IJIO
(*International Journal of Industrial Optimization*)
ISSN 2723-3022 (online) | 2714-6006 (printed)
Vol 5, No. 1, 81-92
https://doi.org/10.12928/ijio.v5i1.8743

Multi-objective elitist spotted hyena resource optimized flexible job shop scheduling

A. N. Senthilvel [a,*], T. Hemamalini [b], G. Geetha [b]

[a] Department of Computer Science and Engineering, Coimbatore Institute of Technology, India
[b] Department of Mathematics, Government Arts College, India
* Corresponding Author: ggeethamuralikumar@gmail.com

ARTICLE INFO

Article history
Received: July 31, 2023
Revised: August 31, 2023
Accepted: July 7, 2023

Keywords
Job shop scheduling;
Multi-objective Elitist Spotted Hyena optimization;
Rate-Monotonic preemptive Scheduling;
Levenberg–Marquardt method.

ABSTRACT

The job shop scheduling problem (JSSP) has drained a lot of consideration since it is one of the most important optimization problems in the manufacturing domain. The scheduling method is crucial for optimizing the objective of minimizing makespan among thousands of jobs, but evaluating machine capacity for achieving this goal remains challenging despite the development of various population-based optimization algorithms for job shop scheduling problems. To improve the efficiency of Job shop scheduling, a novel Multi-objective Elitist Spotted Hyena Monotonic Scheduling (MESHS) technique is introduced. The proposed MESHS technique includes two major processes: machine selection and operation sequences. The number of jobs is considered for solving the scheduling problem. First, the machine selection is performed by applying the Multi-objective Elitist Spotted Hyena optimization technique. The optimization technique selects the optimal machines parallelly based on multiple objective functions such as energy consumption, CPU utilization, and job completion time. The fitness of every machine is calculated based on these multiple objective functions using Levenberg–Marquardt method. Then the Elitist strategy is applied to select the optimal machine based on fitness. After the machine selection, the rate-monotonic preemptive scheduling is modeled to provide a robust operation sequence by assigning high-priority jobs to the optimal machines. As a result, efficient job scheduling is achieved with minimum time. Finally, the experimental valuation is carried out using a benchmark OR-Library dataset with different factors such as job shop scheduling efficiency, job scheduling time, makespan, and memory consumption concerning a number of jobs.

1. Introduction

The job-shop scheduling problem is extensively applied in the real-world manufacturing domain. With the growth of the manufacturing industry, job-shop scheduling is an efficient means to improve machining performances including cost, energy consumption, and so on. Therefore, resource scheduling and job sequencing are significant manufacturing systems processes. Several evolutionary algorithms have been developed for desired scheduling solution in an acceptable time.

A greedy randomized adaptive search procedure (GRASP) was developed in [1] for integrated scheduling of dynamic flexible job shops. But, the efficiency of dynamic flexible job scheduling was a major problem. A multi-objective memetic algorithm (MOMA) was introduced in [2] to minimize the makespan and the total energy consumption. However, performing dynamic scheduling with an efficient metaheuristic's technique was difficult.

A hybrid particle swarm optimization and simulated annealing algorithm (PSOSA) were developed by [3] to evaluate the performance of the heuristic solutions for dynamic scheduling with larger instances. The Quantum Annealing (QA)-based job shop scheduling technique was introduced in [4] to solve flexible job shop scheduling problems with various sizes. But the designed technique failed to solve the multi-objective optimization, increasing the job allocation's complexity.

A Novel Multi-Objective Evolutionary Algorithm was developed in [5] based on Decomposition for energy-aware distributed hybrid flow shop scheduling. But, the designed evolutionary algorithm was inefficient in improving the scheduling process's performance. A two-stage genetic programming hyperheuristics (GPHH) method was developed in [6] to automatically improve dynamic flexible job shop scheduling. However, it failed to find more promising ways by local search to improve the performance of dynamic flexible job shop scheduling.

A hybrid algorithm (HGA-TS) which combines genetic algorithm (GA) and tabu search (TS) was introduced in [7] for flexible job shop scheduling. However the multi-objective optimization was not performed for the job shop scheduling environment. An approach integrating the artificial immune system into ordinal optimization was introduced in [8] for near-optimal scheduling with minimum time. However, the scheduling efficiency was not improved by considering the greater number of jobs. An improved multi-objective optimization algorithm was designed in [9] for flexible job shop scheduling. But it failed to integrate heuristics and self-learning to improve the performance of the flexible job shop scheduling algorithm.

A knowledge-based cuckoo search algorithm (KCSA) was designed in [10] to the scheduling for solving an extended flexible job-shop scheduling. However, it was not concerning the strategy with various multi-swarm optimizations to enhance the performance of the cuckoo search algorithm.

An artificial bee colony algorithm was developed in [11] for job shop scheduling. However, the performance of the artificial bee colony algorithm was not improved. An effective recombinative guidance approach was developed in [12] for genetic programming to improve effective scheduling. But the performance of makespan was not analyzed. A multi-objective genetic algorithm was designed in [13] for job shop scheduling problems by minimizing the total energy consumption. But it failed to improve the applicability of the energy-saving scheduling theory. A squared hyper-heuristic approach was introduced in [14] to solve job shop scheduling problems. But it failed to improve the performance of the total makespan. An improved genetic algorithm was designed in [15] for effective hybrid multi-objective flexible job shop scheduling. But the efficiency of the algorithm was not increased desirably.

