Int J Adv Manuf Technol (2014) 70:1579–1589
DOI 10.1007/s00170-013-5348-4

ORIGINAL ARTICLE

A scheduling model for serial jobs on parallel machines with different preventive maintenance (PM)

S. Nima Mirabedini · Hossein Iranmanesh

Received: 5 September 2012 / Accepted: 25 September 2013 / Published online: 25 October 2013

Abstract In factories during production, preventive maintenance (PM) scheduling is an important problem in preventing and predicting the failure of machines, and most other critical tasks. In this paper, we present a new method of PM scheduling in two modes for more precise and better machine maintenance, as pieces must be replaced or be repaired. Because of the importance of this problem, we define multi-objective functions including makespan, PM cost, variance tardiness, and variance cost; we also consider multi-parallel series machines that perform multiple jobs on each machine and an aid, the analytic network process, to weight these objectives and their alternatives. PM scheduling is an NP-hard problem, so we use a dynamic genetic algorithm (GA) (the probability of mutation and crossover is changed through the main GA) to solve our algorithm and present another heuristic model (particle swarm optimization) algorithm against which to compare the GA's answer. At the end, a numerical example shows that the presented method is very useful in implementing and maintaining machines and devices.

Keywords Scheduling · Reliability · Preventive maintenance · Multi-objective genetic algorithm

1 Introduction

All around the world, failure and deterioration may occur in systems and machines; so maintenance scheduling or planning should be considered for equipment. Preventive maintenance (PM) plays a significant role in the maintenance of equipment and machines and protects tools and pieces against failure. Many authors have presented different explanations for PM; in this paper, we describe it as maintenance of machines or equipment before any failure or deterioration occurs. PM is a schedule of planned maintenance that prevents apparatus from breakdown and failure. We can divide PM into two main maintenance groups "planned" and "condition-based." One difference between these two subgroups is the determination of maintenance time. As PM scheduling is an important problem, and machine failure and breakdown result in much expenditure, both costs and time, many writers have undertaken much research into this problem.

Much work has been undertaken on PM scheduling for a single machine, with minimal repair on machine failure [1] and on PM modeling minimizing total cost and maximizing the reliability of the system [2]. Sadfi et al. presented a new method and developed a modified algorithm maintenance support product team for a single-machine scheduling problem [3]. An important and traditional goal of that paper was to minimize the total completion time (makespan). Based on their algorithm, the worst-case ratio was 20/17 [4]. In their research, Bris et al. considered cost and availability as the system's criteria. They presented a mathematical model to find the best PM schedule and optimized it by including cost as the objective function and availability as the constraint by using a GA [5]. In other research, Shalaby et al. developed an optimization model for the PM scheduling of multicomponent and multi-state systems. They defined a sequence of PM activities as the decision variables and presented multi-objectives including summation of the PM, minimal repair, and downtime costs. In addition, they considered system reliability and minimum intervals between maintenance actions and crew availability all together as the constraints of their model [6]. Later, Yao et al. extended their previous model to be more general, and applied this expanded model in the production line of a semiconductor

S. N. Mirabedini (✉) · H. Iranmanesh
Department of Industrial Engineering, University of Tehran, Tehran, Iran
e-mail: n.mirabedini@ut.ac.ir

H. Iranmanesh
e-mail: Hiranmanesh@ut.ac.ir

manufacturing system, and showed its application via numerical examples [7]. Cassady and Kutanoglu also developed a similar integrated model to minimize the total expected weighted completion time of jobs [8]. Duarte et al. presented a model and an algorithm for the maintenance optimization of a system with series components [9]. Chelbi and Ait-Kadi considered a mathematical model for a repairable production unit, supplying inputs for a subsequent assembly line, operating according to a just in time configuration [10]. The decision variables, the buffer stock size, and the PM period length were obtained by minimizing the sum of the maintenance costs, the inventory holding costs, and the shortage costs. Nakagawa and Nakamura studied an entropy model with the application of a maintenance policy in which machine failure time satisfied a Weibull distribution [11]. Another excellent study in this area was by Tam et al., who developed three nonlinear optimization models as follows: the first model minimizing total cost subject to satisfy a required reliability, the second maximizing reliability for a given budget, and the last minimizing the expected total cost, including expected breakdown-outage costs and maintenance costs [12]. Prasad et al. used a GA approach to solve the multi-objective scheduling problems in a Kanban-controlled flow shop with intermediate buffer and transport constraints [13]. Alardhi et al. presented a binary integer linear programming model in order to find the best PM schedule in separated and linked cogeneration plants. Wu considered one policy in the maintenance scope for optimizing a number of parameters, such as number of contracts [14]. He presented a PM model, analyzing and discussing the special cases of both the PM policy and the bonus function [15].

Kenne et al. formulated an analytical model in which the determination of age-dependent production planning and age-PM is integrated. The objective of this paper is to minimize the overall cost functions, such as inventory holdings, lost sales, and preventive and corrective maintenance costs. However, the production decisions rather revolve around safety stock levels than scheduling job orders [16]. Although most of the relevant research has been about setting PM planning into the scheduling model, it usually considers production scheduling and PM scheduling as two independent problems, which prevent the planning model from being a real integrated model. So, the result may not be optimal. Hence, an improved scheduling model, explicitly integrating production scheduling and PM planning, has been presented. In another article, Qi [17] studied a machine scheduling model that considered machine maintenance and job scheduling simultaneously, and then developed the worst-case bounds of the shortest processing time and earliest due date schedules. There, he defined two objective functions and minimized total completion time and maximum lateness. Mosheiov and Sarighas considered minimum total weighted completion time as the objective function in a single-machine scheduling problem. They proved that this process is NP-hard and proposed a pseudo-polynomial dynamic programming algorithm and an efficient heuristic for calculating the optimal weighted completion time value for a tight lower bound [18]. In another article, Pan et al. presented an integrated scheduling model. They incorporated production scheduling and PM planning together with the objectives in their function to minimize the maximum weighted tardiness. They presented an improved production scheduling model considering PM planning [19]. In another paper, Lapa et al. presented a model with two purposes in the main objective; the first being a methodology for PM based on reliability and the second to optimize the PM for system evaluation. They proposed a GA to attain the optimal solution for their model, concluding probability of the need to repair, the cost of the machine's repair, the impact of the PM on the system, and so on [20]. Berrichi et al. [21] presented an algorithm regarding a production and maintenance scheduling problem as well as ant colony optimization to solve the model. Their model was for the case of a parallel machine, and they presented a reliability model with two objective functions, production, and maintenance. At the end of the article, they offered a multi-objective ant colony optimization model to solve their model and proposed a GA to solve their experimental results. Valdebenito and Schuëller focused on the design of the PM to develop exhaustion cracks. Their problem included the framework of reliability optimization. In their model, they considered maintenance and eventual failure to minimize the costs, taking into account the uncertainty mode in crack propagation phenomenon and inspection activities [22].

