Ant Colony Optimization for Multilevel Assembly Job Shop Scheduling

M. Omkumar, Anna University, Chennai, India

P. Shahabudeen, Anna University, Chennai, India

Abstract

Job shop scheduling is one of the most explored areas in the last few decades. Although it is very commonly witnessed in real-life situations, very less investigation has been carried out in scheduling operations of multi-level jobs, which undergo serial, parallel, and assembly operations in an assembly job shop. In this work, some of the dispatch rules, which have best performances in scheduling multilevel jobs in dynamic assembly job shop, are tested in static assembly job shop environment. A new optimization heuristic based on Ant Colony Algorithm is proposed and its performance is compared with the dispatch rules.

Keywords

Scheduling, Assembly job shops, Ant colony algorithm, dispatching rules, optimization, multilevel jobs, makespan, simulation

Introduction

Over the past few years, a continually increasing number of research efforts have investigated scheduling jobs in job shop based environments under static and dynamic situation. The classic job-shop scheduling problem (JSP) is one of the most well-known machine scheduling problems which can be stated as follows: Given a number of ‘n’ jobs, the jobs have to be processed on ‘m’ machines. Each job consists of a sequence of ‘j’ tasks, i.e., each task of a job is assigned to a particular machine. The tasks have to be processed during an uninterrupted time period of a fixed length on a given machine. A schedule defines the time intervals in which the operations are processed and is feasible only if it complies with the following constraints: each machine can only process one operation at a time and the operation sequence is respected for every job.

Job shop scheduling problem mainly falls under two categories viz. schedule optimization problems to minimize makespan (The length of time required to complete all jobs), minimize tardiness or other cost based functions and development of dispatch rules to improve flowtime based and tardiness based measures of performance. The static job shop scheduling problem assumes that all the n jobs to be processed on m machines are available for processing at the beginning of the planning period i.e., at time t = t0 whereas dynamic scheduling problem allows for the possibility of new job arrival over time.

Assembly job shop is an extension of the Job shop, consisting of an assembly division. Job is the end product or end assembly of several sub-assemblies. These sub-assemblies in turn have sub-sub-assemblies and so on. These are called as multi level assembly jobs. In an assembly job shop, items undergo operations in a serial fashion as per the precedence constrains and wait for the arrival of its mating components at the assembly station, for the assembly operation to start. As the number of levels increases the complexity of scheduling also increases. This makes the assembly job shop scheduling problems (AJSP) quite challenging, when compared to the conventional job-shop scheduling. Scheduling in dynamic assembly job-shops mainly focus on development of dispatch rules for minimizing the flow time of the job as well as the staging delay. Very few reports have been published on development of dispatch rules for scheduling multilevel jobs in dynamic assembly job shop environment; moreover no remarkable work is available on optimizing the schedule of multilevel jobs in assembly job shop under static situation. This paper reports for first time, the comparison of performances of some of the best performing dispatch rules used in scheduling multilevel jobs in dynamic assembly job shop environment with that of a newly developed Ant colony optimisation (ACO) based heuristics. Experiments are conducted in scheduling multilevel jobs in static assembly job shop environment with the objective of minimizing the makespan.

Literature Survey

In the last two decades, many researchers have explored the performance of multi level assembly jobs. Goodwin and Goodwin (1982) have evaluated the performance of priority dispatching rules with the objective of generalizing job shop environment with that of assembly job shops. In this work, assembly jobs made up of sub-assemblies at more than one level were considered for the first time. Several observations concerning costs and the scheduling problems are made by Blackstone, Phillips and Hogg (1982). Russell and Taylor (1985) have evaluated and proposed many sequencing rules with the help of simulation analysis of a hypothetical assembly. Among them, LP + (ROPT)2 was found to perform well with respect to mean staging delay. In this rule the term 'LP' refers to remaining path length of the remaining segment and 'ROPT' denotes the remaining number of operations of the job. The item having smallest value of LP + (ROPT)2 is chosen for loading.

A tie-breaking rule is used when two or more jobs of same priority wait in the queue. Adam, Bertrand and Surkis (1987) have observed the importance of tie breaking rules when they tested the performance of TWKR (total work content remaining or total processing times of remaining operations on the job) rule, with their proposed rules, namely, relative remaining operations (RRO) and relative remaining processing time (RRP).