An alternative mixed integer linear programming model was developed in [16] for job shop scheduling to reduce the makespan as well as total energy consumption. But it was not efficient to improve the performance of job shop scheduling. A robust fuzzy stochastic programming (RFSP) approach was designed in [17] for flexible job-shop scheduling. But it failed to consider the performance of the makespan measure. A new fitness estimation mechanism based on fuzzy relative entropy was introduced in [18] for energy-efficient job-shop scheduling aiming to minimize the makespan and energy consumption. But it failed to consider the multi-objective optimization framework.

A hybrid distributed evolutionary method was developed in [19] for large-scale job-shop scheduling by using the Bayesian grouping method. The designed method failed to apply the scheduling in more complex environments involving machine breakdown and urgent job insertion. An enhanced genetic algorithm with an elite strategy was developed in [20] for complicated flexible job-shop scheduling.

However, the field of energy-aware job-shop scheduling has not yet been extensively explored in the literature, resulting in a major lack of research on comprehending and maximising the energy consumption elements in the context of job-shop scheduling. Therefore, to improve the efficiency of job shop scheduling, this research objective is to develop a novel Multi-objective Elitist Spotted Hyena Monotonic Scheduling (MESHS) technique. The major contribution made in the article for the job shop scheduling is summarized as follows:

- To improve the multi-objective job shop scheduling efficiency, a novel technique called MESHS is introduced by including two contributions namely optimal machine selection and operation sequencing.
- The multi-objective elitist spotted hyena optimization technique is introduced for selecting the optimal machines in a parallel manner based on multiple objective functions such as Energy consumption, CPU utilization, and minimum job completion time. The fitness of the machine is estimated based on Levenberg–Marquardt method with multiple objective functions. Then the Elitist strategy is applied to select the optimal machine with minimum resource utilization.
- Next, Rate-monotonic preemptive scheduling technique is applied in MESHS for operation sequencing by assigning the priority level for each incoming job. The higher priority operation is first assigned to the optimal machine. This helps to improve the task scheduling efficiency and reduce the makespan.
- Finally, extensive simulation is carried out with various performance metrics to highlight the improvement of the proposed MESHS technique over conventional scheduling techniques.

The rest of this article is organized into five different sections as follows. Section 2 explains the proposed job shop scheduling strategy MESHS and provides the simulation settings and dataset description. Section 3 provides the proposed algorithm's performance evaluation compared with existing scheduling strategies. Finally, the paper will be concluded in Section 4.

2. Method

Flexible job shop scheduling is a demanding combinatorial optimization problem due to its complex environment. The scheduling objective is to minimize the total completion time and total cost of the job in computer science and operational research. The scheduling problem consists of a limited number of resources in which each job is processed by numerous machines simultaneously at every stage. As the number of incoming job volumes from all fields increases exponentially, the resources for processing such jobs also rapidly increase. Therefore, an efficient scheduling algorithm is required to process such kinds of jobs with minimum completion time. Based on this motivation, the MESHS technique is introduced. Contrary to the existing optimization, the proposed MESHS solves the multiple objective functions for efficient dynamic scheduling of numerous jobs. The architecture of the MESHS technique is shown in Fig. 1.

Fig. 1 shows the architecture of the proposed MESHS technique for dynamic flexible job shop scheduling into machines with minimum time. In a dynamic job shop scheduling environment, there is a set of jobs $J=\{J_1,J_2,\ldots,J_n\}$ arrived and need to be processed by machines '$M=\{M_1,M_2,\ldots,M_m\}$'. Every job J_i arrived at a time 't' and sequence of operations $O_k=\{O_1,O_2,\ldots,O_k\}$ to be processed. Each operation O_k processed by the optimal machine 'M' with different processing times. The machine processed more than one operation simultaneously. An operation of one job is performed in a particular order. The first operation finishes on the first machine, then the second operation on the second machine, and so on until the n-th operation. It involves two types of decisions that aim to simultaneously make in dynamic flexible job shop scheduling such as optimal machine selection and Operation sequences.

First, the optimal machine selection process is said to be performed by applying a multi-objective Elitist spotted hyena optimization. The proposed optimization technique is a population-based meta-heuristic optimization that worked on the basis of the hunting behavior of the spotted hyenas. The multiple objective functions such as energy consumption, CPU utilization and job completion time into consideration, and lesser time are considered for solving the dynamic flexible job shop scheduling problem. Then, the Elitist strategy is applied to select the optimal machine for processing certain operations of incoming jobs. Finally, the operation sequencing is performed to assign the job operations to the selected optimal machine using Rate-Monotonic preemptive scheduling. The proposed scheduling is a priority assignment algorithm and the static priorities are assigned to the job along with the cycle duration. Therefore, the job with the highest priority gets scheduled first than the others. The proposed MESHS technique is described in the following subsections.

