

Combined Objective Optimization of FMS Scheduling by a Hybrid Genetic Algorithm



K. Gayathri Devi, R. S. Mishra, and A. K. Madan

Abstract Flexibility is an important character of flexible manufacturing systems (FMSs). All the developments made so far aims to improve the flexibility in FMS. Scheduling plays an important part in obtaining that flexibility while implementing a FMS. In real-world scenario, scheduling problems are more complex and hence solving single objective is not enough anymore. This article addresses one such problem where an attempt has been made to minimize the machine idle time and to minimize the overall penalty cost by combined optimization. A new novel hybrid method christened as GAPSOTS has been introduced in this paper. The hybrid technique is an amalgamation of three meta-heuristics approaches, namely genetic algorithm (GA), tabu search (TS) and particle swarm optimization (PSO). To determine the validity of the proposed approach, two large variety problems of 43 jobs 16 machines and 80 jobs 16 machines of FMS system taken from literature are solved for optimal scheduling. The feasibility and adaptability of the hybrid algorithm GAPSOTS are found by comparing experimental outcomes obtained from other hybrid techniques. The optimal schedule obtained by GAPSOTS surpasses other existing meta-heuristic techniques, thereby proving the proposed algorithm is superior in its performance.

Keywords Flexible manufacturing systems (FMS) · Scheduling · Genetic algorithm · Tabu search · Particle swarm optimization (PSO) · Hybrid meta-heuristics

1 Introduction and Literature Review

Increasing customer demand in the market leads to question the efficiency of mass production. Flexible manufacturing system has to achieve success by meeting up the highly increasing competition in the market. An efficient FMS must fulfil the demands of market where customized products are gaining importance. An efficient

K. Gayathri Devi (✉) · R. S. Mishra · A. K. Madan
Department of Mechanical, Delhi Technological University, New Delhi, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022
K. Govindan et al. (eds.), *Advances in Mechanical and Materials Technology*,
Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-16-2794-1_11

125

FMS should have highly reliable flexibility. The scheduling optimization is one of the important problems in FMS as it decides the efficiency and capability of the system. It is highly complex. It differs from conventional scheduling by routing flexibility that is available in the FMS. Scheduling is critical as its primary goal is to minimize the time taken to finish a product within a deadline. The level of difficulty for sequencing and scheduling of FMS depends upon the type of FMS, its process constraints and the performance parameters. Scheduling is one of the “NP” (nondeterministic polynomial time) hard problems, and it is complex in both time-wise and computational-wise.

Scheduling is a problem where jobs are allocated to particular machines with processing times and specific sequence of operations. It has to be ensured that machines are available without any breakdown, and it has to process an operation uninterrupted [1]. The intricacy of scheduling problem in a FMS is difficult than conventional job shop scheduling, as it should allocate jobs among the set of machines available. It is a well-known NP hard problem.

1.1 Earlier Research

FMS can be classified as (a) design problems (b) planning problems (c) scheduling problems (d) control problems. Stecke [2] proposed a framework to study planning and scheduling problems of FMSs. FMS problems are solved by traditional techniques and non-traditional techniques like meta-heuristics. In our proposed approach, we have used meta-heuristic techniques. Hence, the review is mainly done for scheduling problems that used meta-heuristic techniques. Zhao et al. [3] have presented a genetic algorithm with the concepts of virtual and real operations. Chromosome coding and genetic operators of GAs are defined during the problem solving. A minimum weighted tardiness objective function is used to define code fitness, which is used for selecting species and producing a new generation of codes. Kumar et al. [4] have used ACO for the graph-based representation of the FMS scheduling problem. The authors have studied the performance of various parameters and validated it. Noorul Haq et al. [5] have contemplated production and MHS scheduling in combination. The authors studied different scheduling decisions of FMS to check its efficiency by GA and SA. Jerald et al. [6] have proposed four different approaches such as PSO, GA, SA and memetic algorithm (MA) for large variety of problems with 10–43 jobs and 8–16 machines. PSO showed its superior performance over other techniques. Gnavel Babu et al. [7] have made AGV as a part of the scheduling problem. The author demonstrated the effectiveness of proposed differential evolution in obtaining optimal solutions for simultaneous scheduling for the same. Udhayakumar and Kumanan [8] have adopted extended Giffler and Thompson to generate an optimal schedule by an ACO approach. Their algorithm was found to be superior as its results are better than other techniques found in the literature. Sreedhar kumar et al. [9] have solved a combined objective function by bacterial foraging algorithm (BFOA). They moderated the penalty by including a

reward and evaluated the effectiveness of objective function. Nidish Mathew Nidhry et al. [10] have attempted to solve a multi-objective problem. The authors considered 16 machines with 80 job varieties. And they have proved that their algorithm performs better than other algorithms. Reddy and Rao [11] considered the simultaneous scheduling of machine and automated guided vehicles for multi-objectives. The authors considered some benchmark problems reported in the literature and few case studies to test the proposed hybrid GA and found the algorithm to have a better performance. Burnwal and Deb [12] had developed a modified cuckoo search to solve a 43 jobs with 16 machines problem. They have proved that CS requires less time to converge and have the potential to explore further. Chawla et al. [13] have investigated the simultaneous scheduling of AGVs workload and its travel time. They have considered multi-objectives and obtained the optimal schedule by grey wolf optimization algorithm (GWO).

