Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 9 (1): 97-104, 2009
ISSN 1812-5654

A Genetic Algorithm for Scheduling Flexible Manufacturing Cells

[1]M.T. Taghavifard, [2]M. Heydar and [2]S.S. Mousavi
[1]Faculty of Industrial Management, Allameh Tabataba'i University, Tehran, Iran
[2]Member of Young Research Club, Department of Industrial Engineering,
Graduate School of Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran

Abstract: In this study, scheduling of Flexible Manufacturing Cells (FMC) is taken into consideration. This type of production system combines the merit of job shop and flow shop production systems. FMS Scheduling belongs to the class of problems that are known as NP-hard. This study presents a genetic algorithm-based technique to schedule machines and Automated Guided Vehicle (AGV), simultaneously. To generate schedules from a given chromosome, four Priority Dispatching Rules (PDR) are considered. Maximum completion time or makespan is defined as the objective function. The algorithm was coded and many randomly generated problems were solved. The obtained results were compared with optimum values obtained from the most comprehensive mathematical formulation in the literature. The experimental results show that the proposed method performs well in terms of efficiency and quality of solutions. For further study, the researchers will consider this problem in multi-objective environment.

Key words: FMC scheduling, flexible manufacturing systems, genetic algorithm, makespan

INTRODUCTION

Flexibility is a key concept in the management of modern manufacturing systems. The principal motivation is to achieve rapid response to costumer demands by improving the efficiency of a job shop while retaining its flexibility (Blaÿzewicz *et al*., 2007; Groover, 2007). To achieve this goal the term Flexible Manufacturing System (FMS) is defined. Flexible manufacturing systems have many potential advantages including high flexibility, high machine utilization, low work-in-process inventory and is an unsupervised production system. Scheduling is in the heart of this control system and therefore plays a crucial role to achieve intended goals (Liu and MacCarthy, 1997; Groover, 2007). FMS combines the merits of job shop and flow shop production systems. The high level of automation previously reserved for mass production is now also achievable for medium-sized production and the manufacturing flexibility enables companies to react quickly to changes in customer demand (Jerald *et al*., 2006). FMS is an integrated computer controlled complex of automated material handling devices and Numerically Controlled (NC) machine tools that can process medium-sized volumes of a variety of part types (Liu and MacCarthy, 1997; Groover, 2007; Tung *et al*., 1999; Noorul Haq *et al*., 2003). Scheduling FMS problems are more difficult than the conventional production systems. This is because of a number of reasons such as machine setup times, part routing and operations scheduling. Besides, there are resources other than machines to be considered. These resources are material handling devices, buffer storages and tool magazines. FMS scheduling problems are proved to be NP-hard and mathematical programming approaches need to be better suited and improved for real-world FMS scheduling problems (Liu and MacCarthy, 1997; Sankar *et al*., 2003; Kim *et al*., 2004).

Therefore, the success of an FMS lies in the design of an appropriate scheduling procedure that optimizes the performance measures of such a system. As many operational problems are directly linked to scheduling problems, the design of appropriate scheduling mechanisms for FMS is of equal importance to the design of FMS itself. The scheduling problems in FMS are related to the execution of production orders and include raw part input sequencing, machine and vehicle scheduling, monitoring system performance and taking the necessary corrective actions (Chan and Chan, 2001).

In this study, scheduling problem in one special configuration of FMS known as Flexible Manufacturing Cell (FMC) is considered. An FMC consists of a set of Single Flexible Machines (SFM) and only one material handling device that can be used when it is idle (Liu and McCarthy, 1997; Maccarthy and Liu, 1993a, b) and the whole system is under computer control. Moreover, different from the earlier research study, this study focuses on both machine and AGV.

Corresponding Author: Mojtaba Heydar, No. 29, North Iranshahr Street, P.O. Box 11365/4435, Tehran, Iran
Tel: +98 912 6938359 Fax: +98 262 3479

Scheduling of flexible manufacturing systems received enormous attention over the last three decades. Three different approaches are mainly used. One approach is mathematical programming formulation (Liu and McCarthy, 1997; Choi and Lee, 2004). Liu and MacCarthy (1997) developed a comprehensive global MILP for the class of FMSs known as flexible manufacturing cells, or in short, FMCs. The proposed model considers both aspects, of a scheduling procedure, i.e., loading and sequencing and scheduling the machines, material handling systems and storages. Three objective functions including mean completion time, maximum completion time or makespan and maximum tardiness are defined. Based on the model, a global heuristic procedure is described. The results show that the optimality performance of the global heuristic procedure is much better than loading and then sequencing approach. The main disadvantage of this model, as is the case for large scale industrial problems, is the time required to solve the problem as its size increases and therefore recently developed methods must be used.

