International Journal of Physical Sciences Vol. 7(9), pp. 1461 - 1471, 23 February, 2012
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.407
ISSN 1992 - 1950 © 2012 Academic Journals

Full Length Research Paper

Performance optimization of simultaneous machine and automated guided vehicle scheduling using fuzzy logic controller based genetic algorithm

Mostafa Badakhshian*, Shamsuddin B. Sulaiman and Mohd Khairol Anuar B. Mohd Ariffin

Department of Manufacturing Engineering, Faculty of Engineering, UPM, Serdang, Selangor, 43400, Malaysia.

Accepted 26 January, 2012

The current trend in manufacturing technology is considered by two main items automation and flexibility. Flexible manufacturing system (FMS) is one of the most identified systems that include both automation and flexibility criteria. It comprises three principle elements: computer controlled machine tools, an automated material handling system and a computer control system. One of the automated materials handling equipment in FMS is automated guided vehicles (AGVs). Integrated scheduling of AGVs and machines is an essential factor contributing to the efficiency of the manufacturing system in FMS environment. Previously, genetic algorithm (GA) is considered as a heuristic method to solve AGV scheduling problem. GA may not be able to achieve the global optimum due to premature convergence because of control's lack on its operators parameters. Fuzzy logic controller (FLC) is proposed to control the behavior of GA during solving the scheduling problem of AGVs. This paper presents a job-based GA that is based on job sequencing. Through the optimization, the FLC is used to control the GA operators (crossover and mutation rate) simultaneous to solve the AGV scheduling problem.

Key words: Flexible manufacturing system, automated guided vehicle, simultaneous scheduling, genetic algorithm, fuzzy logic controller, optimization.

INTRODUCTION

Flexible manufacturing system (FMS) is an integrated computer-controlled complex arrangement of automated material handling devices and numerically controlled machine tools that can simultaneously process medium-sized volumes of a variety of part types (Stecke, 1983). A set of some numerical controlled machine centers, load/unload station and automated storage/retrieval system (AS/RS) which connecting through an automated guided vehicle (AGV) system are major constituent of FMS. The most advantages of FMS through the flexibility are dealing with machines and tool breakdowns, changes in schedule, product mix and alternative routes (Raj et al., 2007).

AGVs are among various advanced material handling techniques that are finding increasing applications in today's computer integrated manufacturing (CIM) settings. AGVs are battery-powered driverless vehicles and capable of transporting a variety of part types from point to point that is controlled and addressed by computer and move along wire guide paths (flowpath), or by magnetic or optic guidance. They can be interfaced to various other production storage equipment and controlled through an intelligent computer control system. This flexibility and compatibility make AGVs a feasible alternative to traditional material handling methods especially in flexible manufacturing environments. According to Ganesharajah et al. (1998), the first large-scale manufacturing application of an AGV system occurred in 1974 at a Volvo plant in Sweden and the largest application in North America is at the truck assembly plant of General Motors (GM) in Canada, where 1,012 AGVs transport truck engines, bodies and chassis across the 2.7 million square feet plant.

Several researchers have emphasized the importance of the material handling system for the efficiency of the overall system. Attempts to improve the AGVs' design

*Corresponding author. E-mail: m.badakhshian@gmail.com.

process are reported in literature. Transportation systems in FMSs, in general, have sufficient excess capacity to allow machines and vehicles to be scheduled quite independently. Neglecting transportation times at the tactical planning level, as well as lack of appropriate coordination between a schedule for operations and machines and a timetable for vehicle movement, can have severe consequences. An area of equal importance is that of making scheduling of AGVs an integral part of the overall scheduling activity. Obviously, an increase in the performance of the FMS would be expected because of the coordination of the machine and the material handling system during the machine-scheduling phase.

The purpose of this paper is to present a meta-heuristic hierarchical method for optimization of AGV scheduling in FMS environment. The FMS environment is the same as introduced by Reddy and Rao (2006). It includes a load/unload (L/U) station, number of machine that has known sufficient input/output buffer space that is provided at each machine, and number of AGVs that are available at L/U station which carry a single unit load at the time and move along shortest path. A genetic algorithm (GA) application that is controlled by fuzzy logic controller (FLC) and used for the AGV scheduling problem will be presented. The objective is the minimization of makespan denoted by C_{max}. The makespan is defined as the time interval between the pickup of the first part from the L/U station to the finishing time of the last operation.

LITERATURE REVIEW

Recent decades researchers have addressed the AGV design and control in FMS environment. Ganesharajah et al. (1998) reviewed some of the most important example of applying the AGVs. They showed that AGVs are applicable in a wide diversity of service and manufacturing systems. FMS performance can be increased by better synchronization and scheduling of production machines and material handling equipment. Scheduling is defined by allocating the confined resources to tasks overtime and is a determination process that is relative to operations, time, cost and other company objectives (Reddy and Rao, 2006). Scheduling of machines and other resources such as vehicles, personnel, tools etc., has been done with a certain objective to be either minimized or maximized. Egbelu and Tanchoco (1984) introduced the first simulation-based experimental studies that address the scheduling of AGVs. Nevertheless, their system is not an FMS, as such, machine scheduling is not directly considered. Sabuncuoglu and Hommertzheim (1992) developed a simulation model to test different scheduling rules. Their results indicate that scheduling AGVs is as important as scheduling machines. They propose a dynamic dispatching algorithm for scheduling machines and AGVs in an FMS. The scheduling process task is to ascertain the start/end times for the individual operations to optimize a specified performance measure. The minimization of makespan objective is the most frequently used because it is directly related to the efficient utilization of resources (Ganesharajaha et al., 1998).

