Effective Utilization of Job Shop Scheduling in Auto Industries with the aid of Social Spider Optimization

K. B. Gavali¹, Anand K. Bewoor² and Debabrata Barik³

¹*Research Scholar, Department of Mechanical Engineering, Karpagam Academy of Higher Education, Coimbatore-641021, India*

²*Department of Mechanical Engineering, Cummins College of Engineering, Pune-411052, India*

³*Department of Mechanical Engineering, Karpagam Academy of Higher Education, Coimbatore-641021, India*

E-mail: gavaliphd@gmail.com

Received 20 June 2018; Accepted 05 November 2018;

Publication 10 December 2018

Abstract

For promising outcome in auto industries, scheduling place a vital role for effective utilization of jobs allocate to machine. Jobs and machines are two attributes need to schedule for minimize makespan time, for each job we need to schedule the machine. Each job in a machine has its own process time, manipulation of all process time said to be makespan time that should minimized. Job shop scheduling is an effective tool incorporate with NP hard problems to achieve minimized makespan time. To achieve minimized makespan time optimization involve in this process those are namely Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO) and Social Spider Optimization (SSO). While applying these aforementioned optimization techniques, they reveal minimized makespan time compare with benchmark problems. Amid, SSO reveals minimized makespan time for all twelve-bench mark problems compare with other competitive algorithm namely GWO and PSO. These techniques play a vital role to regulate the emissions during real time auto industries trial and error process on job shop

Journal of Green Engineering, Vol. 8_4, 475–496. River Publishers

doi: 10.13052/jge1904-4720.842

This is an Open Access publication. © 2018 the Author(s). All rights reserved.

scheduling. Therefore, the soft computing techniques contribute significant part on conserving environment from air pollution.

Keywords: Makespan time, Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO) and Social Spider Optimization (SSO).

1 Introduction

The automotive industry gives one of a kind and rich setting for contemplating the issue of development of request discharges. Their exactness in light of the particular desires that the automobile original equipment manufacturers (OEMs) have of their suppliers [1]. A normal traveller vehicle from a volume automotive maker can be offered in a few body styles, motor sorts, transmissions, colors, trims, and other selectable choices, creating an amazingly extensive number of potential variations. Despite the fact that clients fluctuate in their optimal conveyance lead-time, reviews demonstrate that automotive makers experience issues in satisfying clients with their craved item variations and depend on offering budgetary motivators to clients to bargain on their specification [2].

The goal of this study is to enhance the comprehension of planning and scheduling along request satisfaction in the automotive industry. We give a general investigation of procedures, exercises, and undertakings identified with planning and scheduling at vehicle creators by applying an arrangement technique as for the time skyline, arranging recurrence and short practical description [3]. Companies having a place with various mechanical segments may utilize diverse methodologies to build their item portfolios [4]. Many specialists have built up various scheduling approaches for machine-obliged fabricating frameworks, yet they dismissed critical imperatives of accessible workers [5]. The Scheduling issue has a vital influence, recently, on account of the shaking purchaser interest for assortment, downsized item life cycles, regularly shifting markets with worldwide rivalry and quick growth of the refined innovations [6].

In job shop scheduling problem (JSSP), the goal is to assign assets as it were, to such an extent that various errands can be finished cost-viably inside a given arrangement of imperatives [7]. Regarding computational intricacy, JSSP is NP-hard in the solid sense. In this manner, notwithstanding for little JSSP cases, it is in no way, shape or form simple to ensure the optimal solution [8]. JSSP is a standout amongst the most broadly examined issues in software engineering, which has incredible significance to assembling

industries with the target to minimize the creation cost. The job shop-scheduling problem (JSP) is one of most vital and troublesome issues in the field of production scheduling [9, 10]. Production scheduling problems include the assignment of rare assets to various undertakings in such path as to upgrade at least one proficiency related objectives [11]. In most cases, these issues are broke down as cases of the Job-Shop Scheduling Problem (JSSP), in which given an arrangement of machines and a rundown of jobs, spoke to as requested successions of operations, to be keep running on the machines, the objective is to minimize, specifically, the completion (processing time) of all employments, known as makespan [12, 13]. Most research on JSSP has concentrated eager for advancement span model (i.e., minimizing the greatest fruition time). Nevertheless, in the make-to-order (MTO) producing an environment, due date related exhibitions clearly more important for leaders, on the grounds that the in-time conveyance of merchandise is indispensable for keeping up a high administration notoriety. Along these lines, the exploration that goes for minimizing delay/lateness in JSSP merits more consideration [14, 15].

