ISSN: 0369-8963

Reducing the Job Shop Scheduling Problem in Real Time Application by Efficient Optimization Method

C.PRABHU[1] AND T.JESUDAS[2*]
[1]*Thiyagarajar Polytechnic College, Salem, Tamil Nadu, India.*
[2]*Mahendra Engineering College (Autonomous), Namakkal, Tamil Nadu, India.*
*Corresponding author: T.JESUDAS

ABSTRACT

The Job-shop Scheduling Problem (JSSP) is an extension of optimal scheduling problem, with the procedures can be accomplished by group of inbuilt machines. This paper aims to propose an efficient optimization approach to resolve the composite scheduling problem in job shop environment. Basically, genetic algorithm based optimization is significant to resolve the scheduling process in job-machine environment so it aims to use genetic algorithm combining with Vickrey–Clarke–Groves (VCG) mechanism, named as Scheduling by Genetic Algorithm with Vickrey–Clarke–Groves Mechanism (SGAVCGM) based on auction and player method with chromosome activities. The chromosome representation of the problem is based on random keys and hence the schedules are constructed using a schedule generation scheme in which the priorities are constructed using optimal path. This approach is tested on a set of standard instances taken from the literature and compared with other approaches.

Keyword- *Scheduling, job-shop, optimization, genetic algorithm, optimal path.*

INTRODUCTION

Job shop scheduling or the job-shop problem (JSP) is an optimization problem in which various manufacturing jobs are assigned to machines at particular times while trying to minimize the makespan [1]. Scheduling has direct impacts on the production efficiency and costs of a manufacturing system, thus it has attracted a great deal of research attentions since 1956. However, JSP is usually a NP combinatorial optimization problem. When scaling up a problem, the existing optimization methods concentrated on centralized scheduling or semi-distributed scheduling meet great challenges in terms of computational stability and time [2]. In order to shifting traditional scheduling into smart distributed scheduling (SDS), the research issues are

(1) What traditional scheduling methods and techniques can be combined and reused in SDS and

(2) What are new methods and techniques required for SDS.

Therefore, in this paper, we first review existing researches aiming to answer the first question and discusses a future research direction in SDS to reduce the complexity of centralized scheduling and support smart manufacturing systems[3].

All feasible schedules can be classified into three classes, i.e., semi active schedules, active schedules, and non-delay schedules [4]. A feasible schedule is defined as a semi active

 https://doi.org/10.37896/pd91.5/91517

ISSN: 0369-8963

schedule if no operation can be started earlier without changing the operation sequence. A semi active schedule is defined as an active schedule if no operation can start earlier without delaying another operation or without violating the precedence constraints [5]. Finally, an active schedule is defined as a non-delay schedule if no machine is kept idle when it can start processing an operation. Thus, the solution space of active schedules is a subset of the solution space of semi active schedules, and the solution space of no delay schedules is a subset of the solution space of active schedules. The solution space of active schedules is surely dominant over the solution space of semi active schedules since it is smaller and also guaranteed to contain an optimal schedule. The solution space of non-delay schedules, which is the smallest solution space, may not contain an optimal schedule [6]. In present, many optimal methods are suggested to solve the JSSP problem, including traditional operations research methods, heuristic rules, artificial intelligence, simulation methods, neural networks, fuzzy theory, Lagrangian relaxation, Meta-heuristic (such as Simulation Annealing SA, Genetic Algorithm GA, Tabu Search TS, Ant Colony System) and so on [7]. Among them, the Genetic Algorithm as a global search algorithm for job scheduling has risen great interest of researchers.

