International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

A Deterministic Flowshop Scheduling Problem to minimizing the Makespan using PA-ACO

Annu Priya, Sudip Kumar Sahana

Abstract: ***Scheduling problems are NP-hard in nature. Flowshop scheduling problems, are consist of sets of machines with number of resources. It matins the continuous flow of task with minimum time. There are various traditional algorithms to maintain the order of resources. Here, in this paper a new stochastic Ant Colony optimization technique based on Pareto optimal (PA-ACO) is implemented for solving the permutation flowshop scheduling problem (PFSP) sets. The proposed technique is employed with a novel local path search technique for initializing and pheromone trails. Pareto optimal mechanism is used to select the best optimal path solution form generated solution sets. A comparative study of the results obtained from simulations shows that the proposed PA-ACO provides minimum makespan and computational time for the Taillard dataset. This work will applied on large scale manufacturing production problem for efficient energy utilization.***

Keywords: ***Permutation Flowshop Scheduling Problem (PFSP), probability of Correct Selection (PS), High-performance computing (HPC)***

I. INTRODUCTION

A PFSP consists of a constant sequence set $\{S \in R^{+}_{\leq 0}\}$ of the non-permutable real-world problem. It also states that jobs 'i' (i =1, 2,… n) have processed on machines 'm' (m = 1, 2… m) having the processing time (t_{ij}). The processing time of the machine is assumed to be '0' if the job doesn't take part in the execution. Then this type of problem is assumed that one machine can process one job at a time, and the jobs available for processing are assumed to be sequence-independent. Here, we consider the PFSP to minimizing makespan [14, 10] of the solution set. The work presented in this paper is to reduce the computational time (CT$_{\text{PJ max}}$) and makespan of the Taillard dataset. Flow shop scheduling [8] with multiprocessor increase the computational capacity and also reduce the cost of the machine. There are various researchers have proposed several heuristic algorithmssuch as Genetic Algorithm (GA), Tabu Search, Particle Swarm Optimization (PSO) algorithms, etc.to provide the near-optimal solution at the relatively minor computational expense [9]. Ant Colony Optimization (ACO) is widely used for solving combinatorial problems.

Revised Manuscript Received on February 15, 2020.
* Correspondence Author
Annu Priya*, Department of Computer Science Engineering, Birla Institute of Technology, Jharkhand (Ranchi), India.
Sudip Kumar Sahana, Assistant Professor, Department of Computer Science and Engineering, Birla Institute of Technology, Jharkhand (Ranchi), India.

In the year 1992, Dorigo introduced the population-based search technique based on the behaviors of ant's hive [13].Ants are the natural food seeker and they use pheromone trail to create the shortest routes for their fellow ants. Ants do not have any visual power instead of they use pheromones to find the shortest route between foods to the nest. It has been experimentally proved that the ants will find the shortest path by using the pheromone trail. The first example of ant pheromone trail search is proposed by Dorigo for traveling salesman problem. In computer science, the colony of artificial ants helps users to finding the optimal solution from the given problem set. There are various versions of ACO algorithms are developed by different researchers to find the optimal results for various datasets.

A. Conventional Ant colony optimization

ACOsolves the complex combinatorial optimization problem using graph theory. The basic structure of the graph is as follows: the ants are set at the nodes and the edge between the nodes is considered for the trails. The higher pheromone concentrationon edgeshaving a maximum probability for next node selection and also identify the shortest path.Travelling Salesman Problem(TSP) is a classic problem that is solved by the ACO algorithm [34]. It consists of cost and distance between the cities. Here, thegraph $G = (i, j)$containscost at the node 'i' and distance between the nodes at edge 'j'. The work of ant has to complete their tour in the graph to find the shortest path. The next visiting node is selected by using a pheromone update. The Ant 'k' selects the node 'v$_i$' to node 'v$_j$' based on the given Eq. 1.

$$[\tau(i,j)]^{\alpha}[n(i,j)]^{\beta} = \max_{v1 \in J_k(i)} \{ [\tau(i,j)]^{\alpha} [n(i,j)]^{\beta}\} \qquad (1)$$

Here,
$\tau(i,j)$ = The pheromone level (i, j)
$\eta(i,j)$ =Cost at the node,
$J_k(i)$ = Visited node by ants
α = Edge Cost
β = Pheromone level
The next node 'v$_j$' is randomly selected based on the probability distribution which is represented in the Eq. 2.