The basic idea of RRP index is to consider the structural complexity in terms of relevant factor to the imminent operations of a multi-level job. Philipoom, Markland, and Fry (1989) have examined many due-date oriented dispatching rules, where each rule had three methods of setting due-date milestones such as job due-dates (JDD), assembly due-dates and operation due-dates (ODD) to control the progression of a job toward completion. Philipoom, Russell and Fry (1991) have proposed importance-ratio rule and evaluated a set of sequencing rules along with IR rule. They have shown that importance ratio (IR) rule when tested with TWKR as tie-breaking rule (IR:TWKR) performed well with respect to mean flowtime and percent tardy jobs. Adam et al (1993) have discussed various due-date assignment procedures in shop environments with multi-level assembly constraints. There are three methods of setting due-date milestones such as job due-dates (JDD), assembly due-dates and operation due-dates (ODD) to control the progression of a job toward completion. They have also proposed a procedure for setting due dates to the assembly jobs known as CPFT (Critical Path Flow Time).

TWKR, First in first out (FIFO) and Earliest completion time (ECT) rules are found to be benchmark rules in studies involving dispatching rules in an assembly based job shops. The combination of these rules with one another was found to minimize flow time based performance measures (Adam, Bertrand and Surkis (1987), Sculli (1987), Phillipoom et al (1991), Reeja and Rajendran (2000a) and Mohanasundaram et al (2002).

Reeja and Rajendran (2000a) proposed a new concept known as operation synchronization date (OSD) which paces the completion of items, accelerates completion time of operations to synchronize them at sub-assembly/assembly stages. The performances of OSD along with some combinations of FIFO, ECT and TWKR have proved to minimize mean and standard deviations of flow time based measures of performances. As per the literature survey it is found that TWKR: OSD (Reeja and Rajendran 2000a) performs best in minimizing the mean flow time and staging delay while the overall performance of ECT: OSD and OSD: ECT rules are better when compared to the performances of other rules.

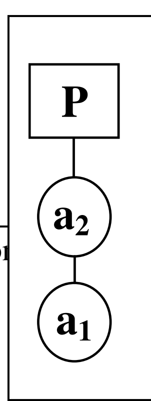
Thiagarajan and Rajendran (2003) have developed weighted dispatch rules to evaluate the measure of performance based on flowtime and cost in utmost three level job structures in the dynamic assembly job shop environment with the considerations of different holding and tardiness cost. The minimization of the total scheduling cost which is the sum of the holding cost and tardiness cost were taken as the primary performance measure while the minimization of mean flowtime and tardiness were the secondary measures. They have shown that the dispatch rules were efficient in minimizing the mean and maximum values of the primary measure, and were quite robust with respect to different job structures and experimental settings.

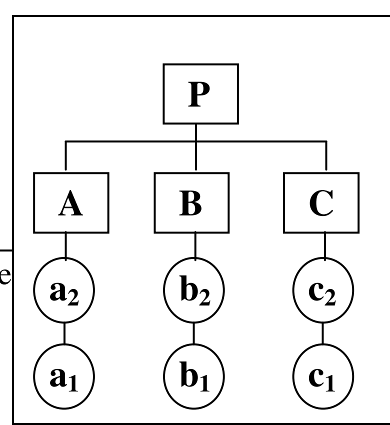
Thiagarajan and Rajendran (2005) have developed new dispatch rules that incorporate weights for earliness, tardiness and flowtime as a follow up of their previous work. Unlike other researchers, they have considered earliness, tardiness and flowtime costs specifically in an assembly shop environment where multi-level jobs are processed. Weighted dispatch rules showing best performance with respect to the minimization of weighted mean sum of weighted earliness and weighted tardiness of jobs, maximum sum of weighted earliness and weighted tardiness of jobs, variance and weighted variance of the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs have been proposed.

Omkumar and Shahabudeen (2006) have developed a new concept known as Available Due Date (ADD) for minimizing tardiness based measures of performance in dynamic assembly job shop environment. Available due date of an item is defined as the time remaining between the due date and the current instant without considering the queuing and staging delays. ADD is found by subtracting the processing time of the completed operations from the JDD of a job. They have shown that ADD along with LF and TWKR as tie breaker, performs well in minimizing mean and maximum of tardiness based measures of performance.

Assembly Job shop environment

In JSP, In a Job shop problem, jobs are finished by processing a specified set of operations on the raw material. Figure 1 shows the product structure in JSP, where a1 and a2 are operations required to complete product P. An assembly job shop consists of a machine shop division and an assembly shop, and deals with scheduling multilevel jobs as shown in Figure 2. A multilevel job requires machining operations in the machine shop, followed by assembly operations in the assembly shop. This goes on until all the levels are completed, and the entity exits the system as a finished product. There are certain unique





problems associated with scheduling assembly type or multi-level assembly jobs that do not arise when dealing with simple string-type of jobs. In a multi-level assembly job, a higher-level item cannot be processed unless all preceding lower level items have been completely processed and assembled together. It implies that an item may have to wait in assembly shop for its matching components, before the required assembly operations can take place. This structural complexity associated with assembly type of jobs introduces problems related to co-ordination and pacing that do not exist when dealing with string-type of jobs considered in conventional job shop scheduling. The scanty availability of research works in static assembly job shop scheduling and the complexity associated with it have motivated the authors in developing an optimization algorithm.