Job-shop scheduling, fundamentally, constitutes a lively NP-finish challenge. Various agents have imagined a few JSP models as per the issue hypotheses [16]. The makespan speaks to the most extreme consummation term of the jobs. The basic motivation behind the JSP is to find a schedule, which fundamentally minimizes the makespan and a decent timetable, thus, requires the compelling minimization of the idle time spent by the machines [17]. It is superbly substantial to expect that employment-handling times are deterministic for some scheduling environments for which algorithm can be utilized to solve the problem [18]. Social Spider Optimization (SSO) defines the inquiry space as the web on which every position speaks to a possible solution for the optimization problem. In addition, every single attainable answer for the issue has relating positions on this web. SSO is utilized for settling obliged optimization problems [19]. The execution evaluation of the optimization strategies is established on the computational insight ideal models for settling the job shop scheduling problem to accomplish the optimized execution time, to diminish the makespan of the job arrangement, along these lines upgrading the practicality of the scheduling method [20]. Leila Asadzadeh [21] had proposed the job shop scheduling problem (JSSP) has pulled in much thought in the field of both information sciences and operations ask about. In that procedure, artificial bee colony algorithm involves a couple expresses that arrange on different hosts of the framework and algorithm finished in various settlements in a parallel method.

The results show that the proposed parallel artificial bee colony algorithm improves the profitability. Hamed Piroozfard *et al.* [22] had arranged that paper delineates the progression of a hybrid genetic algorithm for clarifying the non-preemptive job shop scheduling problems with the objective of minimizing make span. Computational outcomes of the proposed hybrid genetic algorithm demonstrate its sufficiency. Yao Wang *et al.* [23] had proposed a swarm algorithm, called social spider algorithm (SSA), to illuminate the Flexible Job-Shop Scheduling Problem (FJSP). The computational outcome demonstrates that the proposed algorithm delivers preferable outcomes over other creators' algorithms. Adil Baykasoglu *et al.* [24] had foreseen the execution of the TLBO algorithm attempted on some combinatorial change issues, to be particular flow shop (FSSP) and JSSP. Scheduling issues were among the most perplexed combinatorial optimization issues. Investigate comes concerning showing that the TLBO algorithm has an amazing potential once diverged from the best-known heuristic algorithms for scheduling problems.

The primary objective of this proposed strategy is to reduce the time consumed in auto industries, as we probably am aware a few individual procedures fused for the result of complete vehicles. Two essentialness features include in this procedure are job and machine, these two traits ought to be scheduled in the correct allotment for the advancement of minimized makespan time. For scheduling these two features, job shop scheduling procedure is used. For every job, we have to schedule the machine. Every job in every machine has its own particular procedure time, control of all procedure time said to be makespan time that ought to minimize. Job shop, a set of independent jobs processes a sequence of operations on an arrangement of accessible machines for the particular scheduling. The principle target of the job shop scheduling is to minimize the makespan time and find the optimal schedule of the operations. In the event that this procedure happens in manual will took enormous time to complete for the extraction of optimal solution with a specific end goal to minimize the event of huge time consuming, incorporate optimization algorithm, for example, social spider optimization (SSO), Gray wolf optimization (GWO) and Particle Swarm Optimization (PSO) algorithm. By applying these three different optimization techniques, enhanced reveal minimized makespan time optimization technique will consider as proposed technique. The following section incorporates detail description of Social Spider Optimization in Section 2, results and discussion in Section 3 eventually windup with research conclusion.

2 Social Spider Optimization (SSO)

In SSO, the computation of search space optimization problem as hyper-dimensional spider web. Each position on the web signifies a feasible solution to the optimization problem and all feasible solution to the problem has corresponding positions on this web. The web also serves as the transmission media of the vibrations engendered by the spiders. Each spider on the web holds a position and the quality of the solution is based on the fitness function and signified by the possible of finding a food source at the position. The spiders can travel spontaneously on the web. Nevertheless, they cannot leave the web as the portions off the web represent infeasible solution to the optimization problem. When a spider moves to a new position, it generates a vibration, which is propagated over the web. Each vibration holds the information of one spider and other spiders can get the information upon receiving the vibration. This algorithm comprised of two separate spiders namely males and females, every individual gender demeanor by a set of special evolutionary operators, which imitate diverse cooperative behaviors that are frequently suppose within the colony shown in Figures 1 and 2.

