OPTIMAL REPLACEMENT TIME OF COMPONENTS FOR DETERIORATING SYSTEMS

by

Elvan Durdu

B.S., Industrial Engineering, Boğaziçi University, 2022

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
2025

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my thesis advisor, Assoc. Prof. Aybek Korugan, for his unwavering guidance, support, and encouragement throughout this academic journey. His profound expertise and insightful feedback have been instrumental in shaping the direction of my research, and I have gained invaluable knowledge and skills under his mentorship. The lessons I have learned from him extend far beyond academics and will continue to serve as a source of inspiration throughout my life. I consider myself incredibly fortunate to have had the privilege of working under his guidance during this pivotal chapter of my academic career.

I would also like to extend my heartfelt thanks to Prof. Refik Güllü and Prof. Taner Bilgiç for their continuous support and willingness to help whenever I knocked on their office doors. Their knowledge and willingness to help has been invaluable throughout this process. I am also deeply grateful to Assoc. Prof. Wolfgang Hoermann for taking the time to provide valuable feedback, which has significantly enriched my research. Additionally, I would like to express my sincere gratitude to Prof. Fikri Karaesmen and Prof. Murat Fadıloğlu for their constructive feedback and contributions, which have been instrumental in refining this work.

I am very proud to be part of Industrial Engineering Department as a master student and graduate assistant. These three years have profoundly transformed my life, providing opportunities for personal and professional growth beyond what I could have imagined. I would especially like to thank Ezgi and Özge, with whom I faced the similar academic challenges and who were always generous in sharing their experiences with me. Their presence has been invaluable to me. Additionally, I wish to express my sincere appreciation to my colleagues Tuğçe, Zeynep, Büşra, Burak, Abdullah, Ozan and Halis. Their support and companion have made this process much more manageable and meaningful.

I am grateful for the support I received through the TÜBİTAK Science Human Support Programs Presidency (BİDEB) 2211-Domestic Graduate Scholarship Program during my graduate education.

Finally, I would like to express my utmost gratitude to my family, who have been my greatest supporters throughout this journey. I am incredibly fortunate to have my mother, Gülden, my father, Mustafa, and my sister, Nehir, by my side, even from afar. Despite the physical distance, your unwavering love and encouragement have always been a source of strength for me. Thank you for being there for me and for patiently listening to my countless worries during our many calls. I am also profoundly grateful to my grandfather, Ahmet, for his steadfast faith in me and his invaluable help along the way. Having you all in my life has been my greatest blessing, and I could not have achieved this milestone without your support. Lastly, I would like to extend my heartfelt thanks to my dear friend Emine. Your encouragement, understanding, and friendship have meant so much to me, and I am truly grateful for your support throughout this journey.

ABSTRACT

OPTIMAL REPLACEMENT TIME OF COMPONENTS FOR DETERIORATING SYSTEMS

Optimal replacement time problems involve determining the best time to undertake a replacement action within a stochastic system to either minimize associated cost or maximize related rewards of the process, based on limited information. This problem can be encountered in many different application areas such as healthcare, machine maintenance, finance, manufacturing etc. Identifying the appropriate intervention time is crucial for preventing system degradation that may ultimately lead to failure. This study investigates an optimal stopping problem focused on determining the optimal point to replace a component which belongs to a stochastically degrading system. The system's functionality is maintained by replacing deteriorated components with heterogeneous spares of varying quality. The decision framework includes two actions: replace and wait. The objective is to identify the optimal switching point between these decisions to maximize the system's expected reward. The problem is formulated as a Markov Decision Process (MDP) with an average reward criterion. The structural properties of the optimal value function are analyzed to demonstrate the existence of an optimal threshold-type policy. Subsequently, this policy is derived by exploiting the Markovian dynamics of the system. Two alternative methods are proposed to calculate the optimal threshold point in addition to the Value Iteration algorithm and they are implemented in Python to compute optimal solutions across different input parameter settings, illustrating the model's adaptability and robustness. Numerical results are analyzed to assess the sensitivity of system parameters. Additionally, extensions related to system states and quality of spare parts are also included to the model to see their effect on system behavior. The findings offer a comprehensive framework for addressing optimal replacement problems.

ÖZET

BOZULAN SİSTEMLERDE BİLEŞENLERİN OPTİMAL DEĞİŞİM ZAMANI

Optimal değişim zamanı problemleri stokastik bir sistemde değişim kararını almak için en uygun zamanı belirlemeyi içerir. Bu işlem, sınırlı bilgiye dayanarak süreçle ilişkili maliyetleri minimize etmek veya ödülleri maksimize etmek amacı taşır. Sağlık, makine bakımı, finans, üretim vb. birçok farklı alanda bu problemle karşılaşılabilir. Doğru müdahale zamanına karar verilmesi en sonunda arızaya yol açabilecek sistem bozulmalarını önlemek için kritik öneme sahiptir. Bu çalışma, stokastik olarak bozulan bir sisteme ait bir bileşenin değiştirilmesi için en uygun noktayı belirlemeye odaklanan bir optimal durdurma problemini ele almaktadır. Sistem fonksiyonunu sürdürebilmek için bozulan bileşen değişen kalitede heterojen yedek parçalar kullanılarak değiştirilir.Karar çerçevesi iki eylem içermektedir: değiştir ve bekle. Amaç, sistemin beklenen ödülünü maksimize etmek için bu kararlar arasındaki optimal geçiş noktasını belirlemektir. Problem, ortalama ödül kriterine sahip bir Markov Karar Süreci (MKS) olarak modellenmiştir. Optimal değer fonksiyonlarının yapısal özellikleri kullanılarak optimal eşik tipi politikanın varlığı gösterilmiştir. Daha sonra, sistemin Markov yapısından yararlanılarak optimal eşik politikası türetilmiştir. Optimal eşik değerinin hesaplamak için Değer İterasyonu algositmasına ek olarak iki alternatif yöntem sunulmuştur ve bu metodlar değişen girdi parametreleri altında optimal çözümleri hesaplamak için Python'da uygulanmıştır. Bu, modelin uyarlanabilirliğini ve dayanıklılığını ortaya koymaktadır. Numerik sonuçlar, sistem parametrelerinin duyarlılığını değerlendirmek için analiz edilmiştir. Ayrıca, sistem durumları ve yedek parçaların kalitesiyle ilgili genişletmeler de modele dahil edilerek, bunların sistem davranışı üzerindeki etkileri incelenmiştir. Bulgular, optimal değişim problemlerine yönelik kapsamlı bir çerçeve sunmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Reward coefficient related to transition to state (2,0)
b	Cost coefficient for waiting action
c	Reward of transition to state $(1, 0)$
d	Binary decision that denotes selected action
e	Reward coefficient related to transition to state (3,0) for extension 2
f	Non-decreasing function
g	Optimal average expected return per unit time
h	System condition level
k	Number of spare parts
k^*	Optimal threshold point
l	Bounded function
n	Iteration step of the algorithm
p	Type 1 component probability
p_1	Type 1 component probability for extension 1
p_2	Type 2 component probability for extension 1
P	Probability transition matrix of finite state Markov Chain
q^*	Optimal action
q	Any possible action
q_{ij}	Transition probability matrix of a Markov Chain
Q	State transition probability matrix of base model
Q'	State transition probability matrix of extension 1 case 1
Q''	State transition probability matrix of extension 1 case 2
Q'''	State transition probability matrix of extension 2
r	Expected reward of the system
R	Replace action
S	State space
V_n	Value function at iteration n

w	Reward function
W	Wait action
γ	System deterioration rate
λ	Arrival rate of spare parts
$\mathit{\Lambda}$	Total rate of the system
δ	Difference of value function between iteration n and n-1
$\mathit{\Delta} V$	Difference function
μ	Policy
μ^*	Optimal policy
π	Steady state distribution of finite state Markov Chain
$\mathit{\Pi}$	Steady state distribution matrix of finite state Markov Chain
ϕ	Average expected return per unit time

LIST OF ACRONYMS/ABBREVIATIONS

CTMC	Continuos Time Markov Chain
IFR	Increasing Failure Rate
LP	Linear Programming
MC	Markov Chain
MDP	Markov Decision Process
MOMDP	Mixed Observable Markov Decision Process
POMDP	Partially Observable Markov Decision Process

1. INTRODUCTION

Deterioration is an inevitable process that impacts both natural and man-made complex systems, often leading to reduced functionality and eventual system failure over time. To mitigate these adverse effects and ensure the sustainability and efficiency of system operations, it becomes essential to replace degraded components. The process of determining the appropriate time for such replacements is a critical decision, as it directly influences the long-term performance and cost-effectiveness of the system. Optimal component replacement can be conceptualized as a specific case of optimal stopping time problems, which involve identifying the most advantageous moment to take a particular action within a stochastic framework. These problems aim to either minimize the associated costs or maximize the potential rewards of the process, often under conditions of uncertainty and limited information. Optimal stopping problems arise in a wide range of application areas, including but not limited to healthcare, maintenance, finance, and manufacturing. Determining the right time to intervene is crucial for preserving the functionality and efficiency of real-world systems. Failure to do so can result in continued degradation, ultimately leading to undesirable outcomes such as system breakdown or in-operability.

Maintenance is one of the primary areas which optimal replacement problem frequently arises. Optimal replacement time problem corresponds to determining ideal time for maintenance actions. Since the replacement decision requires the utilization of spare parts, they are critical components to ensure effective maintenance operations in many industries. The use of refurbished or re-manufactured spare parts, alongside new components, is a common practice in the maintenance of large systems across various industries. Although using refurbished spare parts can be cost-effective and sustainable solution, their lifespan can be shorter than the new ones, reflecting their relatively lower quality. In certain scenarios, precise information about the nature of available spare parts, whether re-manufactured or new, is not readily accessible. Instead, the type of spare parts can often only be estimated using probabilistic methods.

In such cases, probability distribution of spare part types can be estimated. When spare parts are not identical, the system would not be in the same functioning status after the maintenance procedure. Consequently, the system's post-maintenance condition also remains uncertain and can only be described probabilistically. Maintenance problems can generally be categorized into two main types: preventive maintenance and corrective maintenance. This thesis focuses on the optimal replacement time of components in the context of preventive maintenance, leveraging a recurrent model structure. Similar scenarios are prevalent across various industries, including manufacturing, aviation, energy, transportation, and healthcare. These sectors frequently employ re-manufactured spare parts due to the high cost of manufacturing new components, making re-manufacturing a preferred alternative to disposal.

The optimal replacement time problem with spare parts of varying and uncertain quality is a significant concern in renewable energy sector. In wind energy systems, for instance, the gearbox of a wind turbine is a critical component prone to wear and deterioration over time. The gearbox plays a vital role by converting the low rotational speed of the turbine blades into a higher speed suitable for electricity generation, making it highly susceptible to degradation. It is also demonstrated that the gearbox is associated with the most critical failures in wind turbines, primarily due to its high capital cost [1]. When a gearbox exhibits signs of wear, such as pitting on its gears, it must be replaced with a spare part to ensure continued operation. Operators typically have three spare part options: new, re-manufactured, and reconditioned gearboxes. New gearboxes are factory-produced, offering the highest quality and reliability but also being the most expensive option. Re-manufactured gearboxes are previously used units that have been repaired and restored; their quality can vary significantly depending on the extent of the re-manufacturing process and testing. Reconditioned gearboxes, on the other hand, are used units subjected to minor repairs without full restoration, resulting in lower quality compared to re-manufactured options. The inherent variability in the quality of spare parts, especially re-manufactured and reconditioned gearboxes, introduces uncertainty into maintenance decisions. These spare parts may contain hidden defects or exhibit performance variability due to differences in their repair processes or prior usage. Consequently, the actual quality of a spare part often remains uncertain both before and after replacement. Therefore, the proposed model formulation in this thesis is well-suited for addressing real-world maintenance scenarios, such as those observed in wind turbine systems.

In addition to maintenance, optimal stopping time problems are frequently encountered in the healthcare domain. It is mostly critical to determine best intervention time in many healthcare operations research problems. The level of system improvement is uncertain in many healthcare problems. One such example is organ transplantation, a complex healthcare challenge that has inspired the topic and model formulation of this thesis. The deterioration or failure of vital organs, such as the kidney or liver, necessitates transplantation for patients at the end-stage of organ failure. Transplantation can be performed using either cadaveric organs or organs from living donors. At each decision point, the decision-maker faces the choice of accepting a transplant or waiting for potentially better options. Consequently, the transplantation process can be effectively framed as an optimal replacement time problem for organs. This decision-making process is inherently complex, as it involves multiple interrelated factors, including health-specific determinants, that must be considered to develop a realistic model by utilizing some assumptions for simplification. Moreover, the problem bears a resemblance to maintenance scenarios, where there is an analogy between spare parts in maintenance systems and living donors in transplantation. Furthermore, in cases of living donor transplantation, the possibility of barter exchanges between donors adds another layer of complexity, making such problems more suitable for modeling as optimal stopping time problems. These models often yield optimal control-limit policies that guide decision-makers in selecting the best course of action under uncertainty.

An optimal replacement time problem is modeled in this paper by using Markov Decision Process tools. The aim is to determine optimal control policies for replacement decision by analyzing the interplay between various parameters of a deteriorating complex system. The problem is further extended with alternative formulations and additional enhancements to the base model. Special attention is given to optimal threshold-type policies that guide the decision-maker in choosing between replacement and waiting actions.

This study contributes to the literature on optimal stopping and replacement time problems by constructing an MDP model and employing dynamic programming approaches to analyze the mutual interactions between system parameters and optimal control policies. It is aimed to find a threshold point which effectively balances the trade-off between the rewards of replacement, accounting for the uncertainty in spare part quality, and the costs associated with waiting.

The remainder of the paper is organized as follows. Chapter 2 offers a comprehensive review of the existing literature related to the optimal replacement time problem, with a focus on applications in maintenance and healthcare domains. Chapter 3 describes the optimal replacement problem, formulates the corresponding mathematical models, and presents the main analytical results. Chapter 4 provides a numerical analysis of the system and the derived optimal policies, supporting the findings obtained in Chapter 3. Finally, Chapter 5 concludes the paper with a summary of key insights and concluding remarks.

2. LITERATURE REVIEW

Since the early 1950s, stochastic models have played a key role in the operations research and applied probability literature, focusing on the deterioration of both repairable and non-repairable systems. Our study focuses on the deterioration of systems with replaceable or repairable components which is a widely studied topic in operations research. Such systems are prevalent in various real-life application areas, ranging from manufacturing and maintenance to healthcare operations and management, which underscores their practical significance and the sustained interest of researchers. These systems are often modeled as optimal stopping or replacement time problems, where the objective is to determine the best time to take action to minimize costs or maximize system performance over time.

A rich body of literature explores stochastic models for addressing optimal replacement problems. These models account for the inherent uncertainties in system performance and deterioration, making them suitable for real-world applications. A variety of methodologies have been employed to solve these problems, with dynamic programming and Markov decision processes emerging as particularly prevalent and effective tools. These approaches provide a structured framework for modeling decision-making under uncertainty, allowing for the systematic evaluation of trade-offs between costs and rewards over time.

Optimal replacement problems have been extensively studied in both the maintenance and healthcare operations literature. In the field of maintenance, these problems predominantly focus on determining the optimal time for replacing critical components to minimize system downtime, reduce costs, and maintain operational efficiency. In healthcare operations, optimal replacement problems are similarly critical but take the form of decision-making regarding the time of medical interventions. Examples include the optimal time of organ transplantation to maximize patient outcomes and survival rates, as well as determining the initiation of treatments for chronic diseases to balance effectiveness and side effects.

First, maintenance models are reviewed. Stochastic models that describe the deterioration of repairable systems have a long-standing history in maintenance decision-making problems. İçten et al. [2] addressed a maintenance optimization problem involving the scheduling of a limited number of identical replacements for a critical component in a system, where the component's failure results in system failure. The component's degradation is modeled as a discrete-time Markov chain, and the objective is to determine an optimal replacement policy that maximizes the total expected lifetime of the system, given a finite supply of spare parts. To achieve this, they utilized Markov Decision Process (MDP) and sample path analysis methodologies. A key distinction between their model and ours lies in the treatment of spare parts. While İçten et al. assume a finite and predefined number of spare parts, our model introduces new components that stochastically enter the system with a homogeneous arrival rate. Despite this difference, there is a structural similarity between the two models: both represent the system's states as two-dimensional variables—one dimension representing the condition of the component and the other tracking the number of spare parts. Another critical difference is in the reward and cost structure. In İçten et al.'s model, the reward is constant and identical for the two possible actions, replace and wait, and they do not incorporate costs. In contrast, our model employs state- and action-dependent rewards, which include negative rewards interpreted as costs that are monotone. Additionally, our model operates under the assumption that the system does not have a failure state, allowing it to repeat the process infinitely many times and eventually converge to a steady-state distribution. İçten et al. further examined other performance measures, including the expected number of spare parts used until system failure and the distribution of the system's lifetime, providing additional insights into the dynamics of systems with limited resources.

There are some studies in the literature which consider the effect of another external factor on the deterioration process which is not considered in our study. Kurt and Kharoufeh [3] explored the optimal replacement strategy for a Markovian deteriorating system operating within a controllable environment. Their work is formulated as a discrete-time infinite-horizon discounted Markov Decision Process (MDP), where the system is periodically inspected and decisions are made either to replace the system instantaneously or to allow it to continue operating until the next inspection period. The primary objective of their study is to determine the optimal replacement time to minimize the total expected discounted cost over the planning horizon. A notable distinction between their model and ours is the absence of a reward component in their framework; their analysis focuses solely on minimizing costs associated with actions. Additionally, the deterioration of the system is influenced by an external stochastic process that governs the evolution of the environment, making the deterioration process dependent on both the system's status and the environment. In contrast, our model assumes an invariant environment, which does not take into account the interaction between the system's deterioration process and its surroundings. Kurt et al. also provided sufficient conditions for the existence of an optimal control-limit replacement policy, a common structure in MDPs for maintenance and replacement problems. Furthermore, they illustrated the characteristics of the optimal policy through numerical examples, showcasing its control-limit structure and applicability. This study contributes valuable insights into modeling and solving replacement problems with stochastic deterioration, highlighting the role of environmental dynamics in the decision-making process.

Similar maintenance problem is studied by Ulukus et al. [4] where the system degradation depends on an stochastic environment process. They analyze an MDP model aimed at minimizing the total expected discounted cost and demonstrate the existence of an optimal threshold-type replacement policy for each environmental state. Their objective is to determine the optimal time for preventive maintenance. However, their study does not account for the number and quality of spare parts used in replacements. Additionally, they discretize their CTMC model, similar to the approach we adopted in our analysis.

Another deteriorating system is analyzed by Cekyay and Ozekici [5] where the system performs missions, and its dynamics are formulated as a finite-state Markov decision process. In their study, the system's deterioration follows a Markov chain that is modulated by the mission process, distinguishing it from our approach. Specifically, their model incorporates a Markov process for the mission dynamics that directly influences the deterioration process. The authors first address an optimal repair problem and subsequently examine an optimal replacement problem as a subset of the broader repair framework, a significant departure from our study's focus. They demonstrate that the optimal replacement policy follows a control-limit structure. Furthermore, their analysis includes various repair cost models, aiming to minimize the system's total discounted cost. They also explore the monotonicity properties of the Markov chain and relax several assumptions about this behavior, contributing valuable insights to the literature.

A condition-based optimal maintenance problem in a partially observable multi-component system is studied by Karabağ et al [6]. Unlike our model, which focuses on a single-component system, their study addresses a multi-component system which additionally includes the relationships between components. They define two stochastic processes which show levels of degradation and partial observation of sensor signals. Their aim is to obtain an optimal maintenance and spare part quantity decisions by utilizing aforementioned two processes which is similar to our study which we want to determine the optimal replacement time and also the number of spare parts to wait for. However, we are relaxing the assumption that each spare part is as good as new, allowing for variability in quality such that some components are considered superior to others. An MDP model is formulated to determine optimal intervention policy which minimizes long-run expected cost. Their model is also a representative of a regenerative process. Sensitivity analysis is conducted to observe the effect of system characteristics on the system performance.

Abdul-Malak et al. [7] address the optimal maintenance problem of a stochastically degrading, single-unit system with heterogeneous spare parts. In their model, the available actions include "do nothing," "repair," and "replace." The repair action restores the system to an as-good-as-new condition, while the replace action involves the random selection of a spare part from a heterogeneous supply, resembling the definition of our replace action. The problem is formulated as a mixed observability Markov Decision Process (MOMDP). Similar to our findings, they show the monotonicity of the value function and existence of finite threshold type policies. Although their modeling of deterioration with unit-to-unit variability in spare parts is conceptually aligned with this study, they employ Bayesian inference and belief networks in addition to the MDP tools we have used. Unlike our model, their approach updates quality probabilities over time based on observations.

Byon et al. [8] investigate another optimal maintenance problem, focusing on determining the optimal repair decisions for wind turbines to minimize the system's operations and maintenance costs. They develop a partially observable Markov decision process (POMDP) model to derive an optimal preventive maintenance policy. Their model includes three possible actions: no action, preventive maintenance, and observation. This leads to monotonic four-region policies, which differ from the threshold-type policies identified in our study. Additionally, their model accounts for a stochastic environment influenced by weather conditions, which significantly impact the repair activities for wind turbines. Similar to our approach, they structure their problem as an average expected cost model.

Hu and Yue [9] examined an optimal replacement problem with the goal of minimizing the total discounted cost, albeit from a different perspective. Their model assumes a system that deteriorates according to a semi-Markov process, which operates under the influence of a semi-Markov environment. Similar to the approach in this thesis, they demonstrate the existence of an optimal control policy of the control-limit type. To simplify computations, they also modify the state space to be finite which is parallel to the methodology adopted in this study.

Ciocan and Misic [10] examine an optimal stopping problem by using decision trees which is different than many studies in the literature. Their aim is to find an interpretable optimal policy which shows the relation between states and policies explicitly. They formulated a sample average approximation problem and implemented the algorithm with option pricing data. Their policy is the form of binary tree which is different than the methodology used in our study as well as other papers.

The main distinction between our study and existing maintenance models in the literature is that many of these models incorporate the effect of the environment on the degradation process. As a result, they typically consider two different Markov chains, and their interactions influence the optimal decisions.

Secondly, the related healthcare operations research literature is reviewed. As it is stated before, optimal replacement problem can be considered as a specific form of the optimal stopping problems. The concept of optimal stopping time has also been extensively studied in healthcare applications by numerous researchers.Determining the optimal time for decision-making is essential in many medical contexts. One prominent area where such problems are frequently addressed is organ transplantation, where the timing of key decisions is critical to optimizing patient outcomes and resource utilization. Alagoz et al. [11] address a critical decision problem in the management of end-stage liver disease, focusing on the options of transplantation from a living donor, cadaveric donors, or delaying the procedure for a future period. The authors formulate the problem as a discrete-time, infinite-horizon Markov Decision Process (MDP) model, aiming to maximize the patient's total expected discounted reward. Their analysis results in an optimal policy characterized by a threshold structure. Kidney exchange procedure in a dynamically evolving agent pool with time and compatibility based preferences is studied by Unver [12]. Different than most papers in the relevant literature, he focused on barter exchange and accordingly state space of their model denotes available number of different pair types. However, Bellman equations are solved in order to obtain an matching mechanism which is a threshold type policy.

The optimal time of treatment initiation is another common area in healthcare operations research. Kurt et al. [13] formulated a discrete time infinite horizon MDP to find optimal time of statin initiation for patients with Type 2 diabetes to maximize quality adjusted life years of the patient. Similar to our study, states that denotes health status of patients are discretized and optimal control limit type policy is obtained. Shechter et al. [14] consider similar problem to find the optimal time to initiate HIV therapy. MDP is utilized to model the problem that maximize expected lifetime of the patient which leads to optimal control-limit policies.

The main difference of these healthcare literature and our model is that other studies formulate a transient MDP model while our model is recurrent. Although this recurrent structure of the Markov Chain is not completely suitable for a healthcare problem,it can be implemented to medical decision making problem with additional assumptions such as determining an upper-bound for the number of replacement decision in our model. Since we are formulating a recurrent MDP structure, our model does not include any failure states. It is assumed that the system can continue to operate as long as the deteriorated component is replaced with a spare part. However, the base model is subsequently extended to incorporate states that represent the inoperable condition of the system. These states, though, are not absorbing and do not alter the recurrent nature of the model. The inclusion of these states allows for the observation of system behavior under an externally imposed replacement policy for inoperable states. Furthermore, the Bellman equations of our model are not discounted, in contrast to many models in the literature. Instead, we formulate the Bellman equations based on the average reward criterion for the structured model. When maintenance and healthcare models are considered together, it can be concluded that our model is more directly applicable to maintenance applications. However, with some additional assumptions, it can be easily adapted to healthcare problems.

The reviewed studies are also summarized in the Table 2.1, where they are categorized based on their main context, the problem addressed, and the methods employed.

Table 2.1. Summary of studies in literature

Author	**Context**	**Problem**	**Method**
İçten et al.	Maintenance	Optimal time of limited number of replacements	MDP(discrete-time, absorbing state) and Sample Path Analysis
Kurt and Kharoufeh	Maintenance	Optimal replacement time within a controllable environment	MDP(discrete-time, infinite horizon, discounted cost)
Ulukus et al.	Maintenance	Optimal time for preventive maintenance in a stochastic environment	MDP(continuous time, discounted cost)
Cekyay and Ozekici	Maintenance	Optimal repair and replacement problem under the effect of external missions	MDP(finite-state)
Karabag et al.	Maintenance	Optimal maintenance problem of a multi-component system with environment effect	POMDP and LP
Abdul-Malak et al.	Maintenance	Optimal maintenance problem with heterogeneous spare parts	MOMDP and Belief Networks
Byon et al.	Maintenance	Optimal maintenance problem in a stochastic environment	POMDP
Hu and Yue	Maintenance	Optimal maintenance problem in a stochastic environment	Semi Markov Process
Ciocan and Misic	Maintenance	Optimal stopping time problem for option pricing data	Decision Tree
Alagoz et al.	Healthcare	Optimal time for kidney transplantation	MDP(discrete-time, infinite horizon, absorbing state)
Unver	Healthcare	Optimal barter exchange matching mechanism	MDP(discrete time, finite horizon)
Kurt et al.	Healthcare	Optimal time to statin initiation	MDP(discrete-time, infinite horizon)
Shechter et al.	Healthcare	Optimal time to initiate HIV therapy	MDP(discrete time, absorbing state)

This study builds upon these established methodologies, leveraging dynamic programming and Markov decision processes to analyze and optimize replacement policies for deteriorating systems. Since MDP is a powerful framework for modeling decision making problems under uncertainty and particularly suited to our problem, it is employed in our study. By aligning with existing research, it seeks to extend the understanding of optimal replacement strategies and contribute novel insights to this well-established field. The model we introduced here differs from existing optimal replacement models in that the quality of spare parts is represented as a random variable with predetermined probability parameters which brings stochastic dimension to the replacement procedure and introduces significant challenges to the structure of optimal policies. Although the literature includes models which capture stochastic behavior of different systems, they do not refer to the unknown quality of spare parts and its probability distribution. We aim to find optimal replacement time of components for deteriorating systems where the exact quality of spare parts is unknown and to determine how many spare parts should be waited until replacement action. In this regard, we believe that our model addresses a gap in the literature so far and provides results that can be further developed in future research.

We prove the existence of a threshold type replacement policy and propose different threshold calculation strategies. Numerical analyses are provided to illustrate the structure of the optimal policy and assess the impact of the model parameters on policies by constructing sensitivity analyses. Several numerical examples are provided to show the structure of optimal threshold policies. The model is further extended by incorporating another assumptions and expanding the state space. Additional numeric examples are examined in order to show the impact of these extensions. The effect of an externally dictated optimal policies for some states to the system behavior are also considered and compared with previous results.

3. PROBLEM DEFINITION AND MODEL DESCRIPTION

This study aims to investigate an optimal stopping problem, specifically focusing on determining the best time for initiating a replacement action within a system to maximize the expected long-term reward in a steady-state context. In addition to the optimal time of the decision, the study also addresses the extent of system improvement following replacement. The level of improvement that replacement provides is uncertain both before and after the action, introducing a probabilistic element that adds complexity to the problem.

