J Intell Manuf (2013) 24:695–705
DOI 10.1007/s10845-011-0616-3

Maintenance scheduling incorporating dynamics of production system and real-time information from workstations

Ali Arab · Napsiah Ismail · Lai Soon Lee

Received: 22 January 2011 / Accepted: 6 December 2011 / Published online: 1 January 2012

Abstract In this paper, a new approach to maintenance scheduling for a multi-component production system which takes into account the real-time information from workstations including remaining reliability of equipments as well as work-in-process inventories in each workstation is proposed. To model dynamics of the system, other information like production line configuration, cycle times, buffers' capacity and mean time to repair of machines are also considered. Using factorial experiment design the problem is formulated to comprehensively monitor the effects of each possible schedule on throughput of the production system. The optimal maintenance schedule is searched by genetic algorithm-based optimization engine implemented in a simulation optimization platform. The proposed approach exploits all of makespans of planning horizon to find the best opportunity to perform maintenance actions on degrading machines in a way that maximizes the system throughput and mitigates the production losses caused by imperfect traditional maintenance strategies. Finally the proposed method is tested in a real production line to magnify the accuracy of proposed scheduling method. The experimental results indicate that the proposed approach guarantees the operational productivity and scheduling efficiency as well.

A. Arab
Member of Young Researchers Club, Islamic Azad University, Firouzkooh, Tehran, Iran
e-mail: ali.arab@mavs.uta.edu

N. Ismail (✉)
Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University of Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia
e-mail: napsiah@eng.upm.edu.my

L. S. Lee
Department of Mathematics, Faculty of Science, University of Putra Malaysia (UPM), 43400 Serdang, Selangor, Malaysia

Keywords Maintenance scheduling · System dynamics · Simulation optimization

Introduction

Scheduling in manufacturing systems is a complicated and challenging task requiring the precise consideration of competing alternative resources coupled with the ability to respond rapidly to changing requirements. Optimized scheduling is a key factor and is one of the most crucial planning and operational issues in manufacturing environments to increasing system productivity. There is a trade-off between assigned time to production and available time to perform maintenance actions in production systems (Yang et al. 2007a). Therefore there is a great need to develop a systematic maintenance scheduling support tool to incorporate available information about production and machine failure condition to exploit maintenance opportunities during the shift.

In the recent years, the maintenance of systems has been becoming more and more complicated. The reason of this complexity is that systems have so many components which are interconnected with each other. In one side, interactions among components make it complicated to model and optimize a maintenance system. On the other side, interactions also propose the opportunity as well as group maintenance which may save costs (Nicolai and Dekker 2008). In today's complex manufacturing systems such as automotive assembly line, the maintenance scheduling should not consider only a single machine's condition, but also interconnection between equipments should be taken into account (Wang 2002). In the literature there are two relevant maintenance policies for interconnected and multi-equipment systems which can be categorized into group maintenance policies and opportunistic maintenance policy.

Under the category of group maintenance policy, a group of failed equipments are replaced instead of replacing failed equipments individually. Most of researches on optimal maintenance strategies focus on one-unit systems. However, in many real cases, systems are including groups of identical units. By replacement of groups of failed equipments instead of replacing failed equipments individually, cost reduction can be realized. This cost saving which is known as the economy of scale, results mostly from the quantity discount or reduction of maintenance set-up cost per equipment (Sheu and Jhang 1996).

Under the opportunistic maintenance policy, a maintenance model proposed where repair or replacement of equipment's component is available at an opportunity. An opportunity arises if the repair or replacement of a component in other equipment of system allows the other equipments in question to be repaired (Dagpunar 1996).

