Proc. of the Intl. Conf on Advances in Aeronautical and Mechanical Engineering - AME 2012
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-981-07-2683-6

FLEXIBLE MANUFACTURING SYSTEM SCHEDULING: TABU SEARCH AND ANT COLONY OPTIMIZATION APPROACH

Pradeep Kumar Krishnan
Department of Engineering
Nilai University College
Nilai, Malaysia

Sundar Raj Senthil Kumar
Department of Computing
Nilai University College
Nilai, Malaysia

***Abstract*— Manufacturing is embodied with several key functions and the scheduling activity has become an essential contributor to a successful manufacturing system. Scheduling allocates the time when a particular task or activity is to be processed by a given resource in order to optimize the requirements.**

***Keywords*— *Fl*exible manufacturing systems, tabu search, ant colony optimization, scheduling**

I. Introduction

Scheduling is the process of assigning activities to resources in time. It involves finding a time allocation of competing resources, such as machines, personnel, tools and components, in order to produce a particular commodity and it is considered to be a major task in the manufacturing environment[1]. It is essentially concerned with solving a constraint optimization problem. In simple terms, scheduling attempts to optimize a particular objective function like minimizing the makespan, minimizing tardiness, minimizing the idle time, etc., subject to certain constraint. Scheduling problem is defined as determining the sequence of operations that minimizes the makespan or total completion time..

II. Problem Formulation

The complexity of the scheduling problem.

In this section, the problem is described. In the problem, a job $j \in J$ (where J is a set of jobs, such that $J = \{1, 2, . . . , j, . . . , j'\}$, j' being the maximum number of jobs available from time zero onwards) is to be assigned to a machine $m \in M$ (where M is the set of machines such that $M = \{1, 2, . . . ,m, . . . , m'\}$, m' being the number of machines), such that the makespan is minimum. The job constitutes a sequence of operations and an operation $o_j \in O_j$ (where $O_j = \{o_1, o_2, . . . , o_j, . . . , o'_j\}$, o'_j being the maximum number of operations for job j) can have some flexibility depending on the feasibility of an operation to be performed on a machine. For a feasible operation o of job j on machine m, the processing time can be denoted by $p_{j,o,m.}$ Here, it should be noted that $O_j \in O'$ where O_j gives the set of operations for job j and O' *is* the set of all operations on all jobs. Thus, the objective of the problem is to schedule operations of all the jobs on machines such that the overall makespan is minimized. Keeping this objective in mind, the next section proposes an ACO-based scheduling approach.

III. Problem solving methodology

To solve the combinatorial problem different types of techniques are used. ACO is a technique which is used for solving the hard combinatorial problem [2]. First it is used for solving the Traveling sales man problem and later it is used for solving the scheduling problems [3]. The algorithm is slightly modified according to the type of the application. By using the ACO algorithm the artificial ants are created and the ants construct a solution by visiting the all possible nodes. By visiting the nodes, the ants can find the possible schedules by which the job can be processed [4]. For each schedule, different makespan values are found. By this way the optimum makespan value and the schedule is found.

The combinatorial problem can also be solved by utilizing the Tabu search method. In this step is to create the initial solution. In this step a schedule is created and the makespan value is found. And the neighborhood is identified and is swapped. By changing the neighborhood structure the new schedule is less than the existing makespan value, and then the new schedule is the best schedule. By this way all the schedule is generated and the best result is found.

A. Graph based representation

For a given set of jobs J and set of machines M with corresponding operations of job j as oj, a directed graph is constructed. Let $G = (O', A)$ be the set of arcs connecting all possible combination of nodes [5]. The initial node i is necessary in order to specify the first scheduled job when several jobs have their first operations on the same machine.

Considering N as the total number of operations, $N + 1$ nodes and $N (N .1) /2 + J$ arcs can be obtained, where all nodes are pairwise connected except node I, in Fig. 1 which is connected only to the first operation of each job. Thus, each node corresponds to an operation of a particular job on a particular machine. Each arc is weighted by a pair of numbers $\{\tau_{kl}, \eta_{kl}\}$, where τ_{kl} is the trail level and η_{kl} is the visibility, computed initially according to a desirability measure derived from a greedy problem- specific heuristic like LPT (longest processing time) or LRPT (largest remaining processing time).