$$p_k\,(i,j) = \frac{[\tau(i,j)]^{\alpha}[n(i,j)]^{\beta}}{\sum_0 {}_{v1 \in J_k(i)} [\tau(i,l)]^{\alpha}[n(i,l)]^{\beta}} \qquad (2)$$

If $j \in \; j_k(i)$otherwise
Once the ant tour is completed the local updating rule is applied for pheromone update. Eq. 3, is used to avoid the search of the near best tour.

$$\tau(i,j) = (1-\rho)\tau(i,j) + \rho\Delta\tau(i,j) \qquad (3)$$

Here, ρ = Pheromone evaporation rate$(0, 1)^*$
$\Delta\tau(i,j) = \tau_0$ = Initial pheromone rate
After the ant covers all the nodes and edges the global update rule is applied. Eq. 4, is used to find the global best route.

$\tau(i,j) = (1-\delta)\tau(i,j) + \delta\tau_{gb}(i,j)$ (4)

Here,

δ = Global evaporation rate (0, 1)

$\tau_{gb}(i,j) = \begin{cases} L_{gb}^{-1} \\ 0 \end{cases}$ If edge the (i, j) between source to destination ∊ global best tour, otherwise.

L_{gb}^{-1}= Global best tour by the ant.

The best solution is represented by the pheromone matrix. It consistsof 'jth' job at rows and 'ith' processor at the column.

II. LITERATURE REVIEW

Scheduling isone of the demanding areas for operational research. The researcher recognized that every year hundreds of papers arepublished in this field. In the year 1954 Johnson [31], present a brief study on two-machine flowshop scheduling problems. For flow shop scheduling problemTaillard's datasetconsists of 120 instances and each set of job runs on a different set of machines [6, 7]. There are different datasets present such as job shop scheduling problems, flowshop scheduling problems, etc. [32]. There are various traditional techniques available to solve this type of scheduling problem. However to obtain better results researcher move toward Metaheuristic algorithms like Tabu search [33], evolutionary algorithm [11], and particle swarm optimization. Cheng and Kovalyov [15], deals with batching and flowshop scheduling problems for the machine to reduce total completion time. The single operator flow shop problem studied by Iravani and Teo [16]. They suggested an optimal chain structure schedule to minimizing setup costs, makespan for machine-dependent jobs andmachines. To solve the two-machine flowshop scheduling problem T'kindt et al [17] introduced a technique SACO, it is a hybrid strategy for no wait two machine flowshop to reduce makespan in state transition rule. Shyu et al [18], designed a greedy heuristic technique that includes pheromone initialization, hybrid state transition rule and also local search rule to solve the given problem. The max-min ant system introduced by Rajendran and Ziegler [19], to solve flowshop scheduling problem. They proposed a uniquetechnique to compute the relative distance between the given job position. Graham et al. [20], studied machine environment limitation to minimize the objective function. Zhao and Tang [21] proposed a polynomial-time technique that considers process contingency with a single machine and also investigates scheduling problem deterioration. The objective of this work was to reduce makespan, computational time. The result shows that the proposed technique was reliable and effective for scheduling problems. Yang and Yang [22] suggested a polynomial technique, which has a high capacity to reduce the makespan and find the optimal solution for the given problem. The technique proposed by Yang and Yang [23] for a single machine includesthe aging effect and also maintainsthe position of the variable. Yang [24] developed a Polynomial-time technique that searches for the time-dependent learning effect of the machine which minimizes makespan and also reduces the total absolute deviation of the finished time. The results show that this approach has high efficiency and effectiveness. A mathematical model designed for the economic system to solve the deteriorated problem of the job sequence to find the optimal policy which is efficiently minimizesthe average total cost per unit time and computational time. Liu [26] proposed an algorithm PSO-EDA_PI which provides a 0.65% error rate against the other algorithm. Zhao et al. [27] introduced dynamic particle swarm optimization that found average relative error approximately 1.19-2.39% against the other algorithms. The authors Bauer et.al [28], Herroelen et al. [29] and Merkle et al. [30] studied Ant Colony Optimization for different scheduling problems that applied to RCPSP and also combined with other heuristic techniques to find the near-optimal solution. The concept of hybridization used for job shop, flowshop, one shop and grid computing problems.

