World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:5, No:3, 2011

An Efficient Ant Colony Optimization Algorithm for Multiobjective Flow Shop Scheduling Problem

Ahmad Rabanimotlagh

***Abstract*—**In this paper an ant colony optimization algorithm is developed to solve the permutation flow shop scheduling problem. In the permutation flow shop scheduling problem which has been vastly studied in the literature, there are a set of m machines and a set of n jobs. All the jobs are processed on all the machines and the sequence of jobs being processed is the same on all the machines. Here this problem is optimized considering two criteria, makespan and total flow time. Then the results are compared with the ones obtained by previously developed algorithms. Finally it is visible that our proposed approach performs best among all other algorithms in the literature.

***Keywords*—**Scheduling, Flow shop, Ant colony optimization, Makespan, Flow time

I. Introduction

FLOW shop scheduling (FSS) problem which is one of the most widely studied scheduling problems is defined as processing a set of n jobs on a set of m machines. Sequence of operations belonging to each job is the same for all jobs i.e. all the jobs follow the same route on the machines. The objective is to find a sequence of jobs on each machine which is the optimal solution of the problem under given criterion. The first study concerning flow shop scheduling problem is done in [1] considering 2 machines and n jobs. In [2] it has been shown that from the complexity point of view, flow shop scheduling problem is NP-Complete in a strong sense which causes researchers not to be able to find globally optimal solution for these hard combinatorial optimization problems in a reasonable amount of time. Thus many researches tend to use metaheuristic methods to find good and not necessarily optimal solutions to large size flow shop scheduling problems. In [3], author solved the two-machine flow-shop problem with weighted late work criterion and common due date using simulated annealing, tabu search and variable neighborhood search. In [4], author tried to minimize makespan in a permutation flow shop scheduling problem using a hybrid of genetic algorithm and variable neighborhood search.in [5]Flow shop scheduling to minimize the total completion time with a permanently present operator was considered. They developed an ant colony optimization algorithm to solve this problem. In [6] a novel simulated annealing approach for the flow shop scheduling problem was developed. In [7], Author deployed a hybrid ant colony optimization algorithm to minimize total flow time in a flow shop sequence dependent group scheduling problem. There are several other heuristic approaches developed for flow shop scheduling problem. The interested reader may refer to [8]-[9]-[10]-[11]-[12] and [13].

Many of the studies concerning flow shop scheduling problems try to optimize the problem considering single criterion while in many manufacturing environments it is desired to optimize the problems considering two or more criterions. These are called multi objective problems. In [14] flow shop scheduling problems with the objectives of minimizing makespan and total flow time was modeled as a mixed integer goal programming and then in [15] a branch and bound algorithm to solve this problem was presented where there were two machines. High complexity of this problem is the reason that heuristic methods are desirable to solve it. In [16] a memetic algorithm is developed to solve this problem. In [17], [18] and [19], a bi objective flow shop scheduling problem is solved through the use of heuristic algorithms. But the newest research done in this area is [20]. The flow shop scheduling problem considering three objective functions were done in [21]-[22] and [23]. The newest study considering total machine idle time, total flow time and makespan as three objective functions was done in [24].

This study tries to optimize flow shop scheduling problem considering two criteria, makespan and total flow time. Equally weighting two mentioned criteria, an ant colony optimization approach is developed to optimize the problem and the results are compared with the methodspresented in the past. Computational experiments are conducted on the benchmark problems as the test problem in order to verify the algorithm performance.

The remainder of this paper has been organized as follows. In section 2 terminology and notation used in the paper are introduced and the problem is described. In section 3 the proposed approach to solve the problem is explained. Computational experiments are presented in section 4 and finally conclusions are discussed in section 5.

II. Problem Description

In some manufacturing and assembly systems, there are a lot of jobs, each of which has to undergo a series of operations. Often these jobs must follow the same route. The machines are assumed to be setup in series and the environment is referred to be as a flow shop [25]. This paper deals with thepermutation flow shop scheduling problem, i.e.