Ahluwalia and Ping (1991) proposed a distributed approach to job scheduling in an FMS environment. It is assumed that each machine is equipped with a general purpose computer that controls it during its processing function. Each machine is represented as a node that is capable of communicating with other nodes (machine tools) through its assigned computer. This system is formulated as a linear programming model to solve the scheduling problem. When the system malfunctions, the job rescheduling is based on a non-linear programming model. Results show this approach frees up the main processor for other tasks and is well suited for a large and complex manufacturing system.

Jiang and Hsiao (1994) proposed a new mathematical programming for scheduling an FMS. In their model, operation scheduling and part routing with alternative plans are considered. They presented two models, models A and B, with different objectives. These objectives are absolute deviation of meeting due dates and the minimum of total completion time, respectively.

The second approach is heuristics, dispatching rules and simulation which are very common in practice. Sabuncuoglu and Hommertzheim (1992) proposed a dynamic dispatching rule for on-line simultaneous scheduling of machines and AGVs in an FMS. This dispatching rule uses various priority rules based on the status of jobs. Using this information, decisions are made hierarchically to identify the appropriate part and machine to be served. Mean flow time and mean tardiness are regarded as objective functions. Simulation results indicate that their approach outperforms existing scheduling rules for a number of experimental conditions.

Sridharan and Babu (1998) applied simulation technique to made multi-level decisions for FMS scheduling problem. Then, the results of this simulation model have been used for developing a meta-model which investigates how accurate these results are. They finally concluded that these meta-models are useful for FMS under study so as to evaluate various multi-level scheduling decisions in FMS.

Sabuncuoglu and Karabuk (1998) presented a heuristic algorithm based on the filtered beam search for scheduling flexible manufacturing systems. The main assumptions considered are buffer capacity and routing and sequence flexibility that is used in generating schedules for machines and AGVs. The performance criteria are mean flow time, mean tardiness and makespan. To further explore algorithm efficiency, statistical experiments were designed which shows considerable improvements in system performance.

Chan and Chan (2001) conducted a simulation modeling study on a flexible manufacturing system which minimizes three performance criteria simultaneously, i.e., mean flow time, mean tardiness and mean earliness. They used priority dispatching rules that frequently changed according to the system status. To monitor criteria, three indices were used. These indices, then, were ranked in descending order showing how worse the system condition is. In such case an appropriate rule will be selected to tackle that criterion with largest index. This mechanism is called pre-emptive. Results show that a solution (range of frequency) can always be obtained for changing the dispatching rule, so that the system is better than one which just uses fixed FMS scheduling rules.

The third and last approach is based on artificial intelligence techniques. This includes meta-heuristics, neural networks, fuzzy logic and expert systems. Ulsoy *et al.* (1997) proposed a GA-based approach to schedule jobs and AGVs concurrently in an FMS. Their study is worth considering since a new chromosome representation is used. Another aspect of this GA is the crossover operator that is used for the first time.

Logendran and Sonthinen (1997) presented a tabu search-based approach for the job-shop type flexible manufacturing systems. First, a mixed integer programming is developed and then a strong heuristic algorithm based on the concept known as tabu search is developed to tackle problems of industrial merit. For this, they introduced six different versions of proposed algorithm. To measure the performance of this tabu search-based heuristic, a randomized complete block design is experimented.

Jerald *et al.* (2006) have considered two major resources in FMS, i.e., machine and AGV and developed a genetic algorithm called Adaptive Genetic Algorithm.

The objective function is combined from two parts. The first part minimizes penalty cost and the second minimizes machine idle time. These two aspects, to some extent, are interconnected. In other words, if AGV is properly scheduled, then the idle time of machines can be minimized; as such, their utilizations can be maximized. The penalty cost part of the objective function minimizes not meeting committed due dates.

Noorul Haq *et al*. (2003) proposed a multi level scheduling for FMS to generate realistic schedules for the efficient operation of the FMS. Other resources than machines such as material handling device, AS/AR and tool management is considered. To generate schedules, combined a heuristic method, namely Giffler and Thompson (Sakawa, 2001; Baker, 1974) is combined with GA and Simulated Annealing (SA).