Ganesharajah et al. (1998) merged various lines of research related to AGVs and considered problem arising in flow path design, fleet sizing, job and vehicle scheduling, dispatching and conflict free routing. Chen and Ho (2001) described multi-objective evolutionary optimization of FMS including machines, computers, robots and AGVs. Jawahar et al. (1998) proposed an AGV scheduling integrated with production in FMS by their heuristic algorithm. They considered an FMS that is required to process various types of job loaded at discrete points of time at different processing stations. Le-Anh and Koster (2006) comprehensively presented a review on design and control of AGVs. They addressed mostly key related to guide-path design, determining vehicle requirement, vehicle scheduling and other related options to AGV. Corréa et al. (2007) proposed a hybrid constraint programming approach and mixed integer programming approach for scheduling and routing of AGVs. They used constraint programming for scheduling and mixed integer programming for routing sub problems.

Recent years utilizing heuristic methods for AGV scheduling are noteworthy by authors that the most illustrious is GA. Ulusoy et al. (1997) addressed a GA approach to the simultaneous scheduling machine and AGVs. Abdelmaguid et al. (2004) developed hybrid GA/heuristic approach to the simultaneous scheduling of machines and AGVs. Jerald et al. (2006) presented adaptive GA for simultaneous scheduling of parts and AGVs in an FMS environment. Reddy and Rao (2006) designed a hybrid multi-objective GA for simultaneous scheduling of machine and AGVs in FMS. Subbaiah et al. (2009) addressed the problem of simultaneous scheduling of machines and two identical AGVs in an FMS to minimize makespan and mean tardiness. For solving this problem, they proposed a sheep flock heredity algorithm. Babu et al. (2009) proposed a differential evolution algorithm for scheduling of machines and AGVs. This study focuses on the optimization of scheduling of AGV integrated machines in FMS environment by using FLC-based GA.

CONFIGURATION AND OPERATION ENVIRONMENT OF FMS

Considering the aforementioned factors, the following assumptions were used to formulate the scheduling problem in FMS. The FMS environment assumed for this problem is the multi-machine FMS type, which ensures that the route and vehicle requirement or fleet sizing is fixed and distinct. However, the types and numbers of

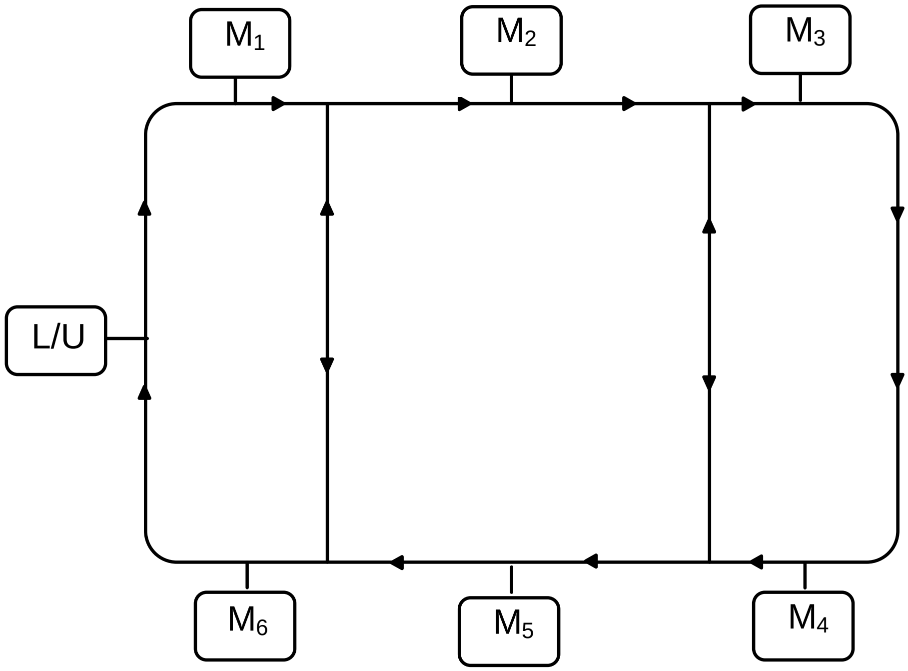

Figure 1. Sample FMS layout.

machines are known. There is sufficient input/output buffer space at each machine. AGVs transfer materials, pallets and fixtures between workstations and L/U stations. An L/U station serves as a distribution center for parts not yet processed and as a collection center for parts finished. There is sufficient input/output buffer space at the L/U station.

The AGV system needs a large number of factors to be considered in their designing and controlling problems. As the main concern of the current research is on the scheduling problem of the AGVs, some of the main factors are assume that made it facile for the author to focus on his research. The main assumptions of the simultaneous scheduling of machines and AGVs problems are:

1) No manufacturing or transportation operation can be preempted.
2) The machines can process one part at a time.
3) Each pallet and thus the AGVs can carry only one part at a time and at most, parts are allowed in the system simultaneously.
5) Transportation times of loaded and unloaded (empty) vehicles are equal.
6) The scheduling is made of assignment periods, which may have unequal time durations.

During each period, the assignment of operations to machines and vehicles to routes (pair of machines) is considered fixed. It is assume that during the scheduling, each machine must complete the assigned operations, and the vehicles must complete the associated transportation tasks determined at the machine loading/part routing level. At the beginning, all the vehicles wait in the central depot, where the central depot, loading and unloading stations are located at one place. Furthermore, number of jobs and number of operations belonging to each job is known. The number of AGVs is given and the transportation times of AGVs are known and real-time issues such as traffic control, congestion, machine failure or downtime, scraps, rework and vehicle dispatches for battery charger are ignored here and left as issues to be considered during real-time control. A typical layout of the proposed FMS environment by Reddy and Rao (2006) is shown in Figure 1. There are six machine centers, an L/U station and two AGVs handle of the material between six machine centers and L/U station. Furthermore, Table 1 shows the sample job set details, time of processing and the operations sequencing per every job in this FMS environment.