Ahmad Rabanimotlagh is with the Industrial Engineering Department, University of Kurdistan, Sanandaj, Iran (e-mail: Ahmad.rabani@ uok.ac.ir).

jobs sequence is the same on all the machines, considering two criteria. In our multi objective permutation flow shop scheduling problem, following notation is used:

n: total number of jobs in the shop floor
m: total number of machines in the shop floor
O_{ij}: the processing of job j on machine i
p_j: the job scheduled at the jth position of the sequence
$C^S(p_j, i)$: the completion time of job at position jth of sequence S on machine$i$
P_{ij}: the processing time of job j on machine i
w_1: the weight corresponding to makespan objective
w_2: the weight corresponding to total flow time objective
S: a complete sequence of n jobs
C^S_{max}: the makespan of the problem when under solution S
F^S: the total flow time of the problem under solution S
Z^S: total objective function corresponding to solution S (the weighted combination of makespan and total flow time corresponding to solution S)

Given a schedule $\{p_1, p_2, \dots, p_n\}$ for anm machine flow shop scheduling problem, considering the properties and constraints, it is obvious that the following equations hold:

$$C^S(p_1, i) = \sum_{l=1}^{i} P_{lp_1} \qquad i = 1,2, \dots, m \quad (1)$$

$$C^S(p_j, 1) = \sum_{l=1}^{j} P_{1p_l} \qquad j = 1,2, \dots, n \quad (2)$$

$$C^S(p_j, i) = Max\{C^S(p_{j-1}, i), C^S(p_j, i-1)\} + P_{ip_j} \quad i = 2,3, \dots, m \text{ and } j = 2,3, \dots, n \quad (3)$$

$$C^S_{max} = C^S(p_n, m) \quad (4)$$

$$F^S = \sum_{j=1}^{n} C^S(p_j, m) \quad (5)$$

And the total objective function value corresponding to solution S may be obtained by the equation stated in (6).

$$Z^S = w_1 C^S_{max} + w_2 F^S \quad (6)$$

III. Description of The Proposed Approach

In this paper an ant colony optimization (ACO) method is presented to solve multi objective flow shop scheduling problem. ACO algorithms are nature inspired metaheuristic methods which simulate foraging behavior of real ant colonies to solve combinatorial optimization problems. ACO is based on the observation of real ant colonies searching for food which use pheromone trails as communication medium. ACO algorithms are essentially constructive algorithms. That is in these algorithms each ant starts with a null solution and step by step adds components to this partial solution until a complete solution is obtained. In each step of construction procedure two types of information is needed for each ant to select the next component:

Pheromone trails, which represent the experienced desirability of adding each of the remaining components to the solution. This kind of information is updated during the construction procedure by ants and depending on the solutions found during the search procedure. The objective of this type of information is to memorize useful information found by good solutions.

Heuristic information, which is essential for generation of high-quality solutions. It represents the heuristic preference of adding each of the remaining components to the solution. It depends on the knowledge of the problem and is not modified during the search procedure.

ACO algorithm was first introduced in [26] and because of its weak performance in its primary applications for complex combinatorial optimization problems, it was then developed in several versions such as Elitist ant system [27], Rank-based ant system [28], Max-Min ant system [29], Ant colony system [30] and etc. to achieve better results.

A. Overview of the approach

This algorithm is written for multi objective flow shop scheduling problem so it is named "ACO-MOFS". In order to describe the proposed the approach clearly, first an overview of the algorithm is presented and then the details are explained.

Algorithm 1 The structure of the ACO-MOFS
Initialize pheromone trails and parameters. SetSJ as the set of schedulable jobs and $\hat{s}$ as the partial schedule.
While (termination condition is not met) **do**
For each ant in the colony **do**
Select the starting job Applying state transition rule.
For (t=2,3,…,n) **do**
Identify all schedulable jobs and include them in SJ.
Apply state transition rule to all the jobs in *SJ* and select one of them.
Apply local updating rule to the pheromone value compatible with the solution being constructed.
end for
Apply global updating rule to the best solution found so far or the iteration best solution.
Apply local search algorithm to the iteration best solution and in case of improvement, update the best solution found so far.
end for
end while

At the start the value of the parameters are fixed and all the pheromone trail values are set to τ_0. Also note that set SJat each iteration contains all the jobs which have not been added to the partial schedule yet.