LITERATURE SURVEY

An effective crossover operator used to improve the algorithm's search ability and prevent premature convergence. A slack-based insertion rescheduling strategy is developed to handle new job insertions in the schedule. Taguchi analysis is employed to identify the best combination of algorithm parameters. Finally, we generate new benchmark instances for the flexible job shop scheduling problem with new job arrivals [8]. Hierarchy Mathematical Modeling Approach (HMMA) has been developed to integrate the production resources planning and job shop scheduling. In which, material requirement planning system for material resource arrangement, employee timetabling module for human resource allocation and manufacturing resource planning for machine allocations are to be considered. For solving the unique hierarchy model, a shuffled frog leaping heuristic algorithm (SFLA) is proposed and implemented for minimizing the overall production cost [9]. A novel Biomimicry Hybrid Bacterial Foraging Optimization Agorithm (BHBFOA) is developed, which is inspired by the behavior of *E. coli* bacteria in its search for food. The developed BHBFOA search method is hybridized with simulated annealing (SA). Additionally, the algorithm has been enhanced by a local search method based on the manipulation of critical operations. Classical dispatching rules have been employed to create the initial swarm of BHBFOA, and a new dispatching rule named minimum number of operations has been devised [10-12]. Timed Petri nets used to describe the dispatching processes of the job shop scheduling scenarios. On this basis, a data mining based scheduling knowledge extraction framework is developed to mine the expected scheduling knowledge from the solutions generated by traditional optimization or approximation method for JSSP. Based on this, we show how to use the extracted knowledge as a new dispatching rule to generate complete schedules. A novel method is further developed to combine the extracted knowledge with traditional heuristics to construct new composite dispatching rules which could gain better

 https://doi.org/10.37896/pd91.5/91517

performance [13-15]. A new job shop scheduling problem is studied with the option of jobs outsourcing. The problem objective is to minimize a weighted sum of make span and total outsourcing cost. With the aim of solving this problem optimally, two solution approaches of combinatorial optimization problems, *i.e.* mathematical programming and constraint programming are examined. Furthermore, two problem relaxation approaches are developed to obtain strong lower bounds for some large scale problems for which the optimality is not proven by the applied solution techniques. It is concluded that constraint programming outperforms mathematical programming significantly in proving solution optimality, as it can solve small and medium size problems optimally [16-17].

RESEARCH METHODOLOGY

In the job shop environment, scheduling is significant to obtain minimized computational time and it is done by proposed method as shown in the Figure-1

