Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics and Management Science

Optimal upgrade planning: An application in a desktop computer

Panagiotis Paspalas (559584)

Supervisor:	Rommert Dekker
Second assessor:	Name of your second assessor
Date final version:	22nd February 2024

Abstract

This paper explores strategies for optimizing the replacement policy of computer components to achieve lifetime extension (LTE) of desktop computers. The research examines a Markov Decision Model (MDM) that evaluates the average replacement costs via time discretisation, utilizing a linear programming (LP) formulation. An extension of the problem introduces dependencies among components, with CPU having a pivotal role in decision making. While these approaches provide optimal solutions, their interpretability may be limited. To address this, the paper introduces a heuristic approach called the Power of Two (PoT) policy, comparing its performance to the MDM. Lastly, the study enhances the optimal replacement policy by incorporating environmental considerations, specifically targeting the major contributor of carbon dioxide (CO_2) during manufacturing phase of computer components.

Keywords: Lifetime extension, Markov Decision Model, Power of Two, CO_2 emissions

Preface

I would like to take this opportunity to express my gratitude to my thesis coordinator Professor Rommert Dekker who supported and guided me all this time. Without his expertise, patience, and encouragement, this research project would not have been possible.

In addition, I want to extend my heartfelt thanks to my parents for their emotional and economical support as well as their sacrifices throughout this journey. Their belief in my potential, their encouragement during moments of self-doubt, and their endless dedication have been my driving force.

Last but certainly not least, I dedicate this work to my beloved grandmother, Helen, who watches over me from the heavens above. Her memory and influence have been a guiding light throughout these years, and without her presence in my thoughts, this significant milestone would not have been attainable.

Contents

Introduction

In a rapidly evolving world, it is of crucial importance to keep up with the latest technological advancements. A prediction in February 2023 indicated that consumers will spend around 685 billion dollars on electronic devices throughout the year (Statista Research Department, 2023c). Only in the Netherlands, the revenue from the electronics market is estimated at around 7 billion euros in 2023 with annual growth rate of 1.21% for the next five years (Statista Research Department, 2023b). This trend indicates the willingness of consumers to spend in order to align with the contemporary technology.

Conversely, the economic impact tends to burden consumers more frequently as a result of rapid technological obsolescence. The accelerated progress of technology and market competition are driving the production of tech products with shorter and shorter lifespans, leading to economical obsolesce (high repair and maintenance costs) even before the end of their life (Umeda, Daimon & Kondoh, 2005). Along these lines, the incorporation of new equipment features and capabilities occurs at a rapid pace (Kang & Schoenung, 2005), rendering the electronic equipment obsolete and necessitating the frequent need for replacement.

While the economic weight on consumers cannot be overlooked, the warnings regarding environmental pollution have triggered a sense of urgency in the global community. In particular, the detrimental effects of over-consumption along with the short life cycles of devices have led to unprecedented hazard in the environment. With the current rate of consumption, the planet requires one and a half year in order to renew the resources used by humans in one year (McLellan, Grooten & Almond, 2012). Furthermore, the extraction and consumption of raw materials skyrocketed from 36 billions of tonnes in 1980 to almost 70 in 2010 (Linster et al., 2015). This upward trend is still ongoing, with the extraction of 85 billions raw materials in 2020 and the projection to surpass 100 in 2030 (Lutz & Giljum, 2017). As we move forward, it is essential for the society to recognize the urgent need for more sustainable practices, by embracing circular economy policies.

To address these challenges, new strategies have been developed to prolong the lifespan of products. These strategies are categorized according to the life cycle phases of systems, namely Beginning of Life (BoL), Middle of Life (MoL) and End of Life (EoL). For each phase, certain methods offer unique opportunities, challenges and economic benefits. In case of EoL, four key actions are the most popular among others, namely remanufacture, refurbish, reuse and recycle (Khan, Mittal, West & Wuest, 2018).

After the implementation of stringent governmental regulations aimed to mitigate environmental pollution, the concept of 'life extension' has become the central point of discussions among scientists. Lifetime extension (LTE) is defined as the extended utilization period of a product, resulting in a slowdown of the flow of materials through economy (Bocken, De Pauw, Bakker & Van Der Grinten, 2016) and is regarded a reuse method during the EoL phase of a component. This further classification seeks to delve into the concept of reuse more comprehensively and investigate life extension in a more detailed manner.

This relatively new concept has led to confusion in distinguishing the terms maintenance and life extension. According to the definition, maintenance is defined as "any activity intended to retain or restore a functional unit in or to a specified state in which the unit can perform its required functions" (Wikipedia, 2023b). In contrast, lifetime extension is defined as "the process of restoring a used product to a better operational state and prolonging its remaining life" (Shafiee, Finkelstein & Chukova, 2011). In other words, life extension seeks to enhance a system with new requirements whereas maintenance intends to keep system's functionality to a certain level.

Despite this terminology difference, the mathematical approaches of LTE, considering deteriorating systems and increased requirements, align with those in the traditional maintenance policies. Furthermore, LTE involves the partial replacement of a system, a fact directly linked with the case of multi-component systems, discussed in the literature as part of maintenance strategies.

This paper aims to delve into the relatively new and under-explored topic of life extension of components within a system. Given the emerging attention of the field, with limited exploration of certain aspects in the past, there is compelling need for the its study. In essence, this paper seeks to provide insights into the following critical questions.

- How can the feasibility of implementing LTE be assessed in terms of costs and CO_2 emissions? Under what circumstances LTE is an applicable strategy? What are the challenges associated with its implementation?
  - How can the concept of life extension be formulated mathematically?
  - Why is it currently challenging to quantify the effects of this strategy on a complex multi-component system?
  - Is there any feasible method to calculate interdependencies among components?
  - Are there mathematical tools that can capture the expected lifetime of a system?
  - How can the measurement of CO_2 emissions be carried out during the implementation of the life extension strategy?

1.1 Problem Statement

The remarkable improvement in computer's performance throughout the years has been attained due to advancements in its individual components. For instance, between 2004 and 2023 the central processing unit (CPU) demonstrated a 25,000-fold improvement,with a projected upward trajectory in the coming years (PassMark, 2023). On the other hand, as every physical object, components are subject to deterioration. Within the existing literature, deterioration types are often classified based on their dependencies: economic, stochastic, and structural (Oakley, Wilson & Philipson, 2022). Given the intricate nature of computers where components degrade at different rates due to technical or technological reasons (Rachaniotis & Pappis, 2008), it would be reasonable to investigate the stochastic dependency among on them. Stochastic dependence arises when the condition or lifespan distributions of other components are influenced by the state of a particular component (Oakley et al., 2022).

In the standard benchmarks sites each component is labeled based on its performance, price and specifications: high end, high to mid, low to mid and low end. For this study the scope falls into the low end computers and components whereas the choice is based on the multiple benefits they offer for an average user such as affordability, easy-use and portability. In reality, a desktop computer requires the assembly of more than ten components, which interact each other through complex connections (Wikipedia, 2023a). However, the lack of consistent data for each component as well as the difficulty of creating a mathematical model for all of them prompts us to focus on assessing three main components of a computer: the central processing unit, video card and hard drive (CPU, GPU, HDD).

Manufacturers design computers with average lifespan varying around four and a half years. (Statista Research Department, 2023a). However, this period can be extended based on several factors. One prominent choice to prolong computer's life is the replacement of its components with more technologically advanced ones in order to be compatible with the latest innovations of technology. Peck (2016) indicated that a minimal increase of 1% of value added by economic activities related to a longer lifetime for products would have an aggregated effect of 7.9 billion EUR per year across the European economy.

Computers are one of the many examples of electronic devices associated with high levels of e-waste and their construction is a highly resource-intensive process. According to United Nations University, a desktop computer with monitor demands 1.8 tons of raw materials during manufacturing phase (Hoang, Tseng, Viswanathan & Evans, 2009) while Williams (2004) estimated the energy use during production up to a staggering 81% compared to the usage stage. As a result, the production and usage of computer accounts 2% of the total greenhouse gases (GHG) emission in the United States (Masanet, Price, Brown, Worrell et al., 2005). Among the greenhouse gases, the major culprit in climate change is considered to be carbon dioxide (CO_2). Since the era of industrialization, these gases have played a substantial role in global warming, with its consequences already beginning to manifest. The planet is in a critical state and decisive measures must be taken to prevent alarming consequences.

The primary goal of this paper is to present a comprehensive plan for PC users in case of their desktop computer no longer meets demand, by either replacing the whole computer or its individual components. When referring to the whole computer, the focus is on the three main components of a desktop computer: the central processing unit (CPU), the graphics processing unit (GPU) and the hard drive disk (HDD). The choice to study these components is motivated by their significant impact on incorporating the latest capabilities introduced in the market into a computer system.

The upgrade strategy involves the replacement of these three components from the period of their integration into a computer up to reaching to a certain condition. Throughout this period, the components may experience deterioration or remain unaffected, or they undergo to preventive replacement at a fixed state. Upon replacement, the components are replaced with new, enhanced counterparts, ensuring that the system always maintains a certain level of performance while it keeps the cost as low as possible.

The objective of this approach is to reduce the average costs during the replacement cycle. Insights derived from the upgrade strategy are intended to empower users in making informed decisions about upgrading their systems to better align with the latest technological requirements.

1.2 Carbon Dioxide Footprint

In the present era, technology companies have adopted the practice of officially keeping records about how their products impact the environment. This transparency serves dual purposes: it shows that companies are serious about being eco-friendly but also puts them in the conversation about environmental responsibility. From an economic point of view, embracing this strategy can provide a competitive advantage, enhancing the brand image and attractiveness of products, especially to environmentally conscious consumers.

This introduction serves as a crucial foundation for comprehending the scope of the upcoming section, aiming to estimate the optimal replacement policy for computer components by considering carbon dioxide emissions during the manufacturing phase. The driving force for this endeavor is rooted in the insights from Ferreboeuf et al. (2019), which reveals that a substantial 70-75% of the overall carbon emissions arise during the manufacturing phase for the case of a desktop computer.

However, it is essential to acknowledge a significant challenge in this pursuit—the scarcity of relevant data. Among supplier companies, there is substantial variation in the published amount of CO_2 of computers during manufacturing. For example, Dell reports a percentage of approximately 50% whereas Hewlett Packard (HP) indicates that the percentage varies around 30%. This discrepancy might stem for the fact that there are more available published data for higher-end models of Dell compared to those of HP, with emphasis on more budget-friendly computers. A naive explanation for this discrepancy could be that the higher the performance of a computer the greater its emissions to the environment.

Finding information, particularly for low-end computers, proved to be a formidable task. This challenge is magnified by the fact that major technology companies focus their production efforts on higher-end models, constructing only a limited number of low-end computers. Within the context of this discussion, the term "low-end computers" primarily refers to laptops. This scarcity of data underlines the need for more comprehensive and accessible information to inform sustainable decision-making, especially concerning the environmental impact of computer components.

The plan aims to formulate an effective strategy for environmentally conscious PC users facing deficiencies in their computer requirements. The proposed approach involves replacing components with upgraded versions to attain a certain level of performance based on their carbon emissions rather than costs. The main components studied for this research - CPU, GPU, HDD - align with those in optimal replacement policy discussed above, given their impact on the CO_2 emissions during PC construction and their significant contribution to overall computer performance. The method emphasizes quantifying the average carbon dioxide emissions associated with the replacement cycle of computer components.

1.3 Relevance

The research on the optimal upgrading of computers and their components investigates a topic that is not only insufficiently explored by scientists but also with plenty of room for further development. Currently, literature refers to the maintenance on 'multi-component system' as a system with more than one component but the limitation lies on most of the published papers, which deal with cases of solely two components. In reality, machines consist of multiple components that require frequent maintenance in order to avoid down times, oversized costs as well as to keep a certain service level. The same problem applies in the study of personal computers, where a wide gap in the literature must be addressed.

In addition to the misconception regarding the number of components, only a limited number of papers address lifetime extension strategies. This is primarily because decision-making proves to be not entirely explainable but also the lack of prior experiences in implementing these types of strategies (Ziegler, Gonzalez, Rubert, Smolka & Melero, 2018). The issue becomes larger, where even fewer papers present mathematical models to describe life extension problems.

The heightened awareness of environmental protection and the potential impacts associated with the entire life cycle of products—from manufacturing and distribution to consumption and end-of-life—has stimulated increased interest in developing methods for a deeper understanding. Nevertheless, the existing literature on assessing the environmental impact of component replacement within a mathematical framework remains relatively sparse. This study aims to address this gap by implementing a replacement policy centered on reducing CO_2 emissions.

1.4 Outline

The remainder of this paper is organized as follows: Section 2 introduces the literature review. Section 3 outlines the methodology, beginning with the problem setup in 3.1. Section 3.2 introduces the implemented mathematical approach, with a specific focus on the Markov Decision Model. Additionally, 3.3 presents a subtle modification to the original problem, while 3.4 proposes a heuristic algorithm. Finally, 3.5 presents the framework for assessing carbon dioxide emissions. In Section 4 the corresponding results are displayed. The paper concludes in Section 5, followed by Section 6 which delves into future additions and discussions.

Literature Review

This chapter presents the relevant literature of the research topic. In Section 2.1 a detailed examination on the life extension strategies and implications applied to computers are introduced. Section 2.2 delves into the environmental aspect, focusing on the assessment of CO_2 footprint through life extension.

2.1 Desktop Computer Literature

In recent decades, the multifaceted significance of systems maintenance have attracted more and more attention from both researchers and industrial sector. Mobley (2002) underlined that maintenance costs could represent up to 60% of the total operating costs. On the same wavelength, Gräber (2004) expressed that maintenance costs could reach up to 30% of the production costs in a power plant.

However, the concept of lifetime extension has diverted the attention of scientists. Fontana et al. (2021) emphasized the implementation of LTE as of highest importance in the field of predictive maintenance. Moreover, Budzinski, Bezama and Thrän (2020) illustrated the benefits of LTE, particularly in reducing environmental hazards. In a chronological progression, scientists initially acknowledged the importance of maintenance and then redirected their endeavors towards LTE. This strategic shift aims to minimize costs and environmental impacts, among other objective goals.

In the literature, an interested reader could find numerous papers related to maintenance systems with a single component. The vast majority of them aim on reducing costs or shortening failures times of a system. Nonetheless, modern systems are way more complex and they consist of multiple components. Two major reasons pivoted scientists' efforts to study multi-component systems: First, the rapid growth of computer technology and the development of analytical techniques gave the opportunity for detailed examination of these systems. Secondly, it turns out that components interact to each other in some multi-component systems. The latter proved to be a crucial element on maintenance decisions (Dekker, Wildeman & Van der Duyn Schouten, 1997).

Although many maintenance planning strategies have been studied thoroughly at latest years, such as corrective maintenance (occurs upon failure of the system) or preventive maintenance (either on a fixed pre-specified age or after T periods of time), the topic of upgradeability is an emerging and relatively new strategy (Khan et al., 2018). The following three papers investigate the topic of lifetime extension: Ziegler et al. (2018) administered interviews with onshore wind turbine firms in four European countries, namely Germany, Denmark, Spain and UK. Their result showed that end-of-life solutions will record a considerable market in the next years. van Noortwijk and Frangopol (2004) made a comparison of two maintenance models: a condition-based and a reliability-based. Both of them aim to find the optimal balance between reliability and life-cycle cost of a deteriorating civil infrastructure. Yang, Frangopol and Neves (2006) proposed a model which assesses the reliability of deteriorating bridges structures based on lifetime functions. Despite the growing attention given to the term of 'upgradability' (Khan et al., 2018), a generic qualitative modelling for life extension is lacking.

An alternative term used to describe LTE is the concept of 'mid-life upgrade', a terminology commonly employed in military. Three papers discussed below incorporate this approach in their research. Kusumo and Sinha (2002) proposed a plan to enhance the performance of military air-crafts to meet the demand for improved mission capabilities. However, this paper lacks of mathematical framework, relying instead on qualitative methods. Jonnalagadda, Sinha, Hoffmann and Schrage (2005) conducted a design analysis of military helicopters in order to develop their on board mission systems and address the demand for technological upgrades. Similarly to the previous paper, this work does not integrate mathematical modelling into its methodology. On the other hand, Nijland, Atyeo and Sinha (2004) assess the flight performance of an upgraded design for helicopters. This paper implements simulation to draw conclusions. Nonetheless, they didn't manage to come up with solid conclusions due to inaccurate helicopter specifications.

In the present era, engineering systems are subject to degradation stemming from both usage and environmental influences. As components interact within a system, the degradation or failure of one component affects the others. De Jonge and Scarf (2020) spotted a gap in existing research, revealing that only a few papers investigate the influence of stochastic dependency on physical entities and all of them miss important information: They either do not consider condition information or limit their work on only two components (Oakley et al., 2022). Keizer, Flapper and Teunter (2017) observe that studies typically isolate dependencies, yet in reality, systems often involve a combination of different dependency types. Equally significant is the oversight in many papers regarding the factor of uncertainty; these studies tend to assume fixed threshold values, overlooking the dynamic nature of component deterioration intervals. This oversight results in obtained findings that paint a distorted picture of the process, potentially leading to misleading conclusions. The three papers highlighted above underline the challenges associated with accurately measuring these dependencies.

Another gap in the literature is spotted on the limited number of papers that utilize mathematical methods for computer upgrading, coupled with a scarcity of research on the implemented methods and derived conclusions. To be more specific, only two papers have been found implementing operations research approaches in the context of computer setups. Generally, as the number of examined components increases, the application of precise mathematical approaches for optimal replacement becomes more challenging. Rachaniotis and Pappis (2008) presented a PC case aimed at determining the optimal upgrade policy to maximize overall computer performance, considering three components. They applied stochastic dynamic programming but this method is computationally intensive. Their findings suggest that if a greater number of components are studied, the problem may be susceptible to the curse of dimensionality, an issue raised from the exponential increase in the amount of data. On the other hand, Zafiropoulos and Dialynas (2004) implemented a meta-heuristic algorithm called simulated annealing in order to find the optimal system structure for an electronic device.

