

**CHALMERS**



**GÖTEBORGS UNIVERSITET**

# Experiments on optimal maintenance operations with emphasis on end-of-contract constraints

*Examensarbete för kandidatexamen i matematik vid Göteborgs universitet*

Sebastian Ahlman

Philip Nilsson

Kamran Tanha

Institutionen för matematiska vetenskaper  
Chalmers tekniska högskola  
Göteborgs universitet  
Göteborg 2017



# Experiments on optimal maintenance operations with emphasis on end-of-contract constraints

*Examensarbete för kandidatexamen i matematik vid Göteborgs universitet*

Kamran Tanha

*Examensarbete för kandidatexamen i tillämpad matematik inom matematikprogrammet vid Göteborgs universitet*

Sebastian Ahlman Philip Nilsson

Handledare: Michael Patriksson  
Quanjiang Yu

Examinator: Marina Axelson-Fisk  
Maria Roginskaya

Institutionen för matematiska vetenskaper  
Chalmers tekniska högskola  
Göteborgs universitet  
Göteborg 2017



# Populärvetenskaplig presentation

## Med teknikens utveckling följer stort ansvar

Överallt i den moderna världen används smartphones, bilar, flygplan och andra ting som underlättar för människan att leva och samverka. Användningen av mänskligt skapade föremål sker flitigt inom bl a infrastruktur, produktion och nöje och är en bidragande påverkan till hur världen förändras kulturellt och ekonomiskt.

Någonting som alla tekniska objekt emellertid har gemensamt är att deras prestanda försämras med tid och användning. Civilisationens alltmer ökade behov av tekniska redskap har medfört att man prioriterar skötseln av dem annorlunda idag än vad man gjorde under första halvan av 1900-talet, då man i princip bara ägnade sig åt att laga eller byta ut det som slutat fungera. Idag pratar man mer om hållbarhet och kostnadsbesparingar, skötseln av dessa tekniska objekt är med noggrann planering en nyckelfaktor till att uppnå båda syften.

## Strävan efter hållbar utveckling

I Sverige är t ex vindkraft en energiform som passar väldigt bra för svenska väderförhållanden. Även om faktorer som låga elpriser påverkar branschen så är det billigare än någonsin att installera vindkraftverk, enligt Energimyndigheten. Dock visar studierna även att det faktiskt är få nya installationer som blir gynsamma, bl a på grund av att komponenterna i kraftverket inte är exakt anpassade för den miljön de jobbar i, vare sig det är kallt, varmt, blåsig havsmiljö eller skog. Speciellt i dessa fall kan planerad underhåll vara väldigt avgörande.

## Dåligt underhåll kan ge negativa kedjereaktioner

Underhåll kan bli dyrt, speciellt med oförutsedda stopp på vindkraftverk som skapar extra utgifter och minskade intäkter. Alltifrån direkta kostnader för reservdelar, utrustning för jobbet, personal för transport och själva lagningen till minskad vinst pga produktionsbortfall pga att vindkraftverket var ur bruk. Ta till exempel vindkraft ut till havs där man även måste transportera delarna ut på havet och utföra underhållet under svåra förhållanden. Av både säkerhetsskäl och affärstänkande ligger det i aktörernas intresse att maskineri överlag fungerar som det ska och att det kostar så minimalt som möjligt. Det är inte hållbart för någon aktör att få stopp i produktionen tack vare dåligt underhåll, vare sig det är Volvo-bandet eller vindkraftverk.

## Planering ger flexibilitet

Systematiskt inleder man med att strategiskt se över hur mycket kapital man har tillgodo för jobbet och hur mycket av det man faktiskt kan utföra själv innan man vänder sig till hjälp utifrån. En taktisk planering inleds där man ser över hur mycket det finns av reservdelar och hur man ska tillsätta en bemanning som klarar av jobbet. När det väl gäller att utföra själva underhållet bör det vara klart i schemat när och hur många gånger man behöver utföra det, vad som behöver lagas eller bytas ut.

Att planera och utföra underhåll överhuvudtaget innebär nuförtiden att man bl a ska kunna förutspå när komponenter ska gå sönder genom att analysera dem tydligt och hitta betéenden i driften som tyder på att något kommer att falla. En sådan process kan ibland liknas vid när man t ex besiktar sin bil, där man får varningar om vissa bildelar, deras hälsa och status, vilka som måste och bör bytas ut snart osv. Om man i förväg vet vilka potentiella problem man riskerar att få så hinner man planera och utföra underhåll utifrån bestämda tillfällen istället för att skicka ut bemanning varje gång någon enstaka komponent går sönder utan förvarning.

## Matematikens roll i planeringen

Ett ungefärligt sätt att hitta de mest kritiska komponenterna är att man teoretiskt med hjälp av matematiken och datorer studerar olika scenarier för den totala kostnaden av att driva ett vindkraftverk. Så småningom visar alla uträkningar och tester ett slags mönsterbetående hos komponenter som gör det lättare för människor att schemalägga underhållstillfällena.

Man ger de kritiska komponenterna en uppskattad livslängd som är baserad på tidigare erfarenheter av samma komponenter. Dessa livslängder och alla priser gällande underhållstillfällena (inklusive reservdelar som tillsätts) använder man i matematiska modeller och så småningom kan man tack vare datorers uträkningar förutspå när underhållstillfällena skall ske. Det är mestadels tre faktorer som avgör när underhåll kommer att utföras: komponentkostnaden, kostnaden att utföra underhåll och livslängderna på komponenterna. Kvoten mellan de två kostnaderna har den största påverkan på när underhåll skall ske. När det gäller svåråtkomliga vindkraftverk ute på havet kan det t ex vara klokast att byta ut i princip alla delar, även delar med lång livslängd kvar då delarna som byts ut på plats faktiskt kan sluta upp billigare än själva underhållstillfallet.

Hela den industriella världen är beroende av tekniken och därmed av underhållsplaneringen som följer. Det är inte alltid lätt att veta detta, speciellt eftersom många människor förknippar underhåll med lagningen som sker när något går sönder utan att ha vetskap att man t ex bör tänka fler steg framåt och att underhåll faktiskt kan bli otroligt dyrt i längden.

## Sammanfattning

För att minimera onödiga kostnader som uppstår i det fall att maskiner eller andra objekt plötsligt fallerar och hastigt behöver bytas ut under mindre fördelaktiga tillfällen via så kallat reaktivt underhåll föredras istället en mer aktiv underhållspolicy genom vilket underhåll bedrivs i förebyggande syfte. Objekten i fråga underhålls alltså kontinuerligt baserat på uppmätta livslängder eller utifrån ett kostnadsperspektiv. För att schemalägga detta underhåll kan matematiska modeller användas med fördel. I dagens läge finns det ingen bevisat bästa modell inom det delfält av Operationsanalys kallat Underhållsplanering.

Syftet med detta projekt är därav via addition av nya bivillkor och simulering att analysera två av de mer prevalenta modellerna inom detta fält: The Opportunistic Replacement Problem(ORP) och The Preventative Maintenance Scheduling Problem with Interval Costs(PMSPIC).

Resultaten påvisar att för stora system i vilket kostnaderna är tidsberoende har schemaläggingskostnaden  $d_t = d$  mycket liten inverkan på schemalaggnings om kvoten  $\frac{d}{c_i}$  är liten. Inversen påvisas även för system bestående av endast ett fåtal komponenter. Det kan således antas att större fokus bör läggas på produktutveckling för att undvika onödigt stora underhållskostnader i system med ett fåtal komponenter och i vilket schemaläggingskostnaden är uppskattningsvis hög. På grund av tidsbrist gick det ej att genomföra någon kvantitativ jämförelse av båda modellerna. Det som presenteras för PMSPIC är således en kvalitativ diskussion kring dess teoretiska egenskaper och huruvida den kan liknas vid ORP.

