MAINTENANCE OPTIMIZATION OF MULTIPLE COMPONENT SYSTEMS USING PROBABILISTIC GRAPHICAL MODELS

by

İpek Kıvanç

B.S., Industrial Engineering, Istanbul Technical University, 2016

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2020

ACKNOWLEDGEMENTS

I would like to thank my thesis advisors Prof. Taner Bilgiç and Asst. Prof. Demet Özgür Ünlüakın who have provided full support from the date we met and started to work together, a depth of gratitude. Their contribution to success of scientific studies we accomplished together and their effort for keeping pace with my complex studying order are unforgettable. I express my gratitude towards Prof. S. Çağlar Aksezer for all his supports and helps. I thank Busenur Türkali and Ayşe Karacaörenli we worked together in Işık University and provided full support to me, for their helps and supports.

This work is also supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant no 117M587.

I particularly owe a great deal to Müge Ertürk who has encouraged and supported me during my educational life. I express my gratitude to Berivan Işık, Peter Teunissen, Cihan Uyanık, Uğur Cucu that we lived and proceeded together during my education life, to Uğurcan Acar who gave all moral support during my application to Boğaziçi University.

I deeply thank Barış Bora who witnessed to all processes of the thesis and never refrained to support.

My family also provided invaluable support for this work. I would like to thank specially to my brother Ömer Cihan Kıvanç who supported and guided me in all my life. I thank my mother Zeliha Kıvanç who is so valuable for me and irreplaceable, for her beliefs and supports to me at my all difficult times. My father Burhan Kıvanç was not able to see this work completed, however his memory was with me all the way to the end.

ABSTRACT

MAINTENANCE OPTIMIZATION OF MULTIPLE COMPONENT SYSTEMS USING PROBABILISTIC GRAPHICAL MODELS

Maintenance optimization is a difficult task in today's manufacturing environment, especially when the system has multiple components. Thus, it is essentially critical to cope with the uncertainty and the complexity of the systems while deciding on the correct maintenance actions. Taking maintenance decisions in a planning horizon is one of the well-known stochastic sequential decision problems under uncertainty.

Partially Observable Markov Decision Processes (POMDPs) are powerful tools for such problems under uncertainty in partially observable stochastic environments. However, since their state spaces can quickly explode with the increasing number of variables, POMDPs may not be preferable for addressing maintenance problems of multi-component systems. Factored representations are used for POMDPs by exploiting the inherent factored structure of the problem. This study aims to demonstrate how to formulate the maintenance problem of systems consisting of partially observable deteriorating components using factored POMDPs on two maintenance problems. The first one is an experimental model to perform in depth sensitivity analyses and to compare with some predefined policies proposed in the study. The second model belongs to a real-life implementation in thermal power plants. Sensitivity analyses are conducted under various scenarios with several settings. The results show that factored POMDPs are advantageous in modeling, solving and analyzing of maintenance problems with multi-components. Furthermore, the generated factored POMDP policies perform considerably better than the myopic policies.

ÖZET

OLASILIKLI GRAFİKSEL MODELLERLE ÇOK BİLEŞENLİ SİSTEMLERİN BAKIM ENİYİLEMESİ

Bakım optimizasyonu, çok bileşenli sistemler için günümüzün üretim dünyasında zor bir görevdir. Bu nedenle, doğru bakım eylemlerine karar verirken sistemlerin belirsizliğini ve karmaşıklığını dikkate almak kritik öneme sahiptir. Bir planlama ufkunda bakım kararları vermek, belirsizlik durumunda stokastik sıralı karar verme problemlerinden biridir.

Kısmen Gözlenebilir Markov Karar Süreçleri (POMDPler), kısmen gözlemlenebilir stokastik ortamlarda belirsizlik altındaki problemler için güçlü araçlardır. Bununla birlikte, durum uzayları değişken sayısı ile hızlı bir şekilde büyüyebildiğinden, çok bileşenli sistemlerin bakım problemlerini ele almak için POMDPler tercih edilmeyebilir. Bu tarz problemler için, problemin doğasında halihazırda mevcut olan faktörlü yapıdan yararlanarak faktörlü POMDPler kullanılır. Bu çalışma, iki ayrı bakım probleminde faktörlü POMDPler kullanılarak kısmen gözlemlenebilir dinamik bileşenlerden oluşan sistemlerin bakım probleminin nasıl formüle edileceğini göstermeyi amaçlamaktadır. İlki, detaylı duyarlılık analizleri yapmak ve çalışmada önerilen bazı tanımlanmış politikalarla karşılaştırmak için deneysel bir modeldir. İkinci model termik santrallerde gerçek hayattaki bir uygulamaya aittir. Duyarlılık analizleri, çeşitli ayarlarla çeşitli senaryolar altında gerçekleştirilmiştir. Sonuçlar, faktörlü POMDPlerin çok bileşenli bakım sorunla-rının modellenmesinde, çözümünde ve analiz edilmesinde avantajlı olduğunu göstermek-tedir. Ayrıca, oluşturulan faktörlü POMDP politikaları miyop politikalardan önemli ölçüde daha iyi performans göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	Action state space
a	Action
B	Belief space
b	Belief
b'	Next belief
Ci	Component i
f	Frequency of the proactive maintenance
o	Observation
r	Reward
S	State space
s	State
s'	Next state
T	Transition function
V	Value Function
X	Random variable
α	Alpha-vector
γ	Discount factor
θ	Observation state space
π	Policy

LIST OF ACRONYMS/ABBREVIATIONS

BB	Ball Bearing
BDD	Binary Decision Diagram
BN	Bayesian Network
CR	Coal Rank
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
DC	Downtime Cost
DP	Dynamic Programming
DT	Decision Tree
FIB	Fast Informed Bound
GH	Games-Howell Post-Hoc Test
HC	Honeycomb
HRG	Hub Reduction Gear
HSVI	Heuristic Search Value Iteration
LMDP	Latent Markov Decision Process
MDP	Markov Decision Processes
MLS	Most Likely State
QMDP	Markov Decision Processes with Q-functions
PBVI	Point-Based Value Iteration
PERSEUS	Randomized Point-Based Value Iteration for POMDPs
PGM	Probabilistic Graphical Model
POMDP	Partially Observable Markov Decision Process
PS	POMDP-solve
RAH	Regenerative Air Heater
RAH Exit	RAH Exit Temperature
RAH Rot	RAH Rotation
RAH Temp	RAH Measured Temperature
RI	RAH Insulation

RR	Rotor Rotation
RS	Rotor Shaft
SARSOP	Successive Approximations of the Reachable Space Under Optimal Policies
SPJ	Symbolic Perseus re-implemented in Java
SPM	Symbolic Perseus implemented in Matlab
WI	Winding Insulation

1. INTRODUCTION

Maintenance optimization and maintenance strategy selection play an essential role in cost optimization with multi-component complex systems. All kinds of systems used in the manufacturing sector, such as machines, equipment, vehicles have a life cycle, and they need to be maintained during their life cycles. As technology evolves, systems have had a more complex structure employing evolving automated technologies and industry, which has increased maintenance costs. Coping with increased maintenance costs requires efficient and effective maintenance planning. Therefore, the importance of the development of effective maintenance strategies has increased.

1.1. Basic Concepts of Maintenance Management

A concise taxonomy of the maintenance strategies is presented in Figure 1.1 [1]. The maintenance strategies can be classified into two categories based on the repair timing: reactive maintenance and proactive maintenance [2].

In reactive or unplanned maintenance, the maintenance activity is performed after a fault, breakdown, or stop occurs to bring the component to its normal operating condition. Reactive maintenance can be divided into two subgroups: corrective maintenance and emergency maintenance. Corrective maintenance is achieved to bring the system to a functional level after a fault has occurred in the system. In corrective maintenance, there is no further effort to keep equipment running in optimal condition. Emergence maintenance is an immediate maintenance activity to prevent unexpected and severe consequences [3]. Although reactive maintenance does not require any investment cost (installation of measurement systems), unplanned production downtime results in increased costs. This method, therefore, is suitable for systems with non-critical equipment or where system performance is not entirely dependent on the reliability of any equipment [4].

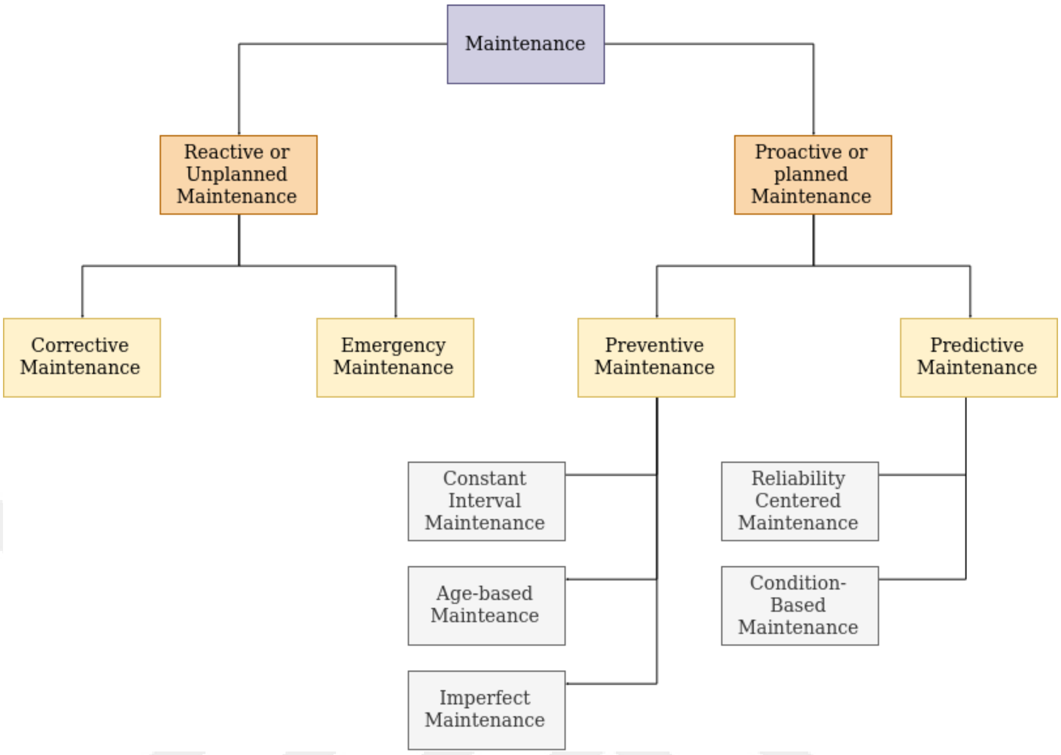


Figure 1.1: Taxonomy of maintenance strategies [1].

Over time, traditional unplanned or reactive maintenance could not meet the demands of effective maintenance and is replaced by proactive maintenance. In proactive maintenance, maintenance activities are scheduled without waiting for system failure. However, unnecessary maintenance activities can result in higher maintenance costs. Thus, to avoid high maintenance costs, the maintenance activity based on observed measurements has come to the forefront. In other words, acting with the "right maintenance principle at the right time" is the first step towards developing more effective maintenance policies. Proactive maintenance activity is examined under two main headings: preventive maintenance and predictive maintenance.

In preventive maintenance, the maintenance activities are achieved at specified intervals based on the probability of the equipment failure. Preventive maintenance is a type of scheduled maintenance activity, intending to improve equipment life and reduce unplanned downtime. Regularly scheduled maintenance helps to avoid possible failures and it results in lower maintenance costs. The downside of preventive maintenance is an unnecessary shutdown or replacement of working equipment. Furthermore, preventive maintenance cannot eliminate reactive repairs [5].

Preventive maintenance can be grouped into three categories: constant-interval maintenance, age-based maintenance, and imperfect maintenance, as shown in Figure 1.1. Constant-interval maintenance is the repair of the system or components at pre-determined time intervals. The intervals are determined based on the system failure time. The components are checked regularly, and any identified situation or condition in the system that may cause failure is rectified. In the case of system failure out of these pre-determined times, corrective maintenance is also performed [6].

Age-based maintenance is performed at the time of a specific age t. If the component fails before reaching a certain age t, corrective maintenance is performed and the coming maintenance is planned after t unit. Thus, the number of maintenance intervals is reduced when compared to the constant-interval maintenance [7]. Imperfect maintenance considers that the system state lies somewhere between the best state and the worst state. The system will not move to its best state after maintenance, but it will be younger. Imperfect maintenance considers the uncertainty of the current state of the component when planning the next actions. The maintenance of deteriorating systems is often imperfect [8]. The minor maintenance activity discussed in Chapter 5 is an example of imperfect maintenance. There, minor maintenance involves small maintenance activities that only require labor costs. It requires much less cost than perfect maintenance; however, it does not provide a gu arantee for improvement.

Predictive maintenance differentiates from the preventive maintenance in scheduling the maintenance activities. In the preventive maintenance, maintenance activities are achieved on predetermined schedules, whereas in the predictive maintenance, maintenance activities are adaptively determined. Predictive maintenance is divided into two types as reliability-centered maintenance and condition-based maintenance.

Reliability-centered maintenance concentrates on maintaining the functional capability of the equipment with respect to cost, safety or environmental goals [9]. The primary purpose is to minimize maintenance costs by focusing on the essential functions of the system and eliminating unnecessary maintenance [10]. Condition-based maintenance is a strategy in which maintenance decisions are determined by monitoring the system state. That is to say, in condition-based maintenance, decisions are made based on the measurements taken from continuously monitored system parameters [11]. The benefits of condition-based maintenance are an early warning of possible failures, improved accuracy in failure prediction and effective inventory control. However, an obvious disadvantage of performing condition-based maintenance is employing monitoring equipment and difficulty of managing decision-making strategies [1]. The main focus of the thesis is to generate effective maintenance policies for the maintenance optimization problems of dynamic systems under the condition-based maintenance strategies, using factored POMDPs.

1.2. Maintenance of Multi-Component Systems

Maintenance optimization is a difficult task in today's manufacturing environment, especially when the system has multiple components, and there exists uncertainty. As technology evolves, systems become more complex, having more components and thus causing the plan of the maintenance activities harder. That is why maintenance management is an essential and critical process for companies with multi-component systems. Since there are different maintenance procedures for each component, selecting a maintenance procedure for the entire system is complicated. Modeling the dependencies among the components is an essential point while studying the maintenance of a system because they directly affect the maintenance policy. Since each piece of equipment can have several maintenance procedures, it is difficult to choose the appropriate maintenance strategy for the entire system. Besides, by increasing the number of components affecting each other, the maintenance optimization problem is getting more difficult. Therefore, the interdependent structure among the components of the system should be defined correctly.

A multi-component system has three types of dependencies: stochastic, structural and economic [12]. Stochastic dependency refers to the effect of the degradation or failure of a component on the lifetime distribution of other components. The stochastic dependency of multi-component systems can be examined in three different types [12]. Type I failure interaction means that the failure of a component can trigger the failure of the other components a probability [13]. In Type II failure interaction, if a component has a failure, this affects the failure rate of the other components [14]. In Type III failure or shock damage interaction, failure of a component causes a random amount of damage (shocks) on the other components [15]. Structural dependency requires the replacement of the working components simultaneously, even if some of them are not in their fail state [16]. Structurally dependent components cannot be repaired independently of each other. Therefore, this is not a dependency on failure. It is a dependency on the repair. If there is economic dependency among components when the components are maintained together, savings or surpluses may incur the total maintenance cost compared with the total of individual costs. In general, economic dependency refers to the joint maintenance of some parts that can be economical than performing them disjointly [17].

1.3. Solution Approaches to Maintenance Problems

Papakonstantinou [18] defines five approaches for categorizing maintenance procedures of aging construction infrastructures. Inspired by this study, it is possible to split maintenance planning approaches into five types of predictive maintenance strategy:

The first group includes approaches based on the simulation of various predetermined policies. According to the simulation results, the scenario with the best performance which has the minimum cost or cost/benefit ratio is selected. The disadvantage of this approach is that although it performs better than other alternatives, the selected policy is hard to be optimal.

The second category has pre-determined safety or risk threshold approaches. The fundamental logic of this approach is to perform maintenance when the simulation model reaches the reliability or risk threshold. In these approaches, the system component and unit breakdown ratio are assumed beforehand. Accordingly, reliability or risk threshold is determined. Thus, optimum maintenance duration is set to keep the reliability above a predetermined threshold. The third category is the approach where default reliability threshold policies are determined with the optimization techniques. In these operations, after identifying maintenance time based on reliability threshold policies, maintenance activities are recommended by using optimization techniques.

The fundamental idea in the fourth category approach is to simulate the disruption based on a steady-state stochastic model. The optimum threshold value is determined for each maintenance activity under defined target function. Perfect inspections are assumed and related activities will take place when a condition state is passed throughout the structure life-cycle. This approach's main weakness is that the system cannot be fully observed with perfect observations and the unrealistic assumption for various real-life problems. Besides, it may be challenging to find the optimal global solution in non-convex areas.

The main interest in this thesis lies in the fifth category and it covers models based on stochastic control and optimal sequential decisions. Generally, these approaches are on a discrete state-space and determine the best maintenance activities by considering real-time data. An MDP is a formulation of the sequential decision problem involving uncertainty where the underlying system state is fully observed [19]. However, MDP models are inadequate, especially for real-life problems where the system state is not completely observed. In such systems, the system state can be estimated via observations that are collected with some signals or measurements. However, the accuracy of these observations is also probabilistic. That is why MDPs have been extended to POMDPs by providing information regarding the system's state through noisy sensors.

1.4. Probabilistic Graphical Models

Probabilistic graphical models (PGMs) are powerful frameworks for encoding probability distributions over complex domains under uncertainty. PGMs intersect graph theory and probability theory and provide a flexible framework to model large numbers of random variables interacting with each other. This section introduces the fundamental concepts of the several types of probabilistic models which can be used for representing deterioration and dependencies among system components such as Bayesian networks (BNs), dynamic Bayesian networks (DBNs), Markov decision processes (MDPs), partially observable Markov decision processes (POMDPs) and factored partially Markov decision processes (Factored POMDPs).

Bayesian Networks are used to describe causal relations between the elements of the system using conditional probability distributions. BNs are directed acyclic graphs (DAGs) in which the variables are represented by nodes and the conditional relationships between variables are represented by directional arrows [20]. BNs consist of two main parts: The qualitative part specifies the probabilistic dependencies between the variables represented as a graph, and the quantitative part defines the conditional relationships identified in the qualitative part. BNs are used to explain the dependency between the components of the system that has failed in decision-theoretic trouble-shooting problems [21]. Reliability is another area where BNs are used frequently [22]. Dynamic Bayesian networks (DBNs) are extended BNs, including the temporal dimension [23]. A DBN is composed of a series of time slots, each of which includes the same BN to represent dynamic behavior between random variables. Temporal arrows indicate the temporal probabilistic dependencies between variables in different time slices. The graphical structure provides an easier way of specifying this conditional independence. Recently DBNs are also used for decision-theoretic trouble-shooting problems, dependability and maintenance domains [24].

The Markov decision processes (MDPs) provide a basis to model sequential decision-making problems in probabilistic domains [25]. MDPs are the extensions of

Markov chains by a number of actions, and state-based rewards. In an MDP model, at each step, an agent in a stochastic world observes the state in the environment as input and determines the action to be selected as the output. The action performed by the agent returns a reward and at the same time provides a stochastically transition to a new state.

Partially observable Markov decision processes (POMDPs) are the extensions of MDPs, where the state of the system is not fully observable. A POMDP provides a framework for modeling probabilistic decision-making problems under partial observability. In such systems, the system state can be estimated via observations that are collected with some signals or measurements [18]. POMDPs are applied widely in robotics, health informatics, artificial intelligence and maintenance domains [26].

POMDPs are limited to solve problems with large state spaces. Factored POMDPs are variants of POMDPs where variables are represented compactly via data structure representations such as decision trees (DTs) [27] and algebraic decision diagrams (ADDs) [28] to reduce the computational complexity. ADDs have conditional independence and context-specific independence that allow representing probability and utility tables in smaller sizes [29]. Factored POMDPs have been widely used to design dialogue management, assistive technology and human-robot interaction and active sensing [30]. Factored POMDPs are particularly suited for maintenance problems of multi-component systems because of the inherent factored structure of such problems [26].

1.5. Motivation and Contribution of the Thesis

In recent years sequential decision-making problems under decision making have been studied extensively in maintenance problems. Maintenance optimization of multi-component systems along a planning horizon is a complex problem because it requires a joint decision of when to make maintenance and what to do at that maintenance time. Since these two decisions are dependent, it is hard to define a predefined policy for such problems. Although maintenance optimization in a dynamic environment can be modeled as such a sequential decision problem, POMDPs are not widely used for tackling maintenance problems of multi-component systems because their state spaces grow exponentially with the increasing number of components. Factored representation of POMDPs, however, allows the complexity of states to be simplified by exploiting the inherent factored structure of maintenance problems. That is why the thesis concentrates on handling two different maintenance problems of partially observable multi-component dynamic systems to investigate the usability and appropriateness of factored POMDP frameworks for such maintenance problems in terms of modeling, solving and analysis.

To the best of our knowledge, there exists no application of maintenance problems through factored POMDPs in the literature which is the main motivation of this study. We focus first on solving an experimental maintenance problem of a multi-component system in a factored partially observable setting by exploiting factored representations and modeling. Relying on the conditional independence in the factored model, we show how the state transition complexity in a POMDP is reduced by factoring in multi-component systems. Then we apply our knowledge to formulate a real-life maintenance problem a regenerative air heater (RAH) system using factored POMDPs. We obtain policies solving the multi-component experimental maintenance problem and also regenerative air heater problem with a factored POMDP solver and perform in-depth sensitivity analyses. Moreover, we develop some myopic predefined policies under reactive and proactive maintenance strategies and compare their performances with the ones obtained by factored POMDPs. The results show promising results both for the experimental model and the real-life system.

The two maintenance POMDP models tackled have differences in the following aspects. The first problem has four independent components which are similar in terms of costs and aging. All have two degrading states in addition to the default states which are full working and failure. The second model is from a real-life maintenance problem and have six components. There exist stochastic dependencies in some of them. The components differ significantly in maintenance costs, durations and aging. Futhermore, there exists exogenous variables reflecting the environmental uncertainty. Although the sensitivity analyses performed on the experimental POMDP model give consistent results in all scenarios, some of the cases in the real-life maintenance problem do not give qualitative policies. However, it should be noted that factored POMDPs are convenient in terms of modeling due to the inherent factored structure of multi-component systems. Because of this, they enable respectively easy formulation and also easy structural changes in the model which would be troublesome if flat POMDPs was to be used.

The rest of this thesis is organized as follows. Chapter 2 briefly overviews the related work on applying the POMDP framework to the maintenance problem. Chapter 3 introduces the theoretical background of POMDPs. The performance comparison of selected POMDP solvers is presented in Chapter 4. Chapter 5 covers the modeling backgrounds and the results of an experimental maintenance problem using POMDPs. Chapter 6 focuses on the problem of maintenance model implementation for a real-life thermal power plant. Finally, Chapter 7 gives conclusions and future research directions.

2. LITERATURE REVIEW

This chapter presents the literature relevant to the concept of this thesis. In this thesis, a partially observable Markov decision process, which is one of the state-of-art sequential decision-making frameworks is used to model and solve the maintenance problems. That is why a brief history of the development of POMDPs and related works of the maintenance applications using POMDPs are provided. Furthermore, this thesis focuses on addressing the maintenance problem in a factored partially observable setting. Thus, applications using factored POMDPs are also presented.

2.1. Historical Development of POMDPs

At the heart of maintenance, planning and decision-making are required. Effective planning and decision-making ensure the improvement of the whole process and the decrease of costs. Decision theory defines how the decision-maker should make decisions based on its priorities. Decision-theoretical planning [31], which is the common working area of operations research and artificial intelligence, focuses on decision making under uncertainty. Operations Research provides a mathematical basis for quantitatively assessing decisions that enable the best decisions to be determined in terms of perceived benefits or effectiveness in achieving the given objectives. On the other hand, in artificial intelligence, agents perceive their environment and make decisions using various search techniques to maximize their chance of successfully achieving their goals. MDPs are an essential framework for modeling sequential decision-theoretic planning. Bellman [25] develops dynamic programming (DP), a recursive method to solve sequential decision problems under uncertainty. Bellman also develops the value iteration algorithm as a DP method. Howard [32] introduces the idea of policy iteration with the average reward for solving infinite horizon problems. A POMDP is the generalization of an MDP allowing imperfect information about the system states. An agent selects an action from the action set based solely on noisy information in each decision period. The agent remembers its past observations and decisions. This is called “all history” and it is difficult to process and keep it since it expands over time. Astrom [33] points out that belief states are enough to summarize the whole history without loss of optimality. Thus, POMDPs become belief-state MDPs, which are a particular case of the continuous MDP through a continuous space containing the probability values instead of discrete state space [34].

Sondik, in his dissertation [35] and in subsequent paper [36], is the first to address and resolve the computational difficulties associated with POMDPs. In these studies, it is proven that for finite horizon POMDPs, the continuous belief state MDP obtained has a piece-wise linear convex value function at each period. He develops a one-pass algorithm to compute the optimal policy and value function for POMDPs on the finite horizon. Besides, Sondik [37] develops a policy-iteration algorithm exploiting exact value function updates to compute ε-optimal policies. This algorithm is later improved by Hansen [38].

Monahan [39] provides comprehensive coverage of the theories, models, and algorithms dealing with POMDPs. In this study, a wide range of models are discussed in areas such as learning and optimal stopping, quality control, internal audit and machine maintenance within the POMDP framework. Moreover, in this study, he proposes an enumeration algorithm to compute the optimal policy and value function for POMDPs on the finite horizon. The main idea of this algorithm is to generate all possible next-horizon vectors and to eliminate the useless ones. Papadimitriou ve Tsitsiklis [40] discuss the computational complexity of POMDPs. The main problem of the optimal policy existence for finite POMDPs is proven to be PSPACE-complete.

Cheng [41] proposes two new exact algorithms called relaxed region and linear support algorithms. These algorithms build on Sondik’s idea; however, they have fewer restrictions. Littman [42] presents the witness algorithm for solving discounted finite-horizon POMDPs exactly using value iteration. Cassandra et al. [43] implement an incremental pruning algorithm with the main idea of combining the Monahan’s enumeration algorithm and the witness algorithm.

Recently, several computational and design challenges for the solution of POMDPs lead to the development of various solution algorithms and procedures. For instance, a series of approximations can achieve a near-optimal result. There are many approximate POMDP solution methods in the literature. These include MDP-based heuristics [44, 45], grid-based methods [46–49], point-based methods [50–52], history-based methods [53] and policy search methods [54]. This study focuses, in particular, on the point-based approaches.