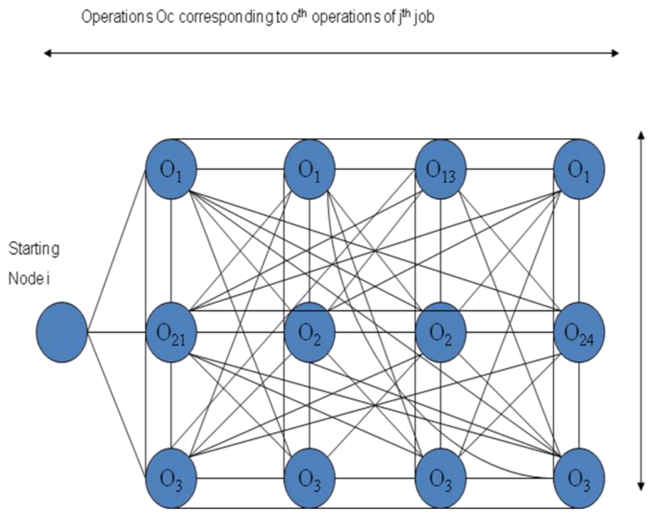


Fig. 1 Graph-based representation of the FMS scheduling

B. *Ant colony optimization* (ACO)

Ant colony optimization (ACO) is a recently proposed metheuristic approach and Model Based Search algorithm for solving hard combinatorial optimization problems [5]. The inspiring source of ACO is the pheromone trail laying and following behavior of real ants which use pheromones as a communication medium. In analogy to the biological example, ACO is based on the indirect communication of a colony of simple agents, called ants, mediated by pheromone trails.

The algorithm imposes a definition the optimization-problem into a graph, in which the ants move along every branch from one node to another node and so construct paths representing solutions. It iteratively moves from one node to another according to the State Transition Rule which is a probability function, which considers the visibility, and amount of pheromone on the edges.

The State Transition Rule is,

$$P_{ij}(t) = \frac{(\tau_{ij}(t))^{\alpha} * (\eta_{ij}(t))^{\beta}}{\sum (\tau_{ij}(t))^{\alpha} * (\eta_{ij}(t))^{\beta}}$$

Where τ is the quantity of the pheromone on the edge between node i and bode j .η is the heuristic distance between node i and node j. α and β tune the relative importance in probability of the amount of the pheromone Vs the heuristic distance.

C. *Tabu search* (TS)

TS is a simple deterministic oriented search procedure that constrains searching and seeks to transcend local optimality by storing the search history in its memory [6]. It forbids moves in the neighborhood having certain attributes, with the aim of guiding the search process away from solutions that appear to duplicate or resemble previously achieved solutions.

The most commonly used tabus involve recording the last few transformations. The attribute that represents an element in sequencing is two operations that have interchanged in recent move. It is the arc (j, i) which is obtained by swapping a pair of operations (i, j). To identify tabu status of the neighbour (j, i), it is looked whether or not it is on the tabu list; if so; it is labeled as “tabu” [7]. Tabu list is updated after each move in so far as the strategic forgetting occurs.

A general TS algorithm is;

Step 1: Start the initial solution; sore it as current seed and best solutions.

Step 2: Generate neighbors of the current seed solutions by a neighborhood structure. Select a neighbor which is not tabu or satisfies a given aspiration criterion and move it as new seed solution.

Step 3: Repeat step 2 until a termination criterion is satisfied.

IV. Solution procedures

The FMS scheduling problem is solved by ACO and the Tabu Search method

A. *Ant colony optimization algorithm*

In ACO algorithm if the parameters (a, β, ρ) are not well tuned, then the ants construct the same solution over and over again, making further exploration of newer paths almost impossible. It is known as stagnation. This derives from excessive trail levels on the edges of one solution, and can be observed in advanced phases of the search process. In particular, stagnation derives from wrong value of parameter p in the original algorithm. If it is too high, stagnation might take place, and if it is too low, little information is conveyed from previous solutions and the algorithm becomes a randomized greedy search procedure. Thus p should be chosen carefully. The algorithm for the ACO is explained below.