## Abstract

To minimize the unnecessary costs that arise when machines or other objects fail, perhaps needing to be replaced during less than ideal occasions through what is known as reactive maintenance, a more active maintenance policy through which objects are continuously maintained in a preventative manner based on life expectancy or from a cost-perspective, is preferred. Mathematical models can be advantageous to utilize when scheduling these maintenance occasion. No model has at this time been shown to be the best within the sub field of Operations Research known as Maintenance Scheduling. The purpose of this project is to, through the addition of constraints and simulation, analyse two of the more prevalent models within this field: The Opportunistic Replacement Problem(ORP) and The Preventative Maintenance Scheduling Problem with Interval Costs(PMSPIC).

Results show that the set-up cost  $d_t$  for large system in which costs are time independent, i.e.  $d_t = d$ , carry little influence over the resulting optimal schedule if the quotient  $\frac{d}{c_i}$ , where  $c_i$  is the replacement cost of a component, is small. The inverse is shown to be true for systems consisting of a small number of components. It can thus be inferred that greater focus should be put on product development to avoid unnecessarily high maintenance costs in systems with few components and in which set-up costs are comparatively high. A quantitative comparison of both models was, unfortunately, not possible to perform due to time constraints. The results presented for the PMSPIC thus take the form of a qualitative discussion of its theoretical properties and how it can be likened to the ORP.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	On the importance of maintenance . . . . .	1
1.3	Purpose . . . . .	1
1.4	Maintenance theory . . . . .	2
1.5	Outline . . . . .	2
<b>2</b>	<b>Theory</b>	<b>3</b>
2.1	Linear integer programming . . . . .	3
2.2	Computational complexity and the concept of NP . . . . .	4
2.3	The Branch & Bound algorithm: An algorithm for the solution of discrete and combinatorial programs . . . . .	4
<b>3</b>	<b>Maintenance models</b>	<b>5</b>
3.1	The opportunistic replacement problem . . . . .	5
3.2	Complexity analysis of the ORP . . . . .	7
3.3	Integrality property . . . . .	7
3.4	The Preventive Maintenance Scheduling Problem with interval Costs . . . . .	8
3.5	Theoretical considerations: The PMSPIC as a generalization of the ORP . . . . .	10
<b>4</b>	<b>Tests and Results</b>	<b>11</b>
4.1	The ORP: Tests and Results . . . . .	11
4.1.1	A system of 500 components with time independent costs . . . . .	11
4.1.2	Prohibiting the scheduling of maintenance occasions during the last $k$ time steps . . . . .	12
4.1.3	A system of 50 components together with time independent costs . . . . .	13
4.1.4	A set of 50 components with the added constraint that a set amount of life $r$ is to be left in each component at the end of the contracted period . . . . .	13
4.1.5	Time dependent component cost $c_{it}$ and set up cost $d_t$ . . . . .	14
4.2	The PMSPIC: Tests and Results . . . . .	16
4.2.1	A system of 5 components with time independent costs . . . . .	16
<b>5</b>	<b>Discussion</b>	<b>17</b>
5.1	A discussion of the time independent cases . . . . .	17
5.2	The cases with time dependent monotonous costs . . . . .	18
5.3	The case with the PMSPIC . . . . .	18
5.4	Comparison of computation times . . . . .	19
5.5	Conclusion and thoughts on future experiments . . . . .	19
<b>6</b>	<b>Appendix</b>	<b>20</b>
6.1	The Traveling Salesman Problem . . . . .	20
6.2	The Set Covering Problem . . . . .	20



# Foreword

This bachelors project was carried out at the Department of Mathematical Sciences at Gothenburg University during the Spring of 2017. We would like to extend a heartfelt thank you to our supervisors PhD student Quanjiang Yu and Professor Michael Patriksson whose continued support and vast knowledge has carried us through many hardships.

Each member of the group has written his own time-log during the course of this project, in addition to a log book that has been updated on a daily basis. Sebastian has been responsible for the implementation of the ORP and the PMSPIC in AMPL and for the creation of graphs as well as writing most of the Latex code. Philip has been responsible for the understanding of the theoretical background of the ORP and the PMSPIC and has also held editorial responsibilities. Kamran has been responsible for understanding the maintenance procedures, and for the writing of the popular science article.

The author or authors of each specific section are specified below

- 1.1 - Kamran
- 1.2 - Main author Sebastian, co-author Kamran
- 1.3 - Main author Philip, co-author Kamran
- 1.4 - Kamran
- 1.5 - Philip
- 2 - Philip
- 3.1 - Sebastian
- 3.2 - Philip
- 3.3 - Philip
- 3.4 - Sebastian
- 3.5 - Philip
- 4.1.1 - Philip
- 4.1.2 - Main author Kamran, co-authors Sebastian and Philip
- 4.1.3 - Sebastian
- 4.1.4 - Sebastian and Philip
- 4.1.5 - Sebastian
- 4.2.1 - Philip
- 5.1 - Sebastian and Philip
- 5.2 - Sebastian and Philip
- 5.3 - Sebastian and Philip
- 5.4 - Philip
- 5.5 - Sebastian and Philip
- 6.1 - Philip
- 6.2 - Philip

# 1 Introduction

## 1.1 Background

Over the past 100 years, technology has evolved considerably and systems have become more complex over time which, in turn has also changed how maintenance is performed. Up to the 1940's, maintenance was mostly about fixing something that went out of order, it was an unavoidable cost, while today, it's viewed from a long-term perspective such as trying to prevent the components from failing within a certain time and actually reduce costs. The operation of modern machinery or facilities is nowadays more demanding than ever, especially when the health of all the components in the system needs to be on a level where they at least operate as they should while making it as cost-effective as possible [6, p. 14-15]. To understand it fully, we're going to talk first about the importance of maintenance, then the purpose of this project and of course about maintenance theory before we get to mathematical models and methods.

## 1.2 On the importance of maintenance

Everywhere in the modern world today you see computers, cars, air planes, energy plants, roads and rail roads, which are all examples of engineered objects. They are used by governments, companies and individuals for different purposes such as infrastructural or economic . However, all engineering objects have one thing in common; their condition and performance degrades with time and usage. Maintenance is therefore a key factor in extending the life-span of these objects.

Performing poor or no maintenance is unfavourable in many aspects, take a factory or a plant for example and how it could result in many important machines failing, which would result in costly repair or replacement. There would also be a lot of loss of production time since important machines are broken, which could also result in a chain of unnecessary expenses. In a plant, the maintenance budget represent 20 percent of the total budget, on average ([3, p. 290])

But the effects of poor maintenance are not always only economical, it can also lead to even worse consequences. Imagine what could happen if an aircraft engine or the brakes of a car were poorly maintained? It could result in a crash, which may be fatal. Maintenance also plays an important role in a well functioning infrastructure: if, for example, train tracks or a road would be in urgent need of service in the morning during high traffic hours, it could result in a total traffic stop which would affect many people in society. By developing a good maintenance policy, an organization may save lives and increase profitability. [6, p. 1-3]

## 1.3 Purpose

The purpose of this project is to study two different mathematical optimization models used within the field of maintenance scheduling, the ORP (opportunistic replacement problem) and the PMSPIC (preventive maintenance scheduling problem with interval costs). These models are to be implemented using varying parameters (such as cost/repair, maintenance occasions, time steps, etc). The results are then to be compared to the expected results and effects, and is then to be analysed and discussed.

## 1.4 Maintenance theory

Maintenance is the technical and administrative actions taken to repair or replace an item to a state where it's capable to perform its activity. Generally, maintenance involves two categories, Preventive Maintenance (PM) and Corrective Maintenance (CM). PM is implemented in order to avoid all type of problems which may affect the objects of a machine or factory. This includes activities like changing the oil of a certain equipment or other minor adjustments just to make sure the object lasts longer so one can avoid everything from failures to safety issues that may detriment the cost efficacy. While the PM is based on preventing future problems through controlling, maintaining and supporting the objects, the CM is different.

