: ABCD**H**GFEF

3.1.6 Repairing

As discussed earlier, some illegal off springs may generate in the above method. For this repairing is needed to resolve the illegitimacy of off springs after mutation. A repairing procedure is utilised for this purpose. It checks the string from left to right. If at any point, a job type repeats more than required and some job type is missing then excess job type at any place is replaced by the missing job type.

Example

String before Repairing: ABAEFFGH

String after Repairing: CBAEFDGH

In the above example, A and F is repeated twice and C and D are missing. So A and F are replaced at any random position by C and D.

3.1.7 Reproduction

Once offspring's are generated after crossover and mutation operations, they along with parent population form the extended population. From this extended population, next generation is formed by taking all the offspring's and remaining individuals are taken from previous population. Those individuals from previous population are taken in which neither crossover nor the mutation has taken place. This process is continued to make successive iterations till the termination criteria is met or there is no further improvement in the fitness value.

3.1.8 Regeneration

In order to avoid premature convergence in the population, regeneration scheme as described below by Alcaraz et al. (2003) is utilised. The scheme is discussed below.

1. Create a sorted list of elements with the make span of the chromosomes in ascending order.
2. Select the chromosomes in the first 20% of elements (0.2% of Population)
3. Next 40% of chromosomes are generated by copying a randomly chosen chromosome from the first 20% of population. This new copied chromosome is mutated once with the SHIFT mutation.
4. The remaining 40 % are completely new random chromosome.

The regeneration procedure is applied if there is no successive improvement in the fitness value for 10 successive iterations

IV. Results And Discussions

In the present work, an attempt is made to optimize Flow Shop Scheduling problem with Sequence Dependent Setup Time and Lot Splitting using Genetic algorithm approach. To demonstrate the working of our adopted procedure as explained in section 3, twelve case studies are taken into consideration. These case studies are either randomly generated or taken from literature. Six case studies represent conventional flow shop where each job requires all machines available on the shop. Remaining six case studies represent a general flow shop where some jobs require few machines available in the shop and other jobs require some other machines available on shop. The range of parameters taken for case studies are summarized in Table 4.1

Table 4.1 Range of Parameters for Case Studies

S.No	PARAMETER	RANGE
1.	Number of Machines	6 -8
2.	Number of Jobs	4-8 in step of 2
3.	Processing Time of jobs on a Machine	8-20
4.	Setup time of jobs on machine	1-7
5.	Production Quantity of each job	1 to 5 in step of 1

For case studies of conventional flow shop, optimization is carried out by taking into consideration, three combinations of crossover and mutation probabilities viz $p_c=0.7$ $p_m=0.05$, $p_c=0.8$ $p_m=0.1$ and $p_c=0.9$ $p_m=0.15$ respectively. Five simulations runs for each combination of parameter are carried out and optimization yields the best makespan as well as jobs sequence among the five simulation runs. Each case study of general flow shop is carried out by taking into consideration of one combination of crossover and mutation probability Viz. $p_c=0.8$ $p_m=0.1$ respectively. Five simulation runs are carried out and optimization yield the best makespan and job sequence.

Case Study 1:

It is a 8 jobs 8 machines flow shop problem has been taken into consideration Table 4.2 and 4.3 provides the details of processing times as well as setup time information for the case study.

Table 4.2 Processing Time of Jobs on Machine

Job \ M/c	1	2	3	4	5	6	7	8
A	17	8	14	9	20	9	11	16
B	12	18	17	7	13	10	9	16
C	19	8	15	16	9	16	14	11
D	13	17	9	13	15	8	19	12
E	16	9	14	14	12	7	19	12
F	7	13	11	16	20	9	12	19
G	11	18	19	8	9	16	20	7
H	11	18	17	10	15	10	16	9

Table 4.3 Setup Time of Jobs on Machine

Job \ M/c	1	2	3	4	5	6	7	8
A	2	3	4	3	1	7	6	1
B	6	4	3	1	3	2	6	4
C	1	4	5	4	7	1	3	2
D	1	2	3	4	5	6	7	4
E	2	6	7	1	4	3	1	2
F	1	2	3	4	3	2	3	1
G	4	3	5	2	1	7	3	1
H	2	4	2	3	4	7	2	3

The schedule for above case study is optimized using the adopted methodology for three combinations of p_c and p_m as described earlier by varying the production quantity of each job from 1 to 5 in step of 1. But the

combination $p_c=0.8$, $p_m=0.1$ yield best and quick results. Five simulation runs for each combination of parameters are performed.