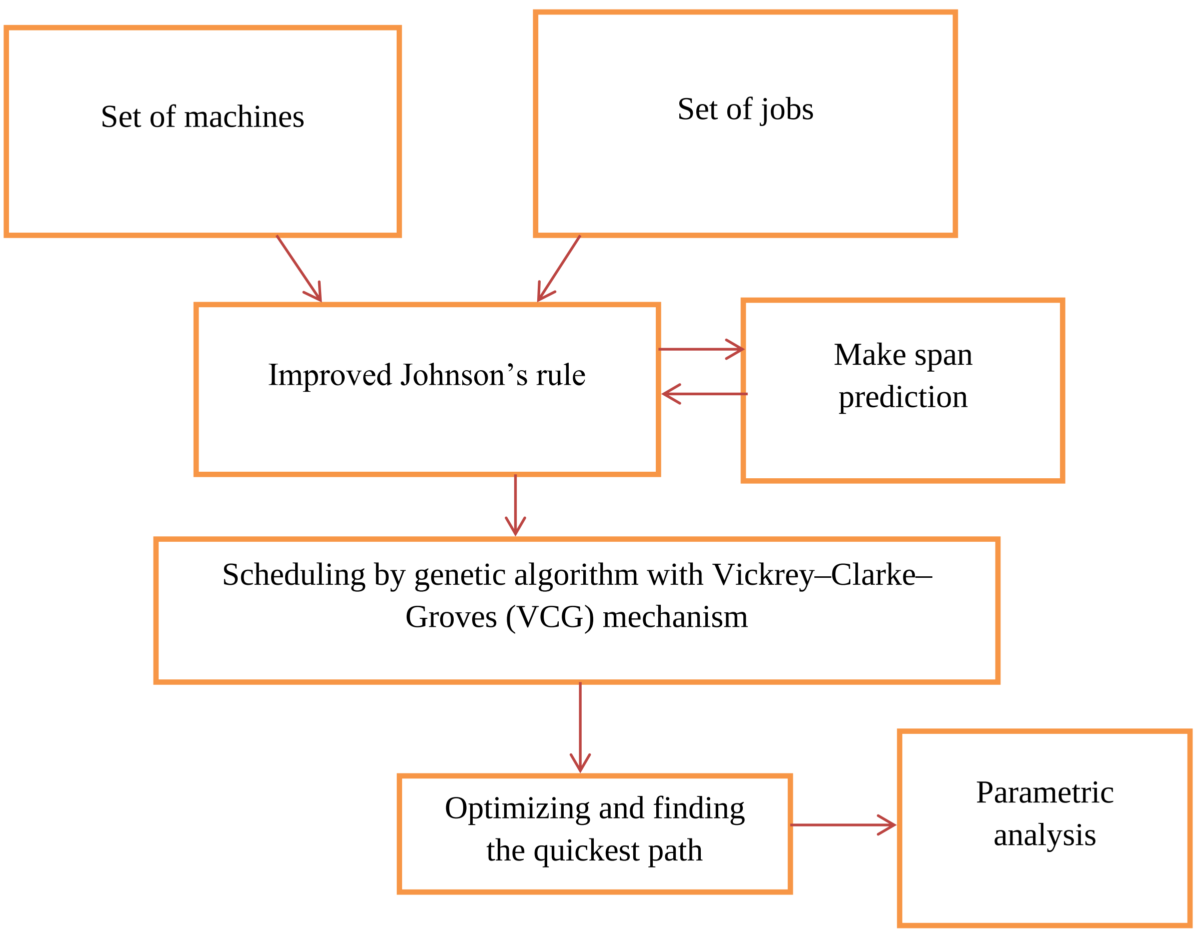

Figure-1 Architecture of proposed mechanism for scheduling

 https://doi.org/10.37896/pd91.5/91517

ISSN: 0369-8963

Initialization of machines

Here, the jobs are indicated as 'X' and machines are indicated as 'Y' where, group of jobs X={1,2,3….x} and group of machines indicated as Y={1,2,3….y}. Initially, the path processing time has to fix for every job. These set of jobs are considered for y number of machines. Secondly, each job is adopted for each machine by means of buffering process. Consider that all the machines are build with buffering capacity. The reason for buffering process is, while running the current job by a machine, this process make available of next machine for another job. This process overcome the problem of blocking, whereas once the finalized job e.g., x gets finished, it can leave from the production process by leaving the buffer stack.

Improved Jhonsons's Rule for Prediction (IJRP)

The IJRP is using here to neglect the productivity problem and buffering problem by considering the following assumptions.

- It is significant to find the processing time of every machine with respect to its order. Automatic migration of job from one machine to another machine is negligible.
- Proper computation of job by a machine is necessary for introducing next job
- It is mandatory of processing single job by a machine within allotted time.
- All jobs must be assumed with notation before undergoing to machine.

Once these assumptions are considered, the initialization of jobs and machines are ready for production. Let us consider two jobs initially, which is indicated as Xi, Xj∈ Y with minimum and maximum notation. The term ai, bi, are the buffer jobs in waiting mode.

Lemma1: Consideration of processing time with fewer periods is adopted for all the unscheduled jobs. If all these unscheduled jobs are for single machine then, make a job schedule for last open position. By this way, remove all the jobs from job set and repeat the process continue till all the jobs get scheduled.

Property 1: consider the job sequence (Xj,Xj+1,…..Xi) with the in equality condition aj ≤ a[j:i] and bj ≤ b[j:i]. For the purpose of apply swapping; there is no need of considering consecutive mode of nodes. So the following cases provide the proper solution.

Case 1: Xj whereas Xj∈ N +

- initially, consider the set of composite job with sequence (Xj,Xj1,….Xjk)
- If X*j is negative, then the sapping process happens between X*j and Xj and hence ends up with new sequence (X*j,X*j1,….X*jk) with the preceding of xi

 https://doi.org/10.37896/pd91.5/91517

ISSN: 0369-8963

- If Xj is positive, then the equation a*j≥aj≥ai is considered by swapping X*j and Xi and hence ends up with the sequence (X*j,X*j1,….X*jk)

Case 2: Xj whereas, Xj∈ N –

- In this case, the set of composite jobs X*I is indicated by (X1,X2,X3,….XkXi) with positive notation and hence swap by X*I and X*j can endsup with the new sequence (X1,X2,X3,….Xk,Xi,Xj) with the preceding of Xj.
- If the set of jobs X*i is negative, then according to the equation b*j≥bj≥bi with swapping concept is done.