B. State transition rule

In the state transition rule step, first a random number q uniformly distributed in the interval [0, 1] is generated. If $q \leq q_0$, the next job among all the ones in $SJ$is selected

according to the formula stated in (7).

$$j = arg\ max\ \left\{(\tau_j^t)^\alpha (\eta_j^t)^\beta\right\} \quad (7)$$

This is called exploitation step. If $q > q_0$, exploration step is done. That is the formula stated in (8) is applied to select the next component of the solution.

$$P_j = \frac{(\tau_j^t)^\alpha (\eta_j^t)^\beta}{\sum_{j' \in SJ} (\tau_{j'}^t)^\alpha (\eta_{j'}^t)^\beta} \qquad \forall\, j \in SJ \quad (8)$$

whereq_0 denotes relative importance of exploitation versus exploration. Also τ_j^t and η_j^t denote the desirability of placing job j at position t of the schedule and heuristic preference of scheduling job$j$at position t respectively. αandβ represent the relative importance of pheromone trail values and problem specific heuristic information respectively. There are a lot of ways of defining heuristic information associated with allocating a job to a position. For example in [24], the distance between two jobs SPIIRIT has been used which was presented in [31]. But as one of the contributions to this paper, in each solution construction step the heuristic preference of selecting job j as the next component of the schedule is calculated using (9).

$$\eta_j^t = \frac{1}{Z^{\hat{s} \cup j} - Z^{\hat{s}}} \quad (9)$$

where $Z^{\hat{s}}$ denotes the total weighted objective function associated with partial solution $\hat{s}$when job j has not been added yet and $Z^{\hat{s} \cup j}$ represents the total weighted objective function associated with this partial solution when job j is added.

C. Local updating rule

At each construction step, as soon as the next job to be scheduled is selected, the local updating rule is applied to corresponding pheromone value. Assume the next job to be scheduled is selected to be job j and this job lies at the position t of the schedule being constructed. Since the objective of this type of updating is to decrease the pheromones on the paths ants traverse to prevent different ants to traverse same paths,pheromone local updating is applied as (10):

$$\tau_j^t = \text{Min}\{\tau_j^t, (1-\rho')\tau_j^t + \rho'\tau_u\} \quad (10)$$

where ρ', a parameter in the interval [0,1] denotes local pheromone evaporation rate and τ_u follows (11):

$$\tau_u = \theta \frac{1}{\rho Z^{GBS}} \quad (11)$$

whereθ denotes a fixed parameter and Z^{GBS} represents the objective function value of the best solution found so far and it is updated whenever the best solution found so far is updated. It is obvious that in the local updating step the pheromone values corresponding to the solution under construction never increase.

D. Global updating rule

After all ants in the colony have completed constructing their solution the global updating rule is applied to the pheromone values compatible with iteration best solution or best solution found so far. Global updating consists of two sequential phases. At first phase, the amount of all pheromone trail values is evaporated as (12):

$$\tau_j^t = (1-\rho)\tau_j^t \qquad j = 1,2,\dots,n \text{ and } t = 1,2,\dots,n \quad (12)$$

This prevents the fast convergence of the algorithm toward local optima. Then in the second phase, if the algorithm is running within the first 100 of its iterations the global updating rule is applied to the iteration best solution according to the formula stated in (13). Otherwise it is applied to the best solution found so far.

$$\tau_j^t = \tau_j^t + \rho \frac{1}{Z^{GBS}} (13)$$

where ρ, a parameter selected in the interval [0,1] represents the global pheromone evaporation rate. As it can be seen in the formula stated in (13), as the objective function value of the iteration best solution or the global best solution improves, more pheromones are deposited on the paths corresponding to these solutions.

E. Local search algorithm

The vast literature on ACO methods has made it clear that a promising approach for obtaining high-quality solutions is to couple a local search algorithm with the main ACO algorithm. It is based on the iterative exploration of neighborhoods of solutions trying to improve the current solution by local changes. The types of local changes that may be applied to a solution are defined by a neighborhood structure. The choice of an appropriate neighborhood structure is crucial for the performance of a local search algorithm and is problem-specific [32]. After doing global updating phase, the local search algorithm is applied to the iteration best solution. Two neighborhood structures are used in our local search algorithm to explore different areas of the solution space of the problem under consideration:

1. Adjacent pairwise interchange method i.e. swapping the position of two adjacent jobs in the sequence
2. Insertion of each job in each possible position of the sequence. Consider the sequence $\{p_1, p_2, \dots, p_t, p_{t+1}, \dots, p_n\}$. Start by removing the job at the first position and placing it at positions 2,3,…,n respectively. Then continue with the job at the second position and placing it at positions 3,4,…,n respectively. Follow this procedure until all jobs are tested at all possible positions.

The first neighborhood structure is used by probability r and the second one with probability $1 - r$.