Figure 4.1 shows the evolution of fitness value (makespan) with the generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 1. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 261 and FCDEBAHG respectively.

Figure 4.1 Evolution of Best Fitness Value with Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=1)

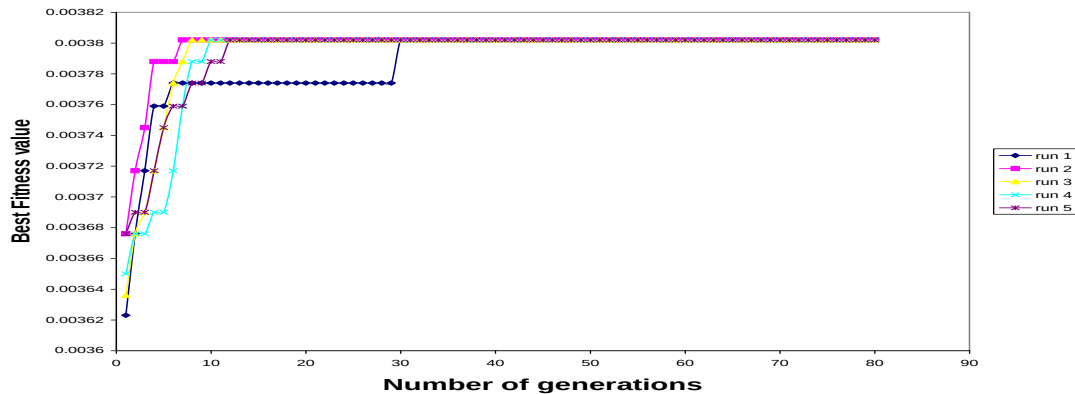


Figure 4.2 shows the evolution of fitness value (makespan) with the generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 2. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 409 and FDADCCBAGBHHEEG respectively.

Figure 4.2 Evolution of Best Fitness Value with Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=2)

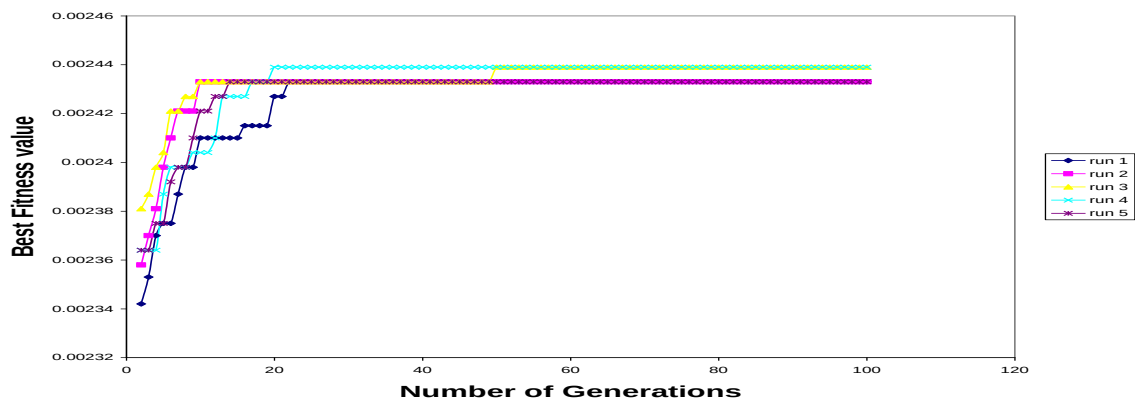


Figure 4.3 shows the evolution of fitness value (makespan) with the generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 3. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 559 and DFDCAHEDFGAAHBBGFBCCEHEG respectively.

Figure 4.3 Evolution of Best Fitness Value with Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=3)

