ISSN:0975-9646

A Hybrid Approach for Job Shop Scheduling Problems

Dr.R. Ramkumar
Department of Computer Applications
Nandha Arts and Science College, Erode, Tamil Nadu

Abstract **- Job shop scheduling problems are NP-hard problems. This paper presents a hybrid algorithm for solving Job-shop Scheduling Problem. Ant Colony Optimization is a meta-heuristic inspired by the searching behavior of ants, which is also used to solve this combinatorial optimization problem. The settings of parameter values have more control to solving an instance of the job shop problem. In this algorithm considers an alternative solution in which each machine is assign one of some fuzzy rules, which heuristically determines the processing order for that machine. The sequence of jobs is scheduled using Fuzzy logic and optimized using Ant Colony Optimization. The makespan, completion time, makespan efficiency, algorithmic efficiency are calculated. The performance of this approach compared by analyzing the JSSP benchmark instances.**

Keywords**- Fuzzy Logic, Scheduling, Makespan, Ant Colony Optimization, Hybrid**

I. INTRODUCTION

Scheduling is the process of generating the schedule and schedule is a physical document and tells the experience of things and shows a plan for the timing of certain activities. The scheduling problem can approach in two steps; in the first step sequence is expected or decides how to choose the next task. In the second stage, planning of start time and perhaps the completion time of each task is performed.

Primary JSSP is a static optimization problem since all information about the production program known in advance. The General job-shop problem is probably most studied one by academic research during the last three decades and is a notoriously difficult problem to solve. The JSSP is an NP (Nondeterministic Polynomial) hard problem and among those optimization problems, it is one of the least tractable known (Garey et al 1979) problem .It is purely deterministic, since processing time and constraints fixed, no questionable actions occur. The Job-shop scheduling problem also illustrates some of the demands required by a wide array of real-world problems. In a shop floor, machine process jobs and each job contain a certain number of operations. Each operation has its individual processing time and has to process on a dedicated machine (ElMaraghy et al 2000). Each job has its machine order then no relation exists between machine orders of any two jobs.

Operations are processed in one machine from an operation sequence for this machine. For a given problem, an operation sequence for each machine is called a schedule (K.Mertins et al 1979). Since each operating sequence can be permuted, and independent of operation sequences of other machines. The problem with n jobs and m machines can have a maximum of m different solutions and the completion time of all jobs as makespan. The objective is to find a reasonable schedule with minimum makespan. Feasible plans obtained by permuting the processing order of operations on machines without violating the technological constraints.

Fuzzy Logic (FL) is essentially a problem-solving control system methodology that implemented in systems ranging from the very basic and small, embedded microcontrollers (Christer Carlsson et al 1996) to large, networked, data acquisition control systems. It implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to controlling problems mimics how a person would make decisions, only much faster.

Ant Colony Optimization (ACO) is an evolutionary metaheuristic to solve combinatorial optimization problems by using principles of communicative behavior found in original ant colonies. Recently the ACO approach has been applied to scheduling problems, as a bus driver scheduling, Job-Shop, Flow-Shop. Many authors have compared an ACO algorithm with several other heuristics to solve the problem (e.g. decomposition heuristics, interchange heuristics, and simulated annealing). They have shown that the ACO algorithm found the optimal solution of several benchmark problems more often than the other heuristics.

Bessem Kordoghli et al., (2010) identified a new scheduling approach of the cloth manufacturing company problem. This method based on the best order scheduling. Wanlei Wang et al., (2008) proposed to solve the problem of job shop scheduling in multi-product different manufacturing workshops. They used fuzzy set theory and create job shop scheduling fuzzy mathematical model, in which the objective function is the maximum customers' satisfaction degree of the delivery date. Surekha. P et al., (2010) proposed an ant colony optimization algorithm for solving the Job-shop Scheduling Problem (JSSP). Ant Colony Optimization (ACO) is a metaheuristic inspired by the foraging behavior of ants, which is also used to solve this combinatorial optimization problem. Bud Fox et al., (2007) proposed ant colony optimization algorithm is a fast suboptimal meta-heuristic based on the behavior of a set of ants that communicate through the deposit of pheromone. It involves a node choice probability that is a function of pheromone strength and inter-node distance to construct a path from a node-arc graph.

