Minimizing the Lateness, Makespan, And Energy Efficient Factors in Job Shop Scheduling

K. Anandapadmanabhan[a] and Dr.V.P. Eshwaramurthy[b]

[a]Assistant Professor & Head Of Computer Science, Sri Vasavi College, Erode, Tamilnadu, India. Research Scholor Bharathiar University (Category B). E-mail:kapn0305@gmail.Com
[b]Assistant Professor of Computer Science, Government Arts and Science College, Komarapalayam, Namakkal District, Tamilnadu, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Half of the worldwide energy is consumed by industries. In the modern era, due to an increase in the energy overhead, it is necessary to plan during electricity production. In this paper, a novel mathematical model is proposed which determine a more efficient scheduling algorithm that reduces the electricity consumption cost (ECC) and make-span for the Flexible Job Shop Scheduling Problem (FJSSP) with a time-of-use (TOU) policy. Another subtask named speed selection, which indicates the selection of varying operation speed is added to already available two subtasks of FJSSP. Subsequently, a Modified Cooperative Co-evolutionary Cuckoo Search Algorithm (MCCCSA) along with Variable Neighborhood Search (VNS) is used for resolving the constrained, unconstrained optimization and engineering issues. The population used for this algorithm is an organization that contains dynamic persons. Experimental analysis is done to check the performance of the proposed model MCCCSA+VNS for improved scheduling result when evaluated with other existing methods.

Keywords: Flexible Job Shop Scheduling Problem (FJSSP), Modified Cooperative Co-evolutionary Cuckoo Search Algorithm (MCCCSA), Variable Neighborhood Search (VNS).

1. Introduction

Schedule is the allocation of the process to the time slots of the machines. A Job shop is a production setting that produces a large range of batch of products in a similar facility (i.e., less volume product). Every order is routed independently to their exclusive work centres. Based on the workflow patterns, the varied quantities and their execution time, work-in-process inventories and queues are generated. The scheduling issues exist in various fields like logistics, planning the production and manufacturing process (Mokhtari & Hasani, 2017). When compared to other production scheduling issues, the job shop problem is a more distinctive and complicated problem.

The job shop problem or job shop scheduling is one of the optimization problems that allocate ideal jobs to the resource at the given time. Multiple objectives are involved in real world job shop problems. Only minimum trials exist to solve the multiple objective JSP. In multi-objective JSP, more than one contradictory objectives are taken. A group of the optimal solution is considered for multi-objective optimization problems i.e., Pareto optimal solution includes two or more optimal solution. Normally, in Pareto optimal solution considering one solution than the other is difficult. For multi-objective optimization problems, the heuristic method of selection can be considered for finding the Pareto optimal solution for multiple criteria (Sundar et al., 2017).

The scheduling process is considered to be a decision-building task for many servicing and manufacturing industries. The main aim of scheduling is to optimize more than one objective by allocating the resources to the ideal task on time. An organization can have a task and their resource in diverse forms. Resources can be of many forms like crews in construction site, machines at the workshop, processing units in the computational area, runways in airports, etc. Similarly, tasks can be stages of construction, function in the production procedure, program execution, landing and take-off in the airport, etc. Every task has a priority level with an early start time and completion date. There is different form of objectives like reduction of completion time of the last job, reducing the number of jobs finished after their corresponding due date (Asadzadeh, 2015). Scheduling is more important in major manufacturing and servicing industries along with information processing systems.

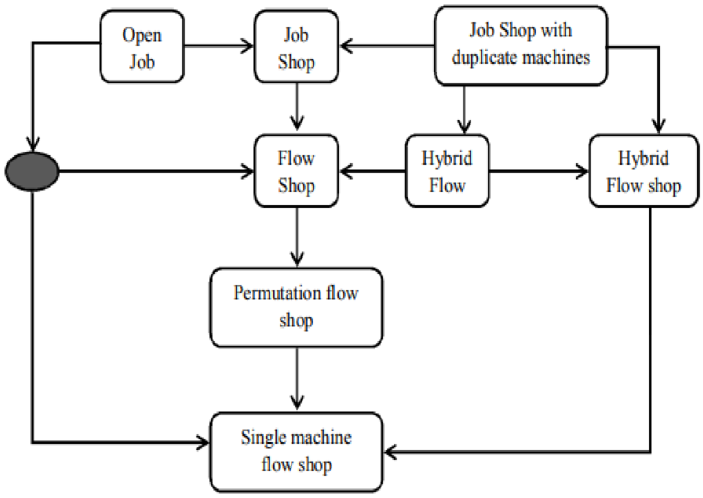

Figure 1. Example of Job Shop Scheduling Problem

Figure 1 shows the example of job shop scheduling problem**.** JS scheduling problem is one of the NP-hard problems. In (Wang, 2012) various genetic operators were considered to solve the JS scheduling problem. A mixed selection operator is used based on the concentration value and fitness to increase the multiplicity of the population. A crossover operator depending on the machine and mutation operator related to a critical path was designed to utilize the characteristics of the problem completely. A new algorithm was used to identify the critical path in the schedule. To improve the local searching ability of the genetic algorithm, a local search operator was used. Considering all the above, a hybrid GA was used and it is proved for its convergence