CM is performed after the occurrence of a problem in order to restore the object to its operational condition, or just simply replace it with another one that is in an operational condition. CM actions can vary from smaller repairs like replacing a lamp that's broken to bigger ones that requires more amount of resources. When increasing PM and paying for the services, the CM costs actually decrease and therefore the total cost of both types of maintenance is affected by the optimality of PM and CM.

## 1.5 Outline

Section 1 of this report provides a brief background to the underlying ideas of this project. Supplied is also a short overarching description of the concept of maintenance. Section 2 provides a general explanation of the theoretical background that is required to properly understand the two optimization models that are to be introduced.

Section 3 introduces the opportunistic replacement problem (ORP) and the preventative maintenance scheduling problem with interval costs (PMSPIC), together with an explanation of the constituents of both.

Section 4 presents the results produced through tests of the ORP and PMSPIC using different set-up cost, replacement cost and time-steps, accompanied by a short discussion on the results of each implementation. Contained within are also a few cases where additional new constraint have been added to each model.

Section 5 takes the form of a discussion of the results obtained through tests and simulations.

## 2 Theory

The beginning of this section introduces the notion of pure and mixed integer programs as well as some of their respective theoretical properties. We also introduce the concept of computational complexity and complexity classes P and NP. Lastly, we present the most common method used to find optimal solutions to integer programs known as the Branch & Bound algorithm.

### 2.1 Linear integer programming

A linear integer program is, in its purest form, a program that can be expressed as

$$\begin{aligned} & \text{maximize} && c^T x, \\ & \text{subject to} && Ax \leq b, \\ & && x \geq 0^n, \quad x \in \mathbb{Z}^n, \end{aligned} \tag{1}$$

where  $c \in \mathbb{R}^n$  is a row vector of coefficients,  $A \in \mathbb{R}^{m \times n}$  being the specific constraint matrix and with  $x \in \mathbb{Z}^n$  being the decision variables relating to the problem and the right-hand side  $b \in \mathbb{Z}^m$  is a column vector. The set  $S := \{x \geq 0^n, x \in \mathbb{Z}^n \mid Ax \leq b\}$  is the set of feasible solutions to the integer program described above, referred to as a pure integer linear set ([2, p. 2]). The focus of this project lies on two specific mathematical models used within the field of maintenance planning: The Opportunistic Replacement Problem (ORP) and the aptly named Preventative Maintenance Scheduling Problem with Interval Costs (PMSPIC). Both of these problems are programs of a mixed integer linear nature, a type of program whose general form can be expressed in a similar fashion as the pure integer programs mentioned above:

$$\begin{aligned} & \text{maximize} && c^T x + h^T y, \\ & \text{subject to} && Ax + Gy \leq b, \\ & && x \geq 0^n, \quad x \in \mathbb{Z}^n, \\ & && y \geq 0^p, \end{aligned} \tag{2}$$

where  $c$ ,  $A$  and  $x$  are the same as in the pure integer program (1),  $h \in \mathbb{R}^p$  is an additional row vector of coefficients,  $G \in \mathbb{R}^{n \times p}$  is a second constraint matrix and  $y \in \mathbb{R}^p$  is a second set of decision variables. The set of feasible solutions pertaining to problems of this kind would then be  $S := \{(x, y) \geq 0^n, (x, y) \in \mathbb{Z}^n \times \mathbb{R}^p \mid Ax + Gy \leq b\}$ . One fact worthy of note is that the pure problem presented above is nothing but a special case of the mixed problem where  $p = 0$ , a choice which kills both  $y$  and  $G$ . What is left is precisely equal to the pure integer program ([2, p. 3]).

One, if not the main question to ask here would be: Are integer programs easier to solve than purely linear programs? The answer to that question is twofold: a tentative yes and a tentative no. Easier in the sense that integer programs are a subclass of discrete programs and it can thus be inferred that the solution space pertaining to an integer program must also be discrete and contain a finite number of elements. Thus, complete enumeration and comparison of enumerated solutions is certainly possible when  $n$  is sufficiently small ([2, p. 4]). The computation time, however, grows incredibly fast as the number of decision variables increases, meaning that complete enumeration quickly becomes impractical. To exemplify this behaviour it is practical to consider a problem that is theoretically simple yet very computationally taxing: The Travelling salesman problem (TSP). A more formal formulation of the TSP can be found under section 6.1. The TSP is an optimization problem in which one asks the question: given a set  $V$  of  $n$  connected nodes, is it possible to find a route that minimizes the cost of a route visiting each node only once while also returning to the initial node? The process of solving such a problem consists of two steps: the first being to produce a feasible solution, and the second being to ask if the given solution is optimal. An increase in the number of nodes leads to a combinatorial explosion  $n!$  of possible

solutions, which makes the problem computationally difficult. This, coupled with the fact that the theory for the solution of linear programs not necessarily being true for programs of the integer variety means that most solution algorithms developed for use with LP's are non-functional or extremely inefficient when dealing with IP's. Thusly, the question becomes not one of difficulty, but of computational efficiency.

## 2.2 Computational complexity and the concept of NP

The Travelling salesperson problem, as presented in the previous section, is part of an overarching structure of problems more commonly known as decision problems. This class of problem can be loosely described as the following: given a set of inputs and their respective outputs, is the answer yes or no? Optimization problems of all forms naturally fall into this category as the underlying idea of optimization is to not only solve a given problem but also to find the best solution to that particular problem. An important complexity class is NP, an abbreviation of "nondeterministic, polynomial time". This class can be used to describe a few types of decision problems, such as the aforementioned Travelling salesperson problem. An intuitive, if somewhat informal way of defining this class is to think of it as all decision problems for which the existence of the answer "yes" can be proven in an efficient manner. In other words, there exists an algorithm that can be used to solve the problem within a reasonable time-frame. Contained within NP are two especially important types of problems that are referred to as NP-hard and NP-complete. A decision problem  $Q$  in NP is said to be NP-complete if all problems  $D$  in NP are reducible to  $Q$  in polynomial time. I.e, there exists a polynomial algorithm that, for every instance  $I$  of  $D$  produces an instance of  $Q$  whose answer is "yes" if and only if the answer to  $I$  is "yes" ([2]). A problem  $Q$ , not necessarily in NP, is said to be NP-hard if all problems  $D$  in NP are reducible to  $Q$  in polynomial time. In particular, integer programming is NP-hard ([2]).

The next section introduces and provides an explanation of the theory and underlying ideas behind one of the more common solution algorithms used within integer programming: The Branch & Bound algorithm.

## 2.3 The Branch & Bound algorithm: An algorithm for the solution of discrete and combinatorial programs

Section 2 laid plain the fact that integer programs are inherently difficult to solve using methods and algorithms developed for the solution of linear programs. This prompted the development of alternative solution methods. One of these methods, which has proven itself to be most effective when it comes to solving integer programs, is the Branch & Bound algorithm ([2, p. 5]). Let  $M$  be the mixed integer linear program defined as

$$z = \max\{c^T x + h^T y : (x, y) \in S\}$$

where  $S = \{(x, y) \in \mathbb{Z}^n \times \mathbb{R}^p : Ax + Gy \leq b\}$  and let  $(x^0, y^0)$ ,  $z^0$  be the optimal solution and value to the linear relaxation of  $M$ , defined as

$$z_{LP} = \max\{c^T x + h^T y : (x, y) \in P_0\}$$

where  $P_0$  is the linear relaxation of  $S$ . The Branch & Bound algorithm operates on the assumption that one or more components of the optimal solution  $(x^0, y^0)$  to this relaxed problem is fractional and the underlying solution technique can be loosely described as follows, while also noting that we wish to define the algorithm such that it is valid for the case of binary variables. Let the index  $j$  be such that  $1 \leq j \leq n$ ,  $j \in \mathbb{Z}$  and choose  $x_j^0$  such that it is fractional. By letting  $f = x_j^0$  define this fractional value it is possible to define the sets

$$S_1 = S \cap \{(x, y) : x_j \leq 0\}, \quad S_2 = S \cap \{(x, y) : x_j \geq 1\}$$

where 0 is the largest integer such that  $0 \leq f$  and 1 is the smallest integer such that  $1 \geq f$ . Note that this definition restricts  $f$  to be in the set  $\{0, 1\}$  and is thus valid for the case of binary variables  $x_j$ . The fact that  $x_j$  is an integer for every pair  $(x, y) \in S$  now implies that