The system under consideration can exist in either operable or inoperable states. The operable states are further categorized based on the system's performance levels. In contrast, the inoperable state signifies system failure, rendering it incapable of functioning. The system undergoes gradual deterioration, transitioning from an initial operable condition toward an inoperable state. While in an operable condition, the system does not require external intervention and continues functioning, albeit with diminishing performance due to natural aging effects or external stressors. Over time, as the system's performance declines, it approaches the inoperable state, necessitating the replacement of its worn component with a spare part to restore its desired functionality. For this analysis, the inoperable state is not included in model formulation, and the system's behavior while operational is examined. The model focuses on a single critical component, whose degradation drives the overall system deterioration. Once the component deteriorates significantly, it can be replaced with a spare part to renew the system's condition. A common assumption in maintenance models is that spare parts (components or subsystems) are drawn from a homogeneous population, where each part shares identical degradation characteristics. However, in practice, replacement parts often exhibit considerable variability in their quality attributes from one unit to another. In our study, the spare parts are assumed to have random and unobservable quality levels, and their precise quality is not fully known. Since multiple types of spare parts are available, a replacement may transition the system into various operable states. Due to the uncertainty in spare part quality, the system's exact future states cannot be directly observed or predicted. During the waiting period for spare parts, the system continues to deteriorate further. It is assumed that deterioration persists until the replacement is performed.

3.1. Continuous Time Markov Chain Model

The replacement problem is modeled as a continuous time Markov Chain (CTMC). The deterioration process evolves as an ergodic continuous-time Markov Chain on an infinite state space. The states have a physical interpretation and a natural ordering. System states, represented as (h, k), have two dimensions: h reflects the system's condition, and k represents the number of spare parts. It is assumed that transitions between states follow independent exponential processes. States having positive h values indicate the system is in operable condition, with higher h values denoting better performance. States having non-positive h values reflect deteriorated state of the system and operation of the system with low performance, where deterioration exceeds the acceptable threshold and requires component replacement. Lower values of h indicate poorer operating conditions for the system. Since k correspond to number of spare parts, it can take non-negative values. Replacement is permitted when k is positive and h is non-positive. If k=0, no spare parts have arrived, so replacement is not possible. It is assumed that there are infinite number of potential spare parts. When h is positive, k is always zero as the system operates effectively and does not require replacements and any spare part.

There are two main processes in the system: the deterioration of the system over time and the arrival of new parts suitable for replacement. At any given time instant, only one event can occur, either an arrival or a deterioration. Furthermore, these events are independent of each other. New parts arrive at the system with a stationary arrival rate λ according to a Poisson process. The Poisson arrival rates mean exponential and memoryless interarrival time of spare parts. Thus, this assumption ensures Markovian nature of the model and also aligns well with real world scenarios where events occur randomly and independently over time. The values of λ can be any nonnegative number. The system degrades at a deterioration rate γ which is also stationary and does not depend on either the number of new components or the condition of the component. The system deteriorates gradually. The values of γ can be any nonnegative number. For boundary states, the system cannot deteriorate further and stays in its current state with rate γ.

At each time the system is observed, the operating condition of the system is determined. If the system operates, no external intervention is required for its operation. If the system starts to deteriorate, it is required to replace the component with a spare part in order to make the system operable again. The time of replacement is evaluated according to the system condition and available spare parts. It is possible to replace the deteriorated component with a spare part instantaneously or let the system operate in a deteriorated condition for a while.

Spare parts may vary in quality, offering different levels of improvement to the system's condition. It is possible to have infinite number of different quality components. It is assumed that there are two types of spare parts with different quality levels—low and high—referred to as Type 1 and Type 2 components. New arrivals are of Type 1 with probability p and of Type 2 with probability 1-p. It is assumed that types of spare parts are determined probabilistically and independently. Hence, observing particular sequence of spare part types does not depend on sequence history which confirms the Markov property. Moreover, it is known that Type 2 component is better than Type 1 component. There is a specific component choosing procedure during replacement. When the deteriorated component is replaced with a spare part, the system aims to select the spare part that maximizes system improvement among all available options. For example, if at least one Type 2 component is available, it is preferred over Type 1 components for replacement. Type 1 and Type 2 spare parts correspond to refurbished and re-manufactured spare parts in real life applications, respectively and it is assumed that new spare parts are not utilized, making the system stochastic in terms of quality levels. It is known that re-manufactured spare parts provide more consistent high quality performance than refurbished spare parts in many areas. On the other hand, refurbished spare parts can be preferred due to their more cost effective and lower environmental impact natures.

Therefore, the degree of system improvement and the next state resulting from replacing a deteriorated component with a new one are determined stochastically. This inherent randomness raises the question of how many new parts' arrivals should be anticipated to maximize the system's improvement and, consequently, the expected system reward. When the overall system is considered, it is seen that the initiative of waiting action is due to having two types of components with different random quality levels where one of them is apparently better than the other one. Otherwise, it would replace the component upon the arrival of first spare part and waiting for more parts would be unnecessary. The probabilistic nature of component types is believed to effectively represent the uncertainty encountered in real-world problems. In many common scenarios, the specific type of available components is not known with certainty, but can be predicted to some degree. For instance, availability of remanufactured and refurbished spare parts mostly depends on production cycles which brings uncertainty to the system.

3.2. Base Markov Decision Process Model

In order to find an optimal solution to the formulated optimal stopping problem, we generate a Markov Decision Process (MDP) model. After constructing the CTMC model, it is discretized using the uniformization technique, which introduces transitions from a state to itself [15] to formulate the problem as an average reward MDP. It is assumed that the system is monitored continuously and components are accordingly replaced. Since we aim to optimize long-term performance in a system without a clear end point, focusing on the average reward per time step is preferred rather than using a cumulative or discounted reward. When decisions are made frequently, the discount rate approaches 1. We develop the MDP as a function of operating condition of the system and the number of spare parts available for replacement. We utilize a dynamic programming approach and Bellman value equations to find the optimal stopping policy. The optimal policies we derived provide robust, state-dependent optimal decisions for given input parameters.

It is a continuous infinite horizon problem. We assume that the system condition can be explicitly observed at any time and the system can continue operating as long as deteriorated components are replaced with functional ones, making it a recurrent MDP model. The component h of the state space is discretized to solve continuous-state MDP problem. The state space is also bounded to decrease the computational burden. The state of the MDP is an ordered pair (h,k) where $h \in \{0,1,2\}$ and $k \in \{0,1,2,....,\infty\}$. States with h$\in \{1,2\}$ indicate the system is in operable condition, with higher h values denoting better performance. Thus, h=2 represents the system's optimal condition. States with h= 0 indicate that the system has deteriorated but remains operational, necessitating component replacement to restore the system's condition. Since the system deterioration is gradual, it occurs only by one unit decrements in h values. Replacement is allowed when $k \geq 1$ and h=0.

Hence, the state space of the system is shown as

$$S = \{(2,0), (1,0), (0,0), (0,1), (0,2), ..., (0,\infty)\}. \tag{3.1}$$

It is assumed that the system is controlled continuously and its state is determined accordingly. If it is performing, no action is required, and it can either deteriorate or continue to function in its current state with transitions occurring according to predefined probabilities. If it shows signs of deterioration, the decision maker can decide to replace the deteriorated component with a new one and restore its functional state, or wait and defer action until the next period. When a new component arrives to the system, the decision making process is triggered and either replace or wait decision is made. It is also probable to stay in the same condition or deteriorate further for the extended model as a result of both actions. The system does not experience a permanent failure state that would definitively terminate its lifetime. However, it may deteriorate to unusable states, rendering it inoperative or incapable of functioning effectively. In such cases, replacing the deteriorated component with a spare part can restore the system to an operational state. While formulating the model, it is assumed that states range from those representing optimal functioning conditions to boundary states that indicate the most degraded states allowable for system operation. Once the system is deteriorated and reaches the boundary states, it does not continue to deterioration and maintains its function with low performance. This setup captures a continuous cycle of deterioration, maintenance, and recovery, ensuring that the system can be sustained indefinitely by appropriate decision-making policies.

In the MDP model action set, q, includes replace and wait decisions, denoted by R and W, respectively. The possible actions are given in equation as

$$q \in \{W, R\}. \tag{3.2}$$

There is a binary decision between two alternative actions. Related decision state is given as

$$d = \left\{ \begin{array}{ll} 0, & \text{if } q^*(h,k) = W \\ 1, & \text{if } q^*(h,k) = R \end{array} \right\}. \tag{3.3}$$

Actions are only defined for the states where h is zero and k is positive. No actions are available in states (2,0) and (1,0) due to the proper system functioning, and in state (0,0) due to the unavailability of replacement components. Therefore, the wait action is the default option in these states. The system continues to its natural degradation without any external intervention due to lack of available actions in these three states. When wait decision is given in boundary states, the system maintains its current state. If wait action is determined for remaining states after an arrival of new component, it corresponds to let the system function for one more period without any replacement.

The replace action, on the other hand, entails instantaneously substituting the existing component with a new one. When the replace decision is made, the available new components in the system are evaluated, and the component of the highest quality is selected for replacement which is consistent with the selection procedure in real-life problems. According to our model definition, Type 1 components transition the system to state (1,0) while Type 2 components transition it to state (2,0). Since Type 2 components are considered superior to Type 1 components in terms of quality, they carry the system to state (2,0). It is also observed that the system prefers directly replacing with the arrival of first component under some parameter definitions which will be elaborated on numerical results section.

When the general operation of the system is considered, deterioration causes a negative impact on the system outcomes and this creates a cost for the system in the long run. Moreover, waiting for new components leads to system to function in a deteriorated condition with lower performance. Hence, increasing waiting time can be associated with higher cost for the system. On the contrary, improving the system state with the help of replacement decision brings a reward to the system. Thus, these rewards and costs can be considered as results of the decisions given. Consequently, it is assumed that replacement action is associated with a reward and this reward depends on the system improvement level, accordingly the type of the new component and its occurrence probability, p. The unknown nature of the replacement decision brings the model a probabilistic reward structure. The quality difference between the two types of spare parts is reflected in the varying levels of replacement rewards they contribute to the system's total expected reward. Waiting action is associated with a cost which depends on how long the system waits that can be deduced from number of new parts available, k. It is straightforward inference that more parts require more waiting time for the system.This also highlights the main trade-off of the problem which can be denoted by the relationship between number of spare parts and transition probability to state (2,0). As k increases, not only the probability of going to state (2,0), 1-p^k, increases but the waiting time also increases. Our objective is to determine the optimal number of components that maximizes the probability of having a Type 2 spare part, which is superior in quality, among the available options.

State transition probabilities represent the likelihood of transitioning from one state to another after a given action is selected, based on the current state of the system. These probabilities depend on both the chosen action and the present state of the system. It is assumed that the system experiences progressive deterioration in the absence of replacement actions; specifically, the system possesses an intrinsic deterioration property and cannot improve without external intervention. Thus, if left alone, the system deteriorates over time, represented by a one-unit decrement in the state parameter h, when h is positive. The system can also remain in its current state with a predefined probability if it does not deteriorate for positive values of h. Under a "wait" action, the system may either stay in its current state (h, k) or transition to a state $(h, k+1)$, where the state remains unchanged or a new part becomes available. If a "replace" action is taken, the system can improve by utilizing one of the highest quality parts available, potentially transitioning to well-functioning states, such as (2,0) or (1,0), or remaining in its current state (h, k). State transition probability matrix, Q, of the system is given as

$$Q = \begin{bmatrix} \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots \\ 0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & \dots & 0 & 0 & \dots \\ 0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & 0 & \dots & 0 & 0 & \dots \\ \frac{(1-p)d\lambda}{\Lambda} & \frac{pd\lambda}{\Lambda} & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & 0 & \dots & 0 & 0 & \dots \\ \frac{(1-p^2)d\lambda}{\Lambda} & \frac{p^2d\lambda}{\Lambda} & 0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & \dots & 0 & 0 & \dots \\ \vdots & & & & & \ddots & & & & \dots \\ \vdots & & & & & & \ddots & & & \dots \\ \vdots & & & & & & & \ddots & & \dots \\ \frac{(1-p^{k-1})d\lambda}{\Lambda} & \frac{p^{k-1}d\lambda}{\Lambda} & 0 & 0 & 0 & 0 & \dots & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & \dots \\ \frac{(1-p^k)d\lambda}{\Lambda} & \frac{p^kd\lambda}{\Lambda} & 0 & 0 & 0 & 0 & \dots & 0 & \frac{\gamma}{\Lambda} & \dots \\ \vdots & & & & & & & & & \ddots \end{bmatrix}. \tag{3.4}$$

The state transition diagram of the model is given in Figure 3.1.

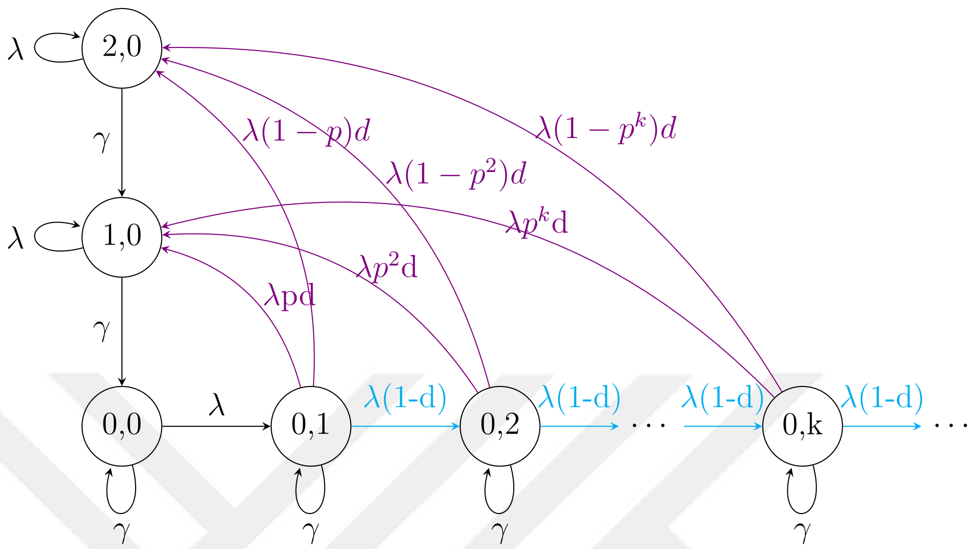

Figure 3.1. Transition Diagram of the Markov Decision Process Model

Accordingly, the rewards and costs associated with the decisions are explicitly defined. General reward function, $w(h, k, q)$, is formulated as a function of the taken action q in the current state (h, k). Related cost of the waiting decision is considered as a negative reward. Thus, it is not stated as another function. Moreover, it is also assumed that reward is not discounted for our model. There are also some constant reward parameters a,c and b. Upon a replacement action, it is assumed that the system transitions to states (1,0) or (2,0), which yield rewards of c and ac units, respectively. Here, c represents the base reward for transitioning to the state (1,0) and is treated as the unit reward for the system, while a is the reward coefficient for the state (2,0). When the waiting decision is implemented, the system incurs a negative reward equivalent to bc, with b representing the cost coefficient. The exact definition of the reward function is given as

$$w(0, k, W) = bck \tag{3.5}$$

$$w(0, k, R) = p^k c + (1 - p^k)ac, \tag{3.6}$$

where b is a non-positive number, c is a positive number and a is a positive number. Bellman equations for each state are formulated as it is shown as

$$\mathbf{V}_n(x,0) = \frac{\lambda}{\Lambda}\mathbf{V}_{n-1}(x,0) + \frac{\gamma}{\Lambda}\mathbf{V}_{n-1}(x-1,0), \tag{3.7}$$

where x ∈ 1,2 and

$$\mathbf{V}_n(0,0) = \frac{\lambda}{\Lambda}\mathbf{V}_{n-1}(0,1) + \frac{\gamma}{\Lambda}\mathbf{V}_{n-1}(0,0), \tag{3.8}$$

$$\mathbf{V}_n(0,k,W) = \frac{\lambda}{\Lambda}\mathbf{V}_{n-1}(0,k+1) + \frac{\gamma}{\Lambda}\mathbf{V}_{n-1}(0,k) + w(0,k,W), \tag{3.9}$$

$$\mathbf{V}_n(0,k,R) = \frac{\lambda}{\Lambda}(p^k\mathbf{V}_{n-1}(1,0) + (1-p^k)\mathbf{V}_{n-1}(2,0)) + \frac{\gamma}{\Lambda}\mathbf{V}_{n-1}(0,k) + w(0,k,R), \tag{3.10}$$

where k ≥ 1. The initial conditions are given as

$$\mathbf{V}_0(2,0) = \mathbf{V}_0(1,0) = \mathbf{V}_0(0,0) = \mathbf{V}_0(h,k,R) = \mathbf{V}_0(h,k,W) = 0, \tag{3.11}$$

where $k \geq 1$ and h =0.

The goal of our study is to determine the optimal number of spares to wait for before replacement becomes superior action compared to waiting action. When the number of components available reach this critical threshold value, the system is not interested in waiting for another extra part and prefers to choose the best component from the available parts. When the logic behind the problem is considered, it is expected to have an optimal solution in the form of control limit type policy. Hence, the aim is to find an optimal switching point between wait and replace decisions. Accordingly, determination of this critical point allows us to obtain robust optimal policies for the system which includes the optimal decisions in each state under the predefined initial conditions of the system. This analysis enables the determination of the optimal number of parts to accumulate before replacement, as well as the maximum duration the system can be allowed to operate in a deteriorated state. Accordingly, the policy makers or responsibles can plan the replacement or maintenance process in advance and anticipate the probable incoming difficulties.

One of the main results of our model is that there is a control limit point k^* that replacement option becomes optimal beyond this critical threshold, i.e. $k \geq k^*$ and wait option is optimal when the number of parts is less than this threshold, i.e. $k < k^*$. Hence, one dimensional control limit point in terms of k can be obtained by solving the balance equations of the model. Another important result is that the obtained control limit points are optimal. Therefore, optimal replacement policies for the system can be found.

3.3. Structural Properties

The optimality of the control limit policies are guaranteed by the structural characteristics of the model. There are some inherent properties of the value function which stem from the structured definition of the model. These properties are important in terms of finding the solution to our problem because it is proven that some of them are required to have an optimal solution. When our model is analyzed, it can be observed that the value function $V_n(h, k)$ is increasing in the given natural degrading order of the states. Moreover, the difference of $V_n(h, k+1)$ - $V_n(h, k)$ is non-increasing for the replace decision of the optimal actions. In other words, when the incremental benefit of extra parts decreases as the number of parts increase, the replace decision becomes optimal.

Since one of the main contributors of the balance equations is transition probability matrix, its structure also takes part in the optimality of the results. Barlow and Proschan [16] proved that optimal replacement rule is a control limit type rule if the underlying Markov chain has an increasing failure rate transition probability matrix (IFR).

Definition 1: A transition matrix is said to be an IFR if P($\mathbf{X}_{n+1} \in B|\mathbf{X}_n = i$) is non-decreasing in i for every set B of the form B = $l, l+1, ..., m$ for some $l = 0, 1, ..., m$; that is

$$f(i) = \sum_{j=l}^{m} \mathbf{q}_{ij} \tag{3.12}$$

is non-decreasing in i for all $l = 0, 1, ..., m$, where $\mathbf{q}_{ij} = P_{(j \in B)}(\mathbf{X}_{n+1} = j|\mathbf{X}_n = i)$.

When the probability transition matrix between states where h is 0 for waiting decision is considered, it is seen that the corresponding matrix has IFR property. Hence, this statement also supports the existence of a threshold type optimal policy for our replacement problem.

The following structural properties are derived under the assumption of a stationary process. Since the model exhibits the Markov property, the obtained optimal policies and underlying Markov Chain are stationary and independent of time in the long run. Hence, demonstrating the stationarity of the derived policies is not considered necessary in this study. Instead, their existence and calculations are examined, as will be shown below.

Ross demonstrated the existence of a stationary optimal policy in his theorem [17] given as

$$g + l(i) = \max_{q}[w(i,q) + \sum_{j=0}^{\infty} P_{ij}(q)l(j)], \tag{3.13}$$

where q denotes any action from action space, $l(i)$ denotes a bounded function, $w(i,q)$ denotes reward function, P denotes probability transitions, and $i \geq 0$. Then there exists a stationary policy μ^* such that

$$g = \phi_{\mu^*}(i) = \max_{\mu} \phi_{\mu}(i), \tag{3.14}$$

for all i ≥ 0, ϕ is average expected return per unit time and μ^* is any policy prescribes an action that maximizes the right hand-side of equation 3.13. In our formulation, r and $V(h,k)$ correspond to g and $l(i)$ functions in the equation 3.13.

In order to check the existence of an optimal control limit type policy in k, $\mathbf{V}_n(0,k,W) - V_n(0,k,R)$ function can be considered. If this difference function is decreasing in k, it can be concluded that there is a control limit type policy. Starting from 1, as the value of k increases, there must be a point where this function becomes zero or negative first time. This point constitutes our critical limit where replace action becomes optimal first time. It is also optimal to replace for k values larger than this threshold. Thus, it is sufficient to show that $\mathbf{V}_n(0,k,W) - V_n(0,k,R)$ function is non-increasing in k to have an optimal control limit solution. The monotonic behavior of the function is shown by using proof by contradiction.

Let us define a difference function $\mathbf{\Delta} V_n(0,k)$ which measures the difference of two actions in each state (0,k) as

$$\mathbf{\Delta} V_n(0,k) = \mathbf{V}_n(0,k,W) - \mathbf{V}_n(0,k,R). \tag{3.15}$$

Therefore it is desired to have the following inequality

$$\mathbf{\Delta} V_{n+1}(0,k) \geq \mathbf{\Delta} V_{n+1}(0,k+1) \tag{3.16}$$

to hold.

A control limit k^* exists iff $\mathbf{\Delta} V_{n+1}(0,k^*-1) > 0$ and $\mathbf{\Delta} V_{n+1}(0,k^*) < 0$. Let us consider any two states (0,k-1) and (0,k). It can be shown that the inequality 3.16 is true for these two states as below

$$\mathbf{V}_{n+1}(0,k-1,W) - \mathbf{V}_{n+1}(0,k-1,R) \geq \mathbf{V}_{n+1}(0,k,W) - \mathbf{V}_{n+1}(0,k,R). \tag{3.17}$$

When $\varDelta V_{n+1}(0,k-1) > 0$, $q^*(0,k-1) = W$ is the optimal action. When $\varDelta V_{n+1}(0,k) < 0$, $q^*(0,k) = R$ will be the optimal result.

Therefore, their value functions satisfy the following in the long run such as

$$\mathbf{V}_{n+1}(0,k-1,W) \geq \mathbf{V}_{n+1}(0,k-1,R), \tag{3.18}$$

$$\mathbf{V}_{n+1}(0,k,R) \geq \mathbf{V}_{n+1}(0,k,W) \tag{3.19}$$

First, the difference of value functions for replace actions is considered for (0,k-1) and (0,k) states as

$$\mathbf{V}_{n+1}(0,k,R) = \frac{\lambda}{\varLambda}(p^k \mathbf{V}_n(1,0) + (1-p^k)\mathbf{V}_n(2,0)) + \frac{\gamma}{\varLambda}\mathbf{V}_n(0,k) + w(0,k,R), \tag{3.20}$$

$$\mathbf{V}_{n+1}(0,k-1,R) = \frac{\lambda}{\varLambda}(p^{k-1}\mathbf{V}_n(1,0) + (1-p^{k-1})\mathbf{V}_n(2,0)) + \frac{\gamma}{\varLambda}\mathbf{V}_n(0,k-1) + w(0,k-1,R), \tag{3.21}$$

$$\begin{aligned} \mathbf{V}_{n+1}(0,k,R) - \mathbf{V}_{n+1}(0,k-1,R) &= \frac{\lambda}{\varLambda}(p^{k-1}(p-1)(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0))) \\ &+ \frac{\gamma}{\varLambda}(\mathbf{V}_n(0,k) - \mathbf{V}_n(0,k-1)) + w(0,k,R) - w(0,k-1,R). \end{aligned} \tag{3.22}$$

When inequalities (3.18) and(3.19) are evaluated and simplified, the inequalities are obtained as

$$\frac{\lambda}{\Lambda}(p^{k-1}\mathbf{V}_n(1,0)+(1-p^{k-1})\mathbf{V}_n(2,0))+w(0,k-1,R)\leq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k)+w(0,k-1,W), \tag{3.23}$$

$$\frac{\lambda}{\Lambda}(p^{k}\mathbf{V}_n(1,0)+(1-p^{k})\mathbf{V}_n(2,0))+w(0,k,R)\geq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)+w(0,k,W). \tag{3.24}$$

Inequality (3.23) is multiplied with -1 such as

$$-\frac{\lambda}{\Lambda}(p^{k-1}\mathbf{V}_n(1,0)+(1-p^{k-1})\mathbf{V}_n(2,0))-w(0,k-1,R)\geq-\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k)-w(0,k-1,W). \tag{3.25}$$

Both sides of inequalities (3.24) and (3.25) are added as

$$\begin{aligned}\frac{\lambda}{\Lambda}(p^{k}\mathbf{V}_n(1,0)+(1-p^{k})\mathbf{V}_n(2,0))+w(0,k,R)-\frac{\lambda}{\Lambda}(p^{k-1}\mathbf{V}_n(1,0)+(1-p^{k-1})\mathbf{V}_n(2,0))\\ -w(0,k-1,R)\geq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)+w(0,k,W)-\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k)-w(0,k-1,W).\end{aligned} \tag{3.26}$$

$\frac{\gamma}{\Lambda}$ $\mathbf{V}_n(0,k)$ - $\frac{\gamma}{\Lambda}$ $\mathbf{V}_n(0,k-1)$ term is added to the both sides of the equation (3.26) as

$$\begin{aligned}\frac{\lambda}{\Lambda}(p^{k}\mathbf{V}_n(1,0)+(1-p^{k})\mathbf{V}_n(2,0))+w(0,k,R)-\frac{\lambda}{\Lambda}(p^{k-1}\mathbf{V}_n(1,0)+(1-p^{k-1})\mathbf{V}_n(2,0))\\ -w(0,k-1,R)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k)-\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k-1)\geq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)+w(0,k,W)\\ -\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k)-w(0,k-1,W)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k)-\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k-1).\end{aligned} \tag{3.27}$$

Right hand-side of the equation (3.27) is arranged and the following is obtained as

$$\begin{aligned}\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k)+w(0,k,W)-\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k)-\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k-1)-w(0,k-1,W)\\ =\mathbf{V}_{n+1}(0,k,W)-\mathbf{V}_{n+1}(0,k-1,W).\end{aligned} \tag{3.28}$$

In conclusion, it is shown that difference equation provides desired property for states $(0, k-1)$ and $(0, k)$ as

$$\mathbf{V}_{n+1}(0,k,R) - \mathbf{V}_{n+1}(0,k-1,R) \geq \mathbf{V}_{n+1}(0,k,W) - \mathbf{V}_{n+1}(0,k-1,W), \quad (3.29)$$

$$\mathbf{V}_{n+1}(0,k-1,W) - \mathbf{V}_{n+1}(0,k-1,R) \geq \mathbf{V}_{n+1}(0,k,W) - \mathbf{V}_{n+1}(0,k,R). \quad (3.30)$$

In addition, assume that there is no control limit type policy and the optimal action for state (0,k+1) is wait such as

$$a^*(0,k+1) = W, \quad (3.31)$$

$$\mathbf{V}_{n+1}(0,k,W) \leq \mathbf{V}_{n+1}(0,k,R), \quad (3.32)$$

$$\mathbf{V}_{n+1}(0,k+1,R) \leq \mathbf{V}_{n+1}(0,k+1,W) \quad (3.33)$$

The difference of replace value functions for state (0,k+1) and (0,k) is found as

$$\mathbf{V}_{n+1}(0,k+1,R) - \mathbf{V}_{n+1}(0,k,R) = \frac{\lambda}{\Lambda}(p^{k+1}\mathbf{V}_n(1,0) + (1-p^{k+1})\mathbf{V}_n(2,0)) + \frac{\gamma}{\Lambda}\mathbf{V}_n(0,k+1)$$
$$+ w(0,k+1,R) - \frac{\lambda}{\Lambda}(p^k\mathbf{V}_n(1,0) + (1-p^k)\mathbf{V}_n(2,0)) - \frac{\gamma}{\Lambda}\mathbf{V}_n(0,k) - w(0,k,R). \quad (3.34)$$

When inequalities (3.29) and (3.30) are evaluated and simplified, the following inequalities are obtained as

$$\frac{\lambda}{\Lambda}(p^k\mathbf{V}_n(1,0) + (1-p^k)\mathbf{V}_n(2,0)) + w(0,k,R) \geq \frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1) + w(0,k,W), \quad (3.35)$$

$$\frac{\lambda}{\Lambda}(p^{k+1}\mathbf{V}_n(1,0) + (1-p^{k+1})\mathbf{V}_n(2,0)) + w(0,k+1,R) \leq \frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+2) + w(0,k+1,W). \quad (3.36)$$

Inequality (3.35) is multiplied with -1 as

$$-\frac{\lambda}{\Lambda}(p^k\mathbf{V}_n(1,0) + (1-p^k)\mathbf{V}_n(2,0)) - w(0,k,R) \leq -\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1) - w(0,k,W). \quad (3.37)$$

Both sides of inequalities (3.36) and (3.37) are added as

$$\frac{\lambda}{\Lambda}(p^{k+1}\mathbf{V}_n(1,0) + (1-p^{k+1})\mathbf{V}_n(2,0)) + w(0,k+1,R) - \frac{\lambda}{\Lambda}(p^k\mathbf{V}_n(1,0) + (1-p^k)\mathbf{V}_n(2,0))$$

$$- w(0,k,R) \leq \frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+2) + w(0,k+1,W) - \frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1) - w(0,k,W). \tag{3.38}$$

$\frac{\gamma}{\Lambda}$ $\mathbf{V}_n(0,k+1)$ - $\frac{\gamma}{\Lambda}$ $\mathbf{V}_n(0,k)$ term is added to the both sides of the above equation (3.38) such as

$$\begin{aligned}
&\frac{\lambda}{\Lambda}(p^{k+1}\mathbf{V}_n(1,0)+(1-p^{k+1})\mathbf{V}_n(2,0))+w(0,k+1,R)-\frac{\lambda}{\Lambda}(p^{k}\mathbf{V}_n(1,0)+(1-p^{k})\mathbf{V}_n(2,0))\\
&-w(0,k,R)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k+1)-\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k) \leq \frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+2)+w(0,k+1,W)-\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)\\
&\qquad - w(0,k,W) + \frac{\gamma}{\Lambda}\mathbf{V}_n(0,k+1) - \frac{\gamma}{\Lambda}\mathbf{V}_n(0,k).
\end{aligned} \tag{3.39}$$

Right hand-side of the equation (3.39) is arranged and the following is obtained as

$$\begin{aligned}
&\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+2)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k+1)+w(0,k+1,W)-\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k+1)-\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k)-w(0,k,W)\\
&\qquad = \mathbf{V}_{n+1}(0,k+1,W) - \mathbf{V}_{n+1}(0,k,W).
\end{aligned} \tag{3.40}$$

In conclusion, the following inequality is obtained as

$$\mathbf{V}_{n+1}(0,k+1,R) - \mathbf{V}_{n+1}(0,k,R) \leq \mathbf{V}_{n+1}(0,k+1,W) - \mathbf{V}_{n+1}(0,k,W) \tag{3.41}$$

$$\mathbf{V}_{n+1}(0,k,W) - \mathbf{V}_{n+1}(0,k,R) \leq \mathbf{V}_{n+1}(0,k+1,W) - \mathbf{V}_{n+1}(0,k+1,R). \tag{3.42}$$

This result implies a contradiction with the non-increasing $\mathbf{\Delta}V_n(0,k)$ function. Hence, the Markov model and its structural properties guarantee a control limit type optimal policy for the given replacement problem.