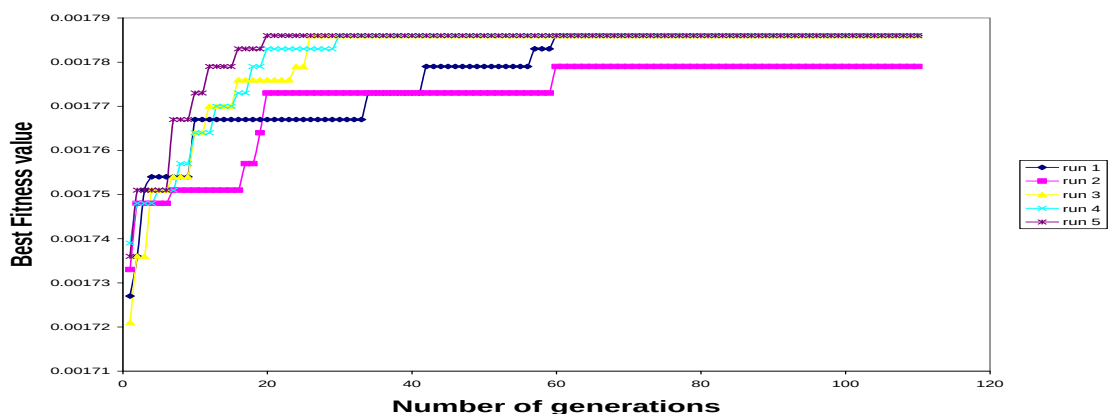


Figure 4.4 shows the evolution of fitness value (makespan) with the generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 4. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 707 and FEDCGDFDAEABHGHFDABAFAGBHCHBEECCG respectively.

Figure 4.4 Evolution of Best Fitness Value with Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=4)

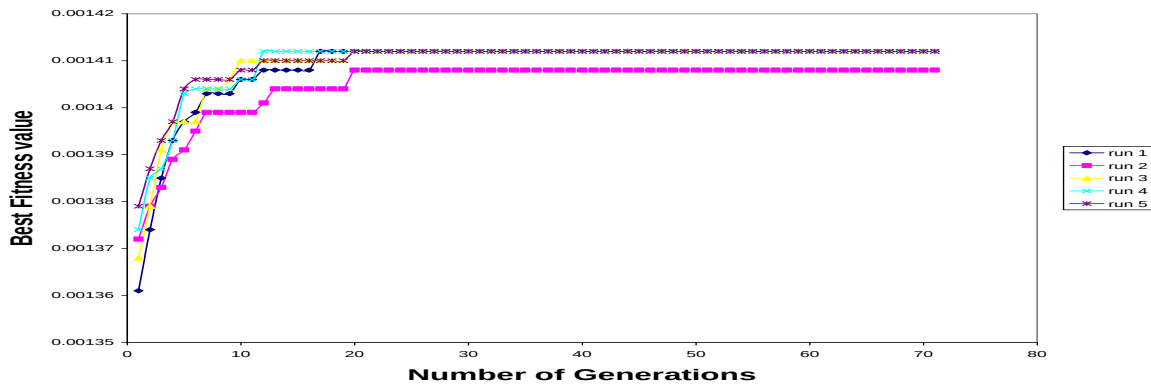
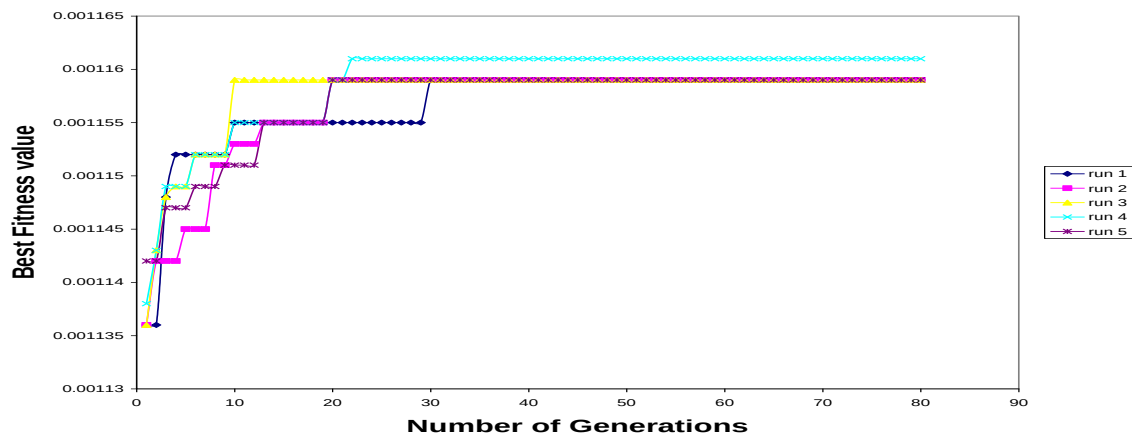


Figure 4.5 shows the evolution of fitness value (makespan) with the generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 5. It clearly reveals that makespan get stabilized at termination criterion. The best makespan & sequence of jobs among five simulation runs are 860 and DDCFEGDCDAECGAFGHFFEBHBAAHHEFGCBBHDCBAEG respectively.

