Int J Adv Manuf Technol (2006) 31: 602–613
DOI 10.1007/s00170-005-0223-6

ORIGINAL ARTICLE

B. S. P. Reddy · C. S. P. Rao

A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS

Received: 15 January 2005 / Accepted: 28 July 2005 / Published online: 30 March 2006

Abstract A carefully designed and efficiently managed material handling system plays an important role in planning and operation of a flexible manufacturing system. Most of the researchers have addressed machine and vehicle scheduling as two independent problems and most of the research has been emphasized only on single objective optimization. Multiobjective problems in scheduling with conflicting objectives are more complex and combinatorial in nature and hardly have a unique solution. This paper addresses multiobjective scheduling problems in a flexible manufacturing environment using evolutionary algorithms. In this paper the authors made an attempt to consider simultaneously the machine and vehicle scheduling aspects in an FMS and addressed the combined problem for the minimization of makespan, mean flow time and mean tardiness objectives.

Keywords Automated guided vehicle · Evolutionary algorithms · Multiobjective · Non-dominated solutions · Scheduling

1 Introduction

In the present day automated manufacturing environment, flexible manufacturing systems (FMS) are agile and provide wide flexibility. FMS are well suited for simultaneous production of a wide variety of part types in low volumes. FMS is a complex system consisting of elements like workstations, automated storage and retrieval systems, and material handling devices such as robots and AGVs [1]. The FMS elements can operate in an asynchronous manner and the scheduling problems are more complex. Moreover the components are highly interrelated and in addition contain multiple part types, and alternative routings etc. FMS performance can be increased by better co-ordination and scheduling of production machines and material handling equipment [2].

Scheduling is concerned with the allocation of limited resources to tasks overtime and is a decision making process that links the operations, time, cost and overall objectives of the company. Scheduling of machines, other resources such as vehicles, personnel, tools etc. has been done with a certain objective, to be either minimized or maximized. Some of these objectives include minimization of makespan, tardiness, earliness, in process inventory etc. [3]. Typically parts in a manufacturing system visit different machines for different operations, and they thus generate demand for the material handling devices. Scheduling of the material handling system in FMS has equal importance as of machines and is to be considered together for the actual evaluation of cycle times. Automated guided vehicles (AGVs) are widely used in flexible manufacturing systems due to their flexibility and compatibility [4]. AGVs can be integrated with the computer controlled production and storage equipment in the shop floor and the entire shop floor operations can be controlled through a computer system.

Most of the real world-scheduling problems involve simultaneous optimization of multiple objectives. Dealing with multiple objectives has received much attention over the last few years, where as scheduling is still dominated by the unrealistic single objective approach. The main goal of a multi-objective optimization problem is to obtain a set of Pareto optimal solutions that satisfy the constraints [5]. Genetic algorithms (GA) are widely used in the last two decades to address the multi criteria decision problems. 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 [6]. In recent years, new algorithms such as non-sorting genetic algorithm–II (NSGA-II) and strength Pareto evolutionary algorithm–2 (SPEA-2) have been developed and are now widely used to address the multi-objective problems.

In this paper the authors have made an attempt to address simultaneously both the machine and vehicle scheduling in

B. S. P. Reddy (✉) · C. S. P. Rao
Manufacturing simulation lab,
Department of Mechanical Engineering,
National Institute of Technology,
Warangal-506004,
Andhra Pradesh, India
e-mail: bspr_kits@yahoo.com
e-mail: csp_rao63@yahoo.com

the flexible manufacturing environment for minimization of makespan, mean flow time and mean tardiness using non-dominated sorting genetic algorithm NSGA-II. The efficacy of the method for the said simultaneous scheduling problem has been thoroughly tested. The following sections describe the development and application of GAs for simultaneous scheduling problems with illustrated examples.

2 Literature review

Most of the researchers have addressed the machine and vehicle scheduling as two independent problems. However, only a few researchers have emphasized the importance of simultaneous scheduling of machines and vehicles. A deterministic off-line scheduling model was presented by Raman et al. [7]. They formulated the problem as an integer programming problem and proposed a solution procedure based on the concepts of project scheduling under resource constraints. Their assumption was that the vehicles always return to the load/unload station after transferring a load which reduces the flexibility of the AGV and influence the overall schedule length. Blazewicz et al. [1] has considered an FMS with parallel identical machines arranged in a loop. They addressed the simultaneous scheduling problem using a dynamic programming approach. Sabuncuoglu and Hommertzheim [4] has tested the different machine and AGV scheduling rules in FMS against the mean flow time criterion. Another off-line model for simultaneous scheduling of machines and material handling system in an FMS for the makespan minimization is presented by Bilge and Ulusoy [8]. The problem was formulated as a non-linear mixed integer-programming model and was addressed using the sliding time window approach. Ulusoy et al. [9] has addressed the same problem using genetic algorithms. In their approach the chromosome represents both the operation number and AGV assignment which requires development of special genetic operators. Anwar and Nagi [2] addressed the simultaneous scheduling of material handling operations in a trip-based material handling system and machines in JIT environment. Abdelmaguid et al. [10] has presented a new hybrid genetic algorithm for the simultaneous scheduling problem for the makespan minimization objective. The hybrid GA is composed of GA and a heuristic. The GA is used to address the first part of the problem that is theoretically similar to the job shop scheduling problem and the vehicle assignment is handled by a heuristic called vehicle assignment algorithm (VAA). Lacomme et al. [11] has addressed the simultaneous job input sequence and vehicle dispatching for a single AGV system. They solved the problem using the branch and bound technique coupled with a discrete event simulation model.