After the system is initialized, it is guaranteed to reach the optimal policy in the long run. When the steady state system is analyzed, it is assumed that the difference between value functions at consecutive iterations converges to a constant value. Bellman's theorem [18] states this asymptotic behavior of the value functions . Bellman [18] obtained the following asymptotic result in his paper such as

$$\mathbf{V}_n(i) \sim Nr, \tag{3.43}$$

as $n \to \infty$ and i $\in$ S. The scalar quantity r is calculated as

$$r = \max_{q} \lim_{N \to -\infty} [\frac{u(q) + A(q)u(q) + ... + u(q)^{N-1}u(q)}{N}], \tag{3.44}$$

where q denotes the possible policies of the system, u is action-state based reward matrix and A is transition probability matrix.

Definition 2: The r quantity for the model is defined as

$$\lim_{n \to \infty} \mathbf{V_{n+1}}(h, n) - \mathbf{V_n}(h, n) = r. \tag{3.45}$$

This r quantity can be interpreted as the average expected reward of the system per iteration. The system gains this amount of expected reward at each iteration in the steady state.

After establishing the existence of an optimal control-limit policy, the next step is to determine the calculation of this control-limit point. Specifically, we seek the smallest k , denoted as k^*, at which the value function for the replacement action first becomes greater than or equal to the value function for the wait action. In other words, the optimal control point k^* is the smallest integer within the set of k values where the difference function $\mathbf{\Delta} V_n(0, k)$ is non-positive. The optimal threshold point is calculated under the assumption of stationary policies exist for average reward criterion. Let $\mathbf{\Delta} V(0, k) = \lim_{n \to \infty} V_n(0, k, W) - V_n(0, k, R)$ and threshold point can be found as

$$\inf\{k : \mathbf{\Delta} V(0, k) < 0\} = k^*. \tag{3.46}$$

The second result determines the decision point of the system. It represents the number of components that corresponds to the first time replace action becomes more preferable than wait action. Whenever the system reaches this state, it is not required to wait for new parts anymore and replace action is optimal after that which can be expressed as

$$\mathbf{V}_{n+1}(0, k^*, R) \geq \mathbf{V}_{n+1}(0, k^*, W), \tag{3.47}$$

$$\frac{\lambda}{\Lambda}(p^{k^*}\mathbf{V}_n(1,0)+(1-p^{k^*})\mathbf{V}_n(2,0))+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k^*)+w(0,k^*,R)\geq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k^*+1)+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k^*)+w(0,k^*,W), \quad (3.48)$$

$$\frac{\lambda}{\Lambda}(p^{k^*}\mathbf{V}_n(1,0)+(1-p^{k^*})\mathbf{V}_n(2,0))+w(0,k^*,R)\geq\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k^*+1)+w(0,k^*,W), \quad (3.49)$$

$$\frac{\lambda}{\Lambda}p^{k^*}(\mathbf{V}_n(1,0)-\mathbf{V}_n(2,0))+w(0,k^*,R)-w(0,k^*,W)\geq\frac{\lambda}{\Lambda}(\mathbf{V}_n(0,k^*+1)-\mathbf{V}_n(2,0)). \quad (3.50)$$

It is assumed that k^* is the decision point where replace action becomes optimal. Hence, it is also optimal to replace thereafter. Since the long run behavior of the system is considered, it can be concluded that replace is the optimal action in state $(0,k^*+i)$ where $i\in 1,2,..,\infty$ that can be shown as

$$\mathbf{V}_{n+1}(0,k^*+1,R)=\frac{\lambda}{\Lambda}(p^{k^*+1}\mathbf{V}_n(1,0)+(1-p^{k^*+1})\mathbf{V}_n(2,0))+\frac{\gamma}{\Lambda}\mathbf{V}_n(0,k^*+1)+w(0,k^*+1,R). \quad (3.51)$$

Moreover, value function for state (0, k^*+1) can be expressed in terms of r value such as

$$\mathbf{V}_{n+1}(0,k^*+1,R)=\mathbf{V}_n(0,k^*+1,R)+r. \quad (3.52)$$

The value functions can be rewritten by using Definition 2 as

$$\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k^*+1,R)+r=\frac{\lambda}{\Lambda}(p^{k^*+1}\mathbf{V}_n(1,0)+(1-p^{k^*+1})\mathbf{V}_n(2,0))+w(0,k^*+1,R), \quad (3.53)$$

$$\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k^*+1,R)=\frac{\lambda}{\Lambda}(p^{k^*+1}\mathbf{V}_n(1,0)+(1-p^{k^*+1})\mathbf{V}_n(2,0))+w(0,k^*+1,R)-r. \quad (3.54)$$

Alternatively, r can be expressed in terms of $\mathbf{V}_n(1,0)$ and $\mathbf{V}_n(2,0)$ such as

$$\mathbf{V}_{n+1}(2,0)=\frac{\lambda}{\Lambda}\mathbf{V}_n(2,0)+\frac{\gamma}{\Lambda}\mathbf{V}_n(1,0), \quad (3.55)$$

$$\mathbf{V}_n(2,0)+r=\frac{\lambda}{\Lambda}\mathbf{V}_n(2,0)+\frac{\gamma}{\Lambda}\mathbf{V}_n(1,0), \quad (3.56)$$

$$\frac{\gamma}{\Lambda}\mathbf{V}_n(2,0) + r = \frac{\gamma}{\Lambda}\mathbf{V}_n(1,0), \tag{3.57}$$
$$r = \frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)). \tag{3.58}$$

Then, equation 3.54 can be rewritten as

$$\frac{\lambda}{\Lambda}\mathbf{V}_n(0,k^*+1,R) = \frac{\lambda}{\Lambda}(p^{k^*+1}\mathbf{V}_n(1,0) + (1-p^{k^*+1})\mathbf{V}_n(2,0)) + w(0,k^*+1,R) - \frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) \tag{3.59}$$

The equation (3.59) is plugged into inequality (3.50) that is shown as

$$\frac{\lambda}{\Lambda}p^{k^*}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*,R) - w(0,k^*,W) \geq \frac{\lambda}{\Lambda}(p^{k^*+1}\mathbf{V}_n(1,0) + (1-p^{k^*+1})\mathbf{V}_n(2,0)) + w(0,k^*+1,R) - \frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) - \frac{\lambda}{\Lambda}\mathbf{V}_n(2,0), \tag{3.60}$$

$$\frac{\lambda}{\Lambda}p^{k^*}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*,R) - w(0,k^*,W) \geq \frac{\lambda}{\Lambda}p^{k^*+1}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + \frac{\lambda}{\Lambda}\mathbf{V}_n(2,0) - \frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*+1,R) - \frac{\lambda}{\Lambda}\mathbf{V}_n(2,0), \tag{3.61}$$

$$\frac{\lambda}{\Lambda}p^{k^*}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*,R) - w(0,k^*,W) \geq \frac{\lambda}{\Lambda}p^{k^*+1}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) - \frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*+1,R), \tag{3.62}$$

$$\frac{\lambda}{\Lambda}p^{k^*}(1-p)(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*,R) - w(0,k^*,W) \geq -\frac{\gamma}{\Lambda}(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) + w(0,k^*+1,R), \tag{3.63}$$

$$(\frac{\lambda}{\Lambda}p^{k^*}(1-p) + \frac{\gamma}{\Lambda})(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) \geq w(0,k^*+1,R) - w(0,k^*,R) + w(0,k^*,W). \tag{3.64}$$

Reward functions for given states and actions are evaluated as

$$(\frac{\lambda}{\Lambda}p^{k^*}(1-p) + \frac{\gamma}{\Lambda})(\mathbf{V}_n(1,0) - \mathbf{V}_n(2,0)) \geq p^{k^*+1}c + (1-p^{k^*+1})ac - p^{k^*}c - (1-p^{k^*})ac + bck^*, \tag{3.65}$$

$$(\frac{\lambda}{\Lambda}p^{k^*}(1-p)+\frac{\gamma}{\Lambda})(\mathbf{V}_n(1,0)-\mathbf{V}_n(2,0)) \geq p^{k+1}c+ac-p^{k^*+1}ac-p^{k^*}c-ac+p^{k^*}ac + bck^*, \quad (3.66)$$

$$(\frac{\lambda}{\Lambda}p^{k^*}(1-p)+\frac{\gamma}{\Lambda})(\mathbf{V}_n(1,0)-\mathbf{V}_n(2,0)) \geq bck^*-p^{k^*}c(1-a)(1-p), \quad (3.67)$$

$$\mathbf{V}_n(1,0)-\mathbf{V}_n(2,0) \geq \frac{\Lambda(bck^*-cp^{k^*}(1-a)(1-p))}{\lambda p^{k^*}(1-p)+\gamma}. \quad (3.68)$$

Minimum k value that satisfy the inequality (3.68) is the control limit point of the system that can be expressed as

$$k^* = \inf\{k : \frac{\Lambda(bck-cp^k(1-a)(1-p))}{\lambda p^k(1-p)+\gamma} \leq \mathbf{V}_n(1,0)-\mathbf{V}_n(2,0)\}. \quad (3.69)$$

The k^* can be expressed by using r value as

$$\frac{\Lambda}{\gamma}r \geq \frac{\Lambda(bck-cp^k(1-a)(1-p))}{\lambda p^k(1-p)+\gamma}, \quad (3.70)$$

$$r \geq \frac{\gamma(bck-cp^k(1-a)(1-p))}{\lambda p^k(1-p)+\gamma}, \quad (3.71)$$

$$k^* = \inf\{k : \frac{\gamma(bck-cp^k(1-a)(1-p))}{\lambda p^k(1-p)+\gamma} \leq r\}. \quad (3.72)$$

Hence, optimal control limit point is found after expected reward of the system, r, is calculated. Alternatively, it can also be deduced from value functions without calculating r value. Numerical computation of optimal policies for different input parameters and related results are provided in section 4, Numerical Results.

3.4. Equivalent Finite State Markov Chain Model

Our analysis yields two primary results which denote the switching behavior of optimal policies at critical point and the property of states above the switching point. First step shows that the replacement problem has an optimal control limit type policy with its defined structure and it is possible to calculate optimal switching points by using parameters and balance equations. When we solve this problem, obtained optimal policies indicate the desired threshold, k*, which denotes the number of components to wait before taking replacement action. Once this threshold, k*, is determined, there is no need to investigate the system further for more components than k*. Since replacement action becomes optimal at and beyond this critical point, it is meaningful to replace the component with one of the already arrived new components at the critical point and not to wait for extra components beyond k*. When replacement decision is given, the component with the highest quality among available components is chosen to replace with the deteriorated one. The probability of choosing the highest quality component is also at its optimal value in this threshold point. Hence, it is meaningful to consider the behavior of the system during the time period until number of new components reaches the critical threshold. Accordingly, we can focus on the system at and before the critical time and neglect the potential next arrivals after the critical point. Since the second dimension of our states represents number of existing new components, state space also enlarges with number of upcoming arrivals and reaches to infinity in the long run. By using this insight, we remodel the fundamental infinite state Markov Decision Process as a finite state Markov chain. Since calculation of r is sometimes cumbersome for the base model, this alternative Markov Chain model also simplifies some computations. For instance, steady state distributions of the system in the long run can be calculated easily by the aid of this second model. Thereafter, the optimal control limit points can be deduced with less effort than it is required for the MDP model.

The system configuration and parameter definitions are also valid for this alternative model. State space is truncated at (0,k*) and the number of states is limited to k*+3. Finite state space is given as

$$\mathbf{S} = \{(2,0),(1,0),(0,0),(0,1),(0,2),....,(0,k^*-1),(0,k^*)\}. \tag{3.73}$$

It is assumed that the states from (0,1) to (0,k^*-1) correspond to the waiting decision and the state (0,k^*) corresponds to the replace decision in the original Markov Decision Process model. This assumption determines the structure of transition probability matrix which is given as

$$
\mathbf{P} = \begin{bmatrix}
\frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & \dots & 0 & 0 \\
0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \dots & 0 & 0 \\
0 & 0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & \dots & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{\gamma}{\Lambda} & \dots & 0 & 0 \\
\vdots & & & & \ddots & & & \vdots \\
\vdots & & & & & \ddots & & \vdots \\
\vdots & & & & & & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} \\
\frac{(1-p^{k^*})\lambda}{\Lambda} & \frac{p^{k^*}\lambda}{\Lambda} & 0 & 0 & 0 & \dots & 0 & \frac{\gamma}{\Lambda}
\end{bmatrix}. \tag{3.74}
$$

The state transition diagram of the model is shown in Figure 3.2.

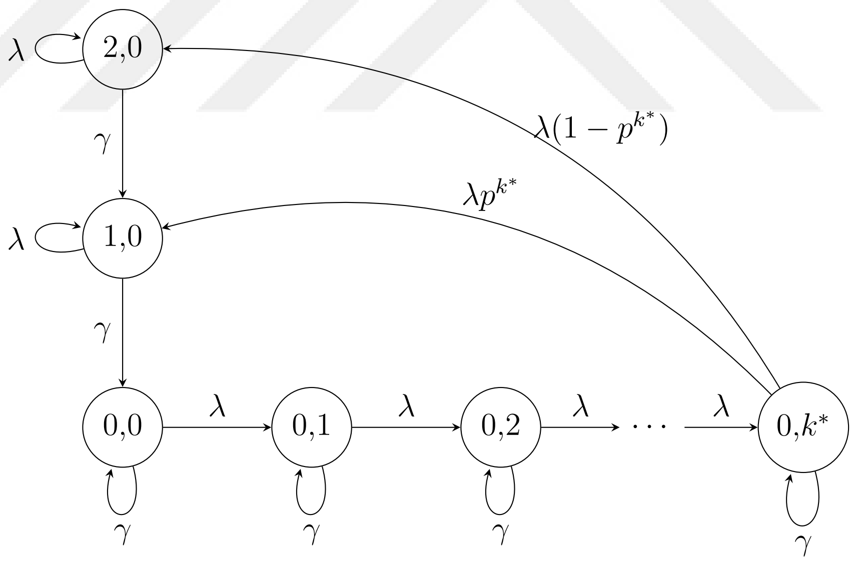

Figure 3.2. Transition Diagram of the Finite State Markov Chain Model

The exact value of k^* is initially unknown and the goal of the problem is to determine its specific value under different circumstances. Other than the base MDP model, this MC model does not include actions. However, the model includes the given decisions in each state inherently. As a result, it is possible to determine the rewards for being in each state, which are related to decisions made in each state in the original Markov Decision Process model. In this model, as it does not include any decisions, the rewards are predefined. It is supposed that w(j) reward is earned whenever the chain is in state j. Our generalized reward function w(j) is given as

$$w = \begin{bmatrix} 0 & 0 & 0 & bck & bck & \dots & p^{k^*}c + (1-p^{k^*})ac \end{bmatrix}, \tag{3.75}$$

where k is number of components in each state so $k \in \{1, 2, ..., k^* - 1, k^*\}$.

Steady state distributions are analyzed to determine optimal control limit point, k^*, and understand the system's long-run behavior. Stationary distributions, $\mathit{\Pi}$, can be calculated by using formulas given as

$$\mathbf{\Pi}' = \mathbf{\Pi}'\mathbf{P}, \tag{3.76}$$

$$\sum_{j \in S} \mathbf{\Pi}_j = \mathbf{1}. \tag{3.77}$$

Equation 3.76 is explicitly evaluated for each state that is given as

$$\pi_{(2,0)} = \frac{\lambda}{\Lambda}\pi_{(2,0)} + \frac{(1-p^{k^*})\lambda}{\Lambda}\pi_{(0,k^*)}, \tag{3.78}$$

$$\pi_{(1,0)} = \frac{\gamma}{\Lambda}\pi_{(2,0)} + \frac{\lambda}{\Lambda}\pi_{(2,0)} + \frac{p^{k^*}\lambda}{\Lambda}\pi_{(0,k^*)}, \tag{3.79}$$

$$\pi_{(0,0)} = \frac{\gamma}{\Lambda}\pi_{(1,0)} + \frac{\gamma}{\Lambda}\pi_{(0,0)}, \tag{3.80}$$

$$\pi_{(0,1)} = \frac{\lambda}{\Lambda}\pi_{(0,0)} + \frac{\gamma}{\Lambda}\pi_{(0,1)}, \tag{3.81}$$

$$\pi_{(0,2)} = \frac{\lambda}{\Lambda}\pi_{(0,1)} + \frac{\gamma}{\Lambda}\pi_{(0,2)}, \tag{3.82}$$

$$\vdots$$

$$\pi_{(0,k^*-1)} = \frac{\lambda}{\Lambda}\pi_{(0,k^*-2)} + \frac{\gamma}{\Lambda}\pi_{(0,k^*-1)}, \tag{3.83}$$

$$\pi_{(0,k^*)} = \frac{\lambda}{\Lambda}\pi_{(0,k^*-1)} + \frac{\gamma}{\Lambda}\pi_{(0,k^*)}. \tag{3.84}$$

By combining equations 3.78 to 3.84 with equation 3.77, stationary distributions are obtained as

$$\pi_{(2,0)} = \frac{(1-p^{k^*})\lambda}{(1-p^{k^*})\lambda + k^*\gamma + \Lambda}, \tag{3.85}$$

$$\pi_{(1,0)} = \frac{\lambda}{(1-p^{k^*})\lambda + k^*\gamma + \Lambda}, \tag{3.86}$$

$$\pi_{(0,0)} = \pi_{(0,1)} = \pi_{(0,2)} = \cdots = \pi_{(0,k^*-1)} = \pi_{(0,k^*)} = \frac{\gamma}{(1-p^{k^*})\lambda + k^*\gamma + \Lambda}. \tag{3.87}$$

It can be observed that in the long run, the probabilities of being in deteriorated states are the same. Since a similar pattern can be tracked in the transition probability matrix for these states, it is expected that they will have the same probabilities. Our system model does not favor having any specific number of new components in the steady state.

Moreover, the probability of being in system state (2,0) is less than or equal to the probability of being in state (1,0). This comparison can be interpreted to mean that the system is less likely to be in a better condition. This is also an expected result because even if the system reaches the (2,0) state, it needs to visit (1,0) state before going into deteriorated states.

Then average reward per unit time, r, can be calculated as

$$r = \sum_{j \in S} w(j) \Pi_j. \tag{3.88}$$

Since the aim is to maximize reward of the system, the k value that maximizes the average reward, r, is the desired critical threshold value. In this sense, average reward r is the equivalence of the average expected reward r in the base model. Numerical results of this model and its comparison to base model is provided in Section 4, Numerical Results. Thus, the equivalent finite state model provides the second step of the analytical results which shows the transience of states after optimal switching point, (0,k^*).

3.5. Extensions to the Base Model

After constructing and thoroughly examining the fundamental MDP model, the model is extended by relaxing some assumptions. Two extensions are considered by enlarging the state space and modifying necessary parameters and definitions. The first extension involves increasing the system's deterioration levels, while the second extension involves expanding the number of component types.

3.5.1. Extension 1: Multilevel Dysfunction and Failed State

Specifically, incorporating multiple levels of system condition and deterioration better reflects the complexity of real-world problems. Since the deterioration levels of individual components can vary, it is beneficial to avoid categorizing each system condition into a single confined state. To better represent these variations, an additional layer of deteriorating states can be integrated into the model. Specifically, (-1, k) states are introduced to capture this complexity. These states are treated as representing two distinct cases: one representing a more deteriorated condition than (0, k) states, and the other indicating the inoperable state of the system.

In the first case, (-1,k) states denote more deteriorated but still functioning condition of the system. These states serve as boundary states, with the same parameter and action definitions applied to this new layer. State space of the extended model is given as

$$S = \{(2,0),(1,0),(0,0),(0,1),(0,2),....,(0,\infty),(-1,0),(-1,1),(-1,2),...,(-1,\infty)\}. \tag{3.89}$$

However, the rewards and costs associated with actions are redefined accordingly. Since $(-1,k)$ states indicate poorer system conditions, a replacement action taken in one of these states yields a higher reward compared to the same action in $(0,k)$ states yields. Additionally, the cost of choosing a waiting action is greater in $(-1,k)$ states than in $(0,k)$ states. Reward function of the model is expressed as

$$w(h,k,W) = bck(1-h), \tag{3.90}$$

$$w(h,k,R) = p^k c + (1-p^k)ac - h, \tag{3.91}$$

where b is non-positive number, c is a positive number, a is a positive integer, $h \in \{0,-1\}$ and k is non-negative integer.

State transition diagram of the extended model is shown in Figure 3.3. State transitions from $(-1,k)$ states to $(2,0)$ and $(1,0)$ states following a replacement decision are highlighted with red dashed arcs to enhance the readability of the diagram.

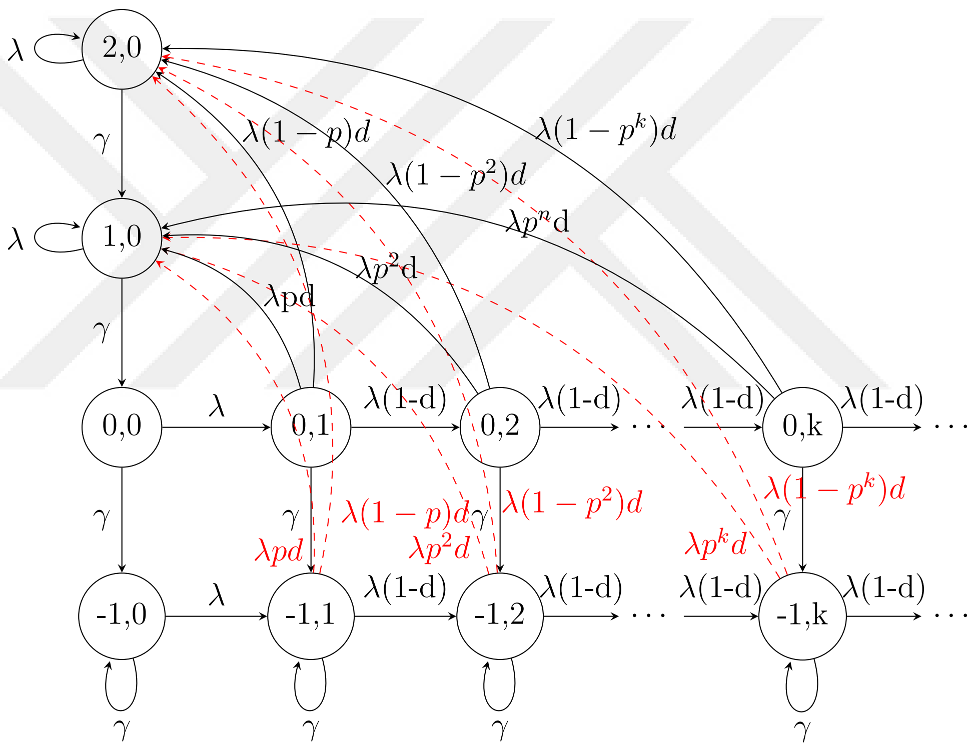

Figure 3.3. Transition Diagram of the MDP Model with Extension 1 Case 1

Probability transition matrix for the first case of Extension 1 is shown in Figure 3.4.

$$
Q' = \begin{array}{c}
\\ (2,0) \\ (1,0) \\ (0,0) \\ (0,1) \\ (0,2) \\ \text{...} \\ (0,k) \\ \text{...} \\ (-1,0) \\ (-1,1) \\ (-1,2) \\ \text{...} \\ (-1,k) \\ \text{...}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccccccccc}
(2,0) & (1,0) & (0,0) & (0,1) & (0,2) & \text{...} & (0,k) & \text{...} & (-1,0) & (-1,1) & (-1,2) & \text{...} & (-1,k) & \text{...}
\end{array} \\
\left[\begin{array}{cccccccccccccc}
\frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
\frac{(1-p)d\lambda}{\Lambda} & \frac{pd\lambda}{\Lambda} & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
\frac{(1-p)^2)d\lambda}{\Lambda} & \frac{p^2d\lambda}{\Lambda} & 0 & 0 & \frac{\gamma}{\Lambda} & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
\frac{(1-p)^k)d\lambda}{\Lambda} & \frac{p^kd\lambda}{\Lambda} & 0 & 0 & 0 & \cdots & \frac{\gamma}{\Lambda} & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \cdots & 0 & \cdots \\
\frac{(1-p)d\lambda}{\Lambda} & \frac{pd\lambda}{\Lambda} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \frac{\lambda}{\Lambda} & \cdots & 0 & \cdots \\
\frac{(1-p)^2)d\lambda}{\Lambda} & \frac{p^2d\lambda}{\Lambda} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \frac{\lambda}{\Lambda} & \cdots & 0 & \cdots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
\frac{(1-p)^k)d\lambda}{\Lambda} & \frac{p^kd\lambda}{\Lambda} & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & \frac{\gamma}{\Lambda} & \cdots \\
\vdots & & & & & \ddots & & & & & & & \vdots &
\end{array}\right]
\end{array}
$$

Figure 3.4. Probability Transition Matrix of Extension 1 Case 1

Given the worse condition and the higher associated rewards and costs, $(-1, k)$ states are expected to have lower optimal control limits compared to $(0, k)$ states. Additionally, with the introduction of multiple deterioration levels, it becomes possible to determine an optimal control limit in terms of h. Consequently, the optimal control limit points become two-dimensional, represented as (k_1^*, k_2^*), when two deterioration levels are present.