Figure 4.5 Evolution of Best Fitness with Generation ($P_c=0.8$, $P_m=0.1$ Lot size is 5)



Similarly, results are generated for $p_c=0.7$, $p_m=0.05$, $p_c=0.9$, $p_m=0.15$ by varying the production quantity of each part from 1 to 5 in step of 1. The evolution of generation curves are not presented here for want of space.

Table 4.4 summarizes the results obtained from the case study. It clearly reveals that as the production quantity of part increases, there are more than one sequence that can yield same makespan. Moreover, the differences in the optimized value of makespan due to various combinations of p_c and p_m is very less and can be ignored. The evolution of generation curves reveals that $p_c=0.8$ and $p_m=0.1$ combination reaches the optimal solution quickly as compared to other two considered combinations.

Table 4.4 Optimum Sequence and Makespan for Case Study

Lot Size	$p_c=0.7$, $p_m=0.05$		$p_c=0.8$, $p_m=0.1$		$p_c=0.9$, $p_m=0.15$	
	Best Makespan	Best Sequence	Best Makespan	Best Sequence	Best Makespan	Best Sequence
1	262	FCHBDEAG	261	FCDEBAHG	262	FDEACGBH
2	412	ADEFDACBGHFCHE BG	409	FDADCCBAGBHHFE EG	409	FDHAEDGAEHFCBCBG
3	562	DCFEDCBAHDFEBG EHFAACBGHG	559	DFDCAHEDFGAAHB BGFBCCEHEG	559	FEDHADFCGBGHADC BBHAFCEG
4	708	FEGDEEDHADAFBG CHHACCFBGDFEHA CBBG	707	FEDCGDFDAEABHG HFDABAFAGBHCHBE ECCG	707	FEDEHDAGFGGDBAHF BBCADHCACEFHCEBG
5	860	FEGGADCHFEGBAD CHFEGBADCHFEBH ADDFAHBBCCEG	860	DDCFEGDCDAECGA FGHFFEBHBAAHHEF GCBHDCBAEG	861	FEDADCGFCHBADEGA EFHECAGHFBHADC EFHBBDCG

Case Study 2

It is a 8 Jobs, 8 Machines General Flow Shop problem. This is a typical flow shop problem where some jobs require few machines available in the shop and other jobs require some other machines available on shop. Table 4.5 shows the processing time of various jobs on the machines. A zero entry in a cell indicates that a job does not require a particular machine. The setup time of jobs on various machines are shown in Table 4.6

Table 4.5 Processing Time of Jobs on Machines

Job \ M/c	1	2	3	4	5	6	7	8
A	17	0	14	9	20	9	0	16
B	12	18	17	0	13	0	9	16
C	0	8	15	16	0	16	14	11
D	13	17	0	13	15	8	19	0
E	16	0	14	0	12	7	19	12
F	0	13	11	16	20	0	12	19
G	11	18	0	8	9	16	20	0
H	11	18	17	10	0	10	0	9

Table 4.6 Setup Time of Jobs on Machines

Job \ M/c	1	2	3	4	5	6	7	8
A	2	0	4	3	1	7	0	1
B	6	4	3	0	3	0	6	4
C	0	4	5	4	0	1	3	2
D	1	2	0	4	5	6	7	0
E	2	0	7	0	4	3	1	2
F	0	2	3	4	3	0	3	1
G	4	3	0	2	1	7	3	0
H	2	4	2	3	0	7	0	3

Figure 4.6 shows the evolution of fitness value (makespan) with generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 1. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 205 and FCBEGAHD respectively.