$(S_1, S_2)$  is a partition of  $S$  ([2, p. 6]). It is now possible to define the following two integer subproblems ([2, p. 6])

$$M_1 : \max\{c^T x + h^T y : (x, y) \in S_1\}, \quad M_2 : \max\{c^T x + h^T y : (x, y) \in S_2\}$$

A consequence of this partitioning of  $S$  is that the optimal solution to  $M$  will now be the best solution among all feasible solutions to  $M_1$  and  $M_2$  ([2, p. 7]). Thusly, the original problem has been reduced to solving two new subproblems and if the optimal solution to either of  $M_1$  or  $M_2$  contains a component that is fractional the partitioning process is repeated. If we let  $(x^i, y^i)$  be an optimal solution to  $M_i$ , the actual algorithmic process can be more formally defined as follows ([2, p. 7])

- Given an integral vector  $x^i$ , the optimal solution  $(x^i, y^i)$  to  $M_i$  provides a lower bound on the optimal solution to  $M$  since  $S_i \subseteq S$ , which implies that  $z_i^* \leq z^*$ .
- If  $x^i$  is not an integral vector and the optimal solution  $z_i$  is smaller than the current best lower bound, then the current branch is discarded since  $S_i$  does not contain a better solution.
- If  $x^i$  is not an integral vector and  $z_i$  is greater than the current best lower bound, it is possible for  $S_i$  to contain an optimal solution to  $M$ . Let  $x_j^i$  be a fractional component of the vector  $x^i$  and define the sets  $S_{i1} = S_i \cap \{(x, y) : x_j' \leq 0\}$ ,  $S_{i2} = S_i \cap \{(x, y) : x_j' \geq 1\}$  and repeat the process.

The next section will serve as an introduction to the main focus of this project, the optimization problem known as The Opportunistic Replacement Problem.

### 3 Maintenance models

This chapter introduces the Opportunistic Replacement Problem as well as the Preventative Maintenance Scheduling Problem with Interval Costs. Included are also brief explanations of some important properties of both models such as the integrality property.

#### 3.1 The opportunistic replacement problem

We present the Opportunistic Replacement Problem as it is presented in Almgren et. al(2012). The opportunistic replacement problem (ORP) is a model made to minimize the cost of maintaining an object during a given period of time. Given the costs, this model will produce an optimal maintenance schedule over a given time period  $\mathcal{T} = \{1, \dots, T\}$  for a system containing a set of components  $\mathcal{N} = \{1, \dots, N\}$ . There are two types of costs to be considered:  $d_t$  and  $c_{it}$ , which are the parameters of the model. These two parameters are defined as follows:

- $d_t$  is the cost incurred when a maintenance occasion is scheduled at time  $t \in \mathcal{T}$ . In other words, the cost to perform maintenance at an occasion excluding the cost to replace components.
- $c_{it}$  is the cost to replace component  $i \in \mathcal{N}$  during a maintenance occasion planned at time  $t \in \mathcal{T}$ .

These parameters are sometimes time-independent and are then denoted  $d$  and  $c_i$ . We define the following variables:

$$x_{it} = \begin{cases} 1, & \text{if and only if component } i \in \mathcal{N} \text{ is to be replaced at time } t, \\ 0, & \text{otherwise,} \end{cases}$$

$$z_t = \begin{cases} 1, & \text{if and only if maintenance occasion is scheduled at time } t \in \mathcal{T}, \\ 0, & \text{otherwise.} \end{cases}$$

We can now express our objective, which is to minimize the total cost. The cost of component replacement,

$$\sum_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{N}} c_{it} x_{it} \right),$$

together with the total cost of scheduling maintenance at each occasion,

$$\sum_{t \in \mathcal{T}} d_t z_t.$$

These two sums define our objective function.

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} \left( \sum_{i \in \mathcal{N}} c_{it} x_{it} + d_t z_t \right), \quad (3a)$$

$$\text{subject to} \quad \sum_{t=l+1}^{T_i+l} x_{it} \geq 1, \quad l = 0, \dots, T - T_i, \quad i \in \mathcal{N}, \quad (3b)$$

$$x_{it} \leq z_t, \quad t = 1, \dots, T, \quad i \in \mathcal{N}, \quad (3c)$$

$$x_{it} \geq 0, \quad t = 1, \dots, T, \quad i \in \mathcal{N}, \quad (3d)$$

$$z_t \leq 1, \quad t = 1, \dots, T, \quad (3e)$$

$$x_{it}, z_t \in \{0, 1\}, \quad t = 1, \dots, T, \quad i \in \mathcal{N}. \quad (3f)$$

The constraints are defined as follows:

- (3b): this constraint ensures that each component is replaced before or at the end of its life;
- (3c): this constraint ensures that the fixed cost  $d$  is paid every time one or more parts are replaced at time  $t$ ;
- (3d),(3e): these constraints are required by necessity;
- (3f): this constraint ensures that the decision variables assume only binary values.

### 3.2 Complexity analysis of the ORP

**Theorem:** *The set covering problem is polynomially reducible to the opportunistic replacement problem.*

We present the following proof as given in Almgren et al.(2012). A short introduction to the Set Covering Problem can be found under section 6.2.

**Proof:** Let  $\{A\}_{t=1}^m$  be a given collection of nonempty subsets of the finite set  $\{1, \dots, n\}$  such that  $\cup_{t=1}^m A = \{1, \dots, n\}$ . By letting  $a_{it} = 1$  if  $i \in A_t$  and 0 otherwise, it is possible to formulate the set covering problem as the following optimization problem:

$$\text{minimize } \sum_{t=1}^m y_t, \quad (4a)$$

$$\text{subject to } \sum_{t=1}^m a_{it} y_t \geq 1, \quad i = 1, \dots, n, \quad (4b)$$

$$y_t \in \{0, 1\}, \quad t = 1, \dots, m. \quad (4c)$$

Now, consider an instance of the ORP where  $N = n$ ,  $T = m$ ,  $d_t = 1$ ,  $T_i = m$ , and  $c_{it} = \delta(1 - a_{it})$ ,  $i = 1, \dots, n$ ,  $t = 1, \dots, m$ , where  $\delta > 1$ . Since  $T_i = T = m$ , each component needs to be replaced at least once between 1 and  $T$ . Furthermore, for every optimal solution and for each  $i$  and  $t$  such that  $a_{it} = 0$ , it must also hold that  $x_{it} = 0$  as  $c_{it} = \delta > 1 = d_t$  and there exists a  $\tilde{t} \in \mathcal{T}$  with  $a_{i\tilde{t}} = 1$ , which implies that  $c_{i\tilde{t}} = 0$ .

Given these specific parameters, the ORP can be reformulated as the following problem:

$$\text{minimize } \left\{ \sum_{t=1}^m z_t \mid \sum_{t=1}^m a_{it} x_{it} \geq 1, \quad i = 1, \dots, n \quad \text{and (3a) - (3f) hold} \right\}. \quad (4d)$$

An optimal solution  $(x^*, z^*)$  is given by:

$$z^* \in \operatorname{argmin}_{z \in \{0,1\}^m} \left\{ \sum_{t=1}^m z_t \mid \sum_{t=1}^m a_{it} z_t \geq 1, \quad i = 1, \dots, n. \right\} \quad (4e)$$

and  $x_{it}^* = a_{it} z_t^*$ ,  $i = 1, \dots, n$ ,  $t = 1, \dots, m$ . Since the programs defined by (4a) - (4c) and (4e) are equivalent, the result follows.  $\square$

As the set covering problem is NP-Complete ([1]), it must follow that the set covering problem, in its optimization problem formulation, is NP-hard. A consequence of this is then that the opportunistic replacement problem is NP-hard.