IV. Computational Experiments

The developed algorithm has been coded in Visual C# 9 under Microsoft Windows XP operating system, running on a Pentium IV, 3.60 GHz PC, with 1 GB memory. To set the parameters, a method has been used that is needed to be explained clearly. First the most effective parameter on the performance of the algorithm has been identified and its primitive valuehas been set. Then starting with the first parameter, the algorithm is executed under different values for that parameter. This is done for the rest of the parameters sequentially. For the sake of stability, this procedure has been again applied to all the parameters and best values for the parameters have been obtained as follows: $\alpha = 0.1,\ \beta = 0.5,\ \rho' = 0.05,\ \rho = 0.2,\ q_0 = 0.9,\ \theta = 0.01,\ r = 0.1,\ \tau_0 = 0.5$ and in each colony there are 20 ants to construct their solutions. The maximum number of iterations as the termination condition is 750. The performance of the algorithm for multi objective flow shop scheduling problem is tested using benchmark problem sets given in [33] from the literature. The number of jobs for all instances is 20 and number of machines varying from 5 to 20. There are 10 instances for each problem size. Because of the random nature of the problem, each test has been run 5 times and the best solution obtained by these runs has been used for the evaluation of the proposed approach (ACO-MOFS). The results obtained by the developed algorithm (ACO-MOFS) are compared by HAMC algorithms (HAMC1, HAMC2, HAMC3) proposed in [19], CR(MC) algorithm proposed in [17], and also the newest developed approach for this of problem, MOACSA proposed in [20].Equal weights are used for each criterion i.e. $w_1 = 0.5$and $w_2 = 0.5$. To evaluate the performance of ACO-MOFS the relative percentage increase in objective function value is calculated applying the formula stated in (14) which has been first introduced in [24].

$$RE(S) = \left(w_1 \left(\frac{C^S_{max} - \text{Min}\,(C^S_{max})}{\text{Min}\,(C^S_{max})} \right) + w_2 \left(\frac{F^S - \text{Min}(F^S)}{\text{Min}(F^S)} \right) \right) * 100 \qquad (14)$$

where$\text{Min}\,(C^S_{max})$ and $\text{Min}(F^S)$ represent best makespan and total flow time obtained after running all the algorithms respectively.

The relative percentage increase for makespan, total flow time separately and then for both is calculated for each test problem.See the obtained results in TABLE 1 to TABLE 3 respectively.Note that ACO-MOFS and MOACSA algorithms are compared indirectly. The right hand side of each table represents the performance of MOACSA while the left hand side represents the performance of ACO-MOFS algorithm in comparison with previously developed algorithms.

From the results shown in the TABLE 1 to 3 it is clear that our developed algorithm, ACO-MOFS performs best among others for makespan, total flow time and multiple objectives. Also this algorithm performs best among others from the time point of view. That is it achieves very good solutions in a very shorter time in comparison with previously developed algorithms. CPU times are shown in TABLE 4 in seconds.

V. Conclusion

The flow shop scheduling problem is one of the classes of scheduling problems that has attracted much attention in recent years. Most studies in the literature have studied on the single criterion form of the problem while sometimes there is a lot of interest to optimize the problem from two points of view i.e. to solve the multi objective form of the problem. Hence in this paper makespan and total flow time have been considered as two objectives to be optimized using an ACO-based heuristic algorithm. In order to verify the performance of the developed algorithm, this algorithm has been tested on the various benchmarks which have been tried to be optimized by different methods in the literature. The comparison of the results obtained by our developed method with those developed before, shows that, our proposed ACO-MOFS algorithm whether in condition of single criterion or multi criterion performs best among others. In addition this algorithm behaves quite appropriate from the time point of view i.e. high quality solutions are found within a very appropriate amount of time.