Figure 4.6 Evolution of Best Fitness Value With Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=1)

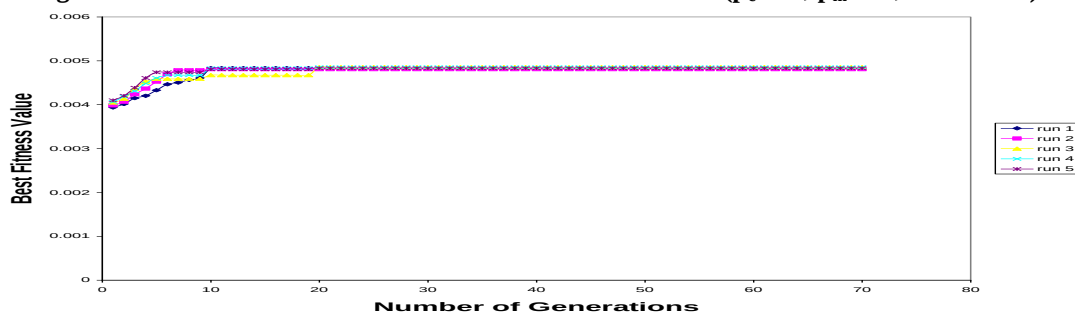


Figure 4.7 shows the evolution of fitness value (makespan) with generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 2. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 346 and FGAHDEFBCGAHBCED respectively.

Figure 4.7 Evolution of Best Fitness Value With Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=2)

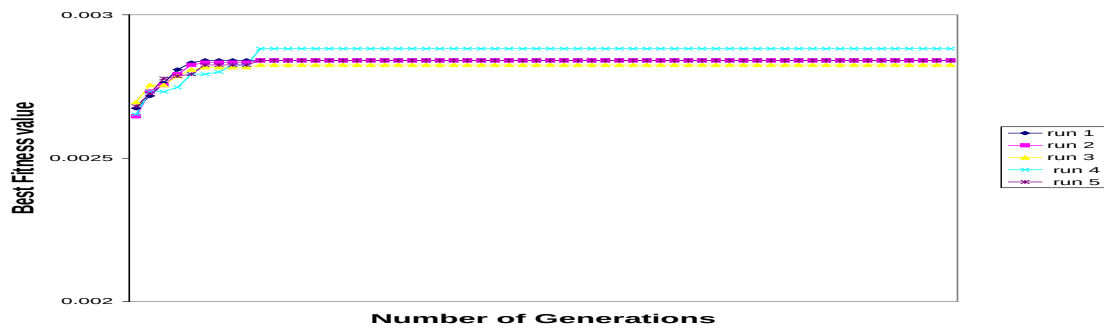


Figure 4.8 shows the evolution of fitness value (makespan) with generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 3. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 455 and AFCCEDGAFCHBEDGAFHHBBEDG respectively.

Figure 4.8 Evolution of Best Fitness Value With Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=3)

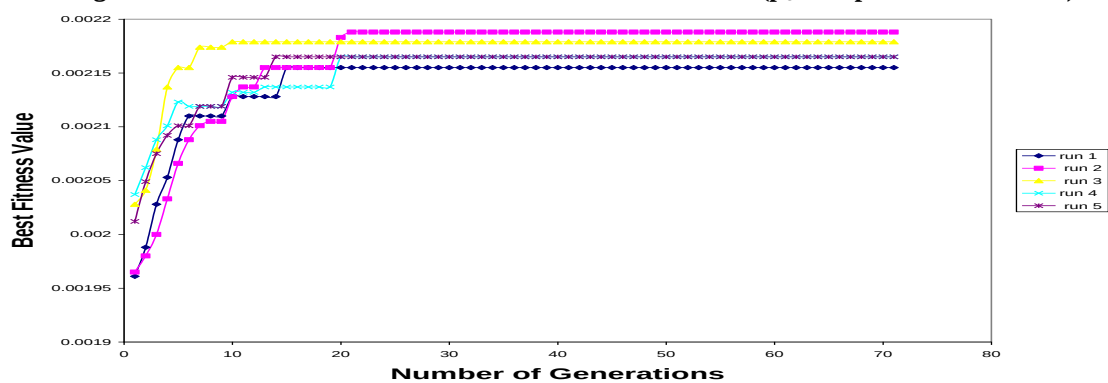


Figure 4.9 shows the evolution of fitness value (makespan) with generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 4. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 584 and CCGBDBHDACFEHGBDAFBHFFCEGEAHDAEG respectively.