Reddy and Rao (2006) developed a hybrid multi-objective GA for scheduling machines and AGV in an FMS concurrently. Three objectives or criteria are considered including makespan, mean flow time and man tardiness. The proposed HGA is combined with a heuristic approach that is used to schedule AGV. As researchers stated, this type of hybridization is capable to reduce the size of strings and the number of constraints and as such, increase algorithm efficiency. Initial population is randomly generated.

Kim *et al.* (2004) proposed a new GA called network-based genetic algorithm for scheduling jobs in FMSs. A static environment is modeled which in scheduling literature means that all jobs are ready or ready time is zero. A mathematical programming with minimization makespan, total flow time and total tardiness as objective functions is presented. Then a network-based hybrid GA combined with a neighborhood search procedure was developed. As numerical experiments show, this algorithm is both effective and efficient for FMS scheduling.

In many real cases there are more than one objective that should be considered simultaneously (Sankar *et al*., 2003; Kim *et al*., 2004; Reddy and Rao, 2006). Prabaharan *et al.* (2006) considers sequencing and scheduling of jobs and tools in a flexible manufacturing cell. To achieve this, two methodologies are used to derive optimal solutions. The first method is commonly used is Priority Dispatching Rules (PDRA) and the second one is Simulated Annealing Algorithm (SAA). One aspect of their proposed algorithm is the use of Giffler and Thompson procedure for active feasible schedule generation. The performance of these two algorithms are compared with makespan and computational time. The analysis reveals that the SAA based heuristic provides an optimal or near optimal solution with reasonable computational time.

Tung *et al*. (1999) presented a hierarchical approach to scheduling Flexible Manufacturing Systems (FMSs) that pursues multiple performance objectives and considers the process flexibility of incorporating alternative process plans and resources for the required operations. In so doing, they proposed a multi-objective priority index that simultaneously considers order tardiness cost, inventory cost, order profit, processing time, due date and order size. Using the just mentioned multi-objective priority index, rough-cut schedules will be generated and evaluated for performance measure at the system level.

FLEXIBLE MANUFACTURING CELLS SCHEDULING

Here, scheduling problem in an FMC environment is described. In so doing, model assumptions and definitions are presented. Moreover, a genetic algorithm-based technique is proposed to solve the scheduling problem in a FMC. This GA differs from existing GA-based techniques in various ways. One major difference is scheduling generator phase that combines four priority dispatching rules, each of which handles one type of resources or constraints inherent in flexible manufacturing cells. In the following, first, model assumptions are presented and, then, the elements of proposed GA are fully described.

Model assumptions and definition: Here, assumptions, based on which under-consideration problem is stated and solved, will be presented. In what follows these assumptions are outlined:

- Processing time of each operation is known in advance. The problem is considered in a static environment
- Transportation times between machines are based on the AGV speed and distance between two different machines
- Loading and unloading times are negligible and can be eliminated
- Setup times in this model are sequence-dependent
- Machines and AGV breakdown are not accounted for
- All machines can process every part and related operations, only if equipped with appropriate tools
- Tooling constraints are not considered. In other words, the considered resources are machines, material handling device and buffers
- Each machine can process only one part at a time
- Preemption is not allowed

- Processing times are scheduling-independent but machine-dependent, i.e., machine eligibility is taken into account. This assumption is considered just here for the first time
- Technological constraints are known a priory
- There are two buffers before and after each machine
- To avoid system dead lock, it is assumed that there is a central buffer with unlimited capacity to keep in-line parts

Based on the above-mentioned assumptions, with some limitations to some extend, Liu and MacCarthy (1997) developed a very comprehensive mathematical programming formulation with seven sets of constraints, each showing one aspect of the flexible manufacturing cell.

In this study a GA-based technique is proposed and developed to generate an (near) optimal solution for scheduling problem. Minimizing makespan or C_{max} is defined as objective function.

Genetic algorithm technique: In this part the proposed GA is outlined. Genetic algorithms are non-deterministic stochastic search methods that utilize the theories of evolution and natural selection to solve a problem within a complex solution space, or more specifically combinatorial optimization problems (Sakawa, 2001; Gen and Cheng, 2000; Gen and Cheng, 1997). The element and mechanism of genetic algorithms are representation, population, evaluation, selection, operator and parameter. The algorithm starts with a randomly generated initial population consisting of sets of chromosomes that represent the solution of the problem. These are evaluated for the fitness function, or equivalently objective function and then selected according to their fitness value. The elements of the proposed GA are explained hereafter.

