A semi-Markov heuristic for total productive maintenance

Shuva Ghosh
Abhijit Gosavi (*Email: gosavia@mst.edu*)
Susan Murray
Department of Engineering Management and Systems Engineering
Missouri University of Science and Technology, Rolla, MO, 65409

Abstract

Total productive maintenance (TPM) plays an important role in minimizing costs in production systems. Maintenance scheduling is a critical step in successfully implementing TPM. We use the classical semi-Markov decision process model for developing the schedule, and propose a heuristic for solution purposes that can be implemented in spreadsheet software.

Keywords: total productive maintenance, semi-Markov decision process, vanishing discount.

Introduction

Total productive maintenance (TPM) originated in Japan in 1971 as a method for improving machine availability through better utilization of maintenance and production resources. It is a methodology to maximize the productivity of equipment for its entire life and fosters an environment where improvement efforts in safety, quality, delivery, cost, and creativity are encouraged. The goal is to maximize overall equipment effectiveness and to reduce equipment downtime while improving quality and capacity. TPM involves the entire organization — from operators to senior management — in order to improve equipment utilization.

One major goal of TPM is to reduce the frequency of unexpected failures of machines. Unexpected failures increase lead times for production cycles and thereby the overall costs. Financial savings can be achieved by successfully implementing TPM (Mckone and Weiss 2001; Askin and Goldberg 2002). Some costs are incurred due to preventive maintenance (PM), but PM reduces the significantly higher costs stemming from unexpected failures. Maintenance managers have to strike a balance between PM costs and the cost savings from reduced failures.

Schouten and Vanneste (1995) have developed an optimization model based on Markov decision theory for a production system with buffer capacity. They have proposed a function to estimate the failure probabilities. Das and Sarkar (1999) have developed a reliability-centered PM model for a single product production-inventory system. They have considered random demand arrivals, production times, maintenance times, and repair times. Sloan (2004) has developed a PM model based on Markov decision processes (MDPs) for a single stage production system. They have considered random demand and equipment condition deteriorating with age, which has a negative impact on product yield. El-Ferik and Ben-Daya (2006) have developed an age-based hybrid model where PM is imperfect. In their model, the life of the production system is reduced after each repair or maintenance and the failure rate increases with time. Gosavi et al. (2011) have developed a budget-sensitive model for scheduling maintenance under a total productive maintenance program based on MDPs for a manufacturing system composed of multiple units. Karamatsoukis and Kyriakidis (2012) have developed a maintenance model based on a discrete-time MDP of a production system with multiple intermediate buffers, where production unit deteriorates stochastically with usage. They have also proposed a function similar to Schouten and Vanneste (1995) to model the failure probabilities.

Scheduling planned maintenance activities is very crucial in a successful implementation of TPM. This can be very challenging because every production system has some randomness associated with it. In this paper, we have developed a PM model based on semi-MDPs (SMDPs) for a single machine production system. The distribution of failure time of a machine, the repair time, the maintenance time are some of the random parameters in our model.

Many real-life scenarios, e.g., queueing control, maintenance management, disaster response management, vehicle routing, etc., can be modeled as SMDPs. Dynamic programming (DP) is a very widely used technique to solve MDPs. Value iteration and policy iteration are well-known DP methods. The existing value iteration methods for solving average reward SMDPs cannot be used without a suitable uniformization transformation that we will discuss later. In this paper, hence, we propose an algorithm based on value iteration to solve average reward SMDPs that bypasses the uniformization transformation and uses a “vanishing discount“ approach instead (Ross 1992). Although, in general, our proposed algorithm will be a heuristic (because of an artificially introduced discount factor, which will be discussed later), we have identified the conditions under which the algorithm can generate an optimal solution.

The rest of this paper is organized as follows. In the following section, we provide some background for average reward SMDPs and limitations of existing value iteration algorithms to solve average reward SMDPs. The PM model and the solution technique developed are described in the sections named “PM model” and “Proposed algorithm” respectively. Numerical results are discussed in the section that follows. In the last section, we present conclusions and directions for some future work.

Background for SMDPs

In this section, we begin with some notation for SMDPs. Then we present the SMDP Bellman equation and finally we discuss some limitations of the existing value iteration algorithms for average reward SMDPs.