FLC BASED GA FOR AGV SCHEDULING OPTIMIZATION

John Holland and his students (1975) as artificial adaptive systems that simulate natural evolution developed GAs. As GA is able to search very large spaces effectively and efficiently, it is increasingly use to attack inherently intractable problems called NP-hard problems. A large portion of machine scheduling

Table 1. Job set details.

Job No.	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
1	3	1	1	3	2	6	4	7	6	3	5	6
2	2	8	3	5	5	10	6	10	1	10	4	4
3	3	5	4	4	6	8	1	9	2	1	5	7
4	2	5	1	5	3	5	4	3	5	8	6	9
5	3	9	2	3	5	5	6	4	1	3	4	1
6	2	3	4	3	6	9	1	10	5	4	3	1

Six jobs each with 6 operations are to be processed on 6 machines Job along with the machine number (M) and processing time (PT) are given alternatively.

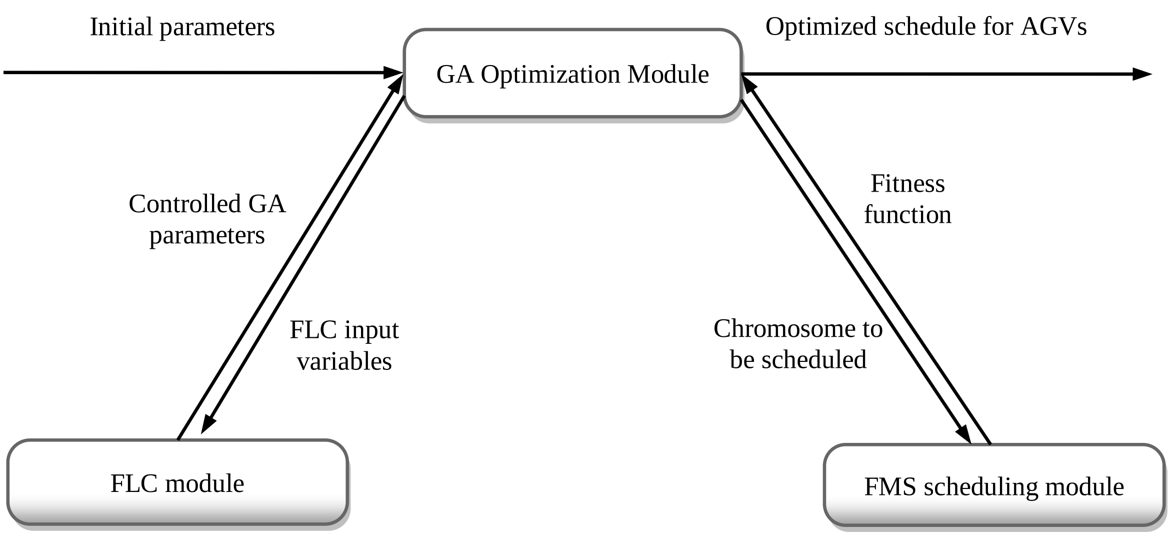

Figure 2. The essential modules for FGA method in AGV scheduling.

problems belongs to the class of NP-hard problems. Thus, optimizing routines suggesting for such problems explode rather quickly with increasing problem size. GA operates on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. GA search and find local optima through problem solving process to obtain the global optimum. When the GA cannot achieve the global optimum and sticks in local optima, premature convergence may occur (Herrera et al., 1999).

FLCs are known for their applicability on controllable systems with complicated mathematical model. The human expertise and knowledge would be useful to increase the capabilities of GA. This expertise generally is vague, incomplete or ill-structured. FLC-based GA proposes to use an FLC whose inputs are any combination of GA performance measures or current control parameters and whose outputs are GA control parameters (de Brito et al., 2006). FL and GA integration have been accomplished by two different approaches; the application of GA in optimization and search problem related fuzzy systems, and the use of fuzzy tools or fuzzy logic-based techniques for modeling different GA components or adapting GA control parameters, respectively, with the goal of improving performance. Generally, this type of integration is called fuzzy GA (FGA) (Herrera et al., 1999). Figure 2 shows the main modules of proposed methodology and their relationship.

AGVs scheduling algorithm

Jobs are scheduled based on the operation sequence derived by the GA. Initially, AGVs carry jobs from the load/unload station to the respective workstations where the first operations are scheduled. AGVs perform two types of trips, a loaded trip where it carries a load and a deadheading trip where the vehicle moves to pick up a load. Deadheading trip can start immediately after the delivery and vehicle demand at different workstations are considered and the subsequent assignments are made. If both AGVs are available, the task should be assigned to the earliest available vehicle. If no vehicle is available, compute the earliest available times of the AGVs and make the assignment. If the vehicle is idle and no job is ready, identify the operation that is going to be completed early and move the vehicle to pick up that job. This type of vehicle scheduling