The aim of this research is to solve the job shop scheduling problem with efficient optimization method. Many research and methods were initiated but the cost reduction and execution speed was not guaranteed. The limitation of existing methodologies is make-span result, energy utilization and lateness. To address the above listed issued, a Modified Cooperative Co-evolutionary Cuckoo Search Algorithm (MCCCSA) along with Variable Neighborhood Search (VNS) is used for solving the constrained and unconstrained optimization issue. The main contribution of the work is FJSSP modeling and using the MCCCSA algorithm to find optimal solution. The proposed work reduces the lateness, energy cost with increasing speed

Remaining sections are structured as: related work on FJSSP and job shop scheduling is explained in Section 2, FJSSP+MCCCSA proposed method is provided in Section 3, experimental analysis and discussion is presented in Section 4, the conclusion is given in Section 5.

2. Related Work

Kaplanoğlu, (2016) has proposed an associate Object-Oriented (OO) approach to solve multi-objective FJSSP along with simulated annealing optimization algorithm. The resolution method within the work used two string cryptography idea to indicate this issue. OO method of FJSSP is done by discrimination of UML type diagram which reduces the issue of cryptography to the organization. Here the operation object of FJSSP contains the information related to varied machines in its hierarchical information structure. Three major objectives were considered employment of the foremost loaded machine, total employment of all machines and most completion time. To improve the projected method benchmark sets were executed.

Authors have examined the Job Shop Scheduling Problem (JSSP) for the transportation time of jobs between the machines. The aim of the basic JSSP is to identify the start and end time of every job where the optimization of the objective function can be done. A number of Automated Guided Vehicles (AGVs) is used to transmit the jobs between the warehouse in the production environment and the machines. Minimizing the completion time i.e. make-span is considered as the objective function. To prove the efficiency of the algorithm and proposed model, the computation outcome of 13 test problems with sensitivity was shown. The results indicate that the ACA is a better meta-heuristic for large-sized problems. The best possible number of AGVs and railways in the production setting is identified by economic analysis (Saidi-Mehrabad et al., 2015).

Kundakcı & Kulak, (2016) debated job shop scheduling mainly focused for past 10 years and most of these ideas were anticipated to address the static job shop scheduling problems. The measures like machine failure, privilege time variation, and random job arrival in production atmosphere are not considered in static job shop scheduling problem.

Authors have established a bi-criteria stochastic flexible flow shop (SFFS) scheduling problem where one criterion is qualitative and the other is quantitative. The analysis considers simulation-optimization approach in mixed integral linear programming (MILP) model for total weighted lateness solution. Based on the customer preference, the alternatives are qualified by ordinal analysis through stochastic multicriteria acceptability analysis with ordinal data (SMAA-O). The integral analysis is done using deterministic SMAA by selecting the alternatives with the best integral characteristics related to reduced lateness, penalty cost, timely completion of due date based on customer strategic significance for the company. Experimental analysis indicates that IAM provides the best way of selecting the alternatives considering the criteria (González-Neira et al., 2016).

Wu et al., (2016) investigated transmitting guideline based GA with fuzzy satisfaction levels (FRGA) to solve multi-target fabricating scheduling problem. The approach focuses on the scheduling issues like reducing make-span, maximal delay, average stream time and total delay. The two-level fuzzy method evaluates each chromosome to indicate the satisfaction level. The outcome of FRGA parameters was analyzed using many trials.

In Ahmadi et al., (2016), authors focused on stable scheduling of multi-objective problem under flexible job shop scheduling with random machine breakdown. The transformation between scheduling and after machine breakdown that makes the usage of resources critical was prevented by the proposed method. Scheduling stability is identified by the deviation between the start and finishing time of every job between realized schedule and pre-schedule. NRGA and NSGA-II algorithms were combined to improve the stability and make-span concurrently.

Authors have proposed an improved adaptive particle swarm optimization algorithm to reduce the make-span for job planning problem. Considering the internal secretion modulation mechanism, adaptive secretion issue is used in the transformation of the equation in PSO. The unfair value of HF makes each particle of the swarm adjust its position to attain a higher value and avoid premature. Experimental result shows that proposed IAPSO attain high quality solution with less computation time (Gu et al., 2012).

Akbari & Rashidi, (2016) proposed a new algorithm for heterogeneous system task scheduling. The proposed method is a multi-objective algorithm using a cuckoo optimization algorithm (MOSCOA). Here, every cuckoo indicates the solution for scheduling like ordering of processor and task allocated to them. Scheduling at every stage is performed by the operators of the cuckoo optimization algorithm where local optima are avoided by enhancing the global search to find an optimal scheduling solution with minimum number of cycles. The target immigration operator is used to move towards global optima and every schedule at each cycle is moved to optimized schedules in order to protect the global optima. The proposed method was analyzed using random graphs and real-time application graph which indicated that MOSCOA performs better than other existing task scheduling algorithms. In MOSCOA, parallel task execution is performed with reduced completion time.