TABLE I
PERFORMANCE SOLUTIONS OF ALGORITHMS FOR MAKESPAN OBJECTIVE

N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	ACO-MOFS	N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	MOACSA
20*5	TA001	1.39	2.46	1.78	3.84	0	20*5	TA001	1.49	3.6	2.27	6.34	1.49
20*5	TA002	0.44	1.76	1.76	0.62	0	20*5	TA002	0	2.62	2.62	0.36	0.73
20*5	TA003	2.85	3.02	3.02	3.9	0	20*5	TA003	5.79	6.14	6.14	7.89	5.53
20*5	TA004	1.06	1.82	1.64	0.73	0	20*5	TA004	4.71	6.27	5.9	4.03	2.91
20*5	TA005	5.68	7.89	7.89	4.55	0	20*5	TA005	2.07	6.12	6.12	0	0.31
20*5	TA006	0.45	2.2	2.2	2.24	0	20*5	TA006	0	3.47	3.47	3.55	0.57
20*5	TA007	1.08	2.51	1.47	3.59	0	20*5	TA007	3.28	6.17	4.06	8.36	1.88
20*5	TA008	0.99	5.63	5.63	3.25	0	20*5	TA008	1.74	10.99	10.99	6.25	0
20*5	TA009	1.66	4.62	4.62	2.77	0	20*5	TA009	0.31	6.06	6.06	2.46	0
20*5	TA010	1.91	6.39	3.04	1.74	0	20*5	TA010	1.36	10.09	3.56	1.02	0
20*10	TA011	1.87	2.98	2.25	0.03	0	20*10	TA011	6.16	8.44	6.94	2.39	0.66
20*10	TA012	0.48	1.21	1.63	4	0	20*10	TA012	2.34	3.83	4.69	9.49	0
20*10	TA013	1.5	5.92	5.89	0.69	0	20*10	TA013	5.59	14.65	14.59	3.92	0
20*10	TA014	1.48	3.63	3.26	1.55	0	20*10	TA014	2.68	6.98	6.24	2.82	0
20*10	TA015	5.3	0	0.93	1	0.87	20*10	TA015	15.57	4.5	6.44	6.58	0
20*10	TA016	1.5	3.48	3.26	0	0.54	20*10	TA016	4.61	8.63	8.18	1.56	1.49
20*10	TA017	2.71	3.7	3.84	3.64	0	20*10	TA017	0.95	2.85	3.1	2.72	0.7
20*10	TA018	0.75	1.62	1.62	6	0	20*10	TA018	2.19	3.95	3.95	12.78	0
20*10	TA019	2.09	4.24	3.68	0.56	0	20*10	TA019	5.74	10.1	8.97	2.63	1.02
20*10	TA020	1.02	3.57	2.46	3.57	0	20*10	TA020	1.45	6.52	4.3	6.52	0
20*20	TA021	1.47	2.46	2.6	3.93	0	20*20	TA021	2.51	4.49	4.77	7.41	0
20*20	TA022	4.13	4.13	6.19	0	1.06	20*20	TA022	11.96	11.96	16.22	3.42	2.56
20*20	TA023	0.4	0.62	2.14	0.17	0	20*20	TA023	0.46	0.91	3.94	0	4.31
20*20	TA024	5.2	7.91	8.54	3.14	0	20*20	TA024	11.41	16.88	18.14	7.25	0.3
20*20	TA025	0	0.68	1.51	0.14	1.1	20*20	TA025	0	1.36	3.02	0.29	2.19
20*20	TA026	2.74	4.48	4.93	0.87	0	20*20	TA026	9.04	12.62	13.56	5.16	0
20*20	TA027	6.3	8.14	7.84	0	5.15	20*20	TA027	12.6	16.28	15.69	0	9.38
20*20	TA028	1.66	1.84	1.82	0.69	0	20*20	TA028	5.12	5.5	5.46	3.16	3.16

TABLE II
PERFORMANCE SOLUTIONS OF ALGORITHMS FOR TOTAL FLOW TIME OBJECTIVE

N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	ACO-MOFS	N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	MOACSA
20*5	TA001	2.26	1.8	1.96	5.63	0	20*5	TA001	1.16	0.28	0.59	7.7	0.96
20*5	TA002	4.44	0.82	0.82	10.12	0	20*5	TA002	7.42	0.27	0.27	18.64	0
20*5	TA003	2.25	2.03	2.03	9.19	0	20*5	TA003	2.94	2.5	2.5	16.6	0
20*5	TA004	2.62	1.59	1.6	5.73	0	20*5	TA004	3.31	1.28	1.3	9.41	0
20*5	TA005	7.01	2.7	2.73	9.04	0	20*5	TA005	8.78	0.55	0.61	12.66	0.38
20*5	TA006	3.44	1.65	1.74	7.98	0	20*5	TA006	3.47	0	0.18	12.26	0.24
20*5	TA007	2.33	1.38	1.39	8.63	0	20*5	TA007	2.67	0.79	0.82	15.02	0
20*5	TA008	2.86	2.36	2.4	7.67	0	20*5	TA008	2.03	1.06	1.13	11.31	0
20*5	TA009	4.99	1.56	1.56	4.07	0	20*5	TA009	7.91	1.17	1.17	6.09	0
20*5	TA010	2.83	1.01	1.12	6.49	0	20*5	TA010	3.57	0	0.21	10.74	0.33
20*10	TA011	3.18	2.64	2.74	4.83	0	20*10	TA011	5.29	4.23	4.43	8.56	0
20*10	TA012	1.41	0.41	0.5	8.13	0	20*10	TA012	2.31	0.31	0.5	15.68	0.43
20*10	TA013	5.31	2.16	2.24	4.68	0	20*10	TA013	8.22	2.07	2.22	7	0.3