Multi-objective optimization has been a subject of interest to researchers of various backgrounds since 1970 and genetic algorithms have received considerable attention as a novel approach to the multiobjective optimization problems. Schaffer [12] has presented a multi-modal EA called vector evaluated genetic algorithm (VEGA), which carries out selection for each objective separately. An approach based on the weighted sum scalarization was introduced by Hajela and Lin [13] to search for multiple solutions in parallel. Horn et al. [14] proposed the niched Pareto genetic algorithm that combines tournament selection and the concept of Pareto dominance. Fonseca and Fleming [15] proposed a multi-objective genetic algorithm (MOGA). Their approach consists of a scheme in which the rank of an individual corresponds to the number of individuals by which it is dominated.

Based on Goldberg's suggestions Srinivas and Deb [16] developed an approach called non-dominated sorting genetic algorithm (NSGA). The non-dominated solutions of a front are assigned the same dummy fitness value and are shared with their dummy fitness values and ignored in the further classification process. Finally, the dummy fitness is set to a value less than the smallest shared fitness value in the current non-dominated front. Then the next front is extracted and the process is repeated until all the individuals in the population are classified. Zitzler and Thiele [17] has proposed the strength Pareto evolutionary algorithm (SPEA). They maintained an external archive which stores all the non-dominated solutions found at every generation from the beginning. The archive solutions are allowed to participate in the genetic operations which lead to the quick convergence of the algorithm. Knowles and Corne [18] developed an approach called Pareto archived evolution strategy (PAES) that incorporates elitism. In their approach non dominance comparison was made between a parent and the child. Deb et al. [19] proposed NSGA-II and showed that it out performs SPEA and PAES methods for certain test problems. Zitzler et al. [20] developed an approach called SPEA2 where the potential weaknesses of its predecessor SPEA are eliminated and an improved fitness assignment scheme is used.

3 Multi-objective optimization

If more than one criterion is to be treated simultaneously, then it is a multi-objective optimization problem. Most of real world decision problems involve multiple and conflicting objectives that need to be tackled while respecting the various constraints [21]. In multi-objective problems, there may not exist one solution, which is best with respect to all objectives. There exists a set of solutions, which are better than the other solutions in the search space when all the objectives are considered but are inferior to the solutions in one or more objectives and these solutions are called non-dominated solutions [22] and are shown in Fig. 1.

Problem formulation For a given FMS environment with machines arranged in a typical layout, the set of jobs to be processed and the vehicles, generate an optimum machine and vehicle schedule that is best with respect to the makespan, mean flow time and mean tardiness.

FMS environment A load/unload station with sufficient input/output buffer space serves as the distribution/collection centre for the parts. Flow path layout and the number

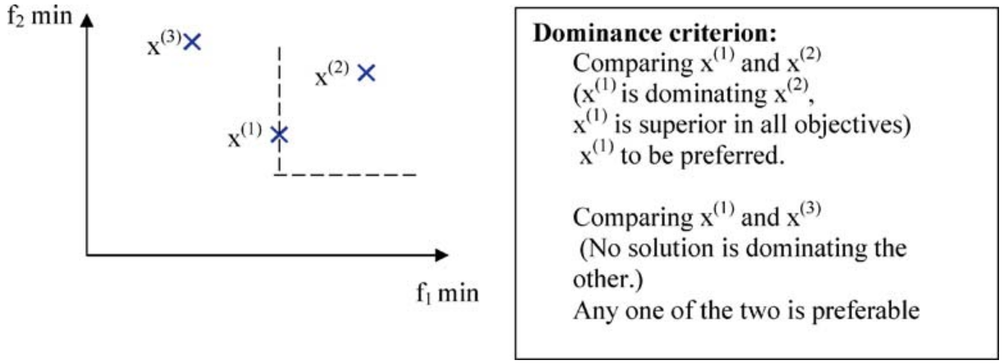


Fig. 1 Dominance criterion

of machines are known. Sufficient input/output buffer space is provided at each machine. Numbers of AGVs are known and initially all the vehicles are available at the load/unload station. AGVs carry a single unit load at a time and move along the shortest path.

Constraints Capacity, precedence and temporal constraints are considered. An order is imposed on operations that are members of one job and no two operations of a job can be done simultaneously. There is only one of each type of machine and no machine can perform more than one operation at a time. The duration of empty trips are sequence dependent and are not known until the vehicle route is specified.

Objective criteria Minimization of makespan, mean flow time and mean tardiness.

4 Hybrid multi-objective genetic algorithm coding

GA working principle Initial population of solutions is randomly generated and is evaluated based on the given objective function. For reproduction individuals are selected based on their fitness and the selected individuals undergo the genetic operations, crossover and mutation. Some of the existing individuals of the current population are replaced with the newly formed ones and this becomes the population for the next generation and this process is continued till the termination criterion is met [6].

Hybrid algorithm A hybrid genetic algorithm was developed to address the combined machine and vehicle-scheduling problem. The hybrid GA proposed is a combination of a genetic algorithm along with a heuristic technique to address different phases of the combined simultaneous scheduling problem. GA is used to address the machine scheduling problem and the vehicle scheduling problem is addressed using the heuristic. This type of hybridization reduces the size of the string length to half the length of the string for the combined problem. It reduces the search space, minimizes the constraints on the search and increases the efficiency of the GA search. Operation based encoding is used for the machine scheduling problem which helps to simplify the representation and does not require any new GA operators. The heuristic scheduling algorithm is embedded in the GA structure and called during the fitness evaluation module.