In the second case, the $(-1, k)$ states denote the system is inoperable rather than being deteriorated and low functioning. An external policy is embedded to the system which dictates the replacement action when the system is in one of the inoperable states with positive k values. Whenever the system transitions into $(-1, k)$ states with $k>0$, the component is immediately replaced by a spare part and waiting for another time period is not an option anymore. State space and reward functions are same as the first case given above. However, there are no action regarding $(-1, k)$ states. State transition diagram of the second case is shown in Figure 3.5. Transitions from $(-1, k)$ states are again shown with red dashed lines. Probability transition matrix for the second case of Extension 1 is shown in Figure 3.6.

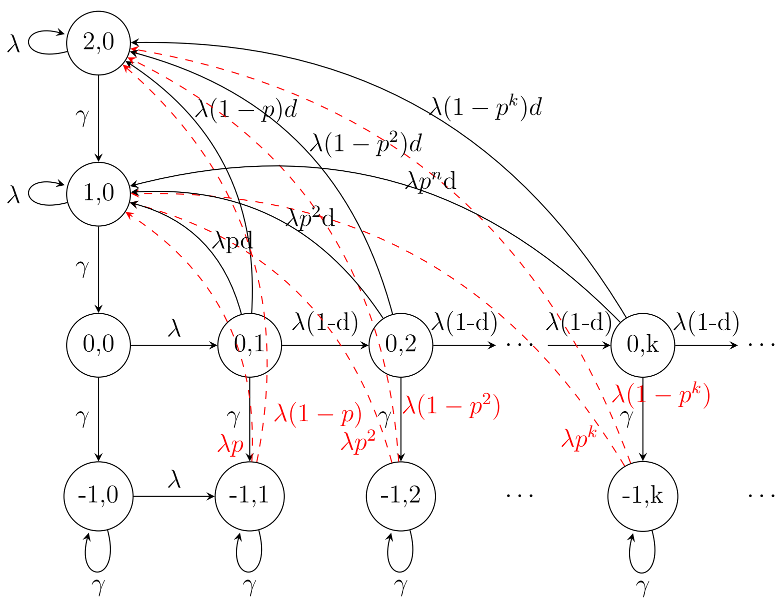

Figure 3.5. Transition Diagram of the MDP Model with Extension 1 Case 2

$$
Q'' = \begin{array}{cc}
 & \begin{array}{ccccccccccccccc} (2,0) & (1,0) & (0,0) & (0,1) & (0,2) & \text{...} & (0,k) & \text{...} & (-1,0) & (-1,1) & (-1,2) & \text{...} & (-1,k) & \text{...} \end{array} \\
\begin{array}{c} (2,0) \\ (1,0) \\ (0,0) \\ (0,1) \\ (0,2) \\ \text{...} \\ (0,k) \\ \text{...} \\ (-1,0) \\ (-1,1) \\ (-1,2) \\ \text{...} \\ (-1,k) \\ \text{...} \end{array} &
\left[\begin{array}{cccccccccccccc}
\frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\frac{(1-p)d\lambda}{\Lambda} & \frac{pd\lambda}{\Lambda} & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\frac{(1-p)^2d\lambda}{\Lambda} & \frac{p^2d\lambda}{\Lambda} & 0 & 0 & \frac{\gamma}{\Lambda} & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
\frac{(1-p)^k)d\lambda}{\Lambda} & \frac{p^kd\lambda}{\Lambda} & 0 & 0 & 0 & \dots & \frac{\gamma}{\Lambda} & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \dots & 0 & \dots \\
\frac{1-p\lambda}{\Lambda} & \frac{p\lambda}{\Lambda} & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\frac{(1-p)^2\lambda}{\Lambda} & \frac{p^2\lambda}{\Lambda} & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots \\
\vdots & & & & & \ddots & & & & & & & \vdots & \\
\frac{(1-p)^k)\lambda}{\Lambda} & \frac{p^k\lambda}{\Lambda} & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & \frac{\gamma}{\Lambda} & \dots \\
\vdots & & & & & \ddots & & & & & & & \vdots &
\end{array}\right]
\end{array}
$$

Figure 3.6. Probability Transition Matrix of Extension 1 Case 2

3.5.2. Extension 2: Three Quality Levels for Spare Parts

In the base model, it is assumed that there are only two types of components. Another extension to the model can be increasing the number of different component quality types. This will make the system more complex and change the system behavior. In many real-life problems, the number of component types is not limited to two and often includes multiple types. In this context, increasing the number of component types makes the model more realistic. Hence, this extension is interesting and worth to consider.

For the extended model, it is assumed that there are three types of components, low, moderate and high quality, instead of two. A new state $(3,0)$ is added to the model to account for this new type of component, Type 3. New state space is given in as

$$S = \{(3,0),(2,0),(1,0),(0,0),(0,1),(0,2),....,(0,\infty),(-1,0),(-1,1),(-1,2)\\ ,...,(-1,\infty)\}. \quad (3.92)$$

Accordingly, two probability parameters, p_1 and p_2, are required. Incoming components are Type 1 component with probability p_1, Type 2 with probability p_2 and Type 3 with probability 1-(p_1+p_2) transitioning the system $(1,0)$, $(2,0)$ and $(3,0)$, respectively.In this context, the state $(3,0)$ represents the optimal functioning level of the system. The system works for all three states but its performance progressively deteriorates as the value of h decreases from 3 to 1. Hence, the reward of the replace action is redefined by accounting the new component type. If the system transitions into $(3,0)$ state, then it yields a reward of ec units where e is a positive integer which denotes the reward coefficient of state $(3,0)$. Reward function is expressed as

$$w(h,k,W) = bck(1-h), \quad (3.93)$$

$$w(h,k,R) = p_1^k c + ((p_1+p_2)^k - p_1^k)ac + (1-(p_1+p_2)^k)ec - h. \quad (3.94)$$

The replacement procedure remains the same, selecting the most qualified available component whenever a replacement decision is made. State transition diagram of the model is shown in figure 3.7. Blue dashed arcs are used to indicate the additional transitions introduced in the diagram for Extension 2.

Transition rates of the states which actions are allowed are not shown in the diagram

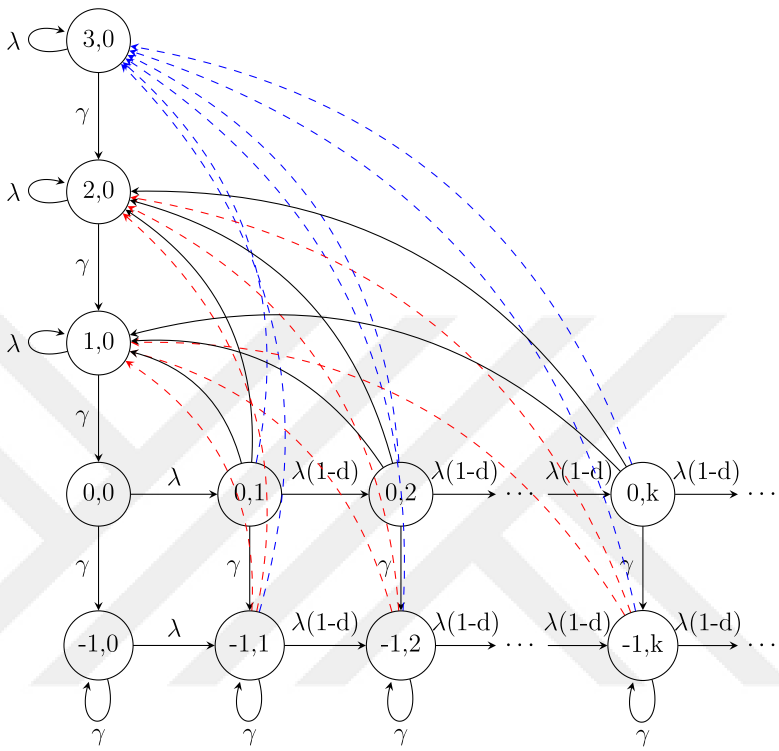

Figure 3.7. Transition Diagram of the MDP Model with Extension 2

due to readability issues. Transition probabilities for this set of states is shown in the transition probability matrix in Figure 3.8. In this case, optimal policies are again sensitive to reward and cost parameters. Numerical results of this extended model are provided in section 4.

In summary, the model formulation presented here captures the complexity of the system by incorporating system functioning levels and number of available components.The base model provides a simple but affective foundational structure, while the extended model enhances realism by introducing additional deterioration states and more component quality types. Moreover, finite state Markov Chain model provides an alternative structure to analyze the system from a different perspective. As a result, the formulation allows for optimal control limit type policies and optimizes decision-making for replacement problems.

$$
Q''' = \begin{array}{cc}
 & \begin{array}{cccccccccccccccc} (3,0) & (2,0) & (1,0) & (0,0) & (0,1) & (0,2) & \ldots & (0,k) & \ldots & (-1,0) & (-1,1) & (-1,2) & \ldots & (-1,k) & \ldots \end{array} \\
\begin{array}{c} (3,0) \\ (2,0) \\ (1,0) \\ (0,0) \\ (0,1) \\ (0,2) \\ \ldots \\ (0,k) \\ \ldots \\ (-1,0) \\ (-1,1) \\ (-1,2) \\ \ldots \\ (-1,k) \\ \ldots \end{array} &
\begin{bmatrix}
\frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
0 & 0 & \frac{\lambda}{\Lambda} & \frac{\gamma}{\Lambda} & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
0 & 0 & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
\frac{(1-p_1-p_2)d\lambda}{\Lambda} & \frac{(p_2)d\lambda}{\Lambda} & \frac{p_1 d\lambda}{\Lambda} & 0 & \frac{\gamma}{\Lambda} & \frac{\lambda(1-d)}{\Lambda} & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
\frac{(1-(p_1+p_2)^2)d\lambda}{\Lambda} & \frac{((p_1+p_2)^2-p_1^2)d\lambda}{\Lambda} & \frac{p_1^2 d\lambda}{\Lambda} & 0 & 0 & \frac{\gamma}{\Lambda} & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
\vdots & & & & & \ddots & & & & \ldots & 0 & 0 & \ldots & 0 & \ldots \\
\frac{(1-(p_1+p_2)^k)d\lambda}{\Lambda} & \frac{((p_1+p_2)^k-p_1^k)d\lambda}{\Lambda} & \frac{p_1^k d\lambda}{\Lambda} & 0 & 0 & 0 & \ldots & \frac{\gamma}{\Lambda} & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots \\
\vdots & & & & & \ddots & & & & \ldots & 0 & 0 & \ldots & 0 & \ldots \\
0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & \ldots & \frac{\gamma}{\Lambda} & \frac{\lambda}{\Lambda} & 0 & \ldots & 0 & \ldots \\
\frac{(1-p_1-p_2)d\lambda}{\Lambda} & \frac{(p_2)d\lambda}{\Lambda} & \frac{p_1 d\lambda}{\Lambda} & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & \frac{\lambda}{\Lambda} & \ldots & 0 & \ldots \\
\frac{(1-(p_1+p_2)^2)d\lambda}{\Lambda} & \frac{((p_1+p_2)^2-p_1^2)d\lambda}{\Lambda} & \frac{p_1^2 d\lambda}{\Lambda} & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & \frac{\lambda}{\Lambda} & \ldots & 0 & \ldots \\
\vdots & & & & & \ddots & & & & \ldots & 0 & 0 & \ldots & 0 & \ldots \\
\frac{(1-(p_1+p_2)^k)d\lambda}{\Lambda} & \frac{((p_1+p_2)^k-p_1^k)d\lambda}{\Lambda} & \frac{p_1^k d\lambda}{\Lambda} & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & \frac{\gamma}{\Lambda} & \ldots \\
\vdots & & & & & \ddots & & & & \ldots & 0 & 0 & \ldots & 0 & \ldots
\end{bmatrix}
\end{array}
$$

Figure 3.8. Probability Transition Matrix of Extension 2

4. NUMERICAL ANALYSIS

The primary aim of this study is to analyze a replacement problem and to identify, if possible, an optimal replacement policy. The model is also analyzed numerically to observe its behavior under different input configurations. Since the problem is formulated as a Markov Decision Process using Bellman's value equations, solving these equations is necessary to derive policy-based solutions. The model is simplified further to obtain plausible numerical results and ease solution of the value equations for policy computations. The base model is constructed as an MDP with infinite number of states. Since analyzing a Markov Chain with an infinite number of states is challenging, an upper bound for the number of states has been set which can also be expressed as an upper bound for the number of new parts that is set to 40. This limit is assumed to be sufficiently high to capture the dynamics of an infinite-state Markov Chain, given the structure of the problem. Observations indicate that the system rarely reaches this maximum component count due to the defined rewards and costs. Thus, it is reasonable to use an MDP model with a large, but finite, number of states. Afterwards, the MDP model is approximated for implementing the Value Iteration algorithm [19] to find optimal policies. The value iteration algorithm is employed in this model because it is particularly well-suited for solving Markov Decision Processes (MDP) with a finite state space. This algorithm iteratively improves the value function estimates for each state by applying Bellman's optimality principle until convergence. By doing so, it ensures that the optimal policy can be derived by identifying the actions that maximize the expected cumulative reward in each state. In cases where there is a tie between actions, the algorithm prioritizes the replace action over the wait action. When truncating MDP model for Value Iteration algorithm, it is also tested whether it is a good approximation by looking at expected reward, r, values for increasing k values. It is observed that r values approximates to a constant value which shows the validity and reliability of the approximation. Additionally, value iteration is computationally efficient and reaches convergence quickly, making it ideal for this replacement problem where we aim to identify optimal decision policies. The pseudo-code of the value iteration algorithm is shown in Figure 4.1.

Algorithm: Value Iteration

1. Select $v^0 \in$ V, specify $\epsilon >$ o and set n=0.

2. For each s $\in$ S, compute v^{n+1}(s) by

$$v^{n+1}(s) = \max_{a \in A_s} \{w(s,a) + \sum_{j \in S} p(j|s,a) v^n(j)\}$$

3. Define

$$\delta^n = v^{n+1} - v^n.$$

If

$$sp(\delta^{n+1} - \delta^n) < \epsilon,$$

go to step 4. Otherwise increment n by 1 and return to step 2.

4. For each s $\in$ S, choose

$$d_\epsilon(s) \in \arg\max_{a \in A_s} \{w(s,a) + \sum_{j \in S} p(j|s,a) v^n(j)\}$$

and stop.

Figure 4.1. Value Iteration Algorithm

The algorithm is based on maximizing the average reward and keeps track of the change in value functions and the difference in changes at every iteration, which is denoted by δ. The stopping condition is to have a negligibly small difference between these changes. The algorithm is implemented in Python and related results are obtained for different parameters. The Python code is given in Appendix A.

In addition to the Value Iteration algorithm, other methods demonstrated to be mathematically rigorous in the previous section are also utilized to determine the optimal threshold point. One of them is to use Bellman's theorem given in equation 3.44. The expected average reward, r, in the long run and optimal policy of the system can be obtained with the aid of this theorem. Since the theorem addresses the steady-state behavior of the system, it serves as the foundation for the finite-state Markov chain model and also provides validation for this model. Following, the finite MC model is utilized as a third computation method for optimal threshold point.

While obtaining optimal policies, three different set of values are used for parameters of reward coefficient, a, arrival rate, λ, and cost coefficient, b. Type 1 probability, p, parameter takes values between 0 and 1 with 0.1 increments. Deterioration rate, γ, and unit reward, c, are fixed to 1 and remaining parameters are structured as coefficients of these unit parameters.

- Reward coefficient(a): 2, 8, 20
- Arrival rate(λ): 1, 5, 10
- Cost coefficient (b): 0, -0.1/k, -0.1

The parameter values are chosen in a way to reflect the potential behavior of the system under low, medium and high values of the parameters a and λ. The values for parameter b is chosen to reflect no waiting cost, constant waiting cost and increasing waiting cost as m increases. Each combination of these parameter values denotes another possible scenario for the system. Hence, different optimal policies are attained for each distinct scenario. The aim is to conduct a sensitivity analysis which allows to observe how changes impact the system outcomes by varying each parameter over a reasonable range. The reason of choosing three values per parameters keeps the total number of scenarios manageable for numeric analysis which allows sufficient exploration without overwhelming computational resources. Parameter sets of three different values are shown in Table 4.1.

Table 4.1. Parameters

	1	**2**	**3**
a	2	8	20
λ	1	5	10
b	0	-0.1/k	-0.1

Accordingly, an experiment set with total 27 different experiments is obtained. We aim to observe the changes in optimal decisions with respect to different parameters. Hence, it is desired to demonstrate the relationship between each parameter and the optimal critical point.

4.1. Analysis of Finite MC

Finite state Markov Chain model is constructed to propose an easier alternative way to calculate optimal critical point for number of components in the system. This model is described in section 3.3. Steady state distributions of the system is utilized for optimal critical point calculations. After stationary distribution of each state is found, the average reward of the system is calculated by using reward function and steady state distributions. Then, the k value which maximizes the average reward functions is determined as the desired threshold point, k^*. The algorithm which is implemented to find k^* is given in Figure 4.2. Numerical results show the concavity of the average reward functions, r. Three different experiments are selected as representatives to show the behavior and concavity property of r functions for each probability value p. The graphs of r functions resulting from three different experiments are shown in Figures 4.3, 4.4 and 4.5. The optimal critical point of each plot is shown with red points. An increasing trend in the optimal control points is observed as the probability values rise from 0.1 to 0.9 in scenarios that incorporate waiting costs.

Algorithm: Optimal Critical Point for Finite Markov Chain Model

1. Set $k_{optimal}$=None and r_{max}=∞.

2. For each k $\in$ (1,40),

compute reward matrix $w(k)$

compute stationary distribution Π(k) by solving ΠP= Π

compute expected reward r_k= $w(k)\Pi$(k).

3. If

$$r_k > r_{max},$$

Update

$$k_{optimal} = k,$$

$$r_{max} = r_k.$$

4. Return $k_{optimal}$ and r_{max}.

Figure 4.2. The Algorithm for Finding Optimal Critical Point for Finite MC Model

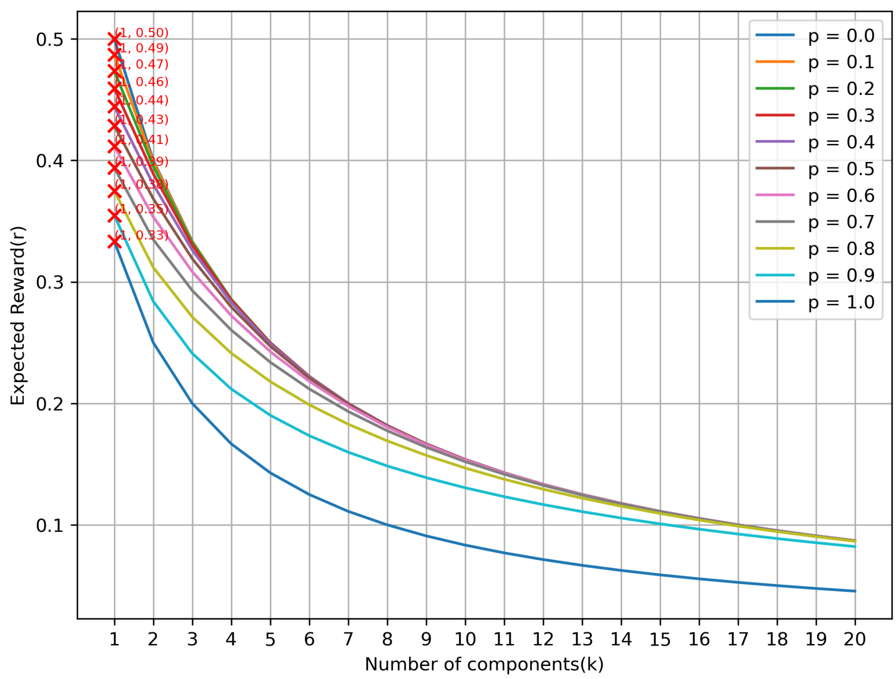

Figure 4.3. Optimal Control Points for a=2, b=0 and λ=1

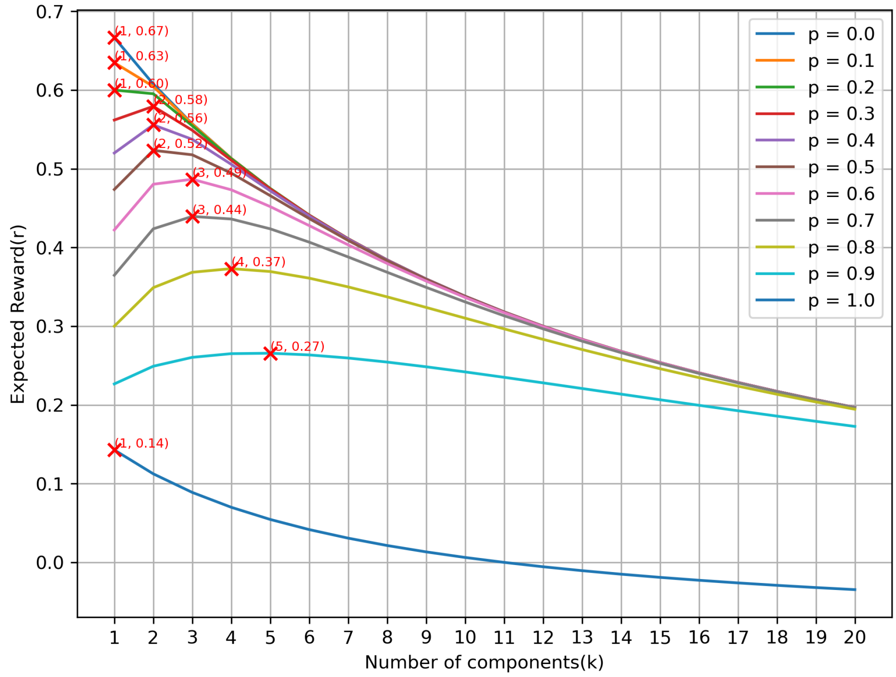

Figure 4.4. Optimal Control Points for a=8, b=-0.1/k and λ=5

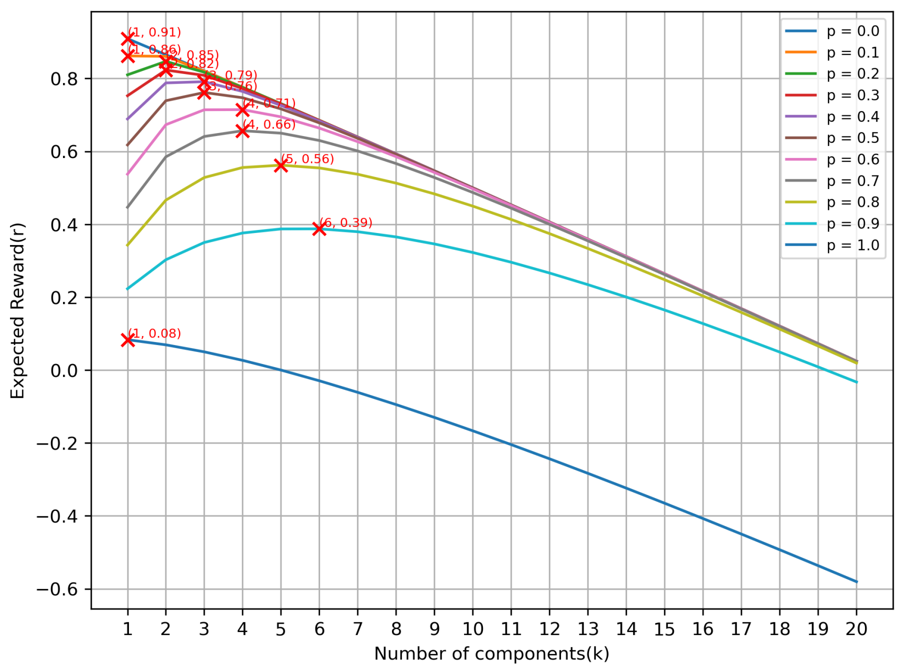

Figure 4.5. Optimal Control Points for a=20, b=-0.1 and λ=10

4.2. Comparison of Different Methods

Then, the validity of the finite state Markov Chain model is considered. The average reward values and optimal control points of three alternative models are compared with each other to check their consistency and justify their calculation methods. For this purpose, average expected reward of the system is calculated by using Value Iteration algorithm, Bellman's theorem and the steady state distributions of finite state Markov Chain. For Value Iteration algorithm, the difference between value functions of state (0,40) in the long-run is considered as the average expected reward of the system. Since the system never requires waiting for 40 components under the selected parameter values, the value equations for the state (0,40) reach a steady state more quickly compared to other states. It is also observed that r values also approximates to same constant for higher values than 40. Thus, it is suitable to use state (0,40) for calculation of expected reward. Thus, average reward and optimal threshold points by using three distinct methods are shown in Tables 4.2, 4.3 and 4.4 for three different experiments selected. The expected reward for threshold search algorithm is calculated by using Bellman's theorem.

Table 4.2. Expected Rewards and Corresponding Optimal Critical Point for a=2, b=0 and λ=1

	Finite MC		Threshold Search Algorithm		Value Iteration	
p	**Expected Reward**	k^*	**Expected Reward**	k^*	**Expected Reward**	k^*
0	0.5	1	0.5	1	0.499	1
0.1	0.487	1	0.487	1	0.487	1
0.2	0.474	1	0.474	1	0.473	1
0.3	0.459	1	0.46	1	0.459	1
0.4	0.444	1	0.444	1	0.445	1
0.5	0.428	1	0.429	1	0.429	1
0.6	0.412	1	0.412	1	0.412	1
0.7	0.394	1	0.394	1	0.394	1
0.8	0.375	1	0.375	1	0.375	1
0.9	0.355	1	0.355	1	0.354	1
1	0.333	1	0.333	1	0.333	1

Table 4.3. Expected Rewards and Corresponding Optimal Critical Point for a=8, b=-0.1/k and λ=5

	Finite MC		Threshold Search Algorithm		Value Iteration	
p	**Expected Reward**	k^*	**Expected Reward**	k^*	**Expected Reward**	k^*
0	0.667	1	0.667	1	0.667	1
0.1	0.635	1	0.635	1	0.635	1
0.2	0.6	1	0.6	1	0.6	1
0.3	0.579	2	0.58	2	0.579	2
0.4	0.556	2	0.556	2	0.556	2
0.5	0.523	2	0.524	2	0.524	2
0.6	0.487	3	0.487	1	0.486	3
0.7	0.439	3	0.44	1	0.44	3
0.8	0.373	4	0.373	1	0.373	4
0.9	0.266	5	0.266	5	0.266	5
1	0.143	1	0.143	1	0.143	1

Table 4.4. Expected Rewards and Corresponding Optimal Critical Point for a=20, b=-0.1 and λ=10

	Finite MC		Threshold Search Algorithm		Value Iteration	
p	**Expected Reward**	k^*	**Expected Reward**	k^*	**Expected Reward**	k^*
0	0.909	1	0.91	1	0.909	1
0.1	0.862	1	0.863	1	0.861	1
0.2	0.847	2	0.848	2	0.846	2
0.3	0.823	2	0.824	2	0.823	2
0.4	0.791	3	0.792	3	0.791	3
0.5	0.762	3	0.763	3	0.761	3
0.6	0.714	4	0.716	4	0.714	4
0.7	0.656	4	0.658	4	0.656	4
0.8	0.562	5	0.562	5	0.562	5
0.9	0.387	6	0.389	6	0.387	6
1	0.083	1	0.083	1	0.083	1

The numeric results, rounded to three decimal places, demonstrate that the optimal control limit points obtained from the three methods are identical. Furthermore, the expected rewards are highly comparable, differing by only a negligible amount of up to 0.002 in some cases. These findings suggest that the finite-state Markov chain model serves as an equivalent representation of the Markov decision process (MDP) model. Given that the analytical results confirm the existence of an optimal control limit-type policy, the system's optimal decision policy can be straightforwardly determined once the critical threshold is identified using the finite-state Markov chain model. This observation underscores the feasibility and validity of representing an MDP model with a finite-state Markov chain.

After r values are calculated by using any of the given three methods above, it is possible to use the equation 3.69 to find optimal threshold point, k^* which provides easier computation than Value Iteration algorithm. The pseudo-code of the proposed threshold search algorithm is given in figure 4.6.