Notation

Let $\mathcal{S}$ denote the finite set of states in the SMDP, $\mathcal{A}(i)$ the finite set of actions allowed in state i, and $\mu(i)$ the action chosen in state i when policy μ is followed, where $\cup_{i\in\mathcal{S}}\mathcal{A}(i) = \mathcal{A}$. Further let $r(.,.,.) : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \rightarrow \Re$ denote the one-step immediate reward, $t(.,.,.) : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \rightarrow \Re$ denote the time spent in one transition, and $p(.,.,.) : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \rightarrow [0,1]$ denote the associated transition probability. Then the *expected* immediate reward earned in state i when action a is chosen in it can be expressed as: $\bar{r}(i,a) = \sum_{j=1}^{|\mathcal{S}|} p(i,a,j)r(i,a,j)$ and the expected time of the associated transition: $\bar{t}(i,a) = \sum_{j=1}^{|S|} p(i,a,j)t(i,a,j)$. We can now define long-run average reward, or simply average reward, as follows.

Definition 1: Consider a stationary, deterministic policy μ. Then we define:

$$R_\mu(i) \equiv \lim_{k\to\infty} \frac{E_{\hat{\mu}}\left[\sum_{s=1}^{k} \bar{r}(x_s,\mu(x_s))|x_1 = i\right]}{k} \text{ and } T_\mu(i) \equiv \lim_{k\to\infty} \frac{E_{\hat{\mu}}\left[\sum_{s=1}^{k} \bar{t}(x_s,\mu(x_s))|x_1 = i\right]}{k}.$$

For regular Markov chains, from Theorem 7.5 in Ross (1992) (pg. 160), the long-run average reward of a policy μ in an SMDP starting at state i can then be defined as

$$\rho_\mu(i) = \frac{R_\mu(i)}{T_\mu(i)}.$$

For regular Markov chains, $\rho_\mu(.)$ can be shown to be independent of the starting state.

SMDP Bellman equation
The average-reward SMDP revolves around finding the policy μ that maximizes ρ_μ. The optimal average reward will be denoted in this paper by ρ^*. The following is a central result that shows the existence of an optimal solution to the SMDP. The result can be found in any standard text on dynamic programming, e.g., Bertsekas (2000).

Theorem 1: For an average reward SMDP in which all Markov chains are regular, there exists a vector $V \equiv \{V(1), V(2), \ldots, V(|\mathcal{S}|)\}$ and a scalar ρ that solve the following system of equations: For all $i \in \mathcal{S}$,

$$V(i) = \max_{a\in\mathcal{A}(i)} \left[\bar{r}(i,a) - \rho\bar{t}(i,a) + \sum_{j=1}^{|\mathcal{S}|} p(i,a,j)V(j)\right] \tag{1}$$

Further ρ equals ρ^*, the optimal average reward of the SMDP, where the average reward of any policy is defined in Definition (1).

Equation (1) is often called the Bellman optimality equation for SMDPs. The above result paves a way for solving the average-reward SMDP, since it implies that if one can find a solution to the vector V and the scalar ρ^*, then the following policy d is optimal, where

$$d(i) \in \arg\max_{a\in\mathcal{A}} \left[\bar{r}(i,a) - \rho^*\bar{t}(i,a) + \sum_{j=1}^{|\mathcal{S}|} p(i,a,j)V(j)\right] \text{ for all } i \in \mathcal{S}.$$

Limitations of value iteration for average-reward SMDPs
The Bellman optimality equation cannot be used directly in value iteration for average reward SMDPs because of the presence of ρ^*, which is unknown at the beginning. The uniformization technique, in which a uniformized Bellman equation is employed, can be used to avoid this term.

The “uniformized" Bellman optimality equation is as follows:

$$V(i) = \max_{a \in \mathcal{A}(i)} \left[\hat{r}(i,a) + \sum_{j=1}^{|\mathcal{S}|} \hat{p}(i,a,j) V(j) \right] \text{ for all } i \in \mathcal{S}, \text{ where } \hat{r}(i,a) = \frac{\bar{r}(i,a)}{\bar{t}(i,a)}, \tag{2}$$

$$\hat{p}(i,a,j) = \frac{\phi p(i,a,j)}{\bar{t}(i,a)} \text{ if } i \neq j \text{ , and } \hat{p}(i,a,j) = \frac{1 + \phi[p(i,a,j) - 1]}{\bar{t}(i,a)} \text{ if } i = j.$$

In the above equations, $\phi \in \Re$ and the following must hold:

$$0 \leq \phi \leq \frac{\bar{t}(i,a)}{1 - p(i,a,j)} \text{for all } i, j \in \mathcal{S} \text{ and } a \in \mathcal{A}. \tag{3}$$

Note that the uniformized version differs from the classical version in Equation (1). Although it is possible to use value iteration with the transformed Bellman equation, i.e., Equation (2), to solve the SMDP, one must experiment with the condition in Equation (3) to find a suitable value for ϕ. In practice, for any problem with a large number of states, *this adds a thick computational layer to the algorithm’s numerical effort*. Another approach to solve the average reward SMDP is to use policy iteration which is more complex than value iteration.