Figure 1 Social Spider Optimization algorithm process.

Figure 2 Flowchart for Social Spider Optimization algorithm.

An attractive characteristic of social-spiders are the extremely female-biased populations. The algorithm begins by characterizing the number of female and male spiders that will portrayed as individuals in the search space. The number of females S_{Nf} randomly chose inside the scope of 65–90% of the entire population N_S . Consequently, S_{Nf} computed by the accompanying condition:

$$S_{Nf} = \text{floor} [(0.9 - \text{rand} \cdot 0.25) \cdot N_S] \quad (1)$$

Though floor () maps a genuine number to an integer number where rand is an irregular number between [0, 1]. The number of male spiders S_{Nm} processed as the supplement amongst N_S and S_{Nf} . It is illustrate below,

$$S_{Nm} = N_S - S_{Nf} \quad (2)$$

Hence, the complete population S_P , composed by N_S elements divided in two sub-groups Female (F) and Male (M).

2.1 Initialization

This initial step comprised of randomly shuffled machine allocate to a job consider as one solution, likewise ten randomly generate solution are made those are consider as male and female operators. The algorithm starts by introducing the set S_P of N_S spider positions. Such values are randomly and consistently distributed between the pre-indicated upper initial parameter bound pr_j^{high} and the lower initial parameter bound pr_j^{low} , also as it portrayed by the accompanying expressions.

$$f_{i,j}^0 = pr_j^{low} + \text{rand}(0, 1) \cdot (pr_j^{high} - pr_j^{low}) \quad (i = 1, 2 \dots S_{Nf}, j = 1, 2, \dots n) \quad (3)$$

$$m_{k,j}^0 = pr_j^{low} + \text{rand}(0, 1) \cdot (pr_j^{high} - pr_j^{low}) \quad (k = 1, 2 \dots S_{Nm}, j = 1, 2, \dots n) \quad (4)$$

While, zero signals the initial population; j , i and k are the parameter and individual index in that order and the function rand generate random values between (0,1). Henceforth, $f_{i,j}$ is the j^{th} parameter of the i^{th} female spider position. By estimating the radius of mating at that point,

$$r = \frac{\sum_{j=1}^n (pr_j^{high} - pr_j^{low})}{2 \cdot n} \quad (5)$$

2.2 Fitness Function

This objective function is utilized to evaluate the performance of above generated random shuffle job-machine solution. The following functions F_i compute to evaluate the performance of random shuffled solution. Eventually, minimum value attained from the fitness is considered to be optimal makespan time and corresponding solution is considered as optimal job-machine scheduling.

```

MST=Zeros (size of problem)
for
row i=1 : Number of jobs
for
column j=1 : Number of machine
if i=j=1
MST (i,j) = spider (i,j)
else if
i≠1 and j=1
MST (i,j) = sum (spider (1: i,j))
else if
i=1 and j≠1
MST (i,j) = sum (spider (i,1: j))
else
Temp 1 = MST (i-1,j)
Temp 2 = MST (i,j-1)
if Temp 1 ≤ Temp 2
MST (i,j) = Temp 2 + spider (i,j)
else
MST (i,j) = Temp 1 + spider (i,j)
end
end
end

Fitness ( $F_i$ ) = MST (end)

```

Where, MST is makespan time

From the aforementioned computation, the result attain for makespan time then weight (w_i) value is use to calculate for obtained fitness to enhance the probability of increasing chance to generate fine tune solution for next iterations in order to attain optimal makespan time in short interval.

$$w_i = \frac{J(S_{pi}) - worst_{S_p}}{best_{S_p} - worst_{S_p}} \quad (6)$$

Where $J(S_{pi})$ is the fitness probability analysis got by the assessment of the spider position S_{pi} concerning the objective function $J(\cdot)$. The values $worst_{S_p}$ and $best_{S_p}$ characterized as takes after (considering a minimization problem)

$$best_{S_p} = \min_{k \in \{1,2,\dots,N\}} (J(S_{pk})) \text{ and } worst_{S_p} = \max_{k \in \{1,2,\dots,N\}} (J(S_{pk})) \quad (7)$$

2.3 Modelling the Vibrations via Communal Web

To transmit data among the colony members, the common web utilized as a component. The vibrations rely on the weight and distance of the spider, which has created them. With a specific end goal to duplicate this procedure, the vibrations saw by the individual i due to the data transmitted by the member j demonstrated by the accompanying condition.

$$Vib_{i,j} = w_j \cdot e^{-d_{i,j}^2} \quad (8)$$

Where the $d_{i,j}$ is the Euclidian distance between the spiders i and j , such that $d_{i,j} = ||S_{pi} - S_{pj}||$.