 10.12928/ijio.v5i1. 8743

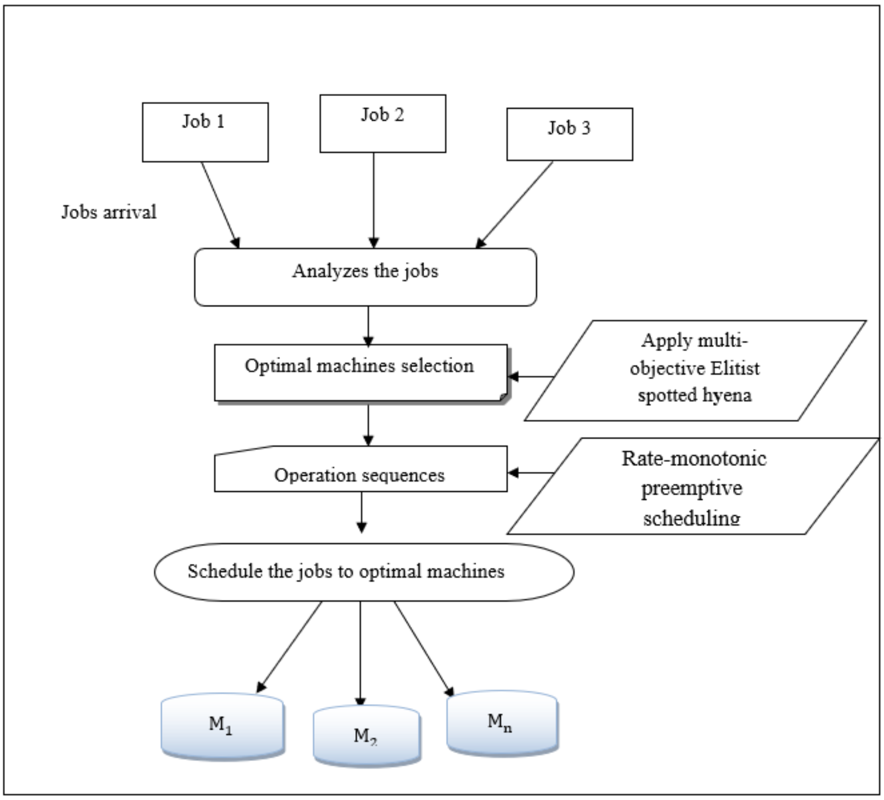

Fig. 1. Architecture of the MESHS technique

2.1. Multi-objective elitist spotted hyena optimization

The proposed MESHS performs the optimal machine selection by using Multi-objective Elitist spotted hyena optimization, a population-based meta-heuristic technique. The main advantage related to the method includes faster convergence time. In the optimization process, spotted hyena act as a search agent and starts to determine the location of prey. Here, the spotted hyena is related to the machines, and the prey is denoted as multiple objective functions. The Elitist selection strategy is used to find the best optimal hyena with the best fitness values.

For each arriving job with operations $O_k = \{O_1, O_2, \dots, O_k\}$, the population of the 'm' number of spotted hyenas (i.e. machines) is initialized in the search space, as shown in Eq. (1).

$$M_i = M_1, M_2, M_3, \dots M_{\mathrm{m}} \tag{1}$$

After the initialization, the proposed optimization algorithm computes the fitness based on multiple objective functions, such as CPU utilization, energy consumption, job completion time. CPU utilization is processing resources referred to as the amount of time for which a machine processes certain operation, as presented in Eq. (2).

$$cpu_t = T\ [processing\ O_k] \tag{2}$$

where, cpu_t denotes a CPU time, T denotes a time, O_k indicates the operations of the particular job.

Energy is the other major resource of the machine to process the operations of the job. The energy consumption factors of machines are different, which represents processing jobs on different machines consume different energy. Energy consumption consists of two parts namely processing energy and idle energy. The formula can be seen in Eq. (3).

$$Ec = E_{proc} + E_{idle} \tag{3}$$

where, Ec denotes energy consumption, E_{proc} denotes processing energy, E_{idle} denotes idle energy.

Job completion time is defined as the time the machines take to complete certain operations. It can be seen in Eq. (4).

$$T_j = T_E - T_S \tag{4}$$

where, T_j denotes a predicted job completion time, T_E represents an ending time to complete the operation of the job and T_S represents a starting time to process the operation of the job.

The optimal machine is selected through the fitness function based on the resources. In the fitness measure, the optimization technique uses the Levenberg–Marquardt method is applied to find the local minimum of a function for selecting the optimal one among the population. It is presented in Eq. (5).

$$Z = arg\ min\left(cpu_t, Ec,\ T_j\right) \tag{5}$$

where, Z denotes a fitness, *arg min* indicates an argument minimum function to find the machine with minimum resource utilization.

Then the Elitist selection strategy is applied to determine the current best machine (i.e. spotted hyenas) from the whole population based on fitness as shown in Eq. (6).

$$R = \begin{cases} Z > \ Z_{th} : select\ best\ machine \\ otherwise\ ;\ reject\ the\ others \end{cases} \tag{6}$$

where, R indicates selection outcomes, Z_{th} denotes a threshold, Z denotes a fitness.

As a result, the machines with higher fitness are selected as the optimal machine and others are removed. This helps to minimize time consumption and improve the performance of the optimal machine selection. Based on the selection outcomes, different behaviors of the spotted hyenas such as encircling the prey, hunting, and attacking the prey are described as give below.