Point-based approaches gain increasing popularity in solving large-scale POMDPs. Due to the remarkable progress through sampling the belief space and approximate computing solutions, state-of-art point-based solvers can solve hundreds of states [55]. Pineau [50, 56] develops a point-based value iteration algorithm (PBVI) which computes the value function only for a limited belief space, accessible from the initial belief state and iteratively added new points to the set as needed. Spaan [51] proposes the Perseus algorithm which is similar to the PBVI algorithm, but instead of updating all belief points in every iteration, it updates only unimproved belief points. Since Perseus updates a small subset of beliefs each time, it can approach a policy quicker. Smith and Simmons [52] presents heuristic search value iteration (HSVI), which uses heuristics based on the value function's higher and lower limits to collect the belief points. Kurniawati [55] implements successive approximations of the reachable space under optimal policies (SARSOP), exploiting the idea of optimally accessible belief spaces to increase computational efficiency.

POMDPs suffer from the curse of dimensionality [34] and curse of history [56]. The former means that the size of the state space grows exponentially with the number of states, whereas the latter one refers to the exponential growth of the number of action-observation histories with the planning horizon. Thus, POMDPs are limited to address problems having large state spaces. To overcome the curse of dimensionality, factored representations have been proposed for systems having inherent factored structures. In a classical POMDP representation, the system is represented by a single node that has multiple states. However, factored POMDPs are efficient tools for reducing the computational complexity since variables are represented compactly via data structure representations such as decision trees (DTs) [57] and algebraic decision diagrams (ADDs) [28]. Poupart [58] presents symbolic Perseus, which is an updated version of Perseus to handle the factored POMDP models. Smith and Simmons propose [59] symbolic HSVI, which is a factored version of the heuristic search value iteration (HSVI) algorithm for solving the factored POMDP models.

This thesis concentrates on showing how to formulate the maintenance problem of systems consisting of partially observable deteriorating components with factored POMDPs and to build successful maintenance strategies for such systems. That is why ready-made POMDP solvers that are able to solve the maintenance models at hand are investigated. Table 2.1 provides a summary of the most frequently used solvers available in the literature. All solvers can be compiled in GNU/Linux and Apple OS-X environments.

2.2. Applications of POMDPs

POMDPs are applied in a wide variety of different real-life problems. In this section, an overview of selected applications is given. Robotics are one of the most widely used application fields which includes navigation and localization of robots [65–69], visual tracking [70], adaptive sensing [30, 71] and robot-assisted health care [72, 73]. Another application domain of POMDPs is health informatics which covers treatment prescription of heart disease [74], kinds of cancer [75], Parkinson's disease [76] and assistance of disabled people [77]. There has also been work on the use of POMDPs in inventory control [78], dynamic pricing strategies [79] and marketing campaigns [80].

POMDPs are less prevalent in the domain of maintenance than other probabilistic graphical approaches because of their high operational complexity. However, POMDPs offer state-based maintenance policies that minimize operating costs and maximize the machine's production capacity. Early theoretical works studied by [81] and [82] are pioneering in POMDP modeling. Monahan's survey [39] presents an overview of the

Table 2.1: POMDP solvers.

Solver	Developer	Language	Last Update	Algorithms
POMDP-solve [60]	Cassandra T.	C	2015	Enumaration Two Pass Linear Support Algortihm Witness Algorithm Incremental Pruining
MADP [61]	Oliehoek F. et al.	C++	2017	Enumaration Incremental Pruining Perseus
ZMDP [62]	Smith T.	C++	2009	FRTDP HSVI2 RTDP LRTDP
APPL [63]	National University of Singapore	C++	2012	SARSOP DESPOT MCVI
JULIA POMDP [64]	Stanford Intelligent Systems Laboratory	Julia	2018	QMDP FIB SARSOP AEMS POMCP DESPOT MCVI
Perseus [51]	Spaan M.	Matlab	2005	Perseus
Symbolic Perseus* [58]	Poupart P.	Matlab Java	2009	Symbolic Perseus

* *uses factored representations.*

models and algorithms dealing with POMDPs. Machine replacement is also discussed within the scope of this study. After these contributions, in literature, the number of articles that provide theoretical frameworks for the management of the maintenance has increased [83–85].

Papakonstantinou and Masanobu present a comprehensive study of two-part for planning structural inspection and maintenance. In the first part, they perform an extensive literature survey in the field of scheduling and maintenance planning through dynamic programming and POMDPs [18]. In the second part, they apply theory from the first part and obtain maintenance and inspection policies for a corroding reinforced concrete structure [86].

Madanat et al. [87] adopt the latent Markov decision process (LMDP) for the maintenance scheduling problem of highway-pavement networks. LMDP is an updated version of the MDP by considering different condition states obtained from visual inspections considering measurement errors. In [88], a bridge inspection model is developed for structural maintenance in infrastructure systems to address the imperfect assessment of the damage conditions due to inspection errors. The same authors develop a model considering fatigue and corrosion as the degradation processes of a highway bridge [89]. They also show how to formulate a POMDP model for risk-based bridge inspection maintenance as a later work [90].

Ivy and Nembhard [91] combine statistical quality control (SQC) and POMDP to determine optimal maintenance policies of deteriorating systems in the case of incomplete information. SQC methods are employed to describe observation distributions for the POMDP by simulating the real world. Memarzadeh and Pozzi [92] propose adaptive maintenance planning based on the POMDP and inspection scheduling based on Value of Information quantification heuristics to gather information for civil infrastructure problems. Aldurgam and Duffuaa [93] use POMDP models to create policy graphs allowing the decision maker to asses the best maintenance action and speed setting for a multi-state, multi-stage machine maintenance problem.

2.3. Applications of Factored POMDPs

There has been work on the use of factored partially observable settings in literature. Factored POMDPs have been widely used to design the dialogue management module of spoken dialogue systems. In dialogue systems, the control process is complicated because automatic speech recognition is not a very secure process and therefore, the state of the speech is unknown. Therefore, several studies are carried out using decision-making models on dialogue management through factored POMDPs. Bui et al. [94] show how to model effective dialogue systems for a single-slot navigation dialogue problem and to develop an emotional dialogue model using factored POMDPs. William et al. [95] present a factored representation for defining POMDPs to handle the spoken dialogue management for buying train tickets. [96] explains how POMDPs can provide a mathematical structure for modeling available uncertainty in spoken dialog systems for booking flights. The assistive technology and human-robot interaction problem have recently become a subject for increasing research interest. Assistive technologies address the problem of elderly people by guiding an older adult using audio or video. Hoey et al. [97] provide guidance of patients suffering from Alzheimer's using factored POMDPs. Jean et al. [98] propose a POMDP model for action scheduling and identification of human error during activities of daily living. Taha et al. [99] adopt POMDPs to model a disabled person's interaction with a robotic wheelchair. Table 2.2 contains a summary of POMDP applications, grouped by the algorithm used, domain area and subject.

Table 2.2: POMDP applications by domain and method used.

Paper	Method	Algorithm	Domain	Subject
Ellis and Corotis (1995) [88]	POMDP	One-Pass	Maintenance	Structural Planning
Reyes et al. (2009) [100]	Factored MDP	SPUDD	Maintenance	Power Supply Systems
Papakonstantinou and Shinozuka (2014) [86, 101]	POMDP	Perseus	Maintenance	Structural Planning
Sheng and Feng (2014) [102]	POMDP	Perseus	Maintenance	Power Supply Systems
Pozzi et al. (2014) [103]	POMDP	SARSOP	Maintenance	Wind Farms
Memarzadeh and Pozzi (2016) [104]	POMDP	SARSOP	Maintenance	Infrastructure Systems
Lin et al. (2016) [102]	POMDP	One-Pass	Maintenance	Power Supply Systems
Papakonstantinou Memarzadeh (2017) [105]	POMDP	SARSOP Perseus HSVI	Maintenance	Structural Planning
Ghandali et al. (2018) [106]	POMDP	Perseus Incremental Pruining	Maintenance	Sustainability
Morato et al. (2018) [107]	POMDP	SARSOP	Maintenance	Wind Farms
Papakonstantinou et al. (2018) [108–110]	POMDP	SARSOP Perseus ZMDP	Maintenance	Structural Planning
Kim et al. (2008) [111]	Factored POMDP	Symbolic HSVI	AI	Dialogue Systems
Williams et al. (2005) [95]	Factored POMDP	Symbolic Perseus	AI	Dialogue Systems
Müller et al. (2012) [112]	Factored POMDP	Symbolic Perseus	AI	Assistive Technologies
Hoey et al. (2007) [113]	Factored POMDP	Symbolic Perseus	AI	Assistive Technologies
Araya-Lopez et al. (2010) [114]	POMDP	Pomdp-Solve	AI	Robotic
Wang et al. (2013) [115]	Factored POMDP	Symbolic Perseus	Ecology	Bird Migration
Rout et al. (2014) [116]	POMDP	Pomdp-Solve	Ecology	Invasive Species
Erdoğdu et al. (2011) [117]	Factored POMDP	Symbolic Perseus	Genetic	Gene Expression
Capitan et al. (2011) [118]	Factored POMDP	Symbolic Perseus SARSOP	Robotic	Navigation
Küçükyazıcı et al. (2015) [119]	POMDP	Pomdp-solve	Health	Personalized Treatment

3. METHODOLOGY

The main focus of this thesis is to model maintenance problems in realistic settings through POMDPs. Besides, the complexity of the states is simplified by exploiting the factored structure that is inherent to maintenance problems. Maintenance management is one of the sequential decision-making processes under uncertainty. Such problems can be solved by mathematical formulation provided by MDPs and POMDPs. That is why, in this chapter, an overview of the MDPs and POMDPs and their solution methods are provided. The basics of factored representations in POMDPs are also covered.

3.1. Sequential Decision-Making Processes

Sequential decision-making processes are a wide field of study which includes applications in operations research, artificial intelligence, maintenance and robotics [18]. A sequential decision process involves an agent to maximize the reward by performing the appropriate actions. The agents interact with the external environment at certain time intervals. After each action taken by the agent, the environment changes in a probabilistic way. At each step, the agent observes the state of the environment as input and immediately determines the action to be selected as the output. The most commonly used models of sequential decision processes are MDPs and POMDPs.

3.2. Markov Decision Processes

MDPs are the extension of Markov chains by a number of decisions, actions and state-based rewards. In an MDP model, the interaction of the agent with the environment is given in Figure 3.1. The agent's major aim is to determine an appropriate action policy to optimize the expected total reward. At any state s, the agent selects an action from the action set. The action taken by the agent returns a reward and provides a stochastic transition from the current state to a new state. A typical MDP model is given in Figure 3.2 where squares, circles and diamonds represent the action, the state and the reward, respectively. The reward is dependent on the state, previous state and the action, while the state is dependent on the previous state and action.

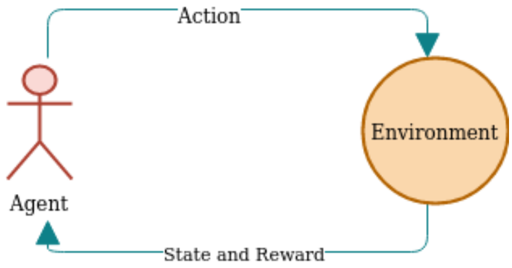


Figure 3.1: Interaction of an agent with the environment in an MDP model.

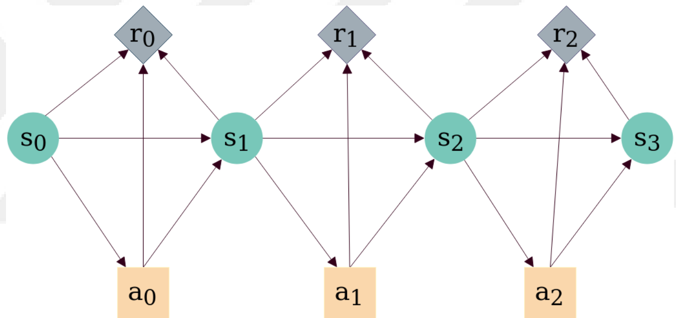


Figure 3.2: A typical MDP.

3.2.1. Formulation

An MDP can be defined by four-tuple: $< S, A, T, R >$

- **States**: System state space S is a set of all possible states. A state is the definition of the environment at any point in the horizon.
- **Actions**: Action state space A is a set of all alternative actions. The agent's main goal is maximizing its reward by performing appropriate actions. At each step an agent cannot take more than one action.
- **Transition Function**: Transition function $T(s'|s,a) = Pr(s'|s,a)$ that determine the agent's next state distribution s', with $Pr(s'|s,a)$ standing for the prob-

ability of moving from state s to state s' when an action a is executed. Transition function satisfy the Markov property, i.e. the probability of moving state s to s' depends solely on the current state-action pair, regardless of past state-action pairs. $Pr(s_{t+1}|s_t, a_t, s_t - 1, a_t - 1, ..., s_0, a_0) = Pr(s_{t+1}|s_t, a_t)$.

- **Reward Function**: $R(s, a, s') : S \times A \times S \rightarrow \Re$ is the function of returning rewards for taking action a when in state s.

The objective of an MDP is to calculate the optimal policy maximizing the expected total reward. To achieve this aim, the agent starts from an initial state s_0 and compares the different alternative action plans in the long run, considering the immediate reward r of each possible action a. Each of these plans is called "policy". The optimal policy that is the policy maximizing the long-term reward is the solution of the MDP. There are two types of policies: stationary and non-stationary. The stationary policy, $\pi : S \rightarrow A$, is a mapping from the states to the actions. In this policy, the selection of actions depends solely on the current states; the decision does not depend on the time step. A non-stationary policy π is a sequence of state-action, sorted by time.

In an MDP model, two different performance criteria can be used to maximize the expected long-term reward, first, finite horizon criterion.

$$E\left[\sum_{t=1}^{T} R(s_t, a_t)\right], \tag{3.1}$$

where r_t is the reward received at the time step t and T is end time of the process. The agent tries to develop an action strategy that maximize long-term reward over the horizon length T. The optimal value function V^* at the state s is calculated as in the Equation 3.2 where $V^\pi(s)$ refers to the value of the π policy starting with state s. The MDP aims to find the optimal value for the initial state s_0.

$$V^*(s) = max_\pi(V^\pi(s)) \tag{3.2}$$

The value function is similar to the policy. The policy π maps an action to each state, while the value function assigns a numeric value for each state. Second, the infinite horizon criterion,

$$E\left[\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t)\right] \tag{3.3}$$

where $T = \infty$. $\gamma \in [0,1]$ is a constant, namely discount factor, which ensures that the sum of the rewards is finite. The objective of a discount factor is to reduce the importance of future rewards by reducing their contributions to the selection of actions to be taken at earlier stages. If an MDP model does not have a specified time horizon length for successive states, or if the time horizon length is large enough to be considered infinite, the infinite horizon performance criterion can be used for the solution. The expected number of time horizons of the agents is always $1/(1-\gamma)$. That is, the expected distance to the horizon never changes and action strategies are not a function of time. Therefore while finite horizon models induce a non-stationary policy (time-independent), infinite horizon models induce a stationary policy (time-dependent) [120].

The optimal value function of the agent moving from state s and by following the policy π is calculated as in the Equation 3.4. The value function $V^{\pi}(s)$ recursively accumulates rewards during time horizon [121].

$$V^{\pi}(s) = R(s,a) + \gamma\sum_{s'\in S}T(s'|s,a)(V^{\pi}(s')) \tag{3.4}$$

where $R(s,a)$ is the immediate reward, $T(s'|s,a)$ probability that taking action a at state s leads to state s' and $V^{\pi}(s')$ is the value of the π policy starting with the new state s'. Let $V^{*}(s)$ be the optimal value function maximizing the expected total revenue of the process in the long-run for state s. Dynamic programming allows to compute the long-term value $V^{*}(s)$ for each (discrete) state s and is summarized by the Bellman

Equation which is given in Equation 3.5 where $0 \leq \gamma \leq 1$ is the discount factor.

$$V^*(s) = \max_a \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)(V^\pi(s') \right\} \tag{3.5}$$

Optimal policy could be expressed by the same recursive equation which is given in Equation 3.6:

$$\pi^*(s) = \text{argmax}_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)(V^\pi(s') \right\} \tag{3.6}$$

3.2.2. Solving MDPs

Dynamic programming (DP) methods is widely used to find optimal value functions in MDPs. Value iteration [122] and policy iteration [32] are two popular DP methods.

3.2.2.1. Value Iteration. MDPs aim to produce a state-to-action transition that represents the best actions to be selected in each case. In an MDP, the value of a state is the expected cumulative reward returned after that state. It is possible to formulate this statement recursively. At a given state, there are a finite number of possible next states. Thus, to calculate the value of the current state, the value of all the next states is needed. If the horizon length in the model is only one, that is, if the agent only needs to decide once for a single state, there will be no future impact on the decision because there is no next step. Thus, without considering the effects of the future, it is decided only based on immediate rewards of all possible actions. However, if the horizon length is more than one in the model, the values of the actions to be selected at the next time steps should be added to the immediate reward of the action by discounting when evaluating all possible actions.

The value iteration algorithm starts with an initial random guess of the value function. Then it repeatedly calculates the next value and updates the overall reward according to Equation 3.7.

$$V^{t+1}(s) = \max_a \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)(V^t(s') \right\} \tag{3.7}$$

<u>3.2.2.2. Policy Iteration.</u> Policy iteration is an algorithm based on improving policy for a decision-making problem. The main idea behind policy iteration is, given policy, it is possible to calculate a value function based on this policy. Unlike value iteration, policy iteration takes any random policy π as the input; there is no initial policy. The value of policy π first is calculated. Moreover, there is a policy improvement phase after the value calculation phase. At this phase, a DP update is performed, and a new policy is obtained. The policy obtained is compared with the current policy. If the policy is improved, a new policy is obtained by DP updating. When an iteration does not improve the policy anymore, the algorithm terminates, converging to the optimal policy.

3.3. Partially Observable Markov Decision Processes

MDPs are the frameworks for modeling sequential decision making under uncertainty where the transition dynamics are known with certainty over the planning horizon. However, if the agent has partial knowledge of the system state, MDPs are inappropriate. In such systems, the state of the system can be estimated via observations that are collected with some signals or measurements. However, the accuracy of these observations is also probabilistic. For this purpose, MDPs are extended to POMDP, by providing information regarding the system's state through noisy sensors [18]. A POMDP is a rich framework for modeling sequential decision-making problems under imperfect observations. In a POMDP model, the interaction of the agent with the environment is given in Figure 3.3. When the agent takes action, it obtains an observation. This observation affects belief state probabilities at the next step. In other words, the agent is obtained a hint for the next state.

A POMDP can be defined by six-tuple: $< S, A, T, R, \theta, O >$

- **States**: System state space S is a finite set of all possible states. A state is the definition of the environment at any point on the horizon. In this thesis, discrete models with a finite number of states will be studied.
- **Actions**: Action state space A is a set of all alternative actions. The agent's main goal is maximizing its reward by performing appropriate actions. At each time step an agent can execute at most one action.
- **Observations:** Observation state space θ is all possible observations.
- **Transition Function**: Transition function $T(s'|s,a) = Pr(s'|s,a)$ that determine the agent's next state s', where $Pr(s'|s,a)$ denoting the probability of moving from state s to s' when an action a is executed. Transition function satisfies the Markov property, i.e. the probability of moving from state s to state s' depends solely on the current state-action pair, regardless of past state-action pairs. $Pr(s_{t+1}|s_t, a_t, s_t - 1, a_t - 1, ..., s_0, a_0) = Pr(s_{t+1}|s_t, a_t)$.
- **Observation Function**: $O : S \times A \times \theta \rightarrow \Delta(O)$ is a function giving the observation probabilities based on the state of the process and actions.
- **Reward Function**: $R(s,a,o,s') : S \times A \times O \times S \rightarrow \Delta(R)$ is the function of returning rewards for executing action a when in state s. In this thesis, reward is based on the current state, action and next state.

A typical POMDP representation is illustrated in Figure 3.4. Circles represent the system and observation nodes as chance nodes. Squares represent the decision nodes and diamonds represent reward (utility) nodes. Arrows demonstrate causal influence.

3.3.1. History and Belief States

In POMDPs, unlike MDPs, the agent cannot to observe the hidden state of the environment; however, the agent can predict its state by receiving observations at each

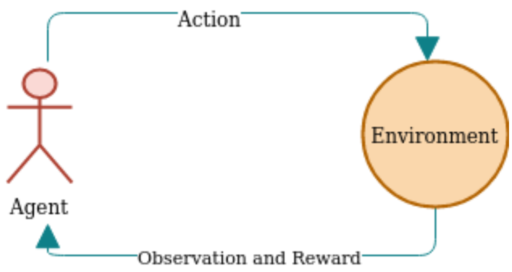


Figure 3.3: Interaction of an agent with the environment in a POMDP model.

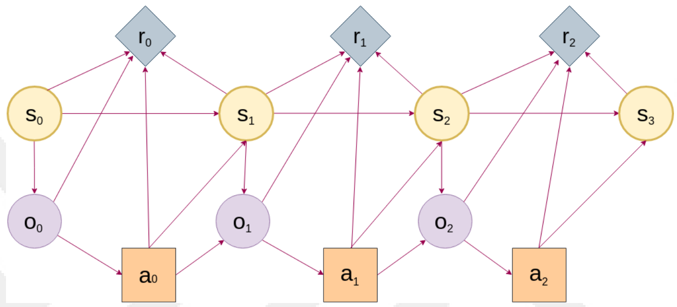


Figure 3.4: A typical POMDP.

step. In the case of incomplete and noisy information, the agent has to remember the full history of the actions and observations in the previous time steps to make the best decision in the current time step. This is called all history" and it is difficult to process and keep it since it expands over time. However, for problems with a large state space and a long horizon, this approach is not scalable. Instead of keeping all history, an alternative representation has been developed that assigns probabilities to every possible state, which are called belief states $b(s)$". Belief state is sufficient to summarize all the history without compromising its optimality with a properly distributed probability distribution over the state space [33]. At each time step, the agent performs an action, it is rewarded accordingly and its belief state needs to be updated using Bayes' rule as follows:

$$
\begin{aligned}
b'(s') &= Pr(s'|o,a,b) \\
&= \frac{Pr(o|s',a,b)Pr(s'|a,b)}{Pr(o|a,b)} \\
&= \frac{Pr(o|s',a)\sum_{s\in S} Pr(s'|a,b,s)Pr(s|a,b)}{Pr(o|a,b)} \\
&= \frac{O(s',a,o)\sum_{s\in S} T(s',a,s)b(s)}{Pr(o|a,b)}
\end{aligned}
\tag{3.8}
$$

where T is the transition probability function from the current belief b to next belief b' after performing an action a and observing o. The term $Pr(o|a,b)$ is used for normalization purpose. A set of all beliefs forms the belief space B.

3.3.2. Belief MDP

The policy of a POMDP is a mapping from belief states to actions. These belief states can be considered as states in MDPs. Thus, POMDPs become belief-state MDPs, which are a special case of continuous-state MDP through a continuous space containing the probability values instead of discrete state space [34]. A belief MDP is defined by a four-tuple: $< S, A, T, R_{belief} >$

- **States**: System state space S is the belief space B of the POMDP
- **Actions**: Finite set of actions
- **Transition Function**: T is transition probability function from current belief b to next belief b' after performing an action a and observing o:

$$
T(b'|o,a,b) = Pr(b'|b,a) = \sum_{o\in O} Pr(b'|b,a,o)Pr(o|a,b) \tag{3.9}
$$

where

$$
P(b'|a,b,o) = \begin{cases} 1, & \text{if } b' = b_o^a. \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}
$$

- **Reward Function**: $R_{belief}(b, a)$ is the function of returning rewards for performing action a when in belief state b:

$$R_{belief}(b, a) = \sum_{s \in S} b(s)R(s, a) \tag{3.11}$$

3.3.3. Value Functions

The value function in POMDPs is calculated on the belief space as in Equation 3.13. Belief space is continuous; however, for a finite horizon, the optimal value function is piecewise-linear and convex. Thus, any finite-horizon solution is represented by a limited set of alpha-vectors due to this property [37]. Alpha-vectors are a set of hyperplanes defining belief functions. For every belief point, the value function is equal to hyperplane gives the maximum value.

$$V^*(b) = \max_{a \in A} \left\{ \sum_{s \in S} R(s, a)b(s) + \gamma \sum_{o \in \theta} \sum_{s \in S} P(o|s, a)b(s)V^*(b') \right\} \tag{3.12}$$

The value function in POMDPs is calculated on the belief space as in Equation 3.13. Belief space is continuous; however, for a finite horizon, the optimal value function is piecewise-linear and convex. Thus, any finite-horizon solution is represented by a limited set of alpha-vectors due to this property [37]. Alpha-vectors are a set of hyperplanes defining belief functions. For every belief point, the value function is equal to hyperplane gives the maximum value.

$$V^*(b) = \max_{a \in A} \left\{ \sum_{s \in S} R(s, a)b(s) + \gamma \sum_{o \in \theta} \sum_{s \in S} P(o|s, a)b(s)V^*(b') \right\} \tag{3.13}$$

3.4. POMDP Solution Algorithms

The solutions of POMDPs are the process of calculating an action policy for a POMDP model. There are two main approaches to generate a POMDP policy: first, value iteration [36] based on exploring in the space of value functions and second, policy iteration [123] based on exploring in the space of policies. In both approaches, the DP update can be calculated exactly or approximately. The use of exact algorithms in large-scale realistic problems is a challenge, as it requires significant calculation and time. It is known that finding optimal policies for finite horizon POMDPs is a PSPACE-complete problem [40] and POMDPs in the infinite horizon are undecidable [124]. However, a series of approximations can achieve a near-optimal result. The main exact and approximate algorithms used in the solution of POMDPs are classified in Figure 3.5.

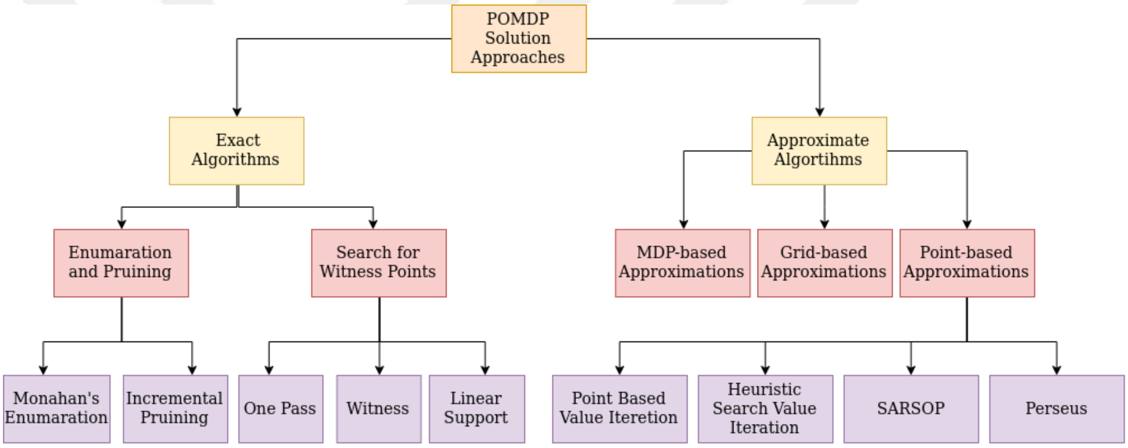


Figure 3.5: POMDP solution approaches.