PM Model

In this section, we will provide the details of the PM problem that we have studied. We have developed a maintenance model for a single machine production system. The output of our model is the optimal maintenance schedule plan for the machine. Most of the dynamic maintenance models in the literature are generally based on MDPs/SMDPs. We have developed a dynamic maintenance model based on SMDPs. The goal is to reduce the frequency of machine failures. The PM is a concept where a working machine is shut down after specific time interval, and then the machine is maintained. This reduces failures during production. The probability of machine failures increases with the cumulative number of production cycles that occur without any repair or maintenance. A production cycle ends when the machine produces one unit. We assume the production time to be deterministic. The mean values of maintenance and repair times are some multiples of the production time (Gosavi et al. 2012). Fixed costs are associated to each maintenance and to each repair. A fixed amount of profit is associated to each unit produced. Some notations are provided next.

C_m =Maintenance Cost; C_r =Repair Cost; T_p =Production Time (per unit); P_u = Profit per unit;
M_m =Co-efficient for Mean Maintenance Time; M_r = Co-efficient for Mean Repair Time.

As in Schouten and Vanneste (1995) and Karamatsoukis and Kyriakidis (2012), we propose a function of time (# of cumulative production cycle) to estimate the probability of failure of the machine. The function is as follows:

P(failure between dth and $(d+1)$th production cycle)=1-ψ^d, where d is the number of cumulative production cycle completed since the last maintenance or repair, and ψ is a constant.

State and action space

The state of the system in our model is the number of cumulative production cycle completed since the last maintenance or the last repair. The number of production cycles ranges from 0 to 99. Hence, the total number of states is 100. After each production cycle, a decision has to be made:

Produce or Maintain. Hence, there are two actions in the action space. Action 1 is to produce, and Action 2 is to maintain. For our numerical experiments, our assumption is that the system will return to the State 1 after each maintenance or repair. We also assume that the machine will fail for sure after 99th production cycle.

Possible transitions
We are going to describe different possible transitions next. If in State 1, Action 1 is chosen, the system can only go to State 2 after successful production or it can remain in State 1 due to failure. From any other State i, where $i \in \{2, 3, \ldots, 99\}$, the system can go to either State $(i + 1)$ after successful production or to State 1 due to failure. From State 100, the system goes back to State 1 after transition.

Transition probability matrix (TPM)
$TPM(i, a, j)$ denotes the transition probability of the system going to state j from state i under action a. The TPM can be expressed in terms of the failure function stated earlier. The expressions are as follows:
For $i \in \{1, 2, \ldots, 99\}$, $a = 1$: $TPM(i, a, 1) = 1 - \psi^{i-1}; TPM(i, a, i + 1) = \psi^{i-1}; TPM(100, a, 1) = 1$
For $i \in \{1, 2, \ldots, 100\}$, $a = 2$: $TPM(i, a, 1) = 1$

Transition reward matrix (TRM)
$TRM(i, a, j)$ denotes the transition reward of the system going to state j from state i under action a. The TRM can be expressed in terms of profit, maintenance and repair cost. The expressions are as follows:
For $i \in \{1, 2, \ldots, 99\}$, $a = 1$: $TRM(i, a, 1) = -C_m; TRM(i, a, i + 1) = P_u; TPM(100, a, 1) = -C_m$
For $i \in \{1, 2, \ldots, 100\}$, $a = 2$: $TRM(i, a, 1) = -C_r$

Transition time matrix (TTM)
$TTM(i, a, j)$ denotes the time taken by the system in going to state j from state i under action a. The TTM can be expressed in terms of the production time, coefficients for maintenance and repair times. The expressions are as follows:
For $i \in \{1, 2, \ldots, 100\}$, $a = 1$: $TTM(i, a, 1) = M_r * T_p; TTM(i, a, i + 1) = T_p$
For $i \in \{1, 2, \ldots, 100\}$, $a = 2$: $TTM(i, a, 1) = M_m * T_p$

