Journal of Quality Engineering and Production Optimization
Vol. 1, No. 1, PP. 1-10, 2015

Flow shop Scheduling Problem with Maintenance Coordination: a New Approach

Mustafa Khatami and Seyed Hessameddin Zegordi
Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran

***Corresponding Author:**Seyed Hessameddin Zegordi (E-mail: zegordi@modares.ac.ir)

Abstract- *This study investigates the coordination of production scheduling and maintenance planning in the flow shop scheduling environment. The problem is considered in a bi-objective form, minimizing the makespan as the production scheduling criterion and minimizing the system unavailability as the maintenance planning criterion. The time interval between consecutive maintenance activities as well as the number of maintenance activities on each machine are assumed to be non-fixed. A mixed integer programming formulation of the problem is presented. A special case of the problem, named as single server maintenance is also studied. Then, a bi-objective ant colony system algorithm is presented to solve the problem in focus. To obtain the appropriate components of the proposed algorithm, two sets of experiments are provided. Firstly, experiments are carried out to select the suitable heuristic method to build the heuristic information part of the algorithm between CDS and NEH. Secondly, experiments are reported to select the local search algorithm between iterated local search and adjacent pair-wise interchange. At last, experiments are generated to evaluate the performance of the proposed algorithm, comparing it to the results of an exhaustive search algorithm.*

Keywords: *Flow shop scheduling, Preventive maintenance, Coordination, Non-fixed time interval, Ant colony system*

I. INTRODUCTION

In classical scheduling problems, machines are assumed to be available through the whole planning horizon, although this assumption may not be true in many practical environments, as machines may become unavailable during certain periods of planning horizon. Preventive maintenance is the main cause of unavailability of machines. There are some researches relaxing this unrealistic assumption, mainly the approach called machine scheduling with availability constraints where the number of preventive maintenance periods and their intervals are fixed and known in advance. In this approach, the maintenance periods are incorporated into the constraints of the problem, with no change in the performance measure. In fact, the constraints are formulated in a way to plan the jobs in the available periods of time.

As an early result for the availability constraint approach, Lee [1] showed that two machine flow shop problem with preemptive jobs becomes NP-hard, if there is a single unavailable period (hole) on one machine. Most of the papers in this area are investigated with two machines in the system. The problem is mostly considered with a limited number of maintenance activities on machines. Ng and Kovalyov [2] presented an FPTAS for the problem with one unavailable period and showed that two cases of the problem are equivalent to similar partition type problems. The same problem was also studied by Breit [3] and an improved approximation algorithm was presented. Allaoui et al. [4] presented the conditions under which the Johnson algorithm results in the optimum solution for the problem. Blazevicz et al. [5] investigated the problem with more than a hole and presented constructive and local search heuristics for the problem. Kubzin et al. [6] presented approximation algorithms for the problem with several holes on the first machine. Yang et al. [7] studied the problem with a separated maintenance constraint. The concept of semi-resumable jobs was incorporated into the problem in Lee [8]. In this approach, the problem is rarely considered with more than two machines in the system. Aggoune [9] presented a heuristic algorithm based on genetic algorithm and tabu search for the problem with non-preemptive jobs. A heuristic algorithm based on the geometric approach was also developed for the problem in Aggoune and Portmann [10]. Choi et al. [11] considered the problem with m machines in the system for very restricted conditions. They investigated the problem in the ordered flow shop scheduling environment. Schmidt

Manuscript Received: 2 June 2014 and revised: 23 July 2014 *ISSN: 2423-3781*
Accepted: 6 Jan 2015

[12] and recently Ma et al. [13] have provided detailed reviews on the problem with availability constraints. The drawback of this approach is that the maintenance planning is always advantageous to the production scheduling. Advantaging maintenance activities to the production jobs may result in systems with poor performance, as the system productivity is highly influenced with both the production and the maintenance decisions. In addition, the production and the maintenance activities are highly-interdependent and ignoring this may result in unsatisfied demand or machine breakdowns [14]. To overcome this drawback, Berrichi et al. [14] proposed the joint production and scheduling approach, in which the simultaneous scheduling of both production and maintenance activities was presented.