3.4.1. Exact Solution Algorithms

The exact algorithms contain all the alpha vectors describing the value function for the whole belief space. Each belief state in B has an alpha-vector associated with it and each hyperplane has an associated action from A. However, as the number of observations increases with each iteration, it leads to the exponential growth in the number of alpha-vectors required to represent the value function [125]. Each α-vector is updated throughout the whole belief space. The value function is represented as a vector of values over the belief space, known as α-vectors. The use of exact algorithms in large-scale realistic problems is a challenge, as it requires significant calculation and time.

3.4.1.1. Enumeration and Pruning Algorithms.

- The main idea of Monahan's enumeration algorithm [39] is enumerating all possible useful belief states and applying value iteration algorithm only considering these belief states. Thus, the algorithm generates all possible vectors and then gets rid of the useless ones. As too many vectors are generated in each iteration, pruning is required.
- Incremental Pruning [43, 126] combines the Monahan's enumeration algorithm and witness algorithms. The similar feature with witness algorithms is constructing sets of vectors for each action individually and then focusing on each one observation at a time.

3.4.1.2. Search for Witness Points.

- One Pass [35] finds a value function and generates a vector component for a single belief state, generate a vector for this point, then explore the belief space, on which this component of the value function is dominating by examining possible actions and outcomes. The region dominated by the vector is not fully known. Thus, the same vector can be created for many belief points.
- Linear Support Algorithm [41] is based on one pass algorithm, but there are fewer restrictions. The linear support algorithm relies on geometric properties of the piece-wise linear value function by ignoring to focus on actions and future action plans.
- In witness algorithm [127], the problem is handled from another perspective. Using the same basic structure of one-pass and linear support algorithm, it defines the regions for a vector and searches a point where this vector is not dominant. However, it uses observations in order to simplify the calculation of the real value of belief points.

3.4.2. Approximate Solution Algorithms

Various computational and design challenges for the solution of POMDPs have led to the emergence of different planning algorithms and techniques in the past years. However, a series of approximations can achieve a near-optimal result. The approximation methods can be divided into two often classes: value-function approximations and policy approximations. In policy approximations, instead of searching the space of all possible policies, a subspace of it, which is believed to contain the optimal solution or a good approximation, is searched. On the other hand, approximating value function for only a subspace of belief space can be much easier than computing the full value function. Value-function approximations are exploiting the advantage of heuristics and branch and bound algorithms. Although value function approximations have no guaranteed precision, in many cases they give upper and lower bounds. Approximate algorithms do not always guarantee that they will approach the optimal result quickly, but often they approach the optimal solution with reasonable time and cost and [128]. There are many value function approximations in the literature. These include MDP-based Heuristics, grid-based methods, point-based methods, history-based methods, and policy search methods.

3.4.2.1. MDP-Based Approximations. As the solution of MDPs is much simpler than a POMDP, a number of methods have been proposed that use heuristics based on the underlying MDP. The "Most Likely State" (MLS) [44] heuristic selects the best action by finding the state of the system with the highest probability.

$$\pi_{MLS}(b) = \pi^*(argmax_s b(s)) \tag{3.14}$$

Another heuristic method QMDP tries to approximate the value function with Q-functions [129]. It selects one single α-vector for each action, for the whole belief space at one time. The policy is created considering a weighted sum over the belief state probability distribution. Thus, QMDP considers only the uncertainty in the current step and assumes full observability in all future steps [18].

$$\pi_{Q_{MDP}}(b) = argmax_a(\sum_s b(s)Q^*(s,a)) \tag{3.15}$$

MLS and QMDP methods allow the state to be observable by ignoring partial observability and their policies do not choose actions gaining more information about the condition of the system. In order to improve these approaches, the Fast Informed Bound (FIB) method [74], integrating the observation probabilities into the update step is proposed. FIB chooses the best action per state with respect to the expected observation.

$$\alpha^a_{t+1} = R(s,a) + \gamma \sum_{o \in Z} max_{\alpha_{t+1} \in \Gamma_t} \sum_{s' \in S} O(o|s',a)T(s'|s,a)\alpha_t(s') \tag{3.16}$$

3.4.2.2. Grid-Based Approximations. Grid-based approximations are based on approximating the value function over a continuous belief space with a finite set of the guide point set. These points are often distributed according to a grid pattern, so the name is grid-based approximation.

3.4.2.3. Point-Based Approximations. Recently, point-based approaches have gained increasing popularity in solving large-scale POMDPS. Due to remarkable progress by sampling the belief space and approximate computing solutions, state-of-art point-based solvers can solve hundreds of states [55]. The key idea of point-based approaches is sampling a set of points from belief space B and using this set for approximating B. Point-based approaches optimize the update process, taking into account only a selected subset of belief points. Therefore, it eliminates the problem of the complexity of large state spaces due to computational issues. There are different point-based algorithms with different characters in literature. Point-based solvers differ in belief space subset selection and the order of value function updates, i.e., which points are updated, and how the backups are sorted [130].

The point-based value iteration (PBVI) algorithm [50] computes the value function only for a subset of belief space, reachable from the initial belief state and iteratively adds more points to the set as needed. The selection of these points is based on the principle that at each iteration, PBVI extends the belief subset by greedily selecting new accessible belief points as far from the existing belief points as possible.

Randomized point-based value iteration for POMDPs (Perseus) [51] is a randomized version of PBVI. The main argument of Perseus is randomly choosing the next point for updating from the set of belief points, which were not yet improved. Perseus creates a fixed set of achievable belief points B in the beginning by sampling the trajectories of randomly chosen actions at each step, starting from the first belief b_0. At each iteration, it backs up at least as belief points as necessary over only on a randomized subset of B. This process is repeated until it guarantees that the value function approach is improved for all points in the initial belief set. Then, iterations continue until the stopping criterion is met.

Heuristic search value iteration (HSVI) [52] uses heuristics based on each the higher and lower limits of the value function to collects the belief points and guide the search in the belief space. As in all point-based solvers, HSVI iteratively forms the value function starting from both the lower and upper limits. Using the heuristics accelerates the step of defining critical belief points [18]. The belief space is approximate and only deals with a subset of the belief points as in point-based value iteration. HSVI is a strong algorithm for solving large POMDP problems with large state spaces. Unlike PBVI and Perseus, it requires pruning process.

3.5. Factoring POMDPs

POMDPs are limited to solve problems with large state spaces. The curse of history is solved by limiting the number of beliefs for value function in point-based value iteration [50]. However, "curse of dimensionality" remained a serious challenge for this algorithm [131]. To address the "curse of dimensionality", factored representations can be used. In a flat POMDP representation used by dynamic programming algorithms, all possible states and state transitions are enumerated. However, factored models can represent POMDP components more compactly by exploiting conditional independence between variables and reduce the computational complexity of various algebraic operations performed on vectors in backup iterations [28]. Thus, problems with large state spaces can be represented and solved more efficiently. Dimensionality reduction can be provided by using "Decision Trees" (DTs) [57] or "Algebraic Decision Diagrams" (ADDs) [28].

The main advantage of using the ADD notation is the high efficiency in basic function addition and multiplication. Therefore, these structures provide the computation time savings. An ADD is an extension of binary decision diagrams (BDDs) [132]. Binary decision diagrams that enable the unification of the branches provide a compact representation of the decision trees. An example of a decision tree converted to ADD is shown in Figure 3.6. In a classical POMDP representation, the system is represented

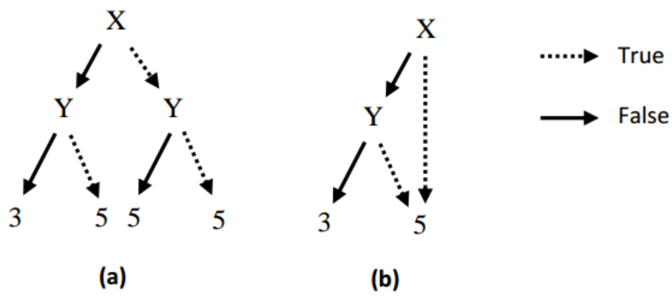


Figure 3.6: An example of decision tree and the corresponding algebraic decision diagram (b) [133].

by a single node that has multiple states. In Figure 3.7, a classical POMDP representation and a factored POMDP for a maintenance model having of four components are illustrated. In (a), the system is represented by a single node that has multiple states, although the POMDP model has been factored into four components in (b). Symbolic Perseus [58] which includes an adapted variant of the point-based value iteration algorithm to solve the factored POMDP models, is used in this thesis.

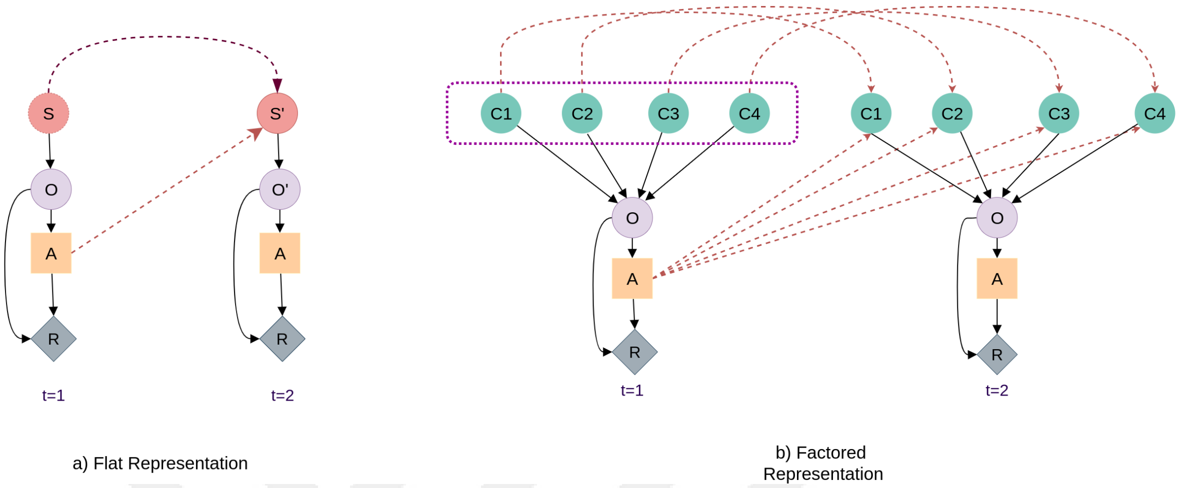


Figure 3.7: POMDP representations.

3.6. Symbolic Perseus

Symbolic Perseus implemented in Matlab (SPM) [58] is used as a policy generator and simulator by exploiting the factored structure already available inherent in the problem. SPM is an approximate point-based POMDP solver, implementing factored representations of value functions and beliefs using algebraic decision diagrams. The innovation of SP is limiting the number of vectors representing the value function without loss of policy quality. Thus, the computational cost of the backup process is reduced. In addition, a belief state approximation is used to solve the dimensionality problem by merging states with values that differ by less than the Bellman error. This is achieved with a compact factored representation of the belief state as a product of the independent marginal of components. SP performs basic ADD operations more efficiently, so belief state backups and updates run faster. In other words, the ADDs used for belief states and vectors provide significant savings in the required memory and calculations.

Real-life problems require POMDP algorithms that are robust against two major challenges: dimensionality and history. Symbolic Perseus successfully overcomes large state-space complexity with a limited backup of accessible beliefs. It also solves the problem of dimensionality by compactly representing alpha vectors and belief states using ADDs. The history problem is solved by limiting the number of vectors for the value function. A sample input of Symbolic Perseus for a 2-component experimental system is given in the Appendix.

3.7. Policy Simulation in SPM

After a policy is obtained, this policy is performed by a simulator to evaluate its performance and to make sensitivity analyses under different scenarios. The flow of the maintenance policy simulation is given in Figure 3.8.

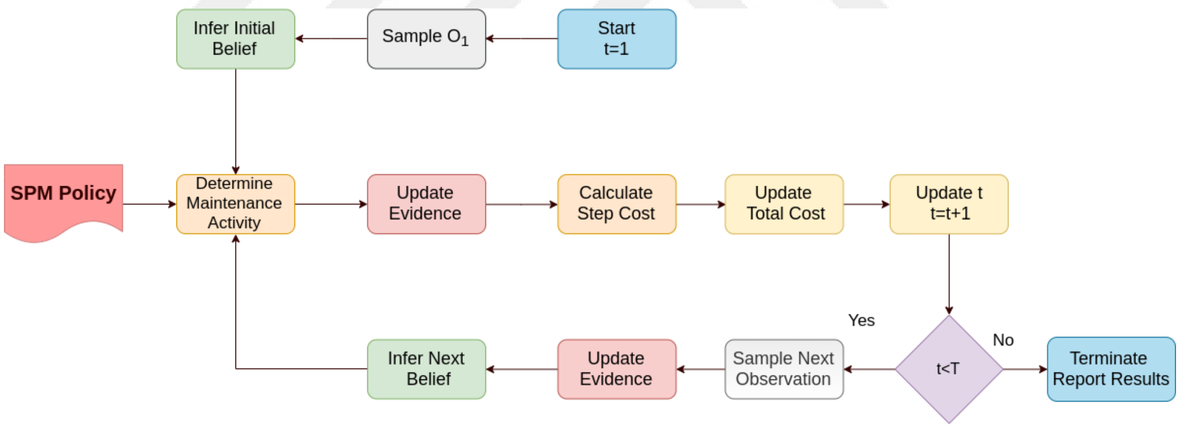


Figure 3.8: SPM maintenance policy simulation flow.

4. PERFORMANCE COMPARISONS OF SELECTED POMDP SOLVERS

In this chapter, comparative analyses of four POMDP solvers having different characteristics in terms of their solution methods for four partially observable stochastically deteriorating maintenance problems setting are included. The primary objective is to reach an exact solution, however, as mentioned, exact methods may not handle problems with large state spaces. Hence, it is resorted to both exact and approximate algorithms to investigate the performance of solution methods through solving problems at different levels of complexity.

4.1. Design of Experiments

To analyze the performance of solvers, four maintenance problem settings consisting of partially observable components deteriorating in time are designed. The main aim is to investigate the limitations of solutions methods by extending the state space of the model. As the number of elements increases in the model, the state space increases and the model becomes more complex. Moreover, the dependencies among the elements of the model increase the complexity further. Model 1, 2, 3 and 4 refer to a system having one, two, three and four components respectively. The relationships between the components and the processes of the models are shown in Figure 4.1 for two periods. There are two states of all components W, F and the observation node G, R and all processes have two states W, F where W and F stand working and fail respectively and G and R stand for green and red respectively. The assumptions about the empirical maintenance models are as follows:

- The components degrade or deteriorate over time. All components degrade or deteriorate with the same probabilities.
- All components can be replaced at any time.

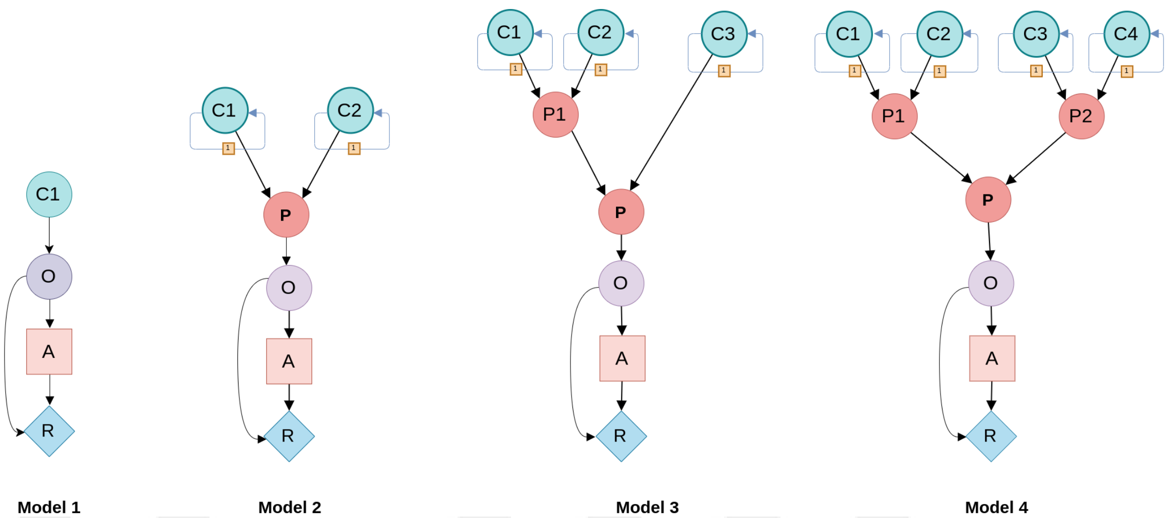


Figure 4.1: Empirical models.

- Direct monitoring of components and processes is not possible. However, they are estimated by the observation node.
- At any time, at most one of the components can be replaced.
- The processes are defined as the result of the interaction between their parents. The main processing node P is directly connected to the observable node used to collect information from the processing node. Model 1 contains only the observation node. Other models have process nodes and observation nodes.
- All components are in their "working" state initially.
- Two different type of maintenance actions are considered: "Do nothing" and "ReplaceCi" where i is the index of components. When a "Do nothing" action is performed, the system experiences the natural deterioration process. Relevant component is replaced to a new one by "ReplaceCi" action.
- Rewards -costs for the maintenance problems- are collected by two means: actions and observations. Maintenance costs depend on the observations received and the maintenance performed. The total maintenance costs are defined as the summation of the downtime cost and cost of the replacement of the relevant component. The downtime cost is the cost of loss in production resulting from system downtime. When a green or yellow signal is observed, the component replacement costs are 100 TL, 200 TL, 300 TL, 400 TL respectively; when a red signal is received, the component replacement costs are 200 TL, 400 TL, 600

TL, 800 TL respectively. When the system receives a green or yellow signal the downtime cost of 2,500 TL incurs, when a red signal is received, more downtime incurs. The total maintenance cost of each component is given in Table 4.5, depending on the observation received and the action performed. Herein, the downtime cost when the red signal is observed is shown parametric.

Transition probabilities of components C1, C2, C3 and C4; the conditional probabilities of nodes P1, P2, P and observation node are given in Table 4.1, Table 4.2, Table 4.4 and Table 4.3 respectively where the transitional and conditional probabilities for each component are represented in each column.

Table 4.1: Transition probabilities of components.

Action Node	**Y**		**N**	
Self [t-1]	**W**	**F**	**W**	**F**
W	1	1	0.95	0
F	0	0	0.05	1

Table 4.2: Conditional probabilities of P1 and P2.

C2	**W**		**F**	
C1	**W**	**F**	**W**	**F**
W	1	0.5	0.5	0
F	0	0.5	0.5	1

Table 4.3: Conditional probabilities of O.

P3	**W**	**NW**
Green	0.95	0
Red	0.05	1

Table 4.4: Conditional probabilities of P.

P2	**W**		**F**	
P1	**W**	**F**	**W**	**F**
W	1	1	1	0
F	0	0	0	1

Table 4.5: Maintenance costs of the experimental model.

Action	**Observation**		
	Green	**Yellow**	**Red**
Do nothing	0	0	Downtime Cost + 200
Replace C1	2,600	2,600	Downtime Cost + 400
Replace C2	2,700	2,700	Downtime Cost + 600
Replace C3	2,800	2,800	Downtime Cost + 800
Replace C4	2,900	2,900	Downtime Cost + 1,000

4.2. POMDP Solvers

Pilot models are formulated as flat and factored POMDPs and solved by four different POMDP solvers which are POMDP-solve (PS) [60], successive approximations of the reachable space under optimal policies (SARSOP) [55], Symbolic Perseus Matlab (SPM) [58] and Symbolic Perseus Java (SPJ) [134]. PS solves flat POMDP problems exactly. It implements a number of POMDP solution algorithms include enumeration [35,39], two pass [35], linear support [41], witness [34,43], incremental pruning [43,135]. It uses the basic dynamic programming approach for all algorithms, solving one stage at a time working backward in time. SARSOP solves also flat POMDP problems approximately. This algorithm uses the point-based value iteration to reach the near-optimal solution. SPM compute the policies factored POMDPs using a point-based value iteration algorithm. It is based on Perseus solver [51], it uses algebraic decision diagrams (ADDs) as a data structure. SPM is implemented in Matlab and Java. SPJ is a re-implemented version of SPM in Java.

4.3. General Policy Simulation

POMDP solvers returns policies and the policy is simulated using DBNs in order to evaluate its performance. We use BNT toolbox [136] to construct the DBN model and to compute the required inferences such as sampling the observations and inferring the belief states. Policy simulations are performed within MATLAB environment. The overview of the maintenance policy simulation is given in Figure 4.2.

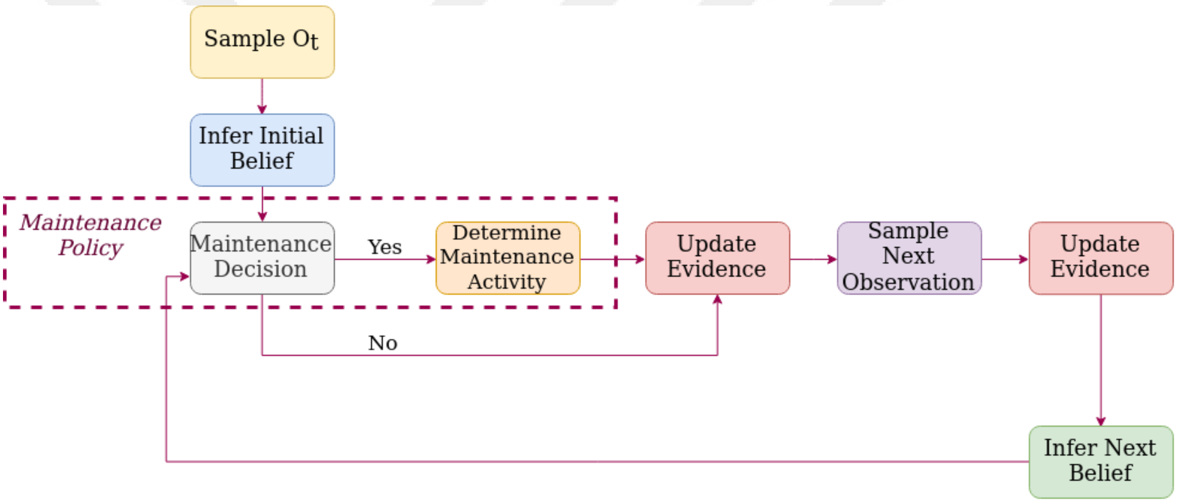


Figure 4.2: Maintenance Policy Simulation.

4.4. Results of Experiments

First PS is run for computing the value of the exact policy. Simultaneously, approximate policies are obtained by SARSOP, SPJ and SPM solvers and they are also simulated for the same planning horizon. The policies generated by PS, SARSOP and SPJ are simulated using DBNs on the planning horizon of 100 days and the simulation is replicated 50 times and also the policies produced by SPM are executed in the SPM's simulator with the same parameters. The discount factor is set as 0.999. Results of sensitivity analyses of models are given in the following tables where TCost, TRed, TRep, DC and ACR denote the total cost, the total number of red signals obtained and the total number of replacements, the downtime cost and the average component replacement respectively in 100 days of a horizon. Average and standard deviation of these measures are reported in the tables. Furthermore, the average total replacements for each component are also given.

4.4.1. Experimental Results of Model 1

Model 1 is the most primitive maintenance problem having one component and one observation node. Table 4.6 shows the results of running the maintenance policies generated by the solvers.

Table 4.6: Simulation results of POMDP solvers for Model 1 when DC=5,000.

	TCost		**TRed**		**TRep**		**ACR**
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std.**	**C1**
SPM	46,074	14,917	9.32	3.01	9.32	3.01	9.32
SPJ	46,912	13,891	9.48	2.81	9.44	2.74	9.44
PS	44,424	14,843	9.00	3.02	8.60	2.70	8.60
SARSOP	47,939	15,402	9.70	3.11	9.04	2.78	9.04

The main experimental focus is comparing policies generated by the POMDP solvers. The performance measure of POMDP policies is the comparison of average total maintenance cost calculated on the planning horizon given in this study. For the statistical comparison, one way ANOVA is used to check whether there is any significant difference between the means of simulation. Assumptions of normality and homogeneity of variance have been checked. Although the assumption of normality is fulfilled homogeneity of variance assumption is violated. For this reason, the Games-Howell post-hoc test (GH) [137] which is used in the case of the lack of homoscedasticity is applied. Results of sensitivity analysis including the groupings given in Table 4.7. The same letter indicates the policies which are insignificant in terms of the total maintenance cost.

Table 4.7: GH test results for model 1 when DC=5,000.

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	44,214	14,726	(40,029, 48,399)	A
SPJ	46,912	13,891	(42,964, 50,860)	A
PS	44,424	14,843	(40,205, 48,642)	A
SARSOP	47,939	15,402	(43,562, 52,316)	A

The results show that the policies tend to be reactive at the downtime cost of 5,000. That is to say, no action is taken to prevent failures in almost all maintenance actions. To initiate proactive policies, the downtime cost has been increased to 10,000. The results are given in Table 4.8.

Table 4.8: Simulation results of POMDP solvers for model 1 when DC=10,000.

	TCost		**TRed**		**TRep**		**ACR**
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**
SPM	91,132	27,061	9.40	2.80	9.40	2.80	9.40
SPJ	86,218	21,252	8.86	2.19	8.86	2.19	8.86
PS	96,896	24,073	10.00	2.50	9.44	2.37	9.44
SARSOP	86,970	28,758	8.98	2.97	8.50	2.84	8.50

Results of the sensitivity analysis are given in Table 4.9. According to the GH post-hoc test, there is no statistically significant difference between the four groups in terms of the average total maintenance costs.

POMDP solvers are expected to produce more proactive behavior at that downtime cost. However, the solvers produce reactive policies. It stems from that the system performance only depends on a single component whose deterioration speed is very low. For this reason, the downtime cost has been increased to a higher value. The results of the simulations at a downtime cost of 55,000 are given in Table 4.10.

Table 4.9: GH test results for model 1 when DC=10,000.

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	91,132	27,061	(83,441, 98,822)	A
SPJ	86,218	21,252	(80,178, 92,258)	A
PS	96,896	24,073	(90,055, 103,738)	A
SARSOP	86,970	28,758	(78,797, 95,144)	A

Table 4.10: Simulation results of POMDP solvers for model 1 when DC=55,000.