2.2. Encircling the prey

Encircling the prey is the basic behavior of the optimization technique. The location of prey is determined by spotted hyenas, which encircle them. Let us consider that the current best candidate solution is the target prey, i.e., objective functions close to the optimum. The mathematical model of encircling the prey behavior is presented in Eq. (7) – Eq. (11).

$$D = \left|h.P_p - P_S\right| \tag{7}$$

$$P_{S+1} = P_p - r.D \tag{8}$$

where, D indicates the distance between the location of the prey 'P_p' and spotted hyena 'P_S', P_{S+1} denotes an updated position of the hyena, $h,\ r$ denotes coefficient vectors.

$$h = 2.v_1 \tag{9}$$

$$r = 2\,G.v_2 - G \tag{10}$$

$$G = 5 - \left(It * \left(\frac{5}{max_{it}}\right)\right) \qquad \text{Where, } It = 1,2,\ldots Mx_{Iter} \tag{11}$$

where, v_1, v_2 denotes a random vector in [0, 1], G is linearly decreased from 5 to 0 during iterations 'It', max_{it} indicates a maximum iteration.

2.3. Hunting behavior

To perform the hunting behavior of spotted hyenas, the best optimal spotted hyenas have an awareness of the location of prey. The remaining hyenas form a group and update their position according to the best hyenas. The mathematical formulation of hunting behavior is shown in Eq. (12) – Eq. (14).

$$X = |h.P_b(S) - P_o\,(S)| \tag{12}$$

$$P_o\,(S) = P_b\,(S) - r.D \tag{13}$$

$$P_{S+1} = P_p - r.D \tag{14}$$

where, $P_b(S)$ indicates the position of the first best-spotted hyena, $P_o\,(S)$ indicates the position of other spotted hyenas.

 10.12928/ijio.v5i1. 8743

2.4. Attacking prey

The final behavior of the optimization is attacking the prey. The spotted Hyenas attack the prey when they are closer to the prey. The mathematical formula for attacking the prey is given in Eq. (15).

$$P_{S+1} = \frac{A}{n} \quad (15)$$

where, P_{S+1} denotes the best solution and updates the positions of other hyenas along with the position of the best machine, A denotes a group of 'n' optimal solutions. Again, the fitness is computed for the newly updated position of hyenas. If the fitness of the updated position of the hyenas is higher than the old position, then it restores the current best as optimal. This process is repeated until the maximum number of iterations gets reached.

2.5. Rate-monotonic preemptive scheduling-based operation sequence

Operation Sequencing is the order of scheduling to be performed in sequence format. Therefore, the proposed technique uses the rate-monotonic scheduling technique for operation sequencing. The rate-monotonic scheduling is a priority assignment algorithm where the static priorities are assigned along with the duration of the execution time of the operation, therefore a shorter time duration results in a higher job priority. Preemptive scheduling is the currently executed operation goes to a waiting state when the high-priority operation arrives.

For each operation 'O', the CPU burst time is calculated as the total time taken by the process for its execution of the operation as presented in Eq. (16).

$$CPU\ Burst\ Time = Operation\ Completion\ Time \ - Waiting\ Time \quad (16)$$

After measuring the CPU burst time, the priority is assigned to the machine's scheduled operation. Then, assigning a higher priority for minimum burst time can be formulated in Eq. (17).

$$\varphi \xrightarrow{highest\ priorty} \arg\min BT \quad (17)$$

where, φ denotes a Rate-monotonic preemptive scheduling output, $\arg$ min denotes a minimum burst time 'BT'.

Table 1. Rate-monotonic preemptive scheduling process

Operations	**CPU burst time (ms)**	**Priority for scheduling**
O_1	5	Second
O_2	7	Third
O_3	9	Four
O_4	4	First

By applying Rate-monotonic preemptive scheduling, the operation O_4 has the shortest CPU time and it has the highest priority for scheduling, followed by O_1, O_2 and finally O_3. In this way, the operation sequencing process is performed by scheduling all the operations to the optimal machine. The algorithmic process of multi-objective elitist spotted hyena monotonic scheduling is given below.

// **Algorithm 1: M**ulti-objective **E**litist **S**potted **Hy**ena Monotonic **S**cheduling
Input: Machine '$M = \{M_1, M_2, \dots, M_m\}$', Job '$J = \{J_1, J_2, \dots, J_n\}$', Operations '$O_k = \{O_1, O_2, \dots, O_k\}$',
Output: Find optimal machine and robust operation sequencing
Begin
<u>**Optimal machine selection**</u>
1. Initialize the population of machines $M = \{M_1, M_2, \dots, M_m\}$
2. **for each** 'M_i'
3. Calculate the fitness 'Z' based on multiple objective functions
4. Apply elitist selection

$if\ (Z > Z_{th})$ **then**
    Select the current best machine
  else
    Remove the others
end if
end for
  While (t < max _it)
    for each current best machine
      Perform Encircling the prey using
      Perform hunting behavior
      Update the position of the current best
end for
  Evaluate fitness function for the new position
  $if\ (Z_i < Z_{i+1})$ **then**
   Replace the old best into the current best
end if
    t=t+1
End while
Return (best optimal machine)
<u>// operation sequencing</u>
_For each operation
  Measure the CPU burst time
   Find the minimum CPU burst time
   Assign high priority to the minimum CPU burst time operation
For each high-priority operation
     Perform preemptive Scheduling
End for
End for
End

The algorithm of MESHS is described in step-by-step process to perform resource-efficient job shop scheduling by using optimal machine selection and operation sequencing. First, the multi-objective elitist spotted hyena optimization is applied to select the resource optimal machine. The populations of machines are initialized randomly. For each machine, the multiple objective functions are measured. Based on the estimation of the objective function, the fitness is computed. Then the current best machine is selected by applying the elitist selection strategy. If the fitness of the current best machine is higher than the threshold, then the machine is selected. Other machines with lesser thresholds are removed. Followed by, the different behaviors are estimated. After that, the current best positions of the spotted hyenas (i.e. machines) are updated. After that, the fitness is estimated for the newly updated position. If the fitness of the previous position (Z_i) is better than the fitness of the new position (Z_{i+1}), then it replaces the new position as optimal. The entire process is repeated until the maximum iteration gets reached. Finally, the optimal machine is selected. After that, the preemptive scheduling process is performed by assigning the job operations to the optimal machine. For each operation, the CPU burst time is computed and assigned priority. The operations with high priority are scheduled first than the others. In this way, efficient job shop scheduling is performed, resulting in a reduced makespan.