3. Proposed Emthodology

In this work, MCCCSA is proposed to improve the lateness, make-span and energy efficient factors for the FJSSP. MCCCSA involve the modeling of FJSSP and optimization.

3.1. Modeling of FJSSP

By (Dai et al., 2019) $n \times m$ FJSP is defined as: with a collection n jobs $j = \{j_1, j_2, , \ldots . j_n\}$ and of m machines $M = \{M_1, M_2, \ldots . M_m\}$. Every job i contains a sequence of operations $\{o_{i1,} o_{i2}, \ldots . . o_{in}\}$ where ni indicates the series of operations that job i contains. Every operation oik is processed by any machine in the machine set $M_{ij} \subseteq M$.

The issue is to identify assignment and operation sequence on the machine to satisfy the criteria. Basically, it has two issues: operation sequencing problem and machine identification problem. Hence the FJSP is a challenging and complicated problem than standard JSP as it needs an appropriate identification of machine among the set of given machines to perform every operation of all the jobs.

Single job contains many operations and they cannot be executed concurrently. One job does not dependent on another so preemption of jobs is not possible and for a time, every resource can be allocated for a single job. At zero time, all the resources and jobs will be available. The setup time in the machine for every operation does not depend on the operation cycle and they are included in the execution time. After processing every job, it is transmitted to other machines for its process and this time is considered to be insignificant.

The limitations for machines and jobs are:

- Every job can visit the machine only once
- No constraints on the priority of operations of varied jobs

- Operation preemption is not permitted
- Every machine can execute one job at a single time
- Every job can be executed in one machine at a time
- The due date or release time of the job is not mentioned

The issue is to identify a schedule that reduces the make-span (time taken to finish all the jobs)

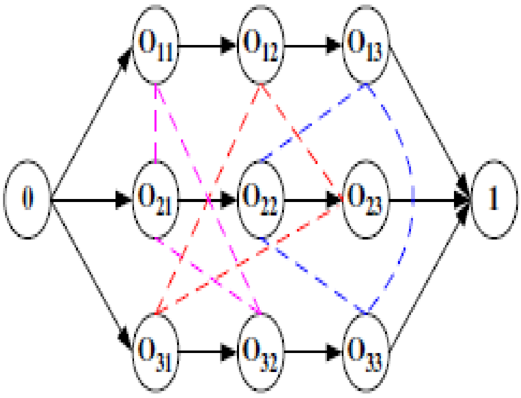

Figure 2. The Model of the Problem

Let W be the model of the problem as indicated in figure 2. When the directions in the non-directed edges Ei (where i=1,2,3) are fixed and create a path connecting all the nodes of Ei, a schedule is identified for this problem (as indicated in Fig.4). Here the directed dashed lines are used to show the edges in Ei (i=1,2,3).

The routing of many operations for the same job and the connection for each pair of successive operation of the similar job is indicated by using conjunctive arcs. Two operations of varied jobs that are to be executed in the same resource are indicated by a pair of disjunctive arcs and these arcs form a circle for every machine. The optimal solution is identified in an acyclic subgraph that has all the conjunctive arcs and only a single disjunctive arc for every pair of disjunctive arcs between any two nodes. The optimal solution is to find a viable subgraph with reduced make-span. Figure 2 depicts an example of a disjunctive graph for the JSSP with three jobs and machines. 3×3 job shop disjunctive graph representation is shown in figure 3 & figure 4 shows the solution of the problem.