III. PFSP FORMULATION

The PFSP formulation [1-5] maintains the identical sequence for processing 'J' jobs on 'M' machine. Each processor executes a single job at a time and it is also assumed that each job is processed at one machine at a given time interval (vice-versa). The execution time of processors is sequence-independentand each ready job is processed at time zero and pre-emption is not allowed at the time of processing. Here, it is assuming that every processor 'P' hasa set of job 'J' sequence and after job allocation, the completion time (CT_{PJ}) and processing time (PT_{PJ}) are calculated for every iteration. Then,

For (P=1 to P)

do

Completion time (CT_{PJ}) = $\max\{C_{(P-1)\,J}, C_{P\,(J-1)}\} + PT_{PJ}$, P=1, 2, 3… P and

J=1, 2, 3…J

Where,

$CT_{P1} = \sum_{P=1}^{P} PT_{P1}$, P=1, 2… P

$CT_{1J} = \sum_{J=1}^{J} PT_{1J}$, J=1, 2… J

IV. DESCRIPTION OF PRPOSED ANT COLONY OPTIMIZATION (PA-ACO)

Ant colony optimization mimics the pheromone trails of a real ant for searching food from source to destination. The proposed algorithm solves PSFP with a good solution. Where each job is to assign with artificial ant with an empty sequence. To construct the complete solution. The ant iteratively depends on the unscheduled job. At each step, the job solution is based on pheromone trails by applying transition rule. The performance quality of the constructed solution is then improved by taking the mean of the job set and building the two subsets (S_{max}, S_{min}) of the set (S) respectively. By using the swap technique new set (S_{new}) is generated. The structure of the swap technique as follows:

A. Proposed Algorithm

Step 1. Input the job sequence

Step 2. Calculate the mean of the job sequence

Step 3. Create two subsets $S_{max}$and $S_{min}$of the set (S) onthe basis of mean

Step 4. Set the parameters and generate the two random seeds for the subsets (S_{min}, S_{max})

Step 5. Swap the job sets of $S_{min}$and S_{max}

Step 6. Initialize pheromone trails

Step 7. The termination conditionis reached by applying the transition rule

Step 8. Apply the local update rule to search the optimal tour and update the solution.
Step 9. Apply global update rule at every iteration and update the trail.
Step 10. Apply the Pareto analysis for best trail selection.
Step 11. End the simulation and return the best solution.

1) Initialization of the Pheromone Trails

Thetrails intensity $(\tau_{ij})$of job 'J'at the ith position of the sequence is calculated by using pheromone rule. Let M^s_{max} be themakespan of the job sequence and he seed sequence (S_{new}) produced by the heuristic method. Then initial pheromone trails intensity calculated for the sequence(S) using the Eq. 5& Eq. 6,

$$\tau_{max}={}^{1}/_{\rho M^{S}_{max}} \qquad (5)$$
$$\tau_{min}=\cup\times {}^{1}/_{\rho M^{S}_{max}} \qquad (6)$$

τ_{min}= lower bound of the pheromone trails.
τ_{max} = upper bound of the pheromone trails.

Where,
ρ= pheromone trial evaporation rate
$\cup$= parameter between [0-1]
τ_{ij}= τ_{min} for the ithposition of job J= [1… N]

The pheromone trail is modified by using the Eq. 7.

$$\tau_{ij}=(1-\rho)\ \tau_{ij}+{}^{\rho}/_{M^{S}_{max}} \qquad (7)$$

2) Transition Rule

The proposed technique, it starts from an empty sequence of job set where each artificial ant constructs their complete solution by iteratively using the transition rule. So, to build a solution ant k, chosen one of the unscheduled jobs at the present position 'i' which is based on transition rule (i.e. pseudo-random proportional rule) as given below,
The pheromone trail of the scheduled job has probability q_0 and $(1-q_0)$ is the probability of an unscheduled job. The next node is selected by the ant is based on the J= arg max (τ_{ij}). Eq. 8, states the probability of next job selection (P^k_{ij}).

$$P^{k}_{ij}(\text{Next job selection probability}) = \left({}^{\tau_{ij}}/_{\sum_{l}\tau_{ij_{k}}}\right) \qquad (8)$$

From the above equation, the selection of jobs within the unscheduled set is calculated by using the selection rule in Eq. 9.
The rule for selection from unscheduled job set

$$= \frac{\sum_{i=1}^{n} processing\ time\ of\ job_{(ij)}}{\sum_{l=(n-m)}^{n} processing\ time\ value\ l\ from\ unscheduled\ job\ set\ at\ i^{th} poition} (9)$$

$$\%\text{deviation } (D_{ij})=\sum_{l\in n_i^k}{}^{J_p}/_{100}\ (10)$$

$$\text{Allocation}=\sum_{j=1}^{N} P_{jk}(11)$$

if
$P_{jk}\geq D_{ij}$
Then,
$P_{jk}\leftarrow 1$

Where,
j = Total no of processor
k = No of available processor

Eq. 10, represents the percentage deviation of each job from unscheduled set and Eq. 11, shows the comparison of percentage deviation of schedule and unscheduled job set in the processing queue. The maximum D^{max}_{ij} of the job is considered to allocate the next job to the machine.