Step 1:[Initialization]
Set t =0; NC = 0; {t is the time counter, NC is the number of algorithm cycles}
For each (i, j), set an initial value$\tau_{i,j}$ = c and $\Delta\tau_{i,j}$ =0
{$\tau_{i,j}$ (t) is the intensity of trail on the edge (i, j) at time }
{$\Delta\tau_{i,j}$ is the quantity of trail laid on edge (i, j) by the k^{th} ant }

Proc. of the Intl. Conf on Advances in Aeronautical and Mechanical Engineering - AME 2012
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-981-07-2683-6

Step 2: [Starting node]
For each ant k: Place ant k on randomly chosen node and store this information is tabu $_k$
Step 3: [Build a tour for each ant]
For each node i: For each ant k: Choose the node j to move to, with the probability $P_{ij}^{k}(t)$ given by the formula. Store this information in tabu $_k$
Step 4: [Update the intensity of trail]
For each ant k: Compute the tour length L_k Compute the quantities $\Delta\tau_{ij}^{k}$ laid on each edge (i, j) according to Q/L_k, For each edge (i , j): Compute the intensity of trail according to equation
$\tau_{ij}(t+1) = (1-\rho).\ \tau_{ij}(t) + \Delta\tau_{ij}^{k}$
$T = t+1$; $NC = NC+1$; $\Delta\tau_{ij}^{k} = 0$;
Step 5: [Termination condition]
Memories the shortest tour found to this point If ($NC< NC_{max}$) and (Not stagnation behavior) Then empty all tabu lists and go to step #2 else stop

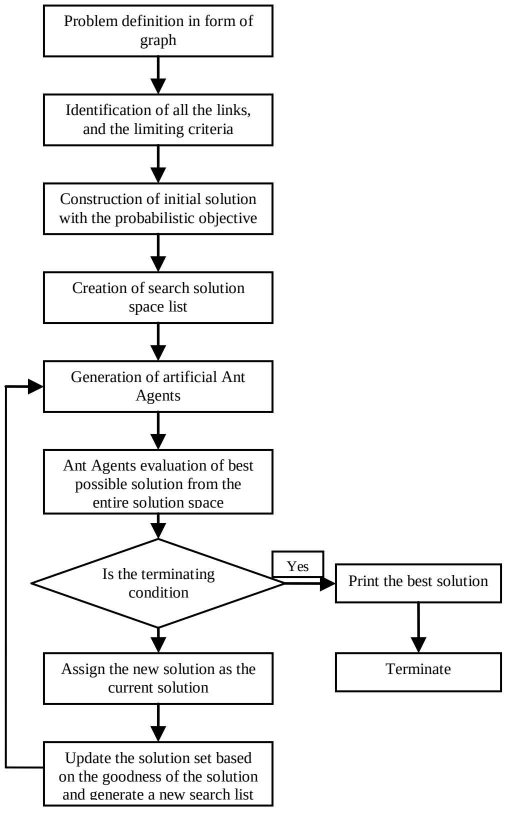

Problem definition in form of graph
Identification of all the links, and the limiting criteria
Construction of initial solution with the probabilistic objective
Creation of search solution space list
Generation of artificial Ant Agents
Ant Agents evaluation of best possible solution from the entire solution space
Is the terminating condition
Yes
Print the best solution
Terminate
Assign the new solution as the current solution
Update the solution set based on the goodness of the solution and generate a new search list

Fig. 2 Flow Chart for ACO algorithm

B. *Tabu Search algorithm*

Tabu Search relies on the systematic use of memory to guide the search process [7]. It uses a local search that at every step makes the best possible move from current solution to a neighbor solution even if the new solution is worse than the current one [8]. TS can explicitly memories recently visited solutions and forbid moving back to them. More commonly, TS forbids the effect of recently applied moves by declaring tabu those solution attributes that changes in the local search.