2.2 Environmental Assessment

With the tremendous prosperity of the electronics industry, an increasing awareness about the environmental impacts related to mass production, electricity use and waste management has been raised. These concerns resonate not only with scientists and environmentalists but also within corporate circles (Choi, Shin, Lee & Hur, 2006). In order to capture the environmental aspects of a product during its lifetime, they developed a methodology called life cycle assessment (LCA). LCA is a systematic approach to assess and quantify the environmental performance associated with all stages of product creation, processes and activities (ISO (International Organization for Standardization), n.d.) and is used in fields like marketing, product selection and strategic planning (Weidema, Wenzel, Petersen & Hansen, 2004).

Due to the growing importance of environmentally friendly practices, many papers have been published using Life Cycle Assessment (LCA), especially when looking at electronic products like TVs, laptops, and smartphones. Most of the published work investigate the complete life cycle of a product, from extraction of resources to the end-of-life disposal. For example Duan, Eugster, Hischier, Streicher-Porte and Li (2009) conducted LCA to desktop computers in China from global level. Figures 2.1 and 2.2 in this study provide key insights. Figure 2.1 makes it clear that manufacturing and use play a dominant role in causing environmental damage during the life cycle. Moreover, it highlights that the most significant impacts are on human health and resources. This pattern holds true for carbon emissions per phase as well, as shown in Figure 2.2, where usage and manufacturing take precedence.

On the other hand, Andrae and Andersen (2010) assess the consistency of studies for consumer electronics. They point out significant deviations among LCA studies for laptop and desktop computers, but also consensus for smartphones and TVs. In a recent study, Loubet et al. (2023) organized LCA focusing on single-board computers (SBC) and desktop computers (PC) within the higher education context. The outcomes of their investigation emphasize the eco-friendly superiority of integrating SBCs for student use, highlighting a significantly reduced environmental footprint linked to this alternative.

A relatively overlooked aspect in the literature pertains to the sustainable analysis of electronic device upgrades. The existing body of literature lacks comprehensive coverage on the sustain-

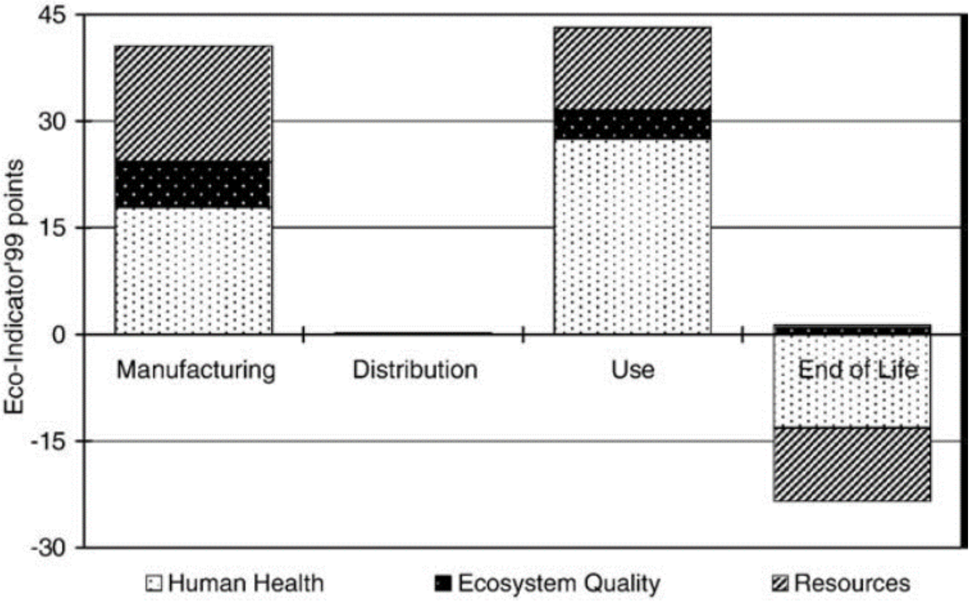

Figure 2.1: Environmental impact per phase (Duan et al., 2009)

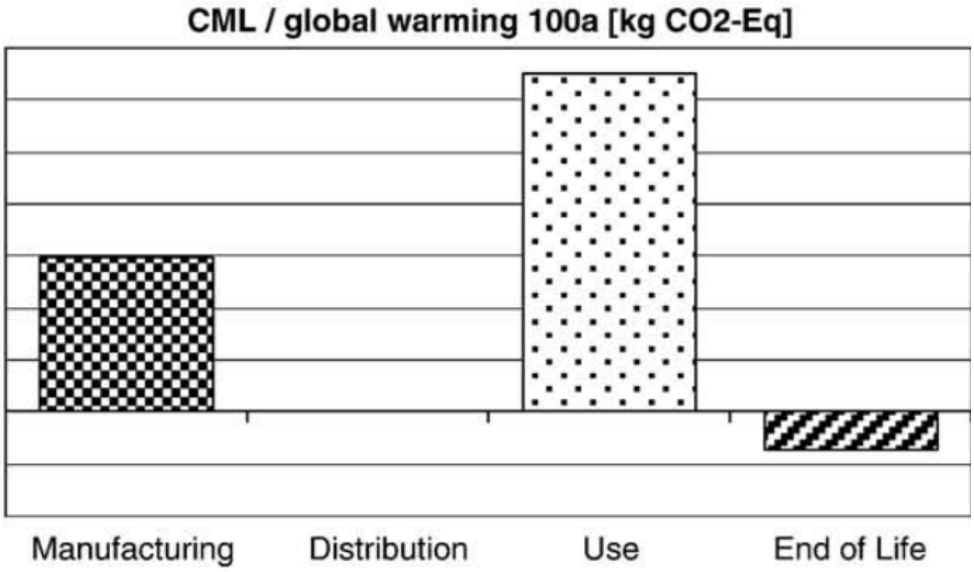

Figure 2.2: CO_2 emissions per phase (Duan et al., 2009)

ability assessment of computer components, with only a handful of papers addressing various facets. Fatimah and Biswas (2016) conducted a survey on small to medium sized computer re-manufacturing enterprises by assessing economic, energy, reliability and unemployment criteria and conducting life cycle assessment. Also, Han et al. (2021) introduced a mathematical framework designed to optimize both greenhouse gas emission savings and profits through the reuse and recycling of end-of-life computers. However, a notable gap persists, as no identified paper specifically addresses optimal upgrade policies with the explicit aim of minimizing the CO_2 footprint, diverging from the traditional focus on cost considerations.

Methodology and Data

In this chapter, a comprehensive overview of the implemented methodologies is provided. Section 3.1 describes the general framework of the problem. In Section 3.2 the implementation of Markov Decision Model is outlined. Moving forward, Section 3.3 delves into a case featuring dependent replacement probabilities whereas Section 3.4 introduces a heuristic algorithm developed for comparative assessments. Concluding this exploration, Section 3.5 illustrates a mathematical framework focused on evaluating the CO_2 footprint resulting from the replacement policy.

3.1 Model and Data

The cornerstone of constructing a Markov Decision Model lies in accurately defining the mathematical framework for the problem at hand. For this context, three components (CPU, GPU, HDD) of a desktop computer are considered. The model is described with finite state and action spaces, a design choice to insert practical functionality to the problem as well as to avoid computational issues. In the broader context, failures of system components tend to adhere to either Weibull or exponential distributions (Zafiropoulos & Dialynas, 2004). For electronic products specifically, their failure rates tend to stabilize over time, allowing for a reliable approximation through the use of an exponential model (Rachaniotis & Pappis, 2008). In this work, exponential distribution is utilized to describe the deterioration level of the components. The goal is to not only broaden model's utility but also simplify computational processes.

While the typical lifespan of a desktop computer is generally estimated at 5.29 years in 2023 (Statista Research Department), this paper adopts a functional lifetime of 5 years for the sake of practicality. The component performance, denoted as $P_i(t)$, represents the relative performance of component i compared to its latest version at time t, determined by the deterioration rate λ. It is mathematically expressed as $P_i(t) = e^{-\lambda_i t} P_i(0) = (\alpha_i)^t P_i(0)$, where $P_i(0) = 100\%$ signifies the initial performance of component i at $t = 0$ and $0 < P_i(t) \leq 1$. The parameter $\alpha_i = e^{-\lambda_i}$ denotes the geometric performance decrease of component i while α_i^t expresses the deterioration level of component i at time t. The system performance P^{sys} is also measured in relation to a complete new system. It is defined as the weighted average of the three individual component performances, expressed as $P_{sys}(t) = \frac{\sum_{i=1}^{3} w_i P_i(0) e^{-\lambda_i t}}{3}$. Minimum performance requirements are set not only for each component but also for the entire system: $P^{sys} \geq \theta^{sys}$ and $P_i(t) \geq \theta_i$ for $i \in 1, 2, 3$.

Recognizing the complexity involved in controlling a continuous state system, a strategic decision is made by discretizing the state space. This approach simplifies the analysis by segmenting the state space into 10 equal pieces, each representing a half-year interval with $t \in 1, 2, ..., 10$. The parameter λ_i signifies the degradation value of component i and is independent of the state of component. To illustrate discretization in probabilities, notation p^i_n is introduced. Component i experiences deterioration from state n to state $n+1$ with probability $P^i_{n,n+1} = (1 - p^i_n)$ or remains at state n with probability $P^i_{n,n} = p^i_n$. Therefore, the system is observed on fixed periods of time and there are two possible transitions from each state.

To establish a connection between the deteriorating performance of the system and the transition probabilities, it is assumed that the expected performance level $\alpha_i P_i(t)$ remains constant and independent of the states. This assumption is modelled by considering partial transitions from states rather than actual. In mathematical notation, it is formulated as

$$\alpha_i P_i(t) = p^i_n P_i(t) + (1 - p^i_n)\beta P_i(t)$$

In this equation, the expected performance of component i at time t is expressed as the probability of being at state n with probability p^i_n and having performance $P_i(t)$ and the probability of deteriorating to state $n+1$ with probability $1 - p^i_n$ with the corresponding performance. By solving in terms of p^i_n, it holds that $p^i_n = \frac{\alpha_i - \beta}{1-\beta}$. Clearly, the transition probabilities remain constant as the state evolve and depend only on parameter α_i of each component as well as parameter β. Here, β denotes the step size of deterioration, it is constant for all components and is independent of the states. Determined arbitrarily by the decision maker, β influences the magnitude of transition probabilities and consequently impacts the average replacement costs.

This approach not only eases the complexity associated with continuous state systems but also provides a structured framework for a more comprehensible examination of the system's dynamics. Lastly, as regard to the action space A, its scope can be narrowed down to the decision of whether a component needs to be upgraded. Therefore, notation $a \in A$ for component i with $i \in 1, 2, 3$ denotes that

$$a_i = \begin{cases} 1 & \text{if component } i \text{ is upgraded} \\ 0 & \text{otherwise} \end{cases}$$

The computational complexity of this problem depends on the number of variables introduced. Indeed, the number of variables grows exponentially with the number of components in the problem. To make it feasible, the deterioration level for all components ranges from 0 to 9. It is assumed that deterioration for each component can be measured in time. Along these lines, Jiajian Yan (2023) proved that the component with the slowest deterioration in a desktop computer is CPU. He also demonstrated that in general HDD has the highest deterioration, followed by GPU among the examined components. However, only for low-end computers, he showed that GPU deteriorates faster than HDD. The latter assumption is considered in this work as well. Mathematically, it is expressed as $\lambda_{CPU} < \lambda_{HDD} < \lambda_{GPU}$.

Furthermore, a constant cost is associated with each component replacement. Since this paper refers to the case of low-end computers, each cost is obtained as the average price from a set of low-end components. All the data utilized in this analysis were sourced from the thesis project of Jiajian Yan (2023). Table 3.1 presents a summary of the relevant values in the dataset.

Table 3.1: Data for prices

Item	Sample size	# of Low-End Comp.	Aver. Price (USD)
GPU	146	36	170 $
CPU	585	144	70 $
HDD	387	98	30 $

The choice to focus on GPU, CPU, and HDD as the primary components for study is driven by their crucial roles in shaping the overall performance and functionality of a computer system. Specifically, CPU stands as the central processing unit, functioning as the computer's brain. Simultaneously, GPU plays a vital role in rendering graphics and processing videos, while HDD serves as the long-term storage location for data. Collectively, these components are considered core elements essential to the operation and functionality of a computer system.

The importance of upgrading/replacing components aims to proactively mitigate the overall deterioration and subsequently failure of a computer, leading to extended lifespan of the system. To formulate the optimal upgrade strategy, a Markov Decision Model is implemented to address the decision making. The central focus is on determining which components should be upgraded and when a complete system replacement is needed, under the condition that the average replacement costs are minimized.

3.2 Replacement Model

To avoid critical failures that can result in production losses, planned replacement of most operating units is essential. Replacement activities undertaken before a failure occurs are termed preventive replacement (PR), while those conducted after a failure are known as corrective replacement (CR) (Nakagawa, 2005). The definition of failure is rather contextual, varying depending on a problem. In this paper, the term ”failure” specifically denotes performance degradation, resulting in a decline in the efficiency or reliability of a system over time, rendering it dysfunctional.

The problem addressed in this paper is formulated as a Markov Decision Model, a fundamental approach in maintenance theory that falls also into the scope of preventive replacement. This affinity is evident in the optimal replacement planning of components, strategically implemented to prevent the system from a non-functional condition. This method will be applied to ensure the continued operational efficiency of our system.

Mathematically, the Markov process comprises the states of all three components $i_1, i_2, i_3 \in I = I_1 \times I_2 \times I_3$. At each combination of states, eight actions can be taken: Do nothing ($a = 0$), replace CPU ($a = 1$), component HDD ($a = 2$), replace GPU ($a = 3$), joint replacement of CPU and HDD ($a = 4$), joint replacement of CPU and GPU ($a = 5$), replacement of HDD and GPU ($a = 6$) or replacement of all three components ($a = 7$). In short, set A is described as

$$A(i_1, i_2, i_3) = \begin{cases} \{0\} & \text{if } i_1 \notin \{0, M\} \text{ and } i_2 \notin \{0, M\} \text{ and } i_3 \notin \{0, M\}, \\ \{1\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \notin \{0, M\} \text{ and } i_3 \notin \{0, M\}, \\ \{2\} & \text{if } i_1 \notin \{0, M\} \text{ and } i_2 \in \{0, M\} \text{ and } i_3 \notin \{0, M\}, \\ \{3\} & \text{if } i_1 \notin \{0, M\} \text{ and } i_2 \notin \{0, M\} \text{ and } i_3 \in \{0, M\}, \\ \{4\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \in \{0, M\} \text{ and } i_3 \notin \{0, M\}, \\ \{5\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \notin \{0, M\} \text{ and } i_3 \in \{0, M\}, \\ \{6\} & \text{if } i_1 \notin \{0, M\} \text{ and } i_2 \in \{0, M\} \text{ and } i_3 \in \{0, M\}, \\ \{7\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \in \{0, M\} \text{ and } i_3 \in \{0, M\}, \end{cases}$$

where constant M denotes the maximum age (deterioration) a component can reach. The transition matrix $p(i_1, i_2, i_3)(j_1, j_2, j_3)(a)$ denotes the probability of going from state (i_1, i_2, i_3) to state (j_1, j_2, j_3) under action a. The probabilities denote independence not only in deterioration among components, but also in replacements actions. Moreover, the renewal points of the process are defined as instances where the system undergoes preventive replacement. Consequently, the process starts anew, resulting in an identical optimal upgrade strategy.

$$p(i_1, i_2, i_3)(j_1, j_2, j_3)(0) = \begin{cases} (1 - p_n^1)(1 - p_n^2)(1 - p_n^3) & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ (1 - p_n^1)(1 - p_n^2)p_n^3 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ (1 - p_n^1)p_n^2(1 - p_n^3) & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ (1 - p_n^1)p_n^2 p_n^3 & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^1(1 - p_n^2)(1 - p_n^3) & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^1 p_n^2(1 - p_n^3) & \text{for } j_1 = i_1, j_2 = i_2, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^1(1 - p_n^2)p_n^3 & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^1 p_n^2 p_n^3 & \text{for } j_1 = i_1, j_2 = i_2, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \end{cases}$$

The probabilities indicate the likelihood of component i deteriorating from state n with probability $1 - p_n^i$ or remaining at the same state (without deterioration) with probability p_n^i. Once state $M = \theta_1 = \theta_2 = \theta_3$ is reached, it implies that the component has exceeded its performance threshold, necessitating an upgrade. Upon replacement, the components start with deterioration level 0. A constant probability for all states, $1 - p_0^i = 1$ with $i \in \{1, 2, 3\}$, implies the release of a new version of component i and $p_0^i = 0$ otherwise. It is assumed that replacements require negligible amount of time to occur. For the rest of the actions, the corresponding probabilities are presented in Appendix B.1.

The cost for each combination (i_1, i_2, i_3) depends only on the action taken. Therefore, each cost depends on the replaced item per action. Note that discount factor $\alpha = 0.9$ is assumed on the joint replacement of two components while factor $\gamma = 0.8$ is set to describe the full replacement of all three components. It is assumed that the full replacement is cheaper than replacing each component separately ($c_{full} < \sum_{i=1}^{n} c_i$). This assumption is grounded in the potential economies of scale in the model, arising from the introduction of completely new systems. The following scheme describes the structure of replacement costs.

$$c(a) = \begin{cases} 0 & \text{if } a = 0 \\ c_1 & \text{if } a = 1 \\ c_2 & \text{if } a = 2 \\ c_3 & \text{if } a = 3 \\ \alpha(c_1 + c_2) & \text{if } a = 4 \\ \alpha(c_1 + c_3) & \text{if } a = 5 \\ \alpha(c_2 + c_3) & \text{if } a = 6 \\ \gamma(c_1 + c_2 + c_3) & \text{if } a = 7 \end{cases}$$

with c_i denoting the purchasing cost of the new version and installation cost in the computer of component i.