The main feature of the so-called joint approach is the simultaneous scheduling of production and maintenance activities. Berrichi et al. [14] presented the joint model in the parallel machine environment, and they formulated it in a bi-objective form to seek for the optimal Pareto front. Minimization of the makespan and minimization of the system unavailability were considered as the two objective functions in their study. The bi-objective form of the problem leads into the similar priority of production and maintenance activities. In the joint production and maintenance scheduling approach, the periodic preventive maintenance is considered in which the period between two consecutive maintenance activities is fixed. The maintenance problem is to find the best maintenance period for each machine in a way to minimize the system unavailability. Moradi and Zandieh [15] investigated the same problem and presented a similarity based genetic algorithm for the problem. The same problem was also investigated by Berrichi et al. [16]. They developed a bi-objective ant colony optimization algorithm to tackle the problem. The model was considered in the flexible job shop scheduling environment by Moradi et al. [17]. Moreover, the model with the minimization of total tardiness as the production scheduling criterion was studied by Berrichi and Yalaoui [18]. The problem with multiple preventive maintenance services was investigated by Mokhtari et al. [19]. For more recent contributions on this area, we refer to Wang and Liu [20] and Cui et al. [21]. In the joint approach, the fixed time intervals between consecutive maintenance activities seem to be problematic. In other words, this assumption may result in the intervention of the production and the maintenance activities. In this paper, we present a new approach to avoid these interventions. In addition, we formulate the model in the flow shop scheduling environment. There is no work investigating the joint model in the flow shop scheduling environment.

In this paper, a new approach is proposed to overcome the problem of simultaneous production and maintenance scheduling in the flow shop scheduling environment. In the proposed approach, the time intervals between maintenance activities as well as their number are decision variables. The problem is formulated in a bi-objective form and the model seeks for the optimal Pareto front. For the production scheduling part, makespan i.e., the finishing time of the last job in order is the performance measure. The performance measure for the maintenance part is the system unavailability. A bi-objective ant colony system approach is presented to solve the problem. To evaluate the efficiency of the proposed algorithm, some computational experiments are carried out using well-known Taillard's benchmark.

The remainder of the paper is organized as follows: Section 2 describes the problem definition and formulation. In Section 3, the proposed bi-objective ant colony system is presented. Results of computational experiments are presented and discussed in Section 4. In Section 5, concluding remarks are provided.

II. PROBLEM DEFINITION AND FORMULATION

Concerning production aspect of the problem, flow shop environment is considered. A flow shop scheduling problem deals with a set of n jobs, $J = \{1, 2, \ldots, n\}$ to be processed on a set of m machines, $R = \{1, 2, \ldots, m\}$. The processing order on the m machines is the same for all N jobs, i.e., only permutation schedules are allowed. Each job j consists of a set of m operations, $O_{j1}, O_{j2}, \ldots, O_{jm}$ to be processed on machines. The processing time of job j on machine r is p_{rj}. All jobs are assumed to be available at time zero, and preemption of jobs is not allowed, i.e., once the execution of an operation is started, it can be interrupted neither by other operations nor by maintenance activities. Every machine processes only one action (e.g., job operation or maintenance activity) at a time. The problem is to find the best permutation of jobs, regarding the minimization of the makespan as the objective function. We define C_j as the completion time of the job j. The makespan (C_{max}) is the completion time of the last job in sequence and can be defined by relation (1):

$$C_{\max} = \max_{j=1,2,\ldots,n} C_j \quad (1)$$

Concerning maintenance aspect of the problem, preventive maintenance is taken into account. Preventive maintenance strategies aim to decrease the probability of failure of the system (i.e., increase the availability of the system). In this paper, the time intervals between two consecutive maintenance activities as well as the number of the activities are not fixed. In other words, the time intervals are assumed to be decision variables. The maintenance problem is to assign maintenance activities into the schedule, in a way to minimize the system unavailability. The availability is defined as "the probability that a system or a component is performing its required function at a given point in time or over a stated period of time when operated and maintained in a prescribed manner" [22]. The availability of a machine r at a given point in time t is defined as relation (2):

$$A(t) = \mathrm{P}(r \text{ is operating at time } t) \tag{2}$$

Unavailability is defined as the opposite of the availability. Therefore, the system unavailability can be defined as (3):

$$\overline{A}(t) = 1 - A(t) \tag{3}$$

The availability of a machine M_r depends on its failure rate λ_r and its repair rate μ_r. Failure rates and repair rates of all machines are assumed to be constant. Moreover, maintenance activities are supposed to restore the machine to as good as new condition. Taking into account these assumptions, the availability of a machine M_r at time t is calculated by relation (4) [22]:

$$A_r(t) = \frac{\mu_r}{\lambda_r + \mu_r} + \frac{\lambda_r}{\lambda_r + \mu_r} \exp(-(\lambda_r + \mu_r)(t - T_r)) \tag{4}$$

where T_r is the completion time of the last performed maintenance activity on machine r. From the initial instant with no performed maintenance activity, $T_r = 0$. The system availability is due to the system structure. For m independent serial components, each component having an availability function $A_r(t)$, the system availability $A_s(t)$ at time t is given by relation (5) [22]:

$$A_s(t) = \prod_{r=1}^{m} A_r(t) \tag{5}$$

Consequently, the system unavailability is:

$$\overline{A_s}(t) = 1 - \prod_{r=1}^{m} A_r(t) \tag{6}$$

Minimization of the system unavailability is the objective function for this aspect of the problem.