STEP 1 : Initialization

Set Cycle = 1// initialize the cycle counter

Set Tabu cycle = 0 // initialize the tabu list

STEP 2 : Generate the initial solution and store the makespan and job sequence in each machine as a tabu and also as a current job order.

STEP 3: If Cycle > maxcycle goto 8 otherwise goto step 4

STEP 4 : Neighbor

Finding the makespan for each neighbors and store it in a neighbor's list

STEP 5 : Move

1. Check whether best neighbor is in a tabu list, if not so, go to step 6 or go to step 5[2].
2. Check whether next neighbor is available in neighbor's list if so store next neighbor in the list as best and go to step 5[1] or go to step 5[3].
3. Find the neighbors with minimum cycle in to visit's list and deletes the neighbor in to visit list.

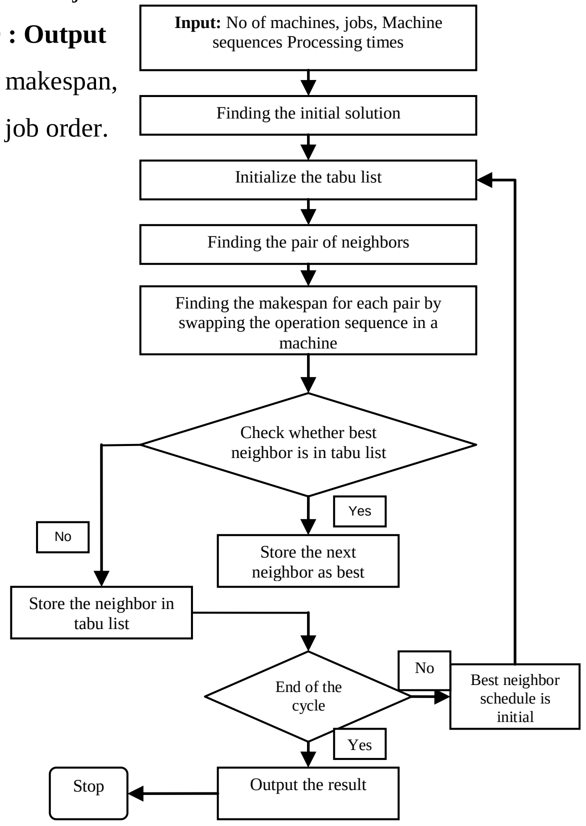
STEP 6: Store best neighbor, its makespan and its job order in the tabu list.

STEP 7: Cycle = cycle + 1 ; and go to step 3.

STEP 8: Finding the minimum makespan in tabu list, Store its makespan and job order as a best.

STEP 9 : Output

- Best makespan,
- Best job order.


Input: No of machines, jobs, Machine sequences Processing times
Finding the initial solution
Initialize the tabu list
Finding the pair of neighbors
Finding the makespan for each pair by swapping the operation sequence in a machine
Check whether best neighbor is in tabu list
Yes
No
Store the next neighbor as best
Store the neighbor in tabu list
End of the cycle
No
Best neighbor schedule is initial
Yes
Output the result
Stop

Proc. of the Intl. Conf on Advances in Aeronautical and Mechanical Engineering - AME 2012
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-981-07-2683-6

Fig. 3 Flow Chart for TS algorithm

V. Results and discussion

The following figure shows the Gantt chart for the problem LD. From this the makespan value is found as 390 seconds for the optimal path achieving by using the ACO method.

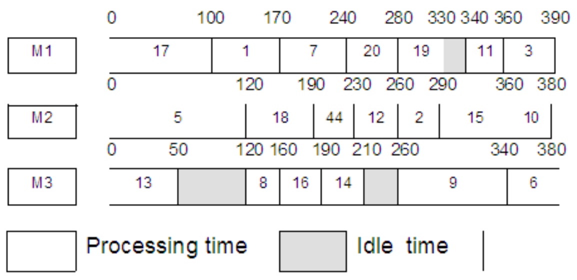


Fig. 4 Gantt chart for LD problem using ACO method

The following figure shows the disjunctive graph for LD problem using ACO method [9].