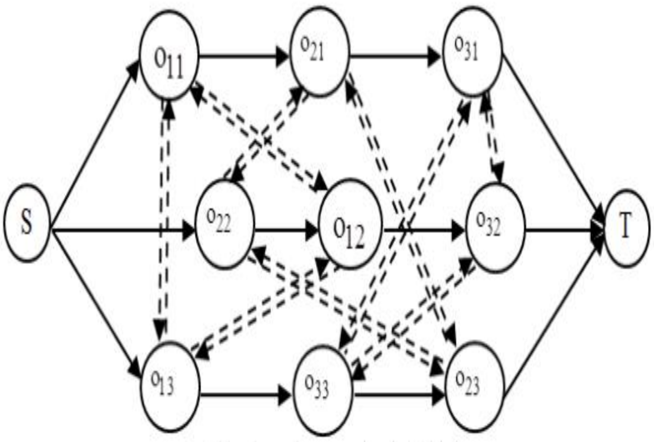

Figure 3. Disjunctive Graph Representation of a 3×3 Job Shop

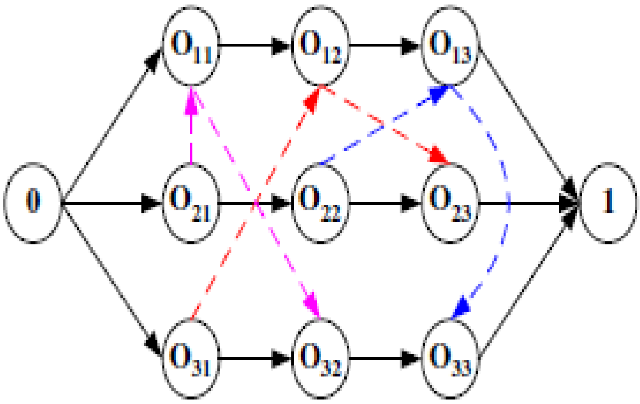

Figure 4. A Solution of the Problem

3.2. Optimal Solution Using MCCCSA Algorithm

This proposed work combines MCCCSA with Variable Neighborhood Search (VNS) for resolving the constrained and unconstrained optimization problems. Motivation from the organizational evolutionary algorithm, a cooperative coevolutionary cuckoo search algorithm (CCCS) is considered that combines both cooperating and annexing operators as the main part of the cuckoo search algorithm. This model aims to increase the performance and diversity of the cuckoo algorithm.

Basically cuckoo birds are more attractive due to their pleasant sound and insistent reproduction. Certain species in cuckoos like Guira and Ani lay the eggs in other nests by removing other bird eggs in order to improve their hatching possibility. The Cuckoo search algorithm is an imitation of brood parasitism of certain species of cuckoo where they lay their eggs in other birds nest as their reproductive strategy. When the host bird identifies the eggs that are not theirs, it throws these eggs and built its nest in a different place (Huang et al., 2016). In the algorithm, every egg in the nest is considered as a solution and the cuckoo eggs are a new solution. Checking whether the cuckoo eggs (i.e new solution) is feasible than the old and replace the best solution from the nest. This algorithm was improvised by Levy flights than by isotropic random walks. The cuckoo's habitat is depicted in figure 5.

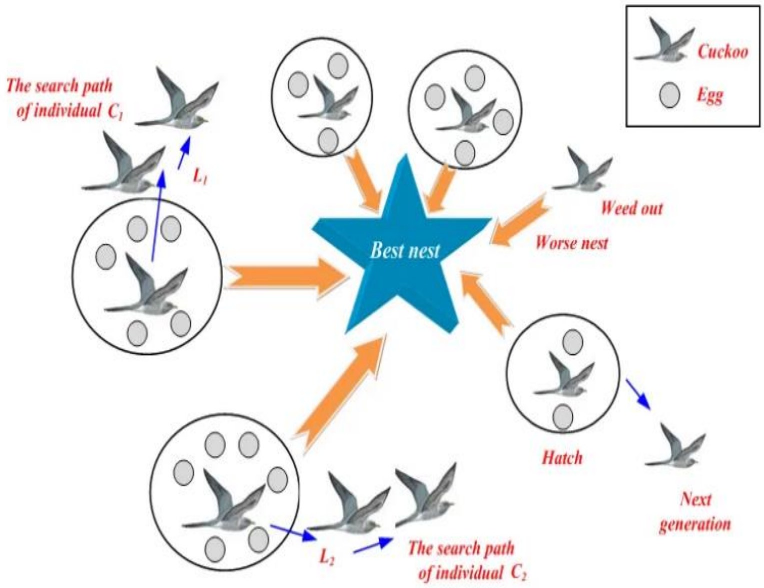

Figure 5. Cuckoo's Habitat

The three ideal rules are followed for the cuckoo search:

- The cuckoo lay only a single egg at a time and place it in a randomly selected nest.
- High feature eggs with the best nest will be considered for the next iteration.
- The number of host nest are preset and the probability of finding the cuckoo's egg by the host is in the range (0,1). The host either throw the egg or built a new nest

The last hypothesis is estimated by the pa fraction of n host nests which are replaced by the new nests (i.e., new random solution). CSO is a simple algorithm with broad search space. The global search is performed by levy flight rather than using the standard random walk to increase the search space.

The CSO algorithm is enhanced by using a different type of nest with many eggs (Kamoona et al., 2018). Basically, cuckoo chooses only three types of host nests to lay their eggs. More common cuckoo bird select a collective host nest that has a similar egg type. Other types of cuckoo choose the host nest with different egg type. Certain cuckoo species lay mysterious egg, that is dark when compared to the host bird eggs to conceal the egg from the eye of the host bird.

1. Initial Population

Here, the selected machines are considered as eggs and the jobs are assigned to them. Based on the top-m rank the machines are selected.

2. Finding New Solutions and Levy Flight

Levy flight is used for CS-based feature selection to identify a new solution based on Equation (4). From the obtained optimal solutions, new solutions are generated through levy walk in order to increase the local search. From levy flight, $x_i^{(t+1)}$ (new solution) is generated for the i^{th} cuckoo and the equation is mention below:

$$x_i^{(t+1)} = x_i^{(t)} + C \oplus \text{Levy}\,(s, \lambda) \tag{1}$$

Where t is the step size. Levy distribution is followed for the step length

$$\text{Levy}\,(S, \lambda) \sim s^{-\lambda}, \quad 1 < \lambda \leq 3 \tag{2}$$