After establishing all the input parameters for the problem, the mathematical formulation is introduced in Formulation 3.2. The Markov decision chain can be characterized as *unichain*, signifying a subset of states with mutual communication. Consequently, the optimal policy will have no two disjoint closed sets and an LP formulation can be used to determine it. The formulation is inspired from the book of Tijms (2003) and aims to minimize average costs. Each variable $x^a_{i_1 i_2 i_3}$ in the formulation represents the long-run probability when the system is in state $(i_1, i_2, i_3) \in I$ under action $a \in A$.

$$\min \sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{i_3 \in I} \sum_{a \in A} c(a) x^a_{i_1,i_2,i_3} \tag{3.1a}$$

$$\text{subject to} \sum_{a \in A} x^a_{i_1,i_2,i_3} = \sum_{j_1 \in I} \sum_{j_2 \in I} \sum_{j_3 \in I} \sum_{a \in A} p_{(j_1,j_2,j_3)(i_1,i_2,i_3)}(a) \cdot x^a_{j_1,j_2,j_3} \quad \forall i_1, i_2, i_3 \in I \tag{3.1b}$$

$$\sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{i_3 \in I} \sum_{a \in A} x^a_{i_1,i_2,i_3} = 1 \tag{3.1c}$$

$$x^a_{i_1,i_2,i_3} \geq 0 \quad \forall i_1, i_2, i_3 \in I, \forall a \in A \tag{3.1d}$$

The objective function 3.2a minimizes the total average costs. Constraint 3.2b expresses the balance equation: The inflow to a specific set of states must equal the outflow. Constraint 3.2c ensures that the probabilities sum up to one. Lastly, 3.2d indicates the non-negative nature of the decision variables.

3.3 Introducing Dependency

As previously mentioned, components within systems often exhibit interdependencies, a common occurrence in real-world scenarios. Capturing these dependencies poses a challenge due to the difficulty in accurately estimating probabilistic dependencies. Furthermore, when multiple components fail simultaneously, determining the optimal sequence for repairs becomes critical, to provide the system with the most favorable characteristics (Thomas, 1986).

The components of a computer interact with each other to achieve a certain performance level. This interaction is crucial because any decrease in the efficiency of one component can have a downgrading effect on overall performance. As a result, a complex network of dependencies intertwines all these components. The key challenge in this aspect lies in the difficulty of quantifying these dependencies accurately. To date, researchers have relied on assumptions to make their results as realistic as possible. Furthermore, scientific literature on the interdependence of computer components remains unexplored.

For this paper, the assumptions are made based on the condition of CPU and are presented below in bullet points. This choice is rooted in its comparatively economical price in contrast to other components. Foremost, it is acknowledged as the most effective method to increase PC performance (The TechSiting, 2018) with the lowest yearly degradation rate among the examined components (Jiajian Yan, 2023). As already discussed, it holds that $\lambda_{CPU} < \lambda_{HDD} < \lambda_{GPU}$.

- If CPU deteriorates, all the other components also deteriorate
- If HDD deteriorates, but not CPU, then GPU deteriorates
- If CPU and HDD do not deteriorate, then GPU can deteriorate

To incorporate the aforementioned information, the transition probabilities from MDM need to be adjusted. The rest of the problem remains unchanged.

$$p(i_1, i_2, i_3)(j_1, j_2, j_3)(0) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^1 - p_n^2 & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^2 - p_n^3 & \text{for } j_1 = i_1, j_2 = i_2, j_3 = i_3 + 1, i_1, i_2, i_3 \notin \{0, M\} \\ p_n^3 & \text{for } j_1 = i_1, j_2 = i_2, j_3 = i_3, i_1, i_2, i_3 \notin \{0, M\} \end{cases}$$

Initially, the matrix represents a scenario where all components experience deterioration, with the probability of degradation set to the lowest degradation probability among components, namely $1-p_n^1$, considering that $1-p_n^3 > 1-p_n^2 > 1-p_n^1$. If only HDD and GPU degrade, excluding

CPU, this event occurs with probability $p_n^1 - p_n^2$. Similarly, if only GPU deteriorates while other components remain unaffected, the probability is $p_n^3 - p_n^2$. Lastly, if none of the components deteriorate, this happens with probability $1 - p_n^3$. This pattern applies to replacement actions as well. For the remaining actions specified in the problem, the transition probabilities are detailed in B.2. The matrix illuminates the symbiotic relationship among components, highlighting that CPU's condition directly affects the condition of others.

3.4 Heuristic Approach

In many real-world scenarios, employing heuristic algorithms yields results that are close to optimal, particularly when dealing with large-scale problems. In the context of this paper, the utilization of a heuristic algorithm is motivated by two primary considerations. First, it enhances interpretability, making the results more readable to individuals without a strong mathematical background. This stands in contrast to the LP formulation, which may be less intuitive for non-mathematical audiences. Secondly, the adoption of heuristic algorithms allows for effective comparison. Such a comparative analysis aids in assessing the 'quality' of the solution obtained through heuristics as opposed to the solution derived from the LP formulation.

For the reasons stated above, the adopted heuristic is referred to as the Power-of-Two (PoT) policy. This algorithm is considered appropriate for deterministic setups (Ekinci & Ornek, 2007) and finds application in manually conducted schedules. It's uniqueness lies in the fact that cycle times are restricted to be powers of two (Ouenniche & Boctor, 2001). More specifically,

$$T = 2^m q \tag{1}$$

with m being an integer number and q the basic period. A significant benefit of utilizing these cycle times is the attainment of relatively straightforward cyclic schedules (Axsäter, 2015). To illustrate the implementation of this approach, Algorithm 1 provides a pseudocode representation.

Algorithm 1 Power of Two Policy

Set λ_{CPU}, λ_{HDD}, λ_{GPU} as expected deterioration for CPU, HDD, GPU
Set c_{CPU}, c_{HDD}, c_{GPU} as replacement costs for CPU, HDD, GPU
Set joint discount factor $\alpha \leftarrow 0.9$, full discount factor $\gamma \leftarrow 0.8$
Set expected lifetime: $E(\text{CPU}) = \frac{1}{\lambda_{\text{CPU}}}$, $E(\text{HDD}) = \frac{1}{\lambda_{\text{HDD}}}$, $E(\text{GPU}) = \frac{1}{\lambda_{\text{GPU}}}$
Define $T, q \in \mathbb{Z}$
$T \leftarrow \max(E(\text{CPU}), E(\text{HDD}), E(\text{GPU}))$
Fix $m \in \mathbb{Z}$ and find relevant q
Round down q if necessary
Set q equal to replacement times for the rest
Fit multiples of q to T
Find number of individual and joint replacements within renewal cycle
cost1 $\leftarrow$ # of Individual Replacements $\times$ Individual Replacement Costs
cost2 $\leftarrow$ # of Joint Replacements $\times$ Joint Replacement Costs
cost3 $\leftarrow$ # of Full Replacements $\times$ Full Replacement Costs
Average replacement cost as a PoT $\leftarrow \frac{\text{cost1}+\text{cost2}+\text{cost3}}{T}$

In essence, the algorithm initiates by computing the expected lifespan of components, determined by their deterioration rate (λ). Equation (1) establishes the common cycle T as the highest lifetime among the components. By fixing $m \in \mathbb{Z}$, the corresponding parameter q is found. In case of q is not integer, then it is rounded down. Each q signifies replacement instances for the remaining components within the range from the shortest lifetime to T. Joint replacements are preferred in such cases, leveraging discounts on the replacement actions. Lastly, the average replacement costs are calculated based on the renewal cycle length, denoted as the common cycle T.

3.5 CO_2 Emissions: Problem Formulation and Solution Approach

To measure the overall carbon emissions, our focus turns to the examination of the HP Pavilion Desktop PC TP01, a desktop computer manufactured by Hewlett Packard (HP). It stands out as one of the limited low-end computers for which we managed to gather precise data, directly published by HP (Table 3.2). However, the data provided lacks a breakdown of carbon emissions for each individual component. To address this limitation, we turn to the insights presented by Loubet et al. (2023), as depicted in Table 3.3(a) and 3.3(b).

Table 3.2 illustrates the percentages and quantities of CO_2 emissions at each life stage of a desktop computer. In our example, the total carbon footprint is measured at 709 Kg over its entire life cycle. From this amount, one-third is attributed to the manufacturing stage while the remaining two thirds are coming from the usage phase. It's worth noting that the other life stages make no contribution to carbon emissions.

Table 3.2: CO_2 Kg eq. emissions / life stage (Hewlett Packard (2021))

Life stage	Percentage (%)	Amount (Kg)
Manufacturing	33.7 %	239
Distribution	0 %	0
Use	66.3 %	470
End-of-Life	0 %	0
Total	100 %	709

The bar chart in Figure 3.1 illustrates carbon emissions across manufacturing phase for various processes and components of a desktop computer. To break it down, the leftmost bar displays an overview of carbon emissions throughout each phase of life cycle, highlighting 'components' as the primary contributors. The middle graph delves deeper into a component-level analysis, revealing that approximately half of the emissions originate from motherboard among other components. Last but not least, the right column depicts the role of individual sub-components of the motherboard in the emissions.

Upon careful review of the middle graph, an absence is observed in the direct reporting of the

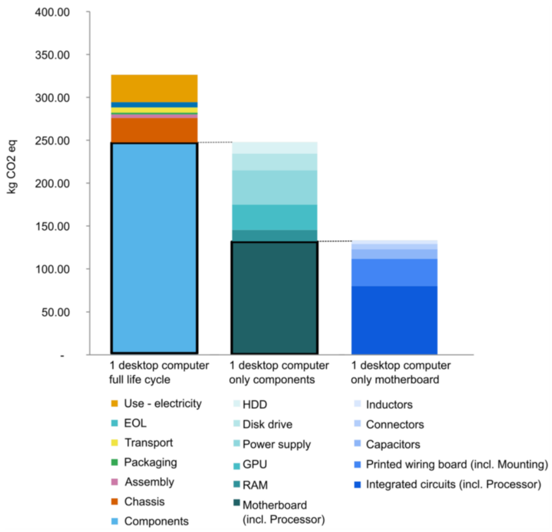

Figure 3.1: Left column illustrates CO_2 emissions across the life cycle of a desktop computer, with EoL components emerging as predominant contributor. Middle column represents classification by components. Right column provides a detailed breakdown into the sub-components of motherboard. (Loubet et al., 2023)

Central Processing Unit (CPU). However, it can be considered as the value of 'Integrated circuits (incl. Processor)', displayed on the right bar of the graph. Hence, the percentage emissions attributed to Integrated circuits serves as the percentage of CPU for this work. Given that CPU is installed on the motherboard and replaceable, unlike motherboard itself, the graph highlights its environmental impact. According to the chart, CPU emerges as the most pollutant component, constituting around 30% of the total emissions and 60% of motherboard's emissions. In comparison to other components addressed in the problem, CPU is two and five times more pollutant than GPU and HDD respectively. The corresponding percentages are more clear in sub - Tables 3.3(a) and 3.3(b).

The data clearly reveal that certain components, such as disk drive or power supply, exhibit higher pollutant levels than HDD and could be investigated. However, the selection of HDD is not solely ruled by its carbon emissions. Instead, it also plays a crucial role in enhancing computer performance through upgrade, while others don't. Despite its relatively lower emissions, HDD is a critical part contributing massively to the overall performance of the computer.

The selection of these components to study is driven by their major contribution to carbon dioxide emissions, the availability of relevant data and their contribution to overall performance. Numerically, the details are visually depicted in Table 3.3. It is important to note that both Tables 3.2 and 3.3 display values of carbon dioxide equivalent (CO_2−eq) instead of carbon dioxide CO_2. While these two terms are related, they represent slightly different concepts. The latter refers to a a specific greenhouse gas while the former is a metric quantifying the potential of various greenhouse gases contributing to global warming, expressed in an equivalent amount of CO_2. For the purpose of this paper, the two terms are treated as equivalent.

(a) CO_2 eq. emissions / component

Item	Percentage (%)
Motherboard	**50**
GPU	**15**
HDD	**6**
RAM	5
Disk drive	8
Power supply	16
Total	100

(b) CO_2 eq. emissions / sub-component of motherboard

Motherboard sub-components	Percentage (%)
Integrated Circuits	**30**
Printed wiring board	10
Capacitors	5
Connectors	3
Inductors	2
Total	50

Table 3.3: Comparison of CO_2 eq. emissions for different components (Loubet et al. (2023))

Table 3.3(a) highlights that the primary contributors to most manufacturing related pollutants are the motherboard, GPU and the power supply, accounting for approximately 80% of the total emissions while Table 3.3(b) delves into a further categorization of the sub-components within the motherboard. It is crucial to note that this classification relies on a single paper (Loubet et al., 2023) and may limits the depiction of reality. Potential inconsistencies in the data underline the need for further research and sources to ensure better assessment.

In Table 3.4, the 'Manufacturing' column quantifies the CO_2 emissions for each component. Components studied for this problem are highlighted with bold letters, along with their respective quantities. The last row presents the total carbon emissions of an entire desktop computer, obtained by summing the emissions of all components. Additionally, the 'Percentage' column displays the contribution percentage of each part during the manufacturing process.

While the total actual manufacturing emissions stand at 239 Kg, the problem utilizes different value. The decision is driven from the significant numerical difference in emissions between individual and full replacement scenarios. Selecting the true number might yield misleading results, favoring individual replacements over full replacement due to the significant gap in the values. Furthermore, since this study involves a three-component system, considering the actual emissions might introduce a disparity in the model. Therefore, for this problem, full replacement emissions are considered the summation of individual emissions from CPU, GPU and HDD, resulting in a value of 121.89 Kg.

Mathematically, the problem works as follows. Formulation 3.2 is implemented as detailed in Section 3.2. The objective of decision-making is to minimize the overall replacement costs over the renewal cycle. In that way, the model creates a solution path constructed from the non-zero decision variables. Each variable denotes the probability of replacing at state (i_1, i_2, i_3) under action a. As a reminder, the decision variables are depicted as $x^a_{i_1,i_2,i_3}$ and the Formulation 3.2 is presented below.

Table 3.4: Carbon emission values of components

Item	Manufacturing (in Kg)	Percentage (%)
Motherboard	119.5	50
CPU	**71.7**	30
Print.Wir.Board	23.9	10
Capacitors	11.95	5
Connectors	7.17	3
Inductors	4.78	2
GPU	**35.85**	15
HDD	**14.34**	6
RAM	11.95	5
Disk drive	19.12	8
Power supply	38.24	16
Total	239	100

$$\min \sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{i_3 \in I} \sum_{a \in A} c(a) x^a_{i_1,i_2,i_3} \tag{3.2a}$$

$$\text{subject to} \sum_{a \in A} x^a_{i_1,i_2,i_3} = \sum_{j_1 \in I} \sum_{j_2 \in I} \sum_{j_3 \in I} \sum_{a \in A} p_{(j_1,j_2,j_3)(i_1,i_2,i_3)}(a) \cdot x^a_{j_1,j_2,j_3} \quad \forall i_1, i_2, i_3 \in I \tag{3.2b}$$

$$\sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{i_3 \in I} \sum_{a \in A} x^a_{i_1,i_2,i_3} = 1 \tag{3.2c}$$

$$x^a_{i_1,i_2,i_3} \geq 0 \quad \forall i_1, i_2, i_3 \in I, \forall a \in A \tag{3.2d}$$

At that point, the CO_2 emissions are introduced in the problem. For each replacement action, the aforementioned decision variables are multiplied by the corresponding carbon emission of the replaced component. This incorporation allows the decision-making to identify the most cost - effective plan for replacement, taking into account the environmental impact. In other words, this approach adds an extra dimension in the model presented in Section 3.2 by taking into consideration green aspects. The corresponding carbon emissions for each replacement action are displayed below.

$$e(a) = \begin{cases} 0 & \text{if } a = 0 \\ e_1 & \text{if } a = 1 \\ e_2 & \text{if } a = 2 \\ e_3 & \text{if } a = 3 \\ e_1 + e_2 & \text{if } a = 4 \\ e_1 + e_3 & \text{if } a = 5 \\ e_2 + e_3 & \text{if } a = 6 \\ e_1 + e_2 + e_3 & \text{if } a = 7 \end{cases}$$

with e_i being the amount of carbon emissions of component i. The specific values for each component can be found in Table 3.4, highlighted in bold.

Structurally, the problem's actions remain the same as replacement decisions refer to the same computer components. Note that discount factors are not applied to joint and full replacement of components, in contrast to Formulation 3.1. The decision lies in the fact that joint or full replacements do not address additional environmental benefits when compared to individual replacements.

Results

In this chapter, the insights from the previously mentioned methodology will be implemented. The presentation of the findings is organized as follows: Section 4.1 demonstrates each of the aforementioned algorithms involving two components. Accordingly, Section 4.2 displays the algorithms considering a three - component system. Lastly, Section 4.3 exhibits the optimal replacement policy based on two factors: replacement costs and carbon emissions during manufacturing of desktop components. For the rest of the paper, the algorithms were written in Java programming language (version 16.0.2) and CPLEX solver (version 22.1.0) employed when needed.