Literature review shows that various meta-heuristics have been applied in order to achieve the best optimization of benchmark problems. Though many meta-heuristic techniques have found success in finding optimal solution, there is still a scope to find better optimal result when we hybridize the meta-heuristic techniques. One such attempt has been made in this article, and an amalgamation of genetic algorithm (GA), particle swarm optimization (PSO), tabu search (TS) has been done. The resulting hybrid technique is christened as GAPSOTS. The main advantage of this hybrid technique is that we have used one global search and two local search methods, thereby preventing the trapping of local optima. GA has good probability for finding better global solutions, PSO is known for solving combinatorial optimization, and TS gives a better neighbourhood search. The novel algorithm GAPSOTS has been applied to generate optimal schedules for the FMS environment taken from the literature for combined objectives, which contradicts each other. The FMS environment considered in our paper is for two different set-ups, a 43 jobs 16 machine problem and 80 jobs 16-machine problem. The outcomes are then compared with existing optimal values taken from the literature for the same problems. The results and future scope have been discussed in results section.

2 Problem Definition

The FMS set-up considered in this work, the assumptions made and objective of the present work are adopted from [6] and are presented in the following points.

1. The FMS set-up taken in this work is shown in Fig. 1. It has five flexible machining cells (FMCs), with two to six computer numerical machines (CNCs) for each FMC, an autonomous tool magazine, one tool changer (ATC) that is automatic and one pallet changer (APC) that is automatic. One to three dedicated robots are assigned for intra cell movement of resources among the jobs. A loading and unloading stations are dedicated exclusively to release parts in batches when required and for moving the finished jobs to the storage. The

automatic storage and retrieval system (AS/RS) are utilized for stocking the jobs that are incomplete. The five FMCs are conjoined by a couple of similar AGVs. These AGVs move any unfinished products to AS/RS and FMC and move finished products to the unloading dock.

2. The following assumptions are made: i) The number of products considered vary from 40 to 80 jobs for a specific mix of tools in tool magazines. Each job must be processed in a specific order, lot size, due date, and a penalty cost is imposed when the product due date is not met. Every job has to follow a particular sequence in a particular machine for specified amount of time. A product mix with processing times is shown in Table 2.
3. This study aims to solve a combined optimization problem for minimization of idle time of a machine and total penalty cost.

Combined Objective Function (COF)

$$\text{COF} = (\omega_1) * \frac{\text{TPC}}{\text{Max PP}} + (\omega_2) * \frac{\text{TMI}}{\text{Max TE}} \quad (1)$$

where

ω_1 and ω_2	weights assigned. The value considered is 0.5 for both weights.
TPC	penalty cost incurred in total.
TMI	machine idle time.
Max PP	maximum penalty that is permissible.
Max TE	maximum elapsed time of the machine.

3 The Proposed Hybrid Algorithm (GAPSOTS)

3.1 GAPSOTS Algorithm Design

An effective multi-objective scheduling approach GAPSOTS is applied, which is a combination of the genetic algorithm (GA), particle swarm optimization (PSO) and tabu search (TS) algorithms (GAPSOTS algorithm). The purpose of using genetic algorithm with multiple objectives is to successfully resolve multiphase process scheduling in FMS setting. Then, PSO algorithm is applied for optimization the scheduling process and TS helps in solving combinatorial optimization issues. This new approach is made by hybrid method with multiple objectives to handle the flexible job scheduling complications with manifold goals. Investigational studies have been utilized to validate the method, and matching the results of the recommended method to specify the compliance/flexibility and supremacy of the present model does a comparative analysis. The proposed algorithms have been implemented and tested by using MATLAB R2016b or beyond, computing environment on an Intel Core™i7, with Windows 10.