2.6. Experimental setup

In this section, experimental evaluation of proposed MESHS and existing GRASP [1] MOMA [2] PSOSA [3] are implemented using MATLAB simulator and run on an Intel Core i7-3520M 2.9 GHz processor with 8 GB of RAM. The benchmark OR-Library dataset is used for dual resource-constrained processes such as optimal machine selection and sequencing. The parameter settings are listed in Table 2.

 10.12928/ijio.v5i1. 8743

Table 2. Parameter settings

S. No	Parameter	Value
1	Number of jobs	150
2	Number of machines	20
3	Number of operations	300
4	Maximal iterations	10

3. Results and Discussion

In this section, the performance of the proposed MESHS and existing GRASP [1], MOMA [2], PSOSA [3] are discussed with different metrics such as Job shop scheduling efficiency, makespan, and energy consumption. The performances of the three methods are discussed with a table or graphical representation.

3.1. Impact of job shop scheduling efficiency

Job shop scheduling efficiency is defined as the number of incoming jobs that are correctly scheduled to the resource-optimized machine. The efficiency is calculated as shown in Eq. (18).

$$JSSE = \left(\frac{Number\ of\ jobs\ correctly\ scheduled}{n}\right) * 100 \tag{18}$$

Where $JSSE$ denotes a Job shop scheduling efficiency, n represents the number of jobs correctly scheduled. The task scheduling efficiency is measured in terms of percentage (%).

Table 3. Comparison of Job shop scheduling efficiency

Methods	Job shop scheduling efficiency (%)
MESHS	98.75
GRASP	91.42
MOMA	94.55
PSOSA	95.22

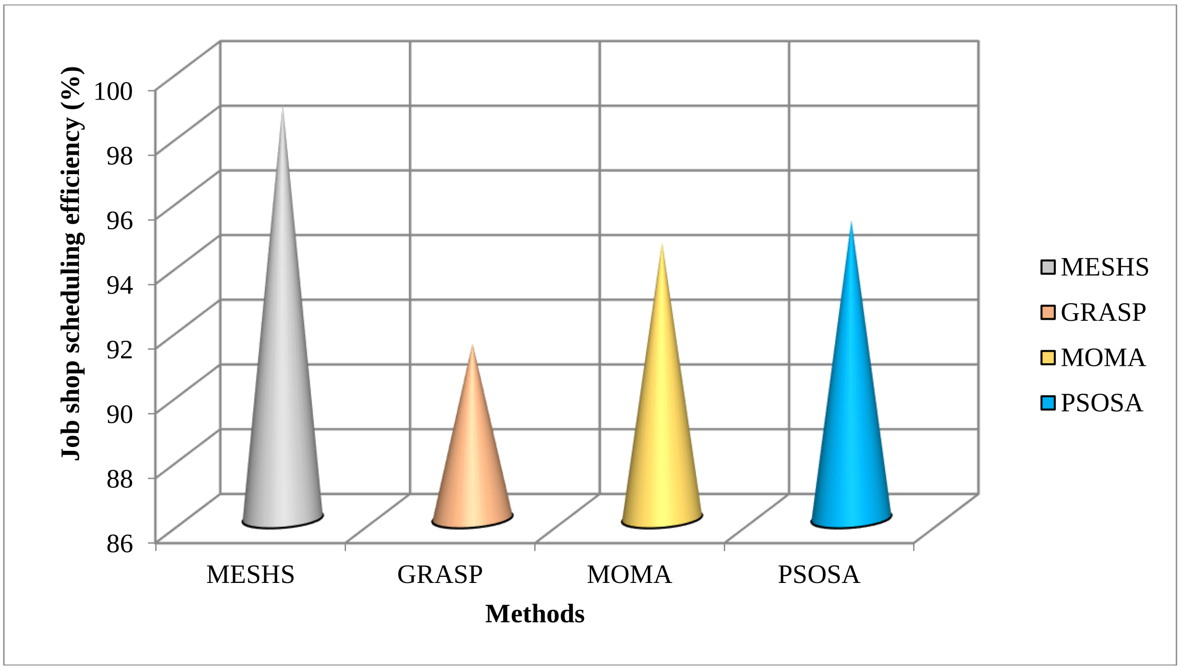

Fig. 2. Performance results of job shop scheduling efficiency

Table 3 and Fig. 2 illustrate the various simulation results of job shop scheduling efficiency by considering the 150 jobs taken from datasets. The scheduling efficiency is measured using four methods namely MESHS and existing GRASP [1], MOMA [2], and PSOSA [3]. Among the three methods, the above-reported results noticeably confirm that the job shop scheduling efficiency of the

MESHS technique is higher than the GRASP [1], MOMA [2], and PSOSA [3]. This is owing to MESHS initially identifying the optimal machine for processing the high-priority operations of the given job. The multi-objective elitist spotted hyena optimization is applied to identify the optimal resource machine by applying the elitist selection strategy. The machine which has minimum resource consumption is selected as optimal to assign the jobs to that virtual machine. As a result of scheduling, high-priority jobs are scheduled first. The efficiency of the MESHS technique is compared to the existing technique. The average value of comparative results indicates that the job shop scheduling efficiency of MESHS is increased by 8%, 5%, and 4% as compared to GRASP [1], MOMA [2], and PSOSA [3], respectively.