Figure – 1 Job structure in JSP

Figure – 2 Job structure in AJSP

So far ACO and many of the dispatch rules used in this work have not been applied for reducing makespan in multi-level static AJSS. The performance of the proposed heuristic is compared with various published dispatch rules used in scheduling of multilevel jobs in AJSS.

Simulation model of assembly Job shop

A simulation model of a hypothetical assembly job shop has been designed and developed for investigation. The test configurations and the multilevel product structures used for the experiment are shown in the Table 1 and Figure 3 respectively. Each workcenter consists of two machines and the assembly station consists of one workcenter. It is assumed that all jobs are available for processing at time t = 0. The processing time for operations on items/sub assemblies are drawn from uniform distribution ranging between 1 and 30, and the number of operations per item/sub assembly is sampled from a uniform distribution in the range of 5 to 8. Simulation runs for each type of machine configuration have been conducted with 10 replications using different data but identical environment.

Table.1: Machine and Product configurations experimented

Job Structure	Number of Jobs (J)	Number of work centers(W)	Configurations
Single level jobs	10	8	10J - 8W
	20	12	20J - 12W

	30	15	30J - 15W
Two level jobs	30	15	30J - 15W
	45	20	45J - 20W
	60	20	60J - 20W
Three level jobs	10	10	10J - 10W
	15	10	15J - 10W
	50	15	50J - 15W

The following assumptions are made in the study:

1. Each machine/workcenter can perform only one operation at a time.
2. Process pre-emption is not allowed.
3. Sequence of machines required and the processing time for every item are known a priori and are assigned before the entry of the job into the shop.
4. No restriction on queue length at any machine.
5. Shop floor interruptions like machine breakdown are not considered.
6. Set-up time is included in the processing time.
7. Items are dependent, i.e. assembly is involved.
8. Routing once generated cannot be changed.

9. No two successive operations of an item can be performed on the same machine.
10. No limiting resources other than machines/workcenters are used.
11. An operation can be undertaken only if its preceding operations are completed.