The optimization methods for multi-component systems are categorized based on planning horizon of maintenance model. In the first category which majority of researches have been done, the time horizon is infinite. It facilitates the mathematical analysis which in the most cases is able to derive analytical expressions for optimal control parameters and corresponding optimal costs. Therefore, in the category of infinite horizon, models policy optimization is the most popular optimization method. In the category of finite horizon, the system is considered in this horizon merely, and assumed that is not used afterwards, unless a residual value for estimation of industrial value of the system at the end of the planning horizon is incorporated. The optimization methods which are applied to finite horizon are either exact methods or heuristics. Exact methods always are able to find the global optimum solution for problems. If the complexity of an optimization problem is high and the computing time of the exact method increases exponentially with the size of the problem, then heuristics can be applied to find a near optimal solution in reasonable time (Nicolai and Dekker 2008). In this context, Langdon and Treleaven (1997) proposed a heuristic which has been combined with genetic algorithm to solve a multi-component maintenance scheduling problem. The combination of their proposed hand coded heuristic and GA demonstrates that the time taken to perform GA fitness evaluations and also program run time grows quickly with problem size and the number of potential failures which should be taken into account. However their proposed heuristic yields good schedules within a reasonable time. In another study Higgins (1998) proposed a model for determining the best allocation of maintenance activities and crews. The proposed model is subject to constraints. The problem has been solved using the tabu search heuristic for which the neighborhood is defined by swapping the order of jobs, maintenance crews, or both. This model also produces a computationally efficient optimal solution. In the problem which has been solved by Papadakis and Kleindorfer (2005) they have formulated the problem by binary programming method, Rhys–Balinski method, max-flow min-cut method, and enforced maintenance formulation. Finally they could find an exact solution which computationally is efficient to solve the problem. Grigoriev et al. (2006) proposed several models for a periodic maintenance scheduling problem that has applications in many different areas. Their approach has been to fix the length of the period to a given constant T. In their study most of mathematical formulations are linear integer programs. They have shown that this formulation can be solved using column generation. This resulted in a branch and price algorithm to find the exact solution for problem. Budai et al. (2006) presented two versions of the preventive maintenance scheduling problem for a multi-component system, first one with fixed intervals between two consecutive executions of the same routine work, and another one with only a maximum interval. Apart from giving a math programming formulation for the maintenance scheduling problem and for its extension they also presented some heuristics which are based on intuitive arguments. Rakowsky (2006) proposed a modeling of reliability-adaptive multi-system operation to increase the reliability and overall performance such system. In his study, two independently operating systems and a single maintenance unit were considered. Using a convolution-based approach, he quantified the problem, and tailored it for fleets of ships, aero-planes, spacecraft, and vehicles. His study indicates that the reliability-adaptive system (RAS) concept makes sense if average system output loss due to lowered performance level is smaller than average loss due to waiting for maintenance in a non-adaptive case. Dietl and Rakowsky (2006) presented a strategy which applies a reliability-adaptive operating strategy in combination with tool derating in order to hold the system harmonization of maintenance actions. Thy compared the output-time functions of a transfer line without reliability-adaptive with functions of a system with reliability-adaptive control. In their study, the economic efficiency of a multi-station transfer line was evaluated in term of the quantity of manufactured parts. Schutz et al. (2009) proposed periodic and sequential preventive maintenance policies for a system which performs different missions based on a dynamic system failure law over a finite planning horizon. In their study, first step is to achieve the set of missions to perform by determining the optimal business plan in order to maximize the profit of missions minus maintenance costs. Therefore, for each plan, maintenance planning is determined by taking into account two maintenance policies. For the periodic maintenance policy the objective is to find the optimal number of preventive maintenance actions. For the sequential maintenance policy, the optimal number of preventive maintenance intervals and the duration of these different intervals are determined. For those interested readers, a thorough review on optimal maintenance of multi-component systems can be found at published paper by Nicolai and Dekker (2008).

Online information about the system status, defined as the dynamic distribution of the work-in-process in the production line to find maintenance work-order prioritization already has been proposed by Yang et al. (2007b). Yang et al. (2008) also introduced another maintenance scheduling method for manufacturing systems using the continuous evaluation and prediction of the performance degradation level of equipment, as well as the complex interaction between the production process and maintenance operations. They used a genetic algorithm based optimization procedure to search for the most cost-effective maintenance schedule, taking into account both production throughputs and maintenance costs. Their algorithm was implemented in a simulated environment and benchmarked against several traditional maintenance strategies, such as corrective maintenance, scheduled maintenance and condition-based maintenance. Their study indicates that their proposed maintenance scheduling method could result in significant gains obtained by optimal maintenance scheduling. In the same year, a systematic approach to find maintenance opportunities utilizing real-time information of production status so that maintenance decisions can be made responsively at all times proposed by Chang et al. (2007).

The importance of real-time information in maintenance scheduling of multi-component systems already has been stressed by Yang et al. (2007b, 2008), and Chang et al. (2007), but it can be much more efficient if we utilize the real-time information from workstations including remaining reliability of equipments in the beginning of each production shift, as well as amount of work-in-process in the production line to find the best maintenance opportunities during entire of the production shift in question which considers the dynamics of production system. The objective of this research is to find the best schedule to perform maintenance actions on machines in a production line that are subject to failure based on given information by predictive embedded devices like Watchdog Agents (Ni et al. 2006) about their condition in the beginning of the shift. Real-time information from workstations is a very important issue in this research. However, a continuous information retrieval from workstations is unrealistic and useless action. Therefore, the real-time information retrieval mechanism is meant to be applied every hour, shift, day, or week, depending on the situation at hand (which in this research is the scheduling baseline at the beginning of planning horizon), and not necessarily in minutes or seconds.

In the continuation of this paper, first, basic concept of dynamics of production system toward maintenance schedule based on a simplified model is discussed. Next, the model in the simulation optimization platform is introduced. Afterwards, the model is implemented using commercial simulation optimization software. Finally, the paper is concluded with experiments, results and discussions.

Dynamics of production system

The purpose of this research is to find the best schedule to perform maintenance actions on machines that are subject to failure. A machine becomes idle when maintenance action is performed on it. Under this circumstances, the buffer feeding into the workstation as shown in Fig. 1, may cause a congestion in the upstream machines, while the buffer fed by the station may become empty, causing a starvation in the downstream machines. The main concern is to utilize the time in an efficient manner, so that the production system's behavior is cost effective and operationally efficient.

Consider a simple production line as described in Fig. 2. The rectangular shapes represent the machines and the circular shape is the buffer. Let the cycle time of machine M_1 and M_2 be 1 min. In fact, it is the processing time to realize one piece from each workstation and transfer it to the next downstream inventory bank. The buffer capacity is one unit only. Let the mean time to repair (MTTR) of machines M_1 and M_2 be 1 min. Assume that there is one work piece as initial buffer level in the buffer B_1 at the beginning of the shift and the shift length is 3 min only. We are to investigate the dynamics of production system in term of the final throughput and the work-in-process distribution at the end of the shift by performing the maintenance actions during the shift which makes a machine idle while another one may continue to function. To do so, a factorial experiment should be designed. Generally in factorial experiment design, experimental trials (or runs) are performed at all combinations of factor levels (Montgomery and Runger 2006). By a factorial experiment we mean that in each complete replication experiment, all possible combinations of the levels of the factors are investigated and the response of each set of factors in a variety of factor levels should be estimated.