4.1 Case of two components

4.1.1 Markov Decision Model

In a two-component system, our focus is directed on CPU and GPU. Even though HDD is also crucial for the overall performance, the choice of studying these specific components is motivated by their multifaceted roles in the functionality of a computer. However, the strongest motivation comes from the existence of two extreme values in the problem, namely the lowest deterioration value of CPU and the highest replacement costs of GPU. It is valuable to investigate the behavior of the model within this problem setup.

For low-end components the deterioration levels for CPU and GPU are established to $\lambda_{CPU} = 0.05$ and $\lambda_{GPU} = 0.18$. These values come up from the research of Jiajian Yan (2023) and imply the significantly higher deterioration of GPU compared to CPU during their life cycles. The parameters $\alpha_i = e^{-\lambda_i}$ are determined based on the magnitude of rate λ_i, resulting in $\alpha_{CPU} = 0.9538$ and $\alpha_{GPU} = 0.8352$. The step size β influences the magnitude of deterioration and is set to $\beta = 0.8$. The corresponding weights influencing the overall system performance are $w_{GPU} = 0.67$ and $w_{CPU} = 0.33$. These weights are determined arbitrarily and reflect the consumer preferences and the performance expectations of each component towards the performance of a desktop computer. Once either the system or a component reaches its minimum performance threshold $\theta_i = \theta^{sys} = M = 6$, a preventive replacement occurs and a new version of it takes place. The exact mathematical formulation can be found in A.1.

The computational time of the algorithm is quite short ($<< 0.01$sec). The reason behind this efficiency is the small number of considered components and consequently the reduced number of decision variables on the problem. As the latter tends to increase exponentially with the former, the running time is reasonable low.

The optimal policy yields an average cost of 27.15$ USD over a half-year period. These costs exhibit variability based on the replacement probabilities associated with each action. Specifically, parameter β influences the magnitude of transition probabilities. Therefore, opting for a higher value of β leads to substantially higher costs.

Given that transition probabilities do not depend on the states but components themselves, the replacement actions take place uniformly throughout the renewal cycle. In other words, each replacement action occurs with identical probability for all states. By conditioning on each action, the likelihood of replacing CPU over a half-year period is measured to 4% while the corresponding probability for GPU replacement is 13.4%. The replacement likelihood for both components together is observed to a negligible 0.6%. Figure 4.1 provides a visual representation on which and when component(s) require upgrade due to deterioration.

Evidently, GPU is more prone to be replaced compared to CPU, primarily due to its higher degradation rate. In particular, the transition probability matrix denotes greater likelihood of replacement for components with tendency for higher degradation, given that $\lambda_{GPU} > \lambda_{CPU}$. Furthermore, the decision-making process is influenced by the replacement cost of each component. Considering that the replacement costs of GPU is 2.5 times higher than those of CPU and the model's objective is to minimize costs, it becomes an important factor in the decision-making framework.

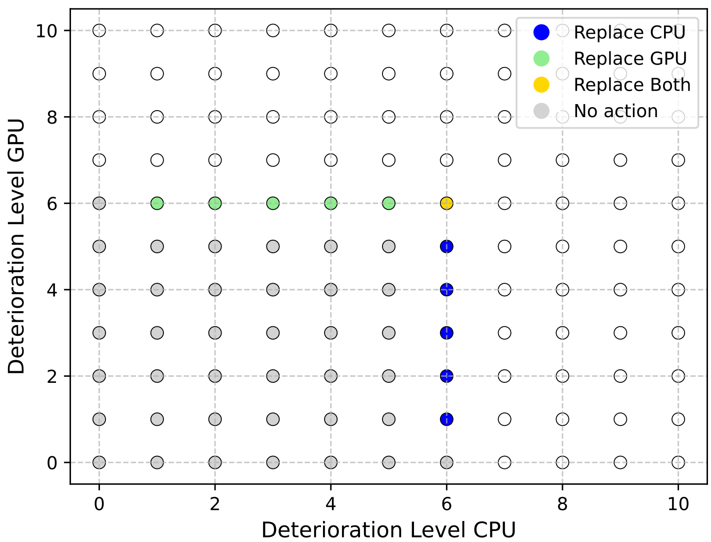

Figure 4.1: Replacement Policy for CPU and GPU ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

In Figure 4.1, the bullet points represent the replacement actions over the renewal cycle. As the states evolve, the replacement probabilities remain stable. This outcome was expected as replacement probabilities were independent of states. Consequently, each combination (i_1, i_2) has identical probability to be replaced for a given a. The colors in the figure signify the action taken for every combination of states, with the hue remaining unchanged as the states progress - each action is executed with the exact same probability.

The computer's expected performance is determined based on individual performances of its components. For CPU, the item's expected performance falls within the range of 100% and 74% before replacement, while the corresponding interval for GPU spans from 100% to 34%. To delve deeper into the system's performance, an assessment will be conducted considering replacement actions for each component. Thus, given that CPU is replaced at state M, the average system performance falls between 91% and 51% with an average of 66%. Similarly for GPU, replacement at state M yields a performance between 55% and 48% with an average of 51%. The overall system performance reaches its minimum level when both components attain their maximum deterioration age at state M, where the overall performance drops to 47%.

Conducting a sensitivity analysis on the algorithm's solution pattern reveals potential variations. In the first scenario, preventive replacement is delayed to state 9 instead of state 6, allowing assessment of the model's behavior over an extended time horizon. Figure 4.2 depicts this adjusted solution path.

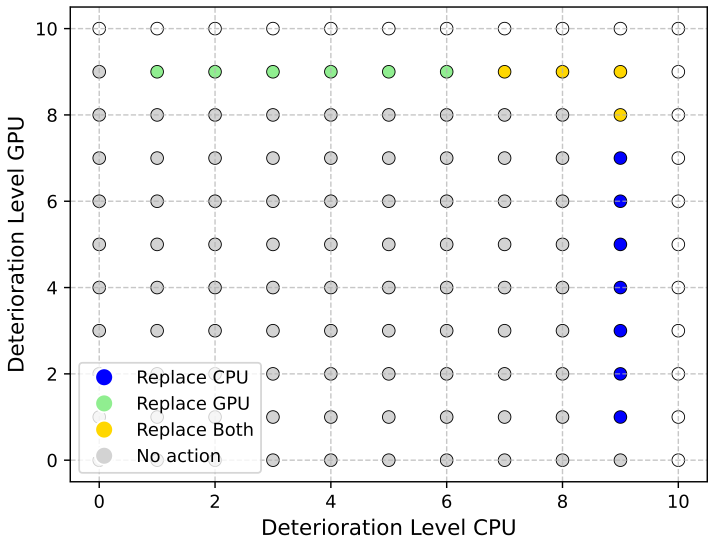

Figure 4.2: Replacement Policy for CPU and GPU at state 9 ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

While the solution path resembles that of Figure 4.1, the impact of the discount factor on the longer time horizon becomes apparent. Gold bullets mark computer replacement, signifying the preference for system replacement at these specific states. The average cost is influenced by the extended time frame, resulting in lower average costs of 17.64\$. A deeper exploration of the matter entails modifying the discount factor to $\delta = 0.8$ while retaining the delayed preventive maintenance. The outcomes of this adjustment are illustrated in Figure 4.3

Comparing Figures 4.2 and 4.3 highlights a heightened occurrence of states necessitating full

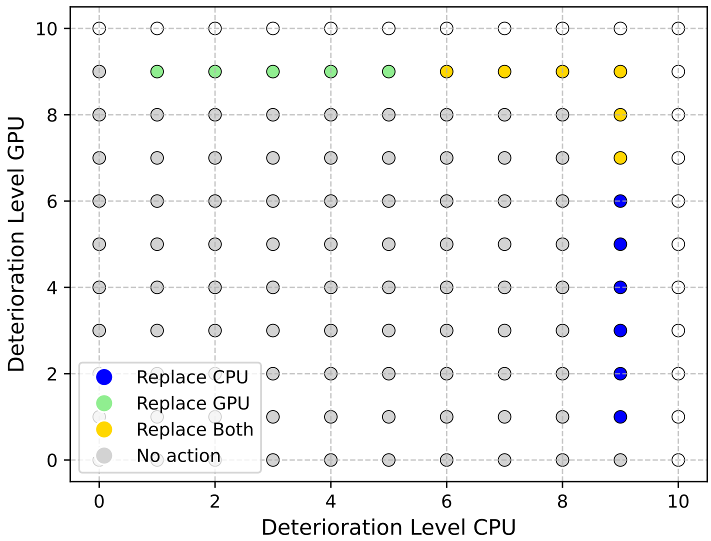

Figure 4.3: Replacement Policy for CPU and GPU at state 9 with $\delta = 0.8$ ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

replacement. More specifically, the algorithm identifies two additional states ((6,9) and (9,7)), driven by higher discount-related replacement costs. This prompts the algorithm to favor full replacement sooner, leading to an overall reduction in replacement costs. Consequently, the average costs in this scenario are computed at 16.86$ per half-year period.

4.1.2 Dependent Deterioration

An identical framework is applied involving dependent probabilities between the components. Both parts have the same λ's as before and they are replaced after an identical period of time ($M = 6$). The formulation remains the same, with only modification being on the transition matrix, denoted as $p(i_1, i_2)(j_1, j_2)$(a).

Although the current formulation has simpler structure than the original MDM, the computational time remains the same ($<< 0.01$sec). The average replacement costs slightly decrease to a value of 26.79$ USD. This kind of dependency introduces unpredictability in the behavior of the model for two reasons. Firstly, the average cost depends on the deterioration speed of CPU. Moreover, the interdependence among components implies also dependency on the actions taken. In the present scenario, the replacement probability of one component varies based on the condition of the other, as illustrated in Figure 4.4.

Across the renewal cycle, replacement actions no longer adhere to a constant probability as in the original model. Figure 4.4 represents these varying probabilities with different colors. For example, the replacement probability of CPU in state (6,5) is higher compared to replacing it in state (6,1). This pattern holds true for GPU replacements as well. The lighter the hue of a color, the greater the corresponding replacement probability. Furthermore, it is worth mentioning that there is a small probability of replacing GPU even in state (6,0). This occurrence is a result of underlying assumptions and is depicted mathematically on the transition probability matrix.

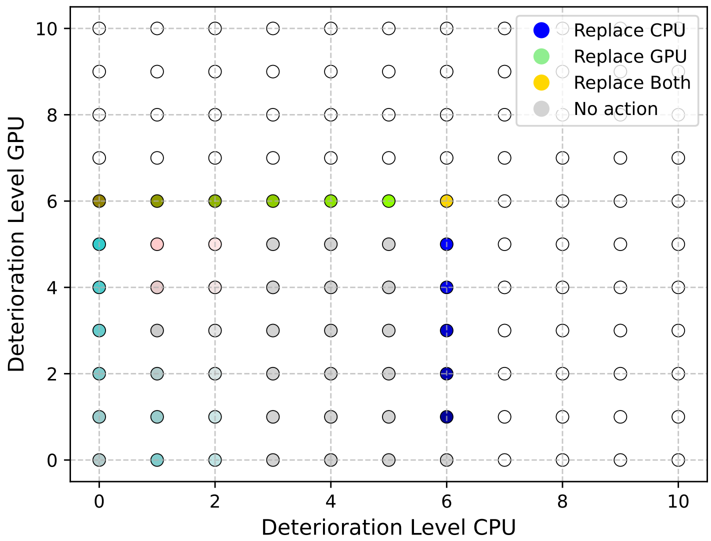

Figure 4.4: Dependent Replacement Policy for CPU and GPU ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

While this may potentially lead to higher average costs compared to the original model, the fluctuations in replacement probabilities throughout the cycle contribute to overall cost reduction.

The remaining points within the rectangular shape denote the probability of taking no action. However, the likelihood of being in state (i_1, i_2) fluctuates for the first three states of each component and then stabilizes, mirroring the behavior observed in the original problem. This difference is spotted due to the interdependence nature of the problem. The minor variance in the overall costs stem from the initial, low probability of replacing at the beginning of the cycle, contrasting to the original model.

To evaluate algorithm's performance across different parameter settings, a sensitivity analysis is conducted, following the approach of the original MDM. As before, the first case involves replacing system components at state 9 and the results are illustrated in Figure 4.5.

In Figure 4.5, color hue intensity correlates with replacement probability. Therefore, sole CPU replacement sees a higher probability at state (9,8), while GPU replacement is more likely at state (6,9). Conversely, full replacement has a higher probability at state (7,9) compared to (9,9), reflecting the diverse deterioration rates. This divergence implies a likelihood of component replacement in different states rather than simultaneously. Additionally, this explains the delay in CPU replacement at early states compared to GPU. Grey states exhibit probabilistic variations, as illustrated in Figure 4.4. Accurately depicting these probability shifts with different hues poses a challenge. For further evidence about the approach, Figure 4.6 explores the impact of setting a higher discount factor on the problem.

Similarly with the original model, a higher discount factor increases the frequency of full replacement actions. Six states indicate full replacement with varying probabilities. The likelihood of replacement at states (6,9) and (9,7) surpasses that at state (9,9), influenced by the interdepend-

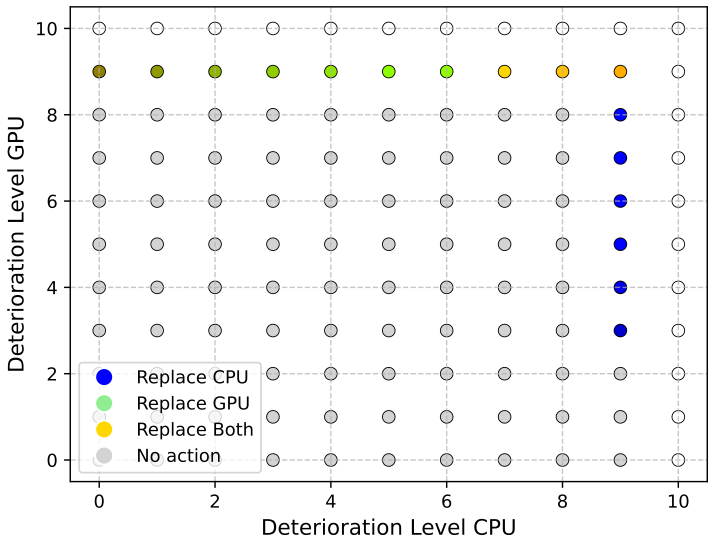

Figure 4.5: Dependent Replacement Policy for CPU and GPU at state 9 ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

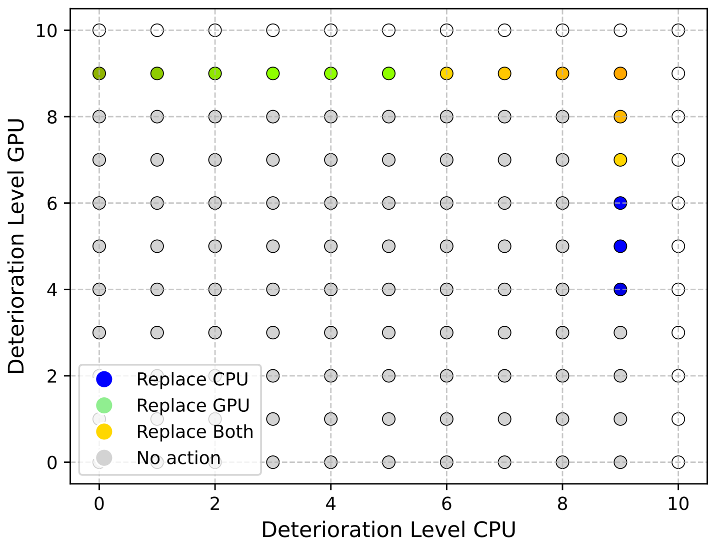

Figure 4.6: Dependent Replacement Policy for CPU and GPU at state 9 with $\delta = 0.8$ ($\lambda_{CPU} = 0.05$, $\lambda_{GPU} = 0.18$)

ence of component actions. While individual CPU replacement occurs less frequently, it does so with smaller probabilities, contributing to an increase in corresponding GPU replacement probabilities.

4.1.3 Power of Two policy

In this section, the final algorithm demonstrated is the Power of Two policy. As previously stated, parameters $\lambda_{CPU} = 0.05, \lambda_{GPU} = 0.18$ denote the deterioration rates of components per half-year, and the replacement costs are displayed in Table 3.1. The expected lifetime of each component is derived from the corresponding exponentially distributed probability function. Utilizing the properties of exponential distribution, $E(CPU) = \frac{1}{\lambda_{CPU}} = 20$ and

$E(GPU) = \frac{1}{\lambda_{GPU}} = 5.55$. The renewal cycle is determined by the least common multiple between 20 and 5.55, resulting a cycle duration of 2220. Throughout this period, CPU is replaced 111 times while GPU 400 times. Evidently, GPU displays higher number of replacements per cycle due to its faster deterioration. The expected cost $E(C)$ is calculated using the formula:

$$E(C) = \frac{\#\text{ of Indiv. Replacements} \times \text{Indiv. Replacem. Costs} + \text{Disc. Full Replacem. Costs}}{\text{Length of Renewal Cycle}}$$

In other words,

$$E(C) = \frac{110 * 70 + 399 * 170 + 216}{2220} = 34.12\$$$

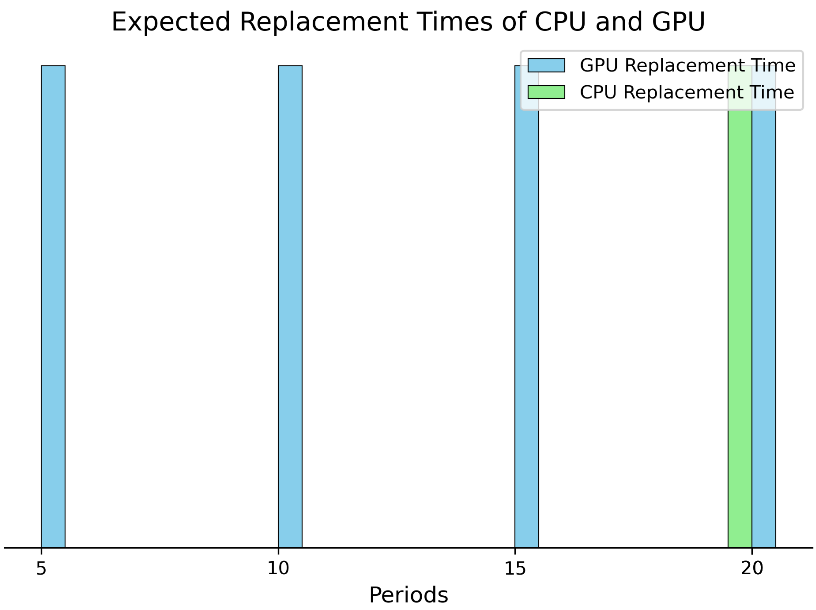

Figure 4.7: Expected Replacements for CPU and GPU with $\lambda_{CPU} = 0.05, \lambda_{GPU} = 0.18$

The result implies that CPU and GPU are replaced 110 and 399 times individually while in period 2220 a full replacement occurs. Upon the full replacement, the system renews and the same actions are repeated. The average cost of this renewal cycle is measured to 34.12$ per half-year. Remember that the model in Section 3.2 expresses the average cost over a half-year period as well.