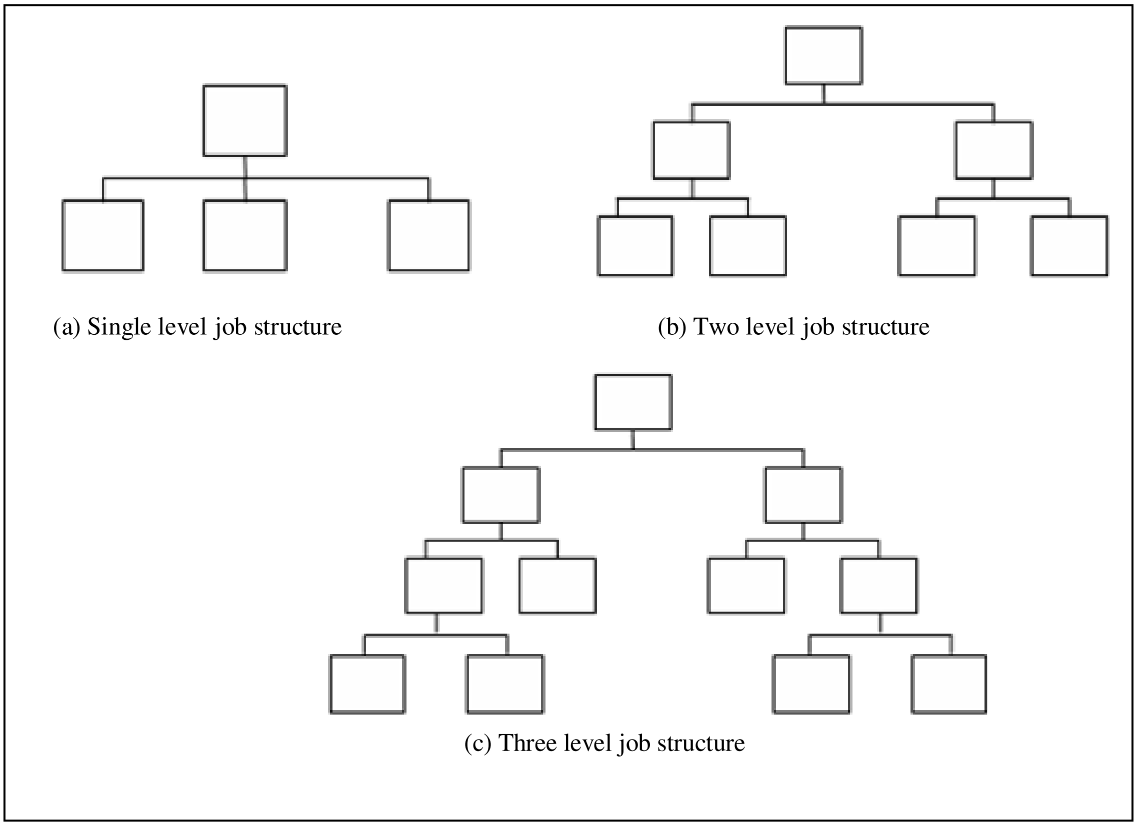


Figure – 3 Multi level job structures

Dispatch rules

When a machine is available, a job is selected from its input queue based on certain priorities known as dispatch rules. Table 2 lists some of the popular dispatch rules. Rules can be classified into static and dynamic. Static rules have priority indices

Table 2 Common Dispatch Rules

NAME	DESCRIPTION
SPT	(Shortest processing times) select the jobs with minimum processing time first.
RAND	Random selection of items for processing.
FIFO	(First in first out) select a job that arrives first to the machine queue.
ECT	(Earliest completion time of the job) Give highest priority for an operation of the job that has the earliest completion time.
LF	(Latest Finish time) Give highest priority for an operation for processing that has the latest finish time.
OSD	(Operation Synchronization date) select an operation for processing that has the least OSD value.
TWKR	(Total work remaining) select a job with smallest total processing time for unfinished operations.

that stay constant as jobs travel through the plant, whereas dynamic rules change with time as the job progresses. TWKR is dynamic, since the remaining processing time decreases when the job progresses through the shop. The rules are tested on individual basis as well as on combination basis and their performances are compared with that of the newly developed ACO heuristics and discussed elaborately in section 7.

Ant Colony Optimisation Algorithm

ACO is based on the co-operative behaviour, the adaptive memory of an ant system while collecting food and storing it in their colony. Consider Figure 4, the ants are in colony C and the resource R is available somewhere nearby. Now, ants from colony C go and explore the area surrounding them in search of food. Initially they move around randomly in search of the resource. The ants while moving deposit a pheromone trail in their path.The ants communicate with the aid of this pheromone trail. After some time some of the ants reach the resource. As soon as they find the resources, they collect some of the food and head back to the colony. After depositing the food in their colony, they go back to the resources and get some more food. Their way back is traced with the help of

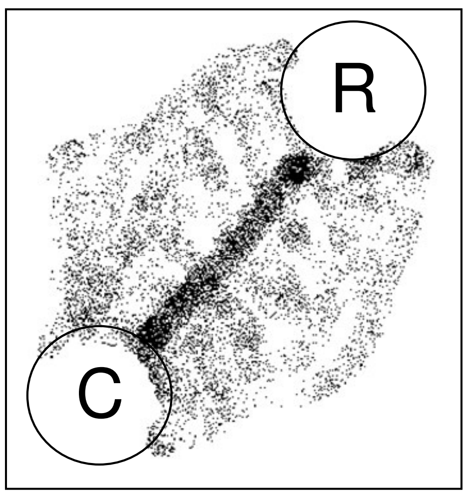


Figure - 4 Ants moving from colony to resources in search of food

pheromone trail. Since all the other ants may at some point sense this pheromone trail of the ants that are collecting resource, they too join the trail. The pheromone trail has the nature to evaporate. Over a period of time, shorter paths and frequently traveled paths accumulate more pheromone deposition, thereby attracting more ants. And after a time, a state of equilibrium is achieved in which all the ants travel along a well defined path which has a very high pheromone trail, and all the other lesser pheromone trail paths diminish.

ACO based AJSP model

The seminal work in ACO by Dorigo(1992) has been used to solve traveling salesman problem. In recent times, ACO has been used to solve scheduling problems in single machine total tardiness problem Merkle and Middendorf (2000), simple job shop problems Colorini et al. (1994) and Dorigo, Maniezzo and Colorni (1996), flow shop problems Stutzle (1998), Rajendran and Ziegler (2004) and Rajendran and Ziegler(2005) and flexible job shop problems Liouane et al. (2007). In this paper, an algorithm based on ACO has been proposed to minimize the makespan in static assembly job shop environment. Initially, a set of ants start from a dummy operation and move from operation to operation, depositing pheromone trail along its way, till all the operations are covered. The schedule generated in such a way, is used in the subsequent cycles to generate the next schedules. The ant chooses the next operation based upon the pheromone trails that it encounters. The pheromone intensity is the probability or desire for placing operation 'i' in position 'j', of the schedule. They are denoted by τ_{ij} (desire of placing operation i in j of schedule). For 'n' operation, these will be n^2 pheromone values.