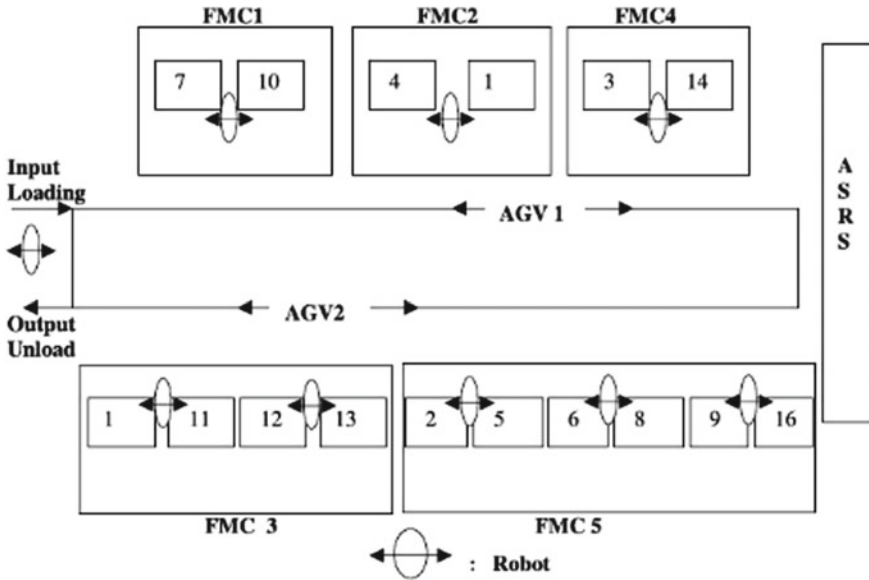


Fig. 1 FMS structure

The genetic algorithm (GA) is an adaptive heuristic exploration technique through which suitable answers to the problems of searching and optimizing can be made on the basis of a fitness function. It is a subsection of evolutionary algorithm, which provides responses to the optimizing issues by naturally stimulated algorithms and includes mutation, reproduction, and cross over, etc. In GA, there is a string of finite length called as chromosome that serves as representation of individual solution. There are position sets that are known as “genes” in a chromosome. Discrete values are given to those genes according to the solution. Crossover and mutation are genetic operators that are used to modify chromosome in order to create a new generation of “kids” that will have the characteristics of both “parents”. Every iteration results in moving towards the multi-objective optimization. Particle swarm optimization is mimicking the flock pattern of birds. In PSO, the probable solutions are termed as particles that will move in problem search space. Fitness values will be assigned to particles, which are then evaluated for fitness function that has to be optimized. Velocities will direct the path that should be taken by particles. The current feasible solutions will be pursued by the particles that will be flown in the problem space. Initialization of PSO is done randomly, and it moves towards optimum solution at each iteration by updating. The updated particles have two best values. One is “pbest”, the best solution achieved by a particular particle so far and “gbest”, the best solution achieved by any particle in the whole set (global best). Tabu search (Glover and Laguna, 1997) is a meta-heuristic method that has been effectively applied in several multi-objective optimization problems in scheduling. It is one of the popular local search algorithms that prevent the trapping of local optima. TS algorithm is

fundamentally a neighbourhood approach and offers a way to clear the inflexible combinatorial problems of optimizing. It helps to get rid of “local optima” problems for such ambience. The process of transfer of present solution to the adjoining solution is termed as move. The neighbourhood/nearby solution providing optimal solution is attained through a “move” in case of TS algorithm.

3.2 The Framework of GAPSOTS

The framework of our hybrid technique GAPSOTS is given as follows:

Step 1: Setting up of parameters.

Step 2: First initial population is generated by GA.

Step 3: The fitness function is evaluated.

Step 4: Has termination criteria been achieved?

If yes End program and display results.

Else goto Step 5.

Step 5: Generation of New populace.

Step 5.1: Application of genetic operators.

Step 5.2: Apply PSO for getting quality particles. Update global best solution.

Step 5.3: Check for termination criteria.

If yes End program.

Else goto step 5.4

Step 5.4: Generation of neighbourhood.

Step 5.5: If aspiration criterion is met, replace and update tabu list.

Step 5.6: Perform steps 5–5.5 until a termination condition is satisfied or reach maximum iteration.

The workflow process of GAPSOTS is given in Fig. 2. Detailed explanation of the proposed GAPSOTS is given in following paragraphs.

3.3 Genetic Operators for GA

To address the optimization problem, genetic operators that is used has to be very rich and so that we can achieve excellent individuals in the population. Selection, crossover and mutation are the three important genetic operators.