The organization of the paper is as follows: Section 2 gives a brief introduction to a job-shop scheduling problem, and Section 3 describes Job-shop scheduling using fuzzy logic. Section 4 explains the hybrid algorithm for JSP. The experimental results analyzed in section 5 and section 6 concludes this paper.

II. Job-Shop Scheduling Problem

The Job shop scheduling problem consists set of jobs J = {1 . . n}, a set of machines M = {1 . . m}, where Ji denotes ith job ($1 \leq i \leq n$) and Mj denotes jth machine ($1 \leq j \leq m$). On the machines M1, M2, … Mm, the jobs J1, J2 … Jn is to be scheduled. Let V is the set of all operations in all jobs. Each job Ji has a set of operations oi1, oi2, …oik, where k is the total number of operations in the job Ji. Operation's precedence constraints are associated with each job and ensure that operation oij will be processed only after the processing of operation oij-1 in a particular job i.

The standard model of n jobs, m machines job shop, is denoted by n/m/φ/P/Cmax. The parameter φ is a technological matrix denoting the processing order of machines for different jobs (Chaabene et al., 2007). The machining order for ith job is given by φij ($1 \leq j \leq m$), where j denotes jth operation in ith job. An example of the technological matrix φ can be represented as Equation (2.1)

$$\varphi = \begin{pmatrix} M2 & M3 & M1 \\ M1 & M2 & M3 \\ M3 & M1 & M2 \end{pmatrix} \qquad (2.1)$$

Each row of the above matrix represents a job. For the first job, the first operation is performed on machine M2, a second operation carried out on machine M3 and the third operation performed on machine M1. Similarly, other jobs are executed on different machines. Matrix P, denoting the processing time of different operations, is represented Equation (2.2): Where pij represents a time in the jth process of ith job.

The technological matrix φ and processing time matrix P given as problem data. The processing order (machine sequence) for machine Mi given by Πik ($1 \leq k \leq n$), where k denotes kth operation to processed on machine Mi.

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \qquad (2.2)$$

A solution to JSSP can represent by a matrix Π indicate processing orders of all machines. A solution of the above problem considered as Equation (2.3)

$$\Pi = \begin{array}{c} M1 \\ M2 \\ M3 \end{array} \begin{pmatrix} J2 & J3 & J1 \\ J1 & J2 & J3 \\ J3 & J1 & J2 \end{pmatrix} \qquad (2.3)$$

According to the above schedule, the first operation of the second job is planned on machine M1, followed by a second operation of the third job and third operation of the first job. Similarly, other machines have schedules represented in second and third rows. Subscript values denoting machine numbers in φ and job numbers in Π are given to formulating a technological matrix and matrix representing a solution respectively. A processing unit of jth operation of ith job on a machine is denoted as oij. Each operation o has at most two direct predecessor operations, a job predecessor PJo and a machine predecessor PMo. The first operation of a machining sequence has no PMo, and the first operation of a job has no PJo. Similarly, each operation has at most two direct successor operations, a job successor SJo and a machine successor SMo. The last operation of a machining sequence has no SMo and the last operation of a job has no SJo. An operation o is called schedulable, if both, PJo and PMo are already scheduled. Let $r_{o_{ij}}$ be the starting time of jth operation of ith job. The Completion time Coij for oij is calculated as in Equation (2.4)

$$C_{o_{ij}} = r_{o_{ij}} + P_{ij}, \; r_{o_{ij}} = \max(C_{PJ_{o_{ij}}}, C_{PM_{o_{ij}}}) \qquad (2.4)$$