The heuristic decodes the GA operation sequence and assigns the tasks to the vehicles based on the sequence. It always checks the vehicle and job status, computes the vehicle availability at the required demand point and assigns tasks.

The results obtained by the proposed hybrid GA are compared with the results reported by Ulusoy et al. [9] and Abdelmaguid et al. [10] for a set of the said simultaneous scheduling problem. The algorithm is found to be performing well. The waiting times are largely reduced by the heuristic vehicle scheduling method used, where the vehicles need not wait until there is a request but the assignments are made based on the calculations of the job completion times and the results and discussion are presented in the Appendix.

Multi-objective optimization The same algorithm is extended to address the multi-objective optimization of mean tardiness and mean flow time along with the makespan. The additional module of non-dominated sorting and fitness assignment are incorporated in the algorithm structure and the algorithm steps are discussed below.

Chromosome representation Representation of solutions is a key issue for implementing genetic algorithms especially for the heavily constrained job shops. A direct representation of candidate solutions to the scheduling problem is used as a chromosome. Care is taken in generating feasible chromosomes that maintain the precedence relations that are crucial in job shop scheduling problems.

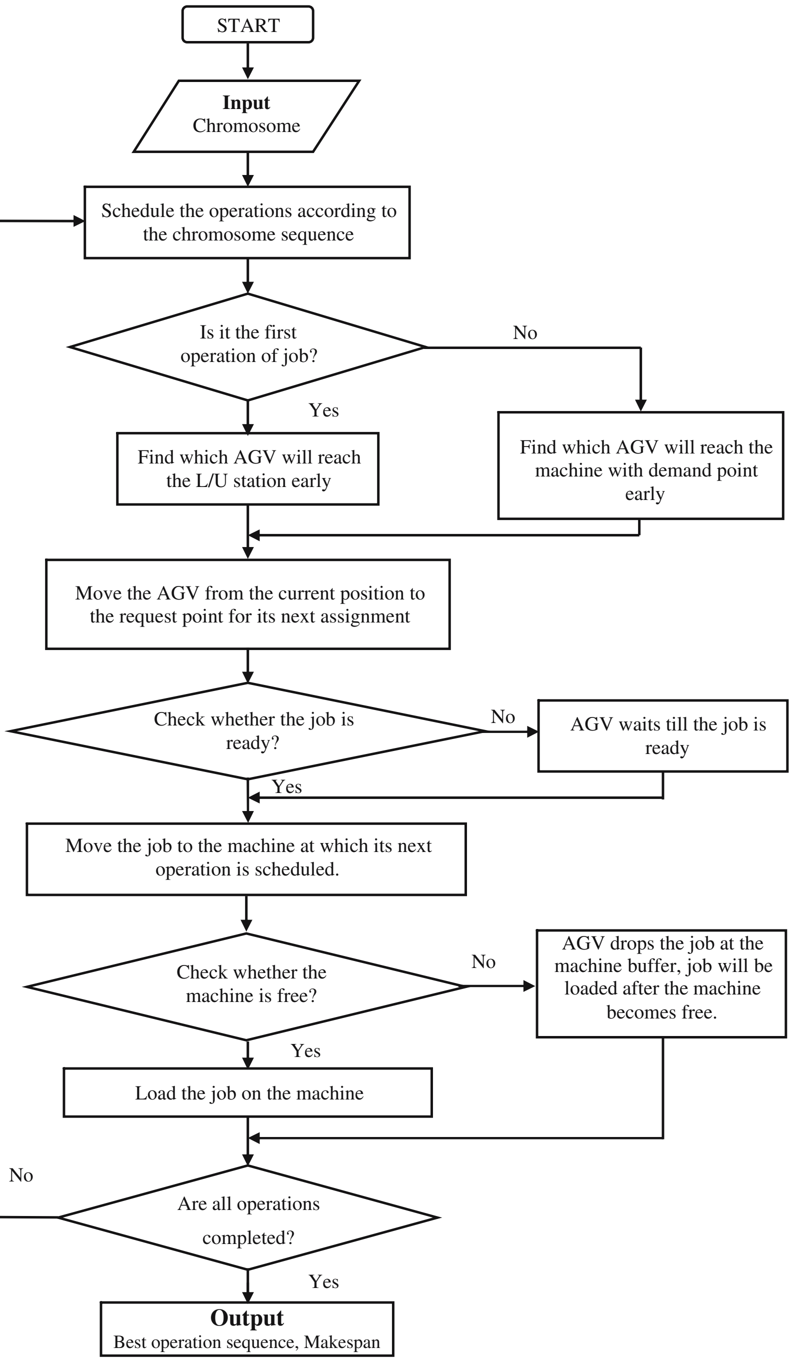
Chromosome represents the operation sequence and each operation denotes the processing of a specific part on a specific machine center, and is expressed by an integer number starting from 1. The chromosome length is equal to the total number of operations of all the jobs put together.

Example A scheduling situation with 4 work centres and 3 work pieces is considered. Total there are 10 operations and the chromosome consists of 10 genes.