3.3.1 Reproduction

The method of rank selection is employed for reproduction. In rank selection method, first the entire population is ranked and the chromosomes are assigned fitness values. The least fitness will be given 1 and next will be 2 and the last will be N. The

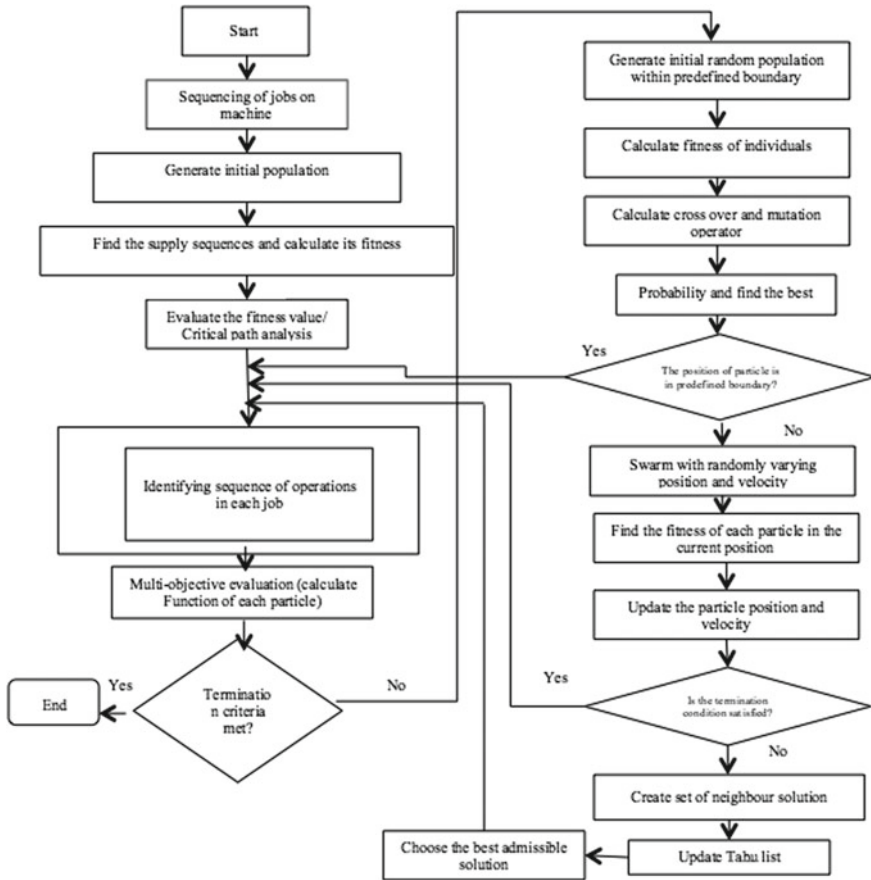


Fig. 2 Workflow process for proposed GAPSOTS

main advantage of rank selection method is it can avoid the algorithm to converge prematurely. Another advantage is that in this method the importance is given to ranks rather than the fitness values. That way the selection pressure is maintained. This method also helps to maintain diversity thereby enhancing the search. The fitness values assigned in this paper range from 0.4 to 1.6.

Once done with above calculation of every ranking, the reproduction is carried out using MATLAB simulation through employing arbitrary numbers.

3.3.2 Crossover

This operation is used after the reproduction is over. For this the “mating pool” is provided with strings. Generally, it is a convergence operation that is anticipated to have a value with regard to either “local” minimum or maximum.

- (1) Randomly select several positions of $R1$ and $R2$ and exchange the genes. Whereas other genes in other locations remain unchanged.
- (2) Check for any conflict to delete the same gene as the exchange position in the original parent gene.
- (3) Unused genes are then filled in the gene vacancy sequentially.

3.3.3 Mutation

Mutation is actually a dissimilarity task. It is just a process of tweaking the chromosome a little bit in order to maintain divergence. The mutation process is the “exploration” of the search universe. It is proposed to periodically break at least one individual from a populace out of a nearby least/greatest space and possibly find a superior least/most extreme space. A random variable is generated for each bit in a sequence to implement mutation. This random variable shows if a particular bit will be modified or not. The main objective of mutation is to avoid the chromosome population becoming too identical to each other thereby preventing the entrapment of local optima.

To boost the local search capability, the mutation operator is utilized and also maintained the population diversity ratio. The ratio of mutation probability is attained by,

$$\text{Mutate} - p = k_3 \frac{f_{\max} - f_i}{f_{\max} - f_{av}} \quad \text{if } f_i \geq f_{av} \quad (2)$$

$$\text{Mutate} - p = k_4 \quad \text{if } f_i < f_{av} \quad (3)$$

where

- $\text{Mutate}-p$ probability mutation.
- f_m maximal value of fitness in the population.
- f_{av} fitness value in average in the entire population.
- k_3 and k_4 are considered as a common value a ($0 < k_3 < 1$, $0 < k_4 < 1$).
- f_i the individual fitness value that will perform the mutation operator.

3.4 Particle Swarm Optimization (PSO) and Tabu Search (TS)

Particle swarm optimization (PSO) algorithm is a meta-heuristic technique based on evolution developed by Eberhart and Kennedy (1995). The mutation done in GA may bring us the good results. The mutation of gene is vital to maintain the divergence of population. But to improve the optimal strategies performance, we have applied PSO and tabu search algorithm.