In this paper, we consider a multi-parallel series machine that performs multiple jobs on every machine. In most published papers, the authors considered one type of PM, and they do not consider the effects of jobs on each other. Here, in this article, we consider two types of PM, PM to change pieces or to repair parts. We study PM scheduling from a new perspective by considering two aspects: whether or not the previous job is under repair (under PM), which will have an effect on the next job repair (with regard to series machines), so we will check the effect of cost, time, and so on in the application of PM to the problem to see, in the event of its application, which kind of PM (repair or changing pieces) can be applied to the jobs. We also study our problem on a multi-machine in parallel mode where multiple jobs are undertaken on each machine; we apply PM to each job whether it needs repair or not (change or not). In Sect. 2, we define our constraints including cost, reliability, and so on. Here, we use the analytic network process (ANP) aid to weight our objectives in an objective function and define it in Sect. 3. At the end, we present a GA to solve our problem and to compare the answer with the particle swarm optimization (PSO) answer and present the best solution. The main contributions are defined as follows:

- Defining two different kinds of PM, being the repair and change modes

- Considering the effects of jobs on each other (increasing or decreasing reliability) whether or not PM is undertaken
- Presenting a new method of applying PM to machines
- Defining a process for PM scheduling and planning
- Using an ANP method to evaluation the objectives and their alternatives
- Presenting a multi-objective function model to increase reliability and decrease time and cost consumption
- Implementing this model in an example and solving it by a multi-objective dynamic genetic algorithm and a multi-objective particle swarm optimization (MOPSO)

2 Problem description

In this section, we define and describe our problem, constraint, and objectives. We use Matlab software to code the problem.

2.1 Problem definition

The following notations are used to describe the problem studied throughout the paper:

Parameters	
tr	Time of PM in repair mode
tc	Time of PM in change mode
tp	Time of process time
cr	Cost of PM in repair mode
cc	Cost of PM in change mode
D_i	Duration of ith task
p [i, k]	Processing time
d [i,k]	Due time
ei	Weight of earliness cost per unit time for job i
bi	Weight of tardiness penalty per unit time for job i
Variables	
H(t)	Hazard function
X(i,j)	$\begin{cases} 1 & \textit{if the ith job sequentially performed is job j} \\ 0 & \textit{O.w for } i,j=1,2,\ldots,n \end{cases}$
$a_{e[0]}$	Initial age of part in PM
RM_j	Reliability of PM
RE	Total reliability
TVT	Total variance tardiness
TVC	Total variance cost

Here, we consider PM on multi-parallel machines that perform multiple jobs on each of them. The jobs are in series mode and affect each other. If we run PM on each job, the reliability of the machine will be increased. So, according to the maintenance scheduling, if we want to apply PM to the next job, first we should check the reliability and, if it is no lower than a certain upper limit, PM will not be undertaken. As PM was undertaken on the previous job, and the reliability of the machine is increased, PM is not required. PM is undertaken in the first (repaired) mode or the second (changed) mode according to the reliability, cost, and make-span amounts. For better scheduling, we need to calculate the hazard function. In this paper, we define methods of computing the hazard function. The first is the Nelson–Aalen estimator, a nonparametric estimator of the cumulative hazard rate function used when data is censored or incomplete. The estimator is given as follows [23]:

$$H(t) = \sum_{t_i \le t} \frac{d_i}{n_i} \tag{1}$$

Where d_i is the number of events at t_i and n_i are the total individuals at risk at t_i. The second kind of hazard function is computed as follows. Suppose that the machine used to process the jobs is subject to random failure, and the time to the failure of the machine follows a Weibull probability distribution having a scale parameter η and shape parameter b (b>1). When b>1, the hazard function is an increasing function, and it may be practical and important to undertake PM for the machine in order to reduce the increasing risk of machine failure.

$$H(k) = \int k \ \ h(t)dt = \left(\frac{k}{n}\right) \ \beta \tag{2}$$

Where h(t) is the hazard function. Having presented two hazard functions above, we use Eq. (2) in this article. For the *i* th job in the job sequence, let p [i, k] be the processing time, d [i, k] be the due time, w [i, k] be the weight (or priority), then we show these variables as follows [19]:

$$P(i,k) = \sum_i \sum_j p(i,j) \times x(i,j) \tag{3}$$

$$W(i,k) = \sum_i \sum_j w(i,j) \times x(i,j) \tag{4}$$

$$D(i,k) = \sum_i \sum_j d(i,j) \times x(i,j) \tag{5}$$

$$X(i,j) = \begin{cases} 1 & \text{if the ith job sequentially performed before job j} \\ 0 & \text{O . w for i , j = 1 , 2 , … , n} \end{cases}$$

In [16] scheduling the PM plan, the maintenance cost and machine availability must be considered. First, we calculate the reliability, time, and cost of the system in repair mode; if reliability is low and time is high, we will change the component. We assume that the second type of PM restores the machine to the "as good as new" condition and the first type improves the reliability of jobs and machines. To calculate the initial age of a part in the PM:

$$a_{e[0]} = ή \times \exp\left[\frac{\ln(-\ln(1-R_0))}{\beta}\right] \tag{6}$$

Where β and ή are fixed amounts of the Weibull distribution, and R0 is the initial reliability of machine (R0∈[0.7,1]).