At that point the Power of Two policy will be presented. From Section 3.4, it is known that

$$T = 2^m q \tag{1}$$

Initially, a slight modification of the algorithm in Section 3.4 is outlined. The parameter q represents the shortest expected lifetime among the components which is $q = 5.55$ - the smallest value among the items under study. Another restriction is imposed by ensuring that the renewal cycle remains smaller than 20; otherwise CPU would cease functioning. In essence, the goal is to identify a common replacement cycle, falling between 5.55 and 20.

The Power of Two heuristic performs well for deterministic environments. Given that, the timing of the coordinated replacements depends on the decision-maker. For simplicity, let's consider a common cycle length $T = 20$. In this scenario, GPU undergoes replacement 4 times while CPU only one. The visual representation of this policy is illustrated in Figure 4.7. At the 20th period, both components are replaced simultaneously, resulting in a discounted replacement. The average cost is calculated as

$$E(C') = \frac{\text{\# of Indiv. Replacements} \times \text{Indiv. Replacem. Costs} + \text{Disc. Full Replacem. Costs}}{\text{Common Cycle}}$$

Or,

$$E(C') = \frac{3 * 170 + 216}{20} = 36.3\$$$

Figure 4.7 depicts that GPU is replaced in periods 5.55, 11.1, 16.65 and 20 while CPU is replaced at period 20. The value of 216 signifies the discounted replacement price for both components. However, the common cycle length can be optimized, reducing the average replacement costs.

The optimization of T involves the decision to replace the whole system when either of the components reach its lifetime limit. In the earlier scenario, the common cycle of 20 represents the cycle when CPU reaches its age limit. However, a more in - depth analysis is required to assess the costs when GPU reaches its limit. This occurs when the life cycles are 5.55, 11.1 and 16.65. These are the only three possibilities, as all the other multiples of 5.55 surpass the lifetime of CPU. Table 4.1 displays all potential common cycles where GPU is replaced at its limit.

Table 4.1: Potential Common Cycles

Common Cycle	CPU	GPU	Total Cost ($)	Average Cost / Half -Year ($)
5.55	1	1	216	43.2
11.1	1	2	386	34.77
16.65	1	3	556	**33.39**

The 'optimal' common cycle, minimizing replacement cost, is determined to be 16.65 periods. Throughout this period, the average replacement costs amount to 33.39$ per half-year. In contrast, the optimal replacement costs from Section 4.1.1 are 27.15$. This indicates that the replacement costs for PoT are approximately 20% higher than those in the LP formulation. This disparity is expected since in PoT, components are not replaced at their optimal times, leading to redundant replacements and higher costs. The advantage of the algorithm lies more in providing interpretable results rather than efficiency in achieving optimal replacements.

Alternatively, considering common cycle and shortest expected lifetime as an integers ($T, q \in \mathbb{Z}$) may offer a more straightforward interpretation. For that reason, sensitivity analysis follows, starting with $q = 5$. In this scenario, GPU is replaced in period 5, 10, 15 and 20 while CPU is replaced in period 20. The replacement times are almost identical to the case presented earlier.

The corresponding average replacement costs are

$$E(C_1) = E(C') = \frac{3 * 170 + 216}{20} = 36.3\$$$

This strategy holds the advantage of replacing components before reaching their expected lifetimes. In practical terms, a component may fail prematurely or after reaching its anticipated value. Opting for GPU replacement every 5 periods minimizes the risk of failure while keeping costs constant. Conversely, considering $q = 6$ proves illogical, as this value exceeds the lifetime of GPU, posing a high risk for the computer's functionality.

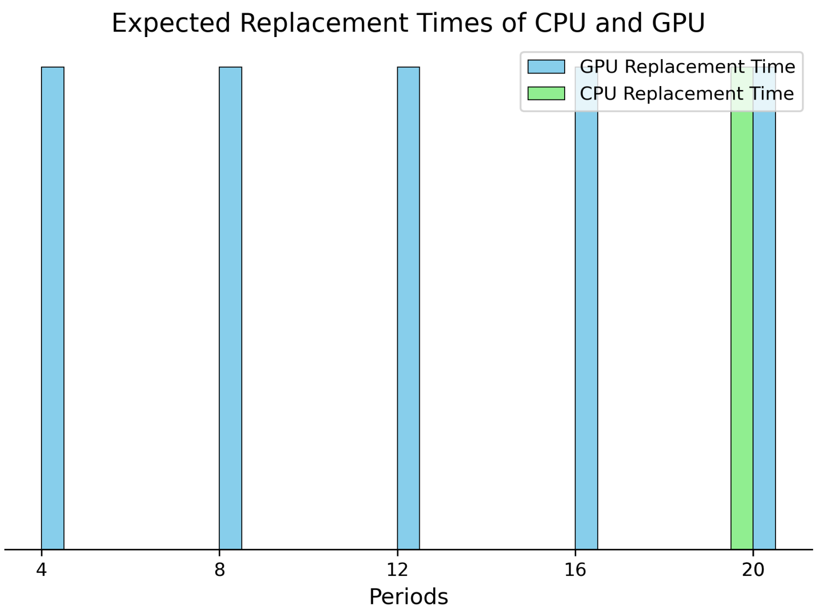

Figure 4.8: Expected Replacements for CPU and GPU with $q = 4$

Another option is to consider an even smaller value for q, further minimizing the risk of failure. However, this also results in more replacements over time, leading to higher replacement costs. Therefore, the case of $q = 4$ is examined. In that case, GPU is replaced at periods 4, 8, 12, 16 and 20 while CPU is replaced at period 20. This implies individual GPU replacements four times and the system replacement at a discount on period 20. The corresponding policy is illustrated in Figure 4.8. Mathematically,

$$E(C_2) = \frac{4 * 170 + 216}{20} = 44.8\$$$

Clearly, the average cost is higher than in the previous cases. This trade-off between replacement times and danger of failure characterizes algorithm's nature. The percentage difference between $E(C_1)$ and $E(C_2)$ is approximately 20%, a significant figure that becomes even more apparent in the long term.

4.2 Case of three components

4.2.1 Markov Decision Model

In the three - component system, the mathematical modelling of Markov Decision formulation becomes somewhat complicated. The introduction of an additional component (HDD) exponentially escalates the complexity of the problem. Therefore, the problem requires higher computational power to be solved. Degradation parameters for low-end components, derived from Jiajian Yan (2023), are defined as follows: $\lambda_{CPU} = 0.05, \lambda_{HDD} = 0.13, \lambda_{GPU} = 0.18$. Consequently, $\alpha_{CPU} = 0.9512$, $\alpha_{HDD} = 0.878$ and $\alpha_{GPU} = 0.8352$ imply the geometric performance decrease for each component. The step size is constant at the value of $\beta = 0.8$ for all components. In prioritizing user requirements, components are assigned specific weights, indicative of users' inclinations to replace each element to enhance overall system performance. Mathematically, they are expressed as $w_{GPU} = 0.5$, $w_{HDD} = 0.25$ and $w_{CPU} = 0.25$. As in the case of two components, when a component or the whole computer attain its minimum performance requirements where $M = \theta_i = 6$, an upgrade is initiated by replacing the relevant counterpart. The replacement costs can be found in Table 3.1 whereas the mathematical framework has been presented in Formulation 3.2.

Regarding the algorithm's running time, solving the problem demands more time, yet it remains reasonably fast ($\approx$ 2.5sec). Compared to the case of two components, the computational time has been remarkably escalated by 25 times. The reason behind this phenomenon is the increase in the number of variables in the problem, soaring from 400 to a staggering 8000!

In terms of average cost, it is measured to 30.05$ USD. This figure is higher than the cost associated with two components. The increase is attributed not only to the study of an extra component but also in the low replacement price of HDD, set at 30$ USD. Conditioning on the action taken, CPU is replaced at a total probability of 3.5%, HDD at 8.9% and GPU at 12% per half-year period. However, the joint replacements of two out of three components are less common, with corresponding percentages residing around 1%. This trend is similarly observed in the case of replacing all components, as illustrated in Figure 4.9.

It is evident that Figure 4.9 is not easily readable. In fact, a 3D depiction of the policy fails to convey all the insights and another type of graph would be more suitable to represent our findings. In general, interpretability is a challenging aspect when presenting systems' replacement of more than two components graphically. In an attempt to enhance clarity of our results, a bar chart is presented in Figure 4.10 as an alternative depiction.

The bars in Figure 4.10 reflect the percentage frequency of individual and joint component replacements, given their replacement actions. More specifically, the discounted replacement of two out of three components occurs less frequently compared to single replacements, highlighting the cost - effective advantage of individual component replacements, as evidenced by the leftmost bars in the graph. The likelihood of replacing the whole system is rather rare, primarily

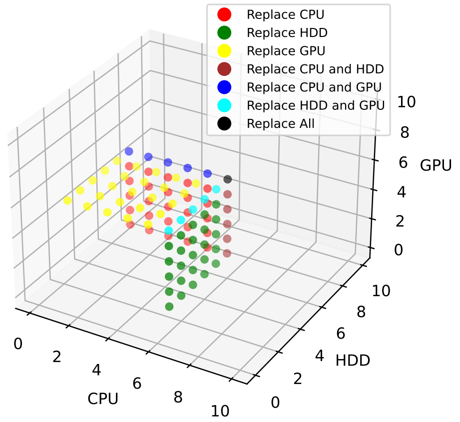

Figure 4.9: Replacement Policy for CPU, HDD and GPU ($\lambda_{CPU} = 0.05, \lambda_{HDD} = 0.18, \lambda_{HDD} = 0.13$)

due to its highest cost.

The probability of replacing either HDD or GPU is 2.5 and 3.5 times higher compared to that of CPU. This discrepancy is attributed to their higher λ's, leading to more potential replacements of the former compared of the latter. The occurrence of minimal percentages in joint and full replacement actions is due to the forced replacement at state M in the problem. If, for instance, system or component replacements were to occur at later stage ($M > 6$), joint or full replacements would become more favorable. However, it is important to note that the current model settings do not showcase this insightful perspective.

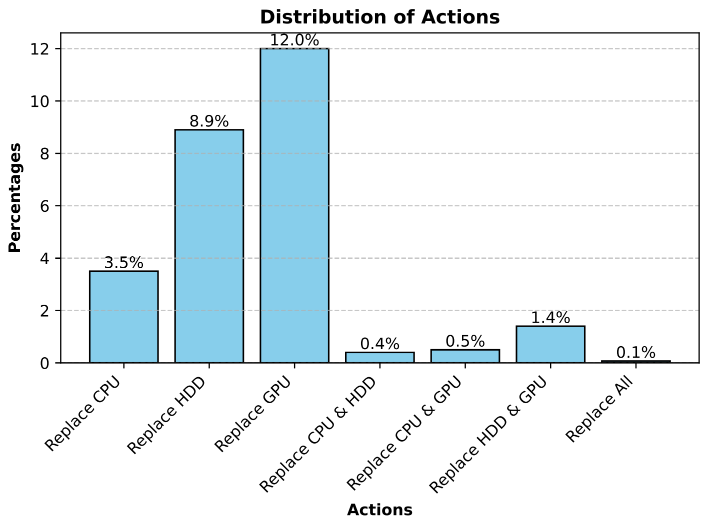

Figure 4.10: Frequency of Replacement Actions for CPU, HDD and GPU

As the system advances, its expected performance gradually diminishes. Following a full replacement, the system commences with 100% performance rate. Given that individual replacement actions have the most significant influence on the average costs, it is valuable to investigate the expected performance of the system as each component approaches its age limit.

Upon reaching its maximum age, CPU yields an overall performance ranging from 93% to 50%. Similarly, for HDD, this interval is 86% to 50% and for GPU, it is 66% and 48%. The average performance for CPU, HDD and GPU is 67%, 64% and 56% respectively. The lowest performance is observed when all components simultaneously reach their age limit at point M, causing the system performance to drop to 47%.

To gain deeper insights into the algorithm, a slight modification is applied to the problem. More specifically, the preventive replacement occurs at state 9 instead of state 6. This adjustment allows for a clearer examination of the impact of the discount factor on replacement actions. Figure 4.11 portrays the total probabilities for each replacement action under this adjustment.

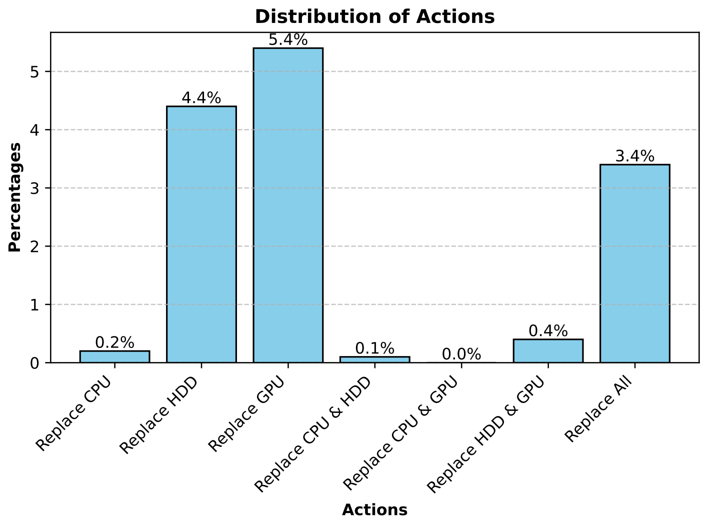

Figure 4.11: Frequency of Replacement Actions for CPU, HDD and GPU at state 9

As the preventive replacement of components is delayed, the probabilities of individual replacements diminish. Notably, replacement probabilities for HDD and GPU decrease by half, while the corresponding decrease is even more intense for CPU. Conversely, the probability of full replacement experiences a significant increase, as the overall cost associated with this action is reduced. In general, the algorithm opts for full replacement over any other kind of replacements. As an alternative, Figure 4.12 depicts the case of replacing at state 9 but with higher discount values.

In that case, with the joint ($\alpha = 0.8$) and the full ($\gamma = 0.7$) replacement discounts, the effect intensifies. Individual and joint replacement probabilities sharply decline to negligible percentages, while the probability of full replacement skyrockets to a substantial 7.6%! Essentially, the decision-making process predominantly leans toward full replacements over other model actions.

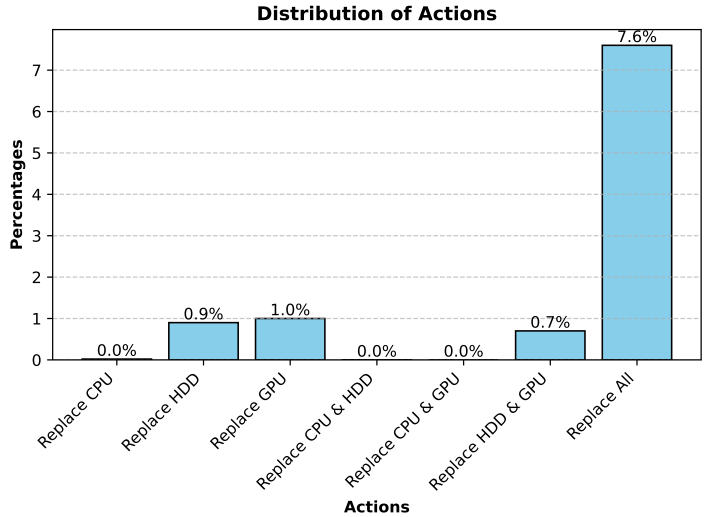

Figure 4.12: Frequency of Replacement Actions for CPU, HDD and GPU at state 9 with $\alpha = 0.8$ and $\gamma = 0.7$

4.2.2 Dependent Deterioration

In the scenario involving dependent probabilities, the identical mathematical approach of Formulation 3.1 is employed. As with the case of two components, the only element changing in this modification is the transition probability matrix $p(i_1, i_2, i_3)(j_1, j_2, j_3)(\text{a})$.

The average cost is recorded to 29.53$ USD. This outcome follows a similar trend to the two-component system, where the average replacement costs are slightly lower than those of the original model. This can be attributed to the higher frequency of taking no action, at a percentage of 74% instead of the original 72%. Individual replacements are favored with slightly reduced percentages compared to the initial problem. Moreover, the symbiotic relationship among components leads to joint replacements more often than the first scenario. Full replacement remains the least probable action. Lastly, the algorithm exhibits slightly faster running time ($\approx$ 2sec). Figure 4.13 depicts the corresponding results.