### 3.3 Integrality property

Assigning binary values to the variable  $z_t$  such that  $\hat{z}_t \in \{0, 1\}$  and defining the set  $\hat{\tau} = \{t \in \tau \mid \hat{z}_t = 1\}$  leads to a decomposition of the ORP over the components  $i \in N$ , which implies that  $x_{it}$  can be relaxed. Thus, the problem is reduced to the following:

$$\text{minimize } \left\{ \sum_{i \in N} \sum_{t \in \hat{\tau}} x_{it} c_{it} \mid x_{it} = 0, \quad t \in \tau \setminus \hat{\tau}, \quad (3b) \text{ and } (3f) \text{ hold} \right\}$$

**Proposition:** *The polyhedron  $P \in \mathbb{R}^{NT}$  defined by (3b) and*

$$x_{it} \leq 1, \quad t \in \hat{\tau}; \quad x_{it} \leq 0, \quad t \in \tau \setminus \hat{\tau} \quad (4f)$$

*possesses integral extreme points.*



**Proof:** The constraint matrix  $A$  that corresponds to the system of inequalities that is defined by (3b) and (4f) possesses the consecutive ones property. That is, for all rows  $i$ , if  $a_{ik} = a_{ij} = 1$  then  $a_{il} = 1$  for all  $k < l < j$ . This property, in turn, implies that  $A$  is totally unimodular, meaning that each square integer submatrix  $B$  of  $A$  is such that  $\det(B) \in \{-1, 1\}$ . Thus, given a totally unimodular matrix  $A$  it follows from the definition that the elements of  $A$  must be 1, 0 or -1. Since the right-hand sides of both (3b) and (4f) are integral this would imply that the corresponding polyhedron is integral ([3]).  
□

A consequence of this property is that when the problem is solved using methods that seek extreme points of polyhedral sets, say, the Simplex method, the constraints  $x_{it} \in \{0, 1\}$  can be relaxed, that is  $x_{it} \in [0, 1]$ , when  $z_t$  is fixed and the solution will remain integral. ([3, p. 296])

### 3.4 The Preventive Maintenance Scheduling Problem with interval Costs

We present the Preventative Maintenance Scheduling Problem with interval Costs as presented by Bangalore(2016). The Preventive Maintenance Scheduling Problem with Interval Costs (PMSPIC) is a model similar to the ORP made to minimize the cost of maintaining an object during a given period of time. Given the costs, this model produces an optimal maintenance schedule over a given time period  $\mathcal{T} = \{1, \dots, T\}$ , in a similar fashion to the ORP. However, even if the models perform the same function there are major differences in how that it is performed. The PMSPIC works in intervals defined as follows:

$$I = \{(s, t) \mid s \in \mathcal{T}, t \in \{s + 1, \dots, T + 1\}\}.$$

If maintenance is scheduled for component  $i$  at times  $s$  and  $t$ , but not in between, then we have a maintenance interval  $(s, t)$ . Every interval has a corresponding interval cost  $C_{st}^i$ . The parameter  $C_{st}^i$  can be explained as the cost of maintaining component  $i$  at time  $s$  and  $t$ , but not in between.  $d_t$  is defined exactly like in the ORP. An advantage that this model holds over the ORP is that the PMSPIC is capable of considering the schedules of a collection of several units, each unit containing a set of components  $\mathcal{N}$ , is denoted as  $m$ , where  $m$  is part of the collection of units  $\mathcal{M}$ . The decision variables can then be defined as follows:

$$x_{st}^{mi} = \begin{cases} 1, & \text{if and only if maintenance is scheduled for component} \\ & i \text{ in unit } m \text{ at times } s \text{ and } t \in I \\ 0, & \text{otherwise,} \end{cases}$$

$$z_t^m = \begin{cases} 1, & \text{if and only if maintenance is scheduled in unit } m \in \mathcal{M} \text{ at time } t \in \mathcal{T} \\ 0, & \text{otherwise.} \end{cases}$$

We can now define our objective. Similarly to the ORP

$$\sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} d_t z_t^m,$$

is the total cost of scheduling maintenance at each occasion and

$$\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{N}} \sum_{(s,t) \in I} C_{st}^i x_{st}^{mi}$$

is the total cost of replacing components in the time interval. Together they add up to:

$$\text{minimize } \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} d_t z_t^m + \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{N}} \sum_{(s,t) \in I} C_{st}^i x_{st}^{mi}, \quad (5a)$$

$$\text{subject to } \sum_{s=1}^{t-1} x_{st}^{mi} \leq z_t^m, \quad i \in \mathcal{N}, \quad m \in \mathcal{M}, \quad t \in \{2, \dots, T\}, \quad (5b)$$

$$\sum_{s=1}^{t-1} x_{st}^{mi} = \sum_{s=t+1}^{T+1} x_{ts}^{mi}, \quad i \in \mathcal{N}, \quad m \in \mathcal{M}, \quad t \in \{2, \dots, T\}, \quad (5c)$$

$$\sum_{s=2}^{T+1} x_{1s}^{mi} = 1, \quad i \in \mathcal{N}, \quad m \in \mathcal{M}, \quad (5d)$$

$$x_{st}^{mi} \in \{0, 1\}, \quad i \in \mathcal{N}, \quad m \in \mathcal{M}, \quad (s, t) \in I, \quad (5e)$$

$$z_t^m \in \{0, 1\}, \quad m \in \mathcal{M}, \quad t \in \mathcal{T}. \quad (5f)$$

The constraints are defined as follows:

- (5b) this constraint ensures that component  $i$  in unit  $m$  is maintained at the time  $t$  when the interval ends;
- (5c) this constraint ensures that when a maintenance interval ends for a component  $i$  in unit  $m$  the next one starts at the same time  $t$  except for when  $t = T + 1$ ;
- (5d) this constraint ensures that every maintenance interval starts at time  $t = 2$  for each component  $i$ ;
- (5e) and (5f) are the same as constraint (3f) in the ORP.

The advantage with the PMSPIC is that if a component fails close to a scheduled maintenance occasion, the maintenance can be performed earlier than scheduled. In other words, the PMSPIC can be used to reschedule planned maintenance occasions such that costs are decreased. The interval costs  $C_{st}^i$  are defined by

$$C_{st}^i = C_i^{PM} + M_i(t - s).$$

Where  $C_i^{PM}$  is the cost of the preventive maintenance and  $M_i(t - s)$  is what is referred to as the deterioration cost for the time interval  $(t - s)$ , which is defined by:

$$M_i(r) := E \left[ \sum_j^K I_{F_j \leq t} G(F_j, r) \right] Rn_i(t).$$

Where  $K \gg 1000$  is a large number and  $Rn_i$  is the total numbers of renewals(replacements) in the interval  $(1, t)$  for component  $i$ , which is calculated by the renewal function.  $F_j$  (In the specific case considered here  $F_j = T_i$ ) is the survival time for component  $i$ , which is a random variable for failures of component  $i$  derived from a Weibull distribution.  $\mathbb{1}$  is the indicator function defined by:

$$\mathbb{1} = \begin{cases} 1, & \text{if and only if } F_j \leq t, \\ 0, & \text{otherwise.} \end{cases}$$

$G(s, t)$  is the cost function defined as follows:

$$G(p, q) = (C_i^{CM} + d) - \left(\frac{p}{q}\right)^\lambda (C_i^{PM} + d), \quad i \in \mathcal{N}, \quad p \in \mathcal{T} \quad q \in \mathcal{T}.$$

The cost function is formulated in a flexible way that allows re-scheduling when a part fails close to a scheduled maintenance occasion, if it will decrease the cost.  $\lambda$  is a parameter used to decide the re-scheduling effect on the cost.  $\lambda$  has a fixed value 3 based on suggestion ([4, p. 49]).