Two aspects can be regarded for the problem, production scheduling and maintenance planning; hence, two decisions must be taken simultaneously. The production decision is to find the best permutation of the jobs to be processed on machines, minimizing the makespan. The maintenance problem is to assign the maintenance activities into the schedule, minimizing the system unavailability, while the number of the maintenance activities on each machine is not fixed. Although the two objective functions are conflicting, the system productivity is concerned with both of them. Assigning maintenance activities into a schedule may increase the makespan but decrease the system unavailability. On the contrary, performing less maintenance activities leads into decrease in makespan and increase in system unavailability.

Let $T = \{0, t_1, t_2, \ldots, t_s, C_{max}\}$ where $t_1, t_2, \ldots, t_s$ are the starting times of the maintenance activities on all machines. As machines are assumed to become "as good as new" after performing a maintenance activity, and the unavailability is an increasing function in each interval $[t_i, t_{i+1}]$, $i = 0, \ldots, s$, with $t_0 = 0$ and $t_{s+1} = C_{max}$, the system unavailability is only computed at the times $t_1, t_2, \ldots, t_{s+1}$. The notations used to formulate the problem are presented in Table 1.

To formulate the problem, the model proposed by Wilson [23] is used. According to Tseng et al. [24], Wilson's model is the second best mixed-integer linear programming formulation for the permutation flow shop scheduling problem. The Wilson's model uses the classic assignment problem constraints. Let Z_{jk} be the binary integer variables for the assignment of jobs in the sequence positions. If job j is assigned to sequence position k, Z_{jk} is 1; otherwise, it is 0. Moreover, let B_{rk} be the integer variables denoting the start time of jobs:

$$B_{rk} : \text{start time of job in sequence position } k \text{ on machine } r \tag{7}$$

TABLE I. TYPE SIZES AND APPEARANCE

Notations	Descriptions
j	Index of jobs
r	Index of machines
k	Index of positions
J	Set of jobs
R	Set of machines
K	Set of sequence positions
n	Total number of jobs
m	Total number of machines
p_{rj}	Process time of job j on machine r

To incorporate the maintenance activities into the schedule, a binary integer variable X_{rk} is defined that is equal to 1 if a maintenance activity is performed after position k on machine r, and 0 otherwise.

By defining this variable, maintenance activities are allowed to perform all over the schedule, between the production jobs. d_r is the duration of a maintenance activity performing on machine r. The maximum number of maintenance activities performing on machine r is shown by P^r_{max}. The whole model is represented as below:

$$F_1 = \min\{C_{\max}\} = B_{mn} + \sum_{j=1}^{n}(p_{mj} Z_{jn}) + X_{mn} d_m \tag{8}$$

$$F_2 = \min\{\max_{t\in T} \overline{A_s}(t)\} \tag{9}$$

subject to:

$$\sum_{k=1}^{n} Z_{jk} = 1 \quad (1 \le j \le n) \tag{10}$$

$$\sum_{j=1}^{n} Z_{jk} = 1 \quad (1 \le k \le n) \tag{11}$$

$$B_{11} = 0 \tag{12}$$

$$B_{1k} + \sum_{j=1}^{n}(p_{1j} Z_{jk}) + X_{1k} t_1 = B_{1,k+1} \quad (1 \le k \le n-1) \tag{13}$$

$$B_{r1} + \sum_{j=1}^{n}(p_{rj} Z_{j1}) = B_{r+1,1} \quad (1 \le r \le m-1) \tag{14}$$

$$B_{rk} + \sum_{j=1}^{n}(p_{rj} Z_{jk}) \le B_{r+1,k} \quad (1 \le r \le m-1; 2 \le k \le n) \tag{15}$$

$$B_{rk} + \sum_{j=1}^{n}(p_{rj} Z_{jk}) + X_{rk} d_r \le B_{r,k+1} \quad (2 \le r \le m; 1 \le k \le n-1) \tag{16}$$