No. of Jobs		1				2			3	
No. of Operations	1	2	3	4	1	2	3	1	2	3
Machines	M1	M3	M4	M2	M2	M3	M1	M1	M4	M3
Representation	1	2	3	4	5	6	7	8	9	10
One feasible chromosome:	1	5	8	2	3	6	9	4	7	10

Vehicle scheduling methodology Jobs are scheduled based on the operation sequence derived by the genetic algorithm. 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 pickup 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 assign the task 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 pickup that job. This type of vehicle scheduling methodology helps in reducing the waiting times and thus helps in improving the resource utilization and the throughput. The flow chart of the vehicle assignment methodology is given in Fig. 2.

Fig. 2 Simultaneous Scheduling flow chart



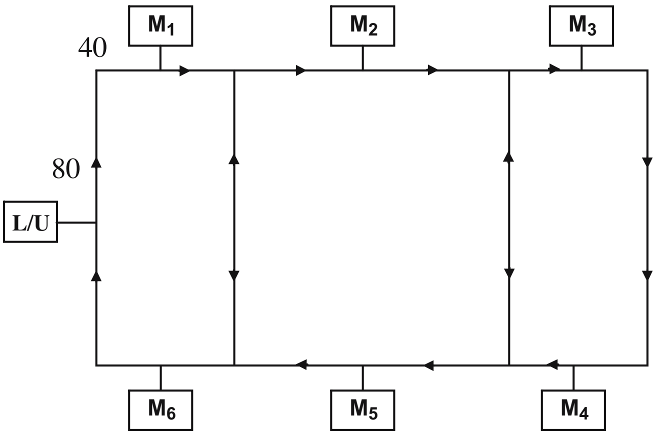


Fig. 3 FMS Layout

Fitness function Every individual in each generation is evaluated for all the three objective values makespan, mean flow time and mean tardiness based on the given GA operation sequence and the heuristic vehicle assignment method.

$$\text{Operation completion time} = O_{ij} = T_{ij} + P_{ij}$$

*j*th operation *i*th job (traveling time + operation processing time)

$$\text{Job completion time } C_i = \sum_{i=1}^{n} O_{ij}$$

$$\text{Makespan} = \text{Max}(C_1, C_2, C_3, \_\_\_\_\_ C_n)$$
$$\text{Flow time} = F_i = C_i - R_i$$
(Completion time – release time)

$$\text{Mean flow time} = \frac{1}{n}\sum_{i=1}^{n} F_i$$

$$\text{Lateness} = L_i = C_i - D_i$$
(Completion time – due date)

$$\text{Tardiness} = T_i = \max\left[L_i, 0\right]$$

$$\text{Mean tardiness} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

Non-dominated sorting All the individuals in the population are sorted based on their non-domination level considering all the objectives. Every solution in the population is compared with the other solutions in the population for all the objectives and a solution is declared as a non-dominated solution if it is superior in at least one objective over the other solution [21]. All such solutions are identified and the fitness of a solution is assigned based on the number of solutions it dominates.

Example	Makespan	Mean flow time	Mean tardiness
Solution 1	168.11	146.392	14.78
Solution 2	170.33	141.734	14.62

Solution 1 is superior with respect to makespan however solution 2 is better with mean flow time and tardiness.

Selection The strings for reproduction are selected using the binary tournament selection process. Two strings are randomly selected and compared, the string with higher fitness is selected to participate in reproduction; thus, the higher fitness strings have a chance to have a larger number of copies selected.

Crossover Job-based crossover is used which never violates 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 off springs generated will maintain the precedence relation.

Step 1:
Randomly select some jobs from the given job set.
Step 2:
Mark the operations of the selected jobs on the parent strings.
Step 3:
Copy the operations of the selected jobs of parent 1 onto the corresponding positions of offspring 2.
Step 4:
Copy the operations of the selected jobs of parent 2 onto the corresponding positions of offspring 1.
Step 5:
Fill the unoccupied 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.

Table 1 Job set details (Fisher and Thompson 6×6 instance **mt06**)

6 jobs each with 6 operations are to be processed on 6 machines Job due date along with the machine number and processing time are given alternatively

Job No	DD	M	PT	M	PT	M	PT	M	PT	M	PT	M	PT
1	35	2	1	0	3	1	6	3	7	5	3	4	6
2	70	1	8	2	5	4	10	5	10	0	10	3	4
3	105	2	5	3	4	5	8	0	9	1	1	4	7
4	140	1	5	0	5	2	5	3	3	4	8	5	9
5	175	2	9	1	3	4	5	5	4	0	3	3	1
6	210	1	3	3	3	5	9	0	10	4	4	2	1

Table 2 Travel time matrix

	L/U	M1	M2	M3	M4	M5	M6
L/U	0	4	6	8	14	12	10
M1	10	0	3	5	11	9	7
M2	12	15	0	3	9	7	9
M3	14	17	15	0	7	9	11
M4	8	11	9	7	0	3	5
M5	6	9	7	9	15	0	3
M6	4	7	9	11	17	15	0

Step 6:
Fill the unoccupied positions of the offspring1 by the operations of the unselected jobs from left to right according to their order of appearance in parent 1.

Example
Chromosomes selected for crossover:
Parent1: 1 5 8 2 6 9 3 7 10 4
Parent2: 5 8 1 9 2 3 6 10 7 4
Let the job selected is 2 and the corresponding operations of job 2 are 5, 6 and 7.
Parent1: 1 5 8 2 6 9 3 7 10 4
Parent2: 5 8 1 9 2 3 6 10 7 4
Resulting off springs:
Offspring1: 5 1 8 2 9 3 6 10 7 4
Offspring2: 8 5 1 9 6 2 3 7 10 4

Mutation operator 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 infeasibilities.