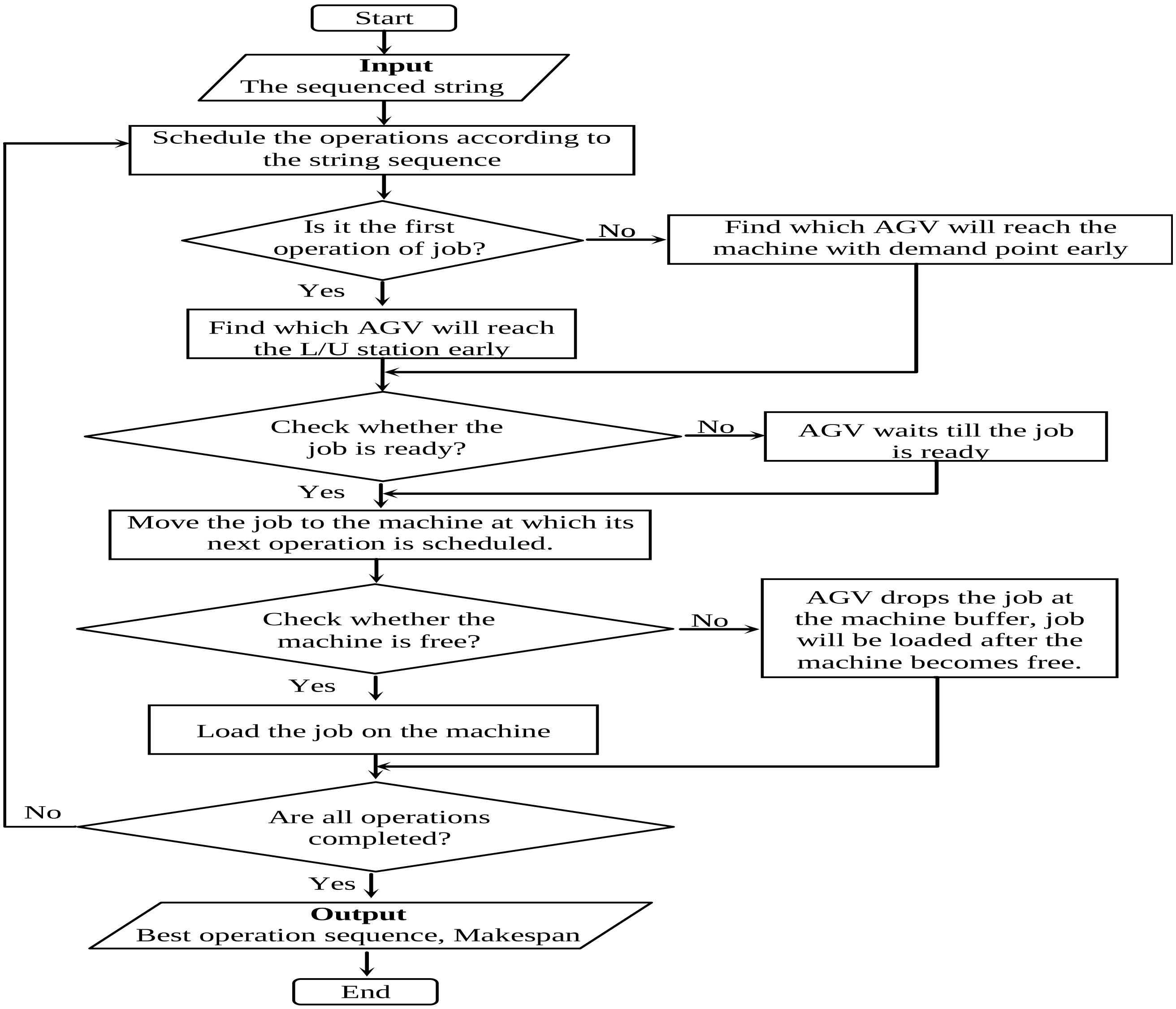

Figure 3. Simultaneous AGV and machine scheduling flow-chart (Reddy and Rao, 2006).

methodology helps in reducing the waiting times and thus, helps in improving the resource utilization and the throughput.

First schedule the operations according to the chromosome sequence, and then find which AGV will reach the L/U station or the machine with demand point early. Move the AGV from the current point to request point for its next assignment. Wait for the AGV until the job is ready if there is no job ready. Move the job to the machine at which the next operation is scheduled. If the machine is busy, AGV drops the job at the machine buffer; job will be loaded after the machine becomes free. Load the job on the machine if the machine was free. Check if all the operation is completed and the scheduling is finished; otherwise find which AGV will reach L/U station or the machine with demand point early. The following steps need to be done during the simultaneous scheduling of machines and AGVs:

1) Schedule the operations according to the sequenced string.
2) Find which AGV reaches the L/U station or the machine with demand point earlier.
3) Move the AGV from the current point to request point for its next assignment.
4) Wait the AGV until the job is ready if there is no job ready.
5) Move the job to the machine at which the next operation is scheduled.
6) If the machine is busy, AGV drops the job at the machine buffer; job will be loaded after the machine becomes free.
7) Load the job on the machine if the machine was free.
8) Check if all the operation is completed.

Figure 3 shows the prescribed scheduling flow-chart.

Job-based GA

The most important items in job-based GA are initial population, crossover operator and mutation operators. These operators should check and ignore the chromosomes that do not observe the operation sequencing in every job during the operation. Table 2 shows details on the proposed job sequence of the FMS example.

Table 2. Representation of Jobs in GA.

No. of job	No. of operation						Machines						Representation					
1	1	2	3	4	5	6	M_3	M_1	M_2	M_4	M_6	M_5	1	2	3	4	5	6
2	1	2	3	4	5	6	M_2	M_3	M_5	M_6	M_1	M_4	7	8	9	10	11	12
3	1	2	3	4	5	6	M_3	M_4	M_6	M_1	M_2	M_5	13	14	15	16	17	18
4	1	2	3	4	5	6	M_2	M_1	M_3	M_4	M_5	M_6	19	20	21	22	23	24
5	1	2	3	4	5	6	M_3	M_2	M_5	M_6	M_1	M_4	25	26	27	28	29	30
6	1	2	3	4	5	6	M_2	M_4	M_6	M_1	M_5	M_3	31	32	33	34	35	36

Accepted chromosome

13,1,7,19,8,9,25,26,14,15,27,16,28,2,3,31,20,10,4,21,17,11,5,22,29,30,23,32,33,34,18,6,24,35,12,36

Denied chromosome

13,1,7,19,8,9,25,26,14,15,27,16,28,2,3,31,20,10,4,21,18,11,5,22,29,30,23,32,33,34,17,6,24,35,12,36

Figure 4. Denied and accepted chromosomes.

Permutation representing method is implemented in current research, for example, operation number four of second job operated on sixth machine is represented as 10 in chromosome string.