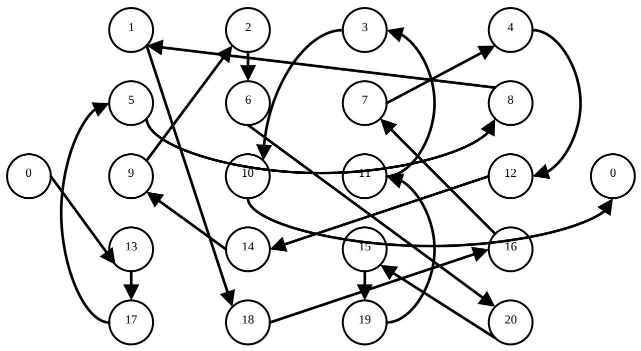


Fig. 5 Ant path

TABLE I. COMPUTATIONAL RESULTS

Test	Best Make span		
	Bench mark	*Obtained*	
		ACO	*TS*
LD	439	390	410
LDK	420	390	410
AK01	5950	5200	5400
AK02	N/A	3800	4700

In ACO algorithms, as the number of cycle increases, the makespan value obtained is near optimum linear value[10]. In Tabu Search algorithms, the variation of makespan value exists and converges at certain cycle.

V. Conclusion

Scheduling can be solved by many techniques like Genetic algorithm, Simulated annealing, Tabu Search, Shifting Bottleneck Procedure, etc., Ant Colony Optimization is being applied to combinatorial optimization problems for solving larger size problems. The optimum makespan is obtained using Ant Colony algorithm for FMS scheduling problem and compared with Tabu search algorithm. From the compared results it is found that the obtained makespan value is less. The parameters like α and β are fine tuned, so that the algorithm yielded the better results. The approach is a promising one because of its generality in nature and its effectiveness in finding very good solutions to the difficult problems.

Acknowledgment

This paper work was supported by Nilai Unversity College, monitored by Dato'Dr. Sothi Rachagan and Dr.R.Logeswaran A/L N.Rajasvaran

References

[1] F. Geyik and Ismail Hakki Cedimoglu (2004) 'The strategies and parameters of tabu search for job-shop scheduling', Journal of Intelligent Manufacturing, Vol. 15, pp. 439 -448.

[2] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, Marco Trubian (1994) Ant System for Job Shop Scheduling', Belgian Journal of Operation Research, Statistics and Computer Science.

[3] Alok R. Chaturvedi (1993), 'FMS Scheduling and Control: Learning to Achieve Multiple Goals' Expert Systems with Applications, Vol. 6, pp. 267- 286.

[4] Kumar.R, Tiwari.M.K, Shankar.R (2003) 'Scheduling of Flexible Manufacturing Systems: An Ant Colony Optimization approach' Proc. Instn Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture, pp. 1443-1453.

[5] Dorigo, Vittorio Maniezzo, Alberto Colorni (1996) 'The Ant System:Optimization by a colony of cooperating agents' IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol.26, No.1,pp.1-13.

[6] F. Geyik and Ismail Hakki Cedimoglu (2004) 'The strategies and parameters of tabu search for job-shop scheduling', Journal of Intelligent Manufacturing, Vol. 15, pp. 439 -448.

[7] Ponnambalam.S. G, Aravindan.P and Rajesh.S. V (2000) 'A Tabu Search Algorithm for Job Shop Scheduling', International Journal of Advanced Manufacturing Technology, Vol. 16, pp. 765-771

[8] Felix T.S. Chan (1999) 'Evaluations of operational control rules in scheduling a Flexible manufacturing system', Robotics and Computer- Integrated Manufacturing, Vol. 15, pp. 121-132.

[9] Mansour Abou Gamila, Saeid Motavalli (2003), 'A modeling technique for loading and scheduling problems in FMS' Robotics and Computer Integrated Manufacturing, Vol. 19, pp. 45-54.

ISBN: 978-981-07-2683-6

[10] E. Taillard (1993) 'Benchmarks for basic scheduling problems'European Journal of Operational Research, Vol. 64, pp. 278-285.