3) Single Point Swap

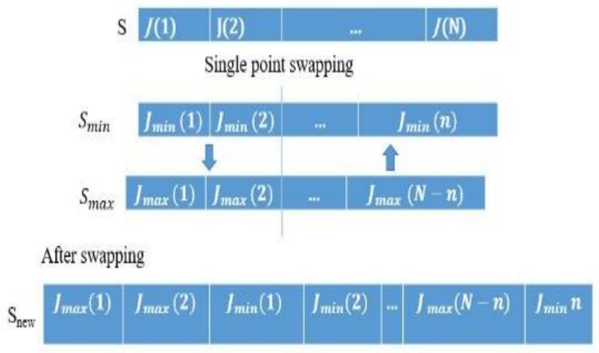

Fig.1.Single Point Swapping

The single point swapping process is a unique technique that uses S_{max} (Maximum processing time) andS_{min}(minimum processing time) of the jobs set.The new setS_{new} is created after a single swapping process.
Where,
S= Set of N jobs
S_{min} = Set of jobs {J_{min1}, J_{min2} … J_{minN}} with minimum processing time i.e. $\sum_{i=0}^{N} J_{min}$(i)
S_{max} = Set of jobs {J_{max1}, J_{max2} … J_{maxM}} with maximum processing time i.e. $\sum_{i=0}^{M} J_{max}$(i)
S_{new} = set of jobs after swapping
$\sum_{i=0}^{N} J_{min} \leq \sum_{i=0}^{M} J_{max}$ (For all the condition)

After applying the local search procedure the new ant solution is buildbased on the modified pheromone trails rule.

4) Local Search Procedure

Step 1. Input job sequence
Step 2. Calculate the mean of the job sequence
Step 3. Create two subsets $S_{max}$and $S_{min}$of the set (S)
Step 4. Swap the set jobs of $S_{min}$and S_{max}
Step 5. A new set sequence is generated (S_{new})
Step 6. Generate uniformly distributed random sequence (R) for mapping.
Step 7. When R ≤ S $_{new}$then
For everyithposition, if the job $'j'$ is not present at i^{th}position, then insert the job without changing sequence order. Start the new ant tour and calculate the makespan. Select only enhanced sequence and exchange the current job with the optimal one.

5) Selection of pheromone trails solutions

The multiple numbers of solutions are constructed by all the ants using local search procedure which shown in Table-I given below for Taillard data set instances.

The next step is to select the best solution from multiple numbers of solutions. Here, Eq. 12 & Eq. 13 is used for the probability of a correct solution (PS) andthe best solution (M-best) selection.

$$PS=\frac{\Delta M-M^{s}_{min}}{M^{s}_{max}-M^{s}_{min}} \qquad (12)$$

Where,

$$\Delta M=\frac{\sum_{i=1}^{n} M^{s}_{i}}{total\ number\ of\ iteration} \qquad (13)$$

M^{s}_{min}= minimum makespan of solution constructed by ants.
M^{s}_{max}= maximum makespan of solution constructed by ants.
ΔM = mean of the best solution (M-best).
Table-I, represents the makespan values obtain for different datasets using the PA-ACO algorithm.

Table-I: Makespan of the proposed algorithm

S.No	20×5	20×10	20×20	50×5	50×10	50×20	100×5	100×10	100×20	200×10	200×20	500×20
1	1278	1648	2345	2735	3116	3882	5472	5769	6529	10982	11562	26845
2	1297	1606	2330	2752	3134	4011	5495	5792	6970	11101	12080	26880
3	1316	1604	2315	2729	3109	4055	5493	5762	6944	11076	12048	26975
4	1322	1600	2291	2740	2990	4021	5519	5899	6899	11002	11959	27009
5	1324	1589	2257	2711	2956	4008	5540	5887	6785	10987	11948	27053
6	1347	1587	2323	2692	2932	4010	5527	5869	6698	10986	11541	27116
7	1367	1605	2321	2674	3156	3916	5529	5748	6893	10953	11516	27198
8	1285	1536	2334	2647	3151	3958	5514	5840	6735	11079	11503	27252
9	1281	1602	2274	2724	3123	4015	5495	5848	6642	10952	11672	27341
10	1289	1474	2272	2706	3143	3896	5514	5813	6536	10950	11684	27413

Let us consider to select the best makespan for simulated data. Here, the upper bound of the Taillard data set (20×5) is considered for makespan selection. According to table-I (20×5) dataset's makespan is considered for PS calculation. For selection the two hypotheses (Z) are constructed for makespan selection.
Z_0 = Makespan lies within the selection set.
Z_1 = Makespan doesn't belong to the selection set
According to Eq. 12 and Eq. 13, the PS is calculated from table-I, which is mention in table-II.