In PSO, the individual solution is called “bird” in the searching space. They are referred to as “particles”. Each and every particle will have a value for fitness that is estimated using the function of fitness. These functions are to made optimum. There is a particular velocity to guide the movement of the particles. The particles will be flowing inside the problematic region in the direction next to the present particles with optimum functions. This algorithm has set of arbitrary particles at initial stage, which are the solutions and then begin to search for further possible optimum values by means of upgrading the generations. The particles are reorganized with new values at every step of iteration with two terms with best values, namely particles’ best, P_{best} and global best, g_{best} . One is the best fitness attained by the particle until then. The next one is the total best value of many such swarms. At every step, the velocity of every single particle is altered to stretch towards P_{best} and g_{best} . With the fresh values of velocity and location, the fitness function will be estimated with new-fangled ordinates.

The best solution with respect to the ordering structure and choosing is needed for the scheduling optimizing of flexibility oriented manufacture. The PSO and TS algorithms are hence utilized for clearing the issues related to “combinatorial” optimizing. The methodology will be adopted for arbitrarily created instances and for practical applications to exhibit the superiority of the suggested method.

The movement of the particle to find the optimal value is given by Eqs. 4 and 5

$$v_{id} = \omega \times v_{id} + G_1 \times r_1 \times (p_{idk}^{best} - p_{idk}) + G_2 \times r_1 \times (p_{gd}^{best} - p_{idk}) \quad (4)$$

$$v_{id} = \omega \times v_{id} + G_1 \times r_1 \times (p_{idk}^{best} - p_{idk}) + G_2 \times r_1 \times (p_{gd}^{best} - p_{idk}) \quad (5)$$

In Eq. (4), v_{id} means the distance moved by idk th particle for one iteration between $[-Mmax; Mmax]$ in which $Mmax$ is the maximum distance moved by a particle in one step. “ ω ” is called inertial weight that is a variable, which calculates the distance moved by a particle in each step. $G1$ is the self-learning factor, meaning a particle learn from its own knowledge p_{idk}^{best} . $G2$ is the social learning factor, meaning a particle’s knowledge globally, i.e. of the entire swarm p_{gd}^{best} . r_1 is a random number generated from an uniform distribution of $[0,1]$. p_{idk} denotes the position of idk th particle.

p_{idk}^{best} denotes the best personal value of idk th particle and p_{gd}^{best} denotes the best global value of all particles of a swarm. The newly generated velocity for idk th particle is obtained by Eq. 4, and updation of position is obtained by Eq. 5

A particle might leave the search area, to prevent this a maximum velocity factor is employed to the PSO so that the velocity is limited within a range. This allows the particle to move freely but in a constrained search area and is given by Eq. 6

$$v_{lmn}^t = \min\{M_{max}; \max\{-M_{max}; v_{lmn}^t\}\}, \quad m = 1, \dots, M; \quad l = 1, \dots, L \quad (6)$$

To achieve global optimum, a larger value of inertia weight ω is preferred and small inertia weight is preferred to explore locally and thus convergence happens

fast. In order to strike a balance between exploration and exploitation, a linear inertia weight that decreases with the convergence of algorithm evolving was adopted that has exhibited good global search ability initially and in later iterations good local search ability.

$$\omega[t] = (\omega_{\text{initial}} - \omega_{\text{final}}) \cdot \left(\frac{g - t}{g} \right)^m + \omega_{\text{final}} \quad (7)$$

where

ω_{initial} initial weight inertia
 ω_{final} final weight inertia
 g maximal number of iterations.
 $\omega_{\text{initial}} > \omega_{\text{final}}$, and $m \in [0.6; 1.4]$ are the nonlinear index [14, 15].

Then the process of tabu search (TS), which was introduced by Glover in 1986, overcomes the local optimality issues based on the evaluation function that has been selected with the highest evaluation solution at each iteration. The initial population of tabu search is done by taking the result of PSO, and some random solutions are introduced. Next step is neighbourhood structure where the existing solutions will be modified to obtain new solutions. The movement of solutions is accepted on the basis of probability function. A tabu list is created to store the properties of the accepted moves, and they will be avoided in later iterations thus called “taboo”. In the new neighbourhood, the best neighbour (solution) will be chosen that is not a tabu. This is applied to avoid cyclic movements. A strategy called forbidding is employed to control and update the tabu list. The parameters used for this hybrid technique GAPSOTS are given in Table 1.