3. Crossover and Mutation

- If the type of cuckoo is common, the crossover is done to produce two eggs and identify best among them.
- If the cuckoo is European, two eggs are formed in the crossover through uniform mutation operator, among which the best is chosen.
- Else cryptic eggs are produced in random

4. Fitness Assignment

The main aim of the fitness function is to identify whether the solution can be transformed to the next generation. The population produced is analyzed based on the given fitness function and a knowledge based operator is used to increase the solution quality of each individual. Fitness function is very important in the selection procedure. By best fitness value, the correct machines are identified from the available resources efficiently. Therefore fitness function should have redundancy and relevance to channel CS in selecting the correct machine and it is signified by Equation (3)

$$\text{fitness (f)} = \text{makespan, lateness and energy consumption (f)} \tag{3}$$

$$Fit.fn\,(Xc) = Cmax - C\,(Xc) \tag{4}$$

Here Xc is the Individual machines with make span as C (Xc) and Cmax represent the maximum value of make span among the entire population

$$Fit.fn\,(Xe) = Emin - E\,(Xe) \tag{5}$$

Here Xe indicates the individual machine that has energy as E (Xe) and Emin represent the minimum energy value in the entire population

$$Fit.fn\,(Xl) = Lmax - L\,(Xl) \tag{6}$$

Xl is the individual machine with L (Xl) as lateness and Lmax is the highest lateness value in the entire population

5. Parameter Pa

In the enhanced cuckoo search, P_a value is modified dynamically using Equation (7)

$$P_a = P_a \max - \frac{P_a \max - P_a \min}{\text{iter\_max}} * \text{iter} \tag{7}$$

Nevertheless, there is a problem of low convergence and computational difficulty. To solve these issues, the proposed method used a modified cooperative co-evolutionary cuckoo search algorithm. In this MCCCSA, the buffer setup time calculation is improvised in order to execute the jobs with an optimal buffer that reduce the lateness of finishing time.

3.2.1. Splitting Operator

For large size organizations, it is divided into many small portions. Consider Maxor as the parameter that controls the maximum size of the organization. In the cooperative co-evolutionary cuckoo search algorithm (CCCS) algorithm, the population is separated into M groups with a leader for every group through collaborative and annexation operation among different organizations. It also applies discrete crossover operator, mutation operator, flip crossover operator, cuboids crossover operator for exchanging information among individuals and promoting the growth of the population. From the simulation results, CCCS optimization shows that it can solve constrained, unconstrained and engineering optimization problems.

3.2.2. Annexing Operator

Let $org_{p1} = \{x_1, x_2, \dots\dots x_M\}$ and $org_{p2} = \{y_1, y_2, \dots .\, y_N\}$ be two organizations that are selected randomly from the present generation. The leaders are chosen using the cuckoo search algorithm. If org$p$1 is success; then org_{p1} will capture org_{p2} to create a new organization, org_c.

3.2.3. Cooperating Operator

Let $org_{p1} = \{x_1, x_2, \dots\dots x_M\}$ and $org_{p2} = \{y_1, y_2, \dots .\, y_N\}$ be two organizations that are selected randomly from the present generation. Consider $CS\epsilon(0,1)$ to be a predetermined parameter. when $rand < CS$, the child organization is produced in Equation 8 and xk, yk are the leaders of organization correspondingly. if not apply (8); where i is a random integer that uses flip operator;

$$q_k = \sigma_k \times x_k + (1 - \sigma_k) \times y_k \quad (8)$$

$$r_k = (1 - \sigma_k) \times x_k + \sigma_k \times y_k \quad (9)$$

$k = 1,2, \dots n$

Here k indicates the selectable machines' collection

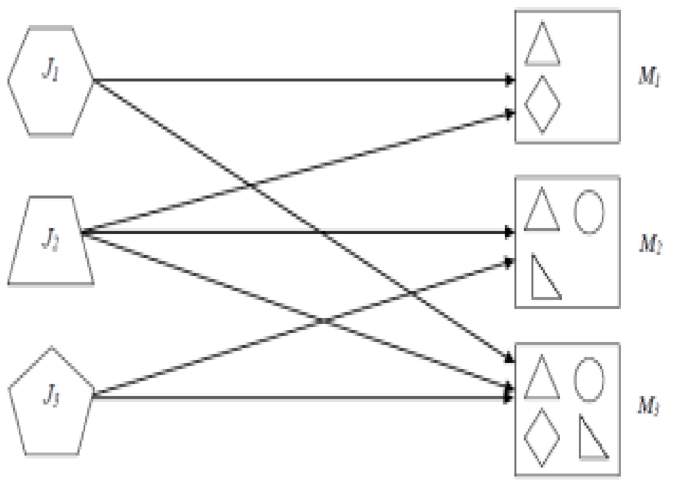

Figure 6. Examples of FJSSP

In figure 6, the problem of 3-jobs on a 3-multipurpose machines are indicated. Here the versatilities and functionalities of different machines are not identical.

Basically, it shows that every jobs do not acquire entire routing adaptability by any resource available at the shop floor.