### 3.5 Theoretical considerations: The PMSPIC as a generalization of the ORP

The PMSPIC is very much an evolution and generalization of the ORP that also considers systems consisting of components that are divided into several units as described in section 3.4. A consequence of this is that the PMSPIC shares some of the properties of its progenitor. One of the more notable being the integrality property described in section 3.3. This is, however, not simply due to similarities in the formulation, but a product of the fact that the PMSPIC can be viewed as a network flow problem when  $z_t$  is fixed and the integrality requirement on  $x_{st}^i$  is relaxed ([5, p. 393]).

Consider a graph with a total of  $N$  subgraphs, where  $N$  is the total number of components in the system considered. Each subgraph consists of  $T + 1$  nodes. The "flow" that is being transported, starting at node 0 and ending at node  $T + 1$ , are the components of the system. Setting  $x_{st}^i = 1$  would thus correspond to connecting node  $s$  in the  $i$ :the subgraph to node  $t$  with a directed arc in the same graph. Setting  $z_t = 1$  would then, in a similar fashion, correspond to allowing access to node  $t$  in each subgraph.

As mentioned in section 3.4, one of the main differences between the ORP and the PMSPIC is that the replacement costs  $c_{st}^i$  are defined as interval costs which behave in a different manner than the costs  $c_{it}$  in the introduction of the ORP in section 3.1. It is however possible to choose these costs such that the PMSPIC becomes equivalent to the ORP ([5, p. 392]). Let  $d_t$  be the set-up cost for an instance of the PMSPIC, and let the interval costs be  $c_{st}^i := c_{it}$  when  $t - s \leq T_i$ , and  $c_{st}^i := T(\max_{u \in T} \{d_u\} + n \max_{j \in N, u \in T} \{c_{ju}\}) + 1$  when  $t - s \geq T_i + 1$ , where  $T_i$  is the lifetime of component  $i \in N$ . In other words, by letting the replacement costs be such that it is cheaper to replace a component as long as the replacement is scheduled within the replacement interval defined by  $T_i$  while also letting the costs increase tremendously if the replacement is scheduled after the replacement interval, then we have arrived at a model that is equivalent to the ORP. Any optimal solution to this instance of the PMSPIC will thus also be optimal to the ORP [5, p. 392].

## 4 Tests and Results

### 4.1 The ORP: Tests and Results

#### 4.1.1 A system of 500 components with time independent costs

We are going to use the ORP model to find the minimal cost for maintaining 500 components in a 10 year time interval. Maintenance can be performed every month, which gives us  $T = 120$  discrete time steps. We have the fixed replacement cost  $c_{it} = c_i$  and specific lifetime  $T_i$  for every component  $i$ . The cost  $d_t = d$  to perform maintenance at each occasions is time independent. The ORP is solved for the cases when  $d = 0, 10, 100, 1000$ .

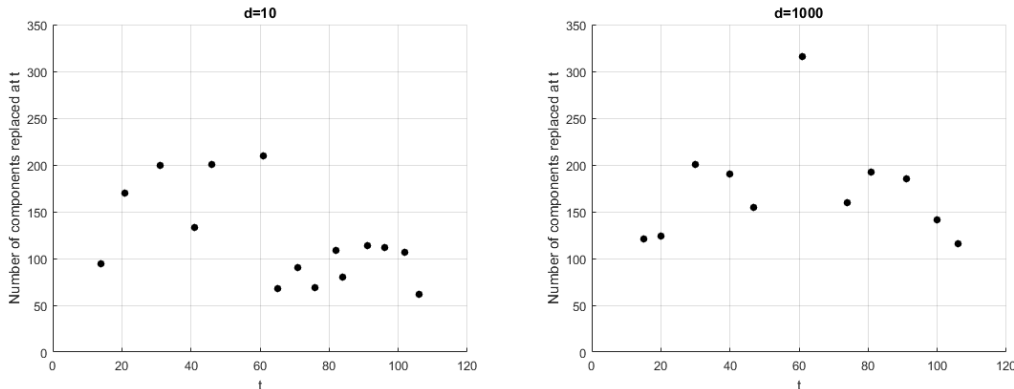


Figure 1: Optimal maintenance schedules for a system of 500 components.

The graphs correspond to  $d_t = 10$  and  $1000$ , respectively. As can be expected, the differing  $d$ 's carry a large influence over the optimal scheduling.

The above set of graphs represent a system of 500 components with time independent replacement costs  $c_{it} = c_i$  for each component  $i \in N$  together with a constant set-up cost  $d_t = d$ . The left graph corresponds to a set-up cost  $d = 10$ . This results in an optimal schedule where each component is replaced close to the end of its lifetime, leading to a total of 15 maintenance occasions and an optimal objective value of 31966, of which 150 is owed due to the set-up cost. Since the set-up cost is very small in comparison to the replacement cost the impact of  $d$  upon where the occasions are scheduled.

Multiplying the set-up cost by a factor of 100 yields a schedule with fewer maintenance occasions as the ratio  $\frac{d}{c_i}$  is now sufficiently large to have an appreciable impact upon the optimal schedule. This new schedule contains a total of 11 schedule occasions and produces an optimal objective value of 44146 of which 11000 is owed to the set up cost. Test were also carried out for  $d = 10$  and  $d = 100$ , respectively. It turns out, however, that the optimal solutions for  $d = 0, 10$  and  $100$  are all optimal for the case when  $d = 0$  as the only difference in optimal value between them is the difference in set-up cost. There is, in other words, some form of degeneracy of solutions at work here and we have thus chosen to not present the schedules for  $d = 0$  and  $d = 100$ . A discussion on the possible causes of this phenomenon can be found in section 5.

### 4.1.2 Prohibiting the scheduling of maintenance occasions during the last $k$ time steps

We introduce a new constraint which purpose is to prohibit the scheduling of maintenance occasions when  $t \in \{(T - k) \dots T\}$ .

$$\sum_{i \in \mathcal{N}} \sum_{t=T-K+1}^T x_{it} = 0$$

This new constraint ensures that no maintenance is performed during the last  $k$  time steps. The  $K$ -value can be no larger than  $\min\{T_i\} - 1$  otherwise the solution will not be feasible. This is because there are  $K + 1$  time steps from  $t = T - K$  to  $T$ , where we don't perform maintenance at the last time step. In our case we chose the value  $K = 10$ .