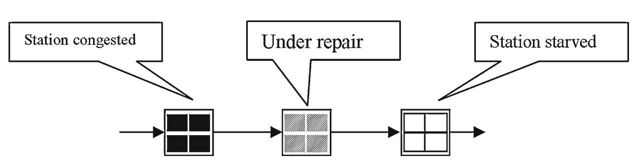

Fig. 1 The effect of a machine shutdown on upstream and downstream workstations

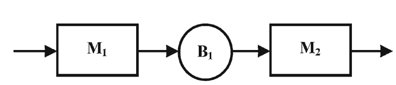

Fig. 2 The shutdown on workstations

Table 1 Combination of factor levels and their corresponding response

Combination[a]	(0, 0)	(0, 1)	(0, 2)	(1, 0)	(1, 1)	(1, 2)	(2,0)	(2,1)	(2,2)
Partially processed	1	1	2	1	1	1	1	0	2
Fully processed	2	2	1	1	2	2	2	2	2
Response value	2.5	2.5	2	1.5[b]	2.5	2.5	2.5	2	3[c]

[a] The first and second elements in parenthesis represent the shutdown time for machine M_1 and M_2 respectively
[b] The minimum response value
[c] The maximum response value

In this experiment, we take the workstations' shutdown times as factors, and the effect of combination of these factor levels (which forms a possible breakdown schedule) on throughput of the production line and work-in-process distribution in the production line as response. Thus, each factor will have three factor levels which are beginning of the 1st, 2nd, and 3rd minutes. Therefore, each replication of crossing the factor levels of these two factors contains nine combinations. The effect of crossing factor levels produces a response which in this experiment is defined to be the number of work pieces which have been fully or partially processed. Let the value of "1" be assigned to fully processed work pieces and value of "0.5" be assigned to partially processed parts (because half of the total process is done at M_1). These values are multiplied by the corresponding number of work pieces and the cumulative amount of all of them makes the response value for the system. Mathematically, $RV = 0.5PP + FP$, where *RV* is the response value of the system, *PP* represents the number of partially processed parts and *FP* is the number of fully processed parts. The derived responses for all of possible combinations calculated through a manual discrete-event simulation are shown in Table 1. The result of this experiment indicates that the highest response (response value of "3") occurs at combination (2, 2) when both machines should go shutdown at the beginning of 2nd minute in order to be repaired, while the lowest response value (response value of "1.5") occurs at combination (1, 0) when machines M_1 and M_2 should go shutdown at the beginning of 1st and 0th minutes respectively. This simple experiment obviously demonstrates that the number of partially and fully produced parts and therefore, the dynamics of production system significantly depends on maintenance schedule and it is worth to find a solution which searches the optimal maintenance schedule to avoid production losses and disruptions to the next shift. In the next section, this solution will be described.

Optimal maintenance scheduling through simulation optimization

In this section, the algorithm for optimal maintenance scheduling will be described. In the real world's problems, implementing such an experiment to find the best response and its corresponding schedule is not so straightforward. For instance, consider a production system consisting ten machines which four of them are subject to repair during the shift. It means that there are four factors. If the mean time to repair for each of these machines is 20 min, then in a shift which consists of 8 h (480 min), the available time range to performing maintenance actions during the shift is within 0th to 460th minutes. Therefore, each factor has 460 factor levels. Logically, in this experiment, the number of complete combinations of factor levels which their response should be investigated will be $460^4 = 44{,}774{,}560{,}000$. Obviously, for such large solution space, analytical calculations using designed experiments are very hard or even impossible to be used. In contrast, discrete-event simulations are able to faithfully represent the behavior of such complicated systems. However, conducting an exhaustive discrete-event simulation for all combinations to search the best (optimal) response is neither effective nor efficient (Ólafsson 2006). One of the most efficient methods to find the optimum and near optimum level of response surface which has been introduced yet is "simulation optimization" (Fu 2002). Simulation optimization is defined as the optimization of performance measures based on outputs from simulations. In this process a general purpose optimization engine is interfaced with a general purpose simulation program (Fu et al. 2005). The optimization engine generates the set of input factors to perform the simulation. The simulation program measures the performance of candidate solutions. This process continues until satisfied solution or termination condition is obtained (Rogers 2005). Figure 3 illustrates the mechanism of simulation optimization for discrete-event systems. The general problem setting for simulation optimization is the following parametric optimization problem:

$$\min_{\theta\in\Theta} \quad J(\theta) = E[L(\theta, \omega)] \tag{1}$$

where $J(\theta) = E[L(\theta, \omega)]$ is the performance measure of interest, $L(\theta, \omega)$ is called the sample performance, ω indicates the stochastic effects of the system, θ is a controllable vector of p parameters, and Θ is the constraint set on θ, either defined explicitly or implicitly (as in mathematical programming formulations), but assumed to be a closed set. The optimum is defined as follows:

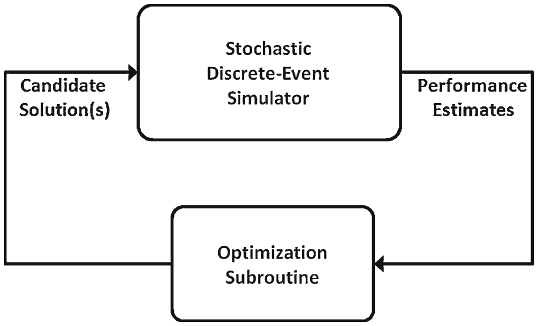

Fig. 3 Mechanism of simulation optimization for discrete-event systems (Fu 2002)

$$\theta_* = \arg\min_{\theta \in \Theta} J(\theta) \quad (2)$$

As an approach to global optimization, genetic algorithm (GA) is applicable to optimization problems which are intractable for exact solutions by conventional methods (Holland 1975). Moreover, most of existing studies in this context are on basis of numerical evaluations and the claim is made that genetic algorithm is robust and therefore applicable to simulation optimization (Ólafsson 2006). Therefore, it can be used to search the near optimal maintenance schedule of this problem; thus, we implement the problem in question in a simulation optimization platform facilitated by GA. The chromosome evolution process to search the near optimal maintenance schedule should be able to perform the repair mechanism to recover those selected chromosomes which are out of constraint (Kampen et al. 1996). To implement the proposed scheduling platform, first the production system should be simulated by discrete-event simulator. Next, the maintenance actions in the production system will be simulated through a series of "workstation waiting" and "workstation starting" events manipulating according to the generated shutdown sets by optimization engine. Then the derived response through simulation is sent to the GA-based optimization engine to evaluate the fitness of solutions. In each generation, based on performance of the current set of solutions, a subset of theses solution is chosen and these solutions are combined into new solutions. In GA, the operators which are used to generate the new solutions are survival, where a solution is carried to the next generation without any change, mutation, where a solution is modified slightly, and crossover, where the properties of two solutions are combined into one solution. The same process is then repeated with the new set of solutions. The mutation and crossover operators depend on the representation of the solution, but not on the evaluation of its performance, while selection of solutions is carried out on basis of their performance. Solutions with higher performance should have more chance of both surviving and being allowed to generate new solutions via crossover. The simplest approach is to order the solutions $J(\theta_{[1]}) \leq J(\theta_{[2]}) \leq \cdots \leq J(\theta_{[n]})$, and only operate on the best solutions. If a strict selection of the top k solutions were needed, this considerably complicates the issue in the simulation optimization context, and substantial simulation effort should be spent to achieve a precise ordering of the solutions (Ólafsson 2006).

With respect to which solutions are chosen to generate the next set, genetic algorithms are quite robust. Nevertheless, a purely deterministic selection of the top k solution is typically not the best approach for deterministic problems, and some randomness is usually involved into the optimization process. Roulette strategy is an instance of this which probability of selecting a solution θ is figured out as follows:

$$P(\theta) = \frac{\hat{J}(\theta)}{\sum_{All\theta} \hat{J}(\theta)} \quad (3)$$

where $\hat{J}(\theta)$ is an estimation for the fitness function $J : \Theta \rightarrow$ R which measures the quality of the solution which should be maximized (because the higher value implies more fit).

With respect to robustness of GA-based simulation optimization platforms, in this study, we have implemented the model in question using ProModel simulation software and its embedded GA-based optimization software called SimRunner which fits the best our simulation modeling paradigm, and optimization requirements as well. SimRunner uses genetic algorithm which is designed to find near optimal control parameters with respect to a defined objective function. SimRunner turns simulation model into an answer machine to perform sophisticated "*what-if*" analysis and optimization automatically. The ActiveX connection of ProModel and SimRunner enables the automation of creating sophisticated and large-scale simulation optimization models, without manually coding the simulation optimization models giving the best answer possible while saving the most time. Furthermore, Simrunner provides a quick and modular control parameters setting option which is a crucial issue in efficiency of GA-based optimization (Harrell et al. 2004).

It should also be pointed out that, since the genetic algorithm yields near optimal solutions, thus in this study, from "optimal maintenance schedule", in fact we mean "the near optimal maintenance schedule".

In order to set the objective function to evaluate the effects of maintenance schedule on production throughput, as described in the simple designed experiment, the effects of maintenance schedule on the number of fully processed and partially processed parts in the system at the end of planning horizon (shift) constructs the backbone of our objective function. To give a value to each work piece in production line, a method needed to faithfully assign these values to work pieces in each workstation. One of such methods called gravity heuristic proposed by Yang et al. (2007a) using the analogy between production line and gravity field, by minimum spanning tree algorithm used in the graph theory. The response value's objective function is as follows:

$$W(t) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(v_{i,j} \cdot C_{i,j}(t)\right) \tag{4}$$

where $W(t)$ is the response value of a production line at a given time t, $v_{i,j}$ is the value of part type j at station i, $C_{i,j}$ is the buffer content level of part type i at station j, n is the number of workstations inside the system, including all machines and buffers, and m is the number of part types the system is capable to produce. Since in this study only one part type exists, Eq. (4) reduced to following equation:

$$W(t) = \sum_{i=1}^{n} (v_i \cdot C_i(t)) \tag{5}$$

where $W(t)$ is the system value of production line at given time t, n is the number of stations in the system, C_i is the number of parts held in station i, at given time t, and v_i is the part value for a part in station i. The part value v_i for each station should be assigned based on layout of production line. It can be calculated based on the shortest time to finish ST_i for station i. It is the minimum needed processing time to finish a part, starting from station i indeed. Therefore v_i can be calculated by following transformation function:

$$v_i = \max(ST_k) - ST_i \tag{6}$$

If the calculated v_i be divided by $\max(v_i)$, it yields the normalized value of v_i which is in the range of [0, 1]. The system response value represents all of the work done by the production system in given time on the existing parts in the system (including work-in-process and finished parts). The higher system value is, the more performance of maintenance schedule.