Table 1 Parameters of proposed hybrid technique—GAPSOTS

Parameters	Value
Population size	500
Crossover probability	0.5
Mutation probability	0.9
The total number of generations	200
Length of tabu list	10
The maximum iteration size	800
Number of particles	20

Table 2 Part mix with due date and penalty cost

Part number	Processing sequence: Machine number (processing time in minutes)	Due date (days)	Batch size (No.)	Penalty (Rs/unit/day)
1	6(1)-7(1)-8(1)-10(2)	17	150	1.00
2	2(1)-6(1)-8(2)-9(2)-14(4)-16(2)	17	200	1.00
3	8(1)-11(3)-13(4)	14	800	1.00
4	9(4)	26	700	2.00
5	4(5)-5(3)-15(4)	11	150	1.00
6	6(5)-14(1)	16	700	1.00
7	3(5)-6(3)-16(5)	26	250	2.00
8	5(4)-6(5)-8(1)	26	850	2.00
9	4(1)-5(5)-8(1)-11(1)	1	100	0.00
10	2(2)-9(1)-16(4)	20	150	2.00
11	8(4)-12(2)	1	250	1.00
12	6(2)-8(4)-10(1)	19	1000	3.00
13	6(1)-7(5)-10(4)	25	700	400
14	4(2)-5(3)-6(2)-15(2)	22	1000	400
15	5(4)-8(3)	15	700	5.00
16	5(3)	27	750	3.00
17	3(1)-6(4)-14(1)	20	650	5.00
18	9(2)-6(3)	24	250	400
19	4(1)-5(5)-6(2)-8(2)-15(5)	5	450	1.00
20	8(2)-11(4)	11	50	5.00
21	4(5)-5(5)-8(4)-15(4)	16	850	3.00
22	12(5)	24	200	5.00
23	4(2)-5(1)-6(5)-8(4)	14	50	400
24	8(4)-11(4)-12(5)-14(4)	7	200	5.00
25	7(3)-10(2)	24	350	1.00
26	10(2)	27	450	0.00
27	8(5)-11(5)-12(4)	22	400	1.00
28	2(1)-8(1)-9(2)	3	950	5.00
29	4(1)-5(5)	7	700	1.00
30	11(3)-12(5)	18	1000	1.00
31	8(2)-10(2)	2	800	2.00
32	2(3)-6(4)-9(3)	15	800	1.00
33	5(4)-6(5)-15(3)	27	500	4.00
34	3(2)-6(2)	12	300	4.00
35	3(4)-14(1)	9	900	2.00

(continued)

Table 2 (continued)

Part number	Processing sequence: Machine number (processing time in minutes)	Due date (days)	Batch size (No.)	Penalty (Rs/unit/day)
36	3(2)	20	700	2.00
37	1(5)-2(2)-6(3)-8(3)-9(2)-16(4)	22	250	4.00
38	2(4)-8(3)-9(2)-16(5)	8	50	1.00
39	6(5)-10(5)	9	500	1.00
40	2(2)-6(4)-9(4)	7	250	5.00
41	5(1)-8(2)-15(1)	22	800	4.00
42	2(5)-6(4)-9(3)-16(1)	19	400	2.00
43	1(3)-5(2)-6(2)-8(2)-15(3)	15	550	3.00

4 Results and Discussions

The FMS problem considered in this work is taken from existing literature [6]. The hybrid procedure GAPSOTS is developed for 43 jobs and 80 jobs for 16 machines using combined objective optimization method. The problem matrix taken for this problem is given in Table 2. MATLAB software version R2016b and above is used for programming. The MATLAB helps the user to solve complex scheduling problems. TOMLAB/CPLEX is also used as it is a subproblem solver and handles quicker solving of mixed-integer linear and quadratic programming (MILP,MIQP), and linear and quadratic programming (LP,QP). Feasible schedule has been achieved for the combinatorial scheduling using the above hybrid approach. A comparison between the proposed GAPSOTS and other algorithms, namely GA, SA, memetic algorithm, cuckoo search (found in the literature) has been analysed and shown in Table 3 and Fig. 3 for 43 jobs 16 machine problem and Fig. 4 for 80 jobs 16 machine problem. The proposed algorithm has been run for different iterations. When the iterations numbers are less the results were not good, but as iterations increased we got better results. We stopped at 800 iterations, as there was no significant change in optimal value later.

5 Conclusion

The following points summarize the authors contribution made in this paper.

- A new hybrid technique named GAPSOTS has been developed by hybridizing GA, PSO, TS. A global search technique has been combined with two local search methods thereby balancing the drawbacks of the said techniques.