After computing the initial age for each machine, we calculate the age of the machine in both types of PM as follows:

$$\mathrm{ae}(\mathrm{i},\mathrm{k}) = \mathrm{ae}(\mathrm{i},\mathrm{k}-1) + \mathrm{P}(\mathrm{i},\mathrm{k}) \quad \mathrm{i} = 1..\mathrm{m}\ \text{number of machine}, \quad \mathrm{k} = 1\ldots\mathrm{n}\ \text{number of jobs}$$

To compute the makespan, we calculate the scheduling completion. The completion time of the ith job in the job sequence is expressed as follows: [19]

$$\mathrm{C}\ (\mathrm{i},\mathrm{k}) = \mathrm{P}(\mathrm{i},\mathrm{k}) + \mathrm{tp} + \mathrm{trx}\left(\left(\frac{\mathrm{ae}(\mathrm{i},\mathrm{k})}{\eta'}\right)\beta - -\left(\frac{\mathrm{ae}(\mathrm{i},\mathrm{k}-1)}{\eta'}\right)\beta\right); \quad (7)$$

Where t_r is the time of the repaired pieces (if the part changes, we use t_c instead of tr) and tp shows the time of the PM.

2.2 Constraint

The level of reliability (for most efficient scheduling) and the time of the PM are the most important problems in this paper; hence, we define the model constraint in this session.

2.2.1 Reliability

The important contributions of reliability are as follows:

1. Using two kinds of PM (repair and change)
2. Considering jobs as series and affected together
3. Studying and presenting a new model for PM scheduling

The level of reliability is the most important constraint in this paper as we want the machines to work well because we do not want the performance of our machines to be lower than a set level. We introduce two constant amounts for reliability and apply PM according to these numbers. If the reliability of the machine is lower than the higher amount, we apply the first type of PM (repairing component), and if the reliability is lower than the smaller amount, we apply the second type of PM (changing component) to the current job. When we apply PM to jobs, the reliability of the jobs increases and affects the reliability of the machine; so, according to scheduling programming, if we want to apply PM on the next job, and reliability of the machine is no lower than the higher or lower amount, PM will not be applied. That is, if the reliability of the machine is greater than a constant amount, PM will not be applied; and if it is lower than a certain level according to the level of reliability and the time of the PM, PM will be applied in change or repair mode. The reliability of each machine decreases when carrying out a job, so we apply PM to the jobs until reliability is increased. If the reliability of the machine is higher (than a constant amount) and the time of the PM is lower (than a constant amount) then: [24]

$$\mathrm{Re} = \mathrm{r} + \mathrm{m2} * (\mathrm{R0} - \mathrm{r}); \quad (8)$$

$$\mathrm{RE} = \mathrm{Re} * \exp\left(\left(\frac{-\frac{1}{\mathrm{m1}*\mathrm{TE}}}{\mathrm{e}}\right)^{\mathrm{b}}\right); \quad (9)$$

Where m_1 and m_2 are random numbers between (0, 1), R0 and r are the initial reliability of the machine and job, respectively, and b and e are constant numbers in the Weibull probability distribution. Equation (9) computes the reliability of one job implemented on each machine. First, we start with the initial reliability for each machine (RM) and assign a reliability amount (r_j) and weight for each job (w_j) because PM is applied to jobs so that their reliability is increased. Then the reliability of each machine is increased as follows:

$$\mathrm{RM}_{\mathrm{j}} = \mathrm{rj} \times \mathrm{wj} + \begin{cases} \mathrm{RM}_{\mathrm{j}-1} & \text{for } \mathrm{j} > 1 \\ \mathrm{RM}_0 & \text{for } \mathrm{j} = 1 \end{cases} \quad (10)$$

As we have assumed that the jobs are located in series mode, we apply PM to other jobs according to the reliability of the machine. r_j is calculated by Eq. (9).

2.2.2 Time of PM

Another important constraint considered in this paper is the time of applying PM. We do not want the PM time to be greater than a constant amount and we want to control our time. So, besides reliability, we consider the time, and if our conditions are satisfactory, then we apply PM.

$$\mathrm{TE} = \mathrm{random}[\mathrm{a} : \mathrm{d}] * \mathrm{z}; \quad (11)$$

In Eq. 11, the time of the PM is computed with z (a random number between (1, 2)).

2.3 Objective function

We have four objectives in the objective function as follows: cost, makespan, total variance tardiness (TVT), and total variance cost (TVC). In this paper, makespan is the most important objective followed by cost, TVT, and TVC, respectively, so we use the ANP aid to calculate the objectives' weights.