$r_{o_{ij}}$ are assigned by zero values for undefined $PJ_{o_{ij}}$ and $PM_{o_{ij}}$. After the scheduling of all operations, the makespan Cmax representing a completion time of all operations is calculated as in Equation (2.5).

$$\text{Cmax} = \max(C_{o_{ij}}) \text{ for all oij} \in V \qquad (2.5)$$

III. Job-Shop Scheduling Using Fuzzy Logic

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise (Colleen Emma Atves 2010). In contrast with "crisp logic", where binary sets have binary logic, the fuzzy logic variables may have a membership value of not only 0 or 1that is, the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values of classic propositional logic.

Fuzzy Logic provides a simple way to arrive at a controlled output based upon vague or noisy input parameters. The output control is a smooth control function despite a broad range of input variations. Fuzzy Logic incorporates a simple, inference type rule-based approach to solving a control problem. Fuzzy Logic conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can build into anything from small, handheld products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data.

A. *Analysis of Input Parameter & Generate Membership Function, Rule Set*

Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, instead of numerical values. A linguistic variable decomposed into a set of linguistic terms. Consider a let Customer Priority (CP), Due Date (DD) & Processing Time (PT) is the linguistic variable that represents the value of the system. To qualify the CP, DD & PT terms are used many values (L, H, VH, VL) in real life. These are the linguistic values of the system. Each member of this decomposition is called a linguistic term and can cover a portion of the overall values of the system.

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs processed, define the functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions inferred, scaled, and combined, they are defuzzified into a crisp output that drives the system. There are different membership functions associated with each input and output response.

In the system, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion. The rule base consists of the database and the linguistic control rule base. The database provides the information that is used to define the linguistic control rules and the fuzzy data manipulation in the fuzzy logic controller. The rule base defines (expert rules) the control goal actions using a set of linguistic rules. In other words, the rule base contains rules such as would be provided by an expert. The FSC looks at the input signals and by using the expert rules determines the appropriate output signals (control actions). The rule base contains a set of if–then rules.

1. IF customer priority is very high AND due date is distant AND processing time is short THEN sequence is very high.
2. IFcustomer priority is low AND due date is distant AND processing time is long THEN sequence is reject.
3. IF customer priority is medium AND due date is close AND processing time is medium THEN Sequence is medium.

B. *Sequence Controller*

The sequence controller (reasoning mechanism) is the kernel of FSC. The capability of simulating the human decision making based on fuzzy concepts and inferring fuzzy control actions by using fuzzy implications, fuzzy logic rules of inference. In other words, once all the monitored input variables are transformed into their respective linguistic variables, the inference engine evaluates the set of if-then rules (given in the rule base), and thus the result is obtained which is again a linguistic value for the linguistic variable.

Once the aggregated fuzzy set representing, the fuzzy output variable determined, an actual crisp control decision must made. The process of decoding the output to produce an actual value for the control signal referred to as the defuzzification. The defuzzification of the data into a crisp output is accomplished by combining the results of the inference process and then computing the techniques of the area. The weighted strengths of each output member function multiplied by their respective output membership function. Finally, this area is divided by the sum of the weighted member function strengths and the result taken as the crisp output. Without defuzzification, the final output of the inference stage would remain a fuzzy set.