20*10	TA014	1.98	1.12	1.27	5.06	0		20*10	TA014	3.51	1.81	2.1	9.65	0
20*10	TA015	2.46	4.51	1.72	5.48	0		20*10	TA015	4.09	8.15	2.63	10.08	0.41
20*10	TA016	2.07	1.25	1.5	4.78	0		20*10	TA016	2.31	0.71	1.2	7.64	0.44
20*10	TA017	3.89	2.35	3.12	6.79	0		20*10	TA017	2.94	0	1.46	8.48	0.45
20*10	TA018	2.95	1.39	1.61	8.82	0		20*10	TA018	4.81	1.71	2.15	16.43	0
20*10	TA019	2.59	0.71	0.78	4.97	0		20*10	TA019	4.46	0.73	0.86	9.19	0
20*10	TA020	0.93	0.28	0.36	8.44	0		20*10	TA020	1.29	0	0.15	16.22	0.6
20*20	TA021	3.36	1.7	2.07	7.24	0		20*20	TA021	5.59	2.31	3.05	13.28	0
20*20	TA022	2.46	2.46	4.06	5.8	0		20*20	TA022	4.46	4.46	7.65	11.12	0
20*20	TA023	1	0	0.78	3.42	1.48		20*20	TA023	1.99	0	1.55	6.84	4.01
20*20	TA024	3.46	6.13	6.66	8.67	0		20*20	TA024	5.04	10.28	11.33	15.28	0.05
20*20	TA025	2.76	0	0.73	3.14	2.04		20*20	TA025	5.52	0	1.46	6.29	5.89
20*20	TA026	6.06	3.05	3.58	6.72	0		20*20	TA026	11.51	5.53	6.58	12.81	0
20*20	TA027	1.74	0.26	0.38	4.36	0		20*20	TA027	3.91	0.93	1.18	9.16	0
20*20	TA028	2.23	2.1	2.13	3.03	0		20*20	TA028	1.48	1.22	1.28	3.02	0