Individual replacements continue to dominate the graph with sole replacement of CPU, HDD and GPU accounting 2.8%, 8.3% and 11.6% per half-year respectively - percentages lower than those observed in the original model. In contrast, joint replacements show an increase in percentages compared to the previous model, reflecting the interdependence among the components as indicated by the transition probabilities. The presumed dependency in the model opts for joint replacements due to the nature of relationship among components. Evidently, the joint replacement of HDD and GPU occurs more frequently compared to the rest joint replacements, given the high deterioration suffered by both components.

The solution path aligns with that observed in the first model. Similar to the initial MDM, presenting the results in 3D graphs remains challenging, rendering Figure 4.9 also inadequate for this case. It is essential to recognize that the decision variables in state (i_1, i_2, i_3) under

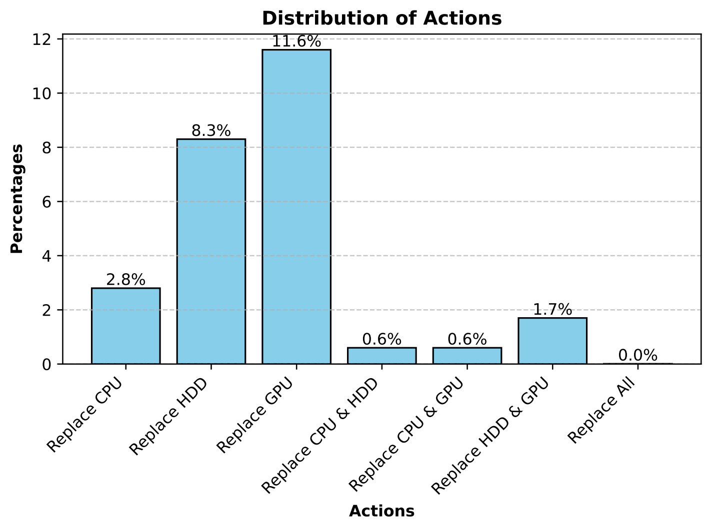

Figure 4.13: Dependent Replacement Actions for CPU, HDD and GPU

action a are not identical for each action. Much like the original model, as states evolve, the optimal strategy opts for replacement of a component when the other components are closer to their maximum deterioration level. Unfortunately, this trend is hard to be depicted with different colors and hues for more than two components.

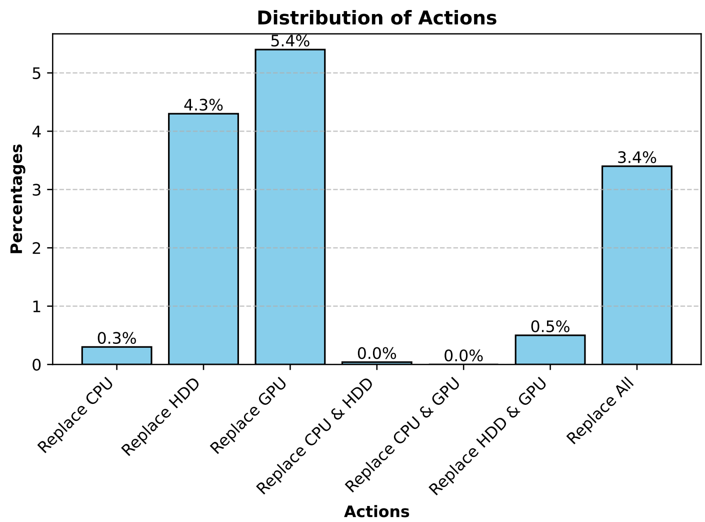

Figure 4.14: Dependent Replacement Actions for CPU, HDD and GPU at state 9

If the preventive replacement occurs later, the model exhibits similar behavior to that discussed in Section 4.2.1. The significant changes lie in the replacement probabilities for each state (i_1, i_2, i_3) which vary throughout the cycle. Depicting this proves to be challenging due to the involvement of numerous variables. Figure 4.14 illustrates the corresponding results, emphasizing a significant percentage in the full replacement action.

Figures 4.11 and 4.14 display evident similarities, prompting the need for further exploration.

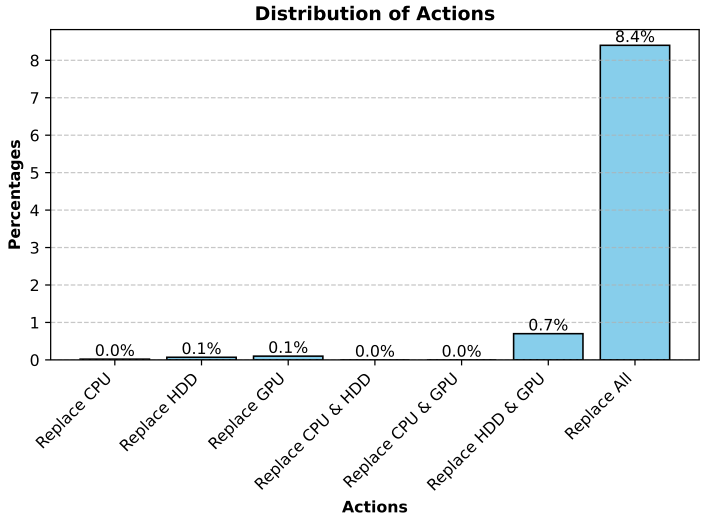

Figure 4.15: Dependent Replacement Actions for CPU, HDD and GPU at state 9 with $\alpha = 0.8$ and $\gamma = 0.7$

To delve deeper, Figure 4.15 is introduced, showcasing the case with higher discounted values, as previously discussed. The discount factors are $\alpha = 0.8$ for joint and $\gamma = 0.7$ for full replacements respectively. This approach interrelates the components to each other, by making decisions based on the condition of CPU. In essence, the expectation is a more frequent occurrence of full replacements than any other action, at a percentage of 8.4%, leaving the rest of actions at minor numbers. This results from the highly discounted replacement costs associated with full replacement and the interdependency among the components. Consequently, the algorithm consistently opts for full replacements.

4.2.3 Power of Two policy

The Power of Two policy operates similarly to the previously described approach, featuring two components. For each component, the parameters are specified as follows: $\lambda_{CPU} = 0.05, \lambda_{HDD} = 0.13, \lambda_{GPU} = 0.18$, with constant replacement costs. The expected lifetimes of CPU, HDD and GPU are $E(CPU) = 20$, $E(HDD) = 7.7$ and $E(GPU) = 5.5$. The renewal cycle is determined by the least common multiple (LCM) of component lifetimes, set at a value of 1540. During the cycle, CPU, HDD and GPU undergo replacement 77, 200 and 280 times respectively.

In Figure 4.16, the expected replacement times for each component are illustrated for the first 20 periods of the renewal cycle. Clearly, GPU leads the replacement times since it lasts shorter than the rest of items. While it appears that GPU is replaced three times more often than CPU, the corresponding proportion is higher in the long run. This trend also applies to HDD, seemingly replaced twice as frequently as CPU in short term.

Replacement costs are calculated similarly to the two - component scenario, incorporating joint

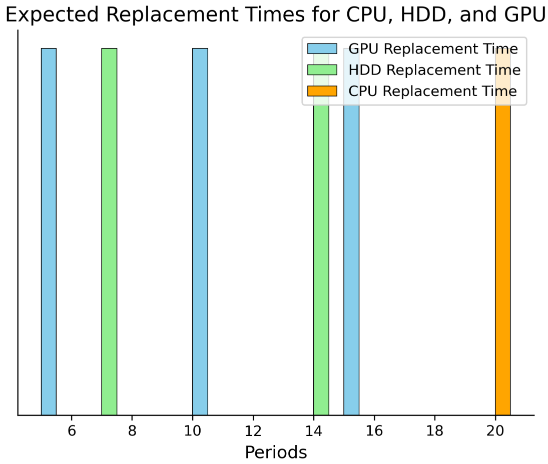

Figure 4.16: Lifetime illustration of computer components

replacements. The expected cost $E(C)$ is calculated using the following formula:

$$E(C) = \frac{\#\text{ of I.Replac.} \times \text{I.Replac.Costs}}{\text{Length of Ren. Cycle}} + \frac{\#\text{ of J. Replac.} \times \text{J. Replac.Costs}}{\text{Length of Ren. Cycle}} + \frac{\text{F. Replac.Costs}}{\text{Length of Ren. Cycle}}$$

Here, I, J, F represent individual, joint and full replacements respectively. Numerically, this implies

$$E(C) = \frac{68 * 70 + 158 * 30 * 232 * 170}{1540} + \frac{1 * 90 + 7 * 216 + 40 * 180}{1540} + \frac{216}{1540}$$

or,

$$E(C) = 37.06\$$$

Naturally, the result is higher than that of the two-component system, primarily due to the additional component introducing extra replacement costs. The joint replacements across the cycle are computed based on the LCM of the corresponding component lifetimes. For example, GPU undergoes a total of 280 replacements throughout the renewal cycle. Joint replacement for CPU and GPU, with LCM of 220, leads to 7 replacements. At the same wavelength, joint replacement of HDD and GPU has LCM 38.5, implying 40 replacements. As a result, individual replacements for GPU amount to 232 (280 - 47 - 1). Accordingly, the same applies for CPU and HDD. Note that the discount factors $\alpha = 0.9$ and $\gamma = 0.8$ are applied also for this approach, signifying the discounted cost of replacing more than one component simultaneously.

As it has already been discussed, parameter q denotes the shortest expected lifetime of components and value T represents the common cycle time for a full replacement occurs and system renewal. Since GPU is the most-deteriorating component, it is assigned the lowest lifetime. Moreover, the common cycle lies in a number between 5.5 and 20, the two extreme numbers of the problem. The Power of Two policy gives the freedom for deterministic replacements at any point of time. One approach could be to set T and q as integer numbers, with T being the highest expected lifetime among the components. By fixing m, the corresponding q is found and rounded down to the nearest integer. In this scenario, the least deteriorating item is replaced at time T with others replaced every q periods.

The optimal q for this case is attained when $m = 2$, resulting in $q = 5$. Following this strategy, GPU and HDD are replaced jointly at times 5, 10, 15, 20 with the full replacement (including CPU) at time 20. In that case, the average replacement costs are

$$E(C'') = \frac{3 * 0.9 * (170 + 30) + 0.8 * 270}{20} = 37.8\$$$

However, this schedule leads to unnecessary frequent replacements for HDD. Despite its expected lifetime being 7.7, the approach dictates replacements every 5 periods. Therefore, HDD is replaced 4 times within the cycle length whereas it could be more advantageous to replace it 3 times (at times 7.7, 15.4 and 20).

In an attempt to improve the replacement costs with as many joint replacements as possible, a different approach will be followed. The common cycle will be fixed based on the expected life of each component and its multiples. In other words,

For the case of CPU,

Table 4.2: Potential Common Cycles with CPU

Common Cycle (T)	Replace CPU	Replace HDD	Replace GPU	Average Cost / Half - Year ($)
20	1	3	4	39.3

For the case of HDD,

Table 4.3: Potential Common Cycles with HDD

Common Cycle (T)	Replace CPU	Replace HDD	Replace GPU	Average Cost / Half - Year ($)
7.7	1	1	2	50.12
15.4	1	2	3	40

For the case of GPU,

Table 4.4: Potential Common Cycles with GPU

Common Cycle (T)	Replace CPU	Replace HDD	Replace GPU	Average Cost / Half - Year ($)
5.5	1	1	1	39.27
11.1	1	2	2	35.67
16.6	1	3	3	**34.7**

Comparing the annual replacement costs presented in Tables 4.2, 4.3 and 4.4, it becomes evident that the most beneficial common cycle for full replacement of the system is at period 16.6. This implies that, upon the third replacement of GPU, a full replacement takes place. At the same time, HDD is replaced 3 times, with the third replacement closely following the second (at periods 15.4 and 16.6 respectively). Practically, the replacement of HDD for third time occurs relatively fast compared to the previous one. However, its low replacement cost contribute only marginally to the average costs.

The optimal replacement costs derived from the Markov Decision Model result to 30.05$ for a half-year interval while PoT indicated an average cost of 34.7$ for the same time period. As with the two-component system, the heuristic yields average costs nearly 14% higher than those in the LP formulation. Although PoT promotes coordinated replacements to leverage joint and full replacement discounts, this strategy may result in redundant replacements, leading to overall costs.

To showcase the significance of discount in joint and full replacements, the scenario where joint replacements are not incorporated is considered. Assume that CPU is replaced every 5 period and HDD every 7, with a common cycle length $T = 20$. This strategy reduces the risk of failure for components before they reach the limit of their expected lifetimes. However, the absence of a discount factor arises as the components are replaced at different intervals. For instance, GPU is replaced in periods 5, 10, 15, HDD in periods 7, 14 and a full replacement occurs at period 20. The average costs are:

$$E(C_3) = \frac{3 * 170 + 2 * 30 + 216}{20} = 39.3\$$$

This result is identical to that of in Table 4.2. The former has the advantage of replacing components slightly earlier than the latter, reducing the risk of failure. However, it does not capitalize on discounts, leading to higher average replacement costs.

The option of joint replacement every 5 periods has been discussed earlier. Another option could be to implement joint replacements every 6 periods; however, this might result to system downtimes, a fact that should be avoided. Therefore, we explore the case of replacements occuring every 4 periods, as illustrated in Figure 4.17.

Figure 4.17 depicts the corresponding joint and full replacement periods, where HDD and GPU are replaced every 4 periods (at times 4, 8, 12, 16), with a full replacement taking place in period 20. This approach offers the advantage of discounted replacements for multiple components. However, the average costs are higher

$$E(C_4) = \frac{4 * 0.9 * (170 + 30) + 0.8 * 270}{20} = 46.8\$$$

Although this case gives the opportunity for discounted costs and 'safe' replacement times, it also implies more frequent replacement actions. In essence, components might be replaced more often than necessary, leading to extra replacements. PoT grants decision makers the freedom to deterministically choose when to replace components.

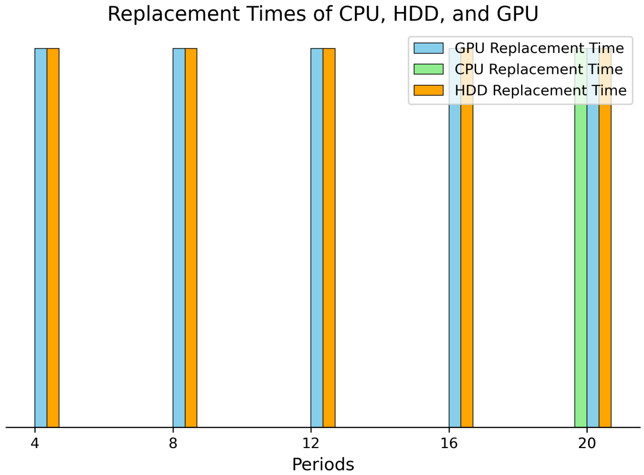

Figure 4.17: Joint and Full Replacement Depiction for CPU, HDD and GPU

4.3 CO_2 Footprint Savings

The mathematical model outlining the optimal replacement strategy has been described in Formulation 3.2. Utilizing consistent deterioration rates - $\lambda_{CPU} = 0.05, \lambda_{HDD} = 0.13, \lambda_{GPU} = 0.18$ - the model employs geometric performance decrease parameters for each component, expressed as $a_i = e^{-\lambda_i}$ with fixed step size of $\beta = 0.8$. The associated replacement costs for each component are provided in Table 3.1.

The average CO_2 emissions stand at 9.96 kg per renewal cycle, a factor introduced as an additional consideration within the Markov Decision Model framework presented in Section 3.2. This value is derived by multiplying the probability of a replacement action, as determined by solving the MDM in Section 4.2.1, by the corresponding carbon emissions, exhibited in Section 3.5. Evidently, the amount of carbon emissions is influenced by the transition probabilities which in turn depend on parameters α_i and β. It is important to highlight that transition probabilities remain constant and independent of states. Figure 4.18 illustrates the average carbon emissions saved for replacement actions at any given state.

The carbon savings, in terms of magnitude, are relatively moderate, a fact that is mainly attributed to the replacement probabilities in the model. GPU stands out with the highest savings of 0.17 Kgs per state due to its highest deterioration and significant contribution to carbon emissions. Regarding CPU there is a noticeable shift: Despite being initially considered the least likely component for individual replacement in the cost model (Figure 4.10), it leaps to the second position, with 0.1 Kgs savings when considering CO_2. This shift is primarily due to its substantial manufacturing carbon release, making it a crucial component to replace for meeting all the goals. In contrast, HDD, despite having the second-highest transition probabilities, is considered the least pollutant item, occupying the last place in terms of individual replacement with a relatively low savings rate at 0.05 Kgs.

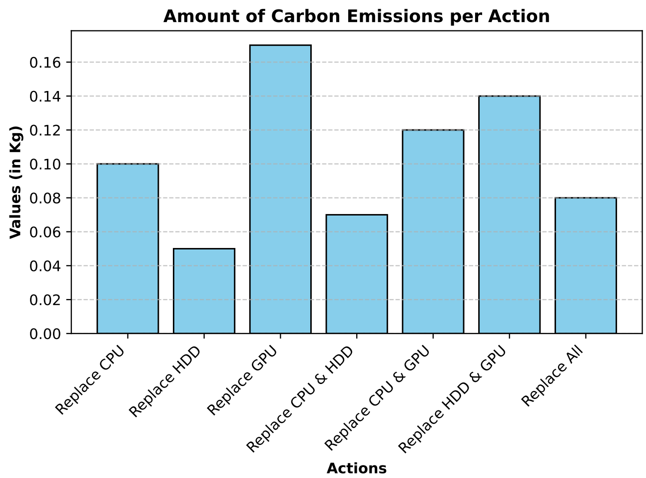

Figure 4.18: Amount of carbon dioxide emissions per state and action for three components (in Kgs)

Joint replacement actions exhibit remarkable saving rates, ranging from 0.07 to 0.14 Kgs. This behavior results as a mixture from the non-discounted emission values and the low replacement probabilities. If the latter were higher, the corresponding numbers would be more significant. The same applies also for the savings in case of full replacement. To enhance clarity, Figure 4.19 represents the corresponding percentages of each replacement action, given that an action is taken in the model.