	TCost		**TRed**		**TRep**		**ACR**
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**
SPM	482,051	132,810	9.02	3.02	10.02	3.02	10.02
SPJ	495,960	116,784	9.40	2.52	10.32	2.51	10.32
PS	511,269	118,109	5.28	2.36	100.00	0.00	100.00
SARSOP	482,104	107,912	4.70	2.16	100.00	0.00	100.00

This scenario's sensitivity analysis results given in Table 4.11 demonstrate that there is no significant difference between performances of POMDP solvers.

Table 4.11: GH test results for model 1 when DC=55,000.

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	482,051	132,810	(458,216, 533,705)	A
SPJ	495,960	116,784	(448,862, 515,241)	A
PS	511,269	118,109	(477,703, 544,835)	A
SARSOP	482,104	107,912	(451,436, 512,772)	A

These results indicate that, for such small problems, the qualities of the exact and the approximate solutions are very close to each other. Exact solution algorithms are expected to achieve better results than approximate algorithms. However, after PS returns the exact policy, it is simulated using DBNs in order to evaluate the total maintenance cost. Since there is the observation sampling process during the simulation, randomness occurs.

It is worth to mention that PS can generate policies for such a small problem. By increasing the number of components in the system we investigate the maximum state space the exact solver can handle.

4.4.2. Experimental Results of Model 2

For the next experiment, it is aimed to analyze the performances of solvers for a system having two components, one process node, and the observation node. The process node is defined as the result of the interaction between the components. The observation node O is used to gather information provided by the process node P in the model. The evaluations of policies generated by POMDP solvers are shown in Table 4.12.

Table 4.12: Simulation results of POMDP solvers for model 2 when DC=5,000.

	TCost		**TRed**		**TRep**		**ACR**	
	Avg	**Std.**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**
SPM	80,398	20,520	15.94	4.07	15.94	4.07	7.28	8.66
SPJ	77,128	20,110	15.36	4.00	15.32	3.98	9.36	5.96
PS	80,093	16,118	15.98	3.20	15.72	2.99	9.26	6.46
SARSOP	75,209	18,307	14.98	3.63	14.78	3.59	8.86	5.92

The solvers are compared in terms of their performance and the results are given in Table 4.13. Although the SARSOP policy achieves better total maintenance costs, there is no significant difference between the four solvers.

Table 4.13: GH test results for model 2 when DC=5,000.

Solver	Avg. Total Cost	Std Dev.	95% CI	GH Group
SPM	80,398	20,520	(74,566, 86,230)	A
SPJ	77,128	20,110	(71,413, 82,843)	A
PS	80,093	16,118	(75,512, 84,674)	A
SARSOP	75,209	18,307	(70,007, 80,412)	A

For the system having two components, the majority of the actions are performed reactively at the downtime cost of 5,000, but still, the small number of proactive maintenance is achieved. To illustrate, the average total red signal received for the PS solver is 15.98; the average total component replacement is 15.72. This difference is detected in the policy file and confirmed during the simulation. When both components are changed in successive periods, the policy proposes the “do nothing” action in the next period, even if the system receives red observation. As this is rare, TRed is slightly smaller than TRep in the replication results. We have increased the downtime cost to 10,000 to favor the policies to behave more proactively and investigate different behaviors of solvers. Performances of the solvers and the GH test results are given in Table 4.14 and Table 4.15 at the downtime cost of 10,000 respectively.

As the number of components in the system increases, the probability of system downtime also increases. Therefore, more proactive policies have been achieved at a downtime cost of 10,000 to avoid the system downtime. As a result of the sensitivity analysis, there is no statistically significant difference between the total maintenance costs obtained by carrying out the policies. As seen in the table, all selected algorithms managed to solve this complex but small scale problem efficiently.

Table 4.14: Simulation results of POMDP solvers for model 2 when DC=10,000.

	TCost		**TRed**		**TRep**		**ACR**	
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**
SPM	140.877	37.483	13,28	3,83	17,70	3,96	9,62	8,08
SPJ	153.786	37.193	13,72	3,45	21,70	5,04	11.24	10,46
PS	138.354	38.749	12,48	3,77	19,04	4.82	10.02	9,02
SARSOP	154.747	38.415	13,96	3,73	21,14	4,72	11,16	9,98

Table 4.15: GH test results for model 2 when DC=10,000.

Solver	**Avg. Total Cost**	**Std.**	**95% CI**	**GH Group**
SPM	140,877	37,483	(130,225, 151,530)	A
SPJ	153,786	37,193	(143,216, 164,356)	A
PS	138,354	38,749	(127,342, 149,367)	A
SARSOP	154,747	38,415	(143,829, 165,664)	A

4.4.3. Experimental Results of Model 3

As a next step, one more component has been added to make the system more complex. As can be seen in Figure 4.1, the process P1 is defined as the result of the interaction between C1 and C2. In model 3, C3 is directly connected to the main processing node P. Table 4.16 summarizes the results based on each solver at a downtime cost of 5,000 and the GH test results are given in Table 4.17 at the downtime cost of 5,000.

PS solver could not handle this POMDP model, whereas the approximate methods generate good policies. Since the real-life model will be more complex than the size of this problem, it has been decided not to use PS as a solver. The results show that SPM and SPJ behave always reactively, while SARSOP occasionally behaves proactively at the downtime cost of 5,000. According to the Games–Howell post hoc test,

Table 4.16: Simulation results of POMDP solvers for model 3 when DC=5,000.

	TCost		TRed		TRep		ACR		
	Avg	Std	Avg	Std.	Avg	Std	C1	C2	C3
SPM	125,910	24,985	24.56	4.82	24.56	4.82	7.98	9.54	7.04
SPJ	120,400	24,598	23.54	4.76	23.54	4.76	9.80	7.32	6.42
SARSOP	128,838	26,854	21.58	4.86	31.80	5.69	6.90	11.08	13.82

Table 4.17: GH test results for model 3 when DC=5,000.

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	125,910	24,985	(118,810, 133,011)	A
SPJ	120,400	24,598	(113,410, 127,391)	A
SARSOP	128,838	26,854	(121,207, 126,470)	A

since the groups have the same letter designation, there are no statistical differences between them.

In order to favor SPM and SPJ solvers to generate policies that recommend having maintenance before the red signal is received, the downtime cost has been increased to 10,000. The comparison of the maintenance policies of the solvers and the GH test results are given in Table 4.18 and Table 4.19 respectively.

Table 4.18: Simulation results of POMDP solvers for model 3 when DC=10,000.

	TCost		TRed		TRep		ACR		
	Avg	Std	Avg	Std.	Avg	Std	C1	C2	C3
SPM	238,303	51,034	22.10	5.08	30.00	5.57	11.90	10.28	7.82
SPJ	230,465	48,016	21.08	4.73	29.58	5.47	10.08	11.30	8.20
SARSOP	239,357	45,057	20.38	4.48	34.70	5.60	11.78	10.26	12.66

Table 4.19: GH test results for model 3 when DC=10,000

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	238,303	51,034	(223,799, 252,806)	A
SPJ	230,465	48,016	(216,818, 244,111)	A
SARSOP	239,357	45,057	(226,552, 252,162)	A

All three solvers behave more proactively as expected. SARSOP policy has performed more proactive actions than the others, however, SPJ gives better results. Notably, the solvers do not statistically distinguish from each other in terms of the total maintenance cost.

4.4.4. Experimental Results of Model 4

For the last experiment, a system having components nodes affecting each other, three process nodes, and one observation node to represent a more complex system is designed. The processes P1 and P2 are defined as the result of the interaction between their predecessor components. The main process node P is directly linked to the observable node O collecting information from the P. The POMDP policies generated via solvers are executed to compare their performances. Simulation summaries and GH test results are given in Table 4.20 and Table 4.21.

Table 4.20: Simulation results of POMDP solvers for model 4 when DC=5,000.

	TCost		**TRed**		**TRep**		**ACR**			
	Avg	**Std**	**Avg**	**Std.**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**
SPM	164,960	24,493	31.72	4.67	31.72	4.67	9.68	7.80	8.08	6.16
SPJ	163,855	26,602	31.50	5.07	31.46	5.10	9.52	8.40	7.16	6.38
SARSOP	167,380	27,164	33.38	5.30	33.16	5.37	9.42	7.58	5.46	10.70

The results demonstrate that although the difference between the total mainte-

Table 4.21: GH test results for model 4 when DC=5,000

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	164,960	24,493	(157,999, 171,921)	A
SPJ	163,855	26,602	(156,294, 171,415)	A
SARSOP	167,380	27,164	(159,660, 175,101)	A

nance costs of solvers is not statistically significant, SPM has achieved better results than the other solvers. It is interesting that SPM, which performs only reactive maintenance actions during the planning horizon, gives the best results, although the other two solvers perform slightly proactive maintenance actions. The solver behaviors are further investigated at the downtime cost 10,000. The comparison of the maintenance policies of the solvers and the GH analysis results are given in Table 4.22 and Table 4.23 at the downtime cost of 10,000.

Table 4.22: Simulation results of POMDP solvers for model 4 when DC=10,000.

	TCost		**TRed**		**TRep**		**ACR**			
	Avg	**Std**	**Avg**	**Std.**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**
SPM	320,049	49,691	29.86	4.74	38.04	5.91	7.44	8.84	11.64	10.12
SPJ	318,678	56,267	28.06	5.29	42.76	6.85	9.58	11.38	12.10	9.70
SARSOP	329,223	56,513	31.20	5.60	37.68	5.90	6.70	11.20	10.38	9.40

Table 4.23: GH test results for model 4 when DC=10,000

Solver	Avg. Total Cost	Std.	95% CI	GH Group
SPM	320,049	49,691	(305,927, 334,171)	A
SPJ	318,678	56,267	(302,687, 334,669)	A
SARSOP	329,223	56,513	(313,163, 345,284)	A

All three solvers generate proactive maintenance policies. However, while SPM and SPJ policies behave in a similar way, it can be said that SARSOP performs less proactive maintenance than the other two solvers. As a result of the sensitivity analysis, there is no statistically significant difference between the costs obtained by carrying out the policies generated by the four different solvers discussed.

4.4.5. Evaluation of Experimental Results

In this chapter, it is focused on comparing the performances of different four POMDP solvers in terms of the total maintenance cost, time and memory needed to solve the problem. The SARSOP solver is run under time constraint of 300 seconds. According to results, the PS solver using exact algorithms, is able to solve a maximum of 2-component model. Moreover, according to results, the quality of selected approximate solvers are very close to each other. As a result of all experiments done, there is no significant difference between the solvers in terms of the total maintenance cost.

The solution has two stages: Policy Generation and Simulation (Evaluation). In the first stage, each solver produce a policy file containing actions and the corresponding α-vectors. The policies generated by PS, SARSOP and SPJ are simulated using DBNs. SPM uses its own simulator to execute the policy. The comparison of solvers in terms of time performance is given in Table 4.24. Due to the time advantage, SPM is preferred as the solver for further studies.

Table 4.24: Computational time performances of the solvers (in seconds).

	PS		**SARSOP**		**SPJ**		**SPM**	
	Solver	**Simulation**	**Solver**	**Simulation**	**Solver**	**Simulation**	**Solver**	**Simulation**
M1	104.24	1,771.48	300	1,819,39	2.24	2,249.70	10,85	5.44
M2	449.81	2,131.69	300	2,154,63	3.21	3,234.01	12,98	7.92
M3	-	-	300	2,738.74	4,48	4,247.45	18,94	14.79
M4	-	-	300	3,499.78	6,82	3,883.68	30,91	24.59

5. POLICY ANALYSIS OF AN EXPERIMENTAL MODEL USING FACTORED POMDPS

5.1. Model Building

In this chapter, main motivation is to show how to formulate a factored POMDP model for the maintenance problem of a multi-component dynamic system and how to simulate and evaluate the obtained policy before implementing it in real life. The model in hand is an empirical dynamic system built to represent the complexity of a real-life maintenance problem. There are symbolically four hidden components degrading over time, three processes and one observable node. The relationships between components and processes are shown in Figure 5.1.

5.1.1. System Structure

Each component has 4 states which are "working", "degraded 1", "degraded 2" and "fail" where "degraded 2" is assumed to be a worse state than "degraded 1". The processes are determined as the result of the interaction of the preceding components and their interactions, and they are unobservable. The main process node P3, which is also not observed, is directly connected to the observable node used to gather information. Direct observation of the components and the processes is not possible. However, their states can be estimated through noisy signals obtained from the observation node. Observation node has 3 states which are "green", "yellow" and "red". It is assumed that all components can be replaced at any time. All components' initial states are "working".

5.1.2. Action Structure

Maintenance activities are defined in accordance with real-life maintenance models. Thus, two different maintenance activities have been defined as "minor" and

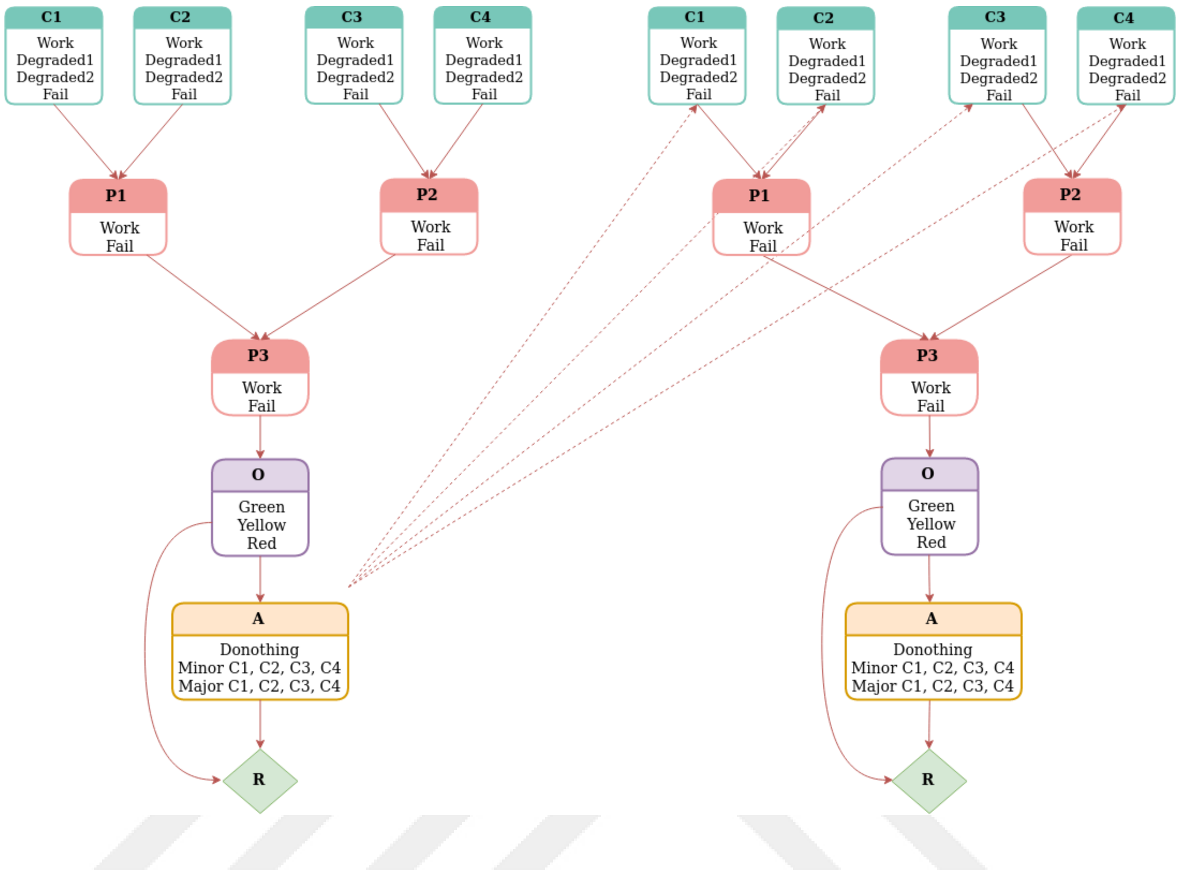


Figure 5.1: Empirical POMDP model for two time slots.

"major". Minor maintenance involves small maintenance activities that require only labor costs. It incurs much less cost than major maintenance; however, it does not provide a guarantee for improvement (imperfect repair). On the other hand, a major maintenance activity behaves as a perfect repair since it replaces the respective component.

5.1.3. Probability Structure

Minor maintenance activities do not guarantee success. If minor maintenance is successful, the state of the component can only pass to the better state. By way of example, a component which is in the "degraded 2" state, after minor maintenance has been performed, passes to "degraded 1" if the maintenance activity is successful. A component in a “fail” state cannot pass to any better states with minor maintenance, it remains in its “fail” state. Major maintenance for a component in a “fail” state is essential. When major maintenance is performed, all components transit to their “working” state with probability 1. Components C1, C3 and components C2, C4 have the same degradation probabilities respectively. Components C2 and C4 deteriorate more slowly than components C1 and C3. Processes P1 and P2 are defined by probabilistic gates, hence, even if one of the components is in its fail state, the related process node can continue to work with a low probability. When both components fail, the related process also fails. The process P3 is defined by the OR gate. Thus, if any of P1 and P2 is in its fail state, P3 is in its fail state. The accuracy of the observations is probabilistic. The observation node is more sensitive to the system downtime. When the system fails, a red signal is received with very close to certainty. Transitional and causal probabilities are given in Tables C.1-C.3 respectively.

5.1.4. Cost Structure

The labor cost of the minor maintenance activity for each component is taken as 500. It is assumed that minor maintenance does not incur production loss. When the green or yellow signal is observed, the total cost of the major maintenance activity for each component consists of the replacement cost of the relevant component, the labor cost and the cost of lost production for a limited time. On the other hand, when a red signal is received, the system is in its fail state with a very high probability. Thus, a downtime cost (DC) incurs because of the system halt. This downtime cost includes the penalty cost due to unrealized production commitments, overtime labor costs and all other types of costs that are caused because of unplanned downtime. That’s why this cost is left as a parameter in this study so that sensitivity analyses are conducted for various levels of downtime costs with respect to different real-life problems. Maintenance costs depending on the observation received and the maintenance activity performed can be seen in Table 5.1 where DC stands for the downtime cost and all costs are in Turkish Lira.

Table 5.1: Maintenance costs.

	Observation		
Action	**Green**	**Yellow**	**Red**
Do nothing	0	0	DC
Minor-C1	500	500	500+DC
Minor-C2	500	500	500+DC
Minor-C3	500	500	500+DC
Minor-C4	500	500	500+DC
Major-C1	2,000	2,000	2,000+DC
Major-C2	2,200	2,200	2,200+DC
Major-C3	2,400	2,400	2,400+DC
Major-C4	2,600	2,600	2,600+DC

5.2. Proposed Methodology

The empirical maintenance problem is first formulated as a factored POMDP and it is solved for a maintenance policy using Symbolic Perseus, a factored POMDP solver. Some smart predefined policies by imitating the behavior of the POMDP policies are proposed to compare the performance of the maintenance policy obtained by the POMDP solver. Sensitivity analyses are conducted under various scenarios with several cost and probability parameters to achieve stronger results. The policy is analyzed by restricting and extending the action space under several downtime cost values and success rates of minor actions. Furthermore, the policies have been analyzed under different aging behaviors of the components to test the robustness of the POMDP policies in various realistic domains.

5.2.1. Factored POMDP Model

The flat POMDP state variable s in the model is a composite variable comprised of four components and three process variables. If the flat representation was to be used, the transition matrix for each action state would be a 2048×2048 square matrix. Because each of the four components has four states and each of the three processes has two states ($4^4 \times 2^3 = 2048$). On the other hand, factored representation of the model reduces this cumbersome transition matrix to factored four 4×4 square matrices (for the four components, each having four states), two 16×2 matrices (for conditional probabilities of P1 and P2) and finally one 4×2 matrix (for the conditional probability of P3). By exploiting the conditional independence in the factored model, the state transition of the flat POMDP state variable s can be calculated by using the conditional probabilities in the factored model as in Equation 5.1. To emphasize how the state transition in a POMDP is reduced by factoring, the sizes of the matrices in the left and right sides of Equation 5.1 are given in Equation 5.2. Let X, be a random variable, x and x' denotes the values of X at time t and $t+1$.

$$\begin{aligned} \mathrm{P}(s'|s,a) = {} & \mathrm{P}(c1'|c1,a)\ .\ \mathrm{P}(c2'|c2,a)\ .\ \mathrm{P}(c3'|c3,a)\ .\ \mathrm{P}(c4'|c4,a) \\ & .\ \mathrm{P}(p1'|c1',c2')\ .\ \mathrm{P}(p2'|c3',c4')\ .\ \mathrm{P}(p3'|p1',p2') \end{aligned} \tag{5.1}$$

$$[\ ]_{2048\times2048} \rightarrow [\ ]_{4\times4}\ ,\ [\ ]_{4\times4}\ ,\ [\ ]_{4\times4}\ ,\ [\ ]_{4\times4}\ ,\ [\ ]_{16\times2}\ ,\ [\ ]_{16\times2}\ ,\ [\ ]_{4\times2} \tag{5.2}$$

In the given maintenance model, the flat POMDP action variable a is also another composite variable which can be factored into four nodes each for one component and having three states (Do nothing, Do minor, Do major). The transition matrix of each of the action state for each of the respective component can be modeled with a decision diagram in the factored representation of the maintenance problem in SPM and can be called in the definition of the actions. Let the following decision diagrams are defined: DefaultC_i is transition probability of C_i when no maintenance is done to C_i, MinorC_i is the transition probability of C_i when minor maintenance is done to C_i, MajorC_i is the transition probability of C_i when major maintenance is done to C_i and finally SameP_i is the conditional probability of P_i which is not affected by the transitions. To give an example, let "DoMajorC3" be an action state in the flat POMDP model meaning that minor maintenance will be performed to C3 and no maintenance will be done to the other components. So other than C3, the others will age with their default transition probabilities.

In the maintenance problem, it is assumed that only one maintenance action can be performed at a time slot. However, one may want to include also the combinations of actions as we will do in the sensitivity analysis. In the flat representation, each extra action will increase the state transition by one 2048×2048 matrix which will cause the explosion of the input file further. By using the factored representation and the decision diagrams, it is possible to define the combination of any actions in the maintenance problem easily as shown in Figure 5.3 where the defined action "DoMajorC1C3" does major maintenance to both C1 and C3, but nothing to C2 and C4 at a time slot.

```
action DoMajorC3
   C1 (DefaultC1)
   C2 (DefaultC2)
   C3 (MajorC3)
   C4 (DefaultC4)
   P1 (SAMEP1)
   P2 (SAMEP2)
   P3 (SAMEP3)
   observe ...
   cost ...
endaction
```

Figure 5.2: Code example for actions in SP.

```
action DoMajorC1C3
    C1 (MajorC1)
    C2 (DefaultC2)
    C3 (MajorC3)
    C4 (DefaultC4)
    P1 (SAMEP1)
    P2 (SAMEP2)
    P3 (SAMEP3)
    observe ...
    cost ...
endaction
```

Figure 5.3: Code example for combined actions in SP.

5.2.2. Predefined Policies

To analyze the performance of the policy generated by the POMDP solver, SPM results are compared with some predefined corrective and proactive maintenance strategies. In the design of predefined policies, three important criteria, which are maintenance decisions of time, component and action, are considered. Time decision is made by means of corrective (Cor) or proactive (Pro) policies. In corrective maintenance, the red signal always triggers the maintenance. In proactive policies, a yellow signal always initiates major or minor maintenance according to the respective policy. Moreover, when a green signal is observed, proactive maintenance decision is taken depending on the predefined number of consecutive green signals.

Once a maintenance time is decided, component selection is done with a random (RND) or an ordered (ORD) method. As their names imply, in the random method, components are selected randomly whereas, in the order method, they are selected in the order of their component numbers. In all predefined policies, when a red signal is received, major maintenance is performed at the selected component. At a proactive maintenance time (when either yellow or green signal is received) after component selection is done, one needs lastly to decide on the type of maintenance that is to select whether major (Maj) or minor (Min) activity. All proposed predefined maintenance policies are illustrated in Figure 5.4. To give an example for the meaning of the names of the predefined policies, in “MinProORD”, when a green or yellow signal is received, a minor maintenance is done by the selecting components in the order of their component numbers, otherwise in the case of receiving a red signal, a major maintenance is performed by selecting components in order of their component numbers.

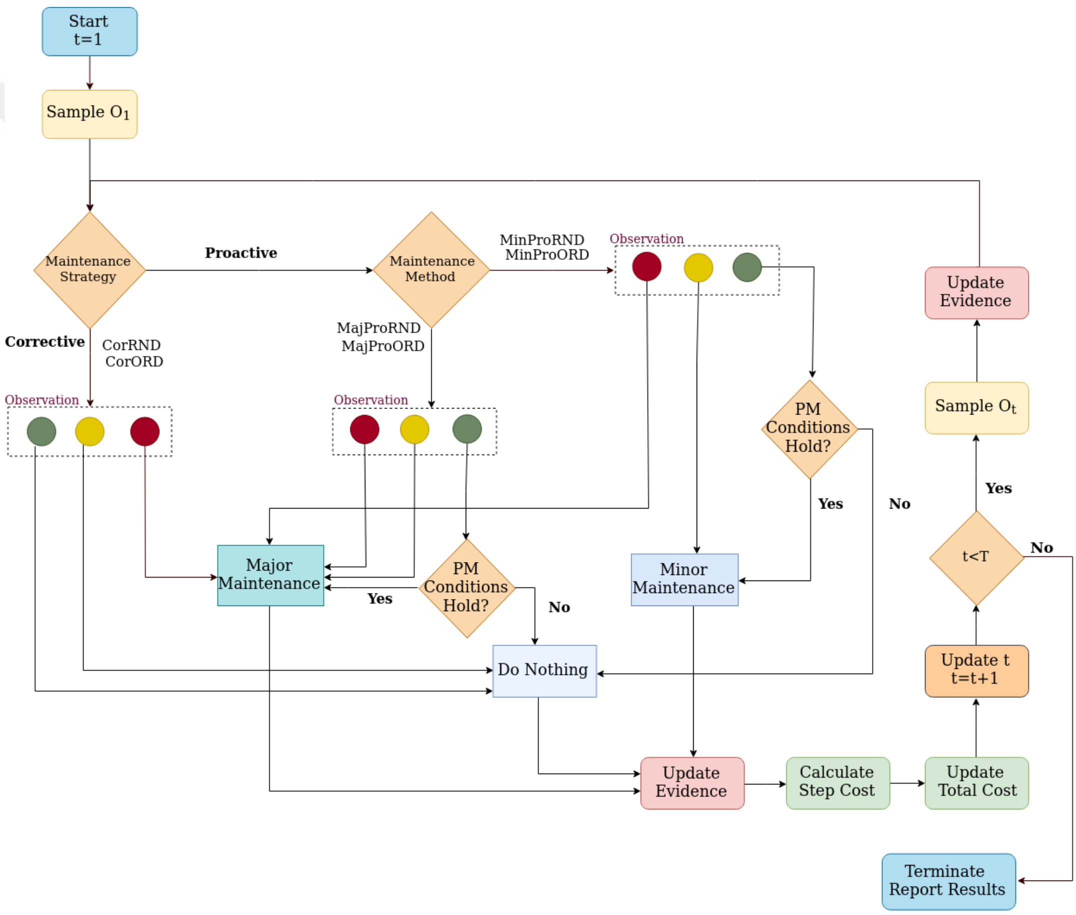


Figure 5.4: Overview of predefined policies.