Representation: Every solution of the problem has equivalent representation in GA domain. To link each solution to a chromosome, a coding scheme is needed. In this study each solution is coded as string of integer numbers (Reddy and Rao, 2006), which is called pheno style (Sankar *et al*., 2003). Care must be taken in generating feasible solution that maintains the precedence relations of operations related to the same job. This is crucial in job shop-based scheduling. The following example shows how this scheme works.

Example: A scheduling situation with 4 work centers and 3 work pieces is considered. There are 10 operations and the chromosomes consist of 10 genes.

Jobs	1				2			3		
Oper.	1	2	3	4	1	2	3	1	2	3
Machine	1	3	4	2	2	3	1	1	4	3
Representation	1	2	3	4	5	6	7	8	9	10

Fitness function: Each individual generated is evaluated for its completion time. The makespan, then, is the maximum of jobs completion times. Mathematically, if completion time is defined as:

$$C_i = \sum_j O_{ij} \tag{1}$$

Makespan would be

$$C_{max} = \max(C_1, \ldots, C_n) \tag{2}$$

Another aspect of GA is operators that play a major role in finding (near -) optimal solution. There are three operators: reproduction or selection, crossover and mutation (Reddy and Rao, 2006).

Crossover: The technique used here to cross over two chromosomes is named job-based crossover which never violates precedence relations between operations (Reddy and Rao, 2006; Gen and Cheng, 2000; Gen and Cheng, 1997). Based on this scheme, once two chromosomes are selected as parents, a job is randomly selected and its corresponding operations are directly copied into respective positions of offspring. This method guarantees that precedence relations are not violated. Then, the remaining unfilled positions are fulfilled with operations of another parent. The example below clarifies the above method.

Example: Chromosomes selected for crossover are P1: 1 5 8 2 6 9 3 7 10 4 and P2: 5 8 1 9 2 3 6 10 7 4.

Let the job selected be 2 and the corresponding operations of job 2 are 5, 6 and 7.

P1: 1 5 8 2 6 9 3 7 10 4
P2: 5 8 1 9 2 3 6 10 7 4

Resulting offspring are:

OS1: 5 1 8 2 9 3 6 10 7 4
OS2: 8 5 1 9 6 2 3 7 10 4

Mutation: Operation swap mutation is used. Two random positions on the chromosome are chosen and the operations associated with these positions are swapped. Operation swap mutation may cause infeasibilities in terms of the precedence relations and a repair function is used to eliminate any such infeasibility (Sankar *et al*., 2003; Reddy and Rao, 2006).

Repair function: A repair function is used to see that the chromosomes do not violate the precedence constraints (Ulsoy *et al*., 1997). The four-step procedure below outlines the repair function in details:

Step 1: Find positions of the operations that violate the precedence relations
Step 2: Compute the distance between violating operations
Step 3: If the distance between them is less than half the chromosome length then swap the operations, else go to step four
Step 4: Randomly pick any one operation and insert it before or after the other depending on the precedence

Selection: The method used here is known as roulette wheel approach that commonly used in practice (Gen and Cheng, 2000). It belongs to the fitness-proportional selection and can select a new population with respect to the probability distribution based on fitness values, i.e., the more fitted a chromosome is, the more chance it has to be selected.

Population and parameters: The initial population is randomly generated. The number of chromosomes in each generation, crossover and mutation rates, number of generation that algorithm should run to give a satisfying solution are considered as GA parameters that must be initialized at the beginning of GA run.

SCHEDULE GENERATOR

Apart from GA methodology, to evaluate each string or solution it is needed to schedule jobs on different machines considering problem constraints. In so doing, four Priority Dispatching Rules (PDR) are combined. These include Earliest Completion Finishing Time (EFT), SPT, Shortest Distance Time (SDT) and Fewest Waiting Jobs for Machine (FWJM). This proposed methodology work as follows: first, a job with earliest finishing time is selected to be processed on the corresponding machine. If there is more than one job, the job with shortest processing time for its subsequent operation is selected. Then tie is broken by considering the distance each job should travel, i.e., the shortest path is selected first by AGV. If again there is a tie, another PDR is taken into account. Based on this rule, the number of jobs in the target machine buffer determines which job should go first.

This GA in conjunction with proposed heuristic approach constructs the methodology presented for scheduling jobs and AGV in a flexible manufacturing cell.