Example
Chromosome selected for mutation: 1 5 8 2 3 6 9 4 7 10
Let the randomly selected positions are 8 and 10.
Mutated chromosome is: 1 5 8 2 3 6 9 10 7 4

Repair function A repair function is used to see that the chromosomes do not violate the precedence constraints.
Repair function [9].

Table 3 Pareto optimal set

Makespan	Mean flow time	Mean tardiness
207	167.16	61.33
221	164	41.5
212	163.83	43
224	162.83	41.16
217	163.16	52
241	161.83	50.83
213	163.66	43.16
245	161.33	50.33

Step 1:
Find the position of the operations, which violate the precedence relations.
Step 2:
Compute the distance between the violating operations.
Step 3:
If the distance between them is less than half the chromosome length then swap the operations.
Step 4:
Otherwise randomly pick any one operation and insert it before or after the other depending on the precedence.

Elitism Elitism is used to retain some of the best individuals from each generation as such individuals can be lost during the crossover and mutation operations and this significantly helps in improving the GA performance. An external archive is maintained where the specified number of best strings in each generation are preserved and the archive is updated after every generation and the process continues till the termination.

5 Hybrid multi-objective genetic algorithm procedural steps

Step 1:
Enter the input data: Number of machines number of AGVs and distance between machines. Job set details- Number of jobs, operations on each job, processing times and due dates.
GA Parameters- Population size, probability of cross over, mutation and termination criteria.
Step 2:
GA procedure: Randomly generate initial population of solutions using the operation based encoding. Maintain an archive (external set) same as the population size.
Step 3:
Evaluate each string in the population for all the objective criteria.
Step 4:
Simultaneous scheduling procedure: Move jobs from the load/unload station according to the given chromosome sequence.

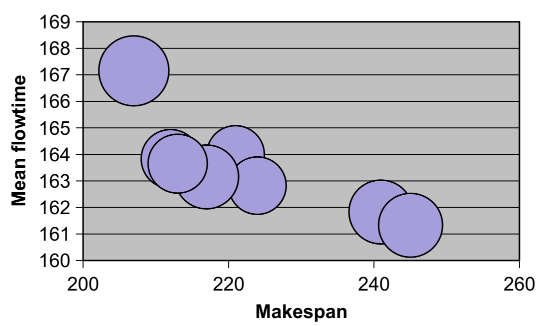


Fig. 4 Pareto front

Table 4 Case study results

Prob. No	Pareto optimal set		
	Makespan	Mean flow time	Mean tardiness
Small size SS1			
FMS layout: 1			
Number of machines: 4	98	82.2	7.0
Travel time matrix: TTM1	96	83.2	8.4
Job set details: EX11 Bilge and Ulusoy 1995	99	78.8	9.2
Number of jobs: 4	97	82.8	9.8
Operations on each job: 3 3 3 2 2			
Total number of operations: 13			
Due date (average makespan): 20			
Small size SS2	394	257.1	101.4
FMS layout: 2	360	265.5	103.9
Number of machines: 5	358	268.4	106.2
Travel time matrix: TTM 2	344	270.8	108.1
Job set details: JSP1 Bagchi 1999	391	265.2	108.9
Number of jobs: 10	360	262.2	111.3
Operations on each job: 5			
Total number of operations: 50			
Due date (average makespan): 37			
Medium size MS3			
FMS layout: 3			
Number of machines: 5			
Travel time matrix: TTM3	993	762.03	276.06
Job set details: G1 Lawrence 15x5 Instance	971.5	810.43	321.36
Number of jobs: 15	950.5	809.16	327.83
Operations on each job: 5			
Total number of operations: 75			
Due date (average makespan): 65			
Medium size MS4			
FMS layout: 4			
Number of machines: 5			
Travel time matrix: TTM4	412	295.075	98.3
Job set details: JSP2 Bagchi 1999	407.5	291.8	98.875
Number of jobs: 20	402.5	294.325	100
Operations on each job: 5			
Total number of operations: 100			
Due date (average makespan): 20			
Medium size MS5			
Number of machines: 10			
FMS layout: 5			
Travel time matrix: TTM5	1190.5	979.2	390.45
Job set details: A5 Lawrance 10x10 Instance	1184.5	995.3	399.95
Number of jobs: 10	1180.5	1014.1	408.05
Operations on each job: 10	1162.5	1006	409.45
Total number of operations: 100			
Due date (average makespan): 115			
Large size LS6	1592	1296.87	530.967
FMS layout: 5	1571	1292.93	532.733
Number of machines: 10	1627	1285.13	535.333

Table 4 (continued)

Prob. No	Pareto optimal set		
	Makespan	Mean flow time	Mean tardiness
Travel time matrix: TTM5	1571	1306.87	542.067
Job set details: B1 Lawrance 15x10 Instance	1530	1322.93	544.267
Number of jobs: 15	1517	1318.53	544.533
Operations on each job: 10	1561	1313.6	548.6
Total number of operations: 150	1530	1293.4	550.733
Due date (average makespan): 100	1528	1315.4	555.4

Step 5:
AGVs return to the load/unload station till all the jobs are moved from the load/unload station.

Step 6:
Whenever a job is ready check the AGV availability and make assignment to the earliest available vehicle. The vehicle picks and moves the job to its next assigned machine.

Step 7:
If no job is ready check which job operation is expected to complete early and move the AGV to that machine.