2.4 Cooperative Operators

There are two types of cooperative operator's namely female cooperative operator and male cooperative operator where discussed below.

2.4.1 Female cooperative operator

A new operator is characterizing in order to imitate cooperative behavior of the female spider. At every iteration, the operator considers the position change of the female spider i . Such position change might attract or surprise the process as a mix of three unique components. The first includes the change concerning the closest part to i that holds a higher weight and creates the vibration $Vibo_i$. The second one considers the change in regards to the best individual of the entire population S_p who delivers the vibration $Vibg_i$. Finally, the third one incorporates a random movement.

$$Vibo_i = w_o \cdot e^{-d_{ii,o}^2}, \quad Vibg_i = w_g \cdot e^{-d_{i,g}^2} \quad (9)$$

A uniform random number r_m created inside the range $[0, 1]$. On the off chance, that r_m is smaller than a threshold PF, a fascination development created; generally, a repulsion movement is delivering. Along these lines, such operator can be display as illustrate below,

$$f_i^{k+1} = \begin{cases} f_i^k + \alpha \cdot Vibo_i \cdot (S_{po} - f_i^k) + \beta \cdot Vibg_i \cdot (S_{pg} - f_i^k) \\ \quad + \delta \cdot (rand - 1/2) \text{ with probability } PF \\ f_i^k - \alpha \cdot Vibo_i \cdot (S_{po} - f_i^k) - \beta \cdot Vibg_i \cdot (S_{pg} - f_i^k) \\ \quad + \delta \cdot (rand - 1/2) \text{ with probability } 1 - PF \end{cases} \quad (10)$$

Though k represents the iteration number where α , β , δ and $rand$ are random numbers between $[0, 1]$. The individual S_{po} and S_{pg} speak to the closest part to i that holds a higher weight and the best individual of the entire population S_p , separately.

2.4.2 Male cooperative operator

Male individuals, with a weight value over the middle value inside the male population, are view as the predominant people D . Then again, those under the middle value are naming as non-prevailing ND males. In order to implement such calculation, the male population M ($M = \{m_1, m_2, \dots, m_{Nm}\}$) is organize by their weight value in diminishing request. The Vibration $Vibq_i$ saw by that individual $i(D_i)$ because of the data passed on by the part $q(D_q)$ with q consistently the nearest female unmistakable to i .

$$Vibq_i = w_q \cdot e^{-d_{i,q}^2} \quad (11)$$

Since lists of the male population M with respect to the entire population S_p are expanded by the number of female members S_{Nf} , the middle weight is ordered by S_{Nf+m} . As indicated by this, change of positions for the male spider can demonstrate as follow.

$$m_i^{k+1} = \begin{cases} m_i^k + \alpha \cdot Vibq_i \cdot (S_{pf} - m_i^k) + \delta \cdot (rand - 1/2) \text{ if} \\ \quad w_{S_{Nf+i}} > w_{S_{Nf+m}} \\ m_i^k + \alpha \cdot \left(\frac{\sum_{h=1}^{S_{Nm}} m_h^k \cdot w_{S_{Nf+h}}}{\sum_{h=1}^{S_{Nm}} w_{S_{Nf+h}}} - m_i^k \right) \text{ if} \\ \quad w_{S_{Nf+i}} \leq w_{S_{Nf+m}} \end{cases} \quad (12)$$

While, $(\sum_{h=1}^{S_{Nm}} m_h^k \cdot W_{SNf+h} / \sum_{h=1}^{S_{Nm}} W_{SNf+h})$ compare to the weighted mean of the male population M , where the individual S_{pq} speaks to the closest female individual to the male part i .

Two unique behaviors created by utilizing this administrator. Initially, set D of particles is magnetizing with others in order to rouse mating and such a performance permits incorporation diversity in to the population. On other hand, the set ND of particles magnetize by means of male population M . This reality is utilizing to somewhat control the search procedure as indicated by the normal execution of a subgroup of the population. Such system goes about as a filter, which maintains a strategic distance from that great person or extremely bad person, affects the search procedure.