3.2.4. Variable Neighborhood Search (VNS)

H.G.A Convergence velocity is reasonably low. Therefore, to improve the quality of each individual before transferring to the next iteration, a good local search method is needed (Adibi & Shahrabi, 2014). A modern metaheuristic method named V.N.S constantly looks for neighborhoods with increasing size to decide superior local optima by shaking. Through an efficient analysis process, V.N.S works on the theory that states the systematic change of neighborhood within the local search. V.N.S identify the isolated neighborhood in the present solution and move towards a new solution if the progress is done (Zhang et al., 2019). There are two types of loops in V.N.S namely inner loop that change and search the problem using shaking (Nk S (x)) and local search (Nl LS (x)) and outer loop that repeat the previous loop after diversification.

By diversification of shaking, the solutions can be transferred to other local neighborhood location. An integer k is used to manage the loop length. The outer loop is repeated after completing the inner loop till the termination condition is met. k and l indices are used for shaking and local search respectively. They include higher limits indicated by kmax and lmax and the ranges are represented as 1 ≤ k ≤ kmax and 1 ≤ l ≤ lmax. The Relative Error Index is calculated as mentioned below:

$$R.E.I = \frac{\text{best makespan} - \text{optimum boundary}}{\text{optimum boundary}} \quad (10)$$

Initially every organization contains one member and the population contains $n0$ organizations. In the evolutionary process, to preserve the diversity of the population the number of the organization is changed. In CCCS, the population is modified in the optimization process whereas in OEA, the population remains unchanged and the cooperating operators of both CCCS and OEA are also dissimilar.

By solving the below mentioned objective function, the unconstrained optimization problems (UCOPs) is planned

$$minimize f(x), \ \ x = (x1, x2, \dots xn) \epsilon s \quad (11)$$

Here $s \subseteq \mathrm{R}n$ indicates the n- dimensional space with parametric constraints.

For the constrained optimization problems (COPs) the following objective function is solved

$$minimize f(x), \ \ x = (x1, x2, \dots xn) \epsilon s \cap \chi \quad (12)$$

Here $s \subseteq \mathrm{R}n$ describes the n- dimensional space bounded with parametric constraints and χ indicates the optimal region

$$\chi = \{x \in Rn \mid gj(x) \leq 0, j = 1, 2, \dots, m\} \quad (13)$$

and $gj(x)$ are constraints with $j = 1, 2, \dots, m$.

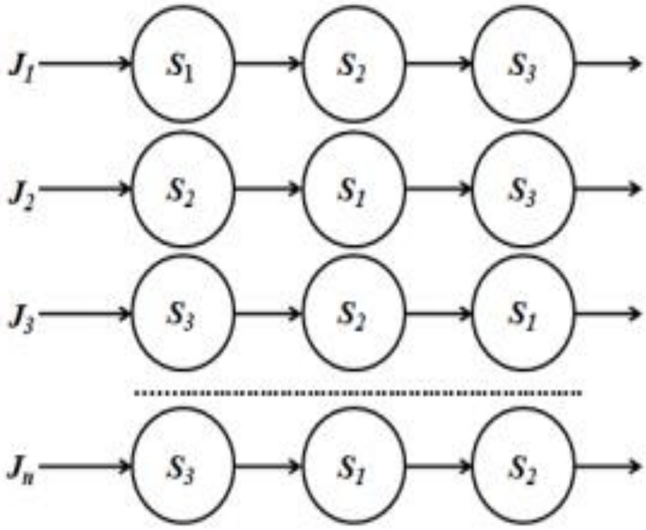

Figure 7: Flexibility in Job Shop Scheduling

Algorithm 1: MCCCSA

1. Begin
2. Random job order array j is generated. Let the process count = 1 and process = 1
3. Let population $p0$ = $n0$ organizations, and every organization with only one member;
4. t=0;
5. While (termination condition not met) do
6. Begin
7. Calculate make-span using Equation (4)
8. Calculate the energy consumption by using the Equation (5)
9. Calculate the lateness value using Equation (6)
10. For job i ϵ j do
11. Discover all the machines M that can execute Oj,i
12. For every job in pt, when the number is greater than 20, perform the dividing operator on it, delete it from pt, and add the child organization to pt+1;
13. While (the number of jobs in pt > 1) do
14. Begin

15. For machines k in M do
16. Select two parent organizations in random from pt as $orgp1$ and $orgp2$
17. Execute the CS and select the leaders;
18. If rand is less than 0.5
19. Perform annexing operator;
20. Else
21. Perform cooperating operator;
22. If $f(w)$ less than $f(x)$ and $f(z)$ less than $f(y)$
23. x=w;
24. y=z;
25. If t1 ≤ t0 then
    - Oj,I is added in Mk starting from t0 and make Ti,k = t0 + tj,i,k
    - Else if it is in between t0 & t1 (time interval between two successive operations) ≥ tj,i,k then
    - Oj,i is added in Mk starting from the end of finished operating time of process to left and make Ti,k = this time + tj,i,k
    - Else
    - Oj,i is added in Mk starting from t1 and Ti,k = t1 + tj,i,k
    - End if
26. End
27. the child organizations are added into pt+1;
28. calculate R.E.I by Equation (10)
29. End
30. $orgp1$and $orgp2$ are deleted from pt;
31. End
32. u organizations are deleted from pt+1;
33. % u = child number of join organizations
34. $pt \leftarrow p(t+1)$;
35. $t \leftarrow t+1$;
36. End
37. identify the minimum stopping time on T and assign Oj,i to that machine.
38. Select best make-span, lateness and energy consumed machine
39. output the best solution in pt
40. End