Case 3: Xj∈ N (–) and Xj∈ N (+)

If the set of job sequence (Xj1, Xj2,….Xjk) is negative, then the swapping process happens between Xi and consecutive job such qas Xi and Xj. Finally the derived sequence is {Xi, Xj,Xi1,Xj1,Xi1,Xj2,….Xjk, Xik}.Now the operation is ready for schedule

Scheduling by genetic algorithm with Vickrey–Clarke–Groves (VCG) mechanism

Once the jobs are ready in buffer the scheduling process is done by genetic algorithm with Vickrey–Clarke–Groves (VCG) mechanism, whereas it depends upon auctioneer ad bidder's concept. The auctioneer is consider as jobs and the bidders are considers as machines.

- Initially, the auctioneer (i.e., job sequence) is search for machines by sending the request For Proposal (RFP) to all machines. This RFP consists of fundamental information such as quality of service, composite parameters and require the list of buffer for scheduling process. The QOS considered are response time of machine, reliability of machine and availability of jobs.
- Once the machine gets the request, it has to submit the pay cost to the required job, if the machine is really interested in particular auction. This can be done by concentrated on all type of quality of service. For example, let ui, j was considered as QoS based parameter with j^{th} auctioneer process. Then it can be indicated as ui,j={α_i,$\beta_{i,j}$, $R_{i,j}$, $cost_{i,j}$} by considering the computational cost, guaranteed service respectively.
- Based on the QOS parameter the schedule will start with the consideration of multi-criteria decision making and winner determination process. As a result, the schedule is done by calculation.
- The calculation is based on payee concept by the auctioneer which concludes the winning machine at the end. The combinational process makes the machine to obtain proper computational cost. Hence, for every machine 'M', it is important to find the optimal constraint which is done by genetic algorithm.

https://doi.org/10.37896/pd91.5/91517

- Let ‘Mx’ and ‘My’ be the set of machines with jobs are arranged in scheduled manner with crossover and mutation point. Let d0 (t) is the available set of jobs at time t and m is the index value of each job. The flow of algorithm is as follows:

Step-1: indicate d0 (t) as available set of jobs at time ‘t’ with length ‘L’ of chromosomes

Step-2: initialize the criteria

1) Set t=t+1
2) Calculate d(t) from d0(t) as follows
   Apply local and global mechanism
   Evaluate the QOS of machine (z)
   Evaluate the objective function /z/
      /z/=aj(m)+z*bj(m)-zm(o)
      Where,
         J(m)∈bj(m) with {i:a*b=1} where ‘m’ is the machine and bj(m) is the bidder of machine (m)
3) Rank the values based on threshold
4) Obtain the decision making method to find the best solution

Step-3: End the calculation

Mi=(m*1,m*2,m*3……m*i-1) with the fitness value as /z/*-z(m)=$\sum_{k=mi} ci - bi$

Step4: Evaluate the present individual p(t) with the value between 0 and 1

Step-5: consider the individual by finalizing the optimal value as,

$$\sum_{i=bm} ci - bi = 0$$

$step -$ 6:Select the individuals po1 and po2with crossover operator

$step -$7: make the offsprings by consider L(x) as$\sum_{j=Bm} cj$

$$If\ L > Lx\ then\ select\ the\ induvidual\ by\ crossing\ the\ mutation$$

$$else\ continue\ and\ save\ the\ optimal\ \ paths.$$

ISSN: 0369-8963

OPTIMIZING THE PATH

The reason for optimizing the path is to repair or replace the fault machine due to technical issues. Due to this, the jobs in schedule can be affected which is in holding time. Particularly, for optimizing path, a strategy named as shifting schedule can be done which makes the entire schedule to push the right direction. Hence, the optimization is done by definitions as below.

• The task of the repaired machine is put separate category as reactive scheduling for further classification.