2.5 Mating Process

In the mating procedure, the weight of each included spider (elements of T^g) characterizes the probability of impact for every person into the new brood. The spiders holding a heavier weight will probably affect the new product, while components with lighter weight have a lower probability. The impact probability P_{Spi} of each member is allocate by roulette strategy, which is characterized as follow.

$$P_{Spi} = \frac{w_i}{\sum_{j \in T^k} w_j} \quad \text{where } i \in T^g \quad (13)$$

Once the new spider is created, it is contrasted with the new spider candidate S_{pnew} holding the worst spider S_{pwo} of the colony, as indicated by their weight values (where $w_{wo} = \min_{l \in \{1,2,\dots,N\}}(w_l)$). In the event that the new spider is superior to the most exceedingly worst spider, the new one replaces the most noticeably worst spider. Something else, the new spider is disposed of and the population does not endure changes. If there should arise an occurrence of substitution, the new spider expects the gender and index from the supplanted spider. Such reality guarantees that the entire population S_p keeps up the first rate amongst female and male individuals. Under this operation, newly created particles locally abuse the search space inside the mating range in order to discover better people.

3 Results and Discussion

The results obtain from different analysis illustrate beneath, those analysis comprised of problem wise makespan time comparison for different

Figure 3 Comparative convergence graph for ft20, la21 and la27.

techniques compared with benchmark solution. From this analysis, it is quite evident that the proposed social spider optimization reveals minimized makespan time compare with other techniques. Analyzing with different problems carrying different size, the proposed SSO reveal almost similar makespan time compare with benchmark result and superior makespan time results than contest techniques. The whole process is implement in the working platform MATLAB with i3 processor 8GB RAM having CPU speed of 2.20 GHz.

In this Figure 3, the comparative convergence shown for ft20 problem size (20×5) , la21 problem size (15×10) and la27 problem size (20×10) . At this point, three optimization algorithms employed such as SSO, GWO and PSO. To predict better scheduling process in these problems, beyond three algorithms are used to minimize the makespan time. From the above graph for each problem the SSO technique converge the fitness value drastically in consecutive iterations and finally reach its minimized makespan time.

In this Figure 4, the comparative convergence appeared for la01 problem size (10×5) , la02 problem size (10×5) , la03 problem size (10×5) and la04 problem size (10×5) . Now, three optimization algorithms utilized,

Figure 4 Comparative convergence graph for la01, la02, la03 and la04.

for example, SSO, GWO and PSO. To anticipate better scheduling process in these problems, beyond three algorithms utilized to minimize the makespan time. From the above diagram for every problem the SSO procedure converge the fitness value radically in successive iterations lastly achieve its minimized makespan time. Evidently revealed that the proposed SSO algorithm is performed better while contrasted and GWO and PSO in this scheduling problems.

Figure 5, illustrate average makespan time for 12 benchmark problems for different optimization techniques along SSO. The overall performance of SSO exhibit 99.77% results HGA1 algorithm reveals 99.75%, GWO algorithm reveals 99.62%, HGA algorithm reveals 99.54%, HGA2 algorithm reveals 99.47%, MA algorithm reveals 99.47%. In PGA algorithm the average value is 98.97%, F&F algorithm exposes 98.83%, MA(GR-RS) algorithm exposes 98.65%, TLBO algorithm exposes 98.54%, GRASP algorithm exposes 97.87%, PSO algorithm exposes 97.71%, Beam Search algorithm expose 96.74% and finally the HIA algorithm exposes 93.41%. Based on this overall performance the proposed SSO algorithm achieves higher average value.

Figure 5 Comparative convergence graph for la06, la07 and la09.

In Figure 6, the comparative convergence appeared for la36 problem size (15×15), and la40 problem size (15×15). Now, three optimization algorithms utilized, for example, SSO, GWO and PSO. To anticipate better scheduling process in these problems, beyond three algorithms are utilized to minimize the makespan time. From the above diagram for every problem, the SSO procedure converge the fitness value radically in successive iterations lastly achieve its minimized makespan time.

From Tables 1 and 2, illustrate the makespan time achieve from different techniques along with benchmark problem makespan results. Here, SSO contest with 13 different optimization techniques (evolutionary and swarm intelligence techniques) in the platform of predicting minimize makespan time. From Tables 3 and 4, it is evident that SSO have superior percentage in major benchmark problem over contest techniques. Here, both swarm intelligence and evolutionary technique perform equally better in entire benchmark problems amid SSO leads slightly better over HGA1. Modifying evolutionary technique make a remarkable change while refurbishing conventional objective function it's evident on modified evolutionary and swarm intelligent techniques. SSO perform literally well on small and medium size problem over big problem size. However, SSO behave literally well in big problem size over contest techniques.