Proposed Algorithm

We now present our proposed value iteration algorithm for solving average reward SMDPs. It seek to overcome the limitations of the existing value iteration algorithms. As stated earlier, our new algorithm is based on the vanishing discount approach (Ross 1992) which is discussed in MDP theory. Our proposed algorithm uses a discount factor. We use this factor, which makes our approach heuristic, because it allows for a unique solution of Equation 1, thereby making our algorithm robust. The algorithm is also based on the idea of step sizes used in artificial intelligence and machine learning (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998; Gosavi 2003).

Steps in the vanishing discount algorithm
In this subsection, we present step-by-step description of the proposed algorithm.
Step 1. Set number of iterations, k, to 1, $\epsilon = 0.001$, and $\rho^k = 0$, where ρ^k is the estimate of the optimal average cost in the kth iteration. Also set $Q^k(i, a) = 0$ and $V^k(i) = 0$ for all $i \in \mathcal{S}$ and all $a \in \mathcal{A}$. Select a scalar η, where $0 < \eta < 1$.

Step 2. Update $Q^k(i,a)$ for all $i \in \mathcal{S}$ and all $a \in \mathcal{A}$ as follows:

$$Q^{k+1}(i,a) = [1-\alpha^k]Q^k(i,a) + \alpha^k \left[\bar{r}(i,a) - \rho^k \bar{t}(i,a) + \eta \sum_{j=1}^{|\mathcal{S}|} p(i,a,j) \max_{b \in \mathcal{A}(j)} Q^k(j,b) \right].$$

Step 3. Compute $\mu(i)$ for all $i \in \mathcal{S}$: $\mu(i) = \arg\max_{a \in \mathcal{A}(i)} Q(i,a)$.
Step 4. Compute the steady-state probabilities of the Markov chain of the policy μ using the well-known invariance equation (see e.g., Ross (1997)).

$$\sum_{i=1}^{|\mathcal{S}|} \Pi_\mu^k(i) p(i,\mu(i),j) = \Pi_\mu^k(i) \text{ for every } j \in \mathcal{S}, \text{ and } \sum_{j=1}^{|\mathcal{S}|} \Pi_\mu^k(j) = 1,$$

where $\Pi_\mu^k(i)$ is the the steady-state probability of i associated with μ in the kth iteration.
Step 5. Update ρ^k using the following equation.

$$\rho^{k+1} = [1-\beta^k]\rho^k + \beta^k \zeta^k, \text{ where } \zeta^k = \frac{\sum_{i=1}^{|\mathcal{S}|} \Pi_\mu^k(i)\bar{r}(i,\mu(i))}{\sum_{i=1}^{|\mathcal{S}|} \Pi_\mu^k(i)\bar{t}(i,\mu(i))}.$$

Step 6. Update $V(i)$ for all $i \in \mathcal{S}$ using: $V^{k+1}(i) = \max_{a \in \mathcal{A}(i)} Q^{k+1}(i,a)$.
Step 7. Set $k \leftarrow k+1$. If $||V^{k+1}(i) - V^k(i)||_\infty < \epsilon$, return μ as the policy and stop; otherwise, return to Step 2.

Conditions: In our proposed algorithm, we need to solve the following version of the Bellman equation:

$$V(i) = \max_{a \in \mathcal{A}(i)} \left[\bar{r}(i,a) - \rho \bar{t}(i,a) + \eta \sum_{j \in \mathcal{S}} p(i,a,j) V(j) \right]. \quad (4)$$

where ρ is assumed to be a constant.

From the theory of discounted reward MDPs, we can easily prove that the above equation has a unique solution when $0 < \eta < 1$. If we replace ρ by ρ^*, as η tends to 1, Equation (4) tends to the Bellman equation for SMDPs, i.e., Equation (1) (Ross 1992). Hence, if the value of η is set close to 1, Equation (4) approximates (behaves like) the Bellman optimality equation for SMDPs. For our proposed algorithm to generate an optimal solution, we need two assumptions on η and the step sizes, which are as follows:

Assumption 1: There exists a value for $\bar{\eta}$ in the interval (0, 1) such that for all $\eta \in (\bar{\eta}, 1)$, the unique solution, V, of Equation (4) with $\rho \equiv \rho^*$ produces a policy d defined as follows

$$d(i) \in \arg\max_{a \in \mathcal{A}(i)} \left[\bar{r}(i,a) - \rho^* \bar{t}(i,a) + \eta \sum_j p(i,a,j) V(j) \right]$$

for all $i \in \mathcal{S}$, whose average reward equals ρ^*.