IV. Hybrid Algorithm For Job-Shop Scheduling

Hybrid Algorithm represented for proposed ant colony optimization method and employs different fuzzy priority rules, tuning of parameters, evaporation level, variation parameter and step pheromone updating strategies described in subsequent sections. The algorithm uses a procedure Chose_Node( ), which is used to select a node during the construction of solutions. The algorithm gets values for parameters α, β, ρ, q0, Q and a problem instance. These parameters can be tuned to produce better results. The algorithm finds an initial solution S* and corresponding makespan value f(S*), and assigned to Cmax keeping the best value up to the current iteration. Initial pheromone value τ0 calculated. The Hybrid algorithm is given below.

```
Procedure Chose_Node( )
Begin
    Assign 0 to b,
    Assign 1 to q
    While ($q \le n$)
     Let $a[q] = \tau(\text{LastVisit}, Z[q, j(q)])\ \alpha\ .\ (\eta(j(q)))\ \beta$
    Let $b = b + a[q]$
    EndWhile
    Assign 1 to q
    While ($q \le n$)
Choose a node $Z[v, j[v]]$ from T with Monte Carlo Probability using $P[q, j(q)]$ for $q = 1 \ldots n$
 Let $\lambda(v) = \lambda(v) - P(v, j(v))$
  Add $Z[v, j(v)]$ to S
 Delete the node $Z[v, j(v)]$ from T
If $Z[v, j(v)+1]$ is successor of $Z[v, j(v)]$ Then
Increment $j[v]$ by 1
Add the node $Z[v, j(v)]$ to T
  End If
End
```

Procedure Chose_Node( ) uses state transition rule to find the probability to choose a node j from a decision point i and adds j to S, which contains a partial solution during the construction of solutions. The procedure also deletes j from T, and a new node z (I, j+1) adds to the set T. The node z(i, j+1) is produced by the conjunctive arc of the graph Ģ. Likewise, all nodes in the set T are being selected and added to the set S.

If there cannot be a node produced by the conjunctive arc of the particular job (for z(i, j), z(i, j+1)= null). The new node not added to the set T. Hence, at this time the set T has allowed nodes corresponding to the n-1 number of jobs. Generally, after completion of r number of jobs in a job shop, T has an n-r number of nodes. Finally, the set S has all nodes visited, and set T would be empty. The set of nodes in S could form a path from the start node to end node.

V. Experimental Results Analyzed

The proposed algorithm tested with standard benchmark JSSP taken from the OR-Library. In the proposed algorithm, the problem representation is simple and compared with other heuristic methods. Also, this algorithm reduces the computational complexity. The experiment conducted for the population of size 100, and the optimal solution (Best Known Solution) in the search space obtained with a minimum number of iterations. The following table1 shows number of iterations required by hybrid algorithms to reach UB value for different classes of problem instances

TABLE I
NUMBER OF ITERATIONS REQUIRED BY HYBRID ALGORITHMS

Problem Instance	Opt (LB UB)	UB	No.of Iterations
LA01	666	666	1894
LA02	655	655	9732
LA03	597	597	19836
LA04	590	590	26764
LA05	593	593	76
LA06	926	926	2925
LA07	890	890	3972
LA08	863	863	6414
LA09	951	951	504
LA10	958	958	523
LA11	1222	1222	4718
LA12	1039	1039	4349
LA13	1150	1150	5196
LA14	1292	1292	796
LA15	1207	1207	5726
LA16	945	969	79917
LA17	784	786	40091
LA18	848	856	48084
LA19	842	859	68290
LA20	902	912	57719

The following table II shows number comparisons of CPU time among hybrid algorithm for the problem instances. Column 1 provides problem instances used for testing whereas the number of jobs and number of machines is specified in column 2 and column 3 respectively. In column 4, a total number of operations for each problem is given. The time required to reach UB for hybrid algorithm specified in column 5, and corresponding UB values are given in parenthesis.

Time required to reach optimal (or upper bound) value is directly proportional to the number of operations in the problem instance. The complexity of the problem also affects the time to reach optimal (or upper bound) value. A number of machines available also increase the complexity of the problem. For example, problems LA11 and LA16 have the same number of operations.