3.2. Impact of makespan

The makespan is the time difference between the start and end times of flexible Job shop scheduling. It is also defined as the overall time consumed for assigning jobs to the respective machines. The formula for calculating the makespan is presented in Eq. (19).

$$MS = \sum_{i=1}^{n} J_i * [ST - FT] \quad (19)$$

where, 'MS' denotes a makespan, the number of jobs involved in the simulation process 'J_i' ST denotes a start time, 'FT' denotes a finish time 'FT' respectively. It is measured in terms of milliseconds (ms).

Table 4. Comparison of makespan

Number of jobs	Makespan (ms)			
	MESHS	GRASP	MOMA	PSOSA
15	92.6	120	112	107.5
30	100.25	135.15	128.44	117.9
45	118	150.25	137.21	128.6
60	132	195.35	165	155.5
75	132.45	235.55	188.66	168.66
90	158.2	250.15	220.45	189.5
105	176.55	295.35	247.89	218.66
120	185.5	310.25	286	245.75
135	204.3	355.15	306.5	284.14
150	247.65	390.25	344.4	312

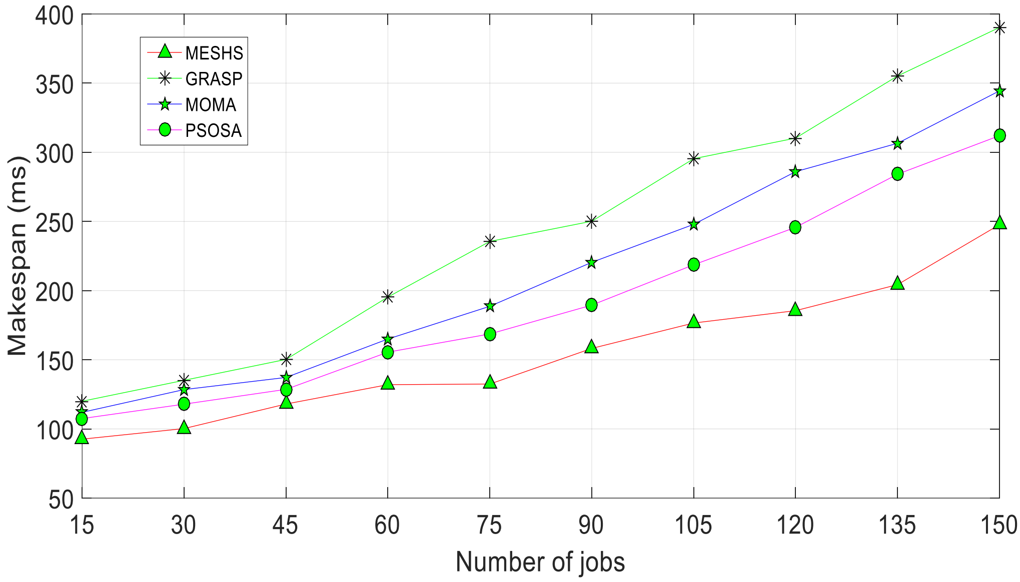

Fig. 3. Performance results of makespan

Table 4 and Fig. 3 illustrate the performance results of makespan in terms of the number of jobs taken in the ranges from 15, 30 … 150. For each method, ten different results are achieved. The

 10.12928/ijio.v5i1. 8743

observed results indicate that the makespan is minimized using MESHS than the existing GRASP [1], MOMA [2], and PSOSA [3] methods respectively. This is verified through statistical examination. By considering 15 jobs in the first iteration, the makespan of MESHS is $92.6ms$. Similarly, by applying GRASP [1], MOMA [2], and PSOSA [3], $120ms, 112ms$ and $107.5ms$ respectively. Ten results are observed and the results are compared. The average of ten results indicates that the makespan of MESHS is considerably minimized by 34% when compared to [1] and 26% when compared to [2] and 18% when compared to [3] respectively. This is because of applying the Rate-monotonic preemptive scheduling technique. The scheduling process is performed by assigning the operations of the jobs to the optimal machine. For each operation, the CPU burst time is computed and then find the high-priority jobs. The operations with high priority are scheduled first than the others. In this way, efficient job shop scheduling is performed and minimizes the makespan.