2.3.1 PM cost

We have two kinds of cost in our problem for the repaired or changed modes. Naturally, the changed mode cost is more than that of the repaired mode. The cost includes the PM cost and the cost of repair or change [25], so:

$$CR(i,k) = cp + cr \times \left(\left(\frac{\mathrm{a_e}(\mathrm{i},\mathrm{k})}{\acute{\eta}}\right)\beta - \left(\frac{\mathrm{a_e}(\mathrm{i},\mathrm{k}-1)}{\acute{\eta}}\right)\beta\right)$$

or

$$CC(i,k) = cp + cc \times \left(\left(\frac{\mathrm{a_e}(\mathrm{i},\mathrm{k})}{\acute{\eta}}\right)\beta - \left(\frac{\mathrm{a_e}(\mathrm{i},\mathrm{k}-1)}{\acute{\eta}}\right)\beta\right);$$

$$for[\mathrm{i} = 1..\mathrm{m}\ \text{number of machine}, \mathrm{k} = 1\ldots n\ \text{number of jobs}]$$

That CR and CC are the costs of repair and change, respectively. Total cost is shown in the following:

$$\mathrm{cos}t = \sum_{i=1}^{m}\sum_{k=1}^{n}[\mathrm{CC(i,k)} + \mathrm{CR(i,k)}] \tag{12}$$

2.3.2 Total makespan

Makespan is the most important objective in our function because we want the processing time of the PM on the jobs to be minimized. It is equivalent to the completion time of the last job in the job sequence.

$$\mathrm{Makespan} = \mathrm{C(n,j)} \quad \mathrm{j} = 1\ldots\mathrm{m}\cdots\text{number of machine}$$

Where C (n, j) represents the completion time of the last job on machine number j. Total makespan is equal to the maximum amount of completion time of the last job on each machine.

$$\text{Total makespan} = \max[\mathrm{C(n,1),C(n,2),\ldots,C(n,m)}] \tag{13}$$

2.3.3 TVT

Tardiness is an important factor in scheduling, so we consider this problem in this article as one of our objective functions. First, we calculate the weighted mean earliness and the weighted mean tardiness as follows [26]:

$$\textit{Weighted mean earliness}(wme) = \frac{\sum_{i=1}^{n}\max[\mathrm{D_i}-\mathrm{C_i};0]\mathrm{e_i}}{\sum_{i=1}^{n}\mathrm{e_i}} \tag{14}$$

$$\textit{Weighted mean tardiness}\,(wmt) = \frac{\sum_{i=1}^{n}\max[\mathrm{C_i}-\mathrm{D_i},0]\mathrm{b_i}}{\sum_{i=1}^{n}\mathrm{b_i}} \tag{15}$$

e_i and b_i are weights of the earliness cost per unit of time for job i and the tardiness penalty per unit of time for job I, respectively. We want to minimize tardiness, so the earliness includes a negative weight and tardiness includes a positive weight.

$$\text{Variance of tardiness} = \lambda * \frac{\sqrt{\sum_{i=1}^{n}\left[\max[\mathrm{C_i}-\mathrm{D_i};0]-(\mathrm{wmt})^2\right]\times \mathrm{b_i}}}{\sum_{i=1}^{n}\mathrm{b_i}} - (1-\lambda) * \frac{\sqrt{\sum_{i=1}^{n}\left[\max[\mathrm{D_i}--\mathrm{C_i};0]-(\mathrm{wme})^2\right]\times \mathrm{e_i}}}{\sum_{i=1}^{n}\mathrm{e_i}}$$

$$\mathrm{i} = 1\ldots\mathrm{n}, \text{numb of job}$$

So TVT is defined as follows:

$$\mathrm{TVT} = \sum_{j=1}^{m}\text{Total variance of tardiness}\cdots\mathrm{j} = 1\ldots\mathrm{m}\,\text{no of machine} \tag{16}$$

2.3.4 TVC

First, we calculate the weighted mean earliness and the weighted mean tardiness as follows [26]:

$$\text{Weighted mean scheduling cost }(\mathrm{w_{mc}}) = \frac{\sum_{i=1}^{n}\max[\mathrm{D_i}--\mathrm{C_i};0]\mathrm{e_i} + \sum_{i=1}^{n}\max[\mathrm{C_i}-\mathrm{D_i};0]\mathrm{b_i}}{\mathrm{n}}$$

We want to minimize the weighted mean earliness scheduling cost.

$$\text{Variance of cost} = \frac{\sum_{i=1}^{n}\left(\sqrt{(\max[\mathrm{D}_i--\mathrm{C}_i;0]-\mathrm{wmc}^2)}\mathrm{e}_i + \sqrt{(\max[\mathrm{C}_i--\mathrm{D}_i;0]-\mathrm{wmc}^2)}\mathrm{b}_i\right)}{n} \quad i = 1\ldots\mathrm{n}, \text{numb of job}$$

So TVC is defined as follows:

$$\mathrm{TVT} = \sum_{j=1}^{m}\text{variance of cost} \quad \mathrm{j} = 1\ldots\mathrm{m}, \text{numb of machine} \tag{17}$$

We present a new method of maintaining the machines and devices and, according to Eqs. (12)–(1), we established a multi-objective model to do so. In this function, we implement a tradeoff between four objectives. $\mathrm{w_{mak}}$, $\mathrm{w_{cost}}$, $\mathrm{w_{tvt}}$, and $\mathrm{w_{tvc}}$ are the weights of the makespan, cost, TVT, and TVC, respectively. Due to Eqs. (3), (6), (7), and (10), our model is as shown as follows:

$$Z = \min(\mathrm{w_{mak}} * \mathrm{mak} + \mathrm{w_{cost}}*\mathrm{cost} + \mathrm{w_{tvt}} * \mathrm{tvt} + \mathrm{w_{tvc}} * \mathrm{tvc}) \tag{18}$$

$$a_{e[0]} = \eta' \times \exp\left[\frac{\ln(-\ln(1-R_o))}{\beta}\right] \tag{19}$$

$$C(i,k) = P(i,k) + tp + tr \times \left(\left(\frac{ae(i,k)}{\eta'}\right)\beta - \left(\frac{ae(i,k-1)}{\eta'}\right)^{\beta}\right); \tag{20}$$

$$\text{Variance of tardiness} = \lambda * \frac{\sqrt{\sum_{i=1}^{n}\left[\max[C_i-D_i;0]-(wmt)^2\right]} \times b_i}{\sum_{i=1}^{n} b_i} - (1-\lambda) * \frac{\sqrt{\sum_{i=1}^{n}\left[\max[D_i-C_i;0]-(wme)^2\right]} \times e_i}{\sum_{i=1}^{n} e_i} \tag{21}$$