TABLE III
COMPARISONS OF CPU TIME AMONG HYBRID ALGORITHM

Problem Instance	n	M	Total Operations	CPU Sec (UB)
LA01	10	5	50	32(666)
LA05	10	5	50	2(593)
LA06	15	5	75	65(926)
LA10	15	5	75	11(958)
LA11	20	5	100	148(1222)
LA14	20	5	100	26(1292)
LA16	10	10	100	1598(969)
LA20	10	10	100	1154(912)
LA21	15	10	150	1857(1072)
LA26	20	10	200	1298(1221)
LA31	30	10	300	3827(1784)
LA36	15	15	225	2438(1298)

But the time required to reach the optimal value for LA16 is more than that of LA11 because LA16 has more machines than LA11. An increasing number of jobs greatly increases the time required per iteration for the algorithm because the number of time, in which state transition rule invoked for the selection of operation during the construction of a solution, is equivalent to the number of jobs. For example, if 10 × 5 is the size of a problem instance, the state transition rule is used for 500 (10 × 10 × 5) time in an iteration to build a solution.

VI. Conclusion

This paper presents the hybrid approach to solve the job shop scheduling problem. To make the problem of flow shop scheduling more applicable, we considered it in a fuzzy state. For this purpose, we examined the process times of jobs on machines in the form of fuzzy numbers. In this problem, our goal was to find an optimum sequence that we could be able to minimize the makespan. The goal of the work is to know the effect of different parameter setting that seem to play a significant role in its performance and the quality of the solution. Once the parameters that properly tuned, the algorithm converges satisfactory, thus accomplishing the stated goal of the work. The performance of the algorithm compared with their pure parents and existing approaches. The algorithm produces the best results among others.

REFERENCES

[1] Bessem Kordoghli, Seifeddine Saadallah, Mohamed Jmali and Noureddine Liouene, "Scheduling optimization in a cloth manufacturing factory using genetic algorithm with fuzzy logic for multi objective decision", Journal of Textile and Apparel Technology and Management, Vol.6, Issue 3, pp.1-12, 2010.

[2] Bud Fox, Wei Xiang and Heow Pueh Lee ,"Industrial applications of the ant colony optimization algorithm", The International Journal of Advanced Manufacturing Technology, Vol. 31, Issue 7-8, pp 805-814, 2007.

[3] Chaabene, M., Ammar.M.B and Elhajjaji.A., "Fuzzy based energy management of a domestic photovoltaic panel", Am. J. Applied Sci., 4: 60-65,DOI: 10.3844/ajassp.2007.60.65, 2007.

[4] Christer Carlsson and Robert Fuller, "Fuzzy multiple criteria decision making: Recent developments", International Journal of Fuzzy Sets and Systems, Vol.78, pp.139-153, 1996, Doi: 10.1016/0165-0114(95)00165-4.

[5] Colleen Emma Atves, B.S.C.E. "A Fuzzy Logic Analysis of Sustainable Concrete Pile Foundations", Graduate Program in Civil Engineering", The Ohio State University,2010.

[6] ElMaraghy, H., V. Patel and I.B. Abdallah.,"Scheduling of manufacturing systems under dualresource constraints using genetic algorithms", J.Manufact. Syst., 19: 186-201, DOI: 10.1016/S0278-6125(00)80011-4, 2000.

[7] Garey, Michael, R. and Johnson David, S. "Computers and intractability: a guide to the theory of NP-completeness", Oxford, Freeman,New York, ISBN: 0-7167-1044-7, 1979.

[8] Mertins.K, Albrecht.R, Wegener.U, Duttenhofer.F., "Set-up scheduling by fuzzy logic", Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and AutomationTechnology, 1994., pp.345-350, Oct, 1994.

[9] Surekha, P. Mohanaraajan, P.R.A.and Sumathi. S., "Ant colony optimization for solving combinatorial fuzzy Job Shop Scheduling Problems", International Conference on Communication and Computational Intelligence (INCOCCI),pp. 295-300,2010.

[10]Wanlei Wang, Changfeng Yuan and Xiaobing Liu, "A Fuzzy Approach to Multi-product Mixed Production Job Shop Scheduling Algorithm", IEEE computer society, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008.DOI - 10.1109/FSKD.2008.420.