5.3. Experimental Results

For all experimental analyses in this study, the policy is simulated for a 100 day decision horizon with 50 simulation runs. The discount factor is set to 0.999.

Table 5.2: Experimental design for SPM.

Exp.	**p**			**DC**		**Action Space**			**Location**
	0.3	**0.5**	**0.7**	**500-6,000**	**10,000-100,000**	**Majors**	**Majors, Minors**	**Majors, Minors, Combinations**	
1	*			*			*		Section 5.3.2
2		*		*			*		
3			*	*			*		
4	*				*		*		Section 5.3.3
5		*			*		*		
6			*		*		*		
7				*	*	*			Section 5.3.4.1
8	*			*				*	Section 5.3.4.2
9			*	*				*	
10	*				*			*	
11			*		*			*	

5.3.1. Scenario Design

In this study, various scenarios have been created and analyzed with several cost and probability parameters. The success probability (p) of minor activities for the scenarios is taken as 0.3, 0.5 and 0.7. The downtime cost (DC) has been experimented at different levels in the range of [500-100,000]. The policy is also analyzed by restricting and extending the action space under several DC values and the success rates of minor actions. The maintenance actions within the action space are analyzed in three levels. In the base scenarios, the action space covers minor and major actions. In the restricted scenarios, the action space is restricted to only major actions whereas, in the extended scenarios, combinations of two major actions are also included in the action space. The experimental design for the SPM policy simulation is tabulated in Table 5.2.

In addition to these experiments, the performances of the predefined proactive and corrective maintenance policies have also been assessed and compared to the poli-

Table 5.3: Experimental design for predefined policies.

Exp.	Aging Behavior		p	DC			f			Location
	Similar	Dissimilar	0.7	1,000	7,000	70,000	1-3	1-9	3-19	
1	*		*	*					*	Section 5.3.5.1
2	*		*		*			*		
3	*		*			*	*			
4		*	*	*					*	Section 5.3.5.2
5		*	*		*			*		
6		*	*			*	*			

cies generated via the POMDP solver for DC values of 1,000, 7,000 and 70,000. In the constructed model, components have almost the same aging probabilities. That is why, we change the deterioration probabilities of the components to obtain dissimilar aging behaviors so that analyses in all kinds of real-life systems are also covered. The experimental design for the predefined strategies is tabulated in Table 5.3 where f denotes the frequency of the proactive maintenance, in consecutive number of green signals received, which is used as a checkpoint for proactive maintenance. For instance, when f=3, it means that the proactive maintenance condition in the predefined policies is satisfied when three consecutive green signals are observed.

5.3.2. Sensitivity to Minor Repair Probability

Success probability (p) of minor maintenance is taken as 0.3, 0.5 and 0.7, respectively of which the results of the sensitivity analyses are given in Tables 5.4-5.6. In the tables, TCost and TRed denote the total cost of the given horizon and the total number of red signals observed in that horizon respectively. TRep is the sum of the total number of repairs due to minor maintenance and major maintenance performed in the given horizon. The averages (Avg) and the standard deviations (Std) of these measurements are given in the tables. Besides, the distribution of the minor and major maintenance among the components are also reported.

Table 5.4: Sensitivity analysis under different DC values with p=0.3.

DC	TCost		TRed		TRep		Minor Maintenance					Major Maintenance				
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**	**Total**	**C1**	**C2**	**C3**	**C4**	**Total**
500	41,921	3,514	88.46	7.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,000	71,241	14,211	27.98	5.12	20.94	4.43	0.00	0.00	0.00	0.00	0.00	7.54	4.22	5.96	3.22	20.94
2,000	102,580	17,822	20.22	4.59	59.28	4.79	17.72	5.82	12.24	1.96	37.74	6.86	5.14	5.68	3.86	21.54
3,000	120,555	18,333	15.82	3.86	91.36	3.03	26.48	4.36	40.60	0.86	72.30	5.98	5.12	3.50	4.46	19.06
4,000	138,562	22,712	15.52	4.11	100.00	0.00	44.74	0.00	31.96	4.64	81.34	4.22	4.90	5.34	4.20	18.66
5,000	153,191	24,269	16.00	3.75	100.00	0.00	3.68	0.00	72.00	6.00	81.68	8.52	4.86	1.78	3.16	18.32
6,000	162,484	30,703	14.86	4.05	100.00	0.00	17.44	13.32	47.68	3.40	81.84	7.26	3.94	2.96	4.00	18.16

Table 5.5: Sensitivity analysis under different DC values with p=0.5.

DC	TCost		TRed		TRep		Minor Maintenance					Major Maintenance				
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**	**Total**	**C1**	**C2**	**C3**	**C4**	**Total**
500	41,921	3,514	88.46	7.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,000	69,105	14,956	28.82	6.15	37.52	6.40	0.00	0.00	12.88	9.90	22.78	6.00	3.78	3.38	1.58	14.74
2,000	95,141	16,052	18.02	4.39	70.64	6.38	3.66	8.58	27.92	13.60	53.76	7.86	4.02	2.62	2.38	16.88
3,000	115,825	18,815	15.24	4.23	100.00	0.00	1.78	0.00	69.84	12.68	84.30	8.32	4.48	1.10	1.80	15.70
4,000	130,648	22,798	14.74	4.20	100.00	0.00	26.82	0.00	37.62	19.78	84.22	2.50	7.08	2.48	3.72	15.78
5,000	146,798	31,319	14.74	4.84	100.00	0.00	43.54	2.66	32.68	4.20	83.08	2.64	6.54	3.44	4.30	16.92
6,000	158,446	35,095	14.88	4.60	100.00	0.00	0.14	4.74	45.34	33.38	83.60	8.80	4.04	1.80	1.76	16.40

Table 5.6: Sensitivity analysis under different DC values with p=0.7.

DC	TCost		TRed		TRep		Minor Maintenance					Major Maintenance				
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**	**Total**	**C1**	**C2**	**C3**	**C4**	**Total**
500	42,130	3,464	87.90	7.51	0.98	1.27	0.42	0.00	0.56	0.00	0.98	0.00	0.00	0.00	0.00	0.00
1,000	67,001	16,155	29.60	6.91	37.86	7.30	0.00	8.28	5.16	11.44	24.88	6.16	2.10	3.64	1.08	12.98
2,000	90,209	16,764	18.78	4.55	64.02	6.10	0.00	7.04	34.08	7.68	48.80	8.32	3.16	1.64	2.10	15.22
3,000	110,011	17,632	14.22	3.95	99.00	0.00	1.36	1.66	54.80	26.94	84.76	7.34	4.30	1.48	1.12	14.24
4,000	121,242	24,606	13.24	4.46	100.00	0.00	41.18	19.22	15.26	10.88	86.54	2.88	2.64	4.60	3.34	13.46
5,000	138,684	28,224	14.28	4.39	100.00	0.00	2.86	38.22	23.22	21.16	85.46	7.94	2.26	2.32	2.02	14.54
6,000	148,062	27,655	13.20	3.66	100.00	0.00	51.54	5.68	27.46	1.36	86.04	1.64	3.88	1.80	6.64	13.96

As can be seen from the tables above, the average total cost increases as expected as the DC increases in all three success probability values. Moreover, as the DC increases, the average total number of red signals decreases and the average total number of maintenance increases. In other words, as the DC increases, the policy tends to maintain more proactively without waiting for a red signal. For the first two levels of success probability (p=0.3 and p=0.5), the DC of 500 is quite low for the system, thus the system stops continuously without performing any maintenance activity. Besides, when the success probability of the minor action is quite small, i.e., p= 0.3, it does not perform minor maintenance even at a DC of 1,000. In all three models, the system is more proactive as the DC increases, thus more minor actions are performed. Furthermore, as the success probability of minor actions increases, the total cost decreases for the same DC value since the number of minor (major) maintenance performed in the policy relatively increases (decreases). Since P3 has an OR gate probability structure, all components should effort to make it work. In all three levels of success probability scenarios, when there is a remarkable imbalance in the major maintenance distribution among the components, the ones with less share there have significantly more share in the minor maintenance distribution. In this context, C3 seems to be more (less) preferred in the minor (major) maintenance.

5.3.3. Sensitivity to Downtime Cost

The average total number of red signals vs the average total number of minor and major maintenance with respect to DC values in [500,6000] are shown in Figure 5.5 for p=0.3, 0.5 and 0.7 respectively. After a DC of 3,000, all policy indicators (TRed, TRep, TRep-Minor, TRep-Major) almost reach a steady state. The policy prefers to perform either minor or major maintenance in all periods of which minor maintenance has a significantly greater share than major maintenance. It can be concluded that the policy behaves almost in a similar way in all three success probability scenarios. Although the model has the opportunity to decrease the total number of red signals received by performing more effective actions such as major maintenance, the policy does not prefer this due to the cost structure of the model.

It would be interesting to see whether the policy tends to perform more major maintenance with considerably increasing DC values. Hence, in the next step, the DC of the system is increased extremely in the range of [10,000;100,000] to encourage the system to perform more major actions. The policy results are depicted graphically in Figure 5.6 for the three levels of success probability of minor actions.

As can be seen from Figure 5.6, as the DC increases, the number of major maintenance actions increases as expected and the number of red signals received falls slightly consequently. In each success probability value, the policy favors more major maintenance actions as the DC increases when compared to the respective policy behavior in Figure 5.5. This justifies that the policy behaves expectedly at extreme DC values. Higher success probabilities of minor maintenance actions lead to a lower deceleration rate of minor maintenance tendency of the policy as seen in Figure 5.6 especially when p=0.7. It should be highlighted that the DC where minor and major maintenance has an equal share (50 of each) in total maintenance number is approximately 90,000 for p=0.7. This value, where the equilibrium is provided, is greater than the two other DC values at p=0.3 and p=0.5. For p=0.3 and p=0.5, when the DC is at its utmost level in the sensitivity study, the policy favors major maintenance instead of the minor ones almost in all periods without waiting for a red signal to escape from the very high DC. For p=0.7, the aforementioned dominance effect of the major maintenance comes later with greater DC values than 100,000. Because minor actions, in these scenarios, has a stronger effect on the system performance due to the fact that their success probability is relatively higher than the other two scenarios. Notably, the policy behavior in the p=0.7 case differentiates obviously than the other two cases for extremely large DC values as seen in Figure 5.6 when compared to smaller DC values as seen in Figure 5.5.

5.3.4. Sensitivity to Action Space

There are three available actions per each component, i.e. *do nothing*, *minor maintenance*, *major maintenance* in the base model. In this section, two different scenarios of action spaces are considered, restricted which includes only major mainte-

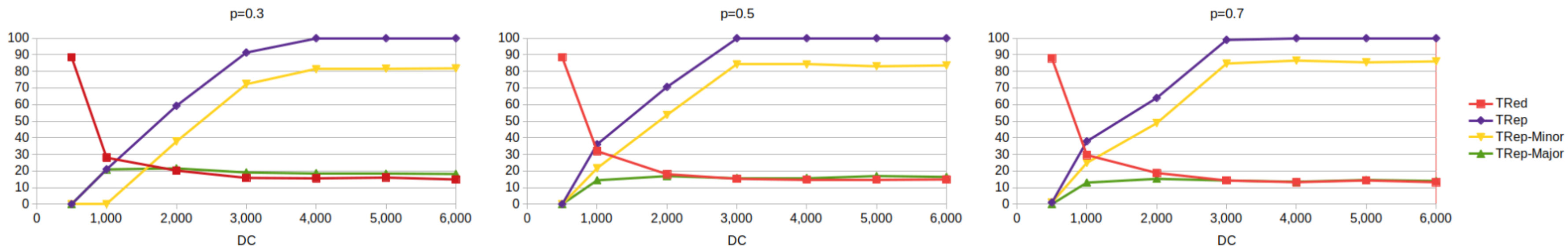


Figure 5.5: Behavior of the SPM policy when DC is in [500; 6,000].

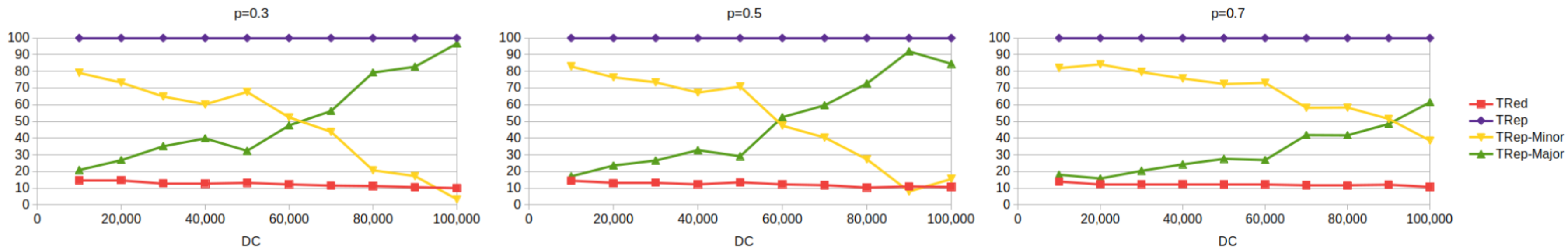


Figure 5.6: Behavior of the SPM policy when DC is in [10,000; 100,000].

nance and extended action space which includes also combined maintenance to carry out sensitivity analyses under different downtime costs.

5.3.4.1. Restricted Action Space. Minor maintenance improves a component state by one with a success probability of p only if it is in the "degraded 2" or "degraded 1" state. Besides, minor maintenance only includes the labor cost of 500 for each component and it does not lead to production loss. To assess the impacts of minor actions on the maintenance policy, minor actions were removed from the model in this section. The results of the sensitivity analyses are given in Table 5.7.

Table 5.7: Sensitivity analysis under different DC values for restricted action space.

DC	TCost		TRed		TRep		Major Maintenance				
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**C1**	**C2**	**C3**	**C4**	**Total**
500	42,467	3,592	89.60	7.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,000	74,304	12,149	26.18	4.08	23.24	3.90	8.78	4.48	6.56	3.42	23.24
3,000	119,750	19,310	24.28	3.96	23.88	3.84	9.52	4.56	6.66	3.14	23.88
7,000	204,448	34,256	21.90	3.94	27.74	4.25	10.96	4.38	9.48	2.92	27.74
10,000	261,555	48,456	19.88	4.25	34.40	4.07	15.04	4.92	10.88	3.56	34.40
30,000	537,587	113,802	12.34	3.91	88.06	1.63	39.08	23.44	8.22	17.32	88.06
70,000	953,894	256,252	11.14	3.84	100.00	0.00	41.22	23.90	11.36	23.52	100.00
100,000	1,194,318	402,139	10.30	4.24	100.00	0.00	36.78	24.62	14.54	24.06	100.00

5.3.4.2. Extended Action Space. As a further study, the assumption of replacing only one component at a time has been extended; new actions that replace two components at a time have been added. Thus, action space has been extended to 14 actions for the system. The new actions added to action space are "replace C1 and C2", "replace C1 and C2", "replace C1 and C4", "replace C2 and C3", "replace C2 and C4" and "replace C3 and C4". For combined major actions, the component replacement costs are calculated by adding the replacement costs of the relevant components. The total cost of major maintenance activities consists of the labor cost of major maintenance, the replacement costs of the relevant components and the downtime cost. It is important to note that one DC incurs when a red signal is received. The results of the sensitivity analyses are given in Table 5.8 and Table 5.9 for p=0.3 and p=0.7 respectively.

Table 5.8: Sensitivity analysis under different DC values for extended action space with p=0.3.

Downtime Cost	**TCost**		**TRed**		**TRep**		**Maintenance**		
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**Minor Total**	**Major Total**	**Combination Total**
500	41,921	3,514	88.46	7.13	0.00	0.00	0.00	0.00	0.00
1,000	70,600	13,674	30.72	5.15	10.76	2.67	0.00	2.30	8.46
3,000	111,844	18,427	15.60	2.57	15.60	2.57	0.00	0.00	15.60
7,000	162,971	28,098	11.34	2.80	100.00	0.00	88.46	1.38	10.16
10,000	195,719	39,248	11.26	2.98	100.00	0.00	88.62	0.66	10.72
30,000	386,263	85,391	9.06	2.71	100.00	0.00	78.80	0.04	21.16
70,000	672,738	173,129	6.38	2.59	100.00	0.00	48.40	0.00	51.60
100,000	806,619	201,054	4.92	2.12	100.00	0.00	24.18	0.00	75.82

Table 5.9: Sensitivity analysis under different DC values for extended action space with p=0.7.

Downtime Cost	**TCost**		**TRed**		**TRep**		**Maintenance**		
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**Minor Total**	**Major Total**	**Combination Total**
500	42,180	3,463	87.90	7.51	1.08	1.34	1.08	0.00	0.00
1,000	71,812	15,842	26.36	5.13	31.18	5.00	22.88	0.00	8.30
3,000	108,777	23,414	12.56	4.13	81.00	2.97	71.56	0.58	8.86
7,000	159,406	30,041	10.72	2.88	100.00	0.00	17.74	0.16	10.56
10,000	175,261	32,466	9.66	2.45	100.00	0.00	90.34	0.22	9.44
30,000	351,958	78,888	9.24	2.41	100.00	0.00	89.14	0.24	10.62
70,000	652,081	172,547	6.62	2.54	100.00	0.00	56.66	0.00	43.34
100,000	813,462	223,462	5.04	2.36	100.00	0.00	24.90	0.00	75.10

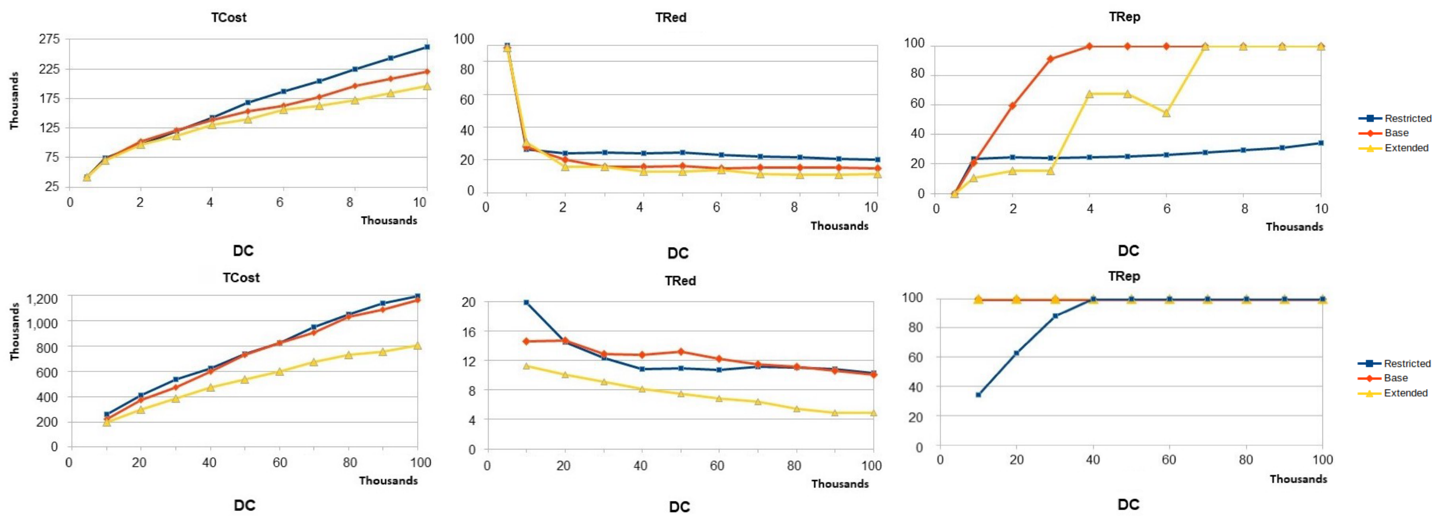


Figure 5.7: Action space scenarios when DC is in [500; 10,000] and [10,000; 100,000] with p=0.3.

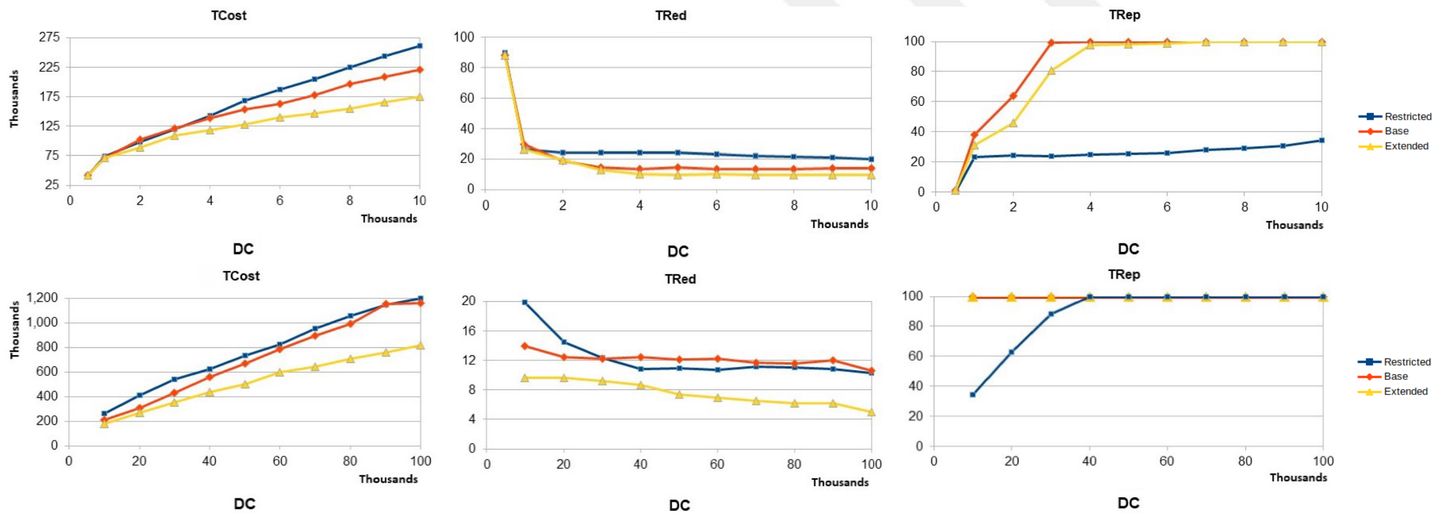


Figure 5.8: Action space scenarios when DC is in [500; 10,000] and [10,000; 100,000] with p=0.7.

5.3.4.3. Comparison of Action Space Scenarios. Comparison of the action space scenarios (restricted, base and extended models) are given in Figure 5.7 and Figure 5.8 with respect to TCost, TRed and TRep with increasing DC values ([500; 10,000] and [10,000; 100,000]) for the two success probability values respectively. In both success probability scenarios, TCost of the extended model is the lowest after a DC value of 3,000 where proactive maintenance starts in the policy. As DC increases step by step, the success of the extended model in reducing TCost, due to having the alternative of performing combined actions at a time point, can be seen more obviously. Moreover, the base model is also more successful than the restricted model due to having the alternative of performing minor actions at a time point. For extreme DC values in the range [10,000; 100,000], the base model approaches the restricted model in TCost, especially after DC=40,000. This is because of the fact that minor maintenance does not help to reduce the total cost and major maintenance starts to replace them for extreme DC values. Hence, it can be concluded that using minor maintenance has its advantage in DC values greater than 4,000 and less than 40,000 when p=0.7.

TRed of the base and the extended model is lower than the TRed of the restricted model until a DC value of 20,000. After this value, the base model does not result in a lower number of red observations whereas it succeeds to keep its cost at most equal to the cost of the restricted model. That is to say, although the base model may result in more TRed, it is still effective in reducing its total cost.

The most remarkable difference between p=0.3 and p=0.7 scenarios with respect to TRep is that the base and the extended model reach the maximum number of maintenance, i.e., TRep=100, quicker when p=0.7. The more a policy favors minor maintenance, the quicker it reaches the full maintenance capacity utilization.

5.3.5. Comparisons with Predefined Policies

In this section, we compare the performance of the policy generated by the POMDP solver with the predefined policies of which the details are already explained in Section 5.2.2. The comparisons are done under two main headings according to the deterioration behavior of the components. In Section 5.3.5.1, the system is handled with the aging probabilities already used in the previous sensitivity analyses where the components deteriorate with similar probabilities. Components C1 and C3 have a working transitional probability of 0.97 whereas components C2 and C4 deteriorate slightly faster with a transitional probability of 0.95. To make our analyses to be valid also for the systems with components aging in dissimilar probabilities, we disturb the balance of the system by differentiating the aging with distinct transitional probabilities which are 0.99, 0.96, 0.93 and 0.90 for C1, C2, C3 and C4 respectively.

Based on the previous sensitivity results, it is observed that the policies result in more minor actions when p=0.7, justifying the inclusion of minor actions in the action space. That is why, in this section, all comparisons are conducted under the success probability of p=0.7. Three downtime cost values are decided, on purpose, according to the previous results such that corrective, minor proactive and major proactive strategies all show up themselves at one of the DC values. The smallest DC value is set to 1,000 from the findings in Table 5.6 where the SPM policy starts doing corrective actions. On the other hand, a high value of DC=70,000, where maintenance is done in all periods, is selected because of the significancy of both minor and major actions in the policy. For an intermediate DC value, 7,000 is chosen to be used in the comparisons.

5.3.5.1. Comparisons under Similar Aging Behaviors. When the components have almost similar deterioration probabilities, the behaviors of the policies in terms of the ratios of the total maintenance costs to the total cost of the SPM policy are given for a DC value of 1,000 with increasing maintenance frequencies in Figure 5.9. At such a small DC value, it is obvious from the SPM policy that maintenance is not required significantly. Furthermore, since the proactive policies convergence to the corrective counterparts as proactive maintenance frequency increases, the maximum frequency f in this part of the study is decided to be set to a reasonably high value. Due to these facts, we take f at values greater than one in the range [3; 19]. For the statistical comparison, a Games-Howell post-hoc test is applied due to the lack of homoscedasticity.

The best promising frequency values of each method are chosen for the statistical test. The grouping results are available in Table 5.10. The same letter indicates the policies which are insignificant at a significance level of 0.05 in terms of the total maintenance cost. Compared with the predefined strategies, SPM policy achieves significantly less maintenance cost than all the others. Each of the RND method, as expected, perform significantly worse than the respective ORD method. Since selecting the right component to be maintained at the right time is very essential in maintenance decisions, random methods are not successful in component selection at a maintenance time. This is because there is a possibility that a component in good condition can be maintained consecutively and a component in poor condition may not be maintained for a long time. Although there is no statistical difference between the strategies using the ORD method, the performance of the CorORD policy is the one closest to the performance of the SPM. This behavior can be explained by the very low DC value. Under the DC value of 1,000, the SPM policy performs proactive maintenance rarely and generally shows a corrective behavior.