In initial population, the chromosomes are composed considering the sequence of operation in each job. Hence, the chromosomes, which do not observe the sequence of operations, should be ignored. For example, operation six of third job cannot appeare prior to operation five of third job. The accepted and denied chromosome in operation sequencing of third job is shown in Figure 4. The number seventeen that presents the operation five of third job cannot appear after number eighteen that present the operation six of third job. The initial population of proposed GA is constructed through checking the sequence of operation during the generating of initial population, which means make the genes in the chromosome one by one by observing the sequence of the operation in jabs. The first position in chromosome should be 1, 7, 13, 19, 25 or 31 randomly. If number seven occupies the first position, then second position can be filled up among 1, 8, 13, 19, 25 or 31. The remaining positions of the chromosome should be filled using the same method.

Makespan or the production completion time of all jobs (produced integrated and simultaneously) is one of the items that evaluate the FMS performance. In the proposed GA, the makespan is evaluated in each generation as fitness function. The makespan value for each chromosome, C_i for i =1 to n, where n is the number of jobs for the time period of scheduling horizon should be calculated. For each operation of the job, two main periods are considered; the traveling times and operation processing time. To calculate C_i, the transportation and processing time of all the operations of the job need to be calculated. The calculation of each job completion time is calculated where $T_{i,j}$ is the transportation time required for the operations of the job and $P_{i,j}$ denotes the processing time of the operation on the corresponding machine. Summation of the aforementioned time provides the C_i for i^{th} job. The greatest value among the C_i , i = 1 to n, denoted the makespan.

Operation completion time = Oij = Tij + Pij

j^{th} operation i^{th} job (traveling time + operation processing time)

Job completion time $C_i = \sum_{i=1}^{n} Oij$

Makespan = Max (C1; C2; C3; . . . Cn)

The crossover is introduced as the main operator of the GA, which is in charge of exploitation of the search space. Two parents are selected through crossover operator and some of the genes of each parent mixed with the other parent to create two new offspring. Job-based crossover is used which never offends the precedence constraints. A job is selected randomly and the operations of the selected job are directly copied in the respective positions of their offspring. As they are directly copied, the positions are not changed during the crossover process and thus, the offsprings generated will maintain the precedence relation. The job-based crossover, which is proposed by Reddy and Rao (2006) is explained. Job-based crossover operation algorithm are as follows:

1) Randomly select one job from the given job set.
2) Mark the operations of the selected jobs on the parent strings.
3) Copy the operations of the selected jobs of parent 1 onto the matching positions of offspring 2.
4) Copy the operations of the selected jobs of parent 2 onto the matching positions of offspring 1.
5) Fill the unfulfilled positions of the offspring 2 by the operations of the unselected jobs from left to right according to their order of appearance in parent 2.
6) Fill the unfulfilled positions of the offspring 1 by the operations of the unselected jobs from left to right according to their order of appearance in parent 1 (Reddy and Rao, 2006).

Figure 5 shows an example for job-based crossover during the scheduling by GA in an FMS environment by three machines and two AGVs, which hold the three jobs.

The mutation is an important operator for the GA procedure, which explores the search space and mainly prevents the premature convergence. In most of the reviewed literature in the area of the order-based GA, usually a general swap mutation operator is used, and if the results of the mutation operator offend the required order of the genes, a repair function is used to correct

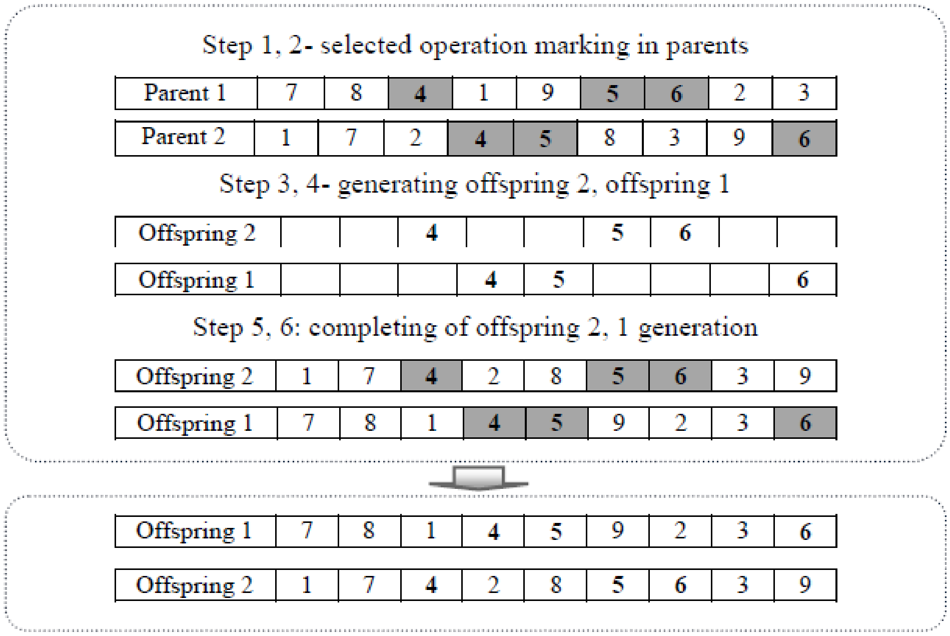

Figure 5. Job based crossover.