The overall block diagram of the proposed system is shown in figure 8

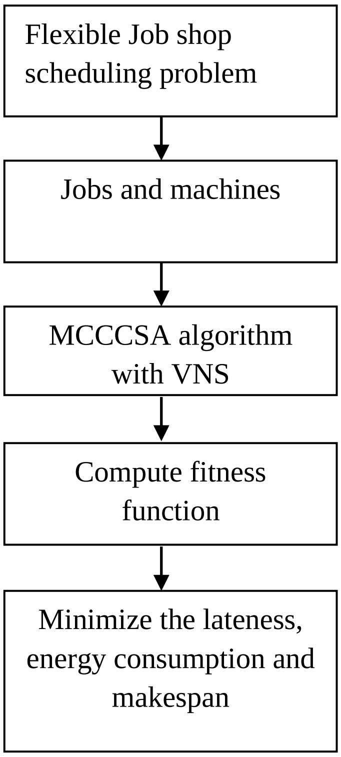

Figure 8. Overall Block Diagram of the Propsoed System

4. Experimental Result

The proposed method MCCCSA with VNS used shop scheduling problem in a flexible job shop with setup times and the performance is analyzed in terms of execution time and job lateness

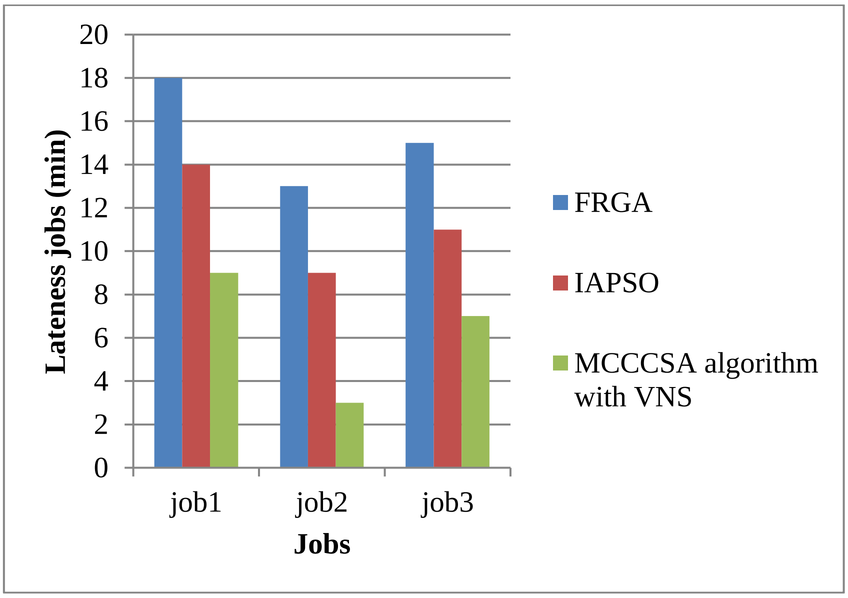

Figure 9. Comparison of the Lateness of Jobs

Fig 9 depicts the comparison metrics evaluated for existing and proposed method in terms of lateness. The x-axis indicates the jobs taken and the y-axis corresponding lateness value. From the graph, the existing methods, FRGA and IAPSO have higher lateness value when compared with the proposed MCCCSA+VNS algorithm. Thus the analysis indicates that MCCCSA with VNS algorithm enhances the performance of FJSSP.