• Any task that needs to be relocated due to the interruption is classified as affected. The set of affected tasks is generated based on the precedence relationships of the tasks. At any instance, if the shifting of a task (because of breakdown) does not affect its successor task, then the successor task is not considered as affected.

PERFORMANCE ANALYSIS

The experiments conducted to test the algorithms use well-known benchmarks. The benchmarks are obtained from the standard job-shop benchmark problems. The parameters used in the algorithms are as follows. The number of initial feasible solutions for each instance, is defined as 200.The number of iterations to optimize each initial solution in the local search, is defined as 100; and parameter, the probability of two solutions exchanging their processing sequences on one or more machines, and is defined as 0.9.

Extensive experiments were carried with the proposed Vickrey–Clarke–Groves Mechanism (SGAVCGM) and the results obtained were compared with the existing algorithm namely Biomimicry Hybrid Bacterial Foraging Optimization Algorithm (BHBFOA) and Hierarchy Mathematical Modeling Approach (HMMA). For parametric analysis scheduling efficiency, power consumption, computational cost and speed are considered.

Scheduling Efficiency

It is defined as the schedule made for each machines by consolidating the ratio of all machines idle time to overall processing time, which is denoted as follows

$$SE=\frac{m,n}{\sum ci,jk} \quad (1)$$

Where, m is the make span

N is the total number of machines

Ci is the idle time

Jk is the processing time

 https://doi.org/10.37896/pd91.5/91517

The table 2 analyzes the network lifetime of the existing BHBFOA, HMMA and proposed methods SGAVCGM method.

Table 2 Analysis of scheduling efficiency

Time in seconds	BHBFOA	HMMA	SGAVCGM
20	29	37	48
40	40	41	64
60	43	46	67
80	46	58	78
100	52	74	93

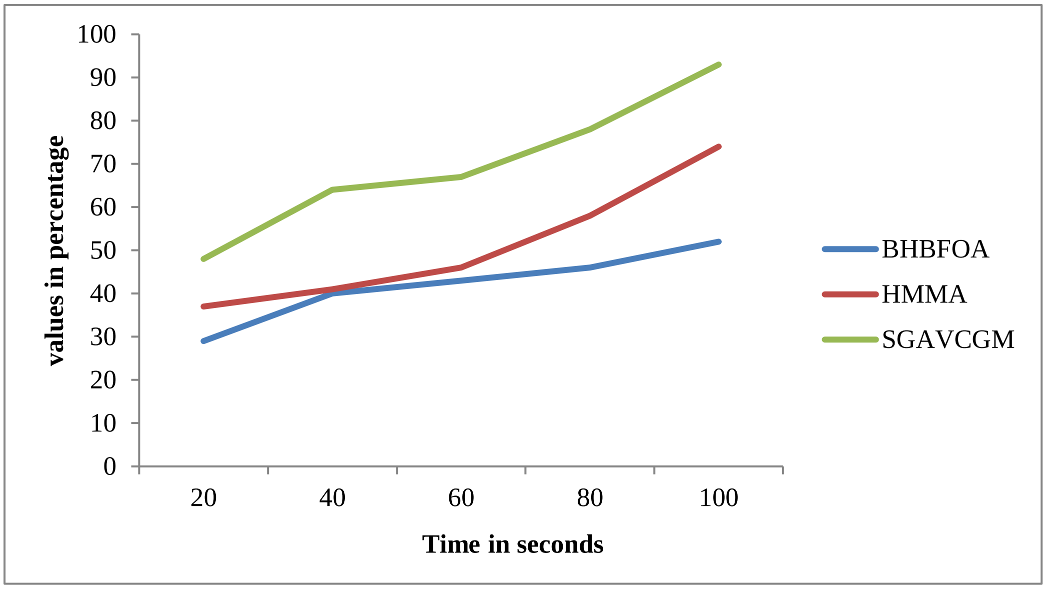

Figure 2 Comparison of scheduling efficiency

Figure 2 depicts the scheduling efficiency comparison of existing BHBFOA, HMMA and proposed methods SGAVCGM method. The X axis and Y axis shows that number of nodes and the values obtained in percentage respectively. The existing method achieves 52% and 74% for 100 seconds and hence the proposed method achieves 93% which is nearly 41% improved than BHBFOA and 19% improved than HMMA.