Figure 4.9 Evolution of Best Fitness Value With Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=4)

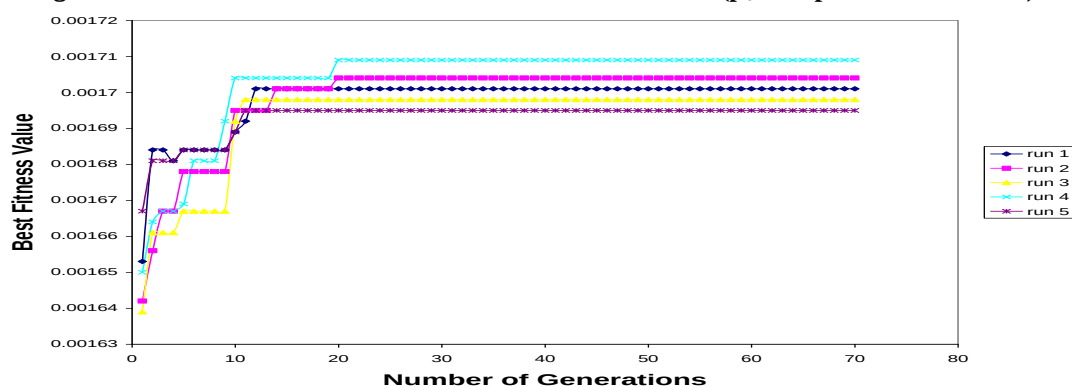
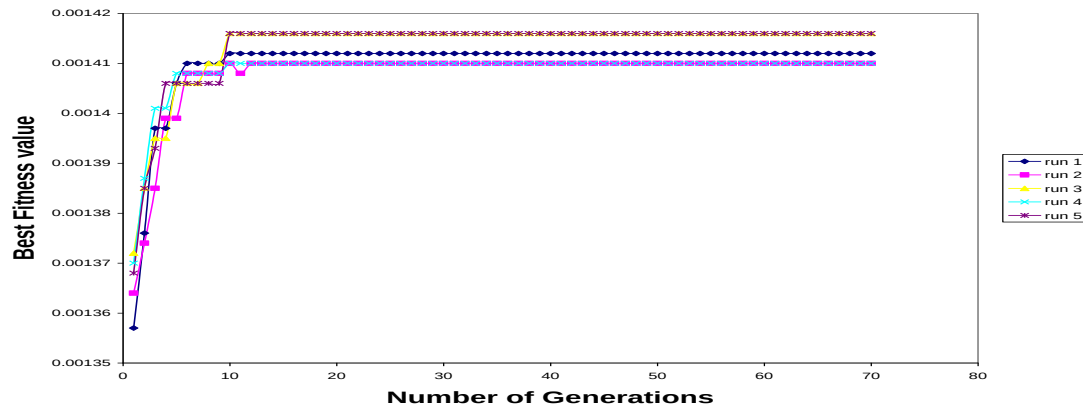


Figure 4.10 shows the evolution of fitness value (makespan) with generation for crossover probability of 0.8, mutation probability of 0.1 and production quantity of each part is 5. It clearly reveals that makespan get stabilized at termination criterion. The best makespan and sequence of jobs among five simulation runs are 584 and CCGBDBHDACFEHGBDAFBHFFCEGEAHDAEG respectively.

Figure 4.10 Evolution of Best Fitness Value With Generation ($p_c=0.8$, $p_m=0.1$, Lot Size=5)



From these figures, Table 4.22 is prepared that shows the sequence as well as optimized makespan for $p_c=0.8$ and $p_m=0.1$. It Clearly reveals that lot splitting assists the scheduler in reducing makespan.

Table 4.6 Optimum Sequence and Makespan for Case Study 7

Lot Size	$p_c=0.8, p_m=0.1$	
	Best Makespan	Best Sequence
1	205	FCBEGAHD
2	346	FGAHDEFBCGAHBCED
3	455	AFCCEDGAFCHBEDGAFHHBBEDG
4	584	CCGBDBHDACFEGHBDABHFFCEGEAHDAEG
5	705	HGHGBAAFGDEEBAFHCDEGAAFHCDEGFCEDCCHBBBDF

V. Conclusions

In the present work, an attempt has been made to solve Flow Shop Scheduling with Sequence Dependent Setup Time and Lot Splitting using Genetic algorithm approach. Twelve case studies are taken into consideration. Six case studies represents a conventional flow shop manufacturing scenario where as remaining six case studies represents general flow shop manufacturing scenario. For conventional flow shop manufacturing scenario, Results are obtained by considering three combinations of crossover and mutation probabilities viz, $p_c=0.7$ $p_m=0.05$, $p_c=0.8$ $p_m=0.1$ and $p_c=0.9$ $p_m=0.15$ respectively. For general flow shop manufacturing scenario, Results are obtained for $p_c=0.8$ $p_m=0.1$ combination. From the analysis of case studies, following conclusions are drawn.