$$\sum_{k=1}^{n} X_{rk} \le P^r_{\max} \quad (1 \le r \le m) \tag{17}$$

Equations (8) and (9) indicate the two objective functions. Equations (10) and (11) are the classical assignment problem constraints, while (10) insures that each job is assigned to just one position and (11) insures that each position is filled with only one job. Constraints (12), (13), and (14) ensure that there is no idle time on machine 1, and job 1 is processed on all M machines with no delay. Constraint (15) ensures that the starting time of each job on machine $r + 1$ is no earlier than its finishing time on machine r. Constraint (16) ensures that the job in the sequence position $k + 1$ does not start on machine r until the job in position k in the sequence has completed its processing on that machine, and the maintenance activity on position k is performed, if it is planned ($X_{rk} = 1$). At last, equation (17) ensures that the number of maintenance activities performing on machine r is no more than P^r_{max}.

Single server maintenance: Consider a setting in which the maintenance activities are performed by a single server. In

other words, no two maintenance activities can be performed simultaneously. The assumption may be caused by a limitation in the human resource or/and the equipment, that may be occurred in many practical situations. We call such a setting as single server maintenance. For the single server maintenance situation, some modifications must be accomplished in the model previously described. In this regard, a new constraint must be added to the model that prevents planning maintenance activities with time interventions. If the distance between the starting time of each two maintenance activities is bigger than the maximum of the processing time of the respective activities, no intervention occurs. Equation (18) indicates the discussed concept, considering two maintenance activities on machines r and q, planned after sequence positions k and l, respectively.

$$|(B_{rk}+\sum_{j=1}^{n}p_{jk}Z_{jk})*X_{rk}-(B_{ql}+\sum_{j=1}^{n}p_{jl}Z_{jl})*X_{ql}| \geq \max(d_r,d_q)*X_{rk}X_{ql} \qquad (1\leq r,q\leq m;1\leq k,l\leq n) \qquad (18)$$

III. SOLUTION APPROACH

A bi-objective ant colony system algorithm is proposed to tackle the problem. Ant colony optimization, first proposed by Dorigo [25], is a swarm intelligence meta-heuristic algorithm, inspired by the cooperating behaviour of real ants. A variation of the basic algorithm is the ant colony system (ACS), proposed by Dorigo and Gambardella [26]. ACS, incorporating an exploration-exploitation mechanism, was first applied to traveling salesman problem. The algorithm is also applied to solve various scheduling problems [27-29]. Neto and Filho [30] have recently prepared a literature review on ant colony optimization algorithms applied to scheduling problems. The proposed bi-objective ACS algorithm employs two colonies of ants, a colony for the production optimization and the other one for the maintenance optimization. The two colonies share information during the solution construction and mutually cooperate to build complete solutions. The general idea of the proposed algorithm is inspired from both the PACO algorithm proposed by Berrichi et al. [16] and the ACO algorithm proposed by Yagmahan and Yenisey [29].

The whole procedure of the applied algorithm is as follows: At first, the parameters, the heuristic information and the pheromone trails are started. In the next step, iteratively, two colonies of ants build their complete solutions. Production ants utilize the state transition rule to select the next job in sequence and maintenance ants do the same to select the maintenance position, respectively. During the solution construction, local pheromone updating rule is applied. The global pheromone-updating rule is applied when all ants have built their complete solutions. Until reaching the stopping criterion, the procedure is repeated. In the following sub-sections, the properties of the proposed algorithm will be explained.

A. Pheromone trails

The pheromone trails are initialized in the first step. For the production part, we define the pheromone trails based on the location of jobs in the sequence. In other words, τ_{jk} is the desirability of placing job j in the kth sequence position. For the maintenance part, we define τ_{rk} as the desirability of placing a maintenance activity after the job in sequence position k on machine r. The initial pheromone trails τ_0 are relatively small and the same for all paths.

B. Heuristic information

The heuristic information is initialized at the first step. For the production part, two scenarios are defined. In this regard, two CDS [31] and NEH [32] heuristics are used. CDS and NEH are well-known heuristics for solving the permutation flow shop scheduling problem [33]. The absolute distance of a job to its position in the best schedule obtained by CDS (NEH) algorithm is regarded as the heuristic information. In other words, if the job j in the CDS (NEH) best schedule is in position k, the heuristic information of placing this job in position v is calculated as $|v - k|$. The heuristic information is fixed during the algorithm for the production part. For the maintenance part, the heuristic information is calculated adaptively. With no maintenance activity in the schedule, the heuristic information increases by the distance to zero point. After placing a maintenance activity, the heuristic information is calculated by the distance to the zero point, as well as the position of maintenance activities. Greater distances result in greater heuristic information values.