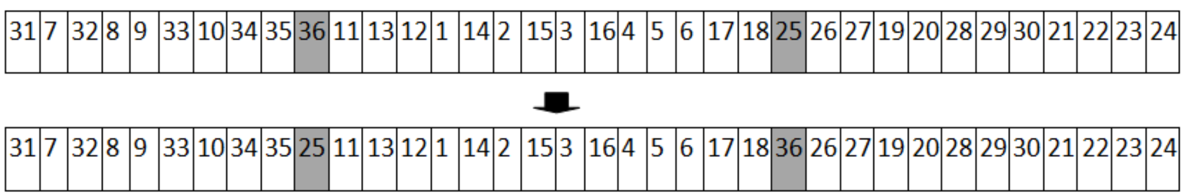

Figure 6. Job-based mutation.

the order of the genes. In this research, a new mutation operator is designed which observe the precedence of the operations in its procedure. “Job-based mutation” which is introduced by the author never offends the precedence constraints. The number of required mutations is determined by the mutation rate. Through, the proposed mutation operator, two operations of two jobs are selected for replacing mutually but must check considering operation sequencing in these two jobs. At first, randomly select two operators from one of the given chromosome that they are in different job. Secondly, change the position of two operators if in new position is between before and after operators otherwise the mutation does not occur. Job-based mutation operation algorithm are as follows:

1) Randomly select two job from the given job set.
2) Randomly select an operation from the first selected job (Mu1).
3) Randomly select an operation from another selected job (Mu2).
4) Check the position of G (Mu1-1) that is should be before the position of Mu2.
5) Check the position of G (Mu1+1) that is should be after the position of Mu2.
6) Check the position of G (Mu2-1) that it should be before the position of Mu1.
7) Check the position of G (Mu2+1) that it should be after the position of Mu1.
8) Put the Mu1 in the position of Mu2 in new chromosome.
9) Put the Mu2 in the position of Mu1 in new chromosome.
10) Fill the unfulfilled positions in chromosome by the other genes, which they are remaining.

Figure 6 shows the job-based mutation during the scheduling by GA.

FLC for GA

The choice of parameters for GAs, such as crossover and mutation rates is a rather hard task, due to the enormous possibilities of variations in the modeling of the problem and fitness function. Traditional GAs base themselves upon the generation of several random factors in the creation of the crossover and mutation. Therefore, two executions with the same initial parameters of execution can produce significantly different results.

The prime objective of using FLCs for GAs is to determine the important parameters of GAs. These parameters can be used during various generations of the GA, for a better performance of GA. The FLC receives the indices of GA periodically as its inputs and through its rule base decides about the GA parameters.

Table 3 shows the details of the rules of the designed FLC module. As it is stated by Herrera and Lozano (2003) and Brito et al. (2006), in cases that the GA reached an acceptable area of

Table 3. Proposed fuzzy rule base for the FGA method.

Rule	IF	BV	And	FBV	And	*MR*	And	*CR*
1		Good		Low		High		High
2		Good		Average		High		Average
3		Good		High		High		Low
4		Average		Low		Average		High
5		Average		Average		Average		Average
6		Average		High		High		Average
7		Poor		Low		Low		High
8		Poor		Average		Average		High
9		Poor		High		High		High

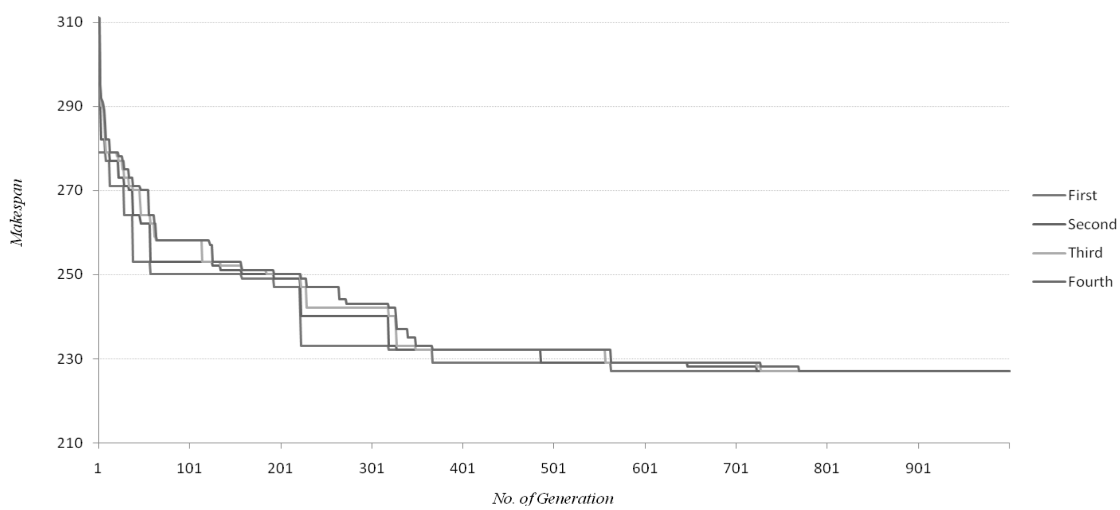

Figure 7. Four of the best makespan values for the first test case (six machines and two AGVs).

optimum values [in terms of best values (BV)], the controller generally should increase the mutation rate to provide a better coverage of the search space by increasing the exploring abilities of the GA. In most of these situations, the number of survived individuals should be increased; hence, the crossover rate is increased in such cases. Rules number one to three indicates this situation, which usually happens after a reasonable number of generations.

The second set of the rules is defined for average values of the BV for the current population. In such occasions, usually the GA should increase its exploring abilities if the premature convergence is too probable in this situation. Generally, the mutation rate is increased slightly and the crossover rate should be changed due to the diversity indicators of the last generations, which are determined by the fuzzy BV (FBV) variable. “Poor” values for BV usually obtained in early iterations of the GA. In such cases, crossover rate should be kept high to exploit more of the search space. The mutation rate in this occasion follows the diversity indicator of the last generations.

RESULTS

In this paper, the problem of simultaneous scheduling of machines and identical AGVs in FMS was addressed by considering the minimization of the makespan objective. An FLC-based GA coding scheme is developed for the studied problem.

From the testing of the proposed GA and the adopted operators, for first test case with six machine and two AGV, the best fitness or the optimal makespan by fuzzy operators (crossover, mutation) is achieved at 227 while it was between 280 to 295 before optimization. Figure 7 shows the experimentation results obtained for the proposed AGV scheduling problem in which the FLC control and protect the GA to stick in local optima. This Figure 7 shows four of the best makespan values for the test case.

The population size is set to 40, number of generations equals to 1000 and tournament selection scheme was used in both methods. Mean of BV of makespan time for 10 times run, standard deviation and best makespan among all the results are shown in Table 4.