3.3. Impact of energy consumption

The machine's energy consumption during the job shop scheduling is measured based on processing energy and idle energy. The total energy consumption of the machine is calculated as shown in Eq. (20).

$$Ec = \sum_{i=1}^{n} J_i * (E_{proc} + E_{idle}) \quad (20)$$

where, Ec denotes energy consumption, E_{proc} denotes processing energy, E_{idle} denotes idle energy. J_i denotes the number of jobs. It is measured in terms of a kilowatt hour (kwh)

Table 5. Comparison of energy consumption

Number of jobs	Energy consumption (kwh)			
	MESHS	**GRASP**	**MOMA**	**PSOSA**
15	615.7	785	756.6	695.5
30	725.3	896.7	842.5	765
45	780.69	945	878.7	832.5
60	840.2	978.3	945.6	895.6
75	863.5	1042	984.3	941.5
90	910.1	1086.6	1045	974.6
105	942.56	1125.1	1087	1009.1
120	962.4	1148	1108	1015
135	1011	1167.5	1152	1108
150	1023.4	1245.35	1210	1145

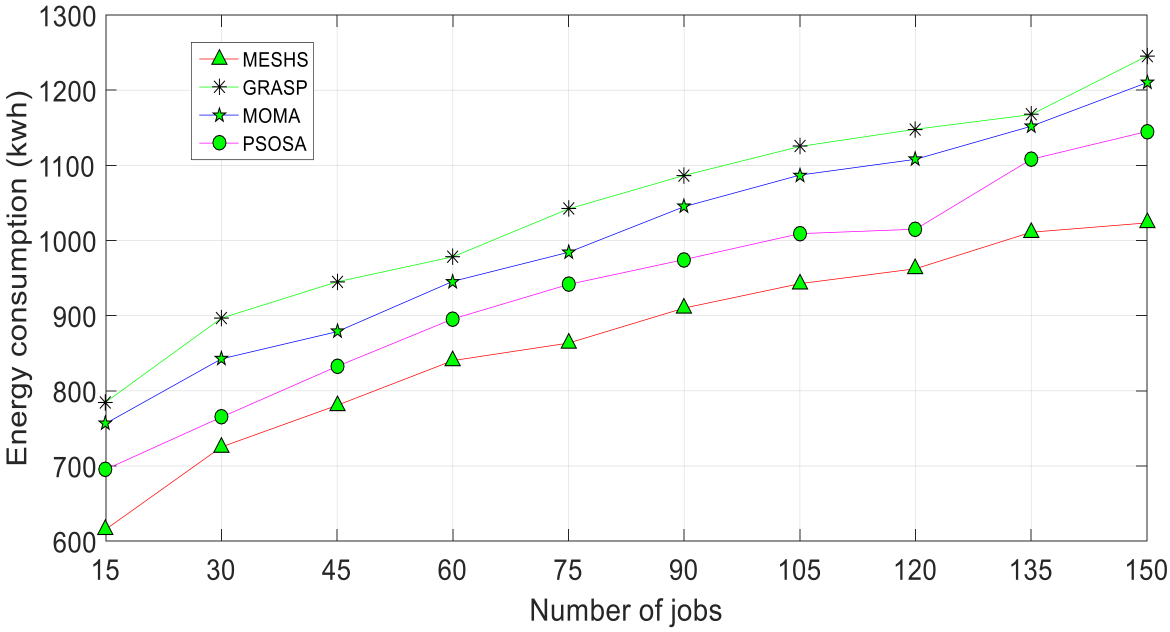

Fig. 4. Performance results of energy consumption

Table 5 and Fig. 4 demonstrate the experimental energy consumption results for dynamic scheduling of the jobs to the optimized machine. From the observed experimental results, it is inferred that the energy consumption is increased while increasing the number of jobs since each job comprises of a number of operations. Among four different methods, the memory consumption of the MESHS is minimal than the other three conventional optimization techniques. Let us consider the 15 jobs for calculating the energy consumption in the first iteration. Firstly, the MESHS technique consumed 615.7kwh of energy for scheduling the 15 jobs. Next, the energy consumption of [1], [2] [3] are observed by 785 kwh,756.6 kwh 695.5 kwh, respectively. As a result, ten results of energy consumption were obtained and compared. The comparison results indicate that the performance of energy consumption using the MESHS technique is considerably reduced by 17%, 13%, and 8% when compared to the GRASP [1], MOMA [2], and PSOSA [3], respectively.

4. Conclusion

In this paper, a novel MESHS technique is developed to schedule the job to a machine for reducing the makespan. The MESHS techniques consist of two major processes such as machine selection and operation sequencing. The multi-objective elitist spotted hyena optimization is applied for selecting resource-optimal machines in a parallel manner for assigning the jobs. After the optimal machine selection, the rate-monotonic preemptive scheduling is applied for assigning the high-priority operations of the jobs with higher efficiency and minimum time consumption. To evaluate the performance, a comparative analysis is performed using the proposed MESHS algorithm and existing optimization algorithms. The experimental outcome illustrates that the MESHS technique provides better performance for scheduling the number of jobs to an optimal machine with higher efficiency and lesser makespan as compared to state-of-the-art works.