Table 3 Performance comparison of GAPSOTS with various techniques

Problem size	Objectives	Genetic algorithm	Simulated annealing	Memetic algorithm	Cuckoo search	GAPSOTS
43 jobs × 16 machines	COF	0.274	0.45936	0.35136	0.2646	0.11967
	Sequence	21; 28; 16; 42; 39; 25; 5; 23; 22; 35; 36; 37; 15; 1; 2; 9; 4; 3; 40; 20; 31; 11; 14; 30; 27; 43; 17; 10; 7; 26; 33; 18; 29; 8; 41; 32; 13; 12;	36 25 14 39 1 33 27 3 13 4 18 22 0 42 28 35 30 38 8 26 11 24 23 41 13 24 37 10 32 9 20 12 9 6 16 17 15 21 7 29 5	10 35 18 27 3 42 5 19 2 39 83 34 30 0 26 36 22 28 38 7 34 11 31 24 23 1 32 1 14 12 5 15 20 9 16 41 21 6 17 29 40 7	8, 14, 28, 31, 3, 42, 26, 33, 22, 20, 5, 24, 12, 41, 18, 7, 10, 19, 23, 38, 4, 35, 40, 37, 15, 15, 17, 39, 6, 2, 34, 1, 29, 27, 16, 36, 30, 25, 32, 13, 43, 11, 10, 9	30 29 8 26 13 31 38 24 37 20 33 7 11 36 23 12 39 18 34 42 35 21 28 36 17 25 5 4 15 22 9 40 2 1 27 14 10 32 19 41 16 43
80 jobs × 16 machines	COF	0.0959237	0.14738715	0.1067266	-	0.08537
	Sequence	34, 50, 59, 76, 48, 69, 31, 11, 1, 26, 56, 23, 47, 25, 62, 20, 3, 78, 55, 53, 24, 51, 70, 71, 75, 46, 58, 74, 4, 2, 28, 18, 38, 22, 54, 5, 72, 12, 13, 40, 10, 60, 65, 79, 43, 37, 66, 27, 42, 7, 64, 61, 29, 35, 36, 73, 39, 49, 67, 2, 41, 15, 17, 44, 45, 63, 21, 6, 80, 9, 19, 57, 14, 33, 16, 68, 52, 77, 8, 32	41, 53, 51, 70, 1, 71, 47, 58, 36, 61, 30, 4, 23, 54, 75, 77, 76, 52, 79, 31, 15, 78, 35, 59, 66, 72, 55, 56, 64, 28, 40, 13, 45, 17, 14, 48, 80, 49, 34, 25, 10, 20, 44, 29, 11, 60, 67, 38, 27, 9, 12, 32, 46, 5, 18, 43, 65, 37, 24, 21, 42, 2, 63, 19, 7, 22, 33, 74, 26, 57, 16, 62, 3, 39, 69, 6, 50, 73, 68, 8	69, 17, 36, 24, 74, 61, 20, 35, 75, 49, 25, 18, 58, 47, 3, 52, 28, 45, 51, 59, 46, 6, 16, 34, 31, 53, 70, 76, 22, 2, 40, 79, 15, 30, 62, 54, 38, 10, 43, 13, 80, 37, 71, 26, 66, 73, 72, 65, 56, 67, 12, 1, 4, 5, 21, 32, 77, 27, 29, 63, 33, 60, 64, 57, 41, 9, 14, 48, 55, 78, 7, 42, 11, 23, 39, 8, 50, 68, 19, 44	-	39, 26, 8, 32, 35, 78, 59, 27, 51, 18, 19, 31, 52, 71, 57, 5, 1, 74, 22, 13, 63, 58, 33, 34, 65, 29, 48, 42, 70, 17, 9, 56, 40, 49, 67, 30, 3, 66, 7, 45, 11, 23, 80, 38, 16, 46, 44, 21, 60, 75, 24, 10, 4, 53, 47, 15, 76, 20, 6, 54, 72, 43, 64, 50, 2, 14, 73, 25, 69, 77, 41, 28, 79, 61, 55, 68, 36, 12, 37, 62

Fig. 3 Comparison of COF for 43 jobs by different approaches

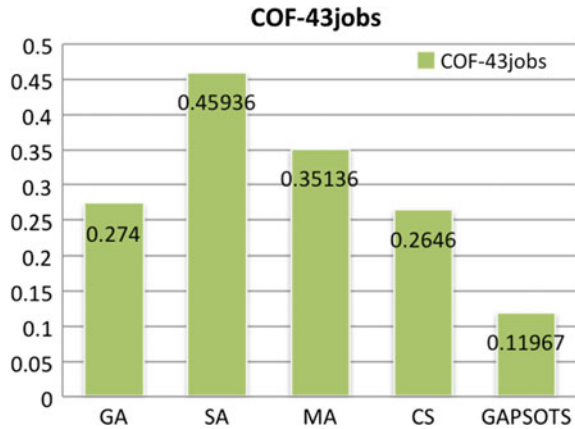
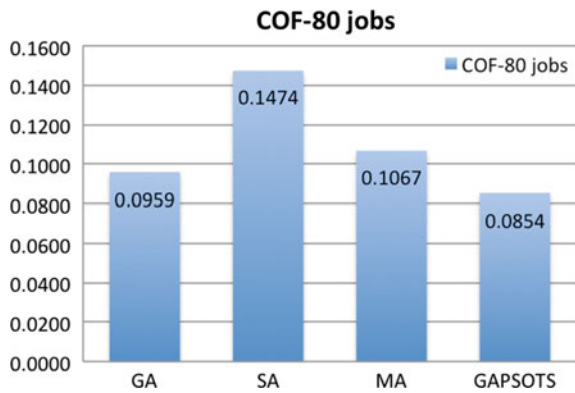