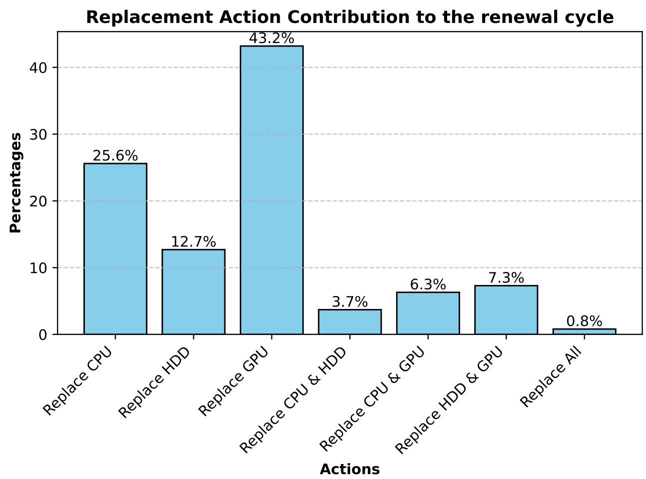

Figure 4.19: Contribution of each action among the replacement actions (in %)

By conditioning on each action taken, it becomes evident that the predominant contributor is the sole replacement of GPU. This dominance is a result of the interplay among the high deterioration probability, cost and carbon emissions. However, when it comes to joint replacements, Figure 4.19 highlights their relatively lower probabilities which weren't transparent in Figure 4.18. This phenomenon appears due to the low discount factor, established in the beginning of the problem.

Conclusion

This paper discusses the optimal replacement policy of low-end computer components during the End-Of-Life (EoL) phase, aiming to extend the overall lifespan of a desktop computer. The complication of the problem necessitates a mathematical formulation through discretization of time, establishing a framework for informed replacement decisions. The deterioration rates of components are captured using an exponential probability function for their modelling. Furthermore, the replacement cost for each component is determined as the average value derived from a set of low-end components obtained from a PC benchmark site (PassMark). Recognizing the complexity in studying all components of a desktop computer, this work focuses on the three major contributors to overall performance: GPU, CPU and HDD.

The Markov Decision Model (MDM) stands as an exact mathematical algorithm applied in this problem. It is described as an LP method where replacement actions are component-dependent. The outcome provides an overview of the average replacement cost over a renewal cycle. Insights derived from this method underscore the crucial role of the most deteriorating component (GPU) in decision making. This trend is clearly manifested in the probabilities of the transition matrix, which is the foundation of problem formulation. Nevertheless, a drawback of this approach lies in its assumption that components deteriorate independently of one another, displaying a non-realistic scenario.

To address this challenge, a slight modification is introduced, by creating interdependence among components. Accurately measuring stochastic dependency or all types of dependencies simultaneously is practically impossible. Therefore, an assumption is made to partially capture this concept: The actions for all components depend on the condition of CPU. In this scenario, the influence of GPU is reduced compared to original MDM, as replacement probabilities depend on the deterioration level of CPU.

In terms of computer performance, the outcome can be interpreted differently. In the two mentioned models, HDD exhibits the lowest probability of replacement per state but also the lowest replacement costs. Essentially, a significant performance boost can be achieved by replacing HDD as opposed to any other component. It is crucial to note that HDD contributes one fourth to the overall performance. Therefore, substituting HDD not only lowers average costs but also extends computer's lifespan, enhancing overall performance.

While MDM yields optimal solutions in terms of costs, its limitation lies in interpretability.

Understanding how this method operates might be challenging for individuals without technical background. Consequently, a heuristic approach called Power of Two is introduced. In this method, each component undergoes replacement after a specific number of periods, following a power of two (2^m) periodicity. Like MDM, deterioration rates of all studied components influence the model, as they affect the overall replacement cycles. Note that replacements in this approach are coordinated, rendering the joint cost of components as a crucial factor for the decision-making. However, this approach leads to sub-optimal results as components are not replaced in their optimal periods.

Last but not least, an alternative optimal replacing policy is presented, focusing both costs and carbon dioxide emissions. The choice of carbon dioxide as a metric stems from its impact among greenhouse gases and the environmental hazard it poses. The mathematical framework is rooted in the original MDM model but adapted to consider carbon emissions. The data for the analysis are sourced from the published CO_2 eq. emissions of a low end computer, the HP Pavilion Desktop PC TP01. It turns out that the component with the highest environmental impact (CPU) emerges as an essential factor in decision making.

Discussion

The paper employs the Markov Decision Model to calculate the average cost over the renewal cycle, focusing on three key components of a desktop computer. However, a computer comprises more than ten components in total, making it impractical to study all components due to extremely high number of states in the problem, and consequently, decision variables. Another limitation is the modelling of deterioration rates using probability functions, a commonly employed method in the literature for electronic products. However, it is acknowledged that this choice might not fully capture reality, as it could omit valuable information.

Furthermore, scientists have yet to comprehensively capture the dependencies among components. Currently, there is no exact method to measure these dependencies due to lack of understanding regarding how components interact. Additionally, there is scarcity of research, assessing two or more kinds of dependencies simultaneously. Therefore, a solid extension of this study could rely on assumptions about all kinds of dependencies with outlook for future improvement.

Another limitation is that the assessment of each component's contribution to carbon dioxide emissions relies on a single paper. Despite the scientific exploration into the subject of greenhouse emissions from electronics, only one study provides a detailed breakdown of CO_2 emissions by components. This particular paper derives its findings from a single computer, with no specification of whether it falls into the category of high or low-end computing. Consequently, the carbon dioxide data presented in this paper are based on a single-computer study without indication about its performance, resulting in a more generic presentation.

Finally, life extension strategies have attained considerably less attention compared to the regular maintenance approaches. This discrepancy can be attributed to the delayed appreciation of the former compared to the latter. However, the primary challenge lies in the utilization of a mathematical framework for decision making. The majority of papers either qualitatively describe or just employ simulation to assess the performance of a system, when implementing LTE.

References

Andrae, A. S. & Andersen, O. (2010). Life cycle assessments of consumer electronics—are they consistent? *The International Journal of Life Cycle Assessment*, *15*, 827–836.

Axsäter, S. (2015). *Inventory control* (Vol. 225). Springer.

Bocken, N. M., De Pauw, I., Bakker, C. & Van Der Grinten, B. (2016). Product design and business model strategies for a circular economy. *Journal of industrial and production engineering*, *33*(5), 308–320.

Budzinski, M., Bezama, A. & Thrän, D. (2020). Estimating the potentials for reducing the impacts on climate change by increasing the cascade use and extending the lifetime of wood products in germany. *Resources, Conservation & Recycling: X*, *6*, 100034.

Choi, B.-C., Shin, H.-S., Lee, S.-Y. & Hur, T. (2006). Life cycle assessment of a personal computer and its effective recycling rate (7 pp). *The International Journal of Life Cycle Assessment*, *11*, 122–128.

De Jonge, B. & Scarf, P. A. (2020). A review on maintenance optimization. *European journal of operational research*, *285*(3), 805–824.

Dekker, R., Wildeman, R. E. & Van der Duyn Schouten, F. A. (1997). A review of multi-component maintenance models with economic dependence. *Mathematical methods of operations research*, *45*, 411–435.

Duan, H., Eugster, M., Hischier, R., Streicher-Porte, M. & Li, J. (2009). Life cycle assessment study of a chinese desktop personal computer. *Science of the total environment*, *407*(5), 1755–1764.

Ekinci, E. & Ornek, A. M. (2007). A heuristic approach for determining lot sizes and schedules using power-of-two policy. *Journal of Applied Mathematics and Decision Sciences*, *2007*.

Fatimah, Y. A. & Biswas, W. K. (2016). Sustainability assessment of remanufactured computers. *Procedia CIRP*, *40*, 150–155.

Ferreboeuf, H., Berthoud, F., Bihouix, P., Fabre, P., Kaplan, D., Lefèvre, L. et al. (2019). Lean ict: Towards digital sobriety. *Report for the Think Tank The Shift Project*, *6*.

Fontana, A., Barni, A., Leone, D., Spirito, M., Tringale, A., Ferraris, M., ... Goncalves, G. (2021). Circular economy strategies for equipment lifetime extension: A systematic review. *Sustainability*, *13*(3), 1117.

Gräber, U. (2004). Advanced maintenance strategies for power plant operators—introducing inter-plant life cycle management. *International Journal of Pressure Vessels and Piping*, *81*(10-11), 861–865.

Han, J., Ijuin, H., Kinoshita, Y., Yamada, T., Yamada, S. & Inoue, M. (2021). Sustainability assessment of reuse and recycling management options for end-of-life computers-korean and japanese case study analysis. *Recycling*, *6*(3), 55.

Hewlett Packard. (2021). *Product carbon footprint report.* https://h20195.www2.hp.com/v2/GetDocument.aspx?docname=c08370555.

Hoang, A., Tseng, W., Viswanathan, S. & Evans, H. (2009). *Life cycle assessment of a laptop computer and its contribution to greenhouse gas emissions* (Unpublished doctoral dissertation). National University, San Diego.

ISO (International Organization for Standardization). (n.d.). *Environmental management-life cycle assessment-general principles and framework. iso 14040, geneva, switzerland.*

Jiajian Yan. (2023, July). *Data collection and optimization of component upgrading plans for personal computers.*

Jonnalagadda, S., Sinha, A., Hoffmann, P. & Schrage, D. (2005). Mission systems identification for the upgrade design analysis of anti-submarine maritime helicopters-an update.

Kang, H.-Y. & Schoenung, J. M. (2005). Electronic waste recycling: A review of us infrastructure and technology options. *Resources, Conservation and Recycling*, *45*(4), 368–400.

Keizer, M. C. O., Flapper, S. D. P. & Teunter, R. H. (2017). Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*, *261*(2), 405–420.

Khan, M. A., Mittal, S., West, S. & Wuest, T. (2018). Review on upgradability–a product lifetime extension strategy in the context of product service systems. *Journal of cleaner production*, *204*, 1154–1168.

Kusumo, R. & Sinha, A. K. (2002). Common development tools for an integrated decision support system for mid-life upgrade analysis. In *22nd congress of the international council of the aeronautical sciences* (Vol. 613, pp. 613–1).

Linster, M., Huq, F., Allen, D., Johnson, K., Doebrich, J., Beillard, C., … others (2015). *Material resources, productivity and the environment.* OECD Green Growth Studies. OECD, Paris.

Loubet, P., Vincent, A., Collin, A., Dejous, C., Ghiotto, A. & Jego, C. (2023). Life cycle assessment of ict in higher education: a comparison between desktop and single-board computers. *The International Journal of Life Cycle Assessment*, *28*(3), 255–273.

Lutz, C. & Giljum, S. (2017). Global resource use in a business-as-usual world up to 2030: updated results from the ginfors model. In *Sustainable growth and resource productivity* (pp. 30–41). Routledge.

Masanet, E., Price, L., Brown, R., Worrell, E. et al. (2005). Optimization of product life cycles to reduce greenhouse gas emissions in california.

McLellan, R., Grooten, M. & Almond, R. (2012). *Living planet report 2012: Biodiversity, biocapacity and better choices.* WWF International. (Includes bibliographical references)

Mobley, R. K. (2002). *An introduction to predictive maintenance.* Elsevier.

Nakagawa, T. (2005). *Maintenance theory of reliability.* Springer Science & Business Media.

Nijland, T. C., Atyeo, S. & Sinha, A. K. (2004). A simulation model for flight performance analysis of helicopter mid-life upgrade designs.

Oakley, J. L., Wilson, K. J. & Philipson, P. (2022). A condition-based maintenance policy for continuously monitored multi-component systems with economic and stochastic dependence. *Reliability Engineering & System Safety*, *222*, 108321.

Ouenniche, J. & Boctor, F. F. (2001). The multi-product, economic lot-sizing problem in flow shops: the powers-of-two heuristic. *Computers & Operations Research*, *28*(12), 1165–1182.

PassMark. (2023, May). *Cpu benchmarks.* https://www.cpubenchmark.net/year-on-year.html.

Peck, D. (2016). Longer lifetime for products: Benefits for consumers and companies.

Rachaniotis, N. & Pappis, C. (2008). Preventive maintenance and upgrade system: Optimizing the whole performance system by components' replacement or rearrangement. *International Journal of Production Economics*, *112*(1), 236–244.

Shafiee, M., Finkelstein, M. & Chukova, S. (2011). On optimal upgrade level for used products under given cost structures. *Reliability engineering & system safety*, *96*(2), 286–291.

Statista Research Department. (2023a, April). *Average age installed base pcs (desktop mobile) in the united states from 2017 to 2022 (in years).* https://www.statista.com/statistics/267474/average-life-of-pc-and-tablets/.

Statista Research Department. (2023b, March). *Consumer electronics - netherlands.* https://www.statista.com/outlook/cmo/consumer-electronics/netherlands.

Statista Research Department. (2023c, February). *Consumer electronics retailers - statistics facts.* https://www.statista.com/topics/7179/consumer-electronics-retailers/.

Statista Research Department. (2023d, September). *Desktop pc replacement cycle length in the united states from 2022 to 2027.* https://www.statista.com/statistics/267465/average-desktop-pc-lifespan/.

The TechSiting. (2018, July). *Pc hardware upgrade guide: Which pc components to upgrade.* https://www.techsiting.com/pc-hardware-upgrade-guide/.

Thomas, L. (1986). A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering*, *16*(4), 297–309.

Tijms, H. C. (2003). *A first course in stochastic models.* John Wiley and sons.

Umeda, Y., Daimon, T. & Kondoh, S. (2005). Proposal of decision support method for life cycle strategy by estimating value and physical lifetimes-case study. In *2005 4th international symposium on environmentally conscious design and inverse manufacturing* (pp. 606–613).

van Noortwijk, J. M. & Frangopol, D. M. (2004). Two probabilistic life-cycle maintenance models for deteriorating civil infrastructures. *Probabilistic engineering mechanics*, *19*(4), 345–359.

Weidema, B., Wenzel, H., Petersen, C. & Hansen, K. (2004). The product, functional unit and reference flows in lca. *Environmental News*, *70*, 1–46.

Wikipedia. (2023a, July). *Computer hardware.* https://en.wikipedia.org/wiki/Computer$_hardware.$

Wikipedia. (2023b, December). *Maintenance.* https://en.wikipedia.org/wiki/Maintenancecite$_note-1037C-5.$

Williams, E. (2004). Energy intensity of computer manufacturing: hybrid assessment combining process and economic input- output methods. *Environmental science & technology*, *38*(22), 6166–6174.

XDA - Developers. (2023, August). *Pc upgrade guide: How to choose the right parts for your machine.* https://www.xda-developers.com/pc-upgrade-choose-right-parts/upgrade-to-an-ssd-leave-th

Yang, S.-I., Frangopol, D. M. & Neves, L. C. (2006). Optimum maintenance strategy for deteriorating bridge structures based on lifetime functions. *Engineering structures*, *28*(2), 196–206.

Zafiropoulos, E. P. & Dialynas, E. N. (2004). Reliability and cost optimization of electronic devices considering the component failure rate uncertainty. *Reliability Engineering & System Safety*, *84*(3), 271–284.

Ziegler, L., Gonzalez, E., Rubert, T., Smolka, U. & Melero, J. J. (2018). Lifetime extension of onshore wind turbines: A review covering germany, spain, denmark, and the uk. *Renewable and Sustainable Energy Reviews*, *82*, 1261–1271.

Appendix A - Two components model

A.1 Markov Decision Model

For the case of studying two components (CPU and GPU), the Markov decision model consists of the states of each component, namely $i_1, i_2 \in I$ with $I = I_1 \times I_2$. At each combination of states an action is taken and there are four possible actions: No action ($a = 0$), replacement of CPU ($a = 1$), replacement of GPU ($a = 2$) or both components are replaced ($a = 3$). With mathematical notation the set A is described as,

$$A(i_1, i_2) = \begin{cases} \{3\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \in \{0, M\}, \\ \{1\} & \text{if } i_1 \in \{0, M\} \text{ and } i_2 \notin \{0, M\}, \\ \{2\} & \text{if } i_1 \notin \{0, M\} \text{ and } i_2 \in \{0, M\}, \\ \{0\} & \text{otherwise.} \end{cases}$$

where the constant M represents the maximum state of replacement. The transition probabilities moving from (i_1, i_2) to (j_1, j_2) by taking action a can be described from the matrix $p(i_1, i_2)(j_1, j_2)$(a).

$$p(i_1, i_2)(j_1, j_2)(0) = \begin{cases} (1 - p_n^1)(1 - p_n^2) & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ (1 - p_n^1)p_n^2 & \text{for } j_1 = i_1 + 1, j_2 = i_2, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ p_n^1(1 - p_n^2) & \text{for } j_1 = i_1, j_2 = i_2 + 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ p_n^1 p_n^2 & \text{for } j_1 = i_1, j_2 = i_2, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{else} \end{cases}$$

$$p(i_1, i_2)(j_1, j_2)(1) = \begin{cases} (1 - p_0^1)(1 - p_n^2) & \text{for } j_1 = 1, j_2 = i_2 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ (1 - p_0^1)^1 p_n^2 & \text{for } j_1 = 1, j_2 = i_2, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ p_0^1(1 - p_n^2) & \text{for } j_1 = 0, j_2 = i_2 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ p_0^1 p_n^2 & \text{for } j_1 = 0, j_2 = i_2, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{else} \end{cases}$$

$$
p(i_1,i_2)(j_1,j_2)(2) = \begin{cases} (1-p_n^1)(1-p_0^2) & \text{for } j_1 = i_1+1, j_2 = 1, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ (1-p_n^1)p_0^2 & \text{for } j_1 = i_1+1, j_2 = 0, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ p_n^1(1-p_0^2) & \text{for } j_1 = i_1, j_2 = 1, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ p_n^1 p_0^2 & \text{for } j_1 = i_1, j_2 = 0, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{else} \end{cases}
$$

$$
p(i_1,i_2)(j_1,j_2)(3) = \begin{cases} (1-p_0^1)(1-p_0^2) & \text{for } j_1 = 1, j_2 = 1, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ (1-p_0^1)p_0^2 & \text{for } j_1 = 1, j_2 = 0, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ p_0^1(1-p_0^2) & \text{for } j_1 = 0, j_2 = 1, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ p_0^1 p_0^2 & \text{for } j_1 = 0, j_2 = 0, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{else} \end{cases}
$$

where $1-p_n^i$ represents the probability of component i deteriorating from state n to state $n+1$ and p_n^i otherwise. Probability $1-p_0^i = 1$ denotes the probability of component i being replaced and starting from state 0.