$$CR\,(i,k) = cp + cr \times \left(\left(\frac{a_e(j,i)}{\acute{\eta}}\right)^{\beta} - \left(\frac{a_e(j,i-1)}{\acute{\eta}}\right)^{\beta}\right); \tag{22}$$

$$CC(i,k) = cp + cc \times \left(\left(\frac{a_e(j,i)}{\acute{\eta}}\right)^{\beta} - \left(\frac{a_e(j,i-1)}{\acute{\eta}}\right)^{\beta}\right); \tag{23}$$

$$\text{Variance of cost} = \frac{\sum_{i=1}^{n}\left(\sqrt{(\max[D_i-C_i;0]-wmc^2)}e_i + \sqrt{(\max[C_i-D_i;0]-wmc^2)}b_i\right)}{n} \tag{24}$$

$$RM_j = \left((r + m2 * (R0-r)) * \exp\left(\left(\frac{-\frac{1}{m1*TE}}{e}\right)^{b}\right)\right) \times W(i,k) + \begin{cases} RM_{j-1} & \text{for } j > 1 \\ RM_0 & \text{for } j = 1 \end{cases} \tag{25}$$

$$\begin{cases} R_l \le RM_j \le R_u & \text{PM is undertaken in repair mode} \\ RM_j \le R_l & \text{PM is undertaken in change mode} \end{cases} \tag{26}$$

$$x(i,j) = \begin{cases} 1 & \text{if the ith job sequentially performed before job } j \\ 0 & O.w \text{ for } i, j = 1, 2, \ldots, n \end{cases} \tag{27}$$

$$\lambda\,(0,1) \tag{28}$$

Equation (18) shows the objective function. Equation (19) shows the initial age of the part. Completion time is computed in Eq. (20). Equation (21) computes the amount of variance of tardiness. Equations (22) and (23) compute the cost of the replace mode and change mode, respectively. Variance of cost is computed in Eqs. (24) and (25) achieves the reliability level for each machine. Equation (26) expresses when PM is undertaken; if reliability is between two specific numbers (R_l, R_u), PM (in repair mode) is undertaken; otherwise if reliability is lower than R_l, PM is undertaken in change mode. x (i, j) is 1 if the ith job is sequentially performed before the job and 0 otherwise; and λ is achieved randomly between (0, 1).

2.4 ANP method for weighting the objective function

ANP is a structured technique for organizing and analyzing complex decisions based on elite viewpoints. ANP is a generalization of the analytic hierarchy process (AHP) used in multi-criteria decision making and analysis. AHP structures a decision problem into a hierarchy with a goal, decision criteria, and alternatives, while the ANP structures it as a network. This method was developed by Thomas L. Saaty [27], and other authors have extended it. While it can be used by individuals working on straightforward decisions, the ANP is most useful where teams of people are working on complex problems involving human perceptions and judgments. This method can be used when we do not have sufficient data for our problem. ANP helps to capture both subjective and objective evaluation measures, providing a useful mechanism for checking the consistency of the evaluation measures and alternatives suggested by the team, thus reducing bias in decision making. It also helps us when making complex decisions involving multiple criteria.

We have 1 goal, 4 objective functions, and 12 alternatives in our problem as follows: cost, TVT, TVC, and makespan (in manufacturing the time difference between the start and finish of a sequence of jobs or tasks) are the objectives; and cost of PM, programming time, cost of project, damage to pieces, cost of repair, number of changed pieces, number of repaired pieces, cost of changed pieces, starting time, time of project, preventive in program, and preventive in precautionary are the alternatives. We use super decision software to compute the objective function coefficients. Figure 1 shows our network structure.

The degrees of importance of our elements are as follows: makespan, cost, TVT, and TVC, in our article, and the value of their coefficients.

2.5 DMOGA

Our objectives in the multi-objective optimization problem (MOP) are always in opposition, so the problem hardly converges on one solution that optimizes all the objectives altogether. A MOP usually produces a set of optimal solutions in which each of them is not downscaled for the decision makers. GA is a known population-based heuristic algorithm and is a suitable method to solve multi-objective problems. Because our model is multi-objective, we use the multi-objective genetic algorithm. Therefore, in this section, we describe our dynamic GA to solve this problem and present a new GA method. When the best solution does not change over a number of iterations, the coefficients of mutation and crossover are changed automatically throughout the GA (the coefficient of mutation is decreased and the coefficient of crossover is increased).

2.5.1 Crossover

Crossover is an important operation in GA in calculating the best answer. A chromosome is defined as a binary string, in PM whether running or not (0 means PM is not undertaken and 1 means PM is undertaken). We use the roulette-wheel method for crossover in this model (produced for the parents and the selected genes), namely one random number is generated, and according to this number crossover is carried out.

2.5.2 Mutation

First, we define one coefficient mutation and then use one-point for mutation, where the value of the gene at the mutation point is changed from 0 to 1 or vice versa

2.6 Multi-objective particle swarm optimization

PSO is a population-based stochastic optimization technique, developed by Eberhart and Kennedy in 1995, inspired by the social behavior of birds flocking or fish shoaling. PSO shares many similarities with evolutionary computation techniques, such as GAs. The system is initialized with a population of random solutions and searches for the optima by updating generations. However, unlike GA, PSO has no evolutionary operators, such as crossover and mutation. In PSO, the potential solutions, called particles fly through the problem space by following the current optimum particles. PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality; therefore, here, we present a PSO algorithm for comparison with the achieved answer above.