TABLE III
PERFORMANCE SOLUTIONS OF ALGORITHMS FOR MULTIPLE OBJECTIVES

N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	ACO-MOFS		N*M	PROBLEM NUMBER	HAMC1	HAMC2	HAMC3	CR(MC)	MOACSA
20*5	TA001	3.65	4.26	3.74	9.47	0		20*5	TA001	1.19	0.56	0.73	7.58	1.53
20*5	TA002	4.88	2.58	2.58	10.74	0		20*5	TA002	6.72	0.38	0.38	17.04	0
20*5	TA003	5.1	5.05	5.05	13.09	0		20*5	TA003	1.95	1.57	1.57	14.56	0.05
20*5	TA004	3.68	3.41	3.24	6.46	0		20*5	TA004	3.1	1.36	1.35	8.65	0
20*5	TA005	12.69	10.59	10.62	13.59	0		20*5	TA005	7.99	0.85	0.9	11.35	0.67
20*5	TA006	3.89	3.85	3.94	10.22	0		20*5	TA006	2.87	0	0.16	11.19	0.12
20*5	TA007	3.41	3.89	2.86	12.22	0		20*5	TA007	2.43	0.96	0.81	14.14	0
20*5	TA008	3.85	7.99	8.03	10.92	0		20*5	TA008	1.76	1.64	1.7	10.64	0
20*5	TA009	6.65	6.18	6.18	6.84	0		20*5	TA009	6.79	1.1	1.1	5.3	0
20*5	TA010	4.74	7.4	4.16	8.23	0		20*5	TA010	3.04	0.49	0.15	9.57	0
20*10	TA011	5.05	5.62	4.99	4.86	0		20*10	TA011	5.18	4.36	4.44	7.94	0
20*10	TA012	1.89	1.62	2.13	12.13	0		20*10	TA012	1.74	0	0.23	14.59	0.19
20*10	TA013	6.81	8.08	8.13	5.37	0		20*10	TA013	7.45	2.41	2.55	6.2	0
20*10	TA014	3.46	4.75	4.53	6.61	0		20*10	TA014	3.07	1.8	2.02	8.75	0
20*10	TA015	7.76	4.51	2.65	6.48	0.87		20*10	TA015	4.66	7.61	2.65	9.54	0.33
20*10	TA016	3.57	4.73	4.76	4.78	0.54		20*10	TA016	2.35	1.17	1.58	7.06	0.4
20*10	TA017	6.6	6.05	6.96	10.43	0		20*10	TA017	2.56	0	1.37	7.8	0.38
20*10	TA018	3.7	3.01	3.23	14.82	0		20*10	TA018	4.15	1.41	1.82	15.64	0
20*10	TA019	4.68	4.95	4.46	5.53	0		20*10	TA019	4.11	1	1.04	8.23	0
20*10	TA020	1.95	3.85	2.82	12.01	0		20*10	TA020	0.84	0.03	0	14.97	0.37
20*20	TA021	4.83	4.16	4.67	11.17	0		20*20	TA021	5.31	2.38	3.08	12.8	0
20*20	TA022	6.59	6.59	10.25	5.8	1.06		20*20	TA022	4.64	4.64	7.88	10.28	0
20*20	TA023	1.4	0.62	2.92	3.59	1.48		20*20	TA023	1.83	0	1.65	6.31	4.27
20*20	TA024	8.66	14.04	15.2	11.81	0		20*20	TA024	5.27	10.52	11.58	14.53	0.35
20*20	TA025	2.76	0.68	2.24	3.28	3.14		20*20	TA025	5.04	0	1.47	5.78	5.69
20*20	TA026	8.8	7.53	8.51	7.59	0		20*20	TA026	11.3	5.96	7	12.26	0
20*20	TA027	8.04	8.4	8.22	4.36	5.15		20*20	TA027	3.73	1.18	1.37	7.85	0
20*20	TA028	3.89	3.94	3.95	3.72	0		20*20	TA028	1.5	1.29	1.34	2.81	0

TABLE IV
CPU TIMES

n	m	Problem number	CPU time	
			ACO-MOFS	MOACSA
20	5	Ta001-010	2.5	4
20	10	Ta011-020	4	6
20	20	TA021-28	7.5	9

REFERENCES

[1] S. M. Johnson.Optimal two- and three-stage production schedules with setup times included,*Naval Research Logistics Quarterly*,Vol. 1, pp 61-68, 1954.

[2] M. RGarey, D. S. Johnson, RSethi. The complexity of flowshop and jobshop scheduling, *Mathematics of Operations Research*,vol. 1, pp.117–129, 1976.

[3] JBlazewicz, EPesch, MSterna, FWerner. Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due date, *Computers & Operations Research*,vol. 35, no. 2, pp. 574-599, 2008.

[4] G.LZobolas, C.DTarantilis, GLoannou. Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm, *Computers & Operations Research,*Vol.36, no. 4, pp. 1249-1267, 2009.

[5] X Li, M. FBaki, Y. PAneja. Flow shop scheduling to minimize the total completion time with a permanently present operator: Models and ant colony optimization metaheuristic Original Research Article, *Computers & Operations Research,*vol. 38, no. 1, pp. 152-164, 2011.
A. C Andreas, CNearchou. A novel metaheuristic approach for the flow shop scheduling problem. *Engineering Applications of Artificial Intelligence,* vol. 17, no. 3, pp. 289-300, 2004.

[6] NSalmasi, RLogendran, M. R. Skandari. Total flow time minimization in a flowshop sequence-dependent group scheduling problem, *Computers & Operations Research,*vol. 37, no. 1, pp. 199-212, 2010.

[7] D. G. Dannenbring. An evaluation of flowshop sequencing heuristics. *Management Science,*vol. 23, no. 11, pp. 1174–1182, 1977.