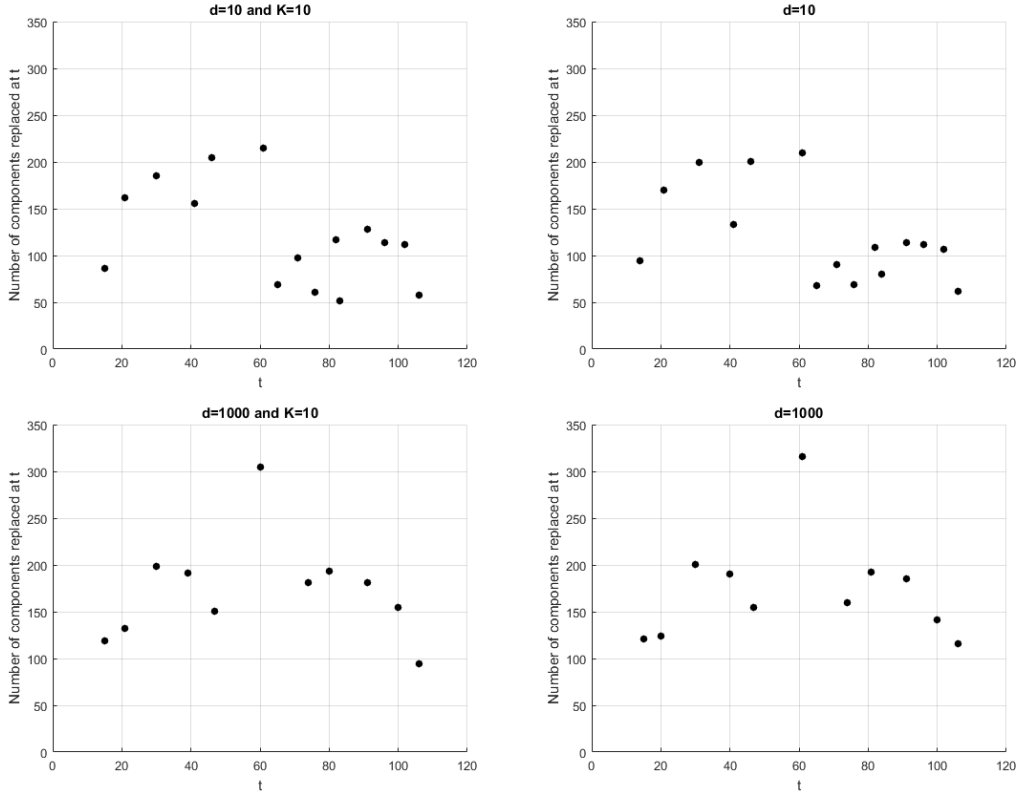


Figure 2: Optimal maintenance schedules for a system of 500 component with the added constraint that no maintenance occasions can be scheduled after some given time-step.

The graph to the upper-left corresponds to the cost  $d = 10$  and  $K = 10$ , which implies that no maintenance is to occur during the last 10 months of the planning horizon. The objective value did not change when adding this constraint and thus remained at 31966. An ocular inspection of these schedules reveal that they are in fact almost identical to the ones produced during the previous test in which this new constraint was not present. There are only marginal differences that are noticeable, such as one or two components being replaced somewhat earlier or later, but the total number of replaced components remains the same. The main problem here is that the optimal schedule does not include any maintenance occasions during the last 14 time steps. The highest  $K$ -value has to be 2 lower than the lowest life length, which would be 13 in our case. By using a high enough  $K$ -value to have an impact, the problem will in our case become infeasible, we therefore conclude that this constraint is unnecessary for our data set.

### 4.1.3 A system of 50 components together with time independent costs

We reduce the number of components from 500 to 50 and perform the tests for  $d = 0, 10, 100, 1000$ .

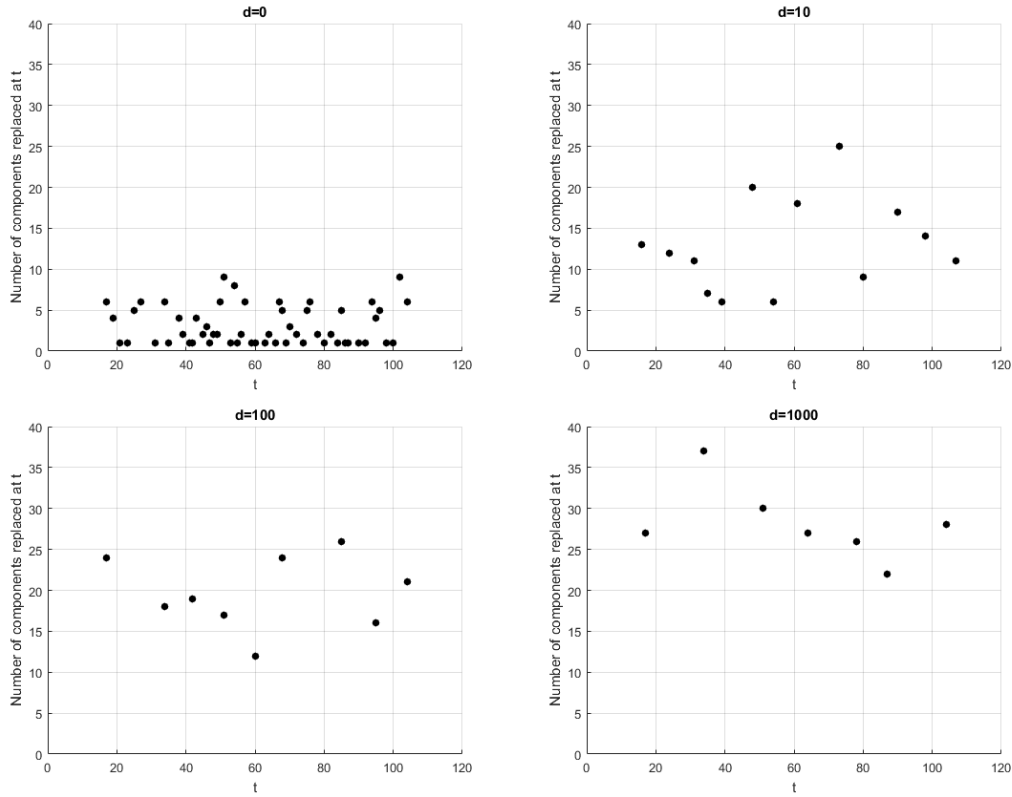


Figure 3: These graphs correspond to a smaller system with only 50 of the 500 components used earlier. The parameters are identical to those used during earlier tests.

The upper left graph represents the system with the value  $d = 0$ . This implementation has an objective value of 3064 and maintenance occurs at 55 occasions. By increasing the set-up cost  $d$  from 0 to 10 the number of maintenance occasions dropped from 55 to 13, this resulted in an increase in objective value from 3064 to 3200. This is because with  $d = 0$ , maintenance was performed at times during which only one or two parts were replaced at each occasion. By increasing  $d$  to 100 the number of maintenance occasions was only 9, four occasions less than the number of occasions corresponding to  $d = 10$  and the objective value increased to 4134. This large increase of the objective value is due to the set-up cost increasing from 130 to 900.

The set-up costs is a considerable part of the objective cost (almost 25% of the cost). We then increase  $d$  to 1000. Now the number of maintenance occasions is only 7 which is the lowest possible. Since the lowest life length of the components is 17, the least possible numbers of replacements of that component is 7. If we replace that same component 6 times it would last for 119 time steps, which is just short of covering the interval. As a consequence of the set-up cost being so violently increased, the objective value increased to 10574, which is an increase by more than doubled compare to the case of  $d = 100$ .

### 4.1.4 A set of 50 components with the added constraint that a set amount of life $r$ is to be left in each component at the end of the contracted period

Here we add a new constraint designed to make sure that each component  $i$  has some specified number of months left of its specific lifetime  $T_i$  at the end of the contracted period, We thus

add the following constraint to the ORP:

$$\sum_{t=T-(T_i-r)+1}^T x_{it} \geq 1$$

where  $r$  is some predetermined minimum life that is required to remain in each component after the end of the contracted period.