An ant is characterized by different parameters. α denotes the probability that the next operation is selected based on pheromone intensity of the next η set of unscheduled operations of the previous best solution, ß denotes the probability of an ant choosing an operation from next η set of unscheduled operations in the schedule based on cumulative pheromone intensity. In this algorithm, 25 ants are used in each cycle. The pheromone trails are updated after each cycle, based on the best schedule generated in that cycle. The termination condition is chosen to be 50 cycles.

The ants are divided into different types based on their parameters. Some ants have high α and ß values, which makes the ant to stick close to the pheromone trails. So the solutions generated by these kinds of ants are small modifications of the previous cycle's solution. Some ants have low α and ß values, which makes the ants to explore more and create wider range of solution.

The rationale in fixing different (α, ß,η) values for different ants is that while some ants work in optimizing the best schedule of the previous cycle, some of the ants concentrate on exploring newer solutions. In this way, the solution does not get entrapped in local optima. The parameters of ants are determined on experimental basis as follows:

α	ß	%(no. ants)
0.8	0.1	25

0.6	0.3	25
0.5	0.4	25
0.4	0.5	25

The procedure begins with a fixed number of ants (25) starting from a dummy operation generating 25 different schedules. Then the makespan is evaluated for all the schedules. The best makespan is taken and based upon it, the pheromone intensities are updated.

$$\tau_{ij(new)} = [\tau_{ij(old)} * (1 - \rho)] + 1/ MS_{Best} \qquad (1)$$

Here ρ denotes the rate at which the pheromone trail evaporates and MS_{Best} indicates the best makespan value of previous cycle. ρ is set to 0.05 in all iterations.

Let 'N' be the total number of operations. An ant chooses an operation for each one of the 'N' positions in its schedule, based on its parameters. First, a random number, 'a' between 0 and 1 is generated. If this number is less than or equal to $\boldsymbol{\alpha}$, then the operation with maximum pheromone intensity for that position, from the first η unscheduled operations, from the previous cycle's best solution is selected by the ant. Here η is taken as 5 percent of the total number of operations.

$$= 0.05 * (N)$$

If 'a' is greater than $\boldsymbol{\alpha}$ and less than or equal to $\boldsymbol{\alpha} + \boldsymbol{\beta}$, an operation from the first η unscheduled operations is selected based on cumulative function (Iij) for operation i at position j.

If 'a' is greater than $\boldsymbol{\alpha} + \boldsymbol{\beta}$, then an unscheduled operation is chosen at random.

```
for (all the ants)
initialize random schedule subject to feasibility conditions
Set α and β values based on the probability defined
  Initialize the pheromone intensity table based on the schedule of the ant with minimum
makespan
  Generate the cumulative pheromone intensity table
    for (all the runs)
   {
      for (all the ants)
      {
         for (all the operations)
         {
                  Compute the pheromone intensity
                  }
         }
       for (all the positions)
         {
            Generate a random number between 0 to 1;
            if (a <= α)
            {
              Schedule among first η unscheduled operations, the operation with maximum
         pheromone intensity
```

```
        }
        else if (a <= α +β)
        {
          Schedule among first ηunscheduled operations, the operation with maximum
      cumulative pheromone intensity
        }
        else
        {
          Schedule among first ηunscheduled operations at random;
        }
      }
      Compute makespan();
    }
    Update trail intensities based on the best makespan
  }
 Select the schedule with the best makespan
```

The makespan of the best schedule in the final cycle is taken to be the makespan of ACO.

Results and discussions

For each configuration given in Table 1 the ACO based model is replicated 10 times. The results obtained for the single level, two level and three structures are shown in Tables 4. The values shown in the result tables are the averages of the ten results. The assembly time is taken to be zero as it does not affect the performance of the dispatch rules tested.

The proposed ACO gives best results compared to all other dispatch rules for all the three levels considered. Among dispatch rules, FIFO outperforms all other dispatch rules in all the three levels. For the single level structure, among the other dispatch rules,