Step 8:
If the job is not completed the AGV waits till the operation is over. AGV moves the job to the machine on which the job next operation is to be performed.

Step 9:
Check whether the machine is free or not. If it is free load the job, else the job waits in the buffer till the machine becomes free.

Step 10:
Check whether all the operations are completed. If not repeat from step 6 onwards.

Step 11:
If all the operations are completed compute the objective function values makespan, mean flow time and tardiness.

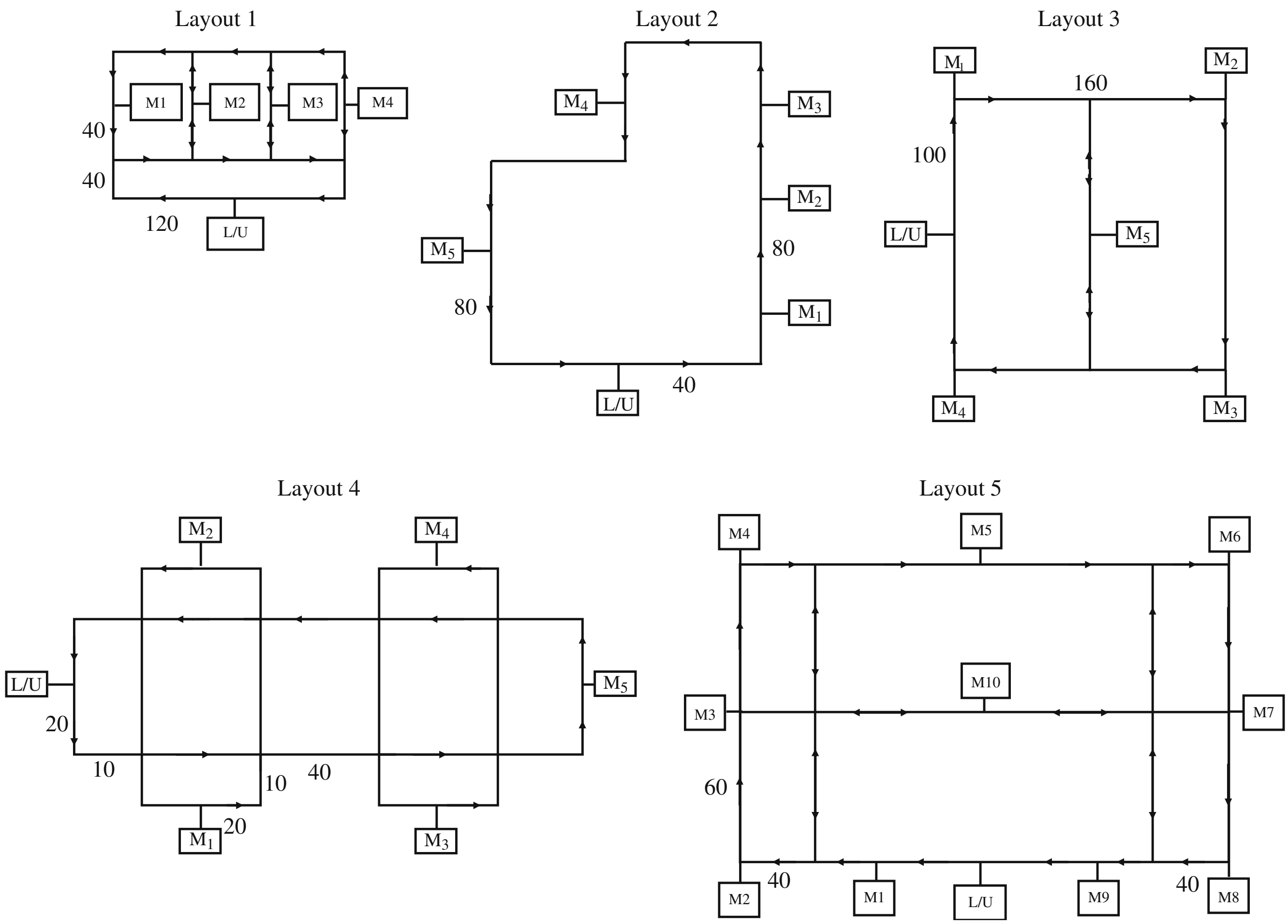


Fig. 5 FMS Layouts (Distances between machines are in meters and dimensions not to scale)

Non-dominated sorting algorithm [19]

Step 12:

Classify the population into different non- domination levels and assign fitness.

Step 13:

Offspring population is created using binary tournament selection, job based crossover and operation swap mutation.

Step 14:

Elitism procedure: Combine parent and child population.

Step 15:

Sort the combined population for non- dominated individuals.

Step 16:

Compute the crowding distance of the chromosomes.