Table-II: PS values for different makespan

S.No	M^{s}_{min}	M^{s}_{max}	ΔM	PS
1	1278	1367	1310	0.35
2	1297			0.185
3	1316			-0.117
4	1322			-0.266
5	1324			-0.325
6	1347			-1.85
7	1285			0.304
8	1281			0.337
9	1289			0.26

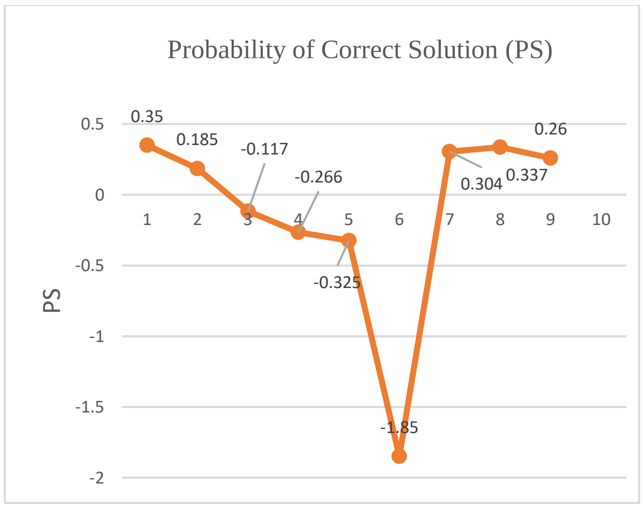

Fig.2. Graphical representation of PS values

Fig. 2 represents the makespan selection probability for (20×5) dataset. The selection is based on maximum probability achieve by individual makespan value. From table-II, it is found that there are four negative values that do not take part in the selection procedure due to their higher value of makespan. The rest of five positive values (0.35, 0.185, 0.304, 0.337, and 0.26) are selected for selection of makespan. The value which is nearer to '1' provides the best probability for selection of makespan. Therefore, 0.35 is selected for the best makespan solution. Initially, two hypotheses are considered as mentioned above Z_0 and Z_1. So, from fig. 1. It is clearly shownthat Z_1 is rejected and the null hypothesis is accepted. Similarly, this procedure is applied for different datasets for the selection of makespan.

6) Update pheromone trails

The solution obtained from the probability of correct solution is modified by using Global update rule. Firstly each pheromone trail is obtained as the best solution after that global update rule is applied for updating the set using Eq. 14.

$$\tau_{ij} = (1-\rho)\,\tau_{ij} + {\rho\, P_Z}/{M^{M-best}_{max}} \qquad (14)$$

Where,
M^{M-best}_{max} = Best makespan of entire ant tour
P_Z = Positive values of tour
Eq. 15, is used to calculate the compatible pheromone trail.

$$\tau_{ij} = (1-\rho)\,\tau_{ij} + {\rho}/{M^{k}_{max}} \qquad (15)$$

Where,
M^{k}_{max} = makespan of the complete sequence of ant k.
Lemma 1. The two parameters ΔM *and* M^{M-best}_{max} *are set such that* $\Delta M \geq M^{M-best}_{max}$
Proof: let $\Delta M_{(best)}$ is the average of pheromone trails calculated from Eq. 13, and at the end PS is calculated which is nearer to 1.

So, before selection of the current M (best), none of the trails is minimum then$\Delta M_{(best)}$.Let us assume thatM_{max}^{M-best}denotes the best makespan at the end of the iteration. According to Eq. 15, $\tau_{max} = {}^{1}/_{\rho M_{max}^{m-best}}$ is used before updating the sequence. Theτ_{ij}provides the M-best solution and theholds the inequality ${}^{P_Z}/_{M_{max}^{M-best}} > \tau_{ij}$. It is clearly adequate that${}^{P_Z}/_{M_{max}^{M-best}} > \tau_{max}$.