Table 4 Results for All the Three Levels

Rules	Level 1			Level 2			Level 3		
	10J - 8W	20J -12W	30J -15W	30J -15W	45J -20W	60J -20W	10J -10W	15J -10W	50J -15W
FIFO	641.7	832.0	1023.0	2568.3	2664.8	3823.2	1169.3	2270.7	3087.6
RAND	795.1	1041.6	1234.1	3001.6	2815.3	3946.2	1174.2	2445.2	3161.9
TWKR:FIFO	685.8	858.5	1057.6	2752.7	2696.7	4038.7	1229.8	2395.5	3220.8
SPT:FIFO	727.4	875.6	1033.8	2637.9	2770.9	3975.0	1212.6	2399.8	3189.2
OSD:FIFO	730.0	894.0	1071.1	2622.7	2750.7	3882.7	1215.1	2277.5	3148.5
LF:FIFO	753.7	931.4	1052.9	2659.5	2667.2	3939.8	1194.8	2240.0	3159.9
ECT:FIFO	749.4	887.3	1089.9	2630.1	2796.0	3935.3	1199.7	2435.2	3139.7
SPT:ECT	721.4	872.1	1040.4	2674.7	2802.2	4048.5	1201.2	2434.3	3224.9
SPT:LF	726.0	910.6	1095.0	2678.7	2724.7	4007.9	1217.6	2330.8	3226.2
SPT:OSD	725.3	866.2	1022.1	2730.4	2777.0	4071.3	1244.5	2374.3	3247.6
SPT:TWK	726.1	869.6	1046.5	2632.4	2772.9	3988.6	1248.1	2416.2	3206.9
OSD:SPT	741.7	911.5	1059.8	2628.6	2759.8	3955.6	1203.7	2274.3	3145.2
OSD:LF	745.3	877.9	1083.7	2629.3	2751.8	3917.1	1193.2	2273.1	3138.9
OSD:ECT	738.3	905.6	1050.4	2620.7	2752.9	3889.6	1216.5	2273.5	3149.1

OSD:TWK	738.1	878.8	1051.1	2627.4	2742.1	3913.0	1221.3	2275.0	3145.3
LF:SPT	734.5	926.9	1043.3	2656.4	2679.1	3946.4	1183.5	2253.5	3162.5
LF:OSD	752.7	901.3	1049.8	2659.9	2678.1	3936.4	1201.6	2253.3	3163.0
LF:ECT	756.2	928.2	1075.5	2674.4	2676.8	3962.1	1164.0	2252.4	3162.4
LF:TWK	759.2	915.5	1075.2	2675.5	2677.2	3952.5	1165.0	2253.2	3160.2
ECT:SPT	757.3	912.0	1106.4	2651.1	2849.6	3901.8	1164.7	2452.4	3144.5
ECT:OSD	752.8	896.0	1112.1	2648.3	2844.7	3951.5	1150.4	2464.3	3143.8
ECT:LF	748.8	909.4	1105.2	2669.2	2826.8	3862.3	1168.9	2450.2	3141.0
ECT:TWK	746.3	890.9	1055.0	2645.2	2829.2	3929.8	1185.1	2479.4	3133.9
TWK:LF	689.7	870.9	1119.2	2735.7	2793.0	4046.8	1214.8	2434.7	3233.2
TWK:ECT	693.0	924.2	1119.9	2732.6	2813.6	3991.9	1264.5	2422.2	3237.4
TWK:SPT	681.5	911.8	1130.2	2697.7	2794.4	4039.7	1238.9	2434.1	3232.1
TWK:OSD	683.2	887.5	1106.5	2737.3	2793.1	3987.3	1225.2	2450.9	3229.0
ACO	**621.8**	**789.5**	**980.6**	**2498.6**	**2626.3**	**3769.7**	**1116.0**	**2220.5**	**3055.4**

In single level structure, once machining operations are completed, the part proceeds to assembly shop for assembly operations and does not return to machine shop. Ignoring the assembly waiting time the situation is exactly similar to the jobs processed in a job shop. This may explain the performance of SPT for single level job structures.

Comparison of results

The following indices are used to compare the makespan values obtained using ACO(note ACO gives the minimum makespan in all the configurations considered so far).

Percentage reduction in makespan (PRM)

PRM which indicates the percentage reduction in makespan by means of ACO compared to any other dispatch rule is given by

$$\mathrm{PRM_d} = \frac{(Makespan\, by\, dispatch\, rule\,(d) - Makespan\, by\, ACO)*100}{Makespan\, by\, ACO}$$

Average of the percentage reduction in makespan (APRM)

It is the average percentage reduction is makespan for a given configuration(c) considering all the dispatch rules (d) over ACO.

$$\mathrm{APRM_c} = \frac{\sum_{d=1}^{k} PRM_d}{\mathrm{k}}$$

NPRM

The percentage reduction in makespan by the dispatch rule ranking

next to ACO is represented by NPRM.

$$\mathrm{NPRM} = \frac{(Makespan\ \ ranking\ next\ to\ ACO - Makespan\ by\ ACO)*100}{Makespan\ by\ ACO}$$

Table.5 Effect of ACO compared to dispatch rules

Structure	Configurations	$APRM_{c,all}$	$APRM_{c,pd}$	NPRM
Single Level	10J - 8W	17.34	17.55	3.20
	20J - 12W	13.94	15.41	5.38
	30J - 15W	9.95	12.05	4.23
Second Level	30J - 15W	7.13	7.95	2.79
	45J - 20W	5.06	5.22	1.47
	60J - 20W	4.97	4.20	1.42
Third Level	10J - 10W	7.75	6.24	3.08
	15J - 10W	6.27	6.75	0.88
	50J - 15W	3.93	3.21	1.05

In Table.5, the values shown in the column $APRM_{c,all}$ gives the APRM values considering all the 27 combinations of dispatch rules while the values in the column $APRM_{c,pd}$ gives the APRM values considering the 7 published combinations of dispatch rules. Column NPRM shows the PRM values of dispatch rules ranking next to the performance of ACO for each configuration.