- **Computational cost (change)**

  It is defined as the total amount of jobs of each machine in sequential order (s) with its consecutive process|(c). Hence, by comparing all the machines, the complexity is depends upon the cost which is denoted as ‘o’ (nlog2)

  The table 2 analyzes the computational cost of the existing BHBFOA, HMMA and proposed methods SGAVCGM method

Table 3: Analysis of computational cost

Time in seconds	BHBFOA	HMMA	SGAVCGM
20	50	40	18
40	59	53	25
60	68	58	44
80	80	66	47
100	89	76	48

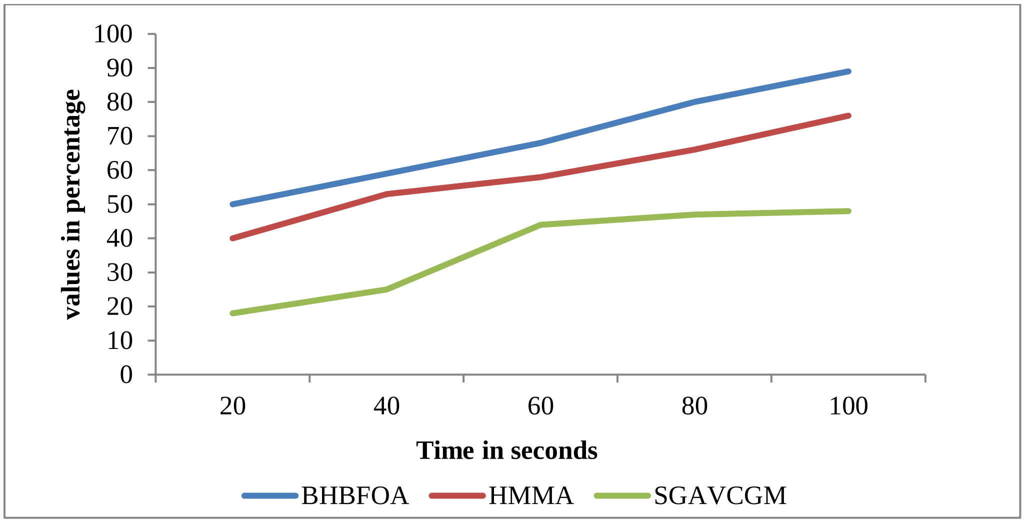

Figure-3 Comparison of computational cost

Figure 3 depicts the computational cost comparison of existing BHBFOA, HMMA and proposed methods SGAVCGM method. The X axis and Y axis shows that number of nodes and the values obtained in percentage respectively. The existing method achieves 89% and 76% for 100 seconds and hence the proposed method achieves 43% which is nearly 41% improved than BHBFOA, method and 19% improved than HMMA method.

Speed

It is calculated based on making the curve by analyzing the running time using the linear scale analysis by neglecting the problem. Hence, the running time of machine is inversely proportional to the optimal problem neglecting.

The Table 4 analyzes the throughput of the existing BHBFOA, HMMA and proposed methods SGAVCGM method

Table 4: Analysis of speed

Time in seconds	BHBFOA	HMMA	SGAVCGM
20	48	51	57
40	55	58	73
60	66	67	83
80	72	81	87
100	75	86	96