Model implementation

In order to use SimRunner, first a simulation model in ProModel must be created. The crucial issues associated with using this simulation optimization tool are selection of control parameters, the objective function and constraints. These are defined as follows:

1. **Objective Function:** The objective function will be to maximize the total throughput. It will include finished goods inventory, and work in process inventory within buffers. As mentioned earlier, it should be in accordance to Eq. 5.
2. **Constraints:** There is a capability in SimRunner to provide some constraints to restrict the solution space. Here, the available range of time within planning horizon to complete the maintenance action on each machine is considered as constraints.
3. **Control Parameters:** The control parameters or the decision variables are the parameters which can be changed from a lower limit to an upper limit. Shutdown time for machines which are subject to repair will be considered as the control parameters. These variables are defined in the SimRunner's "Macro" and "Variable" modules. As described earlier, SimRunner provides modular options to set optimization parameters. In an ad hoc manner, we set all the available optimization parameters on the most precise option given by SimRunner. To do so, we set the convergence percentage on 0.001; the maximum number of generations is set on 100 generations; the number of replication per experiment is set on 100; the confidence level on 99%; the percentage of error on objective function estimation on 1%; and the "Optimization Profile" on "Cautious" mode. Also the MTTRs, cycle times and shutdown variables are set as deterministic values.

Experiments and results

In this section the proposed optimal maintenance scheduling method was tested in seven different simulated scenarios, and finally validated through practicing in a real production system. Scenario 1 is used as benchmark to analyze the dynamics of the production system.

Scenario 1 consider a virtual production line which consists of seven machines and four buffers as shown in Fig. 4. In the beginning of the shift, based one given predicted information about machines condition, machines M_2, M_4, M_5, M_7 are below reliability threshold. Thus, they should be repaired in a near optimal schedule during 8 h shift length. The cycle time for each workstation and the corresponding MTTR for each degraded machine (according to the information from the maintenance database) are described in Table 2. The information about buffers' capacity and their corresponding initial buffer levels distribution remained from previous shift are described in Table 3. In this scenario, half of the capacity of each buffer is full of WIP.

As described earlier in Section "Optimal maintenance scheduling through simulation optimization", the parts value in each workstation should be determined to set the objective function. The part values are calculated according to Eq. (6). In this production line, the $max(ST_k)$ belongs to the first workstation which is equal to total cycle time (115 s). Table 4 illustrates the value of variables used to calculate the normalized values of v_i which are required to set the objective function of problem. According to Eq. (4) the generic form of objective function for this scenario will be as follows:

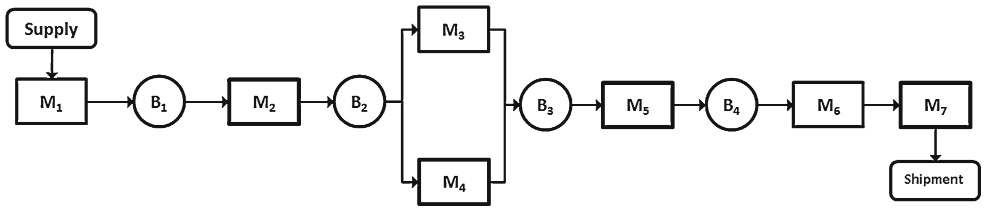

Fig. 4 Layout of production line used in Scenario 1

Table 2 Cycle time and MTTRs for stations in Scenario 1

Station	M_1	M_2	M_3	M_4	M_5	M_6	M_7
Cycle time (s)	15	13	30	30	16	15	16
MTTR (min)	N/A	20	N/A	18	22	N/A	24

Table 3 Buffers' capacity and initial buffer level in Scenario 1

Buffer	B_1	B_2	B_3	B_4
Buffer capacity	100	50	50	100
Initial level	50	25	25	50

Table 4 Part values calculation for Scenario 1

Station	M_1	M_2	M_3	M_4	M_5	M_6	M_7	Finished parts store
ST_i	105	90	77	77	47	31	16	0
v_i	0	15	28	28	58	74	89	105
Normalized v_i	0	0.14	0.26	0.26	0.55	0.70	0.84	1

$$
\begin{aligned}
V &= W(T_e) - W(T_s) = \sum_{i=1}^{n} v_i \cdot C_i(T_e) - \sum_{i=1}^{n} v_i \cdot C_i(T_s) \\
&= [(0.14 \times C_1(T_e)) + (0.26 \times C_2(T_e)) \\
&\quad +(0.55 \times C_3(T_e)) + (0.70 \times C_4(T_e)) \\
&\quad +(1 \times C_7(T_e))] - [(0.14 \times 30) \\
&\quad +(0.26 \times 20) + (0.55 \times 40) + (0.70 \times 10)]
\end{aligned}
$$

where $W(T_s)$ and $W(T_e)$ are the value of the production system in the beginning and at the end of the shift respectively. The reason that the $W(T_s)$ extracted from $W(T_e)$ is to not consider the initial buffer levels which remained from previous shift as the response of production system to the proposed maintenance schedule. The derived near optimal maintenance schedule and its corresponding response of production system are shown in Table 5. In this table SDT_i represents the shutdown time for machine i, $C_i(T_e)$ represents the final amount of buffer at station$i$, and $C_7(T_e)$ represents the number of fully processed parts at the end of the shift. As shown in the table, the highest response value in term of production throughput and final buffer levels distribution is achieved when M_2, M_4, M_5, and M_7 go shutdown on 115th, 343rd, 462nd, and 456th minutes after the beginning of the shift respectively (the shift length is 480 min).