Algorithm: Threshold Search

Input: Parameters $\gamma, b, c, a, p, \lambda, r$

Output: k^* (smallest integer satisfying the condition)

1. Initialize $k = 0$.

2. Compute LHS by

$$LHS = \frac{\gamma(bck - cp^k(1-a)(1-p))}{\lambda p^k(1-p) + \gamma}$$

3. If

$$LHS \leq r$$

stop and return k. Otherwise increment k by 1 and return to step 2.

Figure 4.6. Threshold Search Algorithm

4.3. The Existence of Control Limit Type Policies

As it is stated in the previous section, the Markov model and its structural properties guarantee a control limit type optimal policy for the given replacement problem. When the numerical results of the model is examined after the implementation of the Value Iteration algorithm, the existence of the control limit type policies can be clearly seen as it is expected in the analytical results. Hence, optimal critical thresholds for number of parts, denoted by k^*, can be deduced from resulting policies.

First, $\mathbf{\Delta} V_n(0,k)$ function is studied since its structure is paramount for the existence of optimal control limit-type policies. Numerical experiments provide insight into the behavior of this function, supporting the claims made regarding its properties. In particular, the non-increasing nature of $\mathbf{\Delta} Vn(0,k)$ is evident from the experimental results. The graphs of this function, presented in Figures 4.7 to 4.9, illustrate its monotonic behavior for selected parameter sets, representing three cases from the overall experiment set.

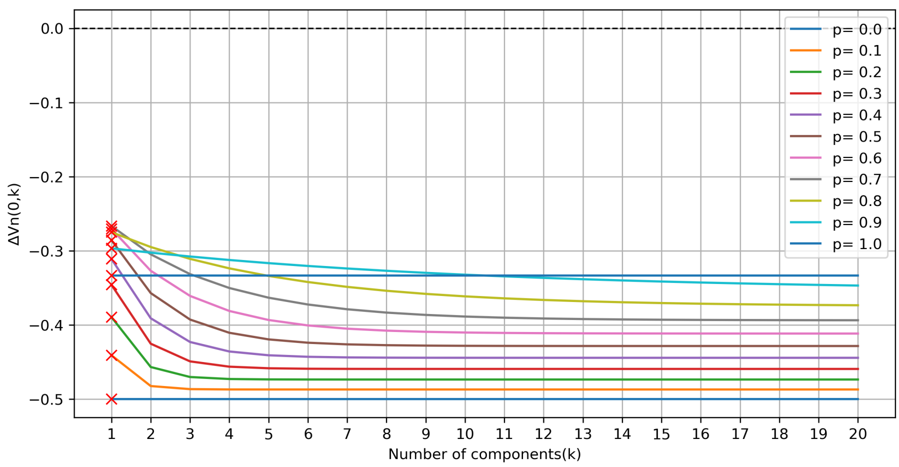

Figure 4.7. $\mathbf{\Delta} V_n(0,k)$ function and corresponding k^* values for a=2, λ=1 and b=0

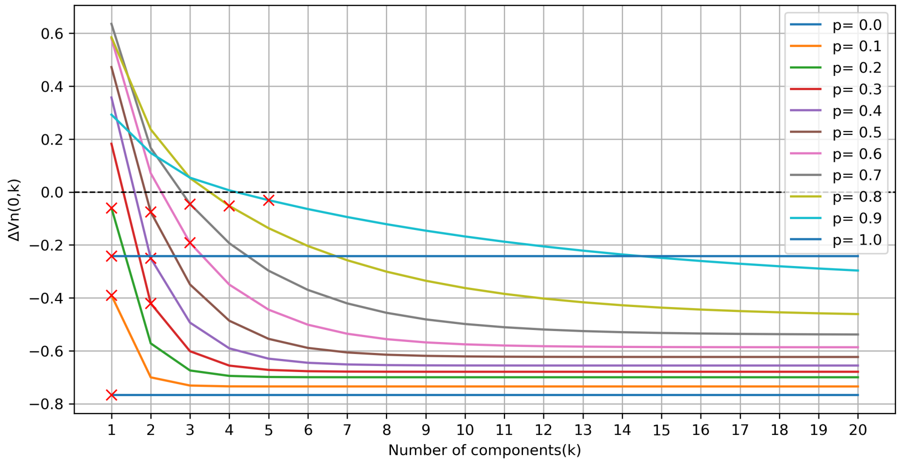

Figure 4.8. $\boldsymbol{\Delta}V_n(0,k)$ function and corresponding k^* values for a=8, λ=5 and b=-0.1/k

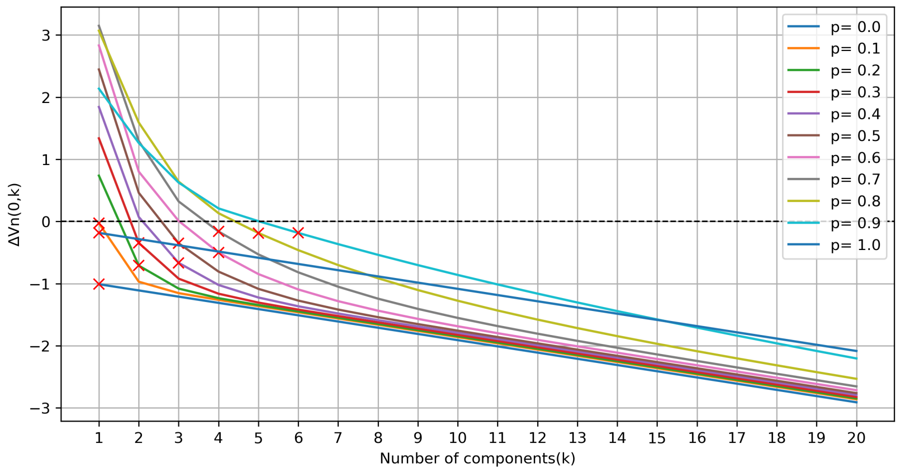

Figure 4.9. $\boldsymbol{\Delta}V_n(0,k)$ function and corresponding k^* values for a=20, λ=10 and b=-0.1

Figures 4.7 - 4.9 include eleven different plots. Each plot presents the $\boldsymbol{\Delta}V_n(0,k)$ function for different Type 1 probability values between 0 and 1. Since negativity of the function denotes that replace action is better than wait action, the k value where the function becomes non-positive first time denotes the optimal critical point. Optimal k^* value of each plot is shown with red dots on graphs. If the waiting cost does not depend on number of components,n, where the cost is either zero or constant, it can be observed that the function approximates to a constant value for higher number of components. Otherwise, $\mathbf{\Delta}V_n(0,k)$ function does not approximate to a specific value for increasing k and can take values from a larger range. Hence, it can be concluded that numerical results verify the analytical results that state the existence of an optimal control limit type policy. It is seen that the function does not increase in numerical experiments which shows a threshold policy for the system. When the optimal action becomes replacement once, it does not change for the states with higher number of components and become wait again.

These policies are of the control limit type and determine threshold values for the number of new parts and also the system condition in the extended version of the model. The optimal action changes exactly at the threshold value and remains the same above this threshold when the decision is changed once at this critical point. This control-limit policy structure simplifies decision-making, as it provides clear action boundaries and reduces the need for frequent recalculations, making it efficient for practical implementation in dynamic, resource-limited environments.

4.4. Sensitivity Analysis of Parameters

Observing the system's performance under varying parameters is essential to understand its sensitivity and adaptability. Therefore, a numerical analysis was conducted to examine how changes in parameters affect the system, with a particular focus on their influence on optimal policies. This analysis provides insights into the robustness and responsiveness of the system's optimal control strategies under different parameter configurations.

4.4.1. The Effect of Type 1 Probability on Optimal Decisions

The policies are determined for probabilities ranging between 0 and 1 with increments 0.1, allowing for an examination of how incremental improvements in probability affect system behavior in each case. Since the randomness in the system arises from this probabilistic feature of the components, understanding its effect on the resulting optimal policy patterns is crucial. Table 4.5 shows the resulted optimal policies and corresponding limit points for different probability values.The analysis focuses on states ranging from (0,1) to (0,40), where decision-making is possible through action selection in these states. As it is stated before, decisions 0 and 1 denote wait and replace actions, respectively. For each row in Table 4.5, the index of the first occurrence of value 1 in the array determines the optimal control point, k^*, of the defined system.

Table 4.5. Optimal Policies and Critical Points where $\lambda = 10$, $a = 8$, b = -0.1/k, c=1, and $\gamma = 1$

p	k^*	**Policy**
1	1	**1**111111111111111111111111111111111111
0.9	7	000000**1**111111111111111111111111111111
0.8	5	0000**1**11111111111111111111111111111111
0.7	4	000**1**111111111111111111111111111111111
0.6	3	00**1**1111111111111111111111111111111111
0.5	3	00**1**1111111111111111111111111111111111
0.4	2	0**1**11111111111111111111111111111111111
0.3	2	0**1**11111111111111111111111111111111111
0.2	2	0**1**11111111111111111111111111111111111
0.1	1	**1**111111111111111111111111111111111111
0	1	**1**111111111111111111111111111111111111

The following Figure 4.10 shows the change of critical points, k^*, with respect to different probability values, p, graphically which correspond to optimal policies given in Table 4.5. When examining the results, an increasing trend is observed between probability values of 0.1 and 0.9. Here, $p = 1$ and $p = 0$ indicate that all arriving components are exclusively of Type 1 and Type 2, respectively. Consequently, probability values at 0 and 1 eliminate the probabilistic nature of the problem, effectively transforming it into a replacement problem involving only a single component type. This renders our structured model less meaningful, as the deterministic nature of component type and

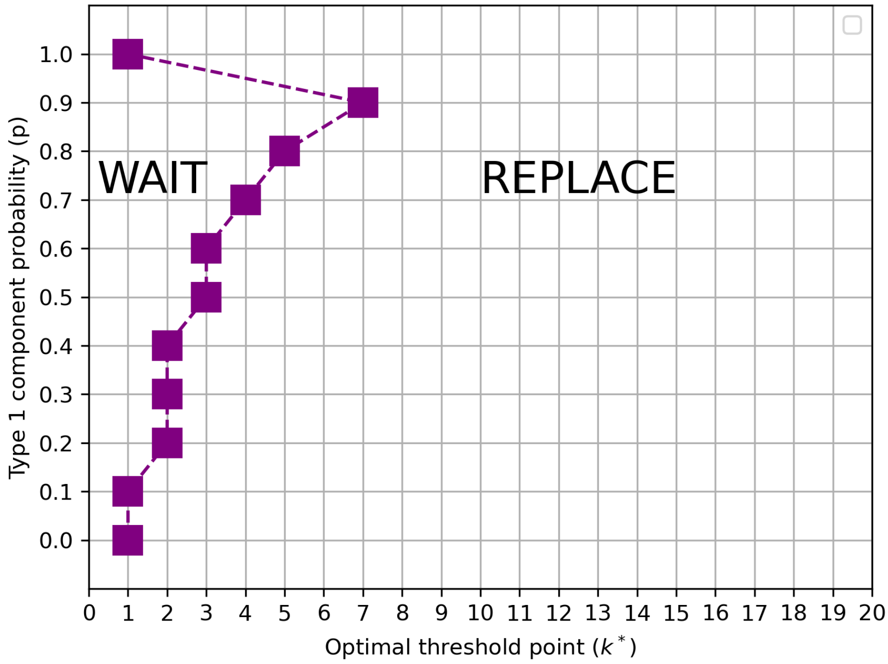

Figure 4.10. Optimal Control Points for Different p Values

replacement outcome leads the system to prefer immediate replacement with the first arrival, removing the necessity to wait for additional components. Hence, the critical point is 1 and wait decision is never optimal in any state for p=0 and p=1 regardless of the values of other parameters. For probability values of p between 0 and 1, the likelihood of receiving a Type 1 component increases with p , while the probability of receiving a Type 2 component, represented by 1 - p , decreases accordingly. Given that Type 2 components have the potential to transition the system to a more favorable state and yield higher expected rewards, the system is inclined to wait for additional arrivals when p is high, as anticipated. When Type 2 components are scarce, the system attempts to offset this scarcity by delaying the replacement decision until more components have arrived. Furthermore, increases in the critical threshold occur more sharply at higher probability values, particularly when p exceeds 0.6. This suggests a heightened sensitivity in the system's replacement decision-making as the likelihood of obtaining a Type 1 component continues to rise.

4.4.2. The Effect of Arrival Rate on Optimal Control Points

The effect of the arrival rate on the optimal critical point is analyzed for 9 different experiments and each probability value. Optimal decisions are obtained and shown in Figures 4.11- 4.17.

The markers in each graph indicate the optimal critical points, k^*. Each figure

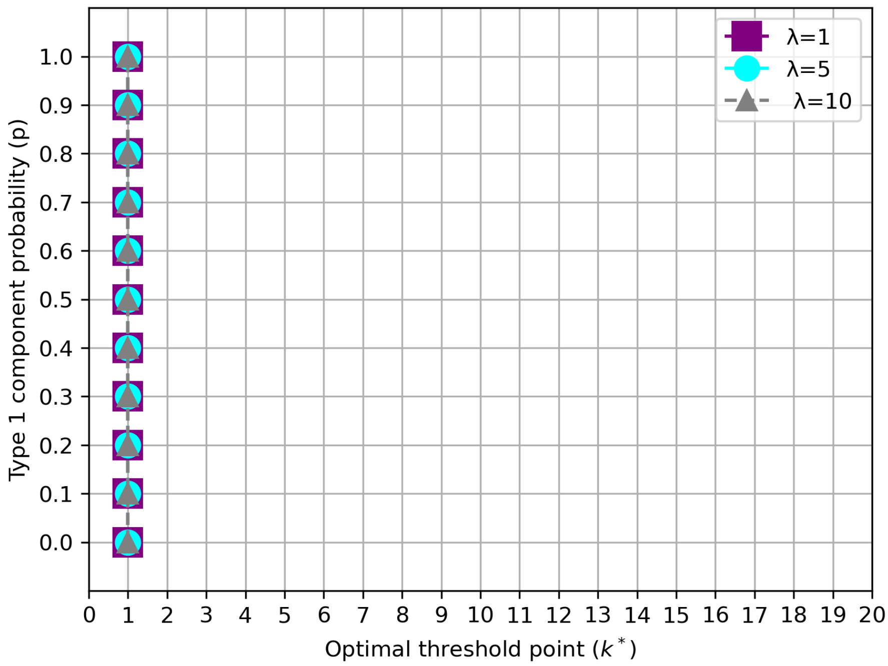

Figure 4.11. Optimal Control Points for a=2 and b=(0,-0.1/k, -0.1)

contains three separate plots, representing three different values of λ. In each plot, the left-hand side of the plot corresponds to the wait decision, while the right-hand side corresponds to the replace decision across all cases. The results indicate that an increase in the arrival rate, λ, leads to a higher optimal critical point value. This effect is particularly pronounced at higher probability values, where the differences between optimal policies for various λ values become more noticeable. In contrast, at lower probability values, the optimal policies for different λ values tend to be more similar, indicating a reduced sensitivity of the critical point to λ value in this range.

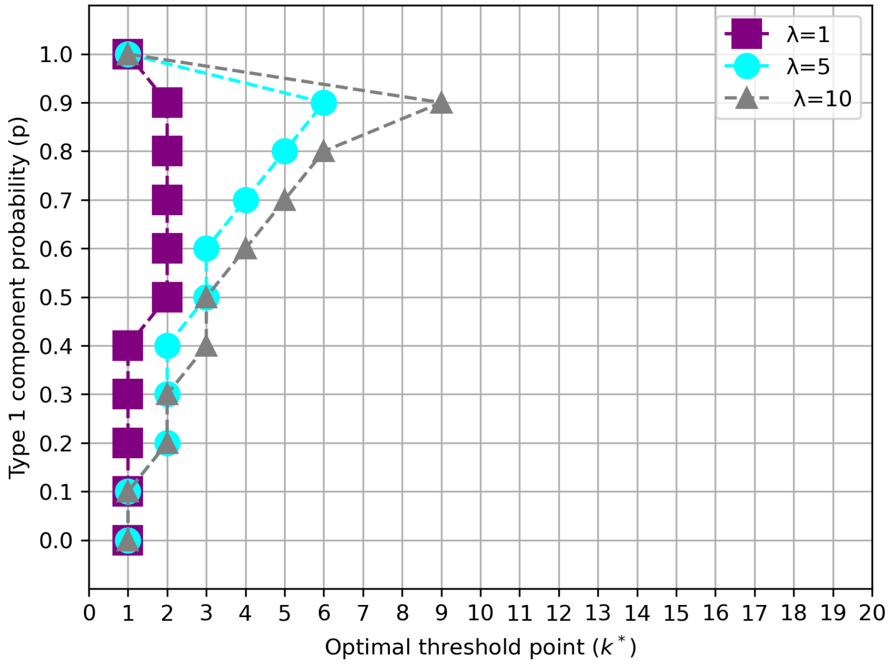

Figure 4.12. Optimal Control Points for a=8 and b=0

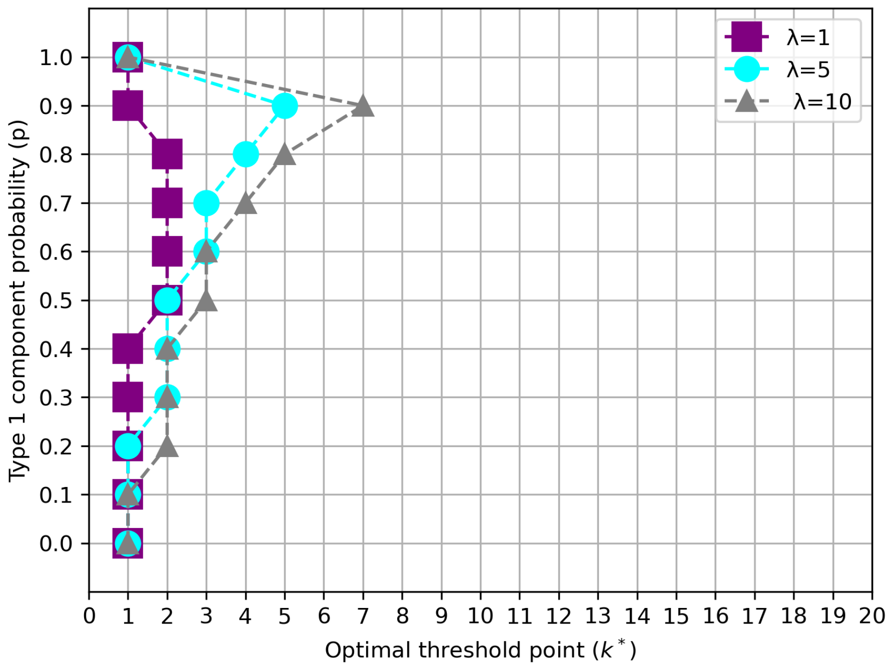

Figure 4.13. Optimal Control Points for a=8 and b=-0.1/k

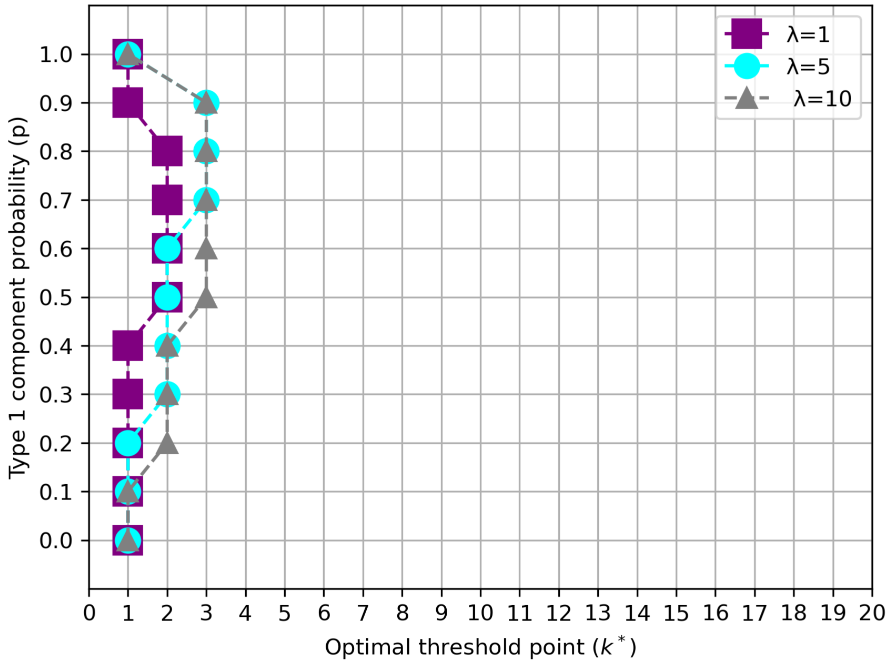

Figure 4.14. Optimal Control Points for a=8 and b=-0.1

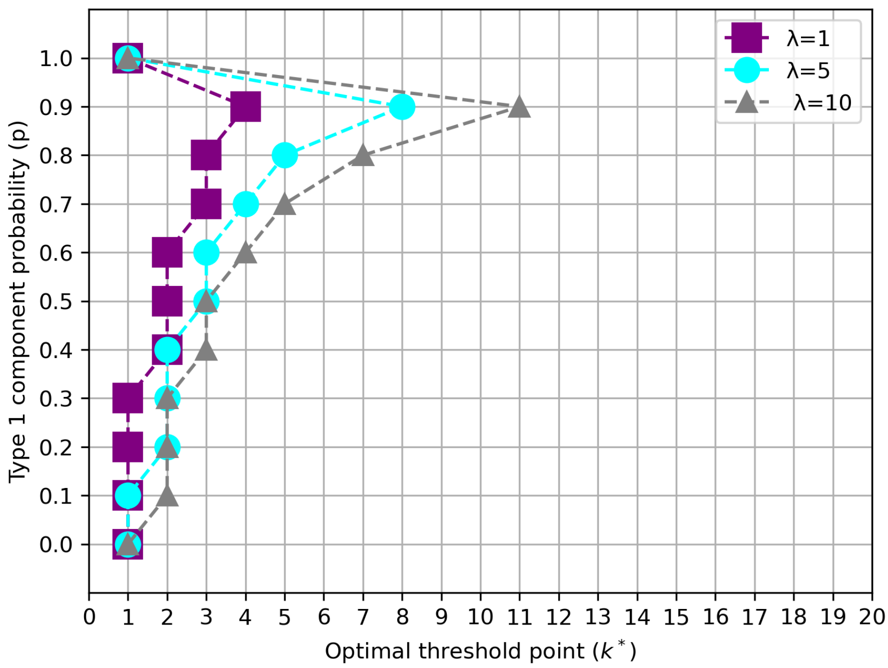

Figure 4.15. Optimal Control Points for a=20 and b=0

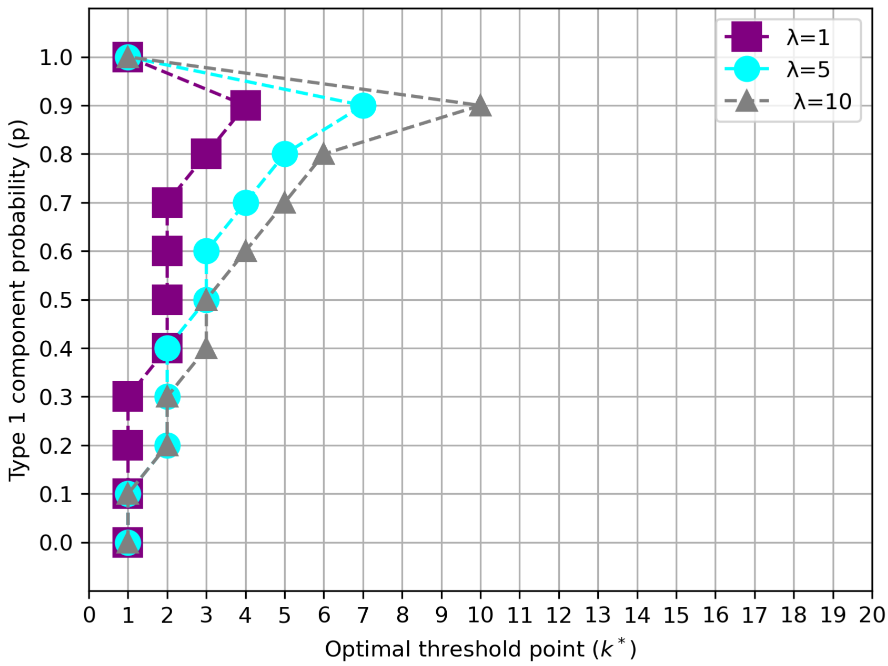

Figure 4.16. Optimal Control Points for a=20 and b=-0.1/k

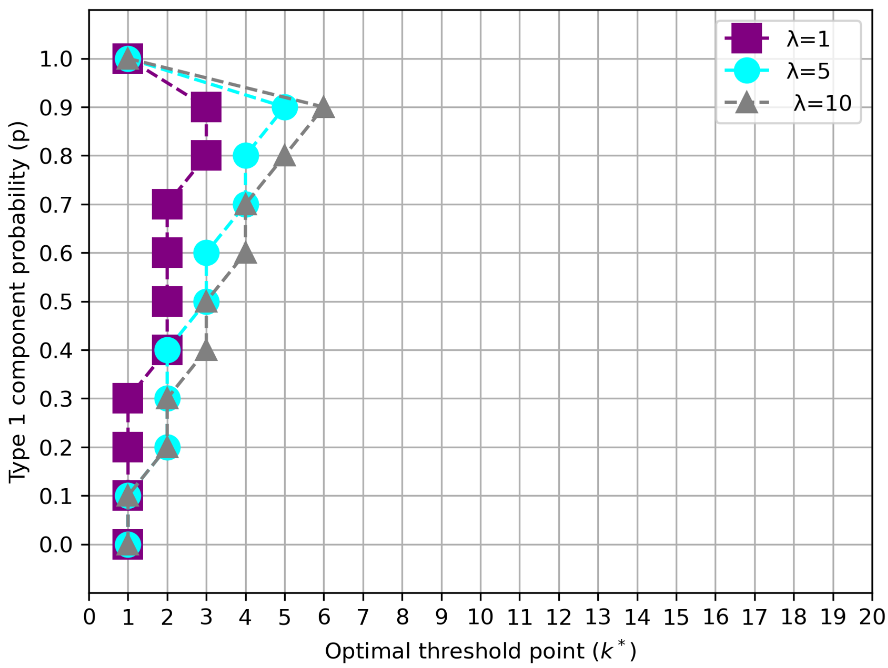

Figure 4.17. Optimal Control Points for a=20 and b=-0.1

4.4.3. The Effect of Reward Coefficient on Optimal Decisions

The reward function is a fundamental component of the model, as it significantly shapes the behavior and decision-making aspects of the system. It dictates the trade-offs the system must make, influencing the timing of replacement decisions. By defining the benefits associated with different actions, the reward function ultimately guides the system towards an optimal balance between waiting for high-quality components and acting based on cost considerations. Thus, the effect of reward coefficient, a, on optimal decisions is analyzed by using the numeric results of 9 different experiments and for each probability value. Optimal decisions are obtained and shown in Figures 4.18-4.26.

Each figure contains three separate plots, representing three different values of

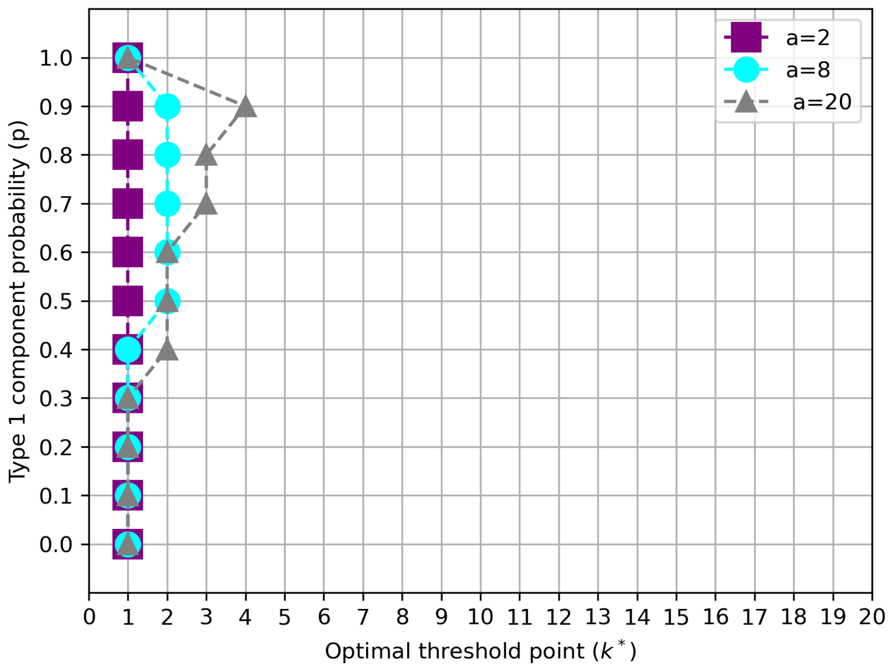

Figure 4.18. Optimal Control Points for λ=1 and b=0

a and markers on plots denote the optimal number of components, k^*. Since higher a values widen the gap between the quality of two different type of components, indicating that the Type 2 component is significantly better than the Type 1 component, it is expected that the system will wait for greater number of components as a value increases. Numerical results reveal this expected behavior for different parameter sets.