Assumption 2: The step sizes α and β must satisfy the following:

$$\lim_{k\to\infty}\sum_{l=1}^{k}\alpha^l=\infty;\quad \lim_{k\to\infty}\sum_{l=1}^{k}\beta^l=\infty;\quad \lim_{k\to\infty}\frac{\beta^k}{\alpha^k}=0.$$

We conjecture (a detailed proof will be subject to future research) that when $\eta > \bar{\eta}$ (Assumption 1), and Assumption 2 holds, using standard convergence theory (Bertsekas 2000), it will be possible to establish convergence to optimal policy. Assumption 1 will essentially guarantee that Equation (4) will also generate an optimal solution.

Numerical Results

This section is devoted to numerical results with our proposed algorithm. First, we have provided numerical results for some test cases to verify our proposed algorithm. Then, we have provided numerical results for the maintenance management problem.

Test cases

Our test cases, i.e., Cases 1, 2, 3, and 4, having ten states and two actions, are drawn from Gosavi et al. (2012) (see Appendix and Table 7 in Gosavi et al. (2012)). We used the following values in our numerical calculations:
$\epsilon = 0.001$; $\eta = 0.99$; $\alpha = 0.90$; and $\beta = \frac{10}{100+k}$, where k is the number of iterations. The steps sizes satisfy Assumption 2.

The optimal policies, values of ρ^*, and the required number of iterations for convergence for all the cases are presented in Table 1; our proposed algorithm, the adaptive critic algorithm, and the Q-learning method (see Gosavi et al. (2012) for description of both) generate identical policies in each of the four cases. Figure 1 shows the gradual convergence of our proposed algorithm for Case 1.

Table 1: *Optimal Policy, Average Reward and Number of Iterations*

Case	*Policy*	ρ^*	*# of Iterations*
1	(2, 1, 2, 1, 1, 2, 2, 2, 2, 2)	0.35	525
2	(2, 1, 2, 1, 1, 2, 2, 2, 2, 2)	0.33	521
3	(2, 1, 2, 1, 1, 2, 2, 2, 2, 2)	0.28	502
4	(2, 1, 2, 1, 1, 2, 1, 2, 2, 1)	0.32	482

It is interesting to note that we found the algorithm to converge to the optimal policy for any value of η, where $\eta \in (0, 1)$, in each of the four cases. However, we also found, experimentally, on some problems from Kulkarni et al. (2011) that in general, the algorithm does *not* converge to the optimal policy if η is below the threshold $\bar{\eta}$. These cases are numbered 5 through 8 in our experiments (numbered 1 through 4 in Kulkarni et al. (2011)). Our values for $\bar{\eta}$ for Cases 5 through 8 coincide with those determined in Kulkarni et al. (2011) (see Table 1 of Kulkarni et al. (2011)); those values of $\bar{\eta}$ are presented in Table 2.

Maintenance management problem

We have developed four cases for the maintenance management model that we have studied. We present the input parameters in Table 3. The values for C_M, C_R, T_p, M_m, and M_r for Case 1 are taken from Gosavi et al. (2012) (see Table 5 of Gosavi et al. (2012)).

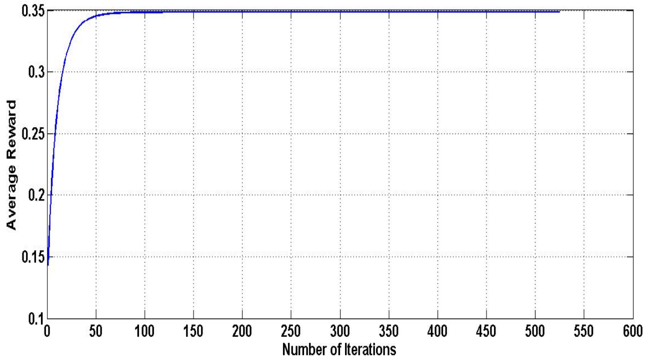

Figure 1: *Convergence for the vanishing discount algorithm for Case* 1.