Reference

[1] A. Baykasoglu and F. S. Madenoglu, "Greedy randomized adaptive search procedure for simultaneous scheduling of production and preventive maintenance activities in dynamic flexible job shops," *Soft Computing*, vol. 25, pp. 14893–14932, 2021, doi: 10.1007/s00500-021-06053-0

[2] C. Lu, B. Zhang, L. Gao, J. Yi, and J. Mou, "A Knowledge-Based Multi-objective Memetic Algorithm for Green Job Shop Scheduling With Variable Machining Speeds," *IEEE Systems Journal*, vol. 16, no. 1, pp. 844–855, 2022, doi: 10.1109/JSYST.2021.3076481

[3] D. B. M. M. Fontes, S. M. Homayouni, and J. F. Gonçalves, "A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources," *European Journal of Operational Research*, pp. 1-18, 2022, doi: 10.1016/j.ejor.2022.09.006

[4] P. Schworm, X. Wu, M. Glatt, and J. C. Aurich, "Solving flexible job shop scheduling problems in manufacturing with Quantum Annealing," *Production Engineering*, pp. 1-11, 2022, doi: 10.1007/s11740-022-01145-8

[5] E. Jiang, L. Wang, and J. Wang, "Decomposition-based multi-objective optimization for energy-aware distributed hybrid flow shop scheduling with multiprocessor tasks," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 646–663, 2021, doi: 10.26599/TST.2021.9010007

[6] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Evolving Scheduling Heuristics via Genetic Programming With Feature Selection in Dynamic Flexible Job-Shop Scheduling," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1797–1811, 2021, doi: 10.1109/TCYB.2020.3024849

[7] Y. Wang and Q. Zhu, "A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem With Sequence-Dependent Setup Times and Job Lag Times," *IEEE Access*, vol. 9, pp. 104864–104873, 2021, doi: 10.1109/ACCESS.2021.3096007

[8] S.-C. Horng and S.-S. Lin, "Apply Ordinal Optimization to Optimize the Job-Shop Scheduling Under Uncertain Processing Times," *Arabian Journal for Science and Engineering*, vol. 47, 2022, pp. 9659-9671, doi: 10.1007/s13369-021-06317-9

[9] W. Xiuli, P. Junjian, X. Zirun, Z. Ning, and W. Shaomin, "An improved multi-objective optimization algorithm for solving flexible job shop scheduling problem with variable batches," *Journal of Systems Engineering and Electronics*, vol. 32, no. 2, pp. 272–285, 2021, doi: 10.23919/JSEE.2021.000024

[10] Z. C. Cao, C. Lin, and M. Zhou, "A Knowledge-Based Cuckoo Search Algorithm to Schedule a Flexible Job Shop With Sequencing Flexibility," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 56–69, 2021, doi: 10.1109/TASE.2019.2945717

[11] X. Liu, J. Chen, X. Huang, S. Guo, S. Zhang, and M. Chen, "A New Job Shop Scheduling Method for Remanufacturing Systems Using Extended Artificial Bee Colony Algorithm," *IEEE Access*, vol. 9, pp. 132429–132441, 2021, doi: 10.1109/ACCESS.2021.3114712

[12] F. Zhang, Y. Mei, S. Nguyen, and M. Zhang, "Correlation Coefficient-Based Recombinative Guidance for Genetic Programming Hyperheuristics in Dynamic Flexible Job Shop Scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 3, pp. 552–566, 2021, doi: 10.1109/TEVC.2021.3056143

[13] H. Wei, S. Li, H. Quan, D. Liu, S. Rao, C. Li, and J. Hu, "Unified Multi-Objective Genetic Algorithm for Energy Efficient Job Shop Scheduling," *IEEE Access*, vol. 9, pp. 54542–54557, 2021, doi: 10.1109/ACCESS.2021.3070981

[14] A. Vela, J. M. Cruz-Duarte, J. C. Ortiz-Bayliss, and I. Amaya, "Beyond Hyper-Heuristics: A Squared Hyper-Heuristic Model for Solving Job Shop Scheduling Problems," *IEEE Access*, vol. 10, pp. 43981–44007, 2022, doi: 10.1109/ACCESS.2022.3169503

[15] J. Fang, "An Effective Hybrid Multi-objective Flexible Job Shop Scheduling Problem Based on Improved Genetic Algorithm," *Scientific Programming*, vol. 2022, pp. 1-10, 2022, doi: 10.1155/2022/2120944

[16] A. Ham, M.-J. Park, and K. M. Kim, "Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming," *Mathematical Problems in Engineering*, vol. 2021, pp. 1-12, 2021, doi: 10.1155/2021/8035806

[17] P. Soofi, M. Yazdani, M. Amiri, and M. A. Adibi, "Robust Fuzzy-Stochastic Programming Model and Meta-Heuristic Algorithms for Dual-Resource Constrained Flexible Job-Shop Scheduling Problem Under Machine Breakdown," *IEEE Access*, vol. 9, pp. 155740–155762, 2021, doi: 10.1109/ACCESS.2021.3126820

[18] L. He, R. Chiong, W. Li, S. Dhakal, Y. Cao, and Y. Zhang, "Multi-objective Optimization of Energy-efficient Job-Shop Scheduling with Dynamic Reference Point-based Fuzzy Relative Entropy," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 600–610, 2022, doi: 10.1109/TII.2021.3056425

[19] J. Liu, Z. Sui, X. Li, and J. Yang, "A Bayesian-Grouping Based Hybrid Distributed Cooperative Evolutionary Optimization for Large-Scale Flexible Job-Shop Scheduling Problem," *IEEE Access*, vol. 9, pp. 69114–69126, 2021, doi: 10.1109/ACCESS.2021.3076732

[20] X. Liang, J. Chen, X. Gu, and M. Huang, "Improved Adaptive Non-Dominated Sorting Genetic Algorithm With Elite Strategy for Solving Multi-Objective Flexible Job-Shop Scheduling Problem," *IEEE Access*, vol. 9, pp. 106352–106362, 2021, doi: 10.1109/ACCESS.2021.3098823