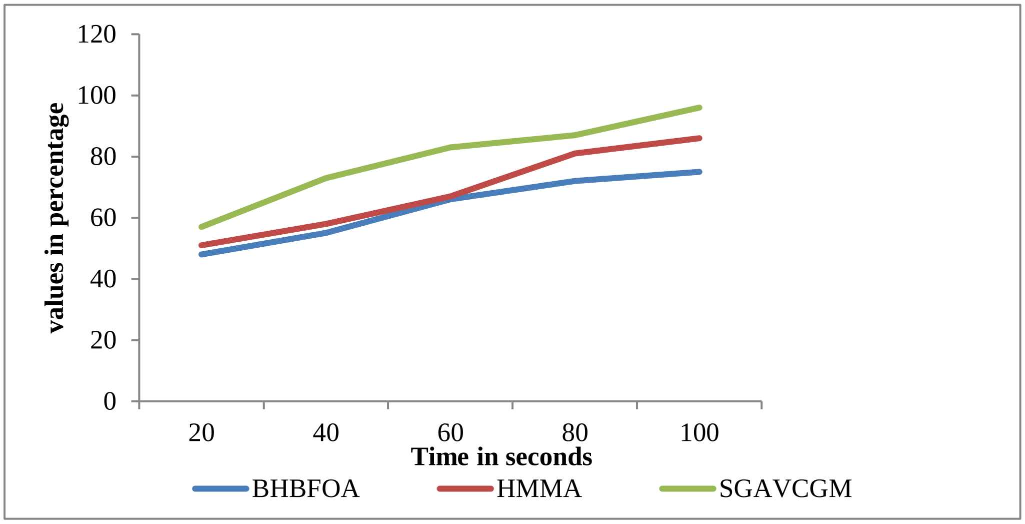

Figure-4 Comparison ofspeed

Figure 4 depicts the speed comparison of existing BHBFOA, HMMA and proposed methods SGAVCGM method. The X axis and Y axis shows that number of nodes and the values obtained in percentage respectively. The existing method achieves 75% and 86% for 100 seconds and hence the proposed method achieves 96% which is nearly 21% improved than BHBFOA, method and 10% improved than HMMA method.

Power efficiency

It is defined as the total amount of power reduces by choosing the optimal path by making the rescheduling method.

$$Power = \frac{1}{p}\sum_{n}^{p} E_n \quad (2)$$

The Table 4 analyzes the power efficiency of the existing BHBFOA, HMMA and proposed methods SGAVCGM method

Table 5: Analysis of power efficiency

Time in seconds	BHBFOA	HMMA	SGAVCGM
20	19	36	59
40	23	46	63
60	31	53	65
80	33	56	67
100	36	57	69

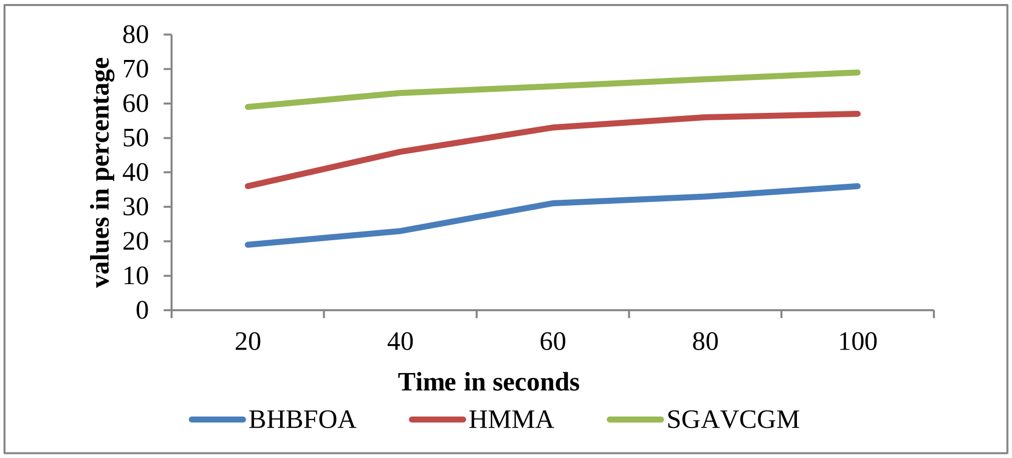

Figure 4: Comparison of power efficiency

Figure 4 depicts the power efficiency comparison of BHBFOA, HMMA and proposed methods SGAVCGM method. The X axis and Y axis shows that number of nodes and the values obtained in percentage respectively. The existing method achieves 36% and 57% for 100 seconds and hence the proposed method achieves 69% which is nearly 33% improved than BHBFOA method and 12% improved than HMMA method.