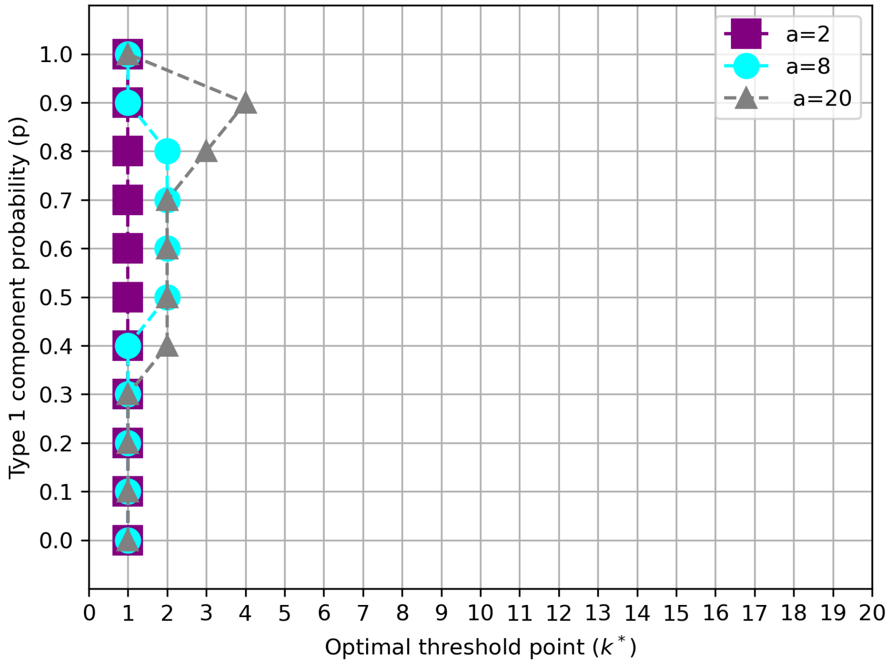

Figure 4.19. Optimal Control Points for λ=1 and b=-0.1/k

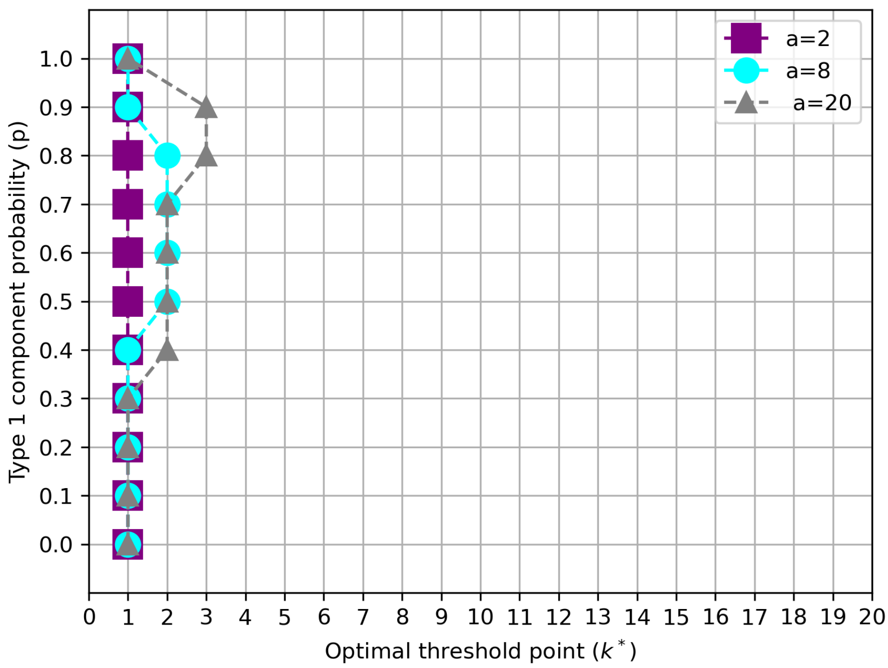

Figure 4.20. Optimal Control Points for λ=1 and b=-0.1

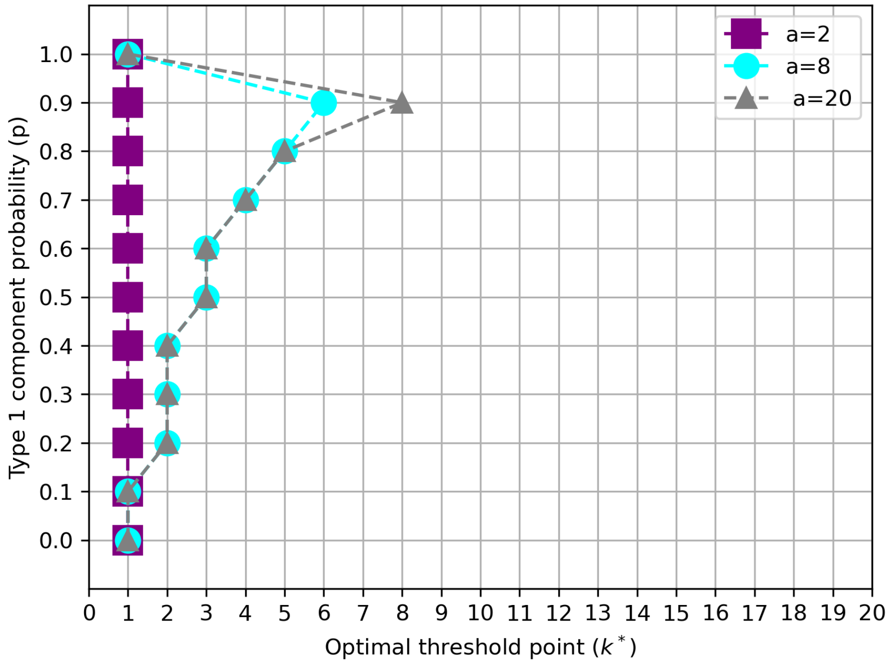

Figure 4.21. Optimal Control Points for λ=5 and b=0

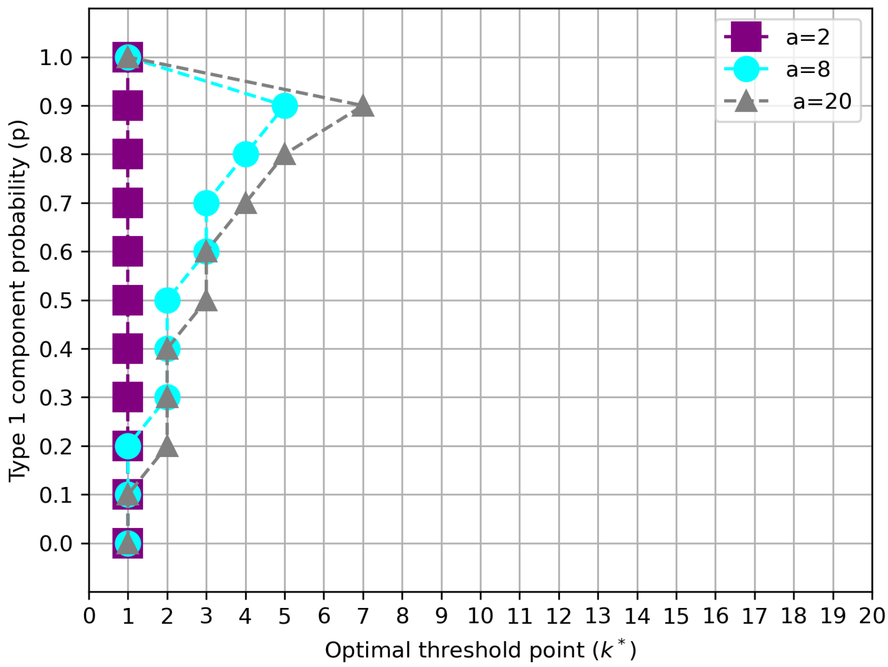

Figure 4.22. Optimal Control Points for λ=5 and b=-0.1/k

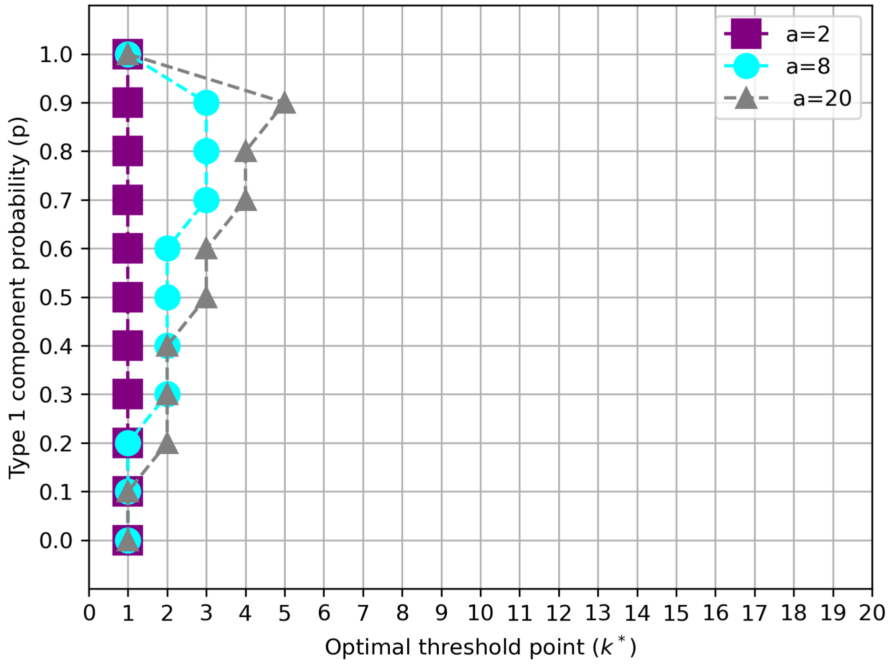

Figure 4.23. Optimal Control Points for λ=5 and b=-0.1

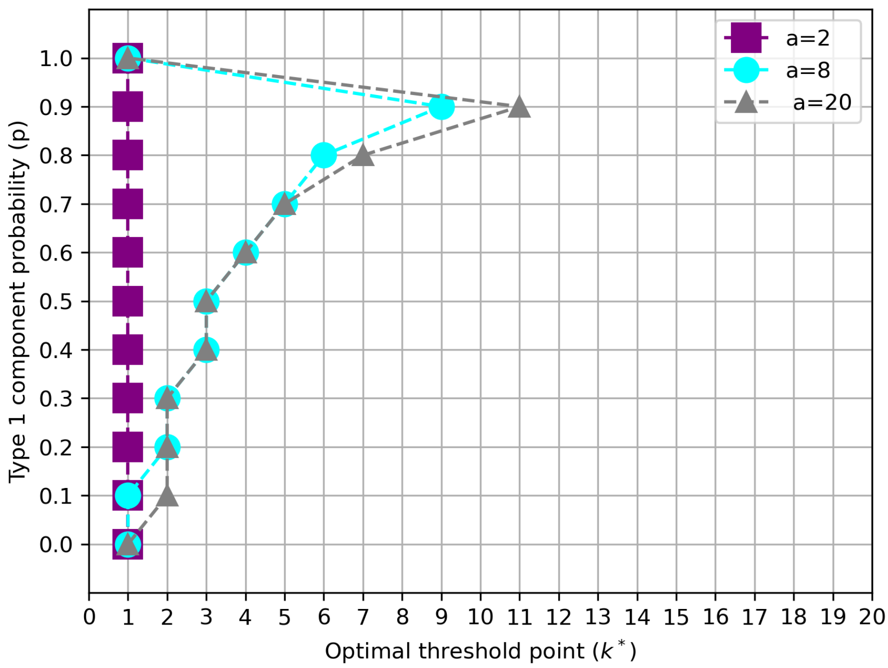

Figure 4.24. Optimal Control Points for λ=10 and b=0

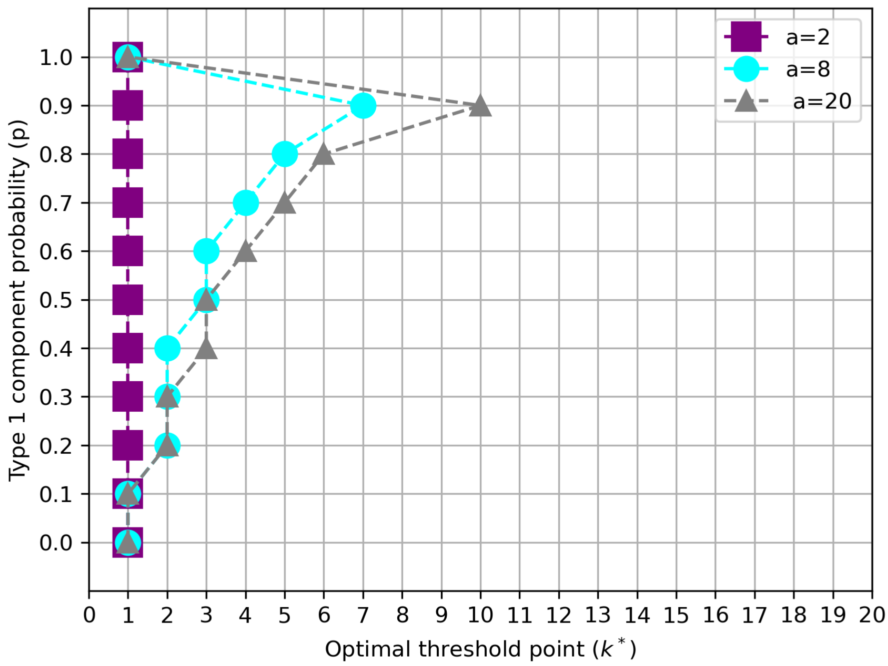

Figure 4.25. Optimal Control Points for λ=10 and b=-0.1/k

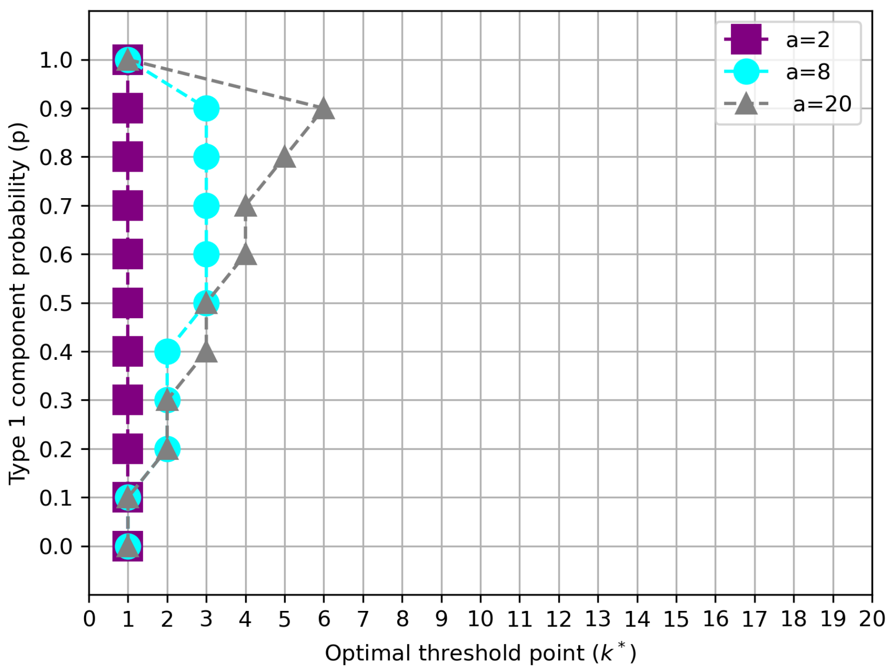

Figure 4.26. Optimal Control Points for λ=10 and b=-0.1

When a value is low, the system's behavior remains unaffected by variations in other parameters. In these cases, k^* remains relatively constant across different scenarios, suggesting that when both component types are of nearly equal quality, the system does not need to wait for a specific type as aggressively. After a value is started to increase, the effect of this parameter on decision space can be observed. It is also seen that the system is more sensitive to changes in a values for higher probability values. When the Type 2 component arrival is rare, i.e. p=0.9, the system responses more quickly to changes in values of a and optimal control point increases rapidly as value a also increases. This rapid increase in k^* with higher a values in high p scenarios emphasizes the system's response to the rarity and value of Type 2 components under such conditions.

4.4.4. The Effect of Cost Coefficient on Optimal Decisions

The cost associated with waiting decision plays a significant role in determining the optimal policies. 9 different experiment results are examined for 3 different cost scenarios which are shown in Figures 4.27 - 4.33.

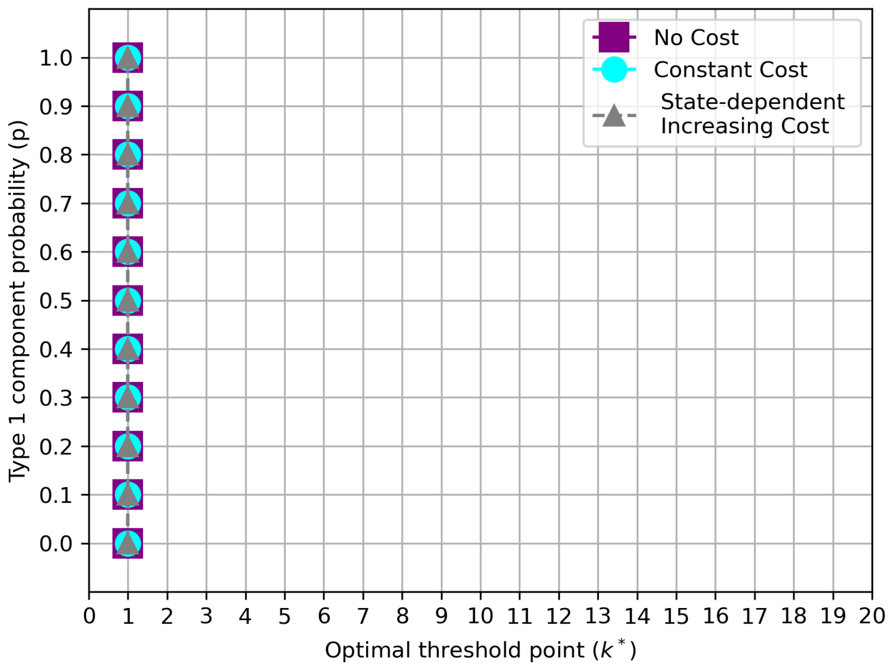

Figure 4.27. Optimal Control Points for a=2 and λ=(1,5,10)

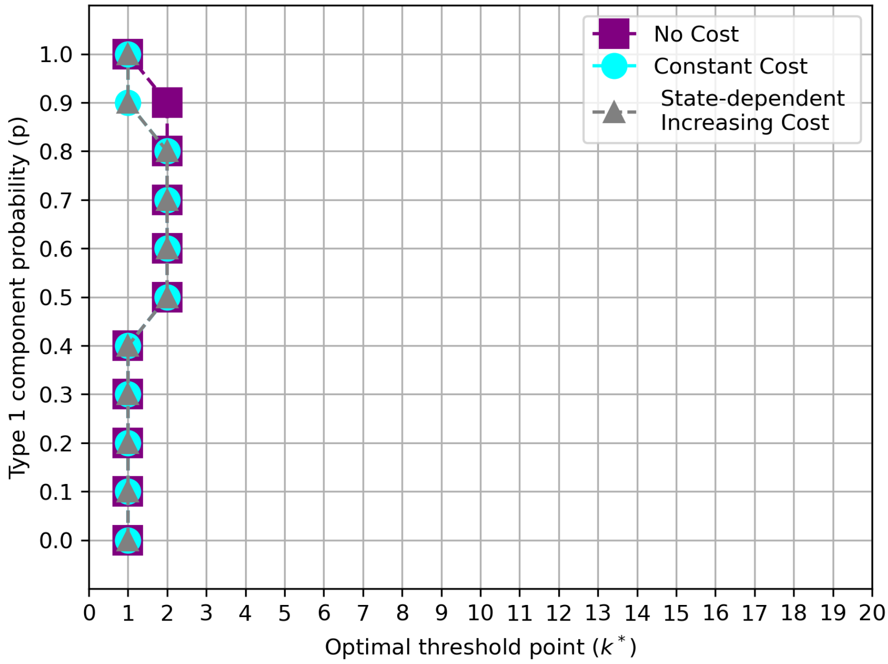

Figure 4.28. Optimal Control Points for a=8 and λ=1

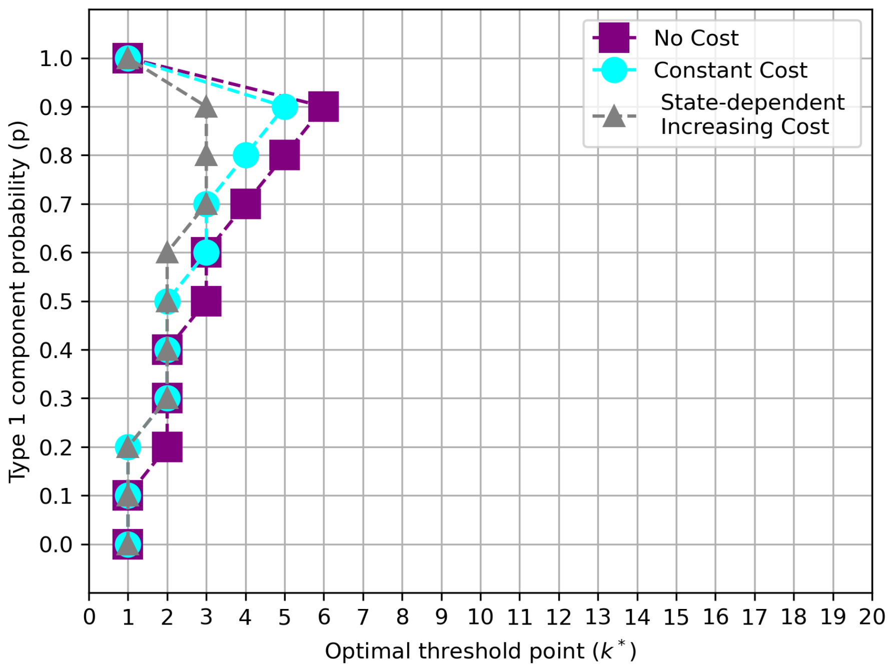

Figure 4.29. Optimal Control Points for a=8 and λ=5

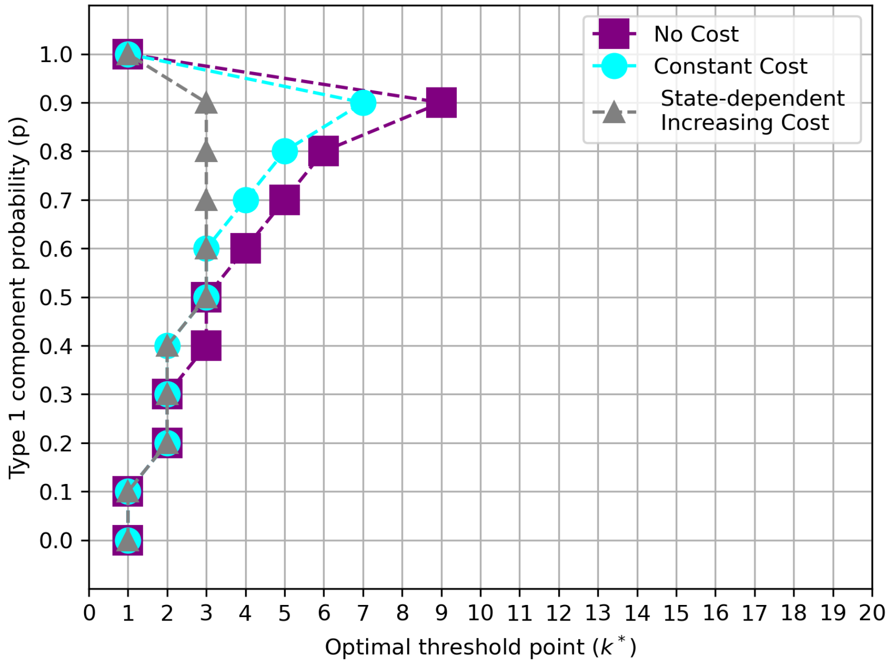

Figure 4.30. Optimal Control Points for a=8 and λ=10

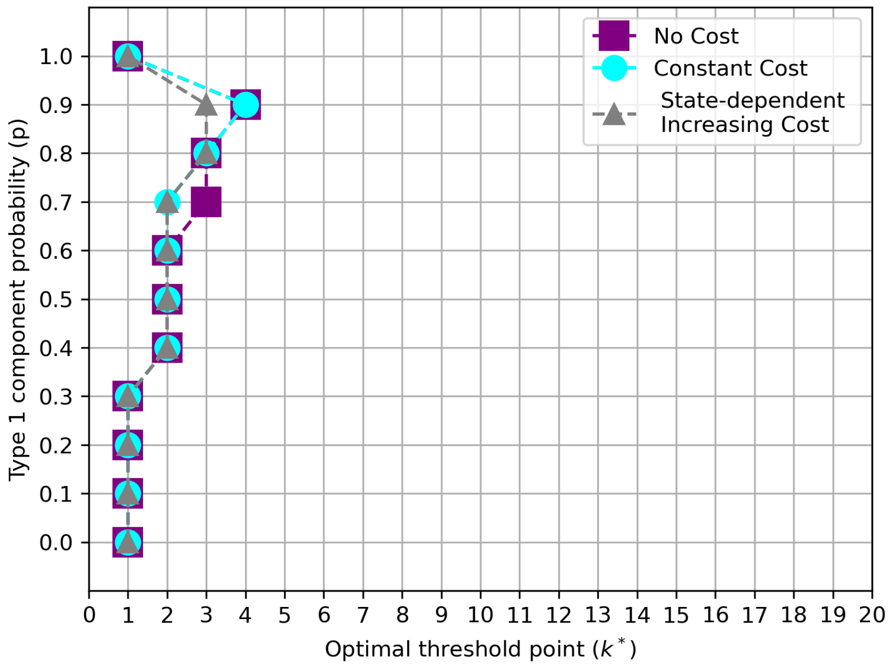

Figure 4.31. Optimal Control Points for a=20 and λ=1

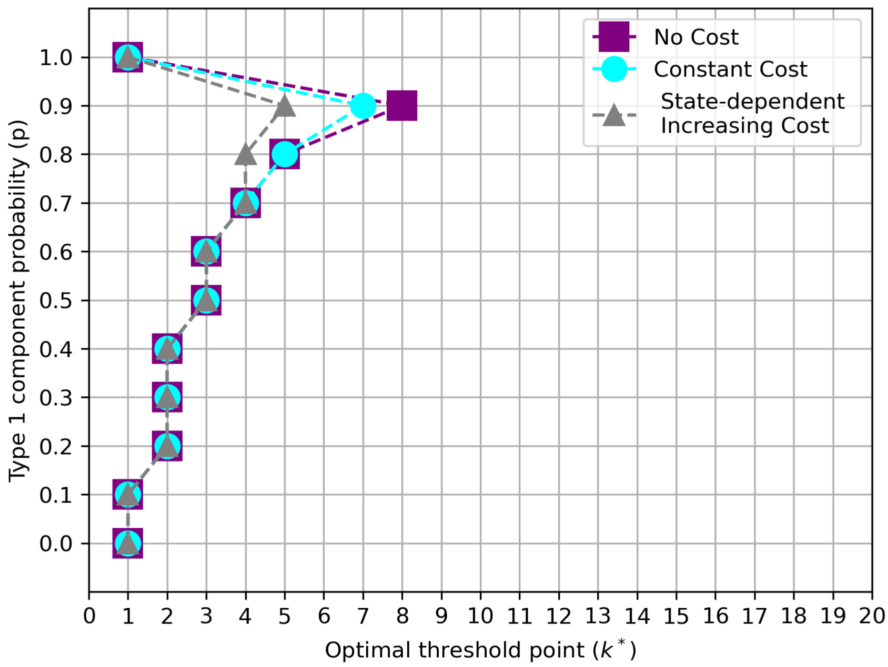

Figure 4.32. Optimal Control Points for a=20 and λ=5

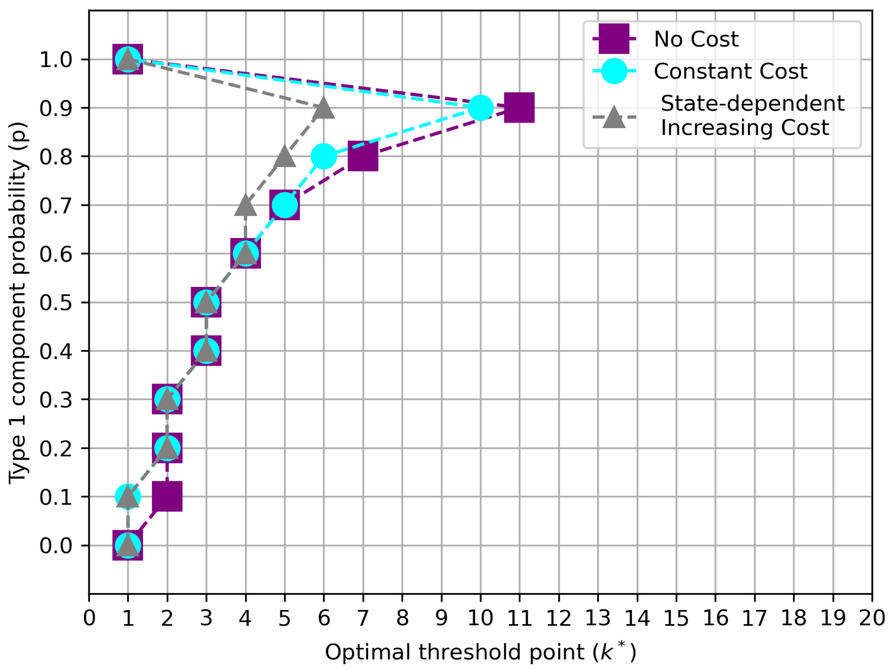

Figure 4.33. Optimal Control Points for a=20 and λ=10

First scenario with b=0 assumes no waiting cost in the system. Other scenario with b=-0.1/k assumes the same constant cost for waiting decision given in each state. Last scenario with b=-0.1 assumes state-dependent increasing cost as number of components increases. When a waiting cost is introduced to the system, the decision to replace components is made at states with a lower number of components. This adjustment reflects the system's attempt to minimize the accumulated waiting costs by acting more promptly, opting for replacement earlier to avoid prolonged waiting times. Consequently, the threshold for initiating replacement decreases, indicating a more responsive strategy in the face of additional costs. Moreover, if the cost increases with respect to the number of components, this leads to a further decrease in the threshold component value. As the cost of waiting for additional components grows, the system becomes more inclined to replace components sooner, prioritizing cost reduction over the probable expected reward of upcoming high quality components. This shift in decision-making behavior further emphasizes the system's sensitivity to increasing costs, driving it to take action at lower component levels to minimize overall expenses. The system is more sensitive to parameter changes for higher probability values which is consistent with the previous findings. For lower probability values, the optimal policies remain relatively consistent across the three different cost scenarios. This indicates that when the probability of obtaining a higher-quality component is high, the system's replacement decisions are less influenced by variations in cost structure. In such cases, the abundance of high-quality components likely dominates the decision-making process, making cost considerations secondary. Consequently, the threshold point for number of components do not vary significantly across the different cost scenarios under low probability conditions.