C. State transition rule

For the production part, an ant s in job j selects job l to move, by applying the following state transition rule:

$$l = \begin{cases} \arg\max_{u \in J_s(j)} \{[\tau(j,u)]^{\alpha}.[\eta(j,u)]^{\beta}\} & if \quad q \le q_0 \\ L & o w . \end{cases} \tag{19}$$

where, τ (j, u) and η (j, u) are the amount of pheromone trail and the heuristic information value on edge (j, u), respectively. J_s (j) is the set of feasible jobs to be selected after job i. q is a random number uniformly distributed in [0, 1], and q_0 is a parameter that determines the relative importance of exploitation versus exploration ($0 \le q_0 \le 1$). α and β are the relative importance of pheromone trail and heuristic information ($\alpha > 0$; $\beta > 0$). L is a random variable selected according to the probability of that ant s chooses job l with larger p_s (j, l) to move from job j:

$$l = \begin{cases} \arg\max_{u \in J_s(j)} \{[\tau(j,u)]^{\alpha}.[\eta(j,u)]^{\beta}\} & if \quad q \le q_0 \\ L & o w . \end{cases} \tag{20}$$

when an ant in job j chooses a job l to move to, it samples a random number q. If $q \le q_0$, then the best job according to relation (18) is selected. Otherwise, the next job is selected according to relation (19). For the maintenance part, the same procedure is performed. The difference is that the maximum number of maintenance activities is not the same for all machines. Hence, for machine r, the procedure is stopped when it reaches P^r_{max}.

D. Local updating rule

While constructing a schedule, an ant decreases the pheromone trail level between selected jobs by applying the following local pheromone-updating rule:

$$\tau(j,u) = (1-\rho_{local}).\tau(j,u) + \rho_{local}.\tau_0 \tag{21}$$

where, τ_0 is the initial pheromone level and ρ_{local} is the local pheromone decay parameter ($0 < \rho_{local} < 1$). The parameters for the two parts may be different.

E. Global updating rule

Global pheromone updating rule is applied after all ants completed their schedules (both parts). Global updating provides a greater amount of pheromone trail between adjacent jobs (maintenance positions) of best Pareto front. The pheromone trail level is updated as follows:

$$\tau(j,u) = (1-\rho_{global}).\tau(j,u) + \rho_{global}.\Delta\tau(j,u) \tag{22}$$

where,

$$\Delta\tau(j,u) = \begin{cases} Q.(f_1)^{-1} & if \quad (j,l) \in best \ \ front \\ 0 & o w . \end{cases} \tag{23}$$

ρ_{global} is the global pheromone decay parameter ($0 < \rho_{global} < 1$) and f_1 is the objective function value of the schedule with best rank containing the (j, u) edge. As the local updating rule, the parameters may be different for the two parts.

F. Local search

To reinforce the performance of the proposed algorithm in the production part, a local search is incorporated into the algorithm. As soon as all the production ants have built their complete solutions, a local search procedure is performed. Two scenarios are considered in this purpose, adjacent pair-wise interchange method (API) and insert local search (ILS) method. In the API method, two adjacent jobs are swapped. In the ILS method, the job in sequence position k is removed and inserted in a new sequence position l.

G. Stopping criterion

The procedure is repeated until the maximum number of iterations is reached as a stopping criterion.

TABLE II: BEST VALUES OF PARAMETERS OF THE ALGORITHM

Parameter	Value in the production part	Value in the maintenance part
α	4	4
β	2	2
q_0	0.5	0.5
ρ	0.1	0.1
σ	0.3	0.3
τ_0	0.05	0.05

IV. COMPUTATIONAL RESULTS

Computational experiments are carried out to evaluate the performance of the proposed ACS algorithm. Experiments are performed in three directions. Firstly, the appropriate heuristic algorithm is selected to build the heuristic information of the algorithm. Afterwards, similar experiments are performed to select the appropriate local search method. Having identified the better heuristic algorithm and local search method, the performance of the proposed algorithm is evaluated in the last sub-section. For the production part of the problem in focus, the well-known Taillard benchmark [34] is used. Test problems with 5, 10 and 20 machines and 20, 50 and 100 jobs are executed. The maintenance characteristics of the problem are as follows: The failure rate and repair rate of all machines is fixed and equal ($\lambda_r = \lambda = 0.01$; $\mu_r = \mu = 0.05$). The duration of a maintenance activity is regarded as 20. Number of ants and iterations are 20 and 100, respectively. In the following, computational experiments carried out in two directions will be discussed. The ant colony algorithm depends on several parameters and so determining the optimal values of all parameters with a experimental design is not tractable. In fact, there are 12 parameters to be calibrated, 6 in the production part of the algorithm and 6 in the maintenance part of the algorithm. Therefore, the best values were obtained by a trial and error approach due to a set of potential values for each parameter. The list of the best value of each parameter is summarized in Table II.