Table 5 Travel time matrices

TTM1

	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

TTM 2

	L/U	M_1	M_2	M_3	M_4	M_5
L/U	0	3	5	7	9	14
M_1	15	0	3	5	8	12
M_2	13	15	0	3	6	10
M_3	11	13	15	0	4	8
M_4	7	9	11	13	0	5
M_5	4	6	8	10	13	0

TTM 3

	L/U	M1	M2	M3	M4	M5
L/U	0	3.5	7.5	12.5	12.5	8
M1	12.5	0	5	10	10	5.5
M2	12.5	15	0	6	10	10.5
M3	7.5	10	10	0	5	5.5
M4	3.5	6	10	15	0	10.5
M5	8	11.5	5.5	10.5	5.5	0

TTM 4

	L/U	M_1	M_2	M_3	M_4	M_5
L/U	0	2.5	4.5	4.5	6.5	5.5
M_1	4.5	0	3.5	3.5	5.5	4.5
M_2	2.5	3.5	0	5.5	7.5	6.5
M_3	6.5	7.5	5.5	0	3.5	2.5
M_4	4.5	5.5	3.5	3.5	0	4.5
M_5	5.5	6.5	4.5	4.5	2.5	0

TTM 5

	L/U	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
L/U	0	2	4	5.5	7	8	11	9.5	11	11	6.5
M1	11	0	3	4.5	6	7	10	8.5	10	10	5.5
M2	11	12	0	2.5	4	7	10	8.5	10	10	5.5
M3	9.5	10.5	4.5	0	2.5	5.5	8.5	7	8.5	8.5	4
M4	11	12	6	4.5	0	4	7	8.5	10	10	9.5
M5	8	9	11	9.5	11	0	4	5.5	7	7	6.5
M6	7	8	10	8.5	10	11	0	2.5	4	6	5.5
M7	5.5	6.5	8.5	7	8.5	9.5	4.5	0	2.5	4.5	4
M8	8	5	7	8.5	10	11	6	4.5	0	3	5.5
M9	2	3	5	6.5	8	9	12	10.5	12	0	7.5
M10	6.5	7.5	5.5	4	5.5	6.5	5.5	4	5.5	5.5	0

Step 17:
: Copy the non-dominated solutions into an archive equal to population size to form next generation population.

Step 18:
: Check whether the termination criterion is satisfied if not repeat the procedure.

Table 6 Results comparison for t/p ratio >0.25

Prob.No	STW	UGA	AGA	PGA
EX11	96	96	96	96
EX21	105	104	102	**100***
EX31	105	105	99	99
EX41	118	116	112	112
EX51	89	87	87	87
EX61	120	121	118	118
EX71	119	118	115	**111***
EX81	161	**152***	161	161
EX91	120	117	118	**116***
EX101	153	150	147	147
EX12	82	82	82	82
EX22	80	76	76	76
EX32	88	85	85	85
EX42	93	88	88	**87***
EX52	69	69	69	69
EX62	100	98	98	98
EX72	90	85	79	79
EX82	151	**142***	151	151
EX92	104	102	104	102
EX102	139	137	136	**135***
EX13	84	84	84	84
EX23	86	86	86	86
EX33	86	86	86	86
EX43	95	91	89	89
EX53	76	75	74	74
EX63	104	104	104	**103***
EX73	91	88	86	83*
EX83	153	**143***	153	153
EX93	110	105	106	105
EX103	143	143	141	**139***
EX14	108	103	103	103
EX24	116	113	108	108
EX34	116	113	111	111
EX44	126	126	126	126
EX54	99	97	96	96
EX64	120	123	120	120
EX74	136	128	127	**126***
EX84	163	163	163	163
EX94	125	123	122	122
EX104	171	164	159	**158***

6 Illustrative example

FMS environment The FMS layout along with the distances between the machines and from the load/unload station is shown in Fig. 3. The FMS consists of 6 machines and 2 AGVs. The job set details are given in Table 1. AGVs move with a speed of 40 m/min and the loading and unloading times of job are 0.5 min each. The travel times

Table 7 Results comparison for t/p ratio <0.25

Prob.No	STW	UGA	AGA	PGA
EX110	126	126	126	126
EX210	148	148	148	148
EX310	150	**148***	150	150
EX410	121	119	119	119
EX510	102	102	102	102
EX610	186	186	186	186
EX710	137	137	137	137
EX810	292	**271***	292	292
EX910	176	176	176	176
EX1010	238	**236***	238	238
EX120	123	123	123	123
EX220	143	143	143	143
EX320	148	145	145	145
EX420	116	114	114	114
EX520	100	100	100	100
EX620	183	181	181	181
EX720	136	136	136	136
EX820	287	**268***	287	287
EX920	174	173	173	173
EX1020	236	238	236	**236***
EX130	122	122	122	122
EX230	146	146	146	146
EX330	149	146	146	146
EX430	116	114	114	114
EX530	99	99	99	99
EX630	184	182	182	182
EX730	137	137	137	137
EX830	288	**270***	288	288
EX930	176	174	174	174
EX1030	237	241	237	**237***
EX140	124	124	124	124
EX241	217	217	217	217
EX340	151	151	151	151
EX341	222	221	221	221
EX441	179	172	172	172
EX541	154	148	148	148
EX640	185	184	184	184
EX740	138	137	137	137
EX741	203	203	203	203
EX840	293	**273***	293	293
EX940	177	175	175	175
EX1040	240	244	240	**240***

are computed and are presented in Table 2, which includes the loading and unloading times of the job.

Multi-Objective criteria Minimization of makespan, mean flow time, mean tardiness.

GA Parameters Population size=200, Archive size=200, Probability of crossover=0.8, Probability of mutation=0.4, Number of generations=100.

7 Results and discussions

The Pareto optimal set obtained by the proposed methodology for the described FMS environment is given in Table 3 and the Pareto front is shown in Fig. 4, where the diameter of the circle represents the mean tardiness of the corresponding solution.