Let us consider the case if solution M-best is not updated during the current iteration i.e. $\Delta M \neq M_{max}^{M-best}$. Then the selection of M-best ≈ M_{max}^{M-best} in given as Eq. 16.

$$ {}^{P_Z}/_{M_{max}^{M-best}} > {}^{1}/_{\rho}\, M_{max}^{M-best(old)} = \tau_{max} (16)$$

V. RESULT AND DISCUSSIONS

The proposed Algorithm PA-ACOis simulated using pythonlanguage at high-performance computing (HPC) [12].HPC environment is consists of 2- Master Node with Intel Xeon processors havinga clock speed of 2.4GHz.It having the 8 cores and 64 GB memory for processing the task. The performance evaluation ofPA-ACO is based on Taillard's benchmark problem dataset. The test problems consist of various range of job sizes from (20 to 100) and it is processed on the machines at (5 to 20) defines (n × m). The performance evaluation is based on the makespan. The parameters for PA-ACO are set as ρ= 0.4, the number of ant = 5, u=0.005, z=20, and also the total number of iterations is taken as 150. The performance measurement is taken for the 5 Trail. Eq. 17, is used for calculating the makespan quality (M) between PA-ACO andTaillard's upper bound (UB).

$$M_{quality} = {}^{(M_{max}^{s} - UB)}/_{UB} \times 100 \qquad (17)$$

Table-III. Shows the performance comparison between proposed technique (PA-ACO) and existing techniques like MMAS [1], M-MAS [2], PACO [2], ACA [4], ACS [3], and NACA [5] for benchmark problems. From table-III, it is observed that the proposed technique (PA-ACO) obtains the better solution in a shorter CPU time period or nearer as compare to ACS and ACA. Which shows the superiority of the PA-ACO technique.

Table –III: Represents Result Evolution

Dataset (n × p)	PA-ACO		NACA		ACA		ACS		M-MMAS	MMAS	PACO
	Best	time	Best	Time	Best	Time	Best	Time			
20 × 5	-1.16	0.72	0.000	0.84	0.368	0.44	1.19	3.67	0.762	0.408	0.704
20 × 10	-0.66	1.12	0.079	1.57	0.831	0.50	1.70	4.00	0.890	0.591	0.843
20 × 20	-0.12	2.91	0.102	3.61	0.944	0.63	1.60	5.33	0.721	0.410	0.720
50 × 5	-0.58	4.62	0.011	5.03	0.085	2.77	0.43	14.67	0.144	0.145	0.090
50 × 10	-0.25	9.36	0.257	11.14	1.241	3.73	1.89	18.00	1.118	2.193	0.746
50 × 20	-0.10	20.44	1.252	22.71	1.990	5.91	2.71	24.33	2.013	2.475	1.855
100 × 5	-0.38	18.34	-0.006	19.46	0.070	14.15	0.22	54.33	0.084	0.196	0.072
100 × 10	1.31	40.17	0.283	43.68	1.059	21.93	1.22	65.67	0.451	0.928	0.404
100 × 20	0.23	90.12	0.761	93.94	1.833	37.79	2.22	88.00	1.030	2.238	0.985
200 × 10	0.23	170.84	0.150	177.39	0.434	141.52	0.64	275.33			
200 × 20	0.12	352.41	0.306	389.67	1.236	254.06	1.30	631.67			
500 × 20	2.90	3040.47	0.230	2447.4	1.444	3744.3	1.68	5133.0			

VI. CONCLUSION

The paper introduced the PA-ACO algorithm for PFSP. The objective of the paper is to reduce the makespan given problem sets. The simulation results evidence that PA-ACO provides the fast local search procedure and achieves high optimal path solutions constructed by artificial ants at a limited CPU time period. Once all the solution is constructed by artificial ants the probability of correct solution is applied to select the trails solution is generated by ants. The makespan is updated by using a global update rule. The results ofPA-ACO for PFSP are very promising and it is suggested that the proposed technique successfully applied for scheduling problems.

ACKNOWLEDGMENT

We would like to thanks Birla Institute of Technology, Mesra, Ranchi for providing me various resources for carrying thisresearch work. We also like to acknowledge our thanks to the publisher of this Journal forthe wide circulation of the research article.