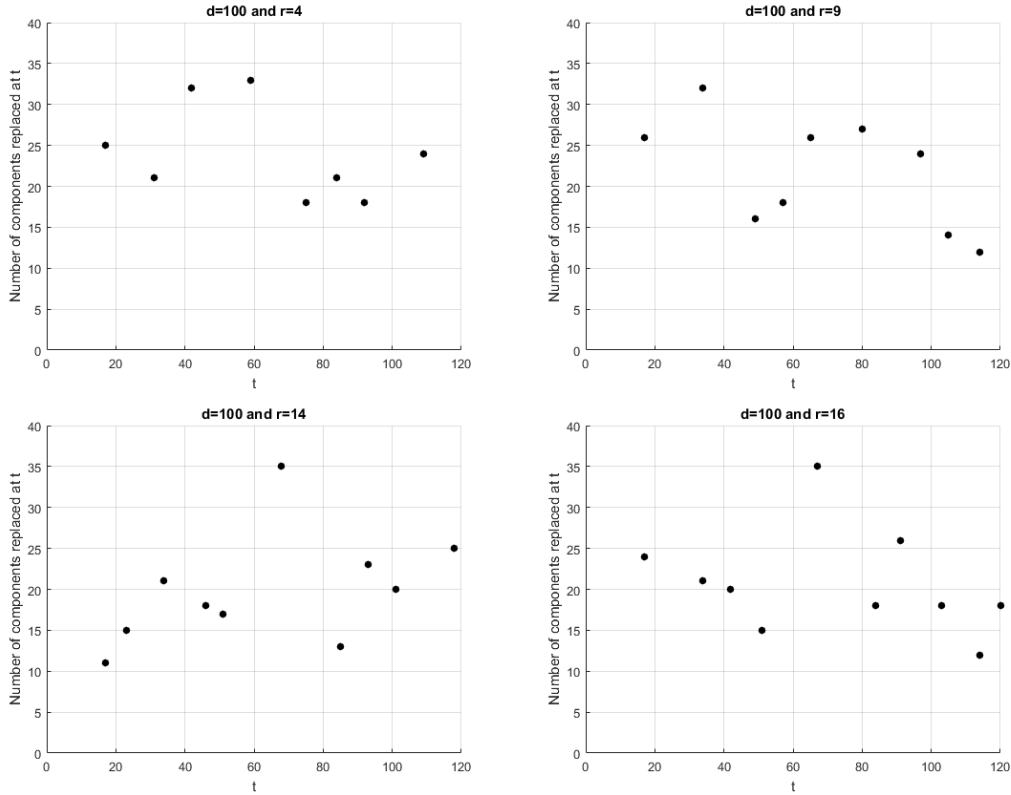


Figure 4: These graphs correspond to the same 50 component system with  $d = 100$  with the new lifetime constraint included.

The upper left graph ( $r = 4$ ) has an objective of 4306 and a total of 8 maintenance occasions. This objective cost is higher than the earlier case without the constraint, but the number of maintenance occasions have been reduced by 1. When the  $r$ -value was increased to 9, the objective value also increased to 4454 and the number of maintenance occasions increased to 9. The  $r$ -value was increased to 14 and the effects were similar. This time the objective value increased to 4665 and the number of maintenance occasions increased by 1. For the last case, the  $r$ -value was increased to 16. This increase is not very high, but it had an impact on the objective value, which increased to 4811. The numbers of maintenance occasions did not get affected by this change. Note that choosing  $r$  16 produces an infeasible problem as the the lowest lifetime among the components in this test was 17.

#### 4.1.5 Time dependent component cost $c_{it}$ and set up cost $d_t$

Earlier all the costs has been time independent with fixed  $d$  and  $c_i$ . Now, both of the costs will be time dependent and monotonously increasing. The set up cost  $d_t$  decrease from 100 to 84 in the case with 10 time steps and from 100 to 71 in the case with 20 time steps. The monotonously decreasing cost replace components  $c_{it}$  varies for the different components, but is never higher than 54 and is never lower than 28 for  $T = 10$  or 22 for  $T = 20$ . In the following test, we will only have 10 components with lower life times that vary between 5 and 9.

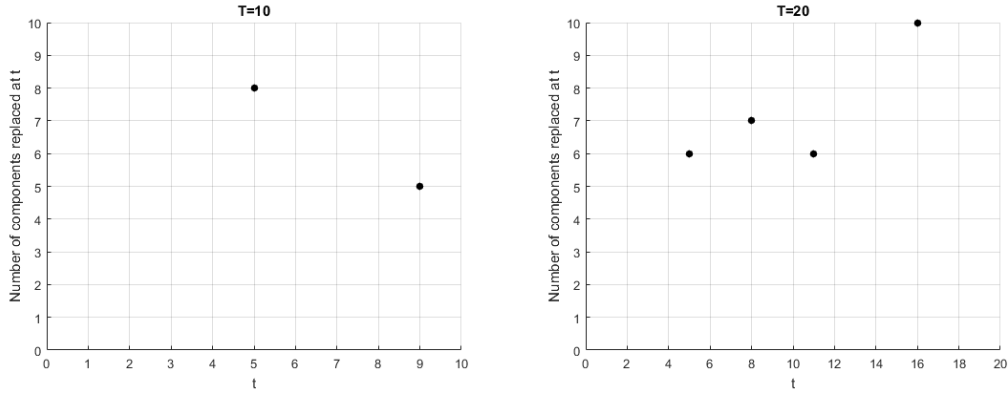


Figure 5: These graphs represent an implementation of the ORP using time dependent costs. The left graph corresponds to a system with  $T = 10$  and the right a system with  $T = 20$ .

In the left graph, the objective is value 696 and maintenance is only performed at 2 occasions. The low number of maintenance occasions has to do with the low number of time steps. The last maintenance occasion is performed at  $T = 9$ , which is different from all the earlier test, where maintenance was rarely done in the late time steps. This has to do with all the monotonous decreasing costs, which make it economically efficient to perform maintenance in the late time steps.

By increasing  $T$  to 20 as in the graph to the right, the number number of maintenance occasions increase to 4 and the objective cost increased to 1375. These changes are not shocking by any means, since the all the costs keeps decreasing all maintenance is performed as late as possible. In the latests maintenance occasion at  $T = 16$  all the parts are replaced at the same time which hasn't been observed so far. This is because there is a low number of components and the costs are at a minimum.

We now add the same extra constraint that was implemented in 4.3:

$$\sum_{i \in \mathcal{N}} \sum_{t=T-K+1}^T x_{it} = 0.$$

The  $K$ -value in this case has to be lower than in the earlier test since the number of time steps and the life length of the components are lower here. As mentioned in the last test with this constraint, the  $K$ -value can at most take the value  $\min\{T_i\} - 1$ . Which in this case equals 5. We will choose the highest  $K$ -value ( $K=4$ ) and the other parameters are identical to those used in section (4.2).

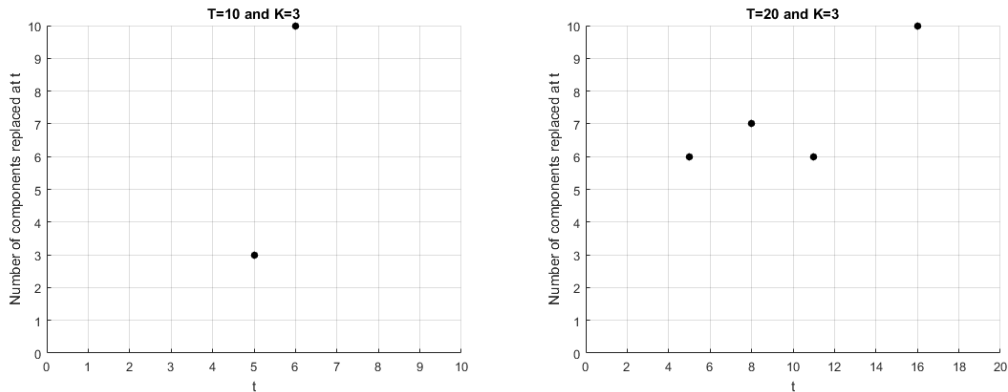


Figure 6: These graphs are the same case as the graphs above, but with the extra constraint added. The graph to the left is different from the same case without the prohibitive constraint, but the right graph is identical.



In the graph to the left the number of maintenance occasions is still 2, but the objective has increased to 717. This is because the maintenance occasions are now scheduled during different time steps. Only 3 components are replaced at  $T = 5$ , while 5 were replaced before. The second maintenance occasion is now scheduled at  $T = 6$  where all 10 components are replaced. The same number of components are replaced in both cases, but are also replaced earlier due to the added constraint, which makes it more expensive. In the right graph it is easy to observe that nothing has happened, even the objective value is precisely the same. This is because no maintenance was done after  $T = 16$  even without the extra constraint.

## 4.2 The PMSPIC: Tests and Results

### 4.2.1 A system of 5 components with time independent costs