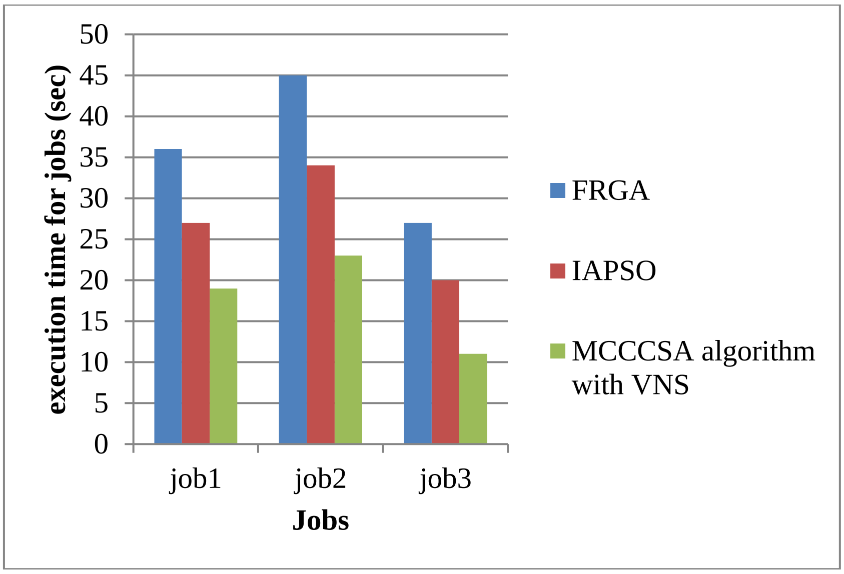

Figure 10. Execution Time for Jobs

Figure 10 illustrates the execution time comparison metric of both existing and proposed method. Here x-axis is plotted for jobs taken and y-axis for the execution time for the jobs. It is observed that, the existing methods FRGA and IAPSO takes longer execution time when compared to the proposed MCCCSA + VNS algorithm. This indicates that the proposed MCCCSA with VNS algorithm improves the performance of FJSSP.

5. Conclusion

In the traditional job shop design, the machines are combined together based on their functions. The scheduling problems are complicated in the case of shop floor. The problems are transportation delay, machine breakdown, material shortage and so on. To solve the problem, many assumptions are made like known jobs, resources are made available all time, known processing time and always remain constant, ignoring the transportation time and so on. This made Job Shop Scheduling Problem (JSSP) to be a research area for decades. In this work, a Modified Cooperative Co-evolutionary Cuckoo Search Algorithm (MCCCSA) along with Variable Neighborhood Search (VNS) is proposed for addressing constrained ad unconstrained optimization problem. The motive of the work is to model FJSSP and identify feasible solution using MCCCSA algorithm. Experimental analysis indicates that the proposed MCCCSA + VNS algorithm reduces the lateness, energy consumption and execution time of the job when compared with the existing methods. Future research will focus on multiple routing flexible job shop environments that have routing flexibility for each operation and the operations of the job are ordered with precedence-constrain. The research will also develop hybrid algorithms by integrating local search mechanisms namely tabu search, simulated annealing, bottleneck shifting procedure, so on to the proposed heuristics to improve the performance.

References

1. Mokhtari, H., & Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Computers & Chemical Engineering*, *104*, 339-352.
2. Sundar, S., Suganthan, P. N., Jin, C. T., Xiang, C. T., & Soon, C. C. (2017). A hybrid artificial bee colony algorithm for the job-shop scheduling problem with no-wait constraint. *Soft Computing*, *21*(5), 1193-1202.
3. Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Computers & Industrial Engineering*, *85*, 376-383.
4. Wang, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, *39*(10), 2291-2299.
5. Kaplanoğlu, V. (2016). An object-oriented approach for multiobjective flexible job-shop scheduling problem. Journal of Expert Systems With Applications, 10322, (56), 1-36
6. Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., & Mahmoodian, V. (2015). An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering*, *86*, 2-13.
7. Kundakcı, N., & Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *Computers & Industrial Engineering*, *96*, 31-51.

8. González-Neira, E. M., García-Cáceres, R. G., Caballero-Villalobos, J. P., Molina-Sánchez, L. P., & Montoya-Torres, J. R. (2016). Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria. *Computers & Industrial Engineering*, *101*, 128-144.
9. Wu, J., Wu, H., Song, Y., Cheng, Y., Zhao, W., & Wang, Y. (2016). Genetic algorithm trajectory plan optimization for EAMA: EAST Articulated Maintenance Arm. *Fusion Engineering and Design*, *109*, 700-706.
10. Ahmadi, E., Zandieh, M., Farrokh, M., & Emami, S. M. (2016). A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. *Computers & Operations Research*, *73*, 56-66.
11. Gu, W., Tang, D., & Zheng, K. (2012). Minimizing makespan in job-shop scheduling problem using an improved adaptive particle swarm optimization algorithm. In *2012 24th Chinese Control and Decision Conference (CCDC)*, pp. 3189-3193.
12. Akbari, M., & Rashidi, H. (2016). A multi-objectives scheduling algorithm based on cuckoo optimization for task allocation problem at compile time in heterogeneous systems. *Expert Systems with Applications*, *60*, 234-248.
13. Dai, M., Tang, D., Giret, A., & Salido, M. A. (2019). Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robotics and Computer-Integrated Manufacturing*, *59*, 143-157.
14. Huang, L., Ding, S., Yu, S., Wang, J., & Lu, K. (2016). Chaos-enhanced Cuckoo search optimization algorithms for global optimization. *Applied Mathematical Modelling*, *40*(5-6), 3860-3875.
15. Kamoona, A. M., Patra, J. C., & Stojcevski, A. (2018). An enhanced cuckoo search algorithm for solving optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-6.
16. Adibi, M. A., & Shahrabi, J. (2014). A clustering-based modified variable neighborhood search algorithm for a dynamic job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, *70*(9-12), 1955-1961.
17. Zhang, G., Zhang, L., Song, X., Wang, Y., & Zhou, C. (2019). A variable neighborhood search based genetic algorithm for flexible job shop scheduling problem. *Cluster Computing*, *22*(5), 11561-11572.