REFERENCES

1. T. Stutzle and H. Hoos, "MAX-MIN Ant System and local search for the traveling salesman problem," Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97), Indianapolis, IN, USA, 1997, pp. 309-314.doi: 10.1109/ICEC.1997.592327.
2. C. Rajendran and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs," European Journal of Operational Research, 2004, vol. 155, no. 2, pp. 426–438. doi:10.1016/S0377-2217(02)00908-6
3. K.-C. Ying and C.-J. Liao, "An ant colony system for permutation flow-shop sequencing," Computers & Operations Research, 2004, vol. 31, no. 5, pp. 791–801.doi:10.1016/s0305-0548(03)00038-8.
4. F. Ahmadizar and F. Barzinpour, "A hybrid algorithm to minimize makespan for the permutation flow shop scheduling problem," International Journal of Computational Intelligence Systems, 2010, vol. 3, no. 6, pp. 853–861. doi:10.2991/ijcis.2010.3.6.15.
5. F. Ahmadizar, "A new ant colony algorithm for makespan minimization in permutation flow shops," Computers & Industrial Engineering, 2012, vol. 63, no. 2, pp. 355–361. doi:10.1016/j.cie.2012.03.015.

6. E. Taillard, “Benchmarks for basic scheduling problems,” European Journal of Operational Research, 1993, vol. 64, no. 2, pp. 278–285.doi:10.1016/0377-2217(93)90182-m.
7. E. Taillard, “Some efficient heuristic methods for the flow shop sequencing problem,” European Journal of Operational Research, 1990, vol. 47, no. 1, pp. 65–74. doi:10.1016/0377-2217(90)90090-x.
8. M. Kurdi, “Ant colony system with a novel Non-DaemonActions procedure for multiprocessor task scheduling in multistage hybrid flow shop,” Swarm and Evolutionary Computation, 2019, vol. 44, pp. 987–1002. doi:10.1016/j.swevo.2018.10.012.
9. H.-S. Woo and D.-S. Yim, “A heuristic algorithm for mean flowtime objective in flowshop scheduling,” Computers & Operations Research, 1998, vol. 25, no. 3, pp. 175–182. doi:10.1016/s0305-0548(97)00050-6.
10. H. D. Pour, “A new heuristic for the n-job, m-machine flow-shop problem,” Production Planning & Control, 2001, vol. 12, no. 7, pp. 648–653. doi:10.1080/09537280152582995.
11. A. Priya and S. K. Sahana, “A Survey on Multiprocessor Scheduling Using Evolutionary Technique,” Lecture Notes in Electrical Engineering, 2018, pp. 149–160.doi: 10.1007/978-981-13-0776-8_14.
12. A. Priya and S. K. Sahana, “Processor Scheduling in High-Performance Computing Environment Using MPI,” Innovations in Soft Computing and Information Technology, 2019, pp. 43–54.doi:10.1007/978-981-13-3185-5_5.
13. M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents,” IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 1996, vol. 26, no. 1, pp. 29–41. doi:10.1109/3477.484436.
14. J. M. Framinan, R. Leisten, and R. Ruiz-Usano, “Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation,” European Journal of Operational Research, 2002, vol. 141, no. 3, pp. 559–569.doi:10.1016/s0377-2217(01)00278-8.
15. T. C. E. Cheng and M. Y. Kovalyov, “Scheduling a Single Server in a Two-machine Flow Shop,” Computing, 2003, vol. 70, no. 2, pp. 167–180.doi:10.1007/s00607-002-1467-8.
16. S. M. R. Iravani and C. P. Teo, “Asymptotically optimal schedules for single-server flow shop problems with setup costs and times,” Operations Research Letters, 2005, vol. 33, no. 4, pp. 421–430.doi:10.1016/j.orl.2004.08.007.
17. V. T’kindt, N. Monmarché, F. Tercinet, and D. Laügt, “An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem,” European Journal of Operational Research, 2002, vol. 142, no. 2, pp. 250–257. doi:10.1016/s0377-2217(02)00265-5.
18. S. J. Shyu, B. M. T. Lin and T. S. Hsiao, "An ant algorithm for cell assignment in PCS networks," IEEE International Conference on Networking, Sensing and Control, 2004, Taipei, Taiwan, 2004, pp. 1081-1086 Vol.2. doi: 10.1109/ICNSC.2004.1297097.
19. C. Rajendran and H. Ziegler, “Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs,” European Journal of Operational Research, 2004, vol. 155, no. 2, pp. 426–438. doi:10.1016/s0377-2217(02)00908-6.
20. R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, “Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey,” Annals of Discrete Mathematics, 1979, pp. 287–326. doi:10.1016/s0167-5060(08)70356-x.
21. C. Zhao and H. Tang, “Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan,” Applied Mathematical Modelling, 2010, vol. 34, no. 3, pp. 837–841. doi:10.1016/j.apm.2009.07.002.
22. S.-J. Yang and D.-L. Yang, “Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities,” Computers & Mathematics with Applications, 2010, vol. 60, no. 7, pp. 2161–2169.doi:10.1016/j.camwa.2010.08.003.
23. S.-J. Yang and D.-L. Yang, “Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities,” Omega, 2010, vol. 38, no. 6, pp. 528–533. doi:10.1016/j.omega.2010.01.003.
24. S.-J. Yang, D.-L. Yang, and T. C. E. Cheng, “Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance,” Computers & Operations Research, 2010, vol. 37, no. 8, pp. 1510–1514. doi:10.1016/j.cor.2009.11.007.
25. J. Sicilia, J. F. Acosta, and M. G. D. la Rosa, “Economic order quantity for a power demand pattern system with deteriorating items,” European J. of Industrial Engineering, 2013, vol. 7, no. 5, pp. 577. doi:10.1504/ejie.2013.057381.
26. H. Liu, L. Gao, and Q. Pan, “A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem,” Expert Systems with Applications, 2011, vol. 38, no. 4, pp. 4348–4360. doi:10.1016/j.eswa.2010.09.104.
27. F. Zhao, J. Tang, J. Wang, and J. Jonrinaldi, “An improved particle swarm optimisation with a linearly decreasing disturbance term for flow shop scheduling with limited buffers,” International Journal of Computer Integrated Manufacturing, 2013, vol. 27, no. 5, pp. 488–499. doi:10.1080/0951192x.2013.814165.
28. A. Bauer, B. Bullnheimer, R. F. Hartl and C. Strauss, "An ant colony optimization approach for the single machine total tardiness problem," Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1445-1450 Vol. 2. doi: 10.1109/CEC.1999.782653.
29. W. Herroelen, B. De Reyck, and E. Demeulemeester, “Resource-constrained project scheduling: A survey of recent developments,” Computers & Operations Research, 1998, vol. 25, no. 4, pp. 279–302.doi:10.1016/s0305-0548(97)00055-5.
30. D. Merkle, M. Middendorf, and H. Schmeck, “Ant colony optimization for resource-constrained project scheduling,” IEEE Transactions on Evolutionary Computation, 2002,vol. 6, no. 4, pp. 333–346. doi:10.1109/tevc.2002.802450.
31. S. M. Johnson, “Optimal two- and three-stage production schedules with setup times included,” Naval Research Logistics Quarterly, 1954, vol. 1, no. 1, pp. 61–68. doi:10.1002/nav.3800010110.
32. E. Demirkol, S. Mehta, and R. Uzsoy, “Benchmarks for shop scheduling problems,” European Journal of Operational Research, 1998, vol. 109, no. 1, pp. 137–141.doi:10.1016/s0377-2217(97)00019-2.
33. E. D. Taillard, “Parallel Taboo Search Techniques for the Job Shop Scheduling Problem,” ORSA Journal on Computing, 1994, vol. 6, no. 2, pp. 108–117. doi:10.1287/ijoc.6.2.108.
34. S. K. Sahana, “Hybrid optimizer for the travelling salesman problem,” Evolutionary Intelligence, 2019, vol. 12, no. 2, pp. 179–188. doi:10.1007/s12065-019-00208-7.