3 Numerical example

In this section, we present a numerical example for our model and solve it by representative GA and PSO and compare the two answers. We solve our problem with five jobs and five machines. If PM is applied to each job, the reliability of the machine will be increased and will affect other jobs. First, we define important values for the problem variables. We compute the initial reliability and age of the machine for the time of the PM in changed mode (t_c), which is defined randomly with interval (70, 75), and for the time of the PM in repair mode, this value is multiplied by one random number (w_{tr}) with an interval (1, 2) ($w_{tr} \in [1, 2]$), namely $t_r = w_{tr} \times t_c$; similarly for the cost of the PM in changed and repaired modes. So, cc (cost of changed mode) = c_r (cost of repaired mode) × w_{cc} ($w_{cc} \in [1, 2]$). η and β (Weibull probability distribution parameters) are 150 and 2, respectively.

In the objective function, because our functions are not of one type, we normalize our objective as far as the sum. Then, we use ANP and ascertain the weight of each objective in terms of importance, so w_{mak}=0.5469 (weight of makespan), w_{cost}=0.2167 (weight of cost), w_{tvt}=0.1302 (weight of TVT), and w_{tvc}=0.1067 (weight of TVC). To solve the model with the GA, we assume initial coefficients of mutation and crossover as 0.2 and 0.7, respectively. If the best solution of the GA does not

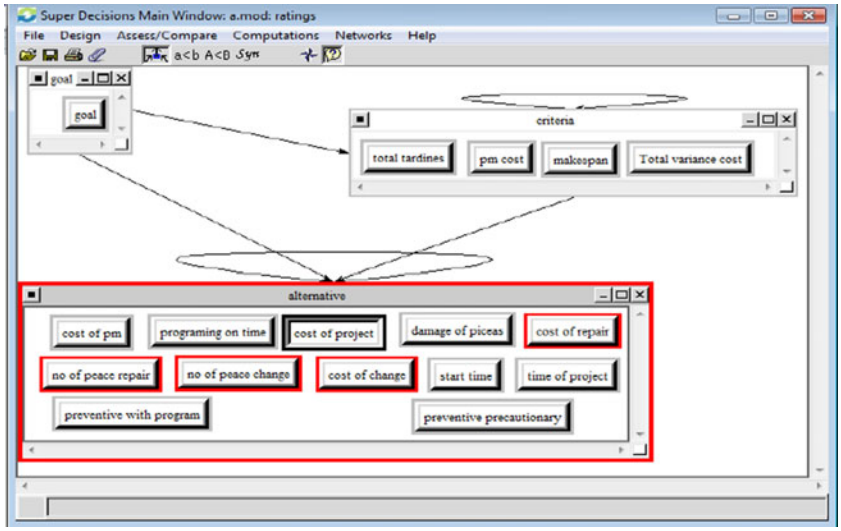

Fig. 1 Problem network for ANP

change for 15 iterations, the coefficients of mutation will be increased to 0.05 and the coefficients of crossover will be decreased to the same amount. Our algorithm starts with an initial RM, and then, according to the time of the PM repair and change mode times, we make the decision to apply PM or not. According to the level of reliability and the age of the machine (using the hazard function), we calculate the cost and conclusion time of each session. At the end, we calculate the makespan with a completion time for each machine and choose the maximum machine makespan for the total makespan. At the end, we normalize our objective function without dimension and then we affect the weight of the objective in the objective function. We use Matlab software to solve our problem.

Table 1 shows the best PM scheduling of the GA. For an amount of 1, we undertake PM and for amount of 0 we do not. Table 2 shows the best PM scheduling of the PSO. In Table 3, we show the value of each objective over ten iterations of the algorithms, and Fig. 1 shows the changes in each objective. For each run, we computed each value of the objective function and show them in Table 3. Figure 2 shows the value of the objective functions for different runs for GA. Figure 3 shows the value of the objective functions for different runs for PSO. For example, in iteration number 3, Table 4 shows the makespan (completion time) information of each job on each machine. Figure 4 shows a Gantt chart of the second and fourth machines and their jobs. Table 5 shows information about the final PM scheduling, the best value for the objective function, and presents the weights of the objectives achieved by the ANP. Figure 5 shows the process for the answers achieved by the GA for different results. Figure 6 shows the process for the answers achieved by the PSO for different results. The tables and figures show that the MOPSO algorithm

Table 1 Final PM scheduling of GA

No of job / No of machine	1	2	3	4	5
1	1	1	0	0	1
2	0	0	1	1	1
3	1	0	1	0	0
4	0	0	0	0	1
5	1	0	1	0	0

Table 2 Final PM scheduling of PSO

No of job / No of machine	1	2	3	4	5
1	0	0	0	1	1
2	1	0	0	1	1
3	0	0	1	1	0
4	0	1	0	0	1
5	1	0	0	1	0

Table 3 Amount of each objective for different runs

Objective run no.	PM cost (GA)	Makespan (GA)	TVT (GA)	TVC (GA)	PM cost (PSO)	Makespan (PSO)	TVT (PSO)	TVC (PSO)
1	9.7228e+003	1.3124e+003	1.3837e+003	1.63E+04	8.7931e+003	1.1529e+003	9.0311e+003	4.62E+03
2	1.2344e+004	408.5847	1.3692e+003	2.10E+03	1.0658e+004	784.2766	6.9868e+003	1.97E+04
3	1.0053e+004	567.5563	4.4329e+003	4.2650e+004	1.0611e+004	250.2367	1.0660e+004	953.2342
4	8.5901e+003	324.9763	1.2548e+004	422.0276	1.1832e+004	426.0966	5.9158e+003	5.70E+03
5	1.1137e+004	1.9691e+003	6.8642e+003	4.37E+03	8.1161e+003	578.8836	8.3699e+003	3.06E+03
6	1.0089e+004	441.4521	1.4234e+004	1.52E+04	1.0003e+004	718.7370	1.0888e+003	3.98E+03
7	1.0665e+004	257.0400	3.9507e+003	249.1547	1.0658e+004	1.0254e+003	7.2118e+003	1.50E+04
8	9.7134e+003	182.5059	1.8676e+004	28.5072	8.9828e+003	420.9371	1.1218e+003	9.65E+03
9	9.4322e+003	974.5392	3.6206e+003	1.25E+04	1.0137e+004	493.9570	4.1934e+003	1.10E+03
10	1.0174e+004	791.6252	1.3142e+004	6.26E+03	1.0661e+004	783.9952	1.3844e+003	6.31E+03

achieved the final answer sooner, and proposes a better answer than the DMOGA algorithm in this problem.