Considering all results collectively, it can be concluded that the system's optimal policy is most sensitive to changes in the reward coefficient, a. Following this, the arrival rates and cost coefficients have a comparatively lesser impact on the policy. This highlights the critical role of a in shaping the system's behavior, as it directly influences the trade-off between waiting for higher-quality components and replacing at lower number of components. Changes in arrival rates and cost coefficients, while significant, exert a secondary and tertiary influence on the system's decision-making framework, respectively. Hence, the reward structure is a significant determinant of the system behavior.

4.5. Analysis of Extended Models

The base model is enhanced with two sequential extensions. The second extension is implemented after the first extension has been incorporated into the model. Both extended models are analyzed numerically to examine their behavior and to assess the impact of the extensions on the system's output in comparison to the base model. First, a state set representing more deteriorated or inoperable conditions of the system is incorporated into the model. Second, the number of component types is increased from 2 to 3.

4.5.1. Extension 1: Multilevel Dysfunction and Failed State

The model is extended by incorporating an additional set of states, $(-1, k)$, where $k \in$ (0,1,...,40), representing a more deteriorated and inoperable condition of the system for two distinct cases. The extended model is detailed in Section 3.4.1. For two different levels of system deterioration, corresponding to the state sets $(0, k)$ and $(-1, k)$, two optimal control points, k_1^* and k_2^* are computed in the first scenario, respectively. Table 4.6 presents the optimal policies and the associated control points derived from a selected experimental example. Furthermore, these optimal decision points are depicted graphically in Figure 4.34. Two plots for $(0, k)$ and $(-1, k)$ state sets with respect to different probability values are shown in the Figure 4.34. As the system's condition deteriorates further, the optimal policy indicates a tendency to reduce the waiting time, as expected. This behavior aligns with the increased urgency for replacement, given that waiting costs rise as the system condition worsens. Consequently, it is rational for the threshold points in the optimal policy to decrease for the states with h=-1 compared to states with h=0.

Table 4.6. Optimal Policies and Critical Points where λ = 10, a = 8, b = -0.1, c=1, and γ = 1

p	h	k	policy
1	0	1	**1**111111111111111111111111111111111111
	-1	1	**1**111111111111111111111111111111111111
0.9	0	4	000**1**111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.8	0	4	000**1**111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.7	0	3	00**1**1111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.6	0	3	00**1**1111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.5	0	3	00**1**1111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.4	0	2	0**1**11111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.3	0	2	0**1**11111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.2	0	2	0**1**11111111111111111111111111111111111
	-1	2	0**1**11111111111111111111111111111111111
0.1	0	1	**1**111111111111111111111111111111111111
	-1	1	**1**111111111111111111111111111111111111
0	0	1	**1**111111111111111111111111111111111111
	-1	1	**1**111111111111111111111111111111111111

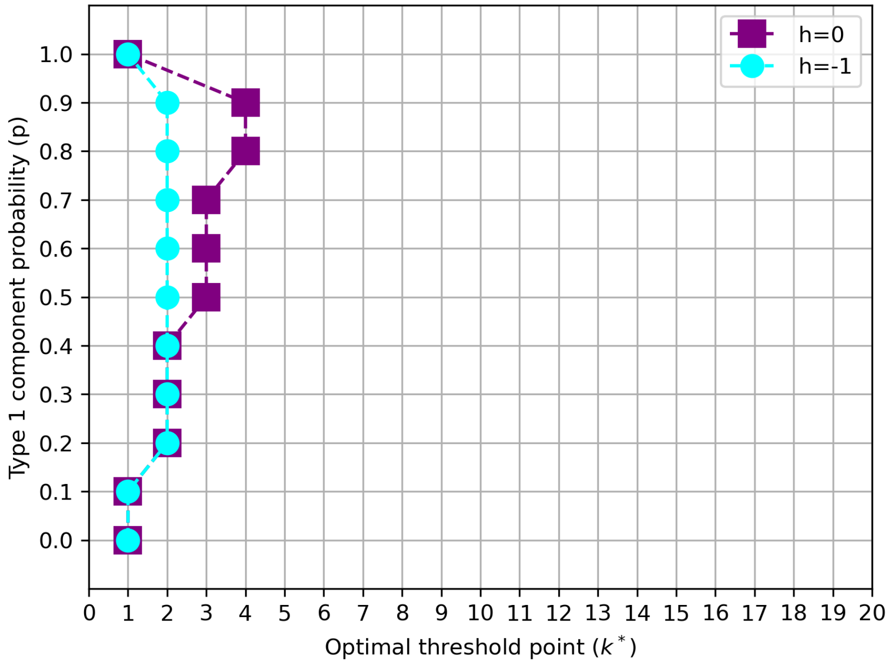

Figure 4.34. Optimal Control Points for different p values

Figures 4.35, 4.36 and 4.37 illustrate the optimal critical points for three different experimental scenarios. The graphs provide clear insight into the system's decision-making process under varying conditions of deterioration and component probabilities. In the absence of a distinct difference between the quality types of components, the system exhibits no preference for waiting, even when no waiting cost is incurred. However, when a waiting cost is introduced, the threshold values for the $(-1, k)$ states are observed to be less than or equal to those for the $(0, k)$ states. This disparity between the threshold values becomes more pronounced at higher probabilities, specifically at 0.8 and 0.9.

For the second scenario, the effect of externally embedded policy on the outcome is analyzed numerically. While actions are available for states with h=0 and k>0, the replacement is predetermined for states with h=-1 and k>0. The optimal policies and corresponding threshold points for selected example are shown in Table 4.7 and Figure 4.38.

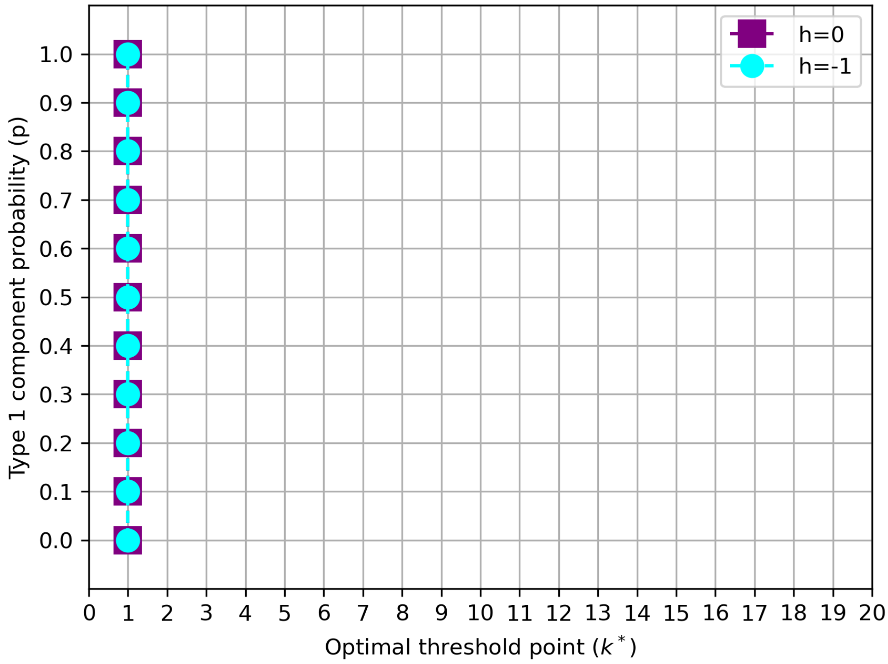

Figure 4.35. Optimal Control Points for a=2, b=0 and λ=1

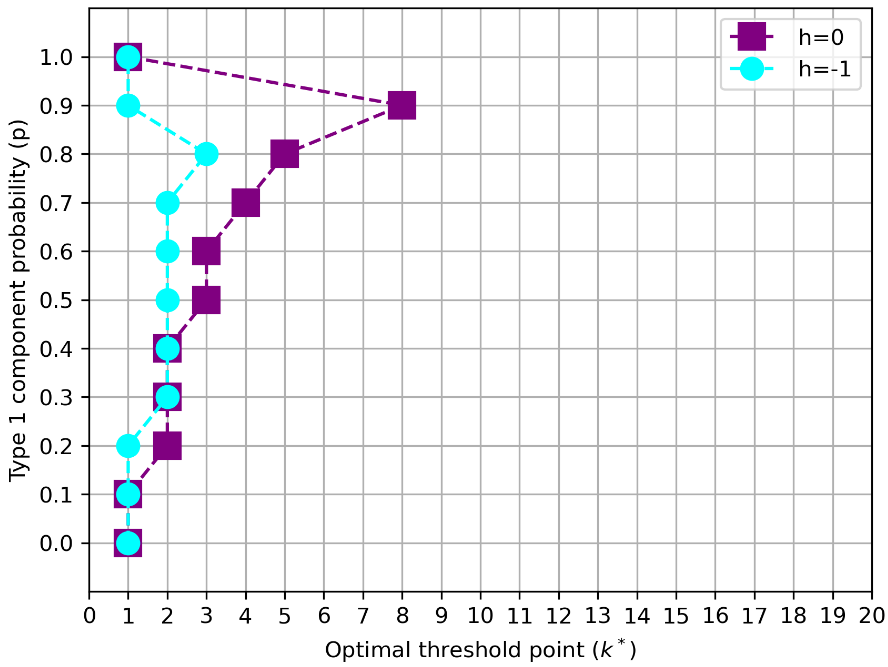

Figure 4.36. Optimal Control Points for a=8, b=-0.1/k and λ=5

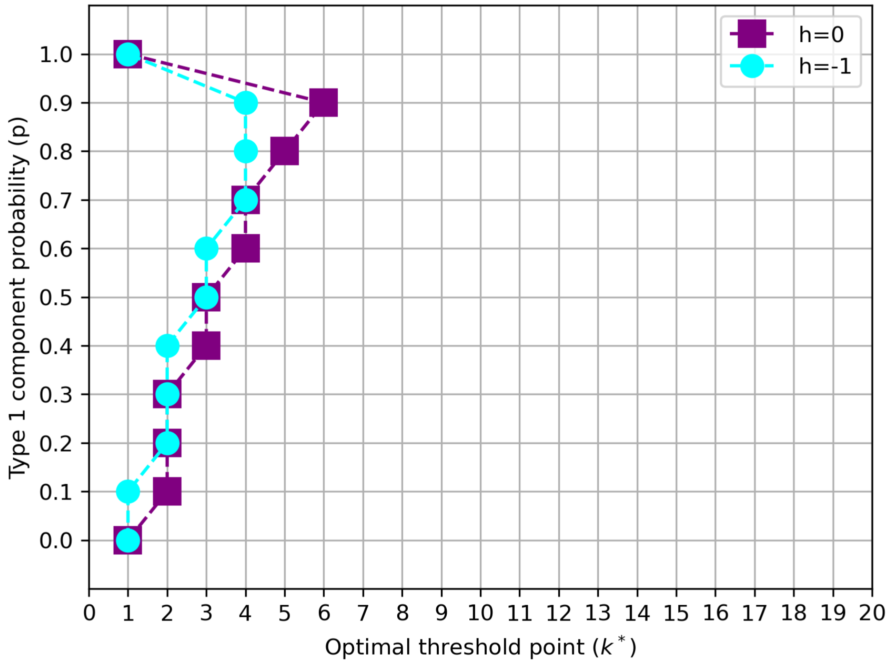

Figure 4.37. Optimal Points for a=20, b=-0.1 and λ=10

Table 4.7. Optimal Policies and Critical Points where λ = 10, a = 8, b = -0.1/k, c=1, and γ = 1

p	k^*	**Policy**
1	1	**1**111111111111111111111111111111111111
0.9	9	00000000**1**1111111111111111111111111111
0.8	6	00000**1**1111111111111111111111111111111
0.7	5	0000**1**11111111111111111111111111111111
0.6	4	000**1**111111111111111111111111111111111
0.5	3	00**1**1111111111111111111111111111111111
0.4	3	00**1**1111111111111111111111111111111111
0.3	2	0**1**11111111111111111111111111111111111
0.2	2	0**1**11111111111111111111111111111111111
0.1	1	**1**111111111111111111111111111111111111
0	1	**1**111111111111111111111111111111111111

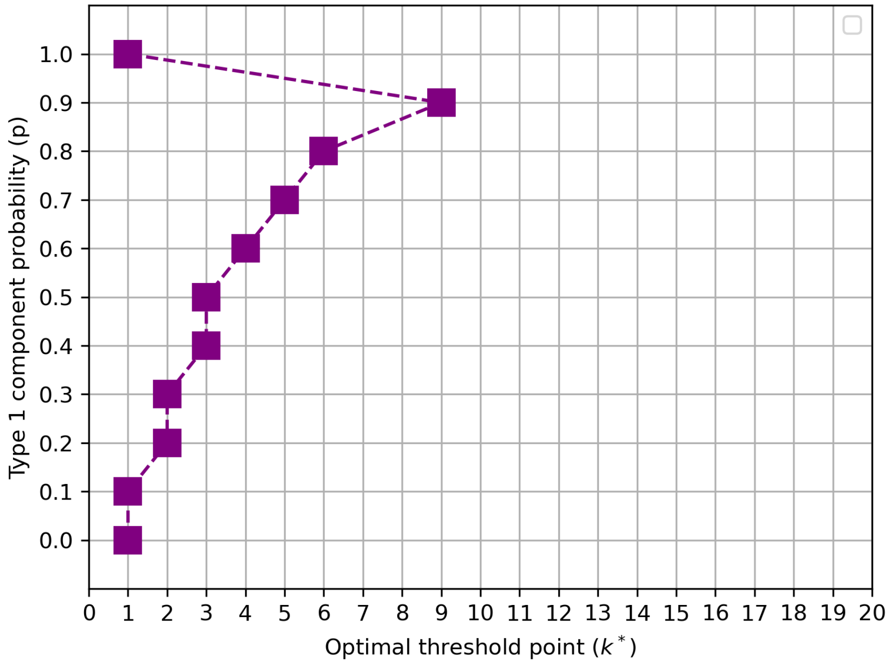

Figure 4.38. Optimal Control Points for different p values

Results of three different experiment are also shown in Figures 4.39,4.40 and 4.41. The results are then compared with the optimal policies obtained for the base model using the same parameter sets to evaluate the effect of the external policy on system outcomes. The analysis reveals that the presence of inoperable states and the incorporation of an externally embedded replacement policy generally lead to an slight increase in the threshold point for some values of p parameter. On the other hand, a comparison with the results of the first case indicates that incorporating inoperable states into the model leads to a noticeable increase in the threshold points, primarily due to the absence of waiting costs associated with the $(1, k)$ states.

4.5.2. Extension 2: Three Quality Levels for Spare Parts

In addition to incorporating the (-1, k) states which denote more deteriorated condition of the system, the model is extended to assume the existence of three types of components, instead of two. Hence, (3,0) state is included in the model to account

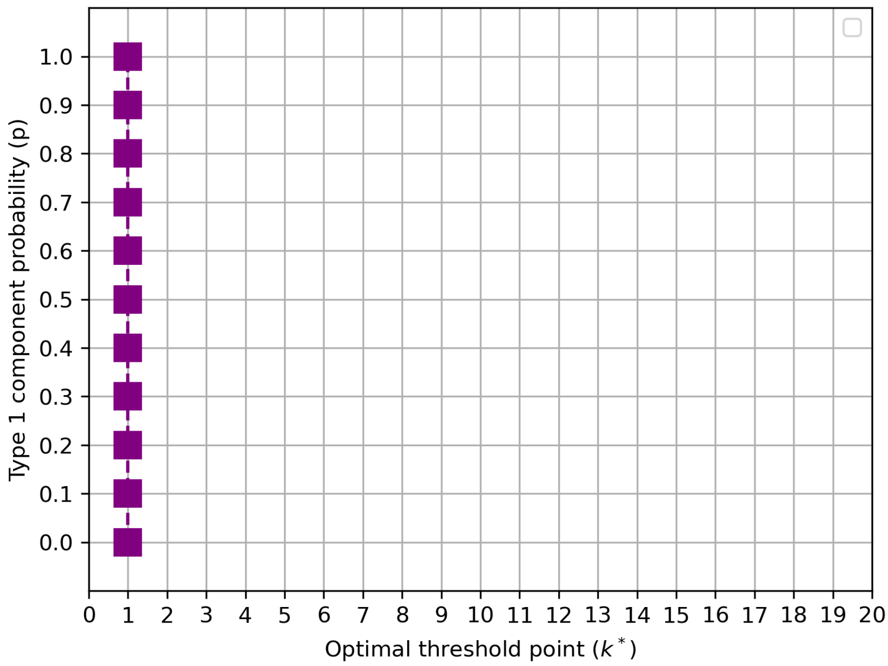

Figure 4.39. Optimal Control Points for a=2, b=0 and λ=1

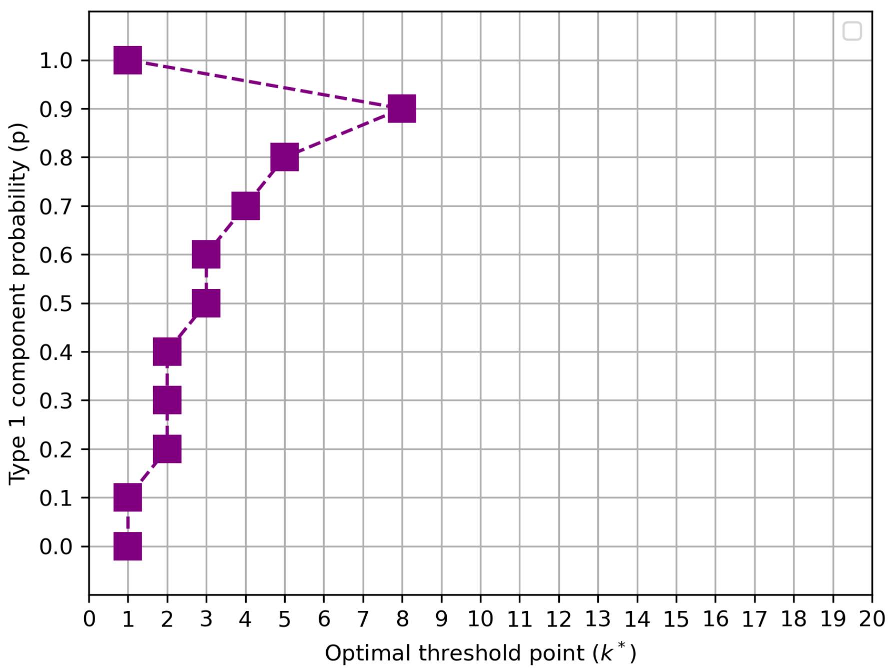

Figure 4.40. Optimal Control Points for a=8, b=-0.1/(k*(1-h)) and λ=5

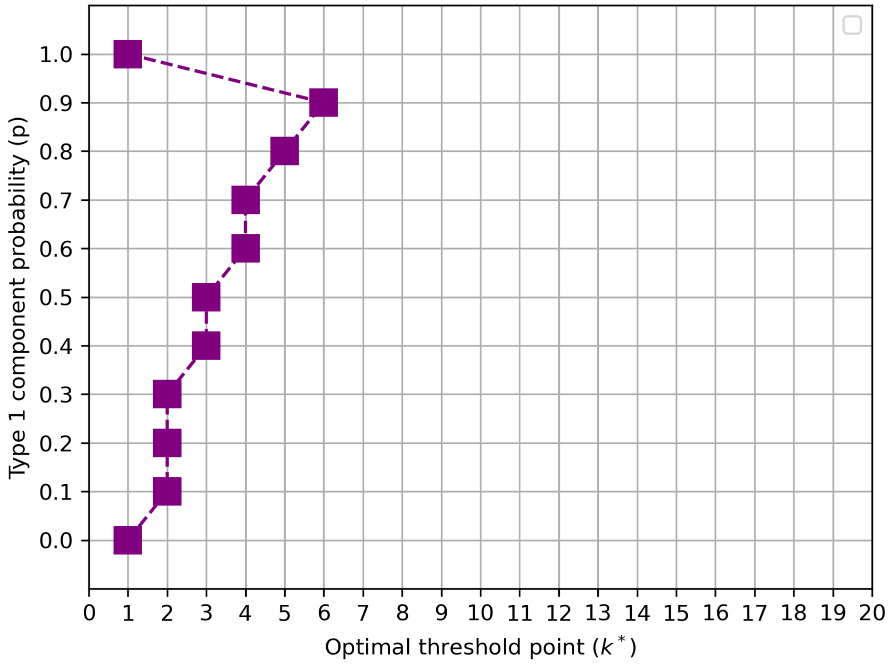

Figure 4.41. Optimal Points for a=20, b=-0.1 and λ=10

for Type 3 component. Two new parameters are defined as a result of addition of (3,0) state which are Type 3 reward coefficient, e, and Type 2 component probability, p_2. Type 1 component probability is denoted by p_1 instead of p. Three different values are determined for these new parameters.

- Type 3 reward coefficient(e): 3, 20, 40
- Type 1 component probability (p_1): 0.2, 0.5, 0.8

The parameter set is updated by these new parameters and it is shown in Table 4.8.

Optimal policies and two related optimal control points of a selected experiment are shown in Table 4.9. Since p_1=0.5 for this experiment, p_2 parameter can take values between 0 and 0.5. The switching points of optimal decisions are shown in Figure 4.42. Two plots show the optimal control points for different p_2 values for h=0 and h=-1 states. When the Type 3 component is incorporated into the system, the system's behavior becomes more intricate yet simultaneously more representative of real-world

Table 4.8. Parameters

	1	**2**	**3**
a	2	8	20
λ	1	5	10
b	0	-0.1/(k*(1-h))	-0.1
e	3	20	40
p_1	0.2	0.5	0.8

dynamics. The inclusion of this component introduces an additional dimension, which significantly influences the interactions among parameters and, consequently, alters the overall system behavior.