AUTHORS PROFILE

First Author Annu Priya, received her B.E. degree in Computer Science Engineering from Bangalore Institute of Technology, India in 20014, and her M.E. (Software Engineering) From BIT Mesra, India in 2016. Currently, she doing her Ph.D. in Computer Science Engineering Department from Birla Institute of Technology, Jharkhand (Ranchi), India. She has worked as a part-time faculty member in the Dept. Computer Science & Engineering, RTC Institute of Technology, Anandi, Ormanjhi, Ranchi.Her current research focuses on scheduling problems in a High-Performance Computing Environment (HPC) and also her research area of interest is Soft & Evolutionary Computing, and Image processing. She has published a journal, book, and several conference papers.

Second Author Sudip Kumar Sahana received his B.E. degree in Computer Technology from Nagpur University, India in 2001, and his M.Tech. (Computer Science) and Ph.D. (Engineering) from the BIT, Mesra, India in 2006 and 2013, respectively. His major field of study is in Computer Science. He is currently working as an Assistant Professor at the Department of Computer Science and Engineering, BIT form 2007. His research and teaching interests include computational intelligence, Swarm and Evolutionary Computing, Distributed Computing, Artificial Neural Network, and Artificial Intelligence. He has authored several research papers in a reputed journal, books and also serves as an editorial member in the reputed journals.