Fig. 4 Comparison of COF for 80 jobs by different approaches



- The developed technique was tested for benchmark problems taken from the literature, which consists of 2 problems of 43 jobs \times 16-machine problem and 80 jobs \times 16 machine problem.
- A less solved combinatorial optimization problem for minimizing the total penalty cost and total machine idleness has been evaluated.
- Optimal results were obtained with less computational time, and the results were found to be superior to other approaches found in the literature. The difference we got is subtle yet the results were promising to explore other adaptations of GA, PSO, TS.
- The main observation is with less number of generations we got a better optimal value. With adaptations made in genetic operators, the premature convergence of algorithm has been prevented.
- In future, the proposed GAPSOTS can be tested for integrated scheduling FMS and can be implemented for any FMS with different set-ups.

References

1. Mathew N, Saravanan NR (2012) Evaluation of genetic algorithm approach for scheduling optimization of flexible manufacturing systems. *Int J Eng Res Appl* 2:437–446
2. Stecke KE (1985) Design, planning, scheduling, and control problems of flexible manufacturing systems. *Ann Oper Res* 3:1–12. <https://doi.org/10.1007/BF02023765>
3. Zhao C, Wu Z (2001) A genetic algorithm approach to the scheduling of FMSs with multiple routes. *Int J Flex Manuf Syst* 13:71–88. <https://doi.org/10.1023/A:1008148313360>
4. Kumar R, Tiwari MK, Shankar R (2003) Scheduling of flexible manufacturing systems: an ant colony optimization approach. *Proc Inst Mech Eng Part B J Eng Manuf* 217:1443–1453. <https://doi.org/10.1243/095440503322617216>
5. Noorul Haq A, Karthikeyan T, Dinesh M (2003) Scheduling decisions in FMS using a heuristic approach. *Int J Adv Manuf Technol* 22:374–379. <https://doi.org/10.1007/s00170-002-1474-0>
6. Jerald J, Asokan P, Prabakaran G, Saravanan R (2005) Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm. *Int J Adv Manuf Technol* 25:964–971. <https://doi.org/10.1007/s00170-003-1933-2>
7. Gnanavel Babu A, Jerald J, Noorul Haq A et al (2010) Scheduling of machines and automated guided vehicles in FMS using differential evolution. *Int J Prod Res* 48:4683–4699. <https://doi.org/10.1080/00207540903049407>
8. Udhayakumar P, Kumaran S (2010) Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm. *Int J Adv Manuf Technol* 50:1075–1084. <https://doi.org/10.1007/s00170-010-2583-9>
9. Kumar AVSS, Veeranna V, Durgaprasad B, Sarma BD (2013) Combined objective optimization of FMS scheduling with non-traditional optimization techniques. *Int J Eng Res Technol (IJERT)* 2:1420–1428
10. Nidhish Mathew Nidhiry DRS (2012) A combined objective genetic algorithm for scheduling of machines in FMS. *Int J Eng Res Appl*
11. 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. <https://doi.org/10.1007/s00170-005-0223-6>
12. Burnwal S, Deb S (2013) Scheduling optimization of flexible manufacturing system using cuckoo search-based approach. *Int J Adv Manuf Technol* 64:951–959. <https://doi.org/10.1007/s00170-012-4061-z>
13. Chawla VK, Chanda AK, Angra S (2019) The scheduling of automatic guided vehicles for the workload balancing and travel time minimization in the flexible manufacturing system by the nature-inspired algorithm. *J Proj Manag* 4:19–30. <https://doi.org/10.5267/j.jpm.2018.8.001>
14. Chatterjee A, Siarry P (2006) Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput Oper Res* 33:859–871. <https://doi.org/10.1016/j.cor.2004.08.012>
15. Durand FR, Abrão T (2013) Particle swarm optimization in WDM/OCDM networks with physical impairments. *J Microwaves Optoelectron Electromagn Appl* 12:336–352. <https://doi.org/10.1590/S2179-10742013000200008>