The replacement costs are independent of the condition of the items, but depend on taken actions. Therefore, a replacement of CPU costs c_1, while a replacement of GPU costs c_2. The summation of these two costs results the full replacement. However, it is assumed that the full replacement is cheaper than replacing each component separately ($c_{full} < \sum_{i=1}^{n} x_i$). Therefore, a discount factor $\delta = 0.9$ is introduced. More precisely,

$$
c(a) = \begin{cases} 0 & \text{if } a = 0 \\ c_1 & \text{if } a = 1 \\ c_2 & \text{if } a = 2 \\ \delta(c_1 + c_2) & \text{if } a = 3. \end{cases}
$$

At this point, all the input data have been defined and the LP approach of the model is displayed (A.1). The mathematical formulation is identical with the one of Formulation 3.1 with its objective aiming to minimize the average costs. The distinction lies in the number of decision variables but also in the size of the transition matrix due to the fewer possible actions of the problem.

$$\min \sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{a \in A} c(a) x^a_{i_1,i_2} \tag{A.1a}$$

$$\text{subj. to} \sum_{a \in A} x^a_{i_1,i_2} = \sum_{j_1 \in I} \sum_{j_2 \in I} \sum_{a \in A} p_{(j_1,j_2)(i_1,i_2)}(a) \cdot x^a_{j_1,j_2} \qquad \forall i_1, i_2 \in I \tag{A.1b}$$

$$\sum_{i_1 \in I} \sum_{i_2 \in I} \sum_{a \in A} x^a_{i_1,i_2} = 1 \tag{A.1c}$$

$$x^a_{i_1,i_2} \geq 0 \qquad \forall i_1, i_2 \in I, \forall a \in A \tag{A.1d}$$

The objective function A.1a is designed to minimize the total average costs. Constraint A.1b outlines the balance equation: The inflow to a certain set of states equals the outflow. Constraint A.1c ensures that the summation of all probabilities equals one. Lastly, A.1d indicates the non-negative nature of the decision variables.

A.2 Dependent Deterioration

For the two-component case, the assumptions are also based on CPU's condition, given the lowest deterioration rate per year compared to others (Jiajian Yan, 2023), the relatively low cost but also the significant role of it in overall performance (XDA - Developers, 2023). The assumptions are summarized below:

1. If CPU deteriorates, GPU also deteriorates
2. If CPU does not deteriorate, GPU can deteriorate

To integrate these information into the problem, a few changes have to be done. Although the formulation and input data remain the same, outlined in A.1, adjustments to the transition probabilities are necessary. The subsequent sections provide detailed configurations within the transition probability matrices.

$$p(i_1,i_2)(j_1,j_2)(0) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ p_n^1 - p_n^2 & \text{for } j_1 = i_1, j_2 = i_2 + 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ p_n^2 & \text{for } j_1 = i_1, j_2 = i_2, i_1 \notin \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{else} \end{cases}$$

$$p(i_1,i_2)(j_1,j_2)(1) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = 1, j_2 = i_2 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{for } j_1 = 1, j_2 = i_2, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ p_n^1 - p_n^2 & \text{for } j_1 = 0, j_2 = i_2 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ p_n^2 & \text{for } j_1 = 0, j_2 = i_2, i_1 \in \{0, M\}, i_2 \notin \{0, M\} \\ 0 & \text{else} \end{cases}$$

$$p(i_1,i_2)(j_1,j_2)(2) = \begin{cases} 1-p_n^1 & \text{for } j_1 = i_1+1, j_2 = 1, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{for } j_1 = i_1+1, j_2 = 0, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ p_n^1 & \text{for } j_1 = i_1, j_2 = 1, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{for } j_1 = i_1, j_2 = 0, i_1 \notin \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{else} \end{cases}$$

$$p(i_1,i_2)(j_1,j_2)(3) = \begin{cases} 1 & \text{for } j_1 = 1, j_2 = 1, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{for } j_1 = 1, j_2 = 0, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{for } j_1 = 0, j_2 = 1, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{for } j_1 = 0, j_2 = 0, i_1 \in \{0,M\}, i_2 \in \{0,M\} \\ 0 & \text{else} \end{cases}$$

It is evident that the probability transition matrix is slightly less complicated than the one indicated in A.1. The non-zero entries in the matrices are determined by the condition of CPU, which dictates the overall system. In all other entries, probabilities have a value of zero.

Appendix B - Three components model

B.1 Transition Probabilities for Independent Deterioration

This section illustrates the remaining transition probabilities introduced in Section 3.2. They are depicted for each action within the problem framework.

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(1) = \begin{cases} (1-p_0^1)(1-p_n^2)(1-p_n^3) & \text{for } j_1=1, j_2=i_2+1, j_3=i_3+1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ (1-p_0^1)(1-p_n^2)p_n^3 & \text{for } j_1=1, j_2=i_2+1, j_3=i_3, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ (1-p_0^1)p_n^2(1-p_n^3) & \text{for } j_1=1, j_2=i_2, j_3=i_3+1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ (1-p_0^1)p_n^2p_n^3 & \text{for } j_1=1, j_2=i_2, j_3=i_3, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ p_0^1(1-p_n^2)(1-p_n^3) & \text{for } j_1=0, j_2=i_2+1, j_3=i_3+1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ p_0^1(1-p_n^2)p_n^3 & \text{for } j_1=0, j_2=i2+1, j_3=i_3, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ p_0^1p_n^2(1-p_n^3) & \text{for } j_1=0, j_2=i_2, j_3=i_3+1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \\ p_0^1p_n^2p_n^3 & \text{for } j_1=0, j_2=i_2, j_3=i_3, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\notin\{0,M\} \end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(2) = \begin{cases} (1-p_n^1)(1-p_0^2)(1-p_n^3) & \text{for } j_1=i_1+1, j_2=1, j_3=i_3+1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ (1-p_n^1)(1-p_0^2)p_n^3 & \text{for } j_1=i_1+1, j_2=1, j_3=i_3, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ p_n^1p(1-p_0^2)(1-p_n^3) & \text{for } j_1=i_1, j_2=1, j_3=i_3+1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ p_n^1(1-p_0^2)p_n^3 & \text{for } j_1=i_1, j_2=1, j_3=i_3, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ (1-p_n^1)p_0^2(1-p_n^3) & \text{for } j_1=i_1+1, j_2=0, j_3=i_3+1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ (1-p_n^1)p_0^2p_n^3 & \text{for } j_1=i_1+1, j_2=0, j_3=i_3, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ p_n^1p_0^2(1-p_n^3) & \text{for } j_1=i_1, j_2=0, j_3=i_3+1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \\ p_n^1p_0^2p_n^3 & \text{for } j_1=i_1, j_2=0, j_3=i_3, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\} \end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(3) = \begin{cases} (1-p_n^1)(1-p_n^2)(1-p_0^3) & \text{for } j_1=i_1+1, j_2=i_2+1, j_3=1, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ (1-p_n^1)p_n^2(1-p_0^3) & \text{for } j_1=i_1+1, j_2=i_2, j_3=1, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ p_n^1(1-p_n^2)(1-p_0^3) & \text{for } j_1=i_1, j_2=i_2+1, j_3=1, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ p_n^1p_n^2(1-p_0^3) & \text{for } j_1=i_1, j_2=i_2, j_3=1, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ (1-p_n^1)(1-p_n^2)p_0^3 & \text{for } j_1=i_1+1, j_2=i_2+1, j_3=0, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ (1-p_n^1)p_n^2p_0^3 & \text{for } j_1=i_1+1, j_2=i_2, j_3=0, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ p_n^1(1-p_n^2)p_0^3 & \text{for } j_1=i_1, j_2=i_2+1, j_3=0, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \\ p_n^1p_n^2p_0^3 & \text{for } j_1=i_1, j_2=i_2, j_3=0, i_1\notin\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\} \end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(4)=\begin{cases}(1-p_0^1)(1-p_0^2)(1-p_n^3) & \text{for } j_1=1,j_2=1,j_3=i_3+1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\(1-p_0^1)p_0^2(1-p_n^3) & \text{for } j_1=1,j_2=0,j_3=i_3+1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\p_0^1(1-p_0^2)(1-p_n^3) & \text{for } j_1=0,j_2=1,j_3=i_3+1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\p_0^1p_0^2(1-p_n^3) & \text{for } j_1=0,j_2=0,j_3=i_3+1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\(1-p_0^1)(1-p_0^2)p_n^3 & \text{for } j_1=1,j_2=1,j_3=i_3,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\(1-p_0^1)p_0^2p_n^3 & \text{for } j_1=1,j_2=0,j_3=i_3,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\p_0^1(1-p_0^2)p_n^3 & \text{for } j_1=0,j_2=1,j_3=i_3,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\\p_0^1p_0^2p_n^3 & \text{for } j_1=0,j_2=0,j_3=i_3,i_1\in\{0,M\},i_2\in\{0,M\},i_3\notin\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(5)=\begin{cases}(1-p_0^1)(1-p_n^2)(1-p_0^3) & \text{for } j_1=1,j_2=i_2+1,j_3=1,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)(1-p_n^2)p_0^3 & \text{for } j_1=1,j_2=i_2+1,j_3=0,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\p_0^1(1-p_n^2)(1-p_0^3) & \text{for } j_1=0,j_2=i_2+1,j_3=1,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\p_0^1(1-p_n^2)p_0^3 & \text{for } j_1=0,j_2=i_2+1,j_3=0,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)p_n^2(1-p_0^3) & \text{for } j_1=1,j_2=i_2,j_3=1,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)p_n^2p_0^3 & \text{for } j_1=1,j_2=i_2,j_3=0,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\p_0^1p_n^2(1-p_0^3) & \text{for } j_1=0,j_2=i_2,j_3=1,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\\p_0^1p_n^2p_0^3 & \text{for } j_1=0,j_2=i_2,j_3=0,i_1\in\{0,M\},i_2\notin\{0,M\},i_3\in\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(6)=\begin{cases}(1-p_n^1)(1-p_0^2)(1-p_0^3) & \text{for } j_1=i_1+1,j_2=1,j_3=1,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_n^1)(1-p_0^2)p_0^3 & \text{for } j_1=i_1+1,j_2=1,j_3=0,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_n^1)p_0^2(1-p_0^3) & \text{for } j_1=i_1+1,j_2=0,j_3=1,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_n^1)p_0^2p_0^3 & \text{for } j_1=i_1+1,j_2=0,j_3=0,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_n^1(1-p_0^2)(1-p_0^3) & \text{for } j_1=i_1,j_2=1,j_3=1,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_n^1(1-p_0^2)p_0^3 & \text{for } j_1=i_1,j_2=1,j_3=0,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_n^1p_0^2(1-p_0^3) & \text{for } j_1=i_1,j_2=0,j_3=1,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_n^1p_0^2p_0^3 & \text{for } j_1=i_1,j_2=0,j_3=0,i_1\notin\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(7)=\begin{cases}(1-p_0^1)(1-p_0^2)(1-p_0^3) & \text{for } j_1=1,j_2=1,j_3=1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)(1-p_0^2)p_0^3 & \text{for } j_1=1,j_2=1,j_3=0,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)p_0^2(1-p_0^3) & \text{for } j_1=1,j_2=0,j_3=1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_0^1(1-p_0^2)(1-p_0^3) & \text{for } j_1=0,j_2=1,j_3=1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\(1-p_0^1)p_0^2p_0^3 & \text{for } j_1=1,j_2=0,j_3=0,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_0^1(1-p_0^2)p_0^3 & \text{for } j_1=0,j_2=1,j_3=0,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_0^1p_0^2(1-p_0^3) & \text{for } j_1=0,j_2=0,j_3=1,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\\p_0^1p_0^2p_0^3 & \text{for } j_1=0,j_2=0,j_3=0,i_1\in\{0,M\},i_2\in\{0,M\},i_3\in\{0,M\}\end{cases}$$

B.2 Transition Probabilities for Dependent Deterioration

This section represents the rest of transition probabilities established in Section 3.3, outlining them per each action within the problem framework.

$$p(i_1, i_2, i_3)(j_1, j_2, j_3)(1) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = 1, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = 1, j_2 = i_2 + 1, j_3 = i_3, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = 1, j_2 = i_2, j_3 = i_3 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = 1, j_2 = i_2, j_3 = i_3, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ p_n^1 - p_n^2 & \text{for } j_1 = 0, j_2 = i_2 + 1, j_3 = i_3 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = 0, j_2 = i_2 + 1, j_3 = i_3, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ p_n^2 - p_n^3 & \text{for } j_1 = 0, j_2 = i_2, j_3 = i_3 + 1, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \\ p_n^3 & \text{for } j_1 = 0, j_2 = i_2, j_3 = i_3, i_1 \in \{0, M\}, i_2 \notin \{0, M\}, i_3 \notin \{0, M\} \end{cases}$$

$$p(i_1, i_2, i_3)(j_1, j_2, j_3)(2) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = i_1 + 1, j_2 = 1, j_3 = i_3 + 1, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = 1, j_3 = i_3, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ p_n^1 - p_n^3 & \text{for } j_1 = i_1, j_2 = 1, j_3 = i_3 + 1, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ p_n^3 & \text{for } j_1 = i_1, j_2 = 1, j_3 = i_3, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = 0, j_3 = i_3 + 1, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = 0, j_3 = i_3, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1, j_2 = 0, j_3 = i_3 + 1, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \\ 0 & \text{for } j_1 = i_1, j_2 = 0, j_3 = i_3, i_1 \notin \{0, M\}, i_2 \in \{0, M\}, i_3 \notin \{0, M\} \end{cases}$$

$$p(i_1, i_2, i_3)(j_1, j_2, j_3)(3) = \begin{cases} 1 - p_n^1 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ p_n^1 - p_n^2 & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ p_n^2 & \text{for } j_1 = i_1, j_2 = i_2, j_3 = 1, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2 + 1, j_3 = 0, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ 0 & \text{for } j_1 = i_1 + 1, j_2 = i_2, j_3 = 0, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ 0 & \text{for } j_1 = i_1, j_2 = i_2 + 1, j_3 = 0, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \\ 0 & \text{for } j_1 = i_1, j_2 = i_2, j_3 = 0, i_1 \notin \{0, M\}, i_2 \notin \{0, M\}, i_3 \in \{0, M\} \end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(4)=\begin{cases}1-p_n^1 & \text{for } j_1=1, j_2=1, j_3=i_3+1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=1, j_2=0, j_3=i_3+1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=0, j_2=1, j_3=i_3+1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=0, j_2=0, j_3=i_3+1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ p_n^1 & \text{for } j_1=1, j_2=1, j_3=i_3, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=1, j_2=0, j_3=i_3, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=0, j_2=1, j_3=i_3, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\\ 0 & \text{for } j_1=0, j_2=0, j_3=i_3, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\notin\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(5)=\begin{cases}1-p_n^1 & \text{for } j_1=1, j_2=i_2+1, j_3=1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=1, j_2=i_2+1, j_3=0, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=0, j_2=i_2+1, j_3=1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=0, j_2=i_2+1, j_3=0, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ p_n^1 & \text{for } j_1=1, j_2=i_2, j_3=1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=1, j_2=i_2, j_3=0, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=0, j_2=i_2, j_3=1, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=0, j_2=i_2, j_3=0, i_1\in\{0,M\}, i_2\notin\{0,M\}, i_3\in\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(6)=\begin{cases}1-p_n^1 & \text{for } j_1=i_1+1, j_2=1, j_3=1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1+1, j_2=1, j_3=0, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1+1, j_2=0, j_3=1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1+1, j_2=0, j_3=0, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ p_n^1 & \text{for } j_1=i_1, j_2=1, j_3=1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1, j_2=1, j_3=0, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1, j_2=0, j_3=1, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\\ 0 & \text{for } j_1=i_1, j_2=0, j_3=0, i_1\notin\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\}\end{cases}$$

$$p(i_1,i_2,i_3)(j_1,j_2,j_3)(7) = \begin{cases} 1 & \text{for } j_1=1, j_2=1, j_3=1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=1, j_2=1, j_3=0, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=1, j_2=0, j_3=1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=0, j_2=1, j_3=1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=1, j_2=0, j_3=0, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=0, j_2=1, j_3=0, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=0, j_2=0, j_3=1, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \\ 0 & \text{for } j_1=0, j_2=0, j_3=0, i_1\in\{0,M\}, i_2\in\{0,M\}, i_3\in\{0,M\} \end{cases}$$