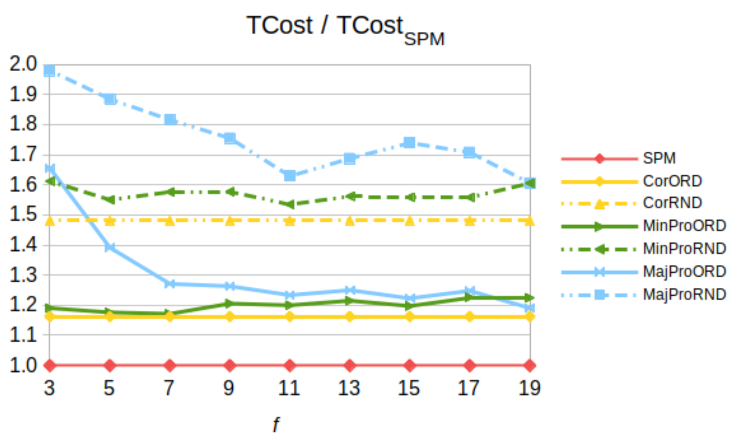


Figure 5.9: Maintenance policies at DC=1,000 with similar aging behaviors.

The performances of the policies are investigated in terms of the total maintenance cost for the DC value of 7,000. We test with different f values in the range [1;9]. The ratios of the total maintenance costs to the total cost of the SPM policy are given in Figure 5.10 whereas post-hoc test results are given in Table 5.11. According to the results, the SPM policy yields the minimum cost compared to the other policies. However, the performances of the SPM and MinProORD (f=1) methods are not

Table 5.10: GH results at DC=1,000 with similar aging behaviors.

Strategy	*f*	TCost	GH
SPM	-	67,001	C
CorORD	-	77,907	B
MinProORD	7	78,498	B
MajProORD	19	79,791	B
CorRND	-	99,212	A
MinProRND	1	101,276	A
MajProRND	19	107,587	A

significantly different. The reason of this is that both SPM and MinProORD (f=1) policies prefer to perform minor proactive maintenance to ensure that the system will work without receiving the red signal. ORD methods yield better results than RND methods. This is because of the similar aging behavior and replacement costs of the components, so that selecting the components in an order provides a more successful policy. In the case of systems with dissimilar aging behavior and costs, ORD methods may not be so successful. At DC=7,000, doing minor proactive maintenance almost at every period unless a red signal is observed, $f = 1$, performs significantly better than both corrective and major proactive counterparts.

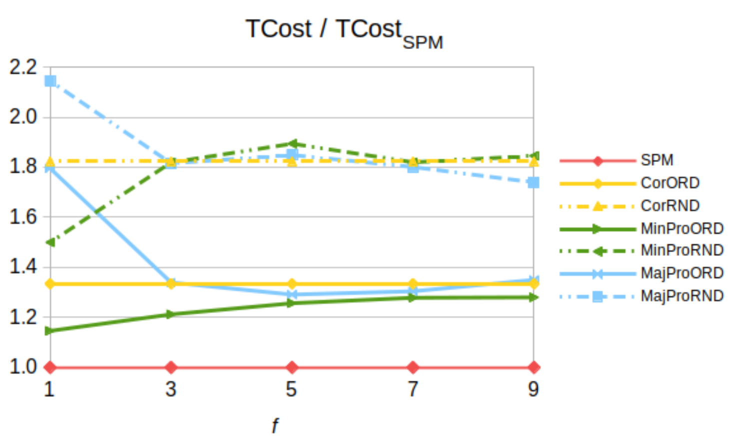


Figure 5.10: Maintenance policies at DC=7,000 with similar aging behaviors.

Table 5.11: GH results at DC=7,000 with similar aging behaviors.

Strategy	***f***	**TCost**	**GH**
SPM	-	162,393	E
MinProORD	1	186,109	D,E
MajProORD	5	209,643	C,D
CorORD	-	216,650	C
MinProRND	1	243,503	B,C
MajProRND	9	282,489	A,B
CorRND	-	296,137	A

We investigate the performance of the SPM policy at an extreme DC value of 70,000. Because of the very high DC value, frequent proactive maintenance should be essential. The behavior of the policies are illustrated in Figure 5.11 for f values of 1 and 3 whereas post-hoc test results are given in Table 5.12. It is interesting to see that both MajProORD and MinProORD results do not distinguish from SPM results significantly at $f = 1$ and DC=70,000. CorORD and CorRND result in worse policies than the other proactive policies at $f = 1$, which shows that corrective methods are not preferable at such a huge DC value. Furthermore, this DC value is the one where proactive (both major and minor) policies are found to be statistically close to the SPM policy.

5.3.5.2. Comparisons under Dissimilar Aging Behaviors. In the previous section, some predefined policies using the ORD component selection method give insignificant results compared to the SPM policy. An interesting question arises whether this will be valid for the systems having components with dissimilar aging behavior. Thus, the deterioration rates of the components are changed such that the transitional working probabilities of the components become as 0.99, 0.96, 0.93 and 0.90 for C1, C2, C3 and C4 respectively. The predefined policies using the ORD method are tested again at the same DC and frequency values as in the previous section. The results are given in

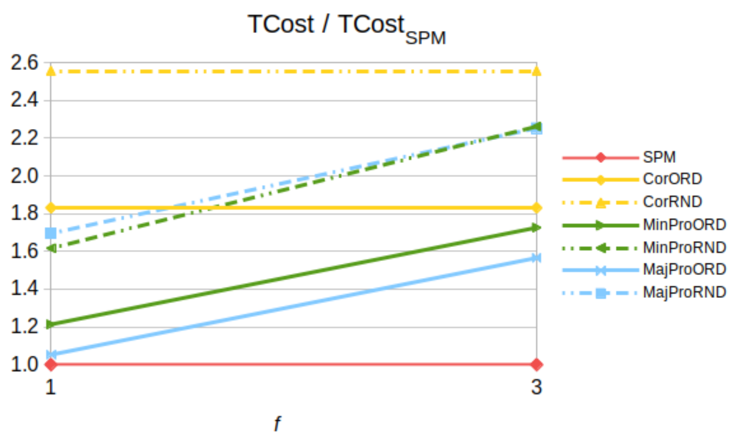


Figure 5.11: Maintenance policies at DC=70,000 with similar aging behaviors.

Table 5.12: GH results at DC=70,000 with similar aging behaviors.

Strategy	***f***	**TCost**	**GH**
SPM	-	891,416	C
MajProORD	1	938,193	C
MinProORD	1	1,081,340	C
MinProRND	1	1,440,899	B
MajProRND	1	1,511,439	B
CorORD	-	1,632,404	B
CorRND	-	2,279,184	A

Figures 5.12, 5.13 and 5.14 Tables 5.13, 5.14 and 5.15 respectively. When the behavior of the policies and post-hoc results are investigated, it can be concluded that SPM policy distinguishes significantly from all the predefined policies at all DC values tested. This result indicates that although the performance of the predefined policies depends on the system structure, SPM always gives robust policies independent of the system structure. One can try to improve the quality of a maintenance policy by scrutinizing the SPM policy up to a point. However, that policy does not guarantee to achieve the best performance in all cases. Another important point to mention is that in a predefined policy, the maintenance decisions of time and components are decoupled

which prohibits finding the optimum joint maintenance decision.

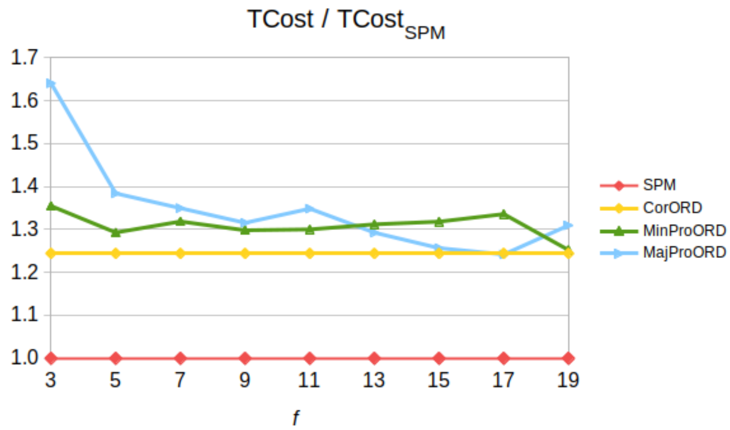


Figure 5.12: Maintenance policies at DC=1,000 with dissimilar aging behaviors.

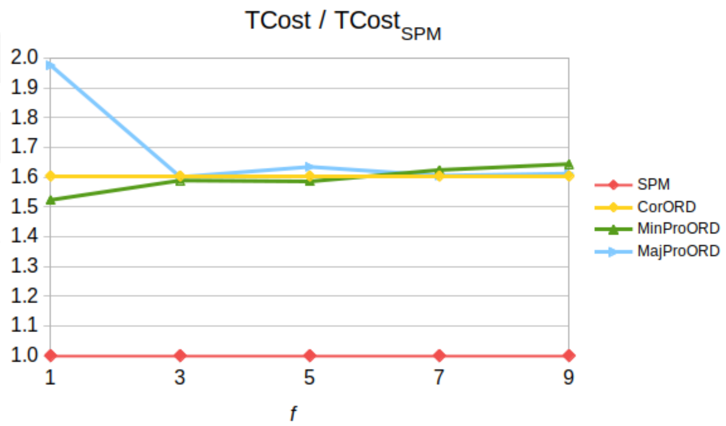


Figure 5.13: Maintenance policies at DC=7,000 with dissimilar aging behaviors.

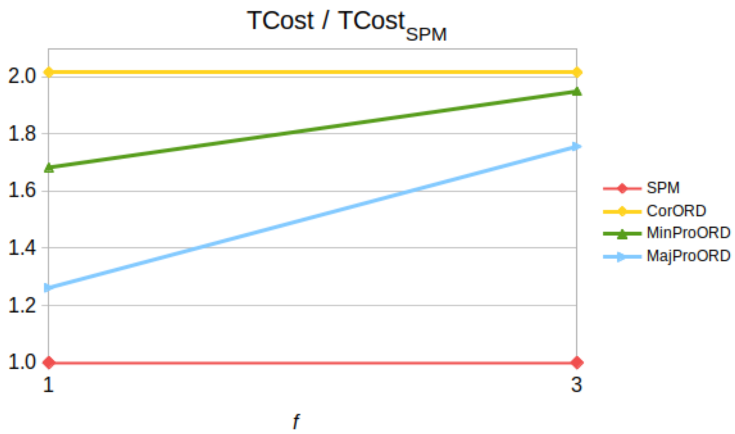


Figure 5.14: Maintenance policies at DC=70,000 with dissimilar aging behaviors.

Table 5.13: GH results at DC=1,000 with dissimilar aging behaviors.

Strategy	***f***	**TCost**	**GH**
SPM	-	73,123	B
MajProORD	17	90,813	A
CorORD	-	90,973	A
MinProORD	19	91,557	A

Table 5.14: GH results at DC=7,000 with dissimilar aging behaviors.

Strategy	***f***	**TCost**	**GH**
SPM	-	159,423	B
MinProORD	1	242,858	A
MajProORD	3	255,163	A
CorORD	-	255,669	A

Table 5.15: GH results at DC=70,000 with dissimilar aging behaviors.

Strategy	***f***	**TCost**	**GH**
SPM	-	961,006	D
MajProORD	1	1,212,615	C
MinProORD	1	1,617,387	B
CorORD	-	1,937,044	A

6. POLICY ANALYSIS OF THE REGENERATIVE AIR HEATER SYSTEM USING FACTORED POMDPS

In the previous chapters, experimental models are built to represent the complexity of a real-life maintenance problem using factored POMDPs. In this chapter, regenerative air heater (RAH), which is one of the major subsystems of a thermal power plant is formulated as a factored POMDP and effective policies are obtained for the maintenance problem of the RAH under various scenarios.

6.1. Model Building

The RAH is the most important element of the air-gas system in coal-based thermal power plants. It consists of a motor group (ball bearing, winding-insulation, rotor-shaft), hub reduction gear, RAH insulation and honeycombs. The working mechanism of the RAH is shown in Figure 6.1. The RAH is used to heat the air. In the RAH system, moving on a rotating component, the air and gas passing through the honeycomb consisting of a set of plates exchange heat. The gas losing its heat goes to the electro filter and the heated air is transferred to the boiler to dehumidify the coal. For detailed information of the RAH system please refer [138, 139] where the RAH system consists of two parallel motor lines and one observation setting related to the RAH exit temperature. In this study, only one complete line of the system is considered. Additionally, an auxiliary observation is added to the rotor shaft to gather information from the motor group in the model.

6.1.1. System Structure

The maintenance problem of the RAH is formulated as a POMDP model. The relationships between nodes are shown in Figure 6.2. The model has four node types: dynamic nodes, process nodes, exogenous nodes and observation node. Dynamic and process nodes are represented by purple and pink respectively, while observation and

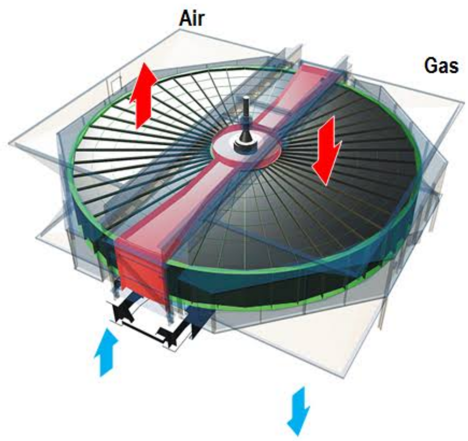


Figure 6.1: Regenerative air heater [140].

exogenous nodes are indicated by orange and blue respectively in the figure. The arrows indicated with “1” denote the temporal relationships between two successive time slices. Other arrows represent the causal relations between the nodes. The condition of the ball bearing affects both the winding insulation and the rotor shaft. Hence, there exits stochastic dependency between them. Furthermore, a structural dependency exists between the honeycomb and the RAH insulation since maintaining the honeycomb requires the maintenance of the RAH insulation. Table 6.1 shows the abbreviations, types and state spaces of all nodes in the POMDP model.

6.1.2. Action Structure

There are six maintenance activities which are represented by green color, as can be seen in Figure 6.2 and Table 6.2 shows the action nodes and their state spaces. When a “Do nothing” action is performed, the system experiences the natural deterioration process. The relevant component is replaced with a new one by “Replace” action. The “Clean” action in the honeycomb symbolizes the special cleaning of the combs using chemicals. The “Grind” action in the rotor shaft also represents the grinding of the rotor in the case of an axis shift.

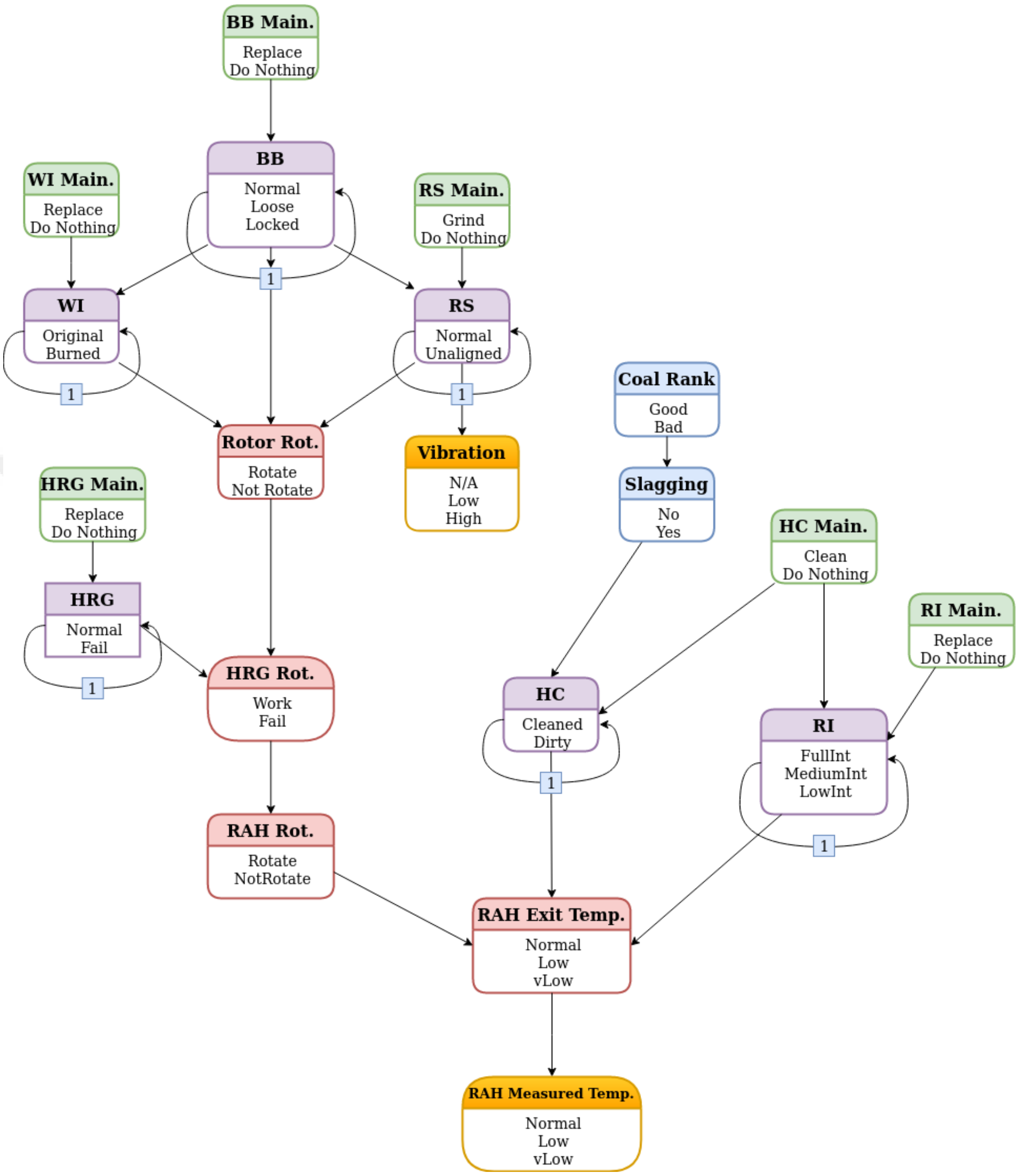


Figure 6.2: POMDP model of the RAH system.

6.1.3. Probability Structure

Transitional probabilities of the components and conditional probabilities of the nodes and the observation node are given in Tables D.1-D.13 respectively. In this study, probabilities in [139] are revisited to handle the problem within the POMDP environment. The accuracy of the observations is probabilistic. The RAH temperature

Table 6.1: Nodes and state spaces of the RAH.

Nodes	Abbrev.	Node Type	Observability	State Space
Ball Bearing	BB	Dynamic	Non-Observable	{Normal, Loose, Locked}
Winding Insulation	WI	Dynamic	Non-Observable	{Original, Burned}
Rotor Shaft	RS	Dynamic	Non-Observable	{Normal, Unaligned}
Hub Reduction Gear	HRG	Dynamic	Non-Observable	{Normal, Fail}
Honeycomb	HC	Dynamic	Non-Observable	{Cleaned, Dirty}
RAH Insulation	RI	Dynamic	Non-Observable	{FullInt, MediumInt, LowInt}
Slagging	Slag.	Exogenous	Non-Observable	{No, Yes}
Coal Rank	CR	Exogenous	Non-Observable	{Good, Bad}
Rotor Rotation	RR	Process	Non-Observable	{Rotate, NotRotate}
HRG Rotation	HRG Rot.	Process	Non-Observable	{Rotate, NotRotate}
RAH Rotation	RAH Rot.	Process	Non-Observable	{Rotate, NotRotate}
RAH Exit Temperature	RAH Exit	Process	Non-Observable	{Normal, Low, VLow}
RAH Measured Temperature	RAH Temp.	Observation	Observable	{Normal, Low, VLow}
Vibration	Vib.	Observation	Observable	{N/A, Low, High}

Table 6.2: Action nodes in the POMDP model.

Action Nodes	Action Space	Affected Component
BB Main.	{Replace, Do Nothing}	BB
WI Main.	{Replace, Do Nothing}	WI
RS Main.	{Grind, Do Nothing}	RS
HRG Main.	{Replace, Do Nothing}	HRG
RI. Main.	{Replace, Do Nothing}	RI
HC Main.	{Clean, Do Nothing}	HC, RI

is more sensitive to the system downtime.

6.1.4. Cost Structure

The maintenance costs incurred at each t depends on the observation received, the maintenance action performed, maintenance duration, and the unit downtime cost. The total maintenance cost consists of the downtime cost and the repair of the relevant component. A unit downtime cost of 10,000 TL per hour incurs when a normal or low temperature measurement is observed, and the system is assumed to have a unit downtime cost of 25,000 TL per hour when a very low (VLow) temperature measurement is received. Maintenance costs of each component including both the downtime costs (obtained by multiplying the maintenance time required with the unit downtime cost) depending on the observations received and also the cost of the action taken are given in Table 6.3 where DC denotes the downtime cost.

Table 6.3: Maintenance costs of the RAH system.

Component	**Main. Dur. (hour)**	**Observation**			
		No/Low		**VLow**	
		Action Cost	**DC**	**Action Cost**	**DC**
BB	1	1,000	10,000	2,000	25,000
WI	4	7,500	40,000	15,000	100,000
RS	4	750	40,000	1,500	100,000
HRG	2	1,000	20,000	2,000	50,000
RI	2	50	20,000	100	50,000
HC	6	800	60,000	1,600	150,000

6.2. Proposed Methodology

The RAH system is formulated as a factored POMDP and it is solved for a maintenance policy using Symbolic Perseus implemented in Matlab (SPM) [58]. To model the RAH with the flat representation, it is required to employ $13,824 \times 13,824$ square matrix for each action to create the state transition matrix. However, by exploiting the conditional independence in the factored model, this huge burden of the transition matrix can be reduced. Equation 6.1 illustrates how the state transition in a flat POMDP is reduced by factoring the elements of the RAH model. Using the factored POMDP enables us to represent the huge matrix demonstrated in the left hand side of the equation with the small instance matrices in the right hand side.

$$
\begin{aligned}
[\,RAH\,]_{13{,}824\times 13{,}824} \rightarrow & [\,BB\,]_{3\times 3}\,,\ [\,WI\,]_{6\times 2}\,,\ [\,RS\,]_{6\times 2}\,,\ [\,HRG\,]_{12\times 2}\,,\\
& [\,HRGRot\,]_{4\times 2}\,,\ [\,RAHRot\,]_{2\times 2}\,,\ [\,CR\,]_{2\times 1},\\
& [\,Slag\,]_{2\times 2},\ [\,HC\,]_{4\times 2},\ [\,RI\,]_{3\times 3},\ [\,RAHExit\,]_{12\times 3}
\end{aligned} \tag{6.1}
$$

6.3. Computational Results

The discount factor is set to 0.9999. SPM is run infinitely to generate maintenance policies and the policies are implemented on the system during a 900 working day horizon (three years excluding the scheduled maintenance duration) with 100 replications. Sensitivity analyses are conducted with several unit downtime costs. Two predefined corrective policies are proposed to compare the performance of the maintenance policy obtained by the POMDP solver. The policy is analyzed to identify the critical components of the system in terms of the probabilities and also maintenance costs. Various scenarios with exogenous variables are designed to highlight their impact on the RAH system. Furthermore, the policy is analyzed by restricting and extending the observation space and by adding an inspection node to the action space.

6.3.1. Comparisons with Predefined Policies

In order to analyze the performance of the policy generated by the POMDP solver, two predefined corrective maintenance policies, CorORD and CorRND proposed in Chapter 5 are used where the components are maintained randomly or in order by node ID when a VLow measurement is observed. Results of sensitivity analysis for the unit downtime cost of 25,000, are given in Table 6.4 where TCost, TVLow, TRep and TRepPro denote the total cost, the total number of VLow measurements received, the total number of replacements and the total number of proactive replacements respectively in 900 days of a horizon. Average and standard deviation of these measures are reported in the table. Furthermore, the average total replacements for each component are also given. In order to make a consistent comparison, besides the base POMDP policy, a policy called "SPM-Cor" is generated which does not provide any cost advantage to the proactive maintenance in the POMDP model, thus forcing the policy to act reactively. The cost of CorRND policy is considerably higher than that of the Order policy indicating the importance of effective component selection. Although the CorORD strategy gives much better results than the Random strategy; its cost is still higher than that of both SPM policies. The reason for this can be explained by the fact that the RAH consists of different components in terms of cost and aging behavior structure. For instance, SPM policy maintains BB more than the others, while the Order strategy leads a balanced selection policy. Moreover, maintaining components in order can cause unnecessary maintenance. For example, as can be seen from Table 6.4, the honeycomb is maintained on the average 3.38 times with the CorORD strategy. Considering that the reactive downtime cost of the honeycomb is 150,000 TL, it is obvious that 570,000 of the total cost comes from the honeycomb maintenance. This situation demonstrate the success of condition-based maintenance strategies. The obtained POMDP policies become more significant since the maintenance actions are determined adaptively using the beliefs from the information provided by the observation and action history. An additional important finding is that the SPM-Base policy is able to perform maintenance proactively before waiting for a VLow measurement when compared to the SPM-Cor policy due to the cost advantage of the proactive maintenance. Thus, a cost saving of 171,183 TL is achieved via the proactive maintenance performed.

Table 6.4: Comparison with predefined policies when unit DC=25,000.

Policy	**TCost**		**TVLow**		**TRep**		**TRepPro**		**Avg. Comp. Replacements**					
	Avg	**Std**	**Avg**	**Std**	**Avg**	**Std**	**Avg**	**Std**	**BB**	**WI**	**RS**	**HRG**	**RI**	**HC**
SPM-Base	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00
SPM-Cor	847,999	353,088	18.12	8.18	18.13	8.17	0.01	0.10	9.61	1.72	1.67	3.84	1.29	0.00
CorORD	1,800,589	949,227	21.26	10.65	21.26	10.65	0.00	0.00	3.94	3.92	3.80	3.46	3.20	2.94
CorRND	3,547,248	1,423,050	44.82	16.38	44.82	16.38	0.00	0.00	7.98	7.68	7.42	7.08	7.08	7.58

6.3.2. Sensitivity to Unit Reactive Downtime Costs

Results of sensitivity analysis with respect to several unit reactive downtime cost values, are given in Table 6.5. According to the results, as downtime cost increases, the average total number of VLow temperature measurements decreases whereas the average total number of replacements increases. It can be said that more replacements are performed proactively before waiting for a VLow measurement as downtime cost increases. An important finding of these replication results is that the policy does not prefer to maintain the honeycomb up to the downtime cost of 100,000. This may be due to the high maintenance duration of the honeycomb. Since honeycomb's maintenance duration is relatively long compared to the other components, its maintenance is costly. Therefore, the general behavior of the policies obtained is to maintain the system by improving the condition of the other components, rather than maintaining the honeycomb at a high cost. As can be seen from the table, the policy recommends the average 1.2 honeycomb maintenance. It is essential to highlight that all of this maintenance is proactive rather than reactive.