Figure 6 Comparative convergence graph for la36 and la40.

Table 1 Comparison of BM with different techniques for scheduling problems

Problems	BM [24]	SSO	HGA1 [24]	GWO [24]	HGA [24]	HGA2 [24]	MA [24]	PGA [24]
FT20	1165	1165	1165	1165	1165	1165	1165	1177
LA01	666	666	666	666	666	666	666	666
LA02	655	655	655	655	655	655	655	666
LA03	597	597	597	600	597	597	597	597
LA04	590	590	590	590	590	590	590	590
LA06	926	926	926	928	926	926	926	926
LA07	890	890	890	890	890	890	890	890
LA09	951	951	951	951	951	951	951	951
LA21	1046	1046	1046	1049	1046	1046	1055	1047
LA27	1235	1237	1236	1239	1256	1256	1261	1260
LA36	1268	1277	1287	1280	1279	1287	1281	1305
LA40	1222	1236	1229	1240	1241	1241	1233	1252

From this Figure 7, the average makespan time for different algorithms evaluated obviously. There are diverse categories of algorithms utilized such as SSO, HGA1, GWO, HGA, HGA2, MA, PGA, F&F, MA(GR-RS),

Table 2 Comparison of BM with different techniques for scheduling problems

Problems	BM [24]	F&F [24]	MA(GR-RS) [24]	TLBO [24]	GRASP [24]	PSO [24]	Beam Search [24]	HIA [24]
FT20	1165	1165	1165	1165	1169	1165	none	1165
LA01	666	666	666	666	666	666	666	666
LA02	655	655	655	655	655	668	704	655
LA03	597	597	597	597	604	606	650	597
LA04	590	590	590	607	590	611	620	590
LA06	926	926	926	926	926	926	926	926
LA07	890	890	890	890	890	890	890	890
LA09	951	951	951	951	951	951	951	951
LA21	1046	1052	1079	1091	1091	1082	1152	1256
LA27	1235	1242	1286	1256	1320	1290	1316	1784
LA36	1268	1381	1307	1332	1334	1332	1401	1281
LA40	1222	1228	1252	1241	1259	1286	1347	1240

Table 3 Percentage for each scheduling problems using different techniques compared with BM

Problems	SSO	HGA1	GWO	HGA	HGA2	MA	PGA
FT20	100	100	100	100	100	100	98.98
LA01	100	100	100	100	100	100	100
LA02	100	100	100	100	100	100	98.34
LA03	100	100	99.50	100	100	100	100
LA04	100	100	100	100	100	100	100
LA06	100	100	99.78	100	100	100	100
LA07	100	100	100	100	100	100	100
LA09	100	100	100	100	100	100	100
LA21	100	100	99.71	100	100	99.14	99.90
LA27	99.83	99.91	99.67	98.32	98.32	97.93	98.01
LA36	99.29	98.52	99.06	99.13	98.52	98.98	97.16
LA40	98.86	99.43	98.54	98.46	98.46	99.10	97.60

TLBO, GRASP, PSO, Beam Search and HIA. The overall average makespan time for SSO algorithm reveal 99.77%, HGA1 algorithm reveals 99.75%, GWO algorithm reveals 99.62%, HGA algorithm reveals 99.54%, HGA2 algorithm reveals 99.47%, MA algorithm reveals 99.47%. In PGA algorithm

Table 4 Percentage for each scheduling problems using different techniques compared with BM

Problems	F&F	MA(GR-RS)	TLBO	GRASP	PSO	Beam search	HIA
FT20	100	100	100	99.65	100	none	100
LA01	100	100	100	100	100	100	100
LA02	100	100	100	100	98.05	93.03	100
LA03	100	100	100	98.84	98.51	91.84	100
LA04	100	100	97.19	100	96.56	95.16	100
LA06	100	100	100	100	100	100	100
LA07	100	100	100	100	100	100	100
LA09	100	100	100	100	100	100	100
LA21	99.42	96.94	95.87	95.87	96.67	90.79	83.28
LA27	99.43	96.03	98.32	93.56	95.73	93.84	69.22
LA36	91.81	97.01	95.19	95.05	95.19	90.50	98.98
LA40	99.51	97.60	98.46	97.06	95.02	90.72	98.54

Figure 7 Average makespan time for all aforementioned problems with different algorithms.

the average value is 98.97%, F&F algorithm exposes 98.83%, MA(GR-RS) algorithm exposes 98.65%, TLBO algorithm exposes 98.54%, GRASP algorithm exposes 97.87%, PSO algorithm exposes 97.71%, Beam Search algorithm expose 96.74% and finally the HIA algorithm exposes 93.41%. Based on this overall performance the proposed SSO algorithm achieves higher average value.