Ga steps for scheduling fmc: Here, steps for scheduling a flexible manufacturing cell are presented.

Step 1: Enter input data including number of machines, distance between machines, number of jobs and corresponding operations, processing and setup times and due dates. Enter GA parameters such as population size, crossover and mutation rates and termination criteria
Step 2: Randomly generate an initial population using the encoding scheme
Step 3: Generate schedules using schedule-generator module
Step 4: Using roulette wheel approach select chromosomes to create mating pool for next generation
Step 5: Generate offspring population using job-based crossover and bit-wise exchange mutation operators. If some precedence relations are violated, go to step 6; otherwise go to step 7
Step 6: In case of any violation as a mutation result, run repair function as described above and go to step 7
Step 7: Evaluate each chromosome in current population for objective function based on the generated schedule
Step 8: Sort chromosomes based on the fitness function value
Step 9: If termination criterion is satisfied, then stop and print the fittest chromosome as the best solution found; otherwise go to step 4 for next generation

NUMERICAL EXAMPLE

In this part, the proposed approach is applied to schedule FMC with varying parameters. The proposed algorithm is coded in Visual C++ 6. Many problems with different parameters and values were considered and solved. The results are tabulated. Since the problem environment is somehow similar to the one considered by Liu and MacCarthy (1997), the MILP was solved by Lingo 8 and the results were compared with those of the heuristic approach.

First, 10 problem examples were randomly generated and shown in Table 1. In each problem example, different job sets with different operations were considered. Based on the mathematical formulation number of variables and constraints are also calculated and provided in this Table 1.

To further study the efficiency of proposed model, four scenarios were defined, based on which both mathematical model and GA methodology are applied. These scenarios consider one parameter that may affect the results.

Table 1: Problem data for experimental study

Problem No.	Jobs	Operations per job	Total operations	Machines	No. of variables	No. of constraints
1	4	2	8	2	208	596
2	6	2	12	2	632	985
3	6	6	36	2	5336	5461
4	7	6	42	2	7233	6308
5	10	3	30	2	3732	4466
6	9	6	54	2	11891	10648
7	10	6	60	2	14652	13241
8	15	3	45	2	8297	10186
9	15	6	90	2	32777	30436
10	20	6	120	2	58102	54681

Table 2: Results for case 1

Problem	Global solution			GA	
No.	Time (sec)	Iteration	C_{max}	Time (sec)	C_{max}
1	8	8734	103.5	30	108.7
2	17	23075	105.5	67	114.0
3	197	381173	48.0	150	49.9
4	623	965846	18.0	400	18.5
5	1987	2800953	23.0	964	24.0
6	3674	3591046	50.0	1630	53.5
7	6743	7541197	190.0	3180	205.0
8	10863	11311795	278.0	5342	297.0
9	15642	18664462	359.0	8334	384.0
10	24651	35462777	406.0	10045	434.0

Table 3: Results for case 2

Problem	Global solution			GA	
No.	Time (sec)	Iteration	C_{max}	Time (sec)	C_{max}
1	8	13998	103.5	20	108.7
2	30	72299	105.5	100	109.7
3	547	1242305	120.0	600	127.0
4	2160	5977290	23.0	1012	25.0
5	4982	16543810	31.0	1807	33.0
6	10030	24815716	69.0	3481	73.0
7	17642	52113003	225.0	5690	245.0
8	26071	96409055	301.0	8519	316.0
9	-	-	-	11047	389.0
10	-	-	-	13609	490.0

Table 4: Result for case 3

Problem	Global Solution			GA	
No.	Time (sec)	Iteration	C_{max}	Time (sec)	C_{max}
1	10	17234	157.8	45	170
2	110	380560	197.8	150	208
3	615	1376554	54.0	700	56
4	3185	7764906	23.0	1285	25
5	7348	14753321	57.0	2059	61
6	15235	32457307	109.0	3942	117
7	29841	63291749	300.0	7002	321
8	-	-	-	9358	400
9	-	-	-	11803	463
10	-	-	-	14391	513

First, an FMC with two machines is considered and iteratively problem size is increased by adding job with varying operations. Processing times are randomly generated from a uniform distribution function (Table 2). Then, the configurations of FMC, in terms of distance between machines or case 2 (Table 3), buffer size or case 3 (Table 4) and speed of AGV or case four (Table 5), were changed. GA parameters were remained unchanged, though their impacts on algorithm performance can be effective.