The average total number of VLow measurements, the average total number of replacements and the average total number of proactive replacements for different downtime cost values are depicted in Figure 6.3 where it is seen obviously that TRep and TProRep increase as downtime cost increases. On the other side, TVLow first does not change, then decreases by the increase in downtime cost. An additional finding is that as the unit downtime cost increases, the system tends to perform more proactive maintenance because high downtime cost triggers the system to behave more proactively to escape from unexpected system halts.

6.3.3. Sensitivity to Costs of Components

The RAH is a complex system that involves components with different aging behaviors, maintenance duration, and dependencies among the components. In this section, a scenario called "SPM-SameCosts" is developed to generate a policy consid-

Table 6.5: Sensitivity analysis under different unit reactive DC values.

URDC	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
25,000	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00
50,000	1,327,721	558,467	14.53	6.56	18.57	7.94	4.04	2.10	8.95	1.42	1.67	3.28	3.25	0.00
75,000	1,926,710	767,268	13.48	5.71	19.74	7.48	6.26	2.56	8.77	1.27	1.81	2.97	4.92	0.00
100,000	2,125,404	924,524	12.22	5.32	20.64	6.28	8.42	2.05	8.09	1.40	1.71	2.91	5.33	1.20

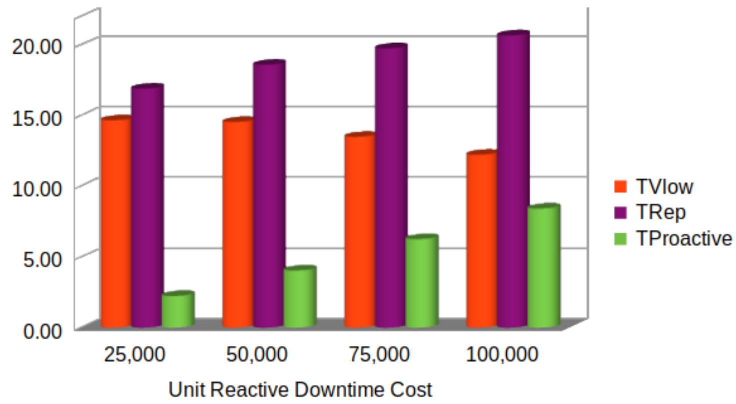


Figure 6.3: TVLow, TRep and TProRep.

ering only probabilities of the components in order to understand the importance of the components regardless of the maintenance cost. This is achieved by setting the costs of all components to the median of all costs in the POMDP model. The results are given in Table 6.6.

It is worthwhile to mention that the policy recommends maintaining the honeycomb more than twice; however, it does not offer any maintenance to the RAH insulation. This is because of the structural dependency; that is the activity of maintaining the honeycomb affects both the honeycomb and the RAH insulation. As a result, after the maintenance of honeycomb, both the honeycomb and the RAH insulation pass to their best states. Since it is possible to perform both maintenances at a single cost, the policy does not prefer to perform maintenance on the RAH insulation. The second important finding is that most of the analyses performed with different settings do not provide honeycomb maintenance, while the SPM-SameCosts policy proposes on the average 4.81 maintenance. This is due to the exclusion of the high maintenance cost of the honeycomb in this scenario.

On the other hand, SPM-SameCosts policy proposes on the average 2.45 ball bearing maintenance while SPM-Base policy proposes on the average 9.55 ball bearing maintenance. In SPM-Base policy including the costs, as the ball bearing is cheaper than the components, the policy tends to improve the state of the system by maintaining the ball bearing much more than the other components. However, giving too much attention to the ball bearing causes the other components to be deprived of the maintenance they need. The results of the average total number of replacements and

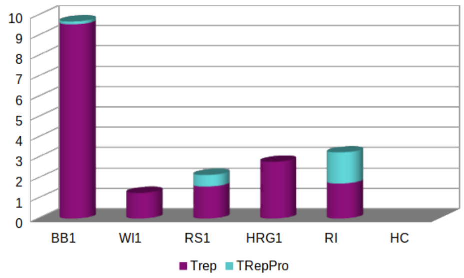
Table 6.6: Sensitivity to costs of the components.

Policy	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
SPM-Base	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00
SPM-SameCosts	1,119,140	500,970	11.30	5.70	13.78	6.25	2.48	1.57	2.45	1.55	1.77	3.20	0.00	4.81

the average total number of proactive replacements proposed by the SPM-Base and the SPM-SameCosts policies are also given in Figure 6.4. As can be seen from the figures, more proactive maintenance activities are offered to the ball bearing and the honeycomb compared to the other components in the SPM-Base policy. This may indicate that these components are critical components of the system. As can be seen in most of the results, the ball bearing is generally maintained much more than the other components.

6.3.4. Sensitivity to Critical Components

According to the results obtained in Section 6.3.3, the major difference between the two cases belongs to maintenance of the ball bearing and the honeycomb. To make further analysis, the honeycomb is favored to be maintained by reducing the maintenance duration to two hours. Another scenario is designed where the ball bearing



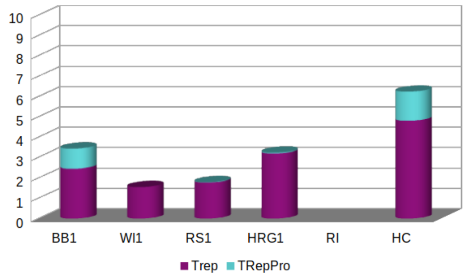


Figure 6.4: Maintenance distribution of the components a) SPM-Base b) SPM-SameCost.

is directed to be maintained less by increasing the maintenance duration to two hours in the base model. The results are given in Table 6.7. As can be seen from the table, the honeycomb is maintained when its duration is reduced. Moreover, the total number of VLow received and the total number of replacements also decrease. In addition, the ball bearing is maintained less. There is also a decrease in the total cost as expected. On the other hand, when the maintenance duration of the ball bearing is increased, it is maintained less. Interestingly, when the total maintenance number of ball bearing reduces, the total maintenance number of the HRG increases. As can be seen in Figure 6.3, HRG is one of the important components of the system. If it does not rotate, the system halts immediately.

Table 6.7: Sensitivity to the maintenance costs.

Policy	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
SPM-Base	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0
SPM-BB (2 hours)	870,357	362,696	13.38	6.07	15.61	6.18	2.23	1.46	2.75	1.65	2.00	5.38	3.83	0.00
SPM-HC (2 hours)	606,882	269,218	10.87	4.82	13.78	5.36	2.91	1.54	5.61	1.34	1.61	3.00	0.00	2.22

6.3.5. Sensitivity to Exogenous Variables

Coal rank and slagging are two important exogenous components for the RAH system. Slagging, which prevents heat exchange, occurs when there exist very small particles of ash on the heat transfer surfaces. The coal quality directly affects slagging. Because low-quality coal first increases the gas temperature and then is slagged. Probability of using low quality coal is taken as 0.5, 0.7 and 0.9 respectively in the sensitivity analysis, of which the results the sensitivity analyses are given in Table 6.8. As the quality of coal decreases, the average total number of VLow signals received and the total maintenance cost also increase. The unexpected results of the case with p=0.9 are discussed in detailed in Section 6.4.

Table 6.8: Sensitivity analysis with different coal rank probabilities.

CR (low)	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
0.5	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00
0.7	769,642	349,826	15.41	7.43	18.28	8.28	2.87	1.85	9.20	1.65	1.89	3.53	2.01	0.00
0.9	9,189,085	6,029,555	353.09	235.25	355.13	235.49	2.04	1.29	349.73	0.98	0.00	2.60	1.82	0.00

As mentioned before, coal quality directly affects slagging. When coal quality is low, the probability of slagging (p) is 0.4 in the base scenario. Sensitivity analyses are conducted for p=0.4, 0.6 and 0.8 respectively and their results are given in Table 6.9. According to the results, as the probability of slagging increases, the average total number of VLow signals received, the total number of maintenance activities performed and the total maintenance cost also increase. One of the noteworthy finding observed in the table is the increase in the number of the RI maintenance as the probability of slagging increases. Under normal circumstances, maintaining the honeycomb is expected because of the causal relation between slagging and the honeycomb. However, the policy avoids to maintain the honeycomb due to its high maintenance cost. Instead, it is preferable to improve the state of the RI, which is relatively cheaper than the honeycomb. To strengthen this statement, a further analysis is conducted by reducing the maintenance duration of HC to four and two hours when of the probability of using low coal quality is 0.9. The results are given in Table 6.10. Obviously, when the honeycomb maintenance duration is reduced to two hours, if the quality of the coal is very low, the policy proposes on the average 2.51 honeycomb maintenance. It is important to state that 2.46 of these maintenance activities are performed proactively.

It can be said that the policy tends to protect the system against the danger of slagging. Therefore, a remarkable decrease in the total number of vlow received is achieved. This is a good example to show the importance of the relationship between costs and probabilities in the maintenance planning of the RAH system.

Table 6.9: Sensitivity analysis with different slagging probabilities.

Slagging	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
0.4	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00
0.6	704,673	311,058	14.67	6.81	17.89	7.66	3.22	1.63	9.53	1.33	1.70	2.83	2.50	0.00
0.8	767,738	315,181	14.96	7.01	18.32	8.00	3.36	1.89	8.68	1.64	1.84	3.43	2.73	0.00

Table 6.10: Sensitivity analysis with HC Duration.

HC Duration	TCost		TVLow		TRep		TRepPro		Avg. Comp. Replacements					
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC
2 hours	660,936	314,515	11.32	5.37	14.70	5.96	3.38	1.48	5.60	1.70	1.68	3.21	0.00	2.51
4 hours	941,999	365,959	19.80	7.08	23.40	7.48	3.60	1.42	12.20	1.80	2.00	4.00	3.40	0.00
6 hours	9,189,085	6,029,555	353.09	235.25	355.13	235.49	2.04	1.29	349.73	0.98	0.00	2.60	1.82	0.00

6.3.6. Sensitivity to Observation Space

The original model [139] has one observation node that only measures the performance of the RAH Exit Temperature. In this study, an auxiliary observation is added to the rotor shaft to improve the quality of the policy by gathering information from the motor group. Two different scenarios are designed. In the "No Auxiliary Obs." scenario, we remove the auxiliary observation receiving signals for each epoch. On the other hand, "Inspection with cost" refers to the scenario that auxiliary observation is received with an inspection cost of 120 TL per inspection action performed. In this case, inspection is defined as an another action in the model. Comparison of the observation space scenarios (no auxiliary observation, inspection and base) are given in Table 6.11 with respect to TCost, TVLow and TRep. Since auxiliary observation provides more information from the engine group, remarkable improvement results are achieved in terms of the total cost, VLow measurements received and the total number of replacements.

Table 6.11: Sensitivity analysis for different observation space.

Policy	TCost		TVlow		TRep		TRepPro		Avg. Comp. Replacements						Ins.
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	BB	WI	RS	HRG	RI	HC	
NoAuxiliaryObs.	895,460	389,846	17.35	7.15	19.91	7.43	2.56	1.16	9.41	2.15	2.31	3.91	2.13	0.00	-
WithInspectionCost	719,480	305,116	15.00	6.77	900.00	0.00	885.00	6.77	9.55	1.34	1.96	2.80	1.86	0.00	882.49
WithAuxiliaryObs. (Base)	676,816	255,864	14.65	6.26	16.89	6.72	2.24	1.36	9.55	1.26	1.57	2.79	1.72	0.00	900

6.4. Discussion

SPM has some major advantages in modeling and solving POMDPs. First, SPM performs basic ADD operations more effectively, thus the backups and updates of the beliefs state are achieved faster. Second, SPM provides the advantage of performing sensitivity analysis in terms of changing in input parameters. For instance, creating, adding or removing elements in the model is highly easy by exploiting its factored structure. However, in some of the experiments performed, we have observed that the SPM is not so successful in generating qualitative policies in some of the scenarios compared to the previous scenario studies achieved in Chapter 5. That is, the generated policy proposes the same component repetitively when successive undesirable observations (very low temperature readings) are received. In order to address this problem, we use more α-vectors and more belief points during solution to improve the quality of the policies obtained from SPM. Although computational effort required increases, the results show no sign of improving in the quality of the policies. The following reasons are believed to be the potential causes of this drawback.

- System structure is complex because of the stochastic dependencies between ball bearing and the two components which are the winding insulation and the rotor shaft.
- The components are not equivalent in terms of the maintenance activities and the downtime effects they cause. For instance, the reactive cost of replacing the

ball bearing is 27,000 TL whereas cleaning the honeycomb incurs 151,600 TL.

- Except for the ball bearing and the RAH insulation, the other four components have two states. Deterioration states such as the ones defined in Chapter 5 (degrade1 and degrade2) would be helpful to improve the quality of the factored POMDP policy.
- Exogenous variables are uncontrollable components of the model. As mentioned in Section 6.3.5, they are highly effective in the model. Therefore, having more states in the modeling of the exogenous variables could improve the quality of the factored POMDP policy. Thus, more adaptive POMDP policies can be produced.

7. CONCLUSION

This research focuses on exploring, formulating, solving and analyzing multi-component maintenance problems through factored POMDPs. Developing maintenance strategies for multi-component systems is quite challenging due to the dependencies in the systems and partial observability inherent in the maintenance problems. POMDPs provide a rich framework for generating effective policies under uncertainty in partially observable stochastic environments and they are particularly suited for tackling maintenance problems. After a comprehensive review of the literature covering a wide range of POMDP solution algorithms and POMDP applications, it is realized that although POMDPs are applied in a wide range of different real-world problems, they have received little attention in the domain of maintenance of multi-component systems because of the fact that POMDP problems suffer from the curse of dimensionality. However, employing factored representations, which is another important finding from the literature, can reduce the complexity of the problems. Although there is a research community actively working on modeling and solving the maintenance problems with one large system node, to the best of our knowledge, there exists no application of maintenance problems through factored POMDPs in the literature, which is the main motivation of this study.

First, existing POMDP solvers available in the platforms are explored and reported in order to decide which is convenient for maintenance problems at hand. To analyze the performance of the selected solvers using flat or factored representations, four maintenance problem settings at increasing complexity levels are designed. POMDP solvers are evaluated by comparing the total maintenance cost and their solution times. According to the results, the exact solver could not generate a successful policy in problems with large state space. Although approximate solvers using both flat and factored representation are successful in solving larger problems, the factored POMDP solver, Symbolic Perseus (SP), stands out with some advantages it provides. In the case of using flat representation, constructing the transition matrix is performed by calculating the joint probabilities for each combination of system state. However, when constructing the state transition function for the factored representation, the complexity is reduced by factoring in multi-component systems by exploiting the conditional independence. Furthermore, SP provides the advantage of performing sensitivity analysis in terms of easy formulation and easy structural changes in the model. For instance, creating, adding revising probabilities in the model, or removing elements and is highly easy by exploiting its factored structure. Therefore, by exploiting factored representations, the eventual goal of this study is to solve the maintenance problems of a multi-component system in a factored partially observable setting using Symbolic Perseus solver.

This study proposes to formulate such maintenance problems as factored POMDPs allowing the complexity of states to be simplified by exploiting the inherent factored structure of maintenance problems. To achieve this purpose, first, an experimental maintenance problem consisting of partially observable components deteriorating in time is designed. The factored POMDP policy is investigated in depth by conducting various sensitivity analyses. Then, the regenerative air heater (RAH) system, which is one of the major subsystems of thermal power plants, is formulated as a factored POMDP.

The first problem involves an empirical dynamic system having symbolically four hidden independent components degrading over time, three processes and one observable node. The experimental model is very rich in terms of the system and action state spaces to provide more suitability to real-life applications. There are two levels of degradation states and there exist two types of maintenance as minor and major, each with its own characteristics. The components are similar in terms of costs and aging. The performance maintenance policy obtained via factored POMDP solver is compared with some smart predefined policies proposed in this study by imitating the behavior of the factored POMDP policies. Sensitivity analyses are conducted under various scenarios with several costs and probability parameters to achieve robust findings. The policy is investigated by restricting and extending the action space under several downtime cost values and success rates of minor actions. Furthermore, the policies are monitored under different aging behaviors of the components to test the robustness of the factored POMDP policy in various realistic domains.

The second model is a variant of a real-life maintenance problem from the literature. It consists of six components having stochastic interdependencies, four processes, two observations, one of them providing auxiliary information and two exogenous variables reflecting the environmental uncertainty. The components have different maintenance costs, maintenance durations and aging. The sensitivity of the policy obtained by the factored POMDP solver is analyzed under several unit downtime costs. The performance of the factored POMDP policy is compared with two predefined reactive policies. The critical components of the system in terms of the probabilities and maintenance costs are identified by various analyses. The effects of the exogenous variables are discussed with different scenarios. The policy is also analyzed by restricting and extending the observation space and by adding an inspection node to the action space.

Results of the experimental study and real-life implementation shows that POMDP-based formulations of maintenance problems are superior to the predefined myopic policies. Number of system halts and total maintenance costs are considerably reduced by factored POMDP policies. However, it should be noted that although the consistency of all scenarios in the experimental factored POMDP model, some of the scenarios in the real-life implementation could not achieve qualitative policies. The main probable reasons are complex stochastic dependencies between some components, huge discrepancy between maintenance costs of the components and lack of deterioration states of components and the effects of exogenous variables.

To sum up, POMDPs are powerful tools to build successful maintenance strategies since the maintenance actions are determined adaptively using the information provided by the observation and action history. On the other hand, factored representations are advantageous in modeling and solving the maintenance problem of multi-component systems due to the inherent factored structure of such problems.

The promising results at hand mostly show that efficient maintenance policies can be generated using factored POMDPs for multi-component maintenance problems.

As a future study, the real-life model can be enriched especially in the problem domain where the system and action state space is extended to enables the belief about the system state to be calculated more correctly. More levels of degradation states can be defined for the components. Parallel to this, action space can be enriched covering also the minor activities. Furthermore, intermediate states can be defined for the exogenous variables.

REFERENCES

1. Kothamasu, R., S. H. Huang and W. H. VerDuin, "System health monitoring and prognostics—a review of current paradigms and practices", *The International Journal of Advanced Manufacturing Technology*, Vol. 28, No. 9-10, pp. 1012–1024, 2006.

2. Kobbacy, K. A. H. and D. P. Murthy, *Complex system maintenance handbook*, Springer Science & Business Media, 2008.

3. Chan, K., R. Lee and J. Burnett, "Maintenance performance: a case study of hospitality engineering systems", *Facilities*, Vol. 19, No. 13/14, pp. 494–504, 2001.

4. Eisemann, R., "Machinery malfunction diagnosis and correction: Vibration analysis and troubleshooting for process industries", , 1998.

5. Mehta, B. and Y. Reddy, "Chapter 18 - System maintenance", B. Mehta and Y. Reddy (Editors), *Industrial Process Automation Systems*, pp. 523 – 545, Butterworth-Heinemann, Oxford, 2015.

6. Gustavsson, E., M. Patriksson, A.-B. Strömberg, A. Wojciechowski and M. Önnheim, "Preventive maintenance scheduling of multi-component systems with interval costs", *Computers & Industrial Engineering*, Vol. 76, pp. 390–400, 2014.

7. Wang, H., "A survey of maintenance policies of deteriorating systems", *European journal of operational research*, Vol. 139, No. 3, pp. 469–489, 2002.

8. Pham, H. and H. Wang, "Imperfect maintenance", *European journal of operational research*, Vol. 94, No. 3, pp. 425–438, 1996.

9. Hipkin, I. B. and C. De Cock, "TQM and BPR: lessons for maintenance manage-

ment", *Omega*, Vol. 28, No. 3, pp. 277–292, 2000.

10. Moubray, J., *Reliability-centered maintenance*, Industrial Press Inc., 2001.

11. Tsang, A. H., "Condition-based maintenance: tools and decision making", *Journal of quality in maintenance engineering*, Vol. 1, No. 3, pp. 3–17, 1995.

12. Nicolai, R. P. and R. Dekker, "Optimal maintenance of multi-component systems: a review", *Complex system maintenance handbook*, pp. 263–286, Springer, 2008.

13. Murthy, D. and D. Nguyen, "Study of two-component system with failure interaction", *Naval Research Logistics Quarterly*, Vol. 32, No. 2, pp. 239–247, 1985.

14. Murthy, D. and D. Nguyen, "Study of a multi-component system with failure interaction", *European Journal of Operational Research*, Vol. 21, No. 3, pp. 330–338, 1985.

15. Zhang, Y., R. Zheng, G. Shen and B. Chen, "Reliability analysis for CNC machine tool based on failure interaction", *International Conference on Intelligent Computing and Information Science*, pp. 489–496, Springer, 2011.

16. Sasieni, M. W., "A Markov chain process in industrial replacement", *Journal of the Operational Research Society*, Vol. 7, No. 4, pp. 148–155, 1956.

17. Dekker, R., R. E. Wildeman and F. A. Van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence", *Mathematical methods of operations research*, Vol. 45, No. 3, pp. 411–435, 1997.

18. Papakonstantinou, K. G. and M. Shinozuka, "Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory", *Reliability Engineering & System Safety*, Vol. 130, pp. 202–213, 2014.

19. Puterman, M. L., *Markov decision processes: discrete stochastic dynamic pro-*

gramming, John Wiley & Sons, 2014.

20. Pearl, J., *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Elsevier, 2014.

21. Heckerman, D., J. Breese and K. Rommelse, *Troubleshooting under uncertainty*, Tech. rep., Technical Report MSR-TR-94-07, Microsoft Research, 1994.

22. Langseth, H. and L. Portinale, "Bayesian networks in reliability", *Reliability Engineering & System Safety*, Vol. 92, No. 1, pp. 92–108, 2007.

23. Murphy, K. P. and S. Russell, "Dynamic bayesian networks: representation, inference and learning", , 2002.

24. Weber, P., G. Medina-Oliva, C. Simon and B. Iung, "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas", *Engineering Applications of Artificial Intelligence*, Vol. 25, No. 4, pp. 671–682, 2012.

25. Bellman, R., "A Markovian decision process", *Journal of mathematics and mechanics*, pp. 679–684, 1957.

26. Özgür-Ünlüakin, D. and İ. Kivanç, "Maintenance policy simulation for a factored partially observable system", *Proceedings of the 2019 Summer Simulation Conference*, p. 26, Society for Computer Simulation International, 2019.

27. Boutilier, C. and D. Poole, "Computing Optimal Policies for Partially Observable Decision Processes Using Compact Representations", *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pp. 1168–1175, AAAI Press, 1996.

28. Hansen, E. A. and Z. Feng, "Dynamic Programming for POMDPs Using a Factored State Representation", *AIPS*, 2000.

29. Hoey, J., R. St-Aubin, A. Hu and C. Boutilier, "SPUDD: Stochastic planning using decision diagrams", *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 279–288, Morgan Kaufmann Publishers Inc., 1999.

30. Hoey, J. and J. J. Little, "Value-directed human behavior analysis from video using partially observable Markov decision processes", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 29, No. 7, pp. 1118–1132, 2007.

31. Dean, T., L. P. Kaelbling, J. Kirman and A. Nicholson, "Planning under time constraints in stochastic domains", *Artificial Intelligence*, Vol. 76, No. 1-2, pp. 35–74, 1995.

32. Howard, R. A., "Dynamic programming and markov processes.", , 1960.

33. Aström, K. J., "Optimal Control of Markov Decision Processes with Incomplete State Estimation", *J. Math. Anal. Appl.*, Vol. 10, pp. 174–205, 1965.

34. Kaelbling, L. P., M. L. Littman and A. R. Cassandra, "Planning and acting in partially observable stochastic domains", *Artificial intelligence*, Vol. 101, No. 1-2, pp. 99–134, 1998.

35. Sondik, E. J., *The optimal control of partially observable Markov processes.*, Tech. rep., STANFORD UNIV CALIF STANFORD ELECTRONICS LABS, 1971.

36. Smallwood, R. D. and E. J. Sondik, "The optimal control of partially observable Markov decision processes over a finite horizon", *Operations Research*, Vol. 21, No. 5, pp. 1071–1088, 1973.

37. Sondik, E. J., "The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs", *Operations research*, Vol. 26, No. 2, pp. 282–304, 1978.

38. Hansen, E. A., "An improved policy iteration algorithm for partially observable

MDPs", *Advances in Neural Information Processing Systems*, pp. 1015–1021, 1998.

39. Monahan, G. E., "State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms", *Management Science*, Vol. 28, No. 1, pp. 1–16, 1982.

40. Papadimitriou, C. H. and J. N. Tsitsiklis, "The complexity of Markov decision processes", *Mathematics of operations research*, Vol. 12, No. 3, pp. 441–450, 1987.

41. Cheng, H.-T., *Algorithms for partially observable Markov decision processes*, Ph.D. Thesis, University of British Columbia, 1988.

42. Littman, M. L., "The witness algorithm: Solving partially observable Markov decision processes", *Brown University, Providence, RI*, 1994.

43. Cassandra, A., M. L. Littman and N. L. Zhang, "Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes", *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pp. 54–61, Morgan Kaufmann Publishers Inc., 1997.

44. Cassandra, A. R., L. P. Kaelbling and J. A. Kurien, "Acting under uncertainty: Discrete Bayesian models for mobile robot navigation", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996.

45. Littman, M. L., A. R. Cassandra and L. P. Kaelbling, "Efficient dynamic-programming updates in partially observable Markov decision processes", , 1995.

46. Drake, A. W., *Observation of a Markov process through a noisy channel*, Ph.D. Thesis, 1962.

47. Brafman, R. I., "A Heuristic Variable Grid Solution Method for POMDPs", *Proceedings of the Fourteenth National Conference on Artificial Intelligence*

and Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI'97/IAAI'97, pp. 727–733, AAAI Press, 1997.

48. Zhou, R. and E. Hansen, "An Improved Grid-Based Approximation Algorithm for POMDPs", , 2001.

49. Bonet, B., "An epsilon-Optimal Grid-Based Algorithm for Partially Observable Markov Decision Processes", *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 51–58, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

50. Pineau, J., G. Gordon, S. Thrun *et al.*, "Point-based value iteration: An anytime algorithm for POMDPs", *IJCAI*, Vol. 3, pp. 1025–1032, 2003.

51. Spaan, M. T. and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs", *Journal of artificial intelligence research*, Vol. 24, pp. 195–220, 2005.

52. Smith, T. and R. Simmons, "Heuristic Search Value Iteration for POMDPs", *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pp. 520–527, AUAI Press, Arlington, Virginia, United States, 2004.