- For conventional flow shop, there is no effect of p_c and p_m combination on the optimal makespan. However, $p_c=0.8$ $p_m=0.1$ combination yields optimal value of makespan quickly (i.e. in less no of generations) as compared to other combinations.
- For conventional flow shop, optimal makespan can be achieved by various jobs sequences instead of one.
- For conventional flow shop and general flow shop, lot splitting assists the scheduler in reducing makespan.

References

- [1] Alcaraz, J., Maroto, C., Ruiz, R. (2003), "Solving the Multi-Mode resource-constraints Project Scheduling problem with Genetic Algorithms" Journal of the Operational Research Society, Vol. 54, Page 614–626.
- [2] Allahverdi A., Ng C.T., Cheng T.C.E., Kovalyov M.Y. (2008), "A Survey of Scheduling Problems with Setup Times or Costs" European Journal of Operational Research, Vol. 187, Page 985-1032.
- [3] BetulYagmahan, Mehmet MutluYenisay: "Scheduling Practice and Recent Developments in Flow Shop and Job Shop Scheduling" Department of Industrial Engineering Gorukle Campus, 16059 Bursa Turkey.
- [4] Błażewicz, Ecker, Pesch, chmidt, Węglarz, J. (1996), "Scheduling computer and manufacturing processes". Springer, Berlin.
- [5] Conway, Miller, Maxwell (1967), "Theory of scheduling". Addison Wesley Publishing Company, Massachusetts.
- [6] Cohn Amy, KurnazSelin, and Yiham Guan (2007), "Flow Shop Sequencing: Beyond Makespan Minimization".
- [7] Cheng, T.C.E., Ding, Q., Lin, B.M.T. (2004), "A Concise Survey of Scheduling with Time-dependent Processing Times" European Journal of Operational Research, Vol. 152, Page 1-13.
- [8] Deb, K. (2003), "Optimization for Engineering Design, Algorithm and Examples", Prentice Hall, New Delhi, India.
- [9] Faith Barnard, Paula deSzoek, RasaratnamLogendran (2005), "Sequence-dependent group scheduling problems in flexible flow shops" Int. J. Production Economics, Vol. 102, Page 66–86.
- [10] Graham, R.L., Lawler, Lenstra J.K. and RinnooyKan (1979), "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey" Annals of Discrete Mathematics, Vol. 5, Page 287-326.
- [11] Goldberg, D.E. (1989), "Genetic Algorithm in search, optimization & Machine Learning", Pearson Education.
- [12] Gen Mitsuo, Cheng R. (1997), "Genetic Algorithm and Engineering Design", John. Willey & Sons.Inc.