Table 2: *Values of* $\bar{\eta}$

Case	$\bar{\eta}$
5	0.93
6	0.77
7	0.65
8	0.98

Table 4 provides the number of iterations to reach an optimal solution. Optimal policies and average rewards for all the cases are presented in Table 5. Optimal policy denotes the number of production cycles after which the machine needs to be maintained. We have also run the policy iteration algorithm, which is guaranteed to generate optimal solution, to gauge the quality of the results generated by our proposed algorithm. It produces the same policies as the policy iteration algorithm. Moreover, the computational time is less with our proposed algorithm.

The algorithm was run on an Intel Pentium Dual Core Processor with 2.00 GHz speed. The RAM size was three GB. For the test cases, it took on average five seconds on each case. For the preventive maintenance cases, it took on average eight seconds for each case. Policy iteration took on average fifteen seconds for each case. Although the code was built in MATLAB, it can just as easily be implemented on a spreadsheet software, e.g., Microsoft Excel. We also solved the baseline case using exhaustive enumeration, which took five minutes. This suggests that our algorithm will provide significant savings of computational time on larger problems.

Table 3: *Input Parameters*

Case	C_M	C_R	T_p	M_m	M_r	ψ	P_u
1	2	10	10	1.25	3	0.95	7
2	4	6	8	1	2	0.90	6
3	3	5	8	1.25	3	0.95	5
4	5	6	6	1	1.5	0.90	6

Table 4: *Number of Iterations*

Case	*# of Iterations*
1	433
2	260
3	368
4	271

Table 5: *Optimal Policy and Average Reward*

Case	*# of Production Cycles*	*Average Reward*
1	3	0.39
2	4	0.13
3	5	0.52
4	6	1.02

Conclusions

The problem of determining the optimal maintenance interval in TPM is often solved via MDP and SMDP models. Policy iteration is an accepted method for solving average reward SMDPs. In this paper, we propose a new heuristic based on value iteration that produces encouraging results, performing faster than policy iteration. We have solved a preventive maintenance problem with our proposed algorithm which produces optimal results. Some directions for future research are as follows: First, establishing convergence proofs for the algorithm should be an interesting avenue to pursue. Second, we propose to implement our algorithm on a complex real-world problem having a large state-action space.

References

Askin, R., J. Goldberg. 2002. *Design and Analysis of Lean Manufacturing Systems*. John Wiley and Sons, New York, USA, 1st edition.

Bertsekas, D.P. 2000. *Dynamic Programming and Optimal Control*. Athena, Belmont, 2nd edition.

Bertsekas, D.P., J.N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena, Belmont.

Das, T.K., S. Sarkar. 1999. Optimal preventive maintenance in a production inventory system. *IIE Transactions* 31:537-551.

El-Ferik, S., M. Ben-Daya. 2006. Age-based hybrid model for imperfect preventive maintenance. *IIE Transactions* 38: 365-375.

Gosavi, A. 2003. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Springer, New York, USA.

Gosavi, A., S. Murray, J. Hu, S. Ghosh. 2012. Model-building adaptive critics for semi-Markov control. To appear in *Journal of Artificial Intelligence and Soft Computing Research*.

Gosavi, A., S. Murray, V. Tirumalasetty, S. Shewade. 2011. A budget-sensitive approach to scheduling maintenance in a total productive maintenance (TPM) program. *Engineering management Journal* 23(3):46-56.

Karamatsoukis, C., E. Kyriakidis. 2012. Optimal maintenance of a production system with L intermediate buffers. *Mathematical Problems in Engineering* 2012.

Kulkarni, K., A. Gosavi, S. Murray, K. Grantham. 2011. Semi-Markov adaptive critic heuristics with application to airline revenue management. *Journal of Control Theory and Applications* 9(3);421-430.

McKone, K.E., E.N. Weiss. 2001. *Innovations in Competitive Manufacturing*. Springer, New York, USA.

Ross, S.M. 1992. *Applied Probability Models with Optimization Applications*. Dover, New York, USA.

Ross, S.M. 1997. *Introduction to Probability Models*. Academic Press, San Diego, CA, USA.

Schouten, F.V.D.D, S. Vanneste. 1995. Maintenance optimization of a production system with buffer capacity. *European Journal of Operational Research* 82(2):323-338.

Sloan, T. 2004. A periodic review production and maintenance model with random demand, deteriorating equipment, and binomial yield. *Journal of the Operational research Society* 55(6):647-656.

Sutton, R., A. Barto. 1998. *reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, USA.