[8] K. C. Ying, and C. J. Liao, Ant colony system for permutation flow-shop sequencing, *Computers & Operations Research,*vol. 31, no. 5, pp. 791-801, 2004

[9] RRuiz, CMaroto, JAlcaraz. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics, *European Journal of Operational Research,*vol. 165, no. 1, pp. 34-54, 2005

[10] AJaniak, EKozan, MLichtenstein, COğuz, Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion, *International Journal of Production Economics*,vol. 105, no. 2, pp. 407-424, 2007.

[11] ENowicki, CSmutnicki. A fast tabu search algorithm for the permutation flow-shop problem, *European Journal of Operational Research,*vol. 91, no. 1, pp. 160-175, 1997.

[12] V. R. Neppalli, C. L. Chen, J. N. D.Gupta. Genetic algorithms for the two stage bicriteriaflowshop problem. *European Journal of Operational Research,*vol. 95, no. 2, pp. 356–373, 1996.

[13] J. M. Wilson. Alternative formulations of a flowshop scheduling problem, *Journal of Operations Research Society,*vol. 40, no. 4, pp. 395–399, 1989.

[14] Nagar A, Haddock, J, Heragu S. S. A branch-and-bound approach for a two-machine flowshop scheduling problem. *Journal of the Operational Research Society,*vol. 46, no. 6, pp. 721–734, 1995.

[15] W. C. Yeh. A memetic algorithm for the n/2/flow shop/αF+βC_max scheduling problem. *The International Journal of Advanced Manufacturing Technology,*vol. 20. No. 6, pp. 464–473, 2002.

[16] CRajendran. Heuristics for scheduling in flowshop with multiple objectives. *European Journal of Operational Research,*vol. 82, pp. 540–555, 1995.

[17] J. M. Framinan,RLeisten, RRuiz-Usano. Efficient heuristics for flowshop sequencing with objectives of makespan and flowtime minimization. *European Journal of Operational Research,* vol. 141, pp. 561–571, 1996.

[18] DRavindran, ANoorulHaq, S. J. Selvakuar, RSivaraman. Flow shop scheduling with multiple objective of minimizing makespan and total flow time. *International Journal of Advanced Manufacturing Technology,*vol. 25, pp. 1007–1012, 2005.

[19] BYagmahan, M. M. Yenisey. A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications,*vol. 37, pp. 1361–1368, 2010.

[20] J. C. Ho, Y. L. Chang. A new heuristic for the n-job, m-machine flow shop problem. *European Journal of Operational Research,*vol. 52, pp. 194–202, 1991.

[21] C. Rajendran. A heuristic algorithm for scheduling in flowshop and flowline-based manufacturing cell with multi-criteria. *International Journal of Production Research,*vol. 32, pp. 2541–2558, 1994.

[22] C. Sridhar, C. Rajendran. Scheduling in flowshop and cellular manufacturing systems with multiple objectives: A genetic algorithmic approach. *Production Planning and Control,*vol. 7, pp. 374–382, 1996.

[23] B. Yagmahan, M. Yenisey. M. Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering,*vol. 54, pp. 411–420, 2008.

[24] M. Pinedo. Scheduling: theory, algorithms and systems. 2nd ed, Englewood Cliffs, NJ: Prentice-Hall; 2002.

[25] M. Dorigo,V. Maniezzo, A. Colorni. Positive feedback as a search strategy,*Technical report 91-016*, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1991.

[26] M. Dorigo. Optimization, Learning and Natural Algorithms [in Italian]. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.

[27] B. Bullnheimer,R. F.Hartl,C. Strauss. A new rank-based version of the Ant System: A computational study. *Central European Journal for Operations Research and Economics*,vol. 7, no. 1, pp. 25–38, 1999.

[28] T. Stutzle, H. H. Hoos. The MAX-MIN Ant System and local search for the traveling salesman problem. In T. Back, Z. Michalewicz, & X. Yao (Eds.), *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation* (ICEC'97) (pp. 309–314), Piscataway, NJ, IEEE Press.

[29] M. Dorigo, L. M. Gambardella. Ant colonies for the traveling salesman problem. *BioSystems*,vol. 43, no. 2, pp. 73–81, 1997.

[30] M. Widmer, A. Hertz. A new heuristic method for the flowshop sequencing problem. *European Journal of Operational Research,*vol. 41, pp. 186–193, 1989.

[31] M. Dorigo, T. Stutzle. Ant colony optimization, *Massachusetts Institute of Technology*,2004, pp. 31-32.

[32] www.lifl.fr/~liefooga/benchmarks/benchmarks/index.html.