Statistical Analysis

To ascertain the significance of improvement in makespan by ACO, ANOVA is performed. F values obtained are given in the Table.6. From the ANOVA results it can be observed that the reduction in makespan is significant. Further to analyze the significance of difference between the pair of means, Fisher Least significant difference test is performed. The results are given in Table.7. It is observed that except for few cases the reduction in makespan obtained using ACO is significant when compared to the other dispatch rules.

Table.6 Summary of the absolute differences of the mean values for the configurations

Structure	Configurations	$F_{tabulated}$	$F_{calculated}$
Single Level	10J - 8W	1.53	9.59
	20J - 12W	1.53	5.29
	30J - 15W	1.53	5.68
Second Level	30J - 15W	1.53	12.93
	45J - 20W	1.53	10.09
	60J - 20W	1.53	5.46
Third Level	10J - 10W	1.53	2.28
	15J - 10W	1.53	4.52
	50J - 15W	1.53	3.39

Rules	**Level 1**			**Level 2**			**Level 3**		
	10J - 8W	20J -12W	30J -15W	30J -15W	45J -20W	60J -20W	10J -10W	15J -10W	50J -15W
	LSD =20.13	LSD =30.26	LSD =26.15	LSD =38.55	LSD =30.83	LSD =33.48	LSD =33.64	LSD =68.37	LSD =42.47
FIFO	19.9*	42.5	42.4	69.7	38.5	53.5	53.3	50.2*	32.2*

RAND	173.3	252.1	253.5	503	189	176.5	58.2	224.7	106.5
TWKR:FIFO	64.0	69	77	254.1	70.4	269	113.8	175	165.4
SPT:FIFO	105.6	86.1	53.2	139.3	144.6	205.3	96.6	179.3	133.8
OSD:FIFO	108.2	104.5	90.5	124.1	124.4	113	99.1	57*	93.1
LF:FIFO	131.9	141.9	72.3	160.9	40.9	170.1	78.8	19.5*	104.5
ECT:FIFO	127.6	97.8	109.3	131.5	169.7	165.6	83.7	214.7	84.3
SPT:ECT	99.6	82.6	59.8	176.1	175.9	278.8	85.2	213.8	169.5
SPT:LF	104.2	121.1	114.4	180.1	98.4	238.2	101.6	110.3	170.8
SPT:OSD	103.5	76.7	41.5	231.8	150.7	301.6	128.5	153.8	192.2
SPT:TWK	104.3	80.1	65.9	133.8	146.6	218.9	132.1	195.7	151.5
OSD:SPT	119.9	122	79.2	130	133.5	185.9	87.7	53.8*	89.8
OSD:LF	123.5	88.4	103.1	130.7	125.5	147.4	77.2	52.6*	83.5
OSD:ECT	116.5	116.1	69.8	122.1	126.6	119.9	100.5	53*	93.7
OSD:TWK	116.3	89.3	70.5	128.8	115.8	143.3	105.3	54.5*	89.9
LF:SPT	112.7	137.4	62.7	157.8	52.8	176.7	67.5	33*	107.1
LF:OSD	130.9	111.8	69.2	161.3	51.8	166.7	85.6	32.8*	107.6
LF:ECT	134.4	138.7	94.9	175.8	50.5	192.4	48	31.9*	107
LF:TWK	137.4	126	94.6	176.9	50.9	182.8	49	32.7*	104.8
ECT:SPT	135.5	122.5	125.8	152.5	223.3	132.1	48.7	231.9	89.1
ECT:OSD	131.0	106.5	131.5	149.7	218.4	181.8	34.4	243.8	88.4
ECT:LF	127.0	119.9	124.6	170.6	200.5	92.6	52.9	229.7	85.6
ECT:TWK	124.5	101.4	74.4	146.6	202.9	160.1	69.1	258.9	78.5
TWK:LF	67.9	81.4	138.6	237.1	166.7	277.1	98.8	214.2	177.8
TWK:ECT	71.2	134.7	139.3	234	187.3	222.2	148.5	201.7	182
TWK:SPT	59.7	122.3	149.6	199.1	168.1	270	122.9	213.6	176.7
TWK:OSD	61.4	98	125.9	238.7	166.8	217.6	109.2	230.4	173.6