4 Conclusion and future research

In this paper, we presented a PM scheduling method according to the reliability level, PM time, and cost. In most previous articles, the writers did not consider the influence of jobs on each other when PM is undertaken on one job. Namely, if PM is undertaken on one job, the reliability of the machine is increased and, although the next job may need PM, PM does not occur because the machine's reliability has increased. First, we considered the PM problem in multi-parallel machines undertaking multiple jobs on each machine (these jobs are in series mode and they affect each other) and assumed two kinds of PM (repair and change mode). We tried to minimize the multi-objective functions of cost, TVT, makespan, and TVC and used ANP to weight these objective functions. In this case, PM was undertaken in the first mode (repair the piece) or second mode (change the piece) according to the level of desired reliability from Eq. (26). With the numerical example, we considered our model with two solution methods (GA and PSO).

This method accordingly considers all aspects of cost, time, and reliability, presents a good maintenance schedule, and considers all influences of the jobs on each other. Also, the achieved results show that this method is very useful for machine maintenance because the reliability of the machine is always high (this amount may be different in various cases), cost and makespan are near their lowest amounts (according to the level of reliability), and variance from cost and tome (from the initial specific amount) are trivial. Hence, the main factors in the maintenance problem (cost, time, reliability, time variance, and cost planning) are considered in this method.

The following may be of interest in future research:

- Solving the problem using other heuristic or meta-heuristic methods and comparing them with our results.
- Considering constraints in undertaking the PM in our model, including those on cost and makespan.
- Using the Nelson–Aalen estimator to compute the hazard function and compare it with our answer.

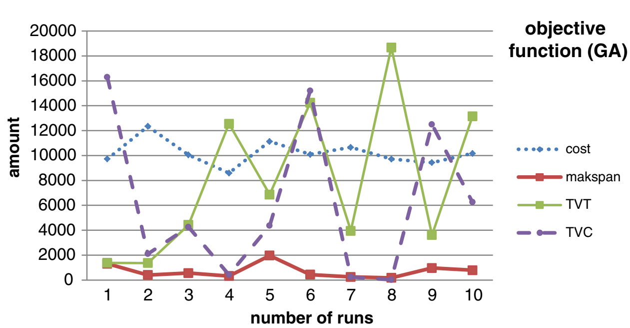

Fig. 2 Amount of objective in different runs, solving with GA

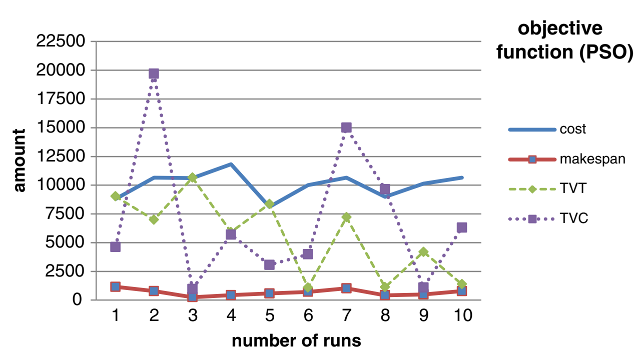

Fig. 3 Amount of objective in different runs, solving with PSO

Table 4 Completion time of jobs in first machine

Job number Makespan	1	2	3	4	5	Amount of makespan
Completion time of 1st machine	29.8831	49.2332	63.4992	124.8533	141.7801	141.7801
Completion time of 2nd machine	267.9911	285.1283	535.2368	550.3332	567.5563	567.5563
Completion time of 3rd machine	18.0015	34.4602	118.8562	130.8410	224.8860	224.8860
Completion time of 4th machine	17.5437	17.5437	17.5437	77.4629	77.4629	77.4629
Completion time of 5th machine	101.2191	116.5028	326.1655	339.5266	339.5266	339.5266

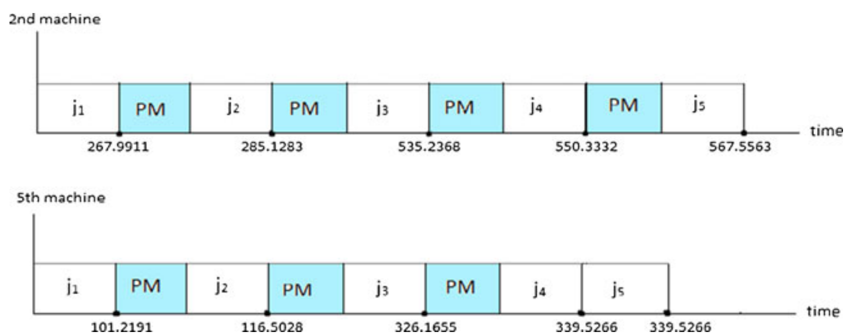

Fig. 4 Gantt chart of 2nd and 5th machines

Table 5 Summary of our output

Objective	Objective weight	Best amount of objective function from GA	0.3926
PM cost	0.447	Best amount of objective function from PSO	0.3764
TVT	0.395	Number of iteration	400
TVC	0.103	Number run of algorithms	10
Makespan	0.056		