This algorithm is coded in Visual C++ 6 along with coded MILP model in Lingo 8. Both were run on a PC with 2.6 GHz CPU and the results are tabulated in the following page.

These 4 cases or scenarios for mathematical model and GA approach are depicted in (Fig. 1, 2), respectively. Figure 1 and 2 depicts and compares four scenarios. It can be seen from these figures that how the configuration and layout of manufacturing cell can increase the problem complexity.

Table 5: Results for case 4

Problem	Global solution			GA	
No.	Time (sec)	Iteration	C_{max}	Time (sec)	C_{max}
1	12	38337	157.8	45	162.5
2	121	406834	197.8	155	205.7
3	774	1648553	120.0	710	126.0
4	3352	8709542	140.0	1315	149.0
5	7853	18725515	156.0	2129	165.0
6	16292	35391234	290.0	4519	311.0
7	30820	63704203	423.0	7132	439.0
8	-	-	-	9680	412.0
9	-	-	-	11923	489.0
10	-	-	-	14571	534.0

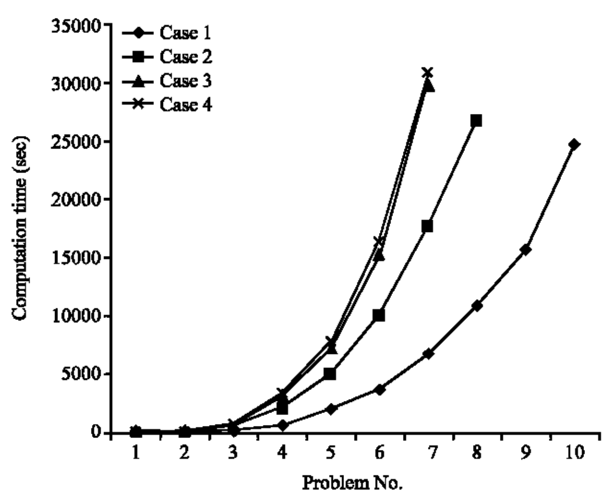

Fig. 1: Computational time needed to solve problem by B and B

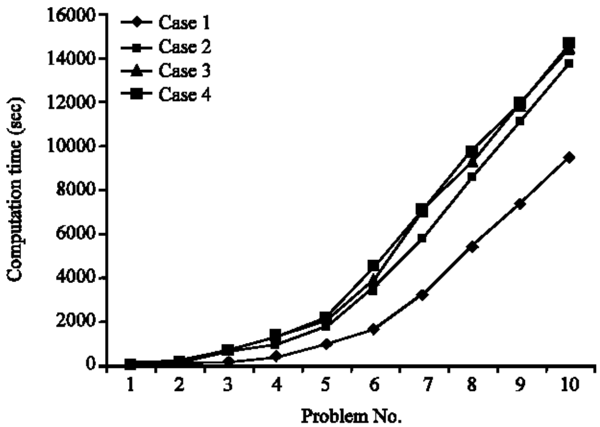

Fig. 2: Computational time for GA

Results: As results show, using GA to solve this problem will reduce time needed to get best objective function dramatically showing that using this technique is promising. Another fact that is worth mentioning is the impact that FMS layout has on production planning in general and scheduling in particular, i.e., the more machines were located away, the greater the makespan is. For this reason layout of a cell must be considered in scheduling system design.

CONCLUSION AND FURTHER RESEARCH

Flexibility is a growing issue in modern industrial firms to respond varying product demand with short lifecycle. Therefore, new approaches are needed to resolve this issue. Since FMS scheduling problems are NP-hard, using heuristic methods are quite justified. In this study a class of FMS known as flexible manufacturing cell is considered. A new GA-based approach is proposed to schedule jobs and AGV for minimizing makespan.

The algorithm is coded in Visual C++ 6 and run for problems of different sizes. The obtained results were compared with mathematical model developed by Liu and MacCarthy (1997). As results show, the proposed model performs better than MILP model. One reason that is worth considering is the required time to solve medium to large size problems that is a crucial issue in industrial firms.

There are several directions to study on for future study and some are suggested here. One aspect of every decision problem, as in the case of scheduling problem, is multiple objectives that must be considered simultaneously. So, it is worth considering more than one objective as the measures of system performance. Another way is to apply other heuristic methods separately or in conjunction with GA.

Currently researchers are working on this problem with hybrid GA as a methodology in multi-objective environment.