Scenario 2 In this scenario, the same system and set of parameters as in Scenario 1 is used, except that the initial buffer levels varies from Scenario 1. In this scenario, the buffers are empty of initial inventories. The aim is to study the effect of initial buffer levels distribution on their lower limit to the outcome of maintenance schedule and production system dynamics. In this scenario, the parts value in each workstation is the same as Scenario 1. The derived near optimal maintenance schedule for this scenario is shown in the Table 6. It indicates that the best response by production system occurs, when machines M_2, M_4, M_5 and M_7 all go shutdown to be repaired at the starting point of the shift. Comparing the results with Scenario 1 indicates that by evacuating the buffers, more throughputs will be obtained. Furthermore, when distribution of WIP varies, the near optimal maintenance schedule has significantly changed.

Scenario 3 In this scenario, the same system and set of parameters as Scenario 1 is used, except that the initial buffer levels varies from Scenario 1. In this scenario, the buffers are full of initial inventories. The aim is to study the effect of initial buffer levels distribution on their upper limits to the outcome of maintenance schedule and production system dynamics. In this scenario, the parts value in each workstation is the same as Scenario 1. The results as indicated in Table 7 shows that the best performance occurs when M_2, M_4, M_5, and M_7 go shutdown on 345th, 0th, 462nd, and 456th elapsed minutes from the beginning of the shift respectively. Furthermore, when there is not any free space in the buffers, the lowest throughputs is obtained.

Scenario 4 In this scenario again, the same system and set of parameters as in Scenario 1 is used, except that the buffers capacity is different from Scenario 1. In this scenario, the buffers' capacities are two times higher than Scenario 1. The objective of designing this scenario is to study the effect of buffer capacity enlargement on outcome of maintenance schedule and production system dynamics. In this scenario also, the parts value in each workstation is the same as Scenario 1. As shown in Table 8, this change has significantly influenced the maintenance schedule and production sys-

Table 5 Derived results of Scenario 1

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
115	343	462	456	64	50	50	71	1,707	1,743.9

Table 6 Derived results of Scenario 2

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
0	0	0	0	0	28	50	5	1,707	1,745.2

Table 7 Derived results of Scenario 3

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
345	0	462	456	100	50	50	100	1,707	1,707

tem's behavior. As shown in the Table 8, in this scenario, the highest production value has been obtained so far.

Scenario 5 In this scenario again, the same system and set of parameters as in Scenario 1 is used, except that the buffers capacity is different from Scenario 1. In this scenario, the buffers' capacities reduced to the half of the Scenario 1. The objective of designing this scenario is to study the effect of decreasing buffer capacity on outcome of maintenance schedule and production system dynamics. As shown in Table 9, by decreasing the buffers capacities, the production value diminishes, and optimal maintenance schedule changes significantly. The interesting point is that the same value as Scenario 3 has been gained. Thus, the more free space in buffers are, the more productive the system works in its near optimal condition.

Scenario 6 In this scenario, consider that in the same systems as Scenario 1, the mean time to repairs have been doubled. Obviously, because of trade-off between assigned time to maintenance and production time, as shown in Table 10, the production system's response to the near optimal maintenance schedule has significantly been decreased. The optimal maintenance actions in this scenario should be performed on 440th, 444th, 436th, and 432nd minutes after beginning of the shift for machines M_2, M_4, M_5, and M_7 respectively. In contrast, the worst derived schedule yields the production value of 1,457, when M_2, M_4, M_5, and M_7 go shutdown on 440th, 333rd, 0th, and 216th minutes respectively. Logically, the longer MTTRs results in less production value.

Scenario 7 In this scenario, consider that in the same systems as Scenario 1, the mean time to repair has reduced by half. As shown in Table 11, the near optimal maintenance actions in this scenario should be performed at 117th, 353rd, 469th, and 468th minutes after beginning of the shift on machines M_2, M_4, M_5, and M_7 respectively. Moreover, because of dedicating more production time, the highest production value has been obtained.

Validation through practicing in real production system

In this section, in a real case of industry, as shown in Fig. 5, a production system consisting thirteen machines and four buffers with adjustable capacities is used to test the proposed maintenance scheduling method via comparing the estimated response by scheduling algorithm, and the real given response by production line to each set of solutions. Under four different conducted circumstances, first the near optimal maintenance schedules were derived by proposed

Table 8 Derived results of Scenario 4

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
345	343	462	456	0	64	100	71	1,707	1,766.1

Table 9 Derived results of Scenario 5

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
230	458	346	342	50	25	25	50	1,707	1,707

Table 10 Derived results of Scenario 6

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
440	444	436	432	100	50	50	63	1,617	1,653.3

Table 11 Derived results of Scenario 7

SDT_2	SDT_4	SDT_5	SDT_7	$C_1(T_e)$	$C_2(T_e)$	$C_3(T_e)$	$C_4(T_e)$	$C_7(T_e)$	Response value
117	353	469	468	38	50	50	52	1,752	1,772