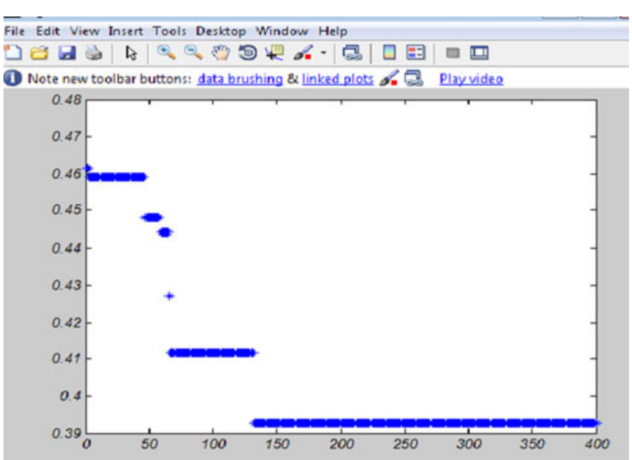

Fig. 5 Genetic algorithm

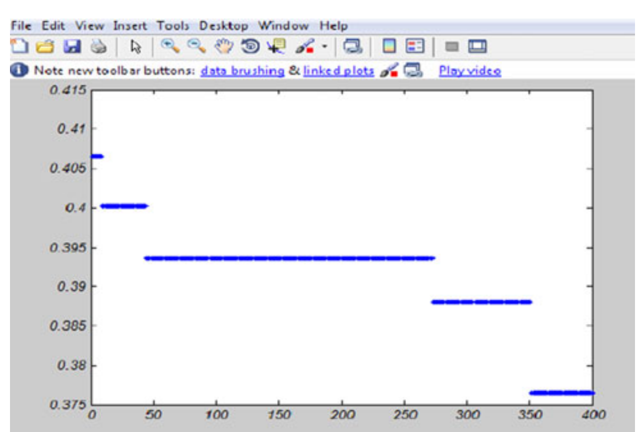

Fig. 6 Particle swarm optimization

References

1. Fitouhi M-C, Nourelfath M (2012) Integrating noncyclical preventive maintenance scheduling and production planning for a single machine. Int J Prod Econ 136:344–351
2. Moghaddam K, Ushe SJ (2011) Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models. Comp Indust Eng 61:64–75
3. Sadfi C, Penz B, Rapine C, Błażewicz J, Formano P (2005) An improved approximation algorithm for the single machine total completion time scheduling problem with availability constraints. Eur J Oper Res 161:3–10
4. Ruiz R, García-Díaz J (2007) Considering scheduling and preventive maintenance in the flow shop sequencing problem. Comp Indust Eng 34:3314–3330
5. Bris RC (2003) New method to minimize the preventive maintenance cost of series–parallel systems. Reliab Eng Syst Saf 83:247–255
6. Shalaby MG (2004) A genetic algorithm for preventive maintenance scheduling in a multiunit multistate system. J Eng Appl Sci 51:795–811
7. Yao XF-G (2004) Optimal preventive maintenance scheduling in semiconductor manufacturing. IEEE Trans Semicond Manuf 17: 345–356
8. Cassady C, Kutanoglu E (2005) Integrating preventive maintenance planning and production scheduling for a single machine. IEEE Trans Reliab 54:304–309
9. Duarte JC (2006) Optimization of the preventive maintenance plan of a series components system. Int J Press Vessel Pip 83: 244–248
10. Chelbi A, Ait-Kadi D (2004) Analysis of a production/inventory system with randomly failing production unit submitted to regular preventive maintenance. Eur J Oper Res 156:712–718
11. Nakagawa TNS (2007) Entoropy model with application to maintenance policy. IEEM, pp. 654–656
12. Tam AC (2006) Optimal maintenance intervals for multicomponent system. Prod Plan Control 17:769–779
13. Prasad SD, Rajendran C (2006) A genetic algorithmic approach to multi-objective scheduling in a Kanban controlled flow shop with intermediate buffer and transport constraints. Int J Manuf Technol 29: 564–576
14. Alardhi MH (2007) Preventive maintenance scheduling for multi-cogeneration plants with production constraints. J Qual Maint Eng 13:276–292
15. Wu S (2012) Assessing maintenance contracts when preventive maintenance is outsourced. Reliab Eng Syst Safety 98:66–72
16. Kenne JP, Gharbi A (2007) Age-dependent production planning and maintenance strategies in unreliable manufacturing systems with lost sale. Eur J Oper Res 178:408–420
17. Qi X (2007) A note on worst-case performance of heuristics for maintenance scheduling problems. Discret Appl Math 155: 416–422
18. Gur Mosheiov AS (2009) Scheduling a maintenance activity to minimize total weighted completion time. Comp Math Appl 57: 619–623
19. Pan E, Liao W, Xi L (2010) Single machine-based production scheduling model integrated preventive maintenance planning. Int J Adv Manuf Tech 50
20. Celso MF, Lapa CM (2006) A model for preventive maintenance planning by genetic algorithms based in cost and reliability. Reliab Eng Syst Saf 91:233–240
21. Berrichi A, Yalaoui F (2010) Bi-objective ant colony optimization approach to optimize production and maintenance scheduling. Comp Oper Res 37:1584–1596
22. Valdebenito MA (2010) Design of maintenance schedules for fatigue-prone metallic components using reliability-based optimization. Comput Methods Appl Mech Eng 199:2305–2318
23. Perez-Marın AM (2008) Empirical comparison between the Nelson-Aalen Estimator and the Naive Local Constant Estimator. University of Barcelona, 7–76
24. Saaty TL, Luis, G (2006) Decision making with the analytic network process: economic, political, social and technological applications with benefits, opportunities, costs and risks
25. Yulan J, Zuhua J, Wenrui H (2008) Multi objective integrated optimization research on preventive maintenance planning and production scheduling for a single machine. Int J Adv Manuf Technol 39:954–964
26. Thiagarajan S, Rajendran C (2005) Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness, and weighted flow time of jobs. Comp Indust Eng 49:463–503
27. Saaty TL (1996) Decision making with dependence and feedback: the analytic network process. RWS Publications