Extensive testing of the algorithm The authors tested the standard job shop benchmark instances contributed by Dirk C. Mattfeld and Rob J.M.Vaessens to the OR library. The test problem instances proposed by Lawrence [23] are tested on different FMS layouts. The problems are categorized into small size problems (SS) involving upto 50 operations, medium size problems (MS) upto 100 operations and large size problems (LS) involving above 100 operations. A few case study results are reported in Table 4, the corresponding FMS layouts are shown in Fig. 5 and the travel time matrices are given in Table 5. 2 AGVs are considered and the due dates of the jobs for the test problems are assumed as the average makespan distributed equally among the jobs. In most of the test cases the results were terminated before 100 generations for the probabilities of crossover and mutation as 0.8 and 0.4, respectively.

8 Conclusions

The present study is focused on the multi-objective scheduling of FMS using genetic algorithms. The simultaneous scheduling of machines and AGVs in FMS are addressed using the multi-objective hybrid GA for the minimization of makespan, mean flow time and mean tardiness objectives. The multi-objective hybrid genetic algorithm is coded in VC++ and the algorithm presents a good number of diversified and non-dominated solutions. In most of the test problems the algorithm is found to be converged within 100 generations for a crossover and mutation probabilities of 0.8 and 0.4, respectively.

Acknowledgement The authors are grateful to the unknown referrers for their constructive suggestions that improved the quality of this paper.

Appendix

The authors tested the performance of the proposed algorithm for single objective makespan minimization with the problems reported in literature by Ulusoy et al. [9] and Abdelamaguid et al. [10] for different t/p ratios (travelling time/processing time). The results reported in the literature are compared with the best results obtained by the proposed method for 20 runs for a population size: 150, number of generations: 100, probability of crossover: 0.8 and probability of mutation: 0.4 and are reported in Tables 6 and 7 for t/p ratio >0.25 and t/p ratio <0.25, respectively.

Excluding the UGA (Ulusoy GA) results of the 7 problems (EX81, EX82, EX83, EX810, EX820, EX830 & EX840), which are proven to be invalid in Abdelmaguid et al. [10] and considering the results obtained by STW (sliding time window) as the optimal solutions, UGA fails to produce better solutions compared to STW in 6 problems. AGA (Abdelmaguid GA) performs better in 22 problems where as UGA in 5 problems out of 82 problems.

The proposed GA (PGA) is superior in most of the cases or at least equal to the other methods. The results of the proposed algorithm are better over the AGA in 12 problems and UGA in 21 problems for the case t/p ratio >0.25.

For t/p ratio <0.25 the proposed algorithm performs same as the AGA and better in 3 problems over UGA, where as UGA performs better in 2 problems. At the outset, out of the 82 problems the proposed method performs better than UGA in 24 problems and better than AGA in 12 problems.

References

1. Blazewicz J, Eiselt HA, Finke G, Laporte G, Weglarz J (1991) Scheduling tasks and vehicles in a flexible manufacturing system. Int J Flex Manuf Syst 4:5–16
2. Anwar MF, Nagi R (1998) Integrated scheduling of material handling and manufacturing activities for JIT production of complex assemblies. Int J Prod Res 36(3):653–681
3. Baker KR (1974) Introduction to sequencing and scheduling. Wiley, New York
4. Sabuncuoglu I, Hommertzheim DL (1992) Experimental investigation of FMS machine and AGV scheduling rules against the mean flow time criterion. Int J Prod Res 30(7): 1617–1635
5. Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York
6. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston
7. Raman N, Talbot FB, Rachamadgu RV (1986) Simultaneous scheduling of machines and material handling devices in automated manufacturing. In: Proc Second ORSA/TIMS Conf. on Flexible Manufacturing Systems
8. Bilge U, Ulusoy G (1995) A time window approach to simultaneous scheduling of machines and material handling system in an FMS. Opns Res 43:1058−1070

9. Ulusoy G, Serifoglu FS, Bilge U (1997) A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput Ops Res 14(4):335−351
10. Abdelmaguid TF, Nassef AO, Kamal BA, Hassan MF (2004) A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int J Prod Res 42: 267−281
11. Lacomme P, Moukrim A, Tchernev N (2005) Simultaneous job input sequencing and vehicle dispatching in a single – vehicle automated guided vehicle system: a heuristic branch -and -bound approach coupled with a discrete events simulation model. Int J Prod Res 43(9):1911–1942
12. Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. Proc First International Conference on Genetic Algorithms, pp 93–100
13. Hajela P, Lin CY (1992) Genetic search strategies in multicriterion optimal design. Struct Optimiz 4:99–107
14. Horn J, Nafpliotis N, Goldberg DE, (1994) A niched pareto genetic algorithm for multi-objective optimization. Proc First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation 1:82–87, Piscataway, NJ
15. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi objective optimization: Formulation, discussion and generalization. Proc Fifth International Conference on Genetic Algorithms, San Mateo, CA, pp 416–423
16. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2 (3):221–248
17. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271
18. Knowles JD, Corne DW (1999) The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization. Congress on Evolutionary Computation (CEC99), vol. 1, Piscataway, NJ, 98–105
19. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL Report No. 200001, Indian Institute of Technology Kanpur, India
20. Zitzler E, Lawmans M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-Report 103, Swiss Federal Institute of Technology (ETH), Zurich
21. Bagchi TP (1999) Multiobjective scheduling by genetic algorithms. Kluwer, Dordrecht
22. Deb K (2000) Multi-objective optimization using evolutionary algorithms. Wiley, New York
23. Lawrence S (1984) Resource constrained project scheduling an experimental investigation of heuristic scheduling technique. Graduate School of Industrial Adminstration. Carnegie-Mellon University, Pittsburgh, PA