REFERENCES

Ahluwalia, R.S. and J. Ping, 1991. A distributed approach to job scheduling in a flexible manufacturing system. Comput. Ind. Eng., 20: 95-103.

Baker, K.R., 1974. Introduction to Sequencing and Scheduling. 1st Edn. John Wiley and Sons, New York, pp: 318 ISBN-10: 0471045551.

Blaÿzewicz, J., K. H. Ecker, E. Pesch, G. Schmidt and J. Weglarz, 2007. Handbook of Scheduling: From Theory to Applications. 1st Edn., Springer, Berlin.

Chan, F.T.S. and H.K. Chan, 2001. Dynamic scheduling for a flexible manufacturing system the pre-emptive approach. Int. J. Adv. Manufac. Technol., 17: 760-768.

Choi, S.H. and J.S.L. Lee, 2004. A sequencing algorithm for makespan minimization in FMS. J. Manufac. Technol. Manage., 15: 291-297.

Gen, M. and R. Cheng, 1997. Genetic Algorithms and Engineering Design. 1st Edn., John Wiley and Sons, New York, ISBN: 0-471-12741-8.

Gen, M. and R. Cheng, 2000. Genetic Algorithms and Engineering Optimization. 1st Edn., Wiley and Sons, New York, ISBN: 0471315311.

Groover, M.P., 2007. Automation, Production Systems, and Computer-Integrated Manufacturing. 3rd Edn., Prentice Hall Inc., Prentice, ISBN: 0132393212.

Jerald, J., P. Asokan, R. Saranavan and A.D.C. Rani, 2006. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. Int. J. Adv. Manuf. Technol., 29: 584-589.

Jiang, J. and W.C. Hsiao, 1994. Mathematical programming for the scheduling problem with alternate process plans in FMS. Comput. Ind. Eng., 27: 15-18.

Kim, K.W., G. Yamazaki, L. Lin and M. Gen, 2004. Network-based hybrid genetic algorithm for scheduling in FMS environments. Artif Life Robotics, 8: 67-76.

Liu, J. and B.L. MacCarthy, 1997. A global MILP Model for FMS Scheduling. Eur. J. Oper. Res., 100: 441-453.

Logendran, R. and A. Sonthinen, 1997. A tabu search-based approach for scheduling job-shop type flexible manufacturing systems. J. Oper. Res. Soc., 48: 264-277.

Maccarthy, B.L. and J. Liu, 1993a. A new classification for flexible manufacturing systems. Int. J. Prod. Res., 31: 299-309.

Maccarthy, B.L. and J. Liu, 1993b. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. Int. J. Prod. Res., 31: 59-79.

Noorul Haq, A., T. Karthikeyan and M. Dinesh, 2003. Scheduling decision in FMS using a heuristic approach. Int. J. Adv. Manufac. Technol., 22: 374-379.

Prabaharan, T., P.R. Nakkeeran and N. Jawahar, 2006. Sequencing and scheduling of job and tool in a flexible manufacturing cell. Int. J. Adv. Manufac. Technol., 29: 729-745.

Reddy, B.S.P. and C.S.P. Rao, 2006. A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. Int. J. Adv. Manufac. Technol., 31: 602-613.

Sabuncuoglu, I. and D.L. Hommertzheim, 1992. Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system. Int. J. Prod. Res., 30: 1059-1079.

Sabuncuoglu, I. and S. Karabuk, 1998. A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. IIE. Trans., 30: 179-191.

Sakawa, M., 2001. Genetic Algorithms and Fuzzy Multi-Objective Optimization. 1st Edn., Kluwer Academic Publisher, Boston, ISBN: 0-7923-7452-5.

Sankar, S.S., S.G. Ponnambalam and C. Rajendran, 2003. A multi-objective genetic algorithm for scheduling a flexible manufacturing system. Int. J. Adv. Manufac. Technol., 22: 229-236.

Sridharan, R. and A.S. Babu, 1998. Multi-level scheduling decisions in a class of FMS using simulation based metamodels. J. Oper. Res. Soc., 49: 591-602.

Tung, L.F., L. Lin and R. Nagi, 1999.. Multiple-objective scheduling for the hierarchical control of flexible manufacturing systems. Int. J. Flexible Manufac. Sys., 11: 379-409.

Ulsoy, G., F.S. Serifoglu and U. Bilge, 1997. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput. Oper. Res., 24: 335-351.