4 Conclusion

As discussed, the scheduling plays a vital role in auto industries to allocate particular job to a machine in particular time. This sort of scheduling certainly improves the work flow in the industries and increases the productivity. Computing these scheduling processes via manual take prolong time and consume enormous energy to allocate an appropriate job to a machine in order to achieve minimize makespan time. Incorporation of optimization technique in scheduling job and machine is an effective way to conserve this energy consumption. Various optimization techniques take part in minimizing makespan time amid swarm intelligence has superior result over evolutionary techniques. Amid different contest optimization techniques SSO exhibit 99.77% similar makespan time with benchmark makespan time. These soft computing techniques certainly eradicate the necessity of real time implementation on trial and error process which causes auto industries emission. In future upcoming researcher tries to apply their own algorithm for the betterment of finding minimized makespan time for job shop scheduling problems.

References

- [1] Yildiz, H., DuHadway, S., Narasimhan, R., and Narayanan, S. (2016). Production Planning Using Evolving Demand Forecasts in the Automotive Industry. *IEEE Transactions on Engineering Management*, 63(3), 296–304.
- [2] Brabazon, P. G., MacCarthy, B., Woodcock, A., and Hawkins, R. W. (2010). Mass customization in the automotive industry: comparing interdealer trading and reconfiguration flexibilities in order fulfillment. *Production and Operations Management*, 19(5), 489–502.
- [3] Staebelin, T., and Aoki, K. (2015). Planning and scheduling in the automotive industry: A comparison of industrial practice at German and Japanese makers. *International Journal of Production Economics*, 162, 258–272.
- [4] Brun, A., and Pero, M. (2012). Measuring variety reduction along the supply chain: The variety gap model. *International Journal of Production Economics*, 139(2), 510–524.
- [5] Li, J., Huang, Y., and Niu, X. (2016). A branch population genetic algorithm for dual-resource constrained job shop scheduling problem. *Computers & Industrial Engineering*, 102, 113–131.

- [6] Xing, L. N., Chen, Y. W., Wang, P., Zhao, Q. S., and Xiong, J. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing*, 10(3), 888–896.
- [7] Horng, S. C., Lin, S. S., and Yang, F. Y. (2012). Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Systems with Applications*, 39(3), 3603–3610.
- [8] Zhang, R. (2011). An Artificial Bee Colony Algorithm Based on Problem Data Properties for Scheduling Job Shops. *Procedia Engineering*, Vol. 23, pp. 131–136.
- [9] Ripon, K. S. N., Siddique, N. H., and Torresen, J. (2011). Improved precedence preservation crossover for multi-objective job shop scheduling problem. *Evolving Systems*, 2(2), 119–129.
- [10] Khowala, K., Fowler, J., Keha, A., and Balasubramanian, H. (2014). Single machine scheduling with interfering job sets. *Computers & Operations Research*, 45, 97–107.
- [11] Immorlica, N., Li, L. E., Mirrokni, V. S., and Schulz, A. S. (2009). Coordination mechanisms for selfish scheduling. *Theoretical Computer Science*, 410(17), 1589–1598.
- [12] Gao, L., Zhang, G., Zhang, L., and Li, X. (2011). An efficient memetic algorithm for solving the job shop scheduling problem. *Computers & Industrial Engineering*, 60(4), 699–705.
- [13] Adibi, M. A., Zandieh, M., and Amiri, M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, 37(1), 282–287.
- [14] Schaller, J., and Valente, J. M. (2012). Minimizing the weighted sum of squared tardiness on a single machine. *Computers & Operations Research*, 39(5), 919–928.
- [15] Noori-Darvish, S., and Tavakkoli-Moghaddam, R. (2012). Minimizing the total tardiness and makespan in an open shop scheduling problem with sequence-dependent setup times. *Journal of Industrial Engineering International*, 8(1), 25, 1–13.
- [16] Ye, L., and Chen, Y. (2010). A Genetic Algorithm for Job-Shop Scheduling. *Journal of Software*, 5(3), 269–274.
- [17] Gu, W., Tang, D., and Zheng, K. (2012). Minimizing makespan in job-shop scheduling problem using an improved adaptive particle swarm optimization algorithm. In *24th Chinese Control and Decision Conference (CCDC)*, (pp. 3189–3193). IEEE.
- [18] Allahverdi, A., and Aydilek, H. (2010). Heuristics for the two-machine flow shop scheduling problem to minimise makespan with bounded