Table 4.9. Optimal Policies and Critical Points where λ = 10, a = 8, b = -0.1, c=1, γ = 1, e=20 and p_1=0.5

p_2	h	k	policy
0.5	0	4	000**1**111111111111111111111111111111111
	-1	3	00**1**1111111111111111111111111111111111
0.4	0	5	0000**1**11111111111111111111111111111111
	-1	4	000**1**111111111111111111111111111111111
0.3	0	5	0000**1**11111111111111111111111111111111
	-1	4	000**1**111111111111111111111111111111111
0.2	0	5	0000**1**11111111111111111111111111111111
	-1	4	000**1**111111111111111111111111111111111
0.1	0	4	000**1**111111111111111111111111111111111
	-1	4	000**1**111111111111111111111111111111111
0	0	4	000**1**111111111111111111111111111111111
	-1	4	000**1**111111111111111111111111111111111

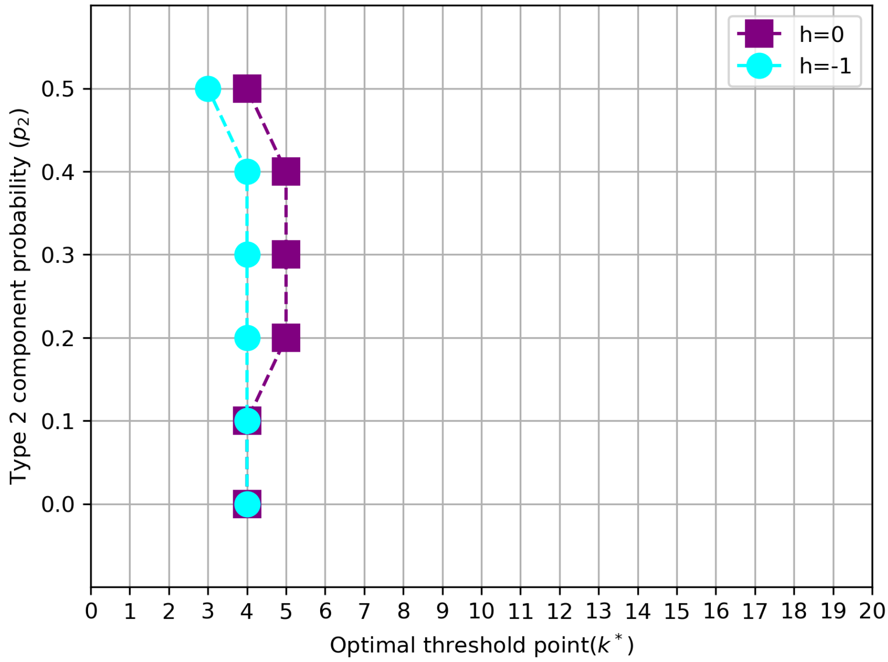

Figure 4.42. Optimal Threshold Points where λ = 10, a = 8, b = -0.1, c=1, γ = 1, e=20 and p_1=0.5

Optimal thresholds for three different experiments are shown in Figures 4.43, 4.44 and 4.45. Compared to the base model, the system tends to wait longer when three types of components are considered. Similar to the model with Extension 1, the optimal control point for the (-1, k) states is lower than or equal to the optimal control point for the (0, k) states. When the Type 3 component offers a higher reward to the system, the difference between the optimal control points, k_1 and k_2, increases. As the quality disparity between the components becomes more pronounced, as indicated by their reward coefficients, the system tends to wait for a larger number of components before deciding on a replacement. In conclusion, alternative models are examined and compared by calculating the expected average rewards and optimal replacement policies for different input settings. the finite Markov Chain model is analyzed numerically. Stationary distributions of the Markov Chain are computed, and the expected rewards are derived using steady-state probabilities under various parameter configurations. The optimal switching point is determined using three alternative methods: Value Iteration Algorithm, Bellman's asymptotic result, and average reward compu-

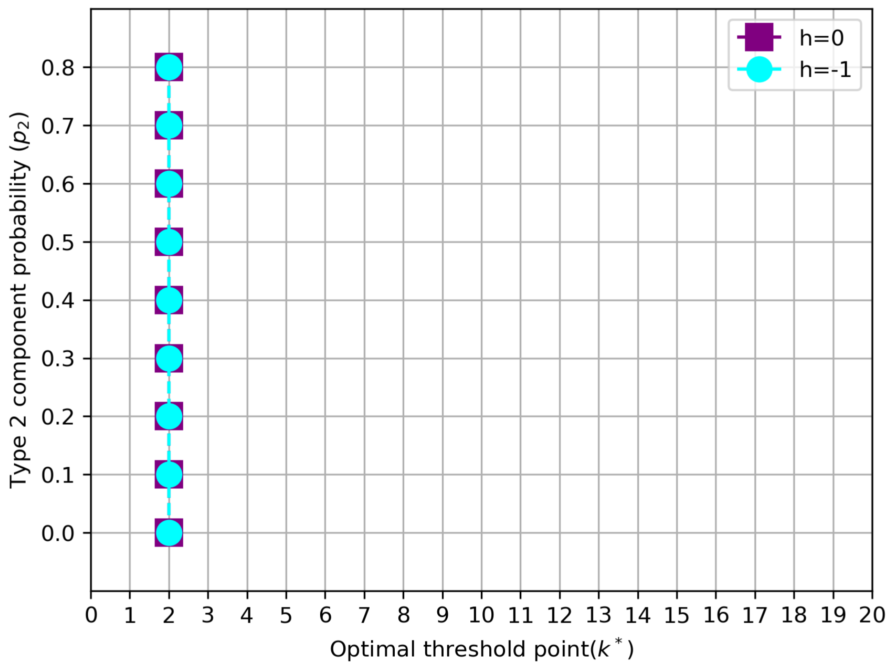

Figure 4.43. Optimal Threshold Points for a=2, b=0, λ=1, e=3 and p_1=0.2

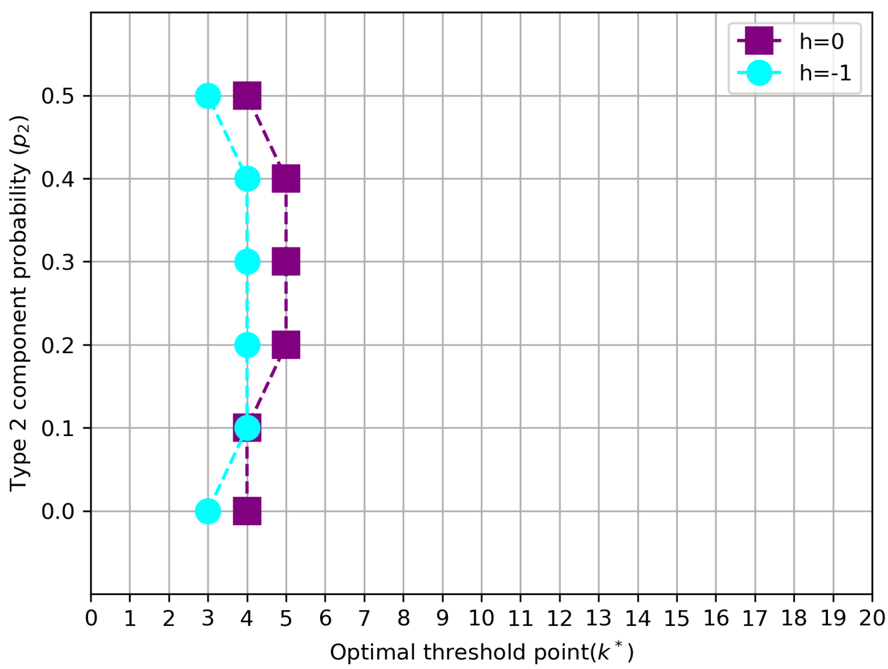

Figure 4.44. Optimal Control Points for a=8, b=-0.1/k, λ=5, e=20 and p_1=0.5

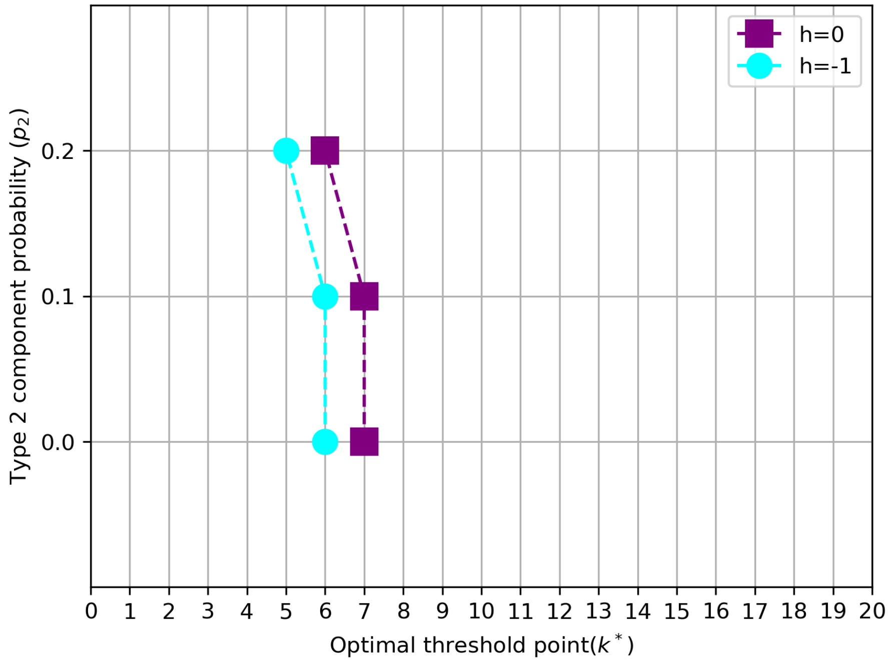

Figure 4.45. Optimal Control Points for a=20, b=-0.1, λ=10, e=40 and p_1=0.8

tation. These methods yield consistent results, validating the numerical calculations. Furthermore, expected rewards are compared across methods to demonstrate their agreement.Moreover, the existence of threshold type optimal policies are also numerically shown in addition to its analytical proof. The numerical analysis underscores the influence of each parameter and their interactions on system behavior and optimal policies. The results reveal the system's sensitivity to parameter variations, providing valuable insights into how changes in inputs affect outcomes. Additionally, the analysis enables the observation of optimal policies under varying parameter configurations, offering a comprehensive understanding of the system's adaptive behavior.

5. CONCLUSION

An optimal replacement problem in a deteriorating system is considered in this study. It is aimed to find the optimal time to make replacement decisions and optimal number of spare parts to accumulate before this decision. The problem is formulated as a Markov Decision Process (MDP) and optimal policies are derived by solving Bellman value equations. The existence of the threshold type optimal solution is proven, demonstrating that below a certain threshold, the optimal action is to wait, while above it, replacement is optimal. The optimal threshold point is analytically deduced by using the definition of value functions. The numerical calculation methods for optimal threshold point are also proposed which are implementation of Value Iteration Algorithm, calculation of average reward of the equivalent finite Markov Chain model and Bellman's [18] asymptotic result. Additionally, a new threshold search algorithm is proposed to determine the optimal threshold point iteratively in cases where the system's expected average reward is either known or can be calculated.

Experiments are conducted using selected parameter values to observe system behavior under different scenarios. The numerical results support the expected system behavior, confirming consistency between analytical deductions and numerical analysis. Sensitivity analyses are performed to examine the impact of parameter changes on optimal policies. It is observed that the optimal threshold point exhibits a positive correlation with the parameter values. The primary stochastic driver of the system is the Type 1 probability parameter, p, which influences system outcomes by dictating the ratio of Type 1 to Type 2 components. As it increases, the optimal threshold point also increases, while extreme values of 0 or 1 eliminate the stochastic nature of the system, making the wait decision non-optimal regardless of other parameters.

The effect of arrival rate, λ, is also investigated. As λ value increases, the optimal threshold point increases in most cases while other parameters are kept constant. When p is higher, the effect of increments in λ on threshold point becomes more obvious.

It is also observed that the reward coefficient, a, is another important determinant on optimal decisions in each state. Numerical analysis reveals that optimal threshold point increases as the the quality gap between two different spare parts widens. Moreover, the quality level of the spare parts are directly related to the reward of the replacement action. On the other hand, waiting cost is the opposite effective dimension on optimal policies compared to the replacement reward. Experiments are made for three different cost scenarios in order to observe the effect of cost coefficient, b, on long-run system behavior. The absence of waiting cost leads to wait for clearly more components when it is compared to scenarios with waiting cost. After the waiting cost is incurred to the system, it is observed that the system is more prone to switch to replace action earlier. Furthermore, if the waiting cost becomes state-dependent which increases as number of spare parts increase, then it accelerates the decision to replace.

New assumptions are also incorporated into the model and analyzed numerically. First, the state space is expanded to account for additional levels of system conditions, leading to multiple optimal switching points. It is observed that replacement occurs earlier as the system deteriorates. Introducing a set of states representing system inoperability and embedding external optimal policies for these states further refines the analysis. Second, the assumption of two spare part types is relaxed to include a third type, increasing the stochastic complexity of the system and the factors influencing optimal policies.

For future research, it can be tried to solve problem with Extension 1 and 2 by using finite Markov Chain model. The model can also be extended to include more intrinsic complexities. For instance, modeling deterioration as a state-dependent process can provide deeper insights. The problem could also be formulated as a transient Markov Chain with an absorbing state representing system failure, allowing for transient analysis. Additional constraints, such as upper bounds on replacement actions, could be explored. Moreover, the problem can be reformulated as a linear program (LP) and solved using optimization techniques. Advanced computational methods, such as reinforcement learning, may also be integrated to enhance decision-making.

For further studies, real life data can be utilized to determine parameter values which makes model more relevant to practical applications. Another interesting extension of the model involves considering a finite number of spare parts, denoted by N, rather than assuming an infinite supply. Under this modification, the arrival rate of spares becomes variable, starting at λN and decreasing as the number of available spare parts diminishes, rather than maintaining a constant rate. This extension would introduce additional complexity into the system dynamics, providing a more realistic framework for scenarios where resource availability is constrained. Analyzing the impact of this finite spare parts assumption could yield valuable insights into system performance and decision-making under resource limitations.

In conclusion, this thesis explored an optimal replacement problem using Markov Decision Process (MDP) tools, aiming to maximize system rewards through optimal decision strategies. This study is significant in terms of giving insight regarding the optimal replacement time of a deteriorating system. Since the deterioration is an inherent process in many natural and industrial systems, the results of our study can be implemented to various fields and utilized by policy makers of these systems despite of its assumptions. It can be very beneficial for preventive care maintenance problems by determining the optimal number of spare parts to wait for under stochastic conditions. Another important finding is that the optimal solution for such a problem is optimal control limit type policy. Optimal control limit type policies provide simple and clear rule for decision-making and these kind of policies are easy to implement. Besides, we provide different calculation methodologies for finding optimal switching point. The proposed methodology can be extended to incorporate additional complexities, such as multi-component systems, time-varying costs, and non-stationary environments. In short, this thesis advances the understanding of optimal replacement policies within stochastic systems and provides a foundation for further studies. It can be employed for practical applications in the industry with further extensions besides of theoretical findings it brings to the literature.

REFERENCES

1. McMillan, D. and G. Ault, “Condition Monitoring Benefit for Onshore Wind Turbines: Sensitivity to Operational Parameters”, *IET Renewable Power Generation*, Vol. 2, No. 1, pp. 60–72, 2008.

2. Icten, Z., S. Shechter, L. Maillart and M. Nagarajan, “Optimal Management of a Limited Number of Replacements Under Markovian Deterioration”, *IIE Transactions*, Vol. 45, No. 2, pp. 206–214, 2013.

3. Kurt, M. and J. Kharoufeh, “Monotone Optimal Replacement Policies for a Markovian Deteriorating System in a Controllable Environment”, *Operations Research Letters*, Vol. 38, No. 6, pp. 273–279, 2010.

4. Ulukus, M., J. Kharoufeh and L. Maillart, “Optimal Replacement Policies Under Environment-Driven Degradation”, *Probability in the Engineering and Informational Sciences*, Vol. 27, No. 1, pp. 139–162, 2013.

5. Cekyay, B. and S. Ozekici, “Optimal Maintenance of Systems with Markovian Mission and Deterioration”, *European Journal of Operational Research*, Vol. 219, No. 2, pp. 123–133, 2012.

6. Karabağ, O., O. Bulut, A. Toy and M.M.Fadıloğlu, “An Efficient Procedure for Optimal Maintenance Intervention in Partially Observable Multi-Component Systems”, *Reliability Engineering and System Safety*, Vol. 244, p. 109914, 2024.

7. Abdul-Malak, D., J. Kharoufeh and L. Maillart, “Maintaining Systems with Heterogeneous Spare Parts”, *Naval Research Logistics*, Vol. 66, No. 6, pp. 485–501, 2019.

8. Byon, E., L. Ntaimo and Y. Ding, “Optimal Maintenance Strategies for Wind Turbine Systems Under Stochastic Weather Conditions”, *IEEE Transactions on*

Reliability, Vol. 59, No. 2, pp. 393–404, 2010.

9. Hu, Q. and W. Yue, “Optimal Replacement of a System According to a Semi-Markov Decision Process in a Semi-Markov Environment”, *Optimization Methods and Software*, Vol. 18, No. 2, pp. 181–196, 2003.

10. Ciocan, D. and V. Misic, “Interpretable Optimal Stopping”, *Management Science*, Vol. 68, No. 3, pp. 1616–1638, 2022.

11. Alagoz, O., L. Maillart, A. Schaefer and M. Roberts, “Choosing among Living-Donor and Cadaveric Livers”, *Operations Research*, Vol. 55, No. 5, pp. 748–763, 2007.

12. Ünver, M., “Dynamic Kidney Exchange”, *The Review of Economic Studies*, Vol. 77, No. 1, pp. 1–32, 2010.

13. Kurt, M., B. Denton, A. Schaefer and N. Shah, “The Structure of Optimal Statin Initiation Policies for Patients with Type 2 Diabetes”, *Operations Research*, Vol. 58, No. 6, pp. 1341–1357, 2010.

14. Shechter, S., M. Bailey, A. Schaefer and M. Roberts, “The Optimal Time to Initiate HIV Therapy Under Ordered Health States”, *Operations Research*, Vol. 56, No. 1, pp. 20–33, 2008.

15. Ross, S. M., *Introduction to Probability Models*, Academic Press, San Diego, California, 2003.

16. Barlow, R. and F. Proschan, *Mathematical Theory of Reliability*, Wiley, New York, NY, 1965.

17. Ross, S. M., *Introduction to Stochastic Dynamic Programming*, Academic Press, New York, New York, 1983.

18. Bellman, R., "A Markovian Decision Process", *Journal of Mathematics and Mechanics*, Vol. 6, No. 5, pp. 679–684, 1957.

19. Puterman, M. L., *Markov Decision Processes Discrete Stochastic Dynamic Programming*, Wiley, Hoboken, New Jersey, 2005.

APPENDIX A: THE VALUE ITERATION ALGORITHM

The following code of Value Iteration Algorithm is implemented in Python to obtain numerical results.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
class ValueIteration:
    def __init__(self,state_space, actions,
       reward_function,cost_function,  transition_model, lam,
       gamma,p,c,a,b ):
        self.state_space = state_space
        self.actions = actions
        self.num_actions = len(actions)
        self.num_states = len(state_space)
        self.probability= p
        self.c=c
        self.a=a
        self.b=b
        self.reward_function = self.reward_func(c,a)
        self.cost_function = self.cost_func()
        self.lam = lam
        self.gamma = gamma
        self.totalrate = self.lam + self.gamma
        self.values = np.zeros(self.num_states)
        self.transition_model = self.transition_matrix()
        self.policy = None
    def one_iteration(self):
        delta_values = []
        delta = 0
        tempvalues=np.zeros(self.num_states)
```

```
for s in range(self.num_states):
    temp = self.values[s]
    v_list = np.zeros(self.num_actions)
    if  self.state_space[s] == (2, 0) or
       self.state_space[s] == (1, 0):
        for a in range(self.num_actions):

            v_list[a] = (self.gamma *
               self.values[s+1]+self.lam *
               self.values[s])/self.totalrate

    elif self.state_space[s] == (0, 0):
        for a in range(self.num_actions):

                v_list[a] =  (self.lam *
                   (self.values[s+1])+self.gamma *
                   self.values[s])/self.totalrate

    elif self.state_space[s] != (0, 40) :
        for a in range(self.num_actions):

            if a == 0:
                v_list[a] = (self.lam *
                   (self.values[s+1]) + self.gamma *
                   (self.values[s]))/self.totalrate +
                   self.cost_function[s]
            else:
                n= self.state_space[s][1]
                pmat = self.transition_model[s, a]
                v_list[a] = self.reward_function[s]+
                   np.sum(pmat * self.values)
```

```
        else :
            for a in range(self.num_actions):
                if a == 0:
                    v_list[a] =  (self.lam *
                       (self.values[s])+self.gamma
                    *self.values[s])/self.totalrate +
                       self.cost_function[s]
                else:
                    n= self.state_space[s][1]
                    pmat = self.transition_model[s, a]
                    v_list[a] = self.reward_function[s] +
                       np.sum(pmat * self.values)

        tempvalues[s]= max(v_list)
    self.values = tempvalues
    delta_values =  abs(temp - self.values)

    print(f'value function = {self.values}')
    return delta_values

def reward_func(self,c,a):

    reward_function = np.zeros(self.num_states)

    for s in range(self.num_states):

        if self.state_space[s] == (2, 0) :
            reward_function[s] =a*c
        elif self.state_space[s] == (1, 0):
            reward_function[s] = c
        else:
            n= self.state_space[s][1]
```

```
                reward_function[s]=((self.probability)**n)
                *reward_function[1]+(1-((self.probability)**n))
                *reward_function[0]

        return reward_function

    def cost_func(self):

        cost_function = np.zeros(self.num_states)

        for s in range(self.num_states):
            if  self.state_space[s] != (2, 0) and
               self.state_space[s] != (1, 0) and
               self.state_space[s] != (0, 0):
                cost_function[s]=-0.1*self.state_space[s][1]

        return cost_function

    def transition_matrix(self):

        transition_matrix = np.zeros((self.num_states,
           self.num_actions, self.num_states))

        transition_matrix[state_space.index((2,0)),
           actions.index('Wait'), state_space.index((2,0))] =
           self.lam / self.totalrate
        transition_matrix[state_space.index((2,0)),
           actions.index('Wait'), state_space.index((1,0))] =
           self.gamma / self.totalrate
        transition_matrix[state_space.index((2,0)),
           actions.index('Wait'), state_space.index((0,0))] = 0
        transition_matrix[state_space.index((2,0)),
           actions.index('Wait'), state_space.index((0,1))] = 0
```

```
transition_matrix[state_space.index((2,0)),
   actions.index('Wait'), state_space.index((0,2))] = 0
transition_matrix[state_space.index((2,0)),
   actions.index('Transplant'), state_space.index((2,0))]
   = self.lam / self.totalrate
transition_matrix[state_space.index((2,0)),
   actions.index('Transplant'), state_space.index((1,0))]
   = self.gamma / self.totalrate
transition_matrix[state_space.index((2,0)),
   actions.index('Transplant'), state_space.index((0,0))]
   = 0
transition_matrix[state_space.index((2,0)),
   actions.index('Transplant'), state_space.index((0,1))]
   = 0
transition_matrix[state_space.index((2,0)),
   actions.index('Transplant'), state_space.index((0,2))]
   = 0
transition_matrix[state_space.index((1,0)),
   actions.index('Wait'), state_space.index((2,0))] = 0
transition_matrix[state_space.index((1,0)),
   actions.index('Wait'), state_space.index((1,0))]=
   self.lam / self.totalrate
transition_matrix[state_space.index((1,0)),
   actions.index('Wait'), state_space.index((0,0))] =
   self.gamma / self.totalrate
transition_matrix[state_space.index((1,0)),
   actions.index('Wait'), state_space.index((0,1))] = 0
transition_matrix[state_space.index((1,0)),
   actions.index('Wait'), state_space.index((0,2))] = 0
transition_matrix[state_space.index((1,0)),
   actions.index('Transplant'), state_space.index((2,0))]
   = 0
transition_matrix[state_space.index((1,0)),
```

```
    actions.index('Transplant'), state_space.index((1,0))]
    = self.lam / self.totalrate
transition_matrix[state_space.index((1,0)),
    actions.index('Transplant'), state_space.index((0,0))]
    = self.gamma / self.totalrate
transition_matrix[state_space.index((1,0)),
    actions.index('Transplant'), state_space.index((0,1))]
    = 0
transition_matrix[state_space.index((1,0)),
    actions.index('Transplant'), state_space.index((0,2))]
    = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Wait'), state_space.index((2,0))] = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Wait'), state_space.index((1,0))] = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Wait'), state_space.index((0,0))] =
    self.gamma / self.totalrate
transition_matrix[state_space.index((0,0)),
    actions.index('Wait'), state_space.index((0,1))] =
    self.lam / self.totalrate
transition_matrix[state_space.index((0,0)),
    actions.index('Wait'), state_space.index((0,2))] = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Transplant'), state_space.index((2,0))]
    = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Transplant'), state_space.index((1,0))]
    = 0
transition_matrix[state_space.index((0,0)),
    actions.index('Transplant'), state_space.index((0,0))]
    = self.gamma / self.totalrate
transition_matrix[state_space.index((0,0)),
```

```
    actions.index('Transplant'), state_space.index((0,1))]
    = self.lam / self.totalrate
transition_matrix[state_space.index((0,0)),
    actions.index('Transplant'), state_space.index((0,2))]
    = 0

for a in range(1,41):
     for b in self.actions:
         if b == 'Wait' and a != 40:
             transition_matrix[state_space.index((0,a)),
                actions.index(b),
                state_space.index((0,a+1))] = self.lam /
                self.totalrate
             transition_matrix[state_space.index((0,a)),
                actions.index(b), state_space.index((0,a))]
                = self.gamma / self.totalrate
         elif b == 'Wait' and a == 40:
             transition_matrix[state_space.index((0,a)),
                actions.index(b), state_space.index((0,a))]
                = 1
         else:
             transition_matrix[state_space.index((0,a)),
                actions.index(b), state_space.index((2,0))]
                = (self.lam*(1-((self.probability)**a)))
             /self.totalrate
             transition_matrix[state_space.index((0,a)),
                actions.index(b), state_space.index((1,0))]
                = (self.lam*((self.probability)**a))
             /self.totalrate
             transition_matrix[state_space.index((0,a)),
                actions.index(b), state_space.index((0,a))]
                = self.gamma / self.totalrate
```

```
        return transition_matrix

    def get_policy(self):
        pi = np.ones(self.num_states) * -1

        for s in range(self.num_states):
            v_list = np.zeros(self.num_actions)
            if  self.state_space[s] == (2, 0) or
               self.state_space[s] == (1, 0):
                for a in range(self.num_actions):

                    v_list[a] = (self.gamma *
                       self.values[s+1]+self.lam *
                       self.values[s])/self.totalrate
                    max_index = []
                    max_val = np.max(v_list)

                max_index = np.where(v_list == max_val)[0]
                if max_index.size > 0:
                    pi[s] = min(max_index)

            elif self.state_space[s] == (0, 0):
                for a in range(self.num_actions):
                    v_list[a] =  (self.lam *
                       self.values[s+1]+self.gamma *
                       self.values[s])/self.totalrate

                    max_index = []
                    max_val = np.max(v_list)

                max_index = np.where(v_list == max_val)[0]
                if max_index.size > 0:
                    pi[s] = min(max_index)
```

```
elif self.state_space[s] != (0, 40) :
    for a in range(self.num_actions):
        if a == 0:
            v_list[a] =  (self.lam *  self.values[s+1]
               + self.gamma *
               (self.values[s]))/self.totalrate +
               self.cost_function[s]
            max_index = []
            max_val = np.max(v_list)
        else:
            pmat = self.transition_model[s, a]
            v_list[a] =  self.reward_function[s] +
               np.sum(pmat * self.values)
            max_index = []
            max_val = np.max(v_list)

    max_index = np.where(v_list == max_val)[0]
    if max_index.size > 0:
        pi[s] = max(max_index)

else :
    for a in range(self.num_actions):

        if a == 0:
            v_list[a] =  (self.lam *
               (self.values[s])+self.gamma
            *self.values[s])/self.totalrate +
               self.cost_function[s]
            max_index = []
            max_val = np.max(v_list)
        else:
            pmat = self.transition_model[s, a]
```

```
                    v_list[a] = self.reward_function[s] +
                       np.sum(pmat * self.values)
                    max_index = []
                    max_val = np.max(v_list)

            max_index = np.where(v_list == max_val)[0]
            if max_index.size > 0:
                pi[s] = max(max_index)

    return pi.astype(int)

def train(self, tol=1e-3):
    epoch = 0
    prevdelta=np.zeros(self.num_states)
    delta = self.one_iteration()
    delta_history = []
    delta_history.append(delta)
    change = abs(np.array(delta)-np.array(prevdelta))
    change_history = []
    change_history.append(change)
    policy_history = [self.get_policy()]
    k=0
    while np.any(change >tol):
        epoch += 1
        prevdelta= delta
        delta = self.one_iteration()
        delta_history.append(delta)
        change = abs(np.array(delta)-np.array(prevdelta))
        change_history.append (change)
        policy_history.append(self.get_policy())
        if np.all(change < tol):
            break
        print(f'Iteration {epoch}: PrevDelta = {prevdelta},
```

```
                Delta = {delta}, Change = {change}, Policy =
                {self.get_policy()}')

        self.policy = self.get_policy()
        for j in range(3,43):
            if j==3 and self.policy[j]==1:
                print(f'REPLACE')
                k=1
                break
            elif j>3 and self.policy[j]==1:
                print(f'WAIT {j-2}')
                k=j-2
                break

        print(f'# iterations of policy improvement:
           {len(delta_history)}')
        print(f'delta = {delta_history}')
        print(f' g values = {delta_history[-1]}')
        print(f'change = {change_history}')
        print(f'policy = {policy_history}')

        return delta_history, policy_history, change_history,k

state_space = [(2,0),(1,0)]
for x in range(41):
    state_space.append((0,x))

actions = ['Wait','Transplant']

transition_model =
   np.zeros((len(state_space),len(actions),len(state_space)))
reward_function = np.zeros(len(state_space))
cost_function = np.zeros(len(state_space))
```

```
lam=10
gam=1
c=1
a=8

pval = (y*0.1 for y in range(0,11))
results = []

for p in pval:
    print(f'probability = {p}')
    mdp= ValueIteration(state_space,actions, transition_model,
       reward_function,cost_function,lam,gam,p,c,a,b)
    k= mdp.train()[3]

    results.append({
            'p': round(p, 1),
            'k': k,
            'lambda': lam,
            'gamma': gam,
            'c':c,
             'a':a,
        })

df = pd.DataFrame(results)
print(df)
fig, ax = plt.subplots()
ax.plot(df['k'], df['p'], 's--',color='purple', markersize=12,
   markerfacecolor='purple')
ax.set_xlabel('Optimal threshold point ($k^*$)')
ax.set_ylabel('Type 1 component probability (p)')
ax.set_xlim(0, 20)
ax.set_ylim(-0.1, 1.1)
ax.set_xticks(range(0, 21, 1))
```

```
ax.set_yticks([i * 0.1 for i in range(11)])
plt.legend()
plt.grid()
plt.savefig('plot.png', format='png', dpi=300, bbox_inches='tight')
plt.show()
```