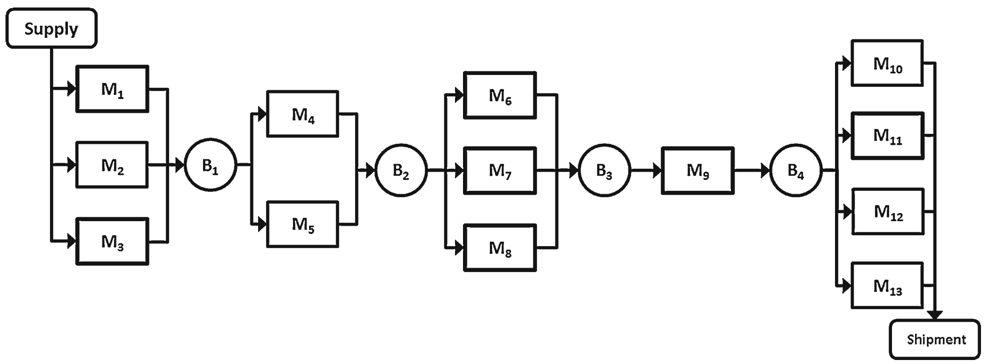

Fig. 5 Layout of production line used as case study

Table 12 Cycle time and MTTRs of stations in the real production system

Station	Cycle time (s)	MTTR (min)
M_1	37	72
M_2	37	N/A
M_3	37	64
M_4	25	N/A
M_5	25	N/A
M_6	36	55
M_7	36	55
M_8	36	55
M_9	11	60
M_{10}	49	N/A
M_{11}	49	42
M_{12}	49	N/A
M_{13}	49	N/A

algorithm, and then the proposed maintenance schedule was practiced in the real production system to measure the accuracy of the proposed maintenance scheduling method. First we explain the circumstances in each experiment, and afterwards the results are presented and compared.

Experiment 1 In this experiment, seven out of thirteen machines are subject to repairs as bolded in the Fig. 5. The cycle time for each workstation and its corresponding MTTR

Table 13 Buffers' capacities and initial level of buffers in Experiment 1

Buffer	B_1	B_2	B_3	B_4
Buffer capacity	100	100	100	100
Initial level	56	64	30	50

Table 14 MTTRs of machines for Experiment 2

Machine name	M_1	M_3	M_6	M_7	M_8	M_9	M_{11}
MTTR (min)	40	25	18	18	18	30	36

(if applicable) are described in Table 12. Also the buffers capacity and their initial level are explained in Table 13.

Experiment 2 In this experiment, the machines go shutdown according to decreased MTTRs given in Table 14, and the rest of conditions are the same as Experiment 1.

Experiment 3 In this experiment, each buffer's capacity is decreased to 65 units. The rest of conditions are the same as Experiment 1.

Experiment 4 In this experiment, each buffer's capacity is increased to 200 units. The rest of conditions are the same as Experiment 1.

Table 15 represents the derived schedule, the real response of system after practicing the proposed schedule by algorithm, and the relative error between former and latter one.

Table 15 The derived schedule from algorithm and experimental results and their relative errors

Experiment	Measures	SDT_1	SDT_3	SDT_6	SDT_7	SDT_8	SDT_9	SDT_{11}	$C_3(T_e)$	$C_5(T_e)$	$C_8(T_e)$	$C_9(T_e)$	$C_{13}(T_e)$	Response value
1	Scheduled	204	0	425	213	213	210	219	48	100	1	31	2,123	2,103.8
	Experimental	204	0	425	213	213	210	219	47	100	1	31	2,124	2,104.6
	Relative error	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.02	0	0	0	0.0005	0.0004
2	Scheduled	220	0	462	231	231	225	222	16	26	1	83	2,293	2,272.5
	Experimental	220	0	462	231	231	225	222	16	26	1	84	2,292	2,272.2
	Relative error	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	0	0	0.01	0.0005	0.0001
3	Scheduled	204	0	425	212	212	210	219	65	65	0	1	2,115	2,065.3
	Experimental	204	0	425	212	212	210	219	65	65	0	1	2,115	2,065.3
	Relative error	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	0	0	1	0	0
4	Scheduled	408	208	213	425	425	210	438	0	107	0	93	2,104	2,117.9
	Experimental	408	208	213	425	425	210	438	0	106	0	91	2,105	2,116.5
	Relative error	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	0.009	0	0.02	0.0005	0.0006

In Experiment 1, the proposed maintenance algorithm suggests that the highest response by production system occurs, when machines M_1, M_3, M_6, M_8, M_9, and M_{11} go shutdown at the beginning of 204th, 0th, 425th, 213th, 213th, 210th, and 219th minutes respectively. After implementing this suggested schedule on real system, almost the same response with relative error of 0.0004 was given. In the Experiment 2, by practicing the proposed schedule, the given response by real production system had only 0.0001 relative errors. In Experiment 3, exactly the estimated response by algorithm yielded by the real system, and in Experiment 4, the amount of relative error was only 0.0006. Like Scenarios 1–4, in the Experiment 1–4, the system had the same behavior toward the similar changes which were imposed to the system. In all of circumstances, the relative error is zero or near to zero, thus it can be ignored.