- [13] Hall Nicholas, Gilbert Laporte, Esaignanisilverajah, ChelliahSriskandarajah (2003), "Scheduling and Lot Streaming in Flowshops with no wait in process" *Journal of Scheduling*, Vol. 6, Page 339-354.
- [14] Jeong In-Jae, Kwanwoo Kim (2007), "Flow shop scheduling with no-wait flexible lot streaming using an adaptive genetic algorithm" *Int J AdvManufTechnol*, Vol. 44, Page 1181–1190.
- [15] Kumar Subodha, Tapan, P., Sriskandarajah, C. (2000), "Lot streaming and scheduling heuristics for m-machine no-wait Flowshops" *Computers & Industrial Engineering*, Vol. 38, Page 149-172.
- [16] Leondes, C.T. (2002) "Computer-aided Design, Engineering and Manufacturing: System Techniques and Applications" *System Techniques and Computational Methods-CRC Press*, Volume 1.
- [17] Liao Ching Jong, Lin Hung (2003), "A case study in a two-stage hybrid flow shop with setup time and dedicated machines" *International Journal of Production Economics*, Vol. 86, Page 133–143.
- [18] Michalweiz, Z. (1999), "Genetic Algorithm+ Data Structures=Evolution Programs", Springer USA.
- [19] Marimuthua, S., Ponnambalam, S.G., Jawahar, N. (2006), "Evolutionary algorithms for scheduling m-machine flow shop with lot streaming" *Robotics and Computer-Integrated Manufacturing*, Vol. 24, Page 125–139.
- [20] Mitchell, M. (2002), "An Introduction to Genetic Algorithm", Prentice Hall, New Delhi, India.
- [21] Man, K. F., Tang, K. S, Kwong, S. (1996): "Genetic Algorithms: Concepts and Applications" *IEEE Transactions on Industrial Electronics*, VOL. 43(5).
- [22] Nasser Aljaber, Cheun-Lung Chen, Venkateswara S., Vempati (1993), "An application of genetic algorithms for flow shop problems" *European Journal of Operational Research*, Vol. 80, Page 389-396.
- [23] Pinedo, M.L. (2005), "Planning and scheduling in manufacturing and services". Springer Science + Business Media, Inc., New York.
- [24] Pranzo Marco (2003), "Batch scheduling in a two-machine flow shop with limited buffer and sequence independent setup times and removal times" *European Journal of Operational Research*, Vol. 153, Page 581–592.
- [25] Ponnambalam, S.G., Marimuthu, S. (2005), "Heuristic search algorithms for lot streaming in a two-machine flowshop" *Int J AdvManufTechnol* 27: 174–180.
- [26] PascualCompanys R. (2003), "Secuenciación" Vol. 1- CPDA, ETSEIB, Barcelona
- [27] Ruiz Rube'n, Nagano Marcelo Seido, Luiz Antonio Nogueira Lorena (2008), "A Constructive Genetic Algorithm for Permutation Flowshop Scheduling" *Computers & Industrial Engineering*, Vol. 55, Page 195–207.
- [28] Rajasekaran, S., Vijaya, Lakshmi G.A. (2004), "Neural Networks, Fuzzy Logic, and Genetic Algorithms, Synthesis and Applications". Prentice Hall, New Delhi, India.
- [29] Ruiz Rube'n, MarotoConcepcio'n (2005), "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility" *European Journal of Operational Research*, Vol. 169, Page 781–800.
- [30] Shiau Der-Fang, Cheng Shu-Chen, Yueh-Min Huang (2008), "Proportionate flexible flow shop scheduling via a hybrid constructive genetic algorithm" *Science Direct Expert Systems with Applications*, Vol. 34, Page 1133–1143.
- [31] Seido Marcelo Nagano, Rube'n Ruiz, Luiz Antonio Nogueira Lorena (2008), "A Constructive Genetic Algorithm for permutation flowshop scheduling" *Computers & Industrial Engineering*, Vol. 55, Page 195–207.
- [32] Sun, J.U. (1998), "A genetic algorithm for a re-entrant flow-shop scheduling problem with sequence-dependent setup times" *Engineering Optimization* Vol. 41(6), Page 505–520.
- [33] Varela Ramiro, Camino, R., Jorge Puente, Alberto Gomez, Ana, M. (2001), "Solving flow-Shop Scheduling Problems by Means of Genetic Algorithms" *Production Planning & Control*, VOL. 12(6), Page 560- 574.
- [34] Wang Ling, Zhang Liang, Zheng Da-Zhong (2005): "An adaptive genetic algorithm with multiple operators for flowshop scheduling" *Int J AdvManufTechnol*, Vol. 27, Page 580–587.
- [35] Yaurima Victor, Burtseva Larisa, Tchernykh Andrei (2008), "Hybrid Flowshop with Unrelated Machines, Sequence Dependent Setup Time and Availability Constraints: An Enhanced Crossover Operator for a Genetic Algorithm" *Production Planning and Advanced Manufacturing*, Page 608-617.
- [36] Yamada Takeshi, Reeves Colin R. (1998), "Genetic Algorithms, Path Relinking, and the Flowshop Sequencing Problem" *Evolutionary Computation*, Vol. 6(1), Page 45-60.
- [37] Zhang Yi, Xiaoping Li, Wang Qian (2009), "Hybrid Genetic Algorithm for Permutation Flowshop Scheduling problems with total Flowtime Minimization" *European Journal of Operational Research*, Vol. 196, Page 869–876.
- [38] Zhang Liang, Wang Ling (2005), "Determining optimal combination of genetic operators for flow shop scheduling" *Int J AdvManufTechnol*, Vol. 30, Page 302–308.
- [39] Zhang Yong, Hong Li Yin, Wang Jiang (2009), "Genetic algorithm with new encoding scheme for flow shop scheduling" *Int J AdvManufTechnol* Vol. 44, Page 977–984.
- [40] Zalinda Othman, KhairanumSubari, NorhashimahMorad (2004), "flow Shop scheduling with alternative machines using genetic algorithms" *Journal Teknologi*, Vol. 41,Page 67–78.