Figure 8 shows the best sequencing for AGV scheduling. As shown in Figure 8, number seven is the

Table 4. Best, overall mean and standard deviation for GA and FGA methods in first test case.

Solution method	Best makespan	Mean of BV	Standard deviation
GA	241	246.1	5.37
FGA	227	232.6	3.41

The last chromosome
13, 31, 14, 32, 25, 33, 1, 2, 15, 7, 3, 16, 26, 4, 27, 5, 8, 34, 9, 10, 19, 35, 28, 20, 21, 22, 6, 11, 17, 23, 18, 29, 12, 24, 36, 30.

Figure 8. The best sequencing for AGV scheduling for first test case.

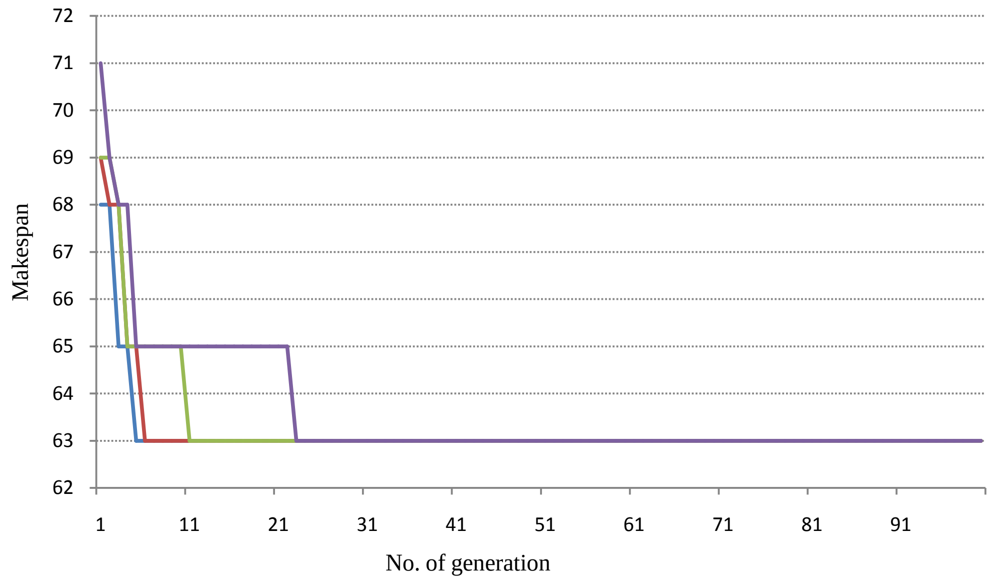

Figure 9. Four of the best makespan values for the second test case (three machines and two AGVs).

first gen in the last chromosome. It shows that the first AGV should bring the job number two to machine number two for operation number one. The operations are sequenced in one job. It was mentioned that operation four of job three came after operation three of job three in last chromosome.

In second test case, there were three machines which were linked by ladder layout, and two AGVs those were moving between the machines and L/U station and handle material between them during the production. IN this experiment, the population size was set to 40 and number of generation was set to 100. Figure 9 shows four of the best makespan values for the second test case.

Table 5 shows three various scheduling chromosomes which obtained the same result. These scheduling strings are obtained through various runs of the FGA. The optimized makespan for all these chromosomes is equal to 63. This is obvious that while the optimum scheduling string for the operations does not occur in a single location, various runs of the same FGA method, with the same characteristics would not provide the same results.

DISCUSSION

In this paper, the proposed methodology attempt to enhance the performance of GAs using FLCs in their application in the area of AGV scheduling in FMS. In simultaneous scheduling of machines and AGVs, a sequence of operations scheduled, due to their precedence and due dates, afterwards the AGVs

Table 5. Different chromosomes with same makespan for second test case.

Test case	Chromosome									Makespan
First	7	8	4	1	5	2	9	6	3	63
Second	7	4	8	1	5	2	9	6	3	63
Third	7	8	1	4	5	2	9	6	3	63

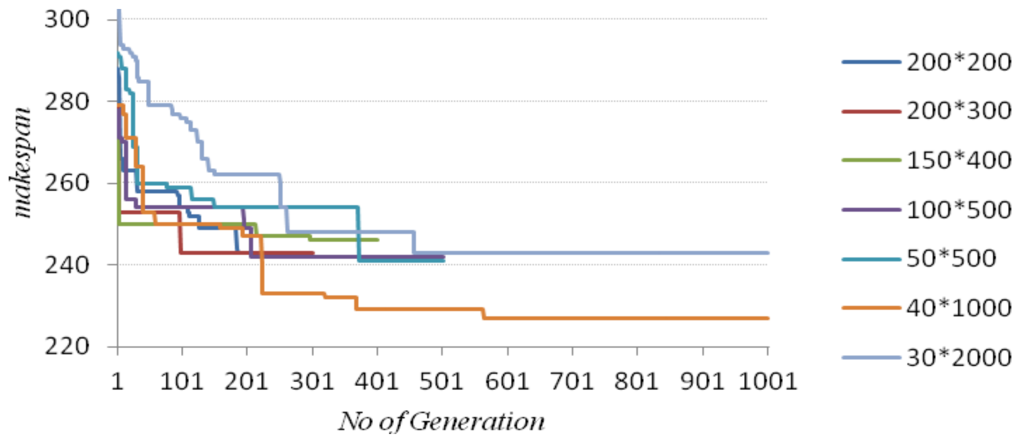

Figure 10. Comparison of various combinations of population sizes and number of generations.

assigned to the operations in a way that total production and traveling time of the sequence of operations minimizes. The literatures proposed GA as a powerful tool for the optimization of the simultaneous scheduling of machines and AGVs in FMS environments. They proposed the algorithms for the scheduling and some of the main operators for the GA such as crossover and mutation operators. In the simultaneous scheduling of machines and AGVs, the performance of GA is enhanced by using a new application of FGA methodology. Moreover, the mutation operator for the GAs is modified, so the GA process for this kind of problem is much easier now. In reviewed literature, the researchers proposed to do a repair after each mutation process, while, the mutation operator is designed in a way that the mutation is executed if and only if it does not violate the precedence of the operations. The performance of the proposed method compared with the performance of the GAs in the same test cases. The results indicate that FLCs can enhance the performance of GA in this kind of scheduling problem.