A. CDS or NEH

Experiments are executed to select the suitable heuristic algorithm between CDS and NEH to construct the heuristic information part of the algorithm. Three metrics are taken into account: the number of Pareto solutions (N.P.S.) obtained by each algorithm, the C metric and the computational time of the heuristic information construction. The C metric, proposed by Zitzler [35], is a measure to differentiate two fronts. The value of C (A, B) states the percentage of solutions in B dominated by at least one solution of A. The C metric is calculated as:

$$C(A,B) = \frac{|\{b \in B \mid \exists a \in A : a \prec b\}|}{|B|} \tag{24}$$

Where $a \prec b$ means that a dominates b. The closer the value of C (A, B) is to 1, the better the front A is compared to B. As the C metric is not symmetric, it is necessary to calculate both C (A, B) and C (B, A) and A is better than B, if C (A, B) > C (B, A).

Table III represents the computational results, comparing the two CDS and NEH scenarios. Computational times are reported in seconds. The N.P.S. metric obtained by the algorithm using both CDS and NEH is not significantly different. Although, comparing the two algorithms under the C metric, the NEH scenario outperforms the CDS scenario with 93% confidence level. Nevertheless, the CDS heuristic is more robust than the NEH heuristic in terms of efficiency. The increasing trend of NEH's computational time by increasing the size of jobs is tangible.

B. API or ILS

In this sub-section, experiments are provided to select the appropriate local search algorithm. Table IV represents the computational results, comparing the two API and ILS scenarios. The N.P.S. metric obtained by the algorithm using both API and ILS is not significantly different. Comparing the two algorithms under the C metric, the API scenario outperforms the CDS scenario in all problem sizes. This is proved with 99% confidence level. In terms of efficiency, the performance of the two API and ILS scenarios is relatively the same.

TABLE III: COMPUTATIONAL RESULTS FOR SELECTING THE HEURISTIC INFORMATION ALGORITHM

$n \times m$	N.P.S.		C metric		Time	
	CDS	NEH	CDS	NEH	CDS	NEH
20 × 5	9.4	8.2	0.38	0.49	0.013	0.016
20 × 10	5.4	6.8	0.43	0.31	0.017	0.026
20 × 20	6.6	9.0	0.28	0.36	0.023	0.037
50 × 5	3.6	5.4	0.33	0.47	0.028	0.088
50 × 10	3.2	3.8	0.39	0.52	0.054	0.115
50 × 20	3.4	2.8	0.32	0.55	0.070	0.145
100 × 5	2.2	2.6	0.33	0.67	0.053	0.425
100 × 10	1.8	1.2	0.44	0.50	0.081	0.524
100 × 20	1.2	1.0	0.60	0.40	0.082	0.749

TABLE IV. COMPUTATIONAL RESULTS FOR SELECTING THE LOCAL SEARCH ALGORITHM

$n \times m$	N.P.S.		C metric		Time	
	API	ILS	API	ILS	API	ILS
20 × 5	8.6	8.2	0.78	0.16	0.029	0.035
20 × 10	8.0	5.4	0.49	0.25	0.036	0.042
20 × 20	6.5	12.0	0.76	0.13	0.054	0.056
50 × 5	8.0	5.7	0.59	0.42	0.095	0.099
50 × 10	7.0	7.4	0.51	0.19	0.134	0.161
50 × 20	6.3	8.2	0.73	0.24	0.205	0.235
100 × 5	5.1	5.4	0.54	0.21	0.255	0.350
100 × 10	7.8	6.3	0.76	0.18	0.390	0.501
100 × 20	7.9	8.6	0.52	0.33	0.715	0.803

C. Performance evaluation

In the two previous sub-sections, experiments were carried out to select the heuristic algorithm and the local search algorithm with higher performance. Therefore, the NEH algorithm is selected to build the heuristic information part of the algorithm and the API algorithm as the local search method. In this part, the performance of the proposed algorithm is evaluated. In this regard, an exhaustive search algorithm is coded and applied to the problem to find the best Pareto front. The search space of the problem immensely increases with the increase in the size of the problem. Therefore, some small sized problem instances are considered to evaluate the performance of the algorithm. Problems with 5 jobs and two machines in the system are considered under four different scenarios of maintenance activities.