Table.7 Fisher Least significant difference test showing the absolute differences of the mean values tabulated against the LSD values. (* indicates values are insignificant)

Conclusion

In this paper, the problem of scheduling static assembly job shop with the objective of minimizing makespan has been studied. The performance of several benchmark dispatch rules of dynamic assembly job shop scheduling and various other combinations have been tested and compared with that of a newly proposed ant colony algorithm for various configurations. It has been found that the ant colony algorithm yields better solutions as against the best known dispatch rules reported in recent research studies. ACO gives the average of the percentage reduction in makespan upto 17.55%. Also statistical analysis is performed to ascertain that the reduction in makespan in significant by ACO. The results of this study encourage the development of ant colony algorithms for various scheduling problems with different objectives. Future work may involve cost calculations, knowledge based scheduling methodologies and development of hybrid algorithms.

References

Adam, N. R., Bertrand, J. W. M., & Surkis, J, (1987). Priority assignment procedures in multi-level assembly job shops, *IIE Transactions*, *Vol.19, No.3*, pp.317–329.

Blackstone, J., Phillips, D. T., & Hogg, G. L, (1982), A state of the art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, *Vol.20*, 27–45.

Colorni, A, Dorigo, M, Maniezzo, V and Trubian, (1994), M, Ant system for job-shop scheduling, *Belgian journal of Operations Research*, *Statistics and Computer Science, Vol. 34*, pp.39-53

Dorigo,M,(1992), *Optimization, Learning and Natural Algorithms, Ph.D. Thesis, Dip. Elettronica e Informazione*, Politecnico di Milano, Italy.

Dorigo, M, Maniezzo, V and Colorni, A, (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.29-41

Goodwin, J. S., & Goodwin, J. C, (1982), Operating policies for scheduling assembled products, *Decision Sciences*, *Vol.13*, pp.585–603

Mohanasundaram.K.M, Natarajan.K, Viswanathkumar.G, Radhakrishnan.P and Rajendran.C, (2002), Scheduling rules for dynamic shops that manufacture multi-level jobs, *Computer & Industrial Engineering*, *Vol.44, No.1*, pp.119–131

Marath K. Reeja and Rajendran.C, (2000a), Dispatching rules for scheduling in assembly job shops Part-1, *International Journal of Production Research., Vol.38, No.9,* pp.2051–2066

Merkle, D, and Middendorf, M, (2000), An ant algorithm with a new pheromone evaluation rule for total tardiness problems. *In: Proceedings of the EvoWorkshops 2000, lecture notes in computer science 1803*, pp. 287–296. Berlin: Springer.

Nabril R. Adam, J.Will M Bertrand, Diane C. Morehead and Julius Surkis, (1993), Due date assignment procedures with dynamically updated coefficients for multi-level assembly job shops, *European Journal of Operational Research*, *Vol.68*, pp.212-227

Omkumar.M and Shahabudeen.P, (2006), *Simulation modeling and analysis of multilevel jobs in dynamic assembly job shops for due date based measures of performance*, CARs & FOF 2006, CAD/CAM Robotics and factories of the future, Narosa Publishing House, pp.902–906

Patrick R. Philipoom, Robertas.Russell, & Timothy D.Fry, (1991), A Preliminary investigation of multi attribute based sequencing rules for assembly shops, *International Journal of Production Research., Vol.29, No.4,* pp.739-753

Rajendran, C, and Ziegler, H, (2004), Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs, *European Journal of Operational Research, Vol.155*, pp.426–438.

Rajendran, C, and Ziegler, H, (2005), Two ant-colony algorithms for minimizing total flowtime in permutation flowshops, *Computers & Industrial Engineering Vol.48,* pp.789–797

Russell, R. S., & Taylor, B. W., III, (1985), An evaluation of sequencing rules for assembly job shop, Decision Sciences, *Vol.16,* pp. 196–212.

Stuetzle, T, (1998), An ant approach for the flow shop problem. *In: Proceedings of the sixth European Congress on intelligent techniques and soft computing (EUFIT '98), Vol. 3,* pp. 1560–1564, Aachen: Verlag Mainz.

Sculli, D, (1987), Priority dispatching rules in an assembly job shop, *OMEGA*, *Vol.15,* pp.49–57.

Thiagarajan.S and Rajendran.C, (2003), Scheduling in Dynamic Assembly Job-Shop with Jobs Having Different Holding and Tardiness Costs, *International Journal of Production Research, Vol.41, No.18*, pp. 4453-4486.

Thiagarajan.S and Rajendran.C, (2005), Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs, *Computers & Industrial Engineering, Vol.49, No.4,* pp. 463–503