CONCLUSIONS

This article addresses the job shop problem with the objective of minimizing the process makespan. An SGAVCGM method is proposed to describe the problem. According to the model, a feasible solutions and a local search is proposed to solve the issue. The local search has two operators they are auctioneer structure and bidder structure based on a Vickrey–Clarke–Groves (VCG) mechanism, focused on reducing or eliminating the waiting time of operations in the buffer..Computational results are presented for a set of benchmark tests, some of which are enlarged by different proportions between the capacity of the buffer and the number of jobs, and the effectiveness of the proposed algorithm is demonstrated by comparing it with two existing algorithms.

 https://doi.org/10.37896/pd91.5/91517

REFERENCES

1. Baykaso glu. A, Hamzadayi. A and Kose S. Y, “Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases”, *Information Sciences*, vol. 276, pp. 276, 204–218, 2014.
2. Harmanani H. M and Ghosn S. B, “An efficient method for the open-shop scheduling problem using simulated annealing (chapter). In Information technology: New generations Berlin: Springer, pp. 1183–1193, 2016.
3. Iwamura, K. and Sugimura N. A, “Study on real-time scheduling for autonomous distributed manufacturing systems. *In 2010 IEEE international conference on systems man and cybernetics (SMC)”*, pp. 1352–1357, 2010.
4. Neto R. F. T and Godinho Filho M, “Literature review regarding ant colony optimization applied to scheduling problems: Guidelines for implementation and directions for future research”, *Engineering Applications of Artificial Intelligence*, vol. 26, no.1, pp. 150-161, 2013.
5. S.-J. Wu and P.-T. Chow, “Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization,” *Engineering Optimization*, vol. 24, no. 2, pp. 137–159, 2005.
6. T. Brys, M. M. Drugan and A. Nowe, “Meta-evolutionary algorithms and recombination operators for satisfiability solving in fuzzy logics,” in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, IEEE, Cancun, Mexico, pp. 1060–1067, 2013.
7. P. Pongchairerks and V. Kachitvichyanukul, “A two-level particle swarm optimization algorithm on job-shop scheduling problems,” *International Journal of Operational Research*, vol. 4, no. 4, pp. 390–411, 2009.
8. Caldeira Rylan H, A. Gnanavelbabu and T. Vaidyanathan. “An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption", *Computers & Industrial Engineering* 149, 106863, 2020.
9. Ramya G., M. Chandrasekaran and P. Arulmozhi. “Optimization of production cost for integrating job shop scheduling with production resources”, *Materials Today: Proceedings*, 2020.
10. Vital-Soto A, Azab A and Baki M. F, “Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility”, *Journal of Manufacturing Systems*, vol. 54, pp. 74-93, 2019.
11. Safarzadeh H and Kianfar F, “Job shop scheduling with the option of jobs outsourcing, *International Journal of Production Research*, vol. 57, no.10, pp. 3255-3272, 2018.
12. Wang C. L, G. Rong, W. Weng and Y. P. Feng, “Mining scheduling knowledge for job shop scheduling problem”, *IFAC-Papers OnLine* , vol. 48, no. 3, pp. 800-805, 2018.
13. M.Babu and T.Jesudas, “An artificial intelligence-based smart health system for biological cognitive detection based on wireless telecommunication”, *Computational Intelligence*, Vol. 38, no. 4, pp. 1365-1378, 2022.

ISSN: 0369-8963

14. M. Senbagavalli and G Tholkappiaarasu, “Opinion Mining for Cardiovascular Disease using Decision Tree based Feature Selection”, *Asian Journal of Research in Social Sciences and Humanities*, vol. 6, no.8, pp. 891-897, 2016.

[15] S. Debnath, K. Ramalakshmi and M. Senbagavalli, “Multimodal authentication system based on audio-visual data: a review”, in *Proce. 2022 International Conference for Advancement in Technology* (ICONAT), IEEE, 21660941, Goa, pp. 1-5, 2022.

[16] Jesudas T, Ramesh S, Arunachalam RM and Senthilkumar K, “Study and optimization of overcut on most influencing process parameters of micro edm, *International Journal of Applied Engineering Research*, vol. 10, no. 50, 13678-13683, 2015.

 https://doi.org/10.37896/pd91.5/91517