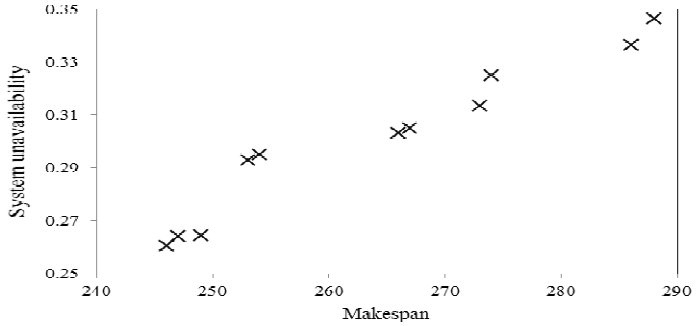

Fig. 1. Optimal Pareto front for a small sized problem instance

The search space of the problem for the four scenarios are equal to 4320, 11520, 11520 and 30720. In all of the four problem instances, the proposed ant colony algorithm found the same Pareto front that was resulted by the exhaustive search algorithm, i.e., optimal Pareto front. This indicates the effectiveness of the proposed algorithm. As an example, the optimal Pareto front of a problem in the fourth scenario is depicted in Fig. 1, gained by both the proposed ant colony algorithm and the exhaustive search method.

V CONCLUDING REMARKS

The problem of simultaneous production scheduling and maintenance planning in the flow shop scheduling environment was investigated in this paper. The concept of non-fixed time intervals between consecutive maintenance activities was introduced and the integer-programming model was presented. Moreover, a special case of the problem was developed. A bi-objective ant colony system was presented to tackle the problem, and some computational experiments were conducted to evaluate the performance of the proposed algorithm.

As a direction for future study, the model can be investigated in other scheduling environments and considering other performance measures.

REFERENCES

1. Lee C.Y. (1997)."Minimizing the makespan in the two machine flowshop scheduling problem with an availability constraint", *Operations Research Letters*, 20(3) pp. 129–139.
2. Ng C.T., & Kovalyov M.Y.(2004). "An FPTAS for scheduling a two-machine flowshop with one unavailability interval", *Naval Research Logistics*, 51(3) pp. 307-315.
3. Breit J.(2004). "An improved approximation algorithm for two-machine flow shop scheduling with an availability constraint", *Information Processing Letters*, vol. 90(6) pp. 273-278.
4. Allaoui H., Artiba A., Elmaghraby S.E., Riane F. (2006)."Scheduling of a two-machine flowshop with availability constraints on the first machine", *International Journal of Production Economics*, 99(1-) pp. 16–27.
5. Blazewicz J., Breit J., Formanowicz P., Kubiak W., & Schmidt, G.(2001). "Heuristic algorithms for the two-machine flowshop with limited machine availability". *OMEGA, The International Journal of Management Science*, 29(6) pp. 599-608.
6. Kubzin M.A., Potts C.N., and Strusevich V.A.(2009). "Approximation results for flow shop scheduling problems with machine availability constraints", *Computers & Operations Research*, 36(2) pp. 379-390.
7. Yang D.-L, Hsu C.-J, and Kuo W.-H.(2008). "A two-machine flowshop scheduling problem with a separated maintenance constraint", *Computers & Operations Research*, 35(3) pp. 876-883.
8. Lee C.-Y.(1999). "Two-machine flowshop scheduling with availability constraints", *European Journal of Operational Research*, 114(2) pp. 420-429.
9. Aggoune R.(2004). "Minimizing the makespan for the flow shop scheduling problem with availability constraints", *European Journal of Operational Research*, 153(3) pp. 534-543.
10. Aggoune R., & Portmann M.-C.(2006). "Flow shop scheduling problem with limited machine availability: A heuristic approach", *International Journal of Production Economics*, 99(1-2) pp. 4-15.
11. Choi B.-C., Lee K., Leung J.Y.-T., Pinedo M.L.(2010)."Flow shops with machine maintenance: Ordered and proportionate cases", *European Journal of Operational Research*, 207 (1) pp. 97-104.
12. Schmidt G.(2000) "Scheduling with limited machine availability", *European Journal of Operational Research*, 121(1) pp. 1-15.
13. Ma Y., Chu C., and Zuo C.(2010). "A survey of scheduling with deterministic machine availability constraints", *Computers & Industrial Engineering*, 58(2) pp. 199–211.
14. Berrichi A., Amodeo L., Yalaoui F., Chatelet E., & Mezghiche M.(2009). "Bi-objective optimization algorithms for joint production and maintenance scheduling: application to the parallel machine problem", *Journal of Intelligent Manufacturing*, vol. 20, pp. 389–400.