Conclusions and future work

A method of maintenance scheduling which incorporates the dynamics of production system and uses real-time information about workstations was developed. The method captures the dynamism of production system and uses information about work-in-process and remaining reliability of equipments and uses them in an optimal control algorithm called simulation optimization. The GA-based optimization engine of the algorithm in a reasonable time searches the solution space to find the best combination of performing maintenance actions on degrading machines which the highest response by production system can be yielded.

Furthermore, it is concluded that, the available space to store WIPs (which depends on number of initial buffers and buffers' capacity), as well as the length of MTTRs influence the dynamics of production system, and consequently affect optimal maintenance schedule. The more free space in buffers, or the shorter MTTRs are, the more productive the system works in its optimal condition.

The research indicates that, the proposed method enables productivity improvement of production system and minimizes production losses which happen due to performing maintenance action. Moreover, it considers the benefit of the system rather than an individual machine.

Our current proposed method does not address the stocasticity, while due to randomness nature of many of production systems like the semiconductor industries, more efficient method should be developed to address randomness of such systems. Also other potential important factors which due to maintenance may affect the dynamics of production system should be studied. This will be addressed in our future work.

Acknowledgments The authors would like to thank two anonymous reviewers for their constructive comments on the manuscript. We also would like to thank Dr. Brian Huff from University of Texas at Arlington for his support and advices.

References

Budai, G., Huisman, D., & Dekker, R. (2006). Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society, 57*, 1035–1044.

Chang, Q., Ni, J., Bandyopadhyay, P., Biller, S., & Xiao, G. (2007). Maintenance opportunity planning system. *Journal of Manufacturing Science and Engineering, 129*, 661–668.

Dagpunar, J. S. (1996). A Maintenance model with opportunities and interrupt replacement options. *Journal of Operations Research Society, 47*(11), 1406–1409.

Dietl, C., & Rakowsky, U. K. (2006). An operating strategy for high-availability multi-station transfer lines. *International Journal of Automation and Computing, 3*(2), 125–130.

Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *Journal on Computing, 14*, 192–215.

Fu, M., Glover, F., & April, J. (2005). Simulation optimization: A review, new developments and applications. In *Proceedings of 37th winter simulation conference*, Orlando (pp. 83–95).

Grigoriev, A., Klundert, J., & Spieksma, F. (2006). Modeling and solving the periodic maintenance problem. *European Journal of Operational Research, 172*, 783–797.

Harrell, C., Ghosh, B., & Bowden, R. (2004). *Simulation using ProModel*. McGraw-Hill.

Higgins, A. (1998). Scheduling of railway track maintenance activities and crews. *Journal of the Operational Research Society, 49*, 1026–1033.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: Michigan Press.

Kampen, A. H. C., Buydens, L. M. C., & Strom, C. S. (1996). Lethalization, penalty and repair function for constraint handling in the genetic algorithm methodology. *Chemommetrics and Intelligent Laboratory Systems, 34*(1), 55–68.

Langdon, W., & Treleaven, P. (1997). Scheduling maintenance of electrical power transmission networks using genetic programming. In K. Warwick, A. Ekwue, & R. Aggarwal (Eds.), *Artificial intelligence techniques in power systems*. Stevenage: Institution of Electrical Engineers.

Montgomery, D. C., & Runger, G. C. (2006). *Applied statistics and probability for engineers*. New York: Wiley.

Ni, J., Djurdjanovic, J., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Journal of Computers in Industry, 52*, 476–489.

Nicolai, R. P., & Dekker, R. (2008). Optimal maintenance of multi-component systems: A review. In K. A. H. Kobbacy & D. N. P. Murthy (Eds.), *Complex system maintenance handbook* (pp. 263–286). Springer.

Ólafsson, S. (2006). Metaheuristics. In S. G. Henderson & B. L. Nelson (Eds.), *Handbook in operations research and management science* (pp. 633–654). Elsevier

Papadakis, I., & Kleindorfer, P. (2005). Optimizing infrastructure network maintenance when benefits are interdependent. *OR Spectrum, 27*, 63–84.

Rakowsky, U. K. (2006). Modelling, reliability-adaptive multisystem operation. *International Journal of Automation and Computing, 3*(2), 192–198.

Rogers, P. (2005). Optimum-seeking simulation in the design and control of manufacturing systems experience with OptQuest for Arena. In *Proceedings of 37th winter simulation conference*, Orlando (pp. 1142–1150).

Schutz, J., Rezg, N., & Léger, J. (2009). Periodic and sequential preventive maintenance policies over a finite planning horizon with a dynamic failure law. *Journal of Intelligent Manufacturing, 22*(4), 523–532.

Sheu, S. H., & Jhang, J. P. (1996). A generalized group maintenance policy. *European Journal of Operational Research, 96*, 232–247.

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research, 139*, 469–489.

Yang, Z., Chang, Q., Djurdjanovic, D., & Ni, J. (2007a). Maintenance scheduling for a manufacturing system of machines with adjustable throughput. *IIE Transactions, 39*(12), 1111–1125.

Yang, Z., Chang, Q., Djurdjanovic, D., Ni, J., & Lee, J. (2007b). Maintenance priority assignment utilizing on-line production information. *Journal of Manufacturing Science and Engineering, 129*, 435–446.

Yang, Z., Djurdjanovic, D., & Ni, J. (2008). Maintenance scheduling in manufacturing systems based on predicted machine degradation. *Journal of Intelligent Manufacturing, 19*(1), 87–98.