It is noteworthy that although this problem has stated in an FMS environment, it is also a valid problem in other environments, which used AGVs as their material handling system such as automated container terminals. For further researches, the authors propose to focus on other input variables for the FLCs such as the frequency of the similar chromosomes and mean value of the population. Study on the other parameters of GA which can be controlled by the FLCs, such as population size, stop criterion and survive percentage is suggested.

The performance of GA is highly affected by selection of its initial parameters. The FLC is proposed to control the mutation and crossover rates. The main advantage of using FLCs to control the key parameters of the GA is that, the main GA developer focuses on the development and improvement in the GA operators, instead of wasting efforts to select these key parameters. The FLCs can be developed and modified for any other application in the scheduling problems. The other researchers in the past literature did not provide unique approach to select the initial parameters for the GA. They proposed various combinations of the initial parameters based on their experience.

To evaluate the performance of proposed method, two test cases are designed. Three main experiments have been designed for these test cases. In second test case as mentioned earlier, the little change in the sequence of the operations may results in a meaningful increase or decreases in obtained makespan of the sequence. On the other hand, many sequences can be found which has the same makespan. Both of these reasons may result in a difficult search process to find the optimal value of the scheduling problems.

To select a proper combination of population size and number of required generations as the stop criterion of the GA, the second experiment has been designed for the second test case. In this experiment, the FGA method using tournament selection scheme was used to solve the scheduling problem. The results are shown in Figure 10.

The problem is run for seven times, with seven various combinations of population size and number of generations. The combinations were selected based on the experiments of the author, and the reviewed literature in the same realm of knowledge. The results showed that increasing the size of population does not assure the modeler to obtain a better result, even by increasing the number of generations, which was examined through comparing the results of 200*200 and 200*300 for population size and number of generations, respectively. It was obvious that population size of 40 and number of generations of 1000 outperforms the other combinations. Decreasing the population size more than 40 does not guarantee to obtain a better result, as may be cleared by comparing the results for 30*2000 and 40*1000 as the population size and number of generations, respectively.

The general results of these experiments show that the FGA cannot outperform the GA method with constant mutation and crossover rates, in small size problems, while it obtains a reasonable good solution of large scale problems. With reference to the standard deviation of the best makespan obtained in ten runs for all the test cases, the FGA may find the optimal values of makespan in most of its runs. It means that in the same number of runs of GA and FGA methods, FGA may find the optimal value with a higher probability than GA method. Totally, the probability of finding the optimal value under the FGA method is higher than GA method.

REFERENCES

Abdelmaguid TF, Nassef AO, Kamal BA, Hassan MF (2004). A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Res., 42(2): 267-281.

Babu AG, Jerald J, Noorul Haq A, Muthu Luxmi V, Vigneswaralu TP (2009). Scheduling of machines and automated guided vehicles in FMS using differential evolution. Int. J. Prod. Res., iFirst, pp. 1-17.

Chen J-H, Ho S-Y (2001). Multi-Objective Evolutionary Optimization of Flexible Manufacturing Systems. In L. Spector et al., E &. San Francisco, CA: Morgan Kaufmann. Proc. Genet. Evol. Comput. Conf., I: 1260-1267.

Corréa AI, Langevin A, Rousseau L (2007). Scheduling and routing of automated guided vehicles: A hybrid approach. Comp. Oper. Res., 34(6): 1688-1707.

de Brito FH, Teixeira AN, Teixeira ON, de Oliveira RCL (2006). A Fuzzy Intelligent Controller for Genetic Algorithms' Parameters. Springer-Verlag Berlin Heidelberg, L. Jiao et al. (Eds.): ICNC 2006, Part I, LNCS 4221: 633-642.

Egbelu PJ, Tanchoco JMA (1984). Characterization of automated guided vehicle dispatching rules. Int. J. Prod. Res., 22: 359-374.

Ganesharajah T, Hallb NG, Sriskandarajah Ch (1998). Design and operational issues in AGV-served manufacturing systems. Ann. Oper. Res., 76: 109-154.

Herrera F, Lozano M, Verdegay, JL (1999). Fuzzy Genetic Algorithms: Issues and Models. Fuzzy Sets Syst., 92: 21-30.

Herrera F, Lozano M (2003). Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy and Future Directions. Soft Comput., Springer-Verlag, 7: 545-562.

Holland JH (1975). Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, Michigan.

Jawahar N, Aravindan P, Ponnambalam SG, Suresh RK (1998). AGV Schedule Integrated with Production in Flexible Manufacturing Systems. Int. J. Adv. Manuf. Technol., 14: 428-440.

Jerald J, Asokan P, Saravanan R, Carolina AD (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.

Le-Anh T, De Koster MBM (2006). A review of design and control of automated guided vehicle systems. Eur. J. Oper. Res., 171: 1-23.

Raj T, Shankar R, Suhaib M (2007). A review of some issues and identification of some barriers in the implementation of FMS. Int. J Flex. Manuf. Sys., 19: 1-40.

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

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

Stecke KE (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. Manag. Sci., 29(3): 273-287.

Subbaiah KV, Rao MN, Rao KN (2009). Scheduling of AGVs and machines in FMS with makespan criteria using sheep flock heredity algorithm. Int. J. Phys. Sci., 4(2): 139-148.

Ulusoy G, Sivrikaya-erifo F, Bilg O (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput. Oper. Res., 24(4): 335-351.