53. McCallum, R. A., "Instance-based utile distinctions for reinforcement learning with hidden state", *Machine Learning Proceedings 1995*, pp. 387–395, Elsevier, 1995.

54. Braziunas, D. and C. Boutilier, "Stochastic local search for POMDP controllers", *AAAI*, pp. 690–696, 2004.

55. Kurniawati, H., D. Hsu and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces", *In Proc. Robotics: Science and Systems*, 2008.

56. Pineau, J., G. Gordon and S. Thrun, *Tractable planning under uncertainty: exploiting structure*, Ph.D. Thesis, Carnegie Mellon University, the Robotics Institute, 2004.

57. Boutilier, C. and D. Poole, "Computing optimal policies for partially observable Markov decision processes using compact representations", *Proceedings of the National Conference on Artificial Intelligence*, pp. 1168–1175, Citeseer, 1996.

58. Poupart, P., *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*, Ph.D. Thesis, Toronto, Ont., Canada, Canada, 2005, aAINR02727.

59. Sim, H. S., K.-E. Kim, J. Kim, D.-S. Chang and M.-W. Koo, "Symbolic heuristic search value iteration for factored POMDPs", *23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference, AAAI-08/IAAI-08*, pp. 1088–1093, Association for the Advancement of Artificial Intelligence, 2008.

60. Cassandra, A., "Pomdp-solve [software, version 5.3]", , 2005.

61. Oliehoek, F. and M. Spaan, *MADP Toolbox 0.2*, Tech. rep., Technical Report, Informatics Institute, Amsterdam University, 2009.

62. Smith, T., "ZMDP software for POMDP and MDP planning", , 2007.

63. Du, Y., D. Hsu, X. Huang, H. Kurniawati, W. S. Lee, S. Ong and S. Png, "Approximate pomdp planning software (appl)", .

64. Egorov, M., Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta and M. J. Kochenderfer, "POMDPs. jl: A framework for sequential decision making under uncertainty", *The Journal of Machine Learning Research*, Vol. 18, No. 1, pp. 831–835, 2017.

65. Pineau, J. and G. J. Gordon, "POMDP planning for robust robot control", *Robotics Research*, pp. 69–82, Springer, 2007.

66. Roy, N., G. Gordon and S. Thrun, "Finding approximate POMDP solutions through belief compression", *Journal of artificial intelligence research*, Vol. 23, pp. 1–40, 2005.

67. Simmons, R. and S. Koenig, "Probabilistic robot navigation in partially observable environments", *IJCAI*, Vol. 95, pp. 1080–1087, 1995.

68. Theocharous, G. and S. Mahadevan, "Approximate planning with hierarchical partially observable Markov decision process models for robot navigation", *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, Vol. 2, pp. 1347–1352, IEEE, 2002.

69. Foka, A. and P. Trahanias, "Real-time hierarchical POMDPs for autonomous robot navigation", *Robotics and Autonomous Systems*, Vol. 55, No. 7, pp. 561–571, 2007.

70. Darrell, T. and A. Pentland, "Active gesture recognition using partially observable Markov decision processes", *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 3, pp. 984–988, IEEE, 1996.

71. Spaan, M. T., T. S. Veiga and P. U. Lima, "Active cooperative perception in network robot systems using POMDPs", *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4800–4805, IEEE, 2010.

72. Pineau, J., M. Montemerlo, M. Pollack, N. Roy and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results", *Robotics and autonomous systems*, Vol. 42, No. 3-4, pp. 271–281, 2003.

73. Boger, J., P. Poupart, J. Hoey, C. Boutilier, G. Fernie and A. Mihailidis, "A decision-theoretic approach to task assistance for persons with dementia", *IJCAI*,

pp. 1293–1299, Citeseer, 2005.

74. Hauskrecht, M. and H. Fraser, “Planning treatment of ischemic heart disease with partially observable Markov decision processes”, *Artificial Intelligence in Medicine*, Vol. 18, No. 3, pp. 221–244, 2000.

75. Maillart, L. M., J. S. Ivy, S. Ransom and K. Diehl, “Assessing dynamic breast cancer screening policies”, *Operations Research*, Vol. 56, No. 6, pp. 1411–1427, 2008.

76. Goulionis, J. E. and A. Vozikis, “Medical decision making for patients with Parkinson disease under average cost criterion”, *Australia and New Zealand health policy*, Vol. 6, No. 1, p. 15, 2009.

77. Mihailidis, A., J. N. Boger, T. Craig and J. Hoey, “The COACH prompting system to assist older adults with dementia through handwashing: An efficacy study”, *BMC geriatrics*, Vol. 8, No. 1, p. 28, 2008.

78. Treharne, J. T. and C. R. Sox, “Adaptive inventory control for nonstationary demand and partial information”, *Management Science*, Vol. 48, No. 5, pp. 607–624, 2002.

79. Aviv, Y. and A. Pazgal, “A partially observed Markov decision process for dynamic pricing”, *Management Science*, Vol. 51, No. 9, pp. 1400–1416, 2005.

80. Rusmevichientong, P. and B. Van Roy, “An analysis of belief propagation on the turbo decoding graph with Gaussian densities”, *IEEE Transactions on Information Theory*, Vol. 47, No. 2, pp. 745–765, 2001.

81. White, C. C., “Procedures for the solution of a finite-horizon, partially observed, semi-Markov optimization problem”, *Operations Research*, Vol. 24, No. 2, pp. 348–358, 1976.

82. Rosenfield, D., "Markovian deterioration with uncertain information", *Operations Research*, Vol. 24, No. 1, pp. 141–155, 1976.

83. Madanat, S., "Incorporating inspection decisions in pavement management", *Transportation Research Part B: Methodological*, Vol. 27, No. 6, pp. 425–438, 1993.

84. David, I., L. Friedman and Z. Sinuany-Stern, "A simple suboptimal algorithm for system maintenanceunder partial observability", *Annals of Operations Research*, Vol. 91, pp. 25–40, 1999.

85. Makis, V. and X. Jiang, "Optimal replacement under partial observations", *Mathematics of Operations Research*, Vol. 28, No. 2, pp. 382–394, 2003.

86. Papakonstantinou, K. G. and M. Shinozuka, "Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation", *Reliability Engineering & System Safety*, Vol. 130, pp. 214–224, 2014.

87. Madanat, S. and M. Ben-Akiva, "Optimal inspection and repair policies for infrastructure facilities", *Transportation science*, Vol. 28, No. 1, pp. 55–62, 1994.

88. Ellis, H., M. Jiang and R. B. Corotis, "Inspection, maintenance, and repair with partial observability", *Journal of Infrastructure Systems*, Vol. 1, No. 2, pp. 92–99, 1995.

89. Jiang, M., R. B. Corotis and J. H. Ellis, "Optimal life-cycle costing with partial observability", *Journal of infrastructure systems*, Vol. 6, No. 2, pp. 56–66, 2000.

90. Corotis, R. B., J. Hugh Ellis and M. Jiang, "Modeling of risk-based inspection, maintenance and life-cycle cost with partially observable Markov decision processes", *Structure and Infrastructure Engineering*, Vol. 1, No. 1, pp. 75–84, 2005.

91. Ivy, J. and H. Nembhard, "A Modeling Approach to Maintenance Decisions Using Statistical Quality Control and Optimization", *Quality and Reliability Engineering International*, Vol. 21, 06 2005.

92. Memarzadeh, M. and M. Pozzi, "Integrated Inspection Scheduling and Maintenance Planning for Infrastructure Systems", *Computer-Aided Civil and Infrastructure Engineering*, 10 2015.

93. AlDurgam, M. M. and S. O. Duffuaa, "Optimal joint maintenance and operation policies to maximise overall systems effectiveness", *International Journal of Production Research*, Vol. 51, No. 5, pp. 1319–1330, 2013.

94. Bui, T. H., J. Zwiers, M. Poel and A. Nijholt, "Affective dialogue management using factored POMDPs", *Interactive Collaborative Information Systems*, pp. 207–236, Springer, 2010.

95. Williams, J. D., P. Poupart and S. Young, "Factored partially observable Markov decision processes for dialogue management", *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 76–82, 2005.

96. Young, S., "Using POMDPs for dialog management", *2006 IEEE Spoken Language Technology Workshop*, pp. 8–13, IEEE, 2006.

97. Hoey, J., P. Poupart, C. Boutilier and A. Mihailidis, "POMDP models for assistive technology", *Assistive Technologies: Concepts, Methodologies, Tools, and Applications*, pp. 120–140, IGI Global, 2014.

98. Jean-Baptiste, E. M., P. Rotshtein and M. Russell, "POMDP based action planning and human error detection", *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 250–265, Springer, 2015.

99. Taha, T., J. V. Miró and G. Dissanayake, "A POMDP framework for modelling human interaction with assistive robots", *2011 IEEE International Conference*

on Robotics and Automation, pp. 544–549, IEEE, 2011.

100. Reyes, A., M. T. Spaan and L. E. Sucar, "An intelligent assistant for power plants based on factored MDPs", *2009 15th International Conference on Intelligent System Applications to Power Systems*, pp. 1–6, IEEE, 2009.

101. Papakonstantinou, K. and M. Shinozuka, "Optimum inspection and maintenance policies for corroded structures using partially observable Markov decision processes and stochastic, physically based models", *Probabilistic Engineering Mechanics*, Vol. 37, pp. 93–108, 2014.

102. Lin, S., A. Zhang and D. Feng, "Maintenance decision-making model based on POMDP for traction power supply equipment and its application", *Prognostics and System Health Management Conference (PHM-Chengdu), 2016*, pp. 1–6, IEEE, 2016.

103. Memarzadeh, M., M. Pozzi and J. Zico Kolter, "Optimal planning and learning in uncertain environments for the management of wind farms", *Journal of Computing in Civil Engineering*, Vol. 29, No. 5, p. 04014076, 2014.

104. Memarzadeh, M. and M. Pozzi, "Integrated inspection scheduling and maintenance planning for infrastructure systems", *Computer-Aided Civil and Infrastructure Engineering*, Vol. 31, No. 6, pp. 403–415, 2016.

105. Memarzadeh, M. and K. Papakonstantinou, "Optimal maintenance and inspection planning for structural components under mixed and partial observability", , 01 2017.

106. Ghandali, R., M. H. Abooie and M. Fallahnezhad, "A POMDP framework to find Optimal Policy in Sustainable Maintenance", *Scientia Iranica*, 2018.

107. Morato, P. G., Q. A. Mai, P. Rigo and J. S. Nielsen, "Point-based POMDP risk based inspection of offshore wind substructures", *Proceedings of Ialcce 2018: the*

Sixth International Symposium on Life-cycle Civil Engineering, 2018.

108. Papakonstantinou, K., C. Andriotis and M. Shinozuka, "Point-based POMDP solvers for lifecycle cost minimization of deteriorating structures", *Proceedings of the 5th International Symposium on Life-Cycle Civil Engineering (IALCCE 2016)*, p. 427, 2016.

109. Papakonstantinou, K. G., C. P. Andriotis and M. Shinozuka, "POMDP solutions for monitored structures", *Proc. of IFIP WG-7.5 Conference on Reliability and Optimization of Structural Systems*, 2016.

110. Papakonstantinou, K. G., C. P. Andriotis and M. Shinozuka, "POMDP and MOMDP solutions for structural life-cycle cost minimization under partial and mixed observability", *Structure and Infrastructure Engineering*, Vol. 14, No. 7, pp. 869–882, 2018.

111. Kim, D., H. S. Sim, K.-E. Kim, J. Kim, H. Kim and J. W. Sung, "Effects of user modeling on POMDP-based dialogue systems", *9th Annual Conference of the International Speech Communication Association, 2008*, pp. 1169–1172, Interspeech, 2008.

112. Müller, F., C. Späth, T. Geier and S. Biundo, "Exploiting expert knowledge in factored POMDPs", *Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 606–611, IOS Press, 2012.

113. Hoey, J., A. Von Bertoldi, P. Poupart and A. Mihailidis, "Assisting persons with dementia during handwashing using a partially observable Markov decision process", *Proc. Int. Conf. on Vision Systems*, Vol. 65, p. 66, 2007.

114. Araya-López, M., V. Thomas, O. Buffet and F. Charpillet, "A closer look at MOMDPs", *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, Vol. 2, pp. 197–204, IEEE, 2010.

115. Nicol, S., O. Buffet, T. Iwamura and I. Chadès, "Adaptive management of migratory birds under sea level rise", *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

116. Rout, T. M., J. L. Moore and M. A. McCarthy, "Prevent, search or destroy? A partially observable model for invasive species management", *Journal of applied ecology*, Vol. 51, No. 3, pp. 804–813, 2014.

117. Erdogdu, U., R. Alhajj and F. Polat, "The benefit of decomposing POMDP for control of gene regulatory networks", *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*, pp. 381–385, IEEE Computer Society, 2011.

118. Capitán, J., L. Merino and A. Ollero, "Multi-robot coordinated decision making under mixed observability through decentralized data fusion", *Proceedings of the 11th International Conference on Mobile Robots and Competitions (Robotica 2011)*, 2011.

119. Ibrahim, R., B. Kucukyazici, V. Verter, M. Gendreau and M. Blostein, "Designing personalized treatment: An application to anticoagulation therapy", *Production and Operations Management*, Vol. 25, No. 5, pp. 902–918, 2016.

120. Littman, M. L., *Algorithms for sequential decision making*, Brown University Providence, RI, 1996.

121. Bellman, R., "Dynamic programming", *Science*, Vol. 153, No. 3731, pp. 34–37, 1966.

122. Bellman, R. *et al.*, "The theory of dynamic programming", *Bulletin of the American Mathematical Society*, Vol. 60, No. 6, pp. 503–515, 1954.

123. Hansen, E. A., "Solving POMDPs by searching in policy space", *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 211–219,

Morgan Kaufmann Publishers Inc., 1998.

124. Madani, O., S. Hanks and A. Condon, "On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision processes", *AAAI/IAAI*, pp. 541–548, 1999.

125. Ross, S., J. Pineau, S. Paquet and B. Chaib-Draa, "Online planning algorithms for POMDP", *Journal of Artificial Intelligence Research*, Vol. 32, pp. 663–704, 2008.

126. Zhang, J. and Z. Liu, "Calculating some inverse linear programming problems", *Journal of Computational and Applied Mathematics*, Vol. 72, No. 2, pp. 261–273, 1996.

127. Cassandra, A. R., L. P. Kaelbling and M. L. Littman, "Acting optimally in partially observable stochastic domains", *AAAI*, Vol. 94, pp. 1023–1028, 1994.

128. Lusena, C., J. Goldsmith and M. Mundhenk, "Nonapproximability Results for Partially Observable Markov Decision Processes", *J. Artif. Intell. Res.*, Vol. 14, pp. 83–103, 2001.

129. Littman, M. L., A. R. Cassandra and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up", *Machine Learning Proceedings 1995*, pp. 362–370, Elsevier, 1995.

130. Shani, G., J. Pineau and R. Kaplow, "A survey of point-based POMDP solvers", *Autonomous Agents and Multi-Agent Systems*, Vol. 27, No. 1, pp. 1–51, 2013.

131. Dibangoye, J. S., B. Chaib-draa and A.-i. Mouaddib, "Topological orders based planning for solving POMDPs", *In AAAI*, 2008.

132. Bryant, R. E., "Symbolic Boolean manipulation with ordered binary-decision diagrams", *ACM Computing Surveys (CSUR)*, Vol. 24, No. 3, pp. 293–318, 1992.

133. Koltunova, V., *Active Sensing for Partially Observable Markov Decision Processes*, Master's Thesis, University of Waterloo, 2013.

134. Hoey, J., "Home", `https://cs.uwaterloo.ca/ jhoey`.

135. Zhang, N. L. and W. Liu, *Planning in stochastic domains: Problem characteristics and approximation*, Tech. rep., Technical Report HKUST-CS96-31, Hong Kong University of Science and Technology, 1996.

136. Murphy, K. *et al.*, "The bayes net toolbox for matlab", *Computing science and statistics*, Vol. 33, No. 2, pp. 1024–1034, 2001.

137. Games, P. A. and J. F. Howell, "Pairwise multiple comparison procedures with unequal n's and/or variances: a Monte Carlo study", *Journal of Educational Statistics*, Vol. 1, No. 2, pp. 113–125, 1976.

138. Özgür-Ünlüakın, D., İ. Kıvanç, B. Türkali and Ç. Aksezer, "A DBN Based Prognosis Model for a Complex Dynamic System: A Case Study in a Thermal Power Plant", *The International Symposium for Production Research*, pp. 75–84, Springer, 2018.

139. Özgür-Ünlüakın, D., B. Türkali, A. Karacaörenli and S. Ç. Aksezer, "A DBN based reactive maintenance model for a complex system in thermal power plants", *Reliability Engineering & System Safety*, Vol. 190, p. 106505, 2019.

140. "Regenerative Air Pre-Heater Regenerative Air Pre-Heater", `https://www.lnthowden.com/product-services/regenerative-air-pre-heater`.

APPENDIX A: SYMBOLIC PERSEUS INPUT FORMAT

```
(variables
        (System fail work)
        (C2 fail work)
        (C1 fail work))

(observations  (Obs green red))

init [*  (C1     (fail    (0.0))  (work(1.0)))
         (C2     (fail    (0.0))  (work(1.0)))
         (C1     (fail            (C2     (fail          (System (fail  (1.0))  (work  (0.0))))
                                          (work          (System (fail  (1.0))  (work  (0.0))))))
                 (work            (C2     (fail          (System (fail  (1.0))  (work  (0.0))))
                                          (work          (System (fail  (0.0))  (work  (1.0)))))))]

dd defaultC1
         (C1     (fail            (C1'    (fail    (0.85)) (work   (0.15))))
                 (work            (C1'    (fail    (0.00)) (work   (1.00)))))
enddd

dd defaultC2
         (C2     (fail            (C2'    (fail    (0.85))  (work   (0.15))))
                 (work            (C2'    (fail    (0.00))  (work   (1.00)))))
endd

dd System
(C1'     (fail            (C2'    (fail            (System' (fail(1.00))(work(0.00))))
                                  (work            (System' (fail(1.00))(work(0.00)))))
         (work            (C2'    (fail            (System' (fail(1.00))(work(0.00))))
                                  (work            (System' (fail(0.00))(work(1.00))))))
enddd

dd ObsCPT
         (System'         (fail    (Obs' (green (0.92))(red (0.08))))
                          (work (Obs' (green (0.02))(red (0.98)))))
enddd

action donothing
         C1               (defaultC1)
         C2               (defaultC2)
         System           (System)
         observe
                 Obs(ObsCPT)
         endobserve
 cost (System             (fail    (7500))
                          (work  (0)))
endaction

action replaceC1
         C1               (C1     (fail            (C1'(fail(0.0))(work (1.0))))
                                  (work            (C1'(fail(0.0))(work (1.0)))))
         C2               (defaultC2)
         System           (System)
```

```
        observe
                Obs(ObsCPT)
        endobserve
        cost (System            (fail     (7700))
                                (work   (2600)))
endaction

action replaceC2
        C1              (defaultC1)
        C2              (C2     (fail           (C2'(fail(0.0))(work (1.0))))
                                (work           (C2'(fail(0.0))(work (1.0)))))
        System          (System)
        observe
                Obs(ObsCPT)
        endobserve
        cost (System            (fail     (7900))
                                (work   (2700)))
endaction

discount 0.999
```

APPENDIX B: POMDP-SOLVE INPUT FORMAT

```
discount: 0.999
values: reward
states: ww wf fw ff
actions: dn replaceC1 replaceC2
observations: green red

T:dn
0.7225 0.1275 0.1275 0.0225
0      0.85   0      0.15
0      0.85   0      0.15
0      0      0      1


T: replaceC1
0.85   0.15   0      0
0      1      0      0
0.85   0.15   0      0
0      1      0      0

T: replaceC2
0.85   0      0.15   0
0.85   0      0.15   0
0      0      1      0
0      0      1      0


O: * : ww : green 0.92
O: * : wf : green 0.02
O: * : fw : green 0.02
O: * : ff : green 0.02
O: * : ww : red 0.08
O: * : wf : red 0.98
O: * : fw : red 0.98
O: * : ff : red 0.98

R: dn : * : * : green 0
R: dn : * : * : red -7500
R: replaceC1 : * : * : green -2600
R: replaceC1 : * : * : red -7700
R: replaceC2 : * : * : green -2700
R: replaceC2 : * : * : red -7900
```

APPENDIX C: TRANSITIONAL AND CONDITIONAL PROBABILITIES OF EXPERIMENTAL MODEL

Table C.1: Probabilities of C1 and C3.

Action	**Major**				**Minor**				**Do nothing**			
Self [t-1]	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**
W	1	1	1	1	1	0.5	0	0	0.95	0	0	0
Deg1	0	0	0	0	0	0.5	0.5	0	0.04	0.90	0	0
Deg2	0	0	0	0	0	0	0.5	0	0	0.07	0.85	0
F	0	0	0	0	0	0	0	1	0.01	0.03	0.15	1

Table C.2: Probabilities of C2 and C4.

Action	**Major**				**Minor**				**Do nothing**			
Self [t-1]	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**
W	1	1	1	1	1	0.5	0	0	0.97	0	0	0
Deg1	0	0	0	0	0	0.5	0.5	0	0.02	0.92	0	0
Deg2	0	0	0	0	0	0	0.5	0	0	0.06	0.87	0
F	0	0	0	0	0	0	0	1	0.01	0.02	0.13	1

Table C.1: Probabilities of P1 and P2.

C2	**W**				**Deg1**				**Deg2**				**F**			
C1	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**	**W**	**Deg1**	**Deg2**	**F**
W	1	0.7	0.5	0	0.8	0.5	0.3	0	0.6	0.4	0.1	0	0	0	0	0
F	0	0.3	0.5	1	0.2	0.5	0.7	1	0.4	0.6	0.9	1	1	1	1	1

Table C.2: Probabilities of P3.

P2	**W**		**F**	
P1	**W**	**F**	**W**	**F**
W	1	0	0	0
F	0	1	1	1

Table C.3: Probabilities of O1.

P3	**W**	**F**
Green	0.96	0.005
Yellow	0.03	0.015
Red	0.01	0.98

APPENDIX D: TRANSITIONAL AND CONDITIONAL PROBABILITIES OF RAH SYSTEM

Table D.1: Transition probabilities of BB.

BB Main.	Replace			Do Nothing		
Self [t-1]	Normal	Loose	Locked	Normal	Loose	Locked
Normal	1	1	1	0.997503	0	0
Loose	0	0	0	0.001665	1	0
Locked	0	0	0	0.000832	0	1

Table D.2: Transition probabilities of WI.

WI Main.	Replace						Do Nothing					
BB	Normal		Loose		Locked		Normal		Loose		Locked	
Self [t-1]	Normal	Locked	Normal	Locked	Normal	Locked	Normal	Locked	Normal	Locked	Original	Burned
Original	1	1	1	1	1	1	0.993356	0	0.2	0	1	0
Burned	0	0	0	0	0	0	0.006644	1	0.8	1	0	1

Table D.3: Transition probabilities of RS.

RS Main.	Replace						Do Nothing					
BB	Normal		Loose		Locked		Normal		Loose		Locked	
Self [t-1]	Normal	Unaligned	Normal	Unaligned	Normal	Unaligned	Normal	Unaligned	Normal	Unaligned	Normal	Unaligned
Normal	1	1	1	1	1	1	0.999667	0	0.02	0	1	0
Unaligned	0	0	0	0	0	0	0.000333	1	0.98	1	0	1

Table D.4: Conditional probabilities of RR.

RS	Normal						Unaligned					
WI	Original			Burned			Original			Burned		
BB	Normal	Loose	Locked	Normal	Loose	Locked	Normal	Loose	Locked	Normal	Loose	Locked
Rotate	1	0.5	0	0	0	0	0.3	0.1	0	0	0	0
Not Rotate	0	0.5	1	1	1	1	0.7	0.9	1	1	1	1

Table D.5: Conditional probabilities of HRG.

HRG Main.	Replace		Do Nothing	
Self [t-1]	**Normal**	**Fail**	**Normal**	**Fail**
Normal	1	1	0.998335	0
Fail	0	0	0.001665	1

Table D.6: Conditional probabilities of HRG Rot.

HRG	Normal		Fail	
RR	**Rotate**	**Not Rotate**	**Rotate**	**Not Rotate**
Work	1	0	0	0
Fail	0	1	1	1

Table D.7: Conditional probabilities of RAH Rotation.

HRG Rot.	Work	Fail
Rotate	1	0
Not Rotate	0	1

Table D.8: Transition probabilities of RI.

RI Main.	Replace						Do Nothing					
HC Main.	Clean			Do Nothing			Clean			Do Nothing		
Self [t-1]	fullInt	mediumInt	lowInt	fullInt	mediumInt	lowInt	fullInt	mediumInt	lowInt	fullInt	mediumInt	lowInt
fullInt	1	1	1	1	1	1	1	1	1	0.997669	0	0
mediumInt	0	0	0	0	0	0	0	0	0	0.001665	0.996672	0
lowInt	0	0	0	0	0	0	0	0	0	0.000666	0.003328	1

Table D.9: Conditional probabilities of Slagging.

Coal Rank	Good	Bad
No	0.95	0.6
Yes	0.05	0.4

Table D.10: Transition probabilities of HC.

HC Main.	**Replace**				**Do Nothing**			
Self [t-1]	**Cleaned**		**Dirty**		**Cleaned**		**Dirty**	
Slagging [t-1]	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**	**No**	**Yes**
Cleaned	1	1	1	1	0.999334	0.996672	0	0
Dirty	0	0	0	0	0.000066	0.003328	1	1

Table D.11: Conditional probabilities of RAH Exit Temp.

RI	**fullInt**				**mediumInt**				**lowInt**			
HC	**Cleaned**		**Dirty**		**Cleaned**		**Dirty**		**Cleaned**		**Dirty**	
RAH Rot.	**R**	**NR**	**R**	**NR**	**R**	**NR**	**R**	**NR**	**R**	**NR**	**R**	**NR**
Normal	1	0	0.95	0	0.99920	0	0.92	0	0.00019	0	0.0014	0
Low	0	0	0.04	0	0.00035	0	0.05	0	0.68	0	0.69	0
VLow	0	1	0.01	1	0.0045	1	0.03	1	0.31981	1	0.3086	1

Table D.12: Observational probabilities of RAH Measured Temp.

RAH Exit Temp.	**Normal**	**Low**	**VLow**
No	0.99989	0.00015	0.00005
Low	0.0001	0.99	0.0099
VLow	0.00001	0.00985	0.99

Table D.13: Observational probabilities of Vibration.

Rotor Shaft	**Normal**	**Unaligned**
N/A	0.99989	0.00015
Low	0.0001	0.00985
VLow	0.00001	0.99