- processing times. *International Journal of Production Research*, 48(21), 6367–6385.
- [19] Gupta, S., and Arora, S. (2015). A Hybrid Firefly Algorithm and Social Spider Algorithm for Multimodal Function. *Intelligent Systems Technologies and Applications*, 384, 17–30.
 - [20] Surekha, P., and Sumathi, S. (2010). Solving Fuzzy based Job Shop Scheduling Problems using GA and ACO. *Journal of Emerging Trends in Computing and Information Sciences*, 1(2), 95–102.
 - [21] Asadzadeh, L. (2016). A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy. *Computers & Industrial Engineering*, 102, 359–367.
 - [22] Hamed, P., Wong, K. Y., and Hassan, A. (2016). A Hybrid Genetic Algorithm with a Knowledge-Based Operator for Solving the Job Shop Scheduling Problems. *Journal of Optimization*, 1–13.
 - [23] Wang, Y., Zhu, L., Wang, J., and Qiu, J. (2015). An improved social spider algorithm for the Flexible Job-Shop Scheduling Problem. *International Conference on In Estimation, Detection and Information Fusion (ICEDIF)*, (pp. 157–162). IEEE.
 - [24] Baykasoglu, A., Hamzadayi, A., and Köse, S. Y. (2014). Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases. *Information Sciences*, 276, 204–218.

Biographies

K. B. Gavali is a Ph.D. student at Department of Mechanical Engineering, Karpagam Academy of Higher Education, Karpagam University, Coimbatore since 2013. He attended Shivaji University Kolhapur where he received his B.E. in Mechanical Engineering in 1998. Gavali then went on to purchase his M. Tech. in Mechanical Engineering from Dr. Babasaheb Ambedker

Technological University Lonere Raighad, Maharashtra in 2005. K. B Gavali has held position of Lecturer and Assistant professor in Mechanical engineering from 1999. There after worked as Principal for polytechnic college for span of 8 years and currently as a Associate Professor and Head of Department, Mechanical Engineering at Trinity Academy of Engineering Pune. K.B. Gavali is currently completing a Doctorate in Mechanical Engineering at the Karpagam Academy of Higher Education, Karpagam University, Coimbatore. His Ph.D. work job shop scheduling is effective to auto industries for optimization of makespan time.

Anand K. Bewoor is presently working as Professor of Department of Mechanical Engineering, Cummins College of Engineering, Pune. He holds a bachelor's degree in Mechanical, a master's degree in Mechanical Engineering, with specialization in Production Engineering, from Walchand C. o. E., Sangli and PhD in Mechanical Engineering. He has worked as Vendor Development and Quality Control Engineer in the industry before serving as a faculty in engineering colleges of Pune University. He has published several books in the field of Mechanical Engineering viz. Metrology & Measurements, Quality Control, Production Planning and Control, Industrial Engineering and Management, Manufacturing Processes, and Industrial Fluid Power. He has presented several technical/research papers at national and international conferences and published papers in journals of national and international repute. Apart from these, Dr. Bewoor has also filed two patents. He is also a member of various professional bodies.

Debabrata Barik is an Associate Professor, Department of Mechanical Engineering, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India. He has teaching and research experience of over 10 years in various universities and completed his doctorate degree from National Institute of Technology Rourkela, India. His research interests include Combustion Engines, Waste to Energy, Renewable Energy, Regulated and Unregulated Emissions, and Heat Transfer among many other topics. Dr. Barik has published more than 20 international journals and conference proceedings and is an active reviewer for more than 30 international journals. Dr. Barik also published a book entitled “Energy from Toxic Organic Waste for Heat and Power Generation”. He is currently a member of the Indian Society for Technical Education (ISTE) and International Society for Energy, Environment and Sustainability (ISEES).