15. Moradi E., & Zandieh M.(2010). "Minimizing the makespan and the system unavailability in parallel machine scheduling problem: a similarity-based genetic algorithm", *Intelligent Journal of Advanced Manufacturing Technology*, vol. 51, pp. 829–840.

16. Berrichi A., Yalaoui F., Amodeo L., & Mezghiche M.(2010). "Bi-objective ant colony optimization approach to optimize production and maintenance scheduling", *Computers & Operations Research*, 37(9) pp. 1584–1596.

17. Moradi E., Fatemi Ghomi S.M.T., and Zandieh M.(2011). "Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem", *Expert Systems with Applications*, 38(6) pp. 7169–7178.

18. Berrichi A. and Yalaoui F.(2013). "Efficient bi-objective ant colony approach to minimize total tardiness and system unavailability for a parallel machine scheduling problem", *International Journal of Advanced Manufacturing Technology*, vol. 68, pp. 2295–2310.

19. Mokhtari H., Mozdgir A. & Nakhai Kamal Abadi I.(2012). "A reliability/availability approach to joint production and maintenance scheduling with multiple preventive maintenance services", *International Journal of Production Research*, 50(20) pp. 5906–5925.

20. Wang S. & Liu M.(2014). "Two-stage hybrid flow shop scheduling with preventive maintenance using multi-objective tabu search method", *International Journal of Production Research*, 52(5) pp. 1495-1508.

21. Cui, W.-W., Lu, Z., and Pan, E.(2014). "Integrated production scheduling and maintenance policy for robustness in a single machine", *Computers & Operations Research*, 47, pp. 81–91.

22. Ebeling C.E., An Introduction to Reliability and Maintainability Engineering, McGraw-Hill: USA, 1997.

23. Wilson J.M.(1989). "Alternative formulations of a flow-shop scheduling problem", *Journal of the Operational Research Society*, 40(4) pp. 395–9.

24. Tseng F.T., Stafford E.F. Jr., & Gupta J.N.D.(2004) "An empirical analysis of integer programming formulations for the permutation flowshop", *Omega*, 32(4) pp. 285–293.

25. Dorigo M., Optimization, Learning and Natural Algorithm. Ph.D. Thesis, DEI: Politecnico di Milano, 1992.

26. Dorigo M., and Gambardella L.M.(1997). "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, 1(1) pp. 53–66.

27. Ying K.C., and Liao C.J.(2003). "An ant colony system approach for scheduling problems", *Production Planning & Control*, 14(1), pp. 68–75.

28. Ying K., and Lin S., "Multi-heuristic desirability ant colony system heuristic for non-permutation flowshop scheduling problems", *International Journal of Advanced Manufacturing Technology*, vol. 33, pp. 793–802, 2007.

29. Yagmahan B., and Yenisey M.M.(2008). "Ant colony optimization for multi-objective flow shop scheduling problem", *Computers & Industrial Engineering*, 54(3) pp. 411–420.

30. Tavares Neto R.F., and Godinho Filho M.(2012). "Literature review regarding ant colony optimization applied to scheduling problems: guidelines for implementation and directions for future research", *Engineering Applications of Artificial Intelligence*, 26(1) pp. 150–161.

31. Campbell H.G., Dudek R.A., and Smith M.L. (1970). "A heuristic algorithm for the *n* job, *m* machine sequencing problem", *Management Science*,61(3) pp. B630–B637.

32. Nawaz M., Enscore Jr.E.E., and Ham I. (1983) ."A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem", *Omega*, 11(1) pp. 91–95.

33. Ruiz R., and Maroto C.(2005). "A comprehensive review and evaluation of permutation flowshop heuristics", *European Journal of Operational Research*, 165(2) pp. 479–94.

34. Taillard E. (1993). "Benchmarks for basic scheduling problems", *European Journal of Operational Research*, 64(2) pp. 278–85.

35. Zitzler E., Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications. Ph.D. Thesis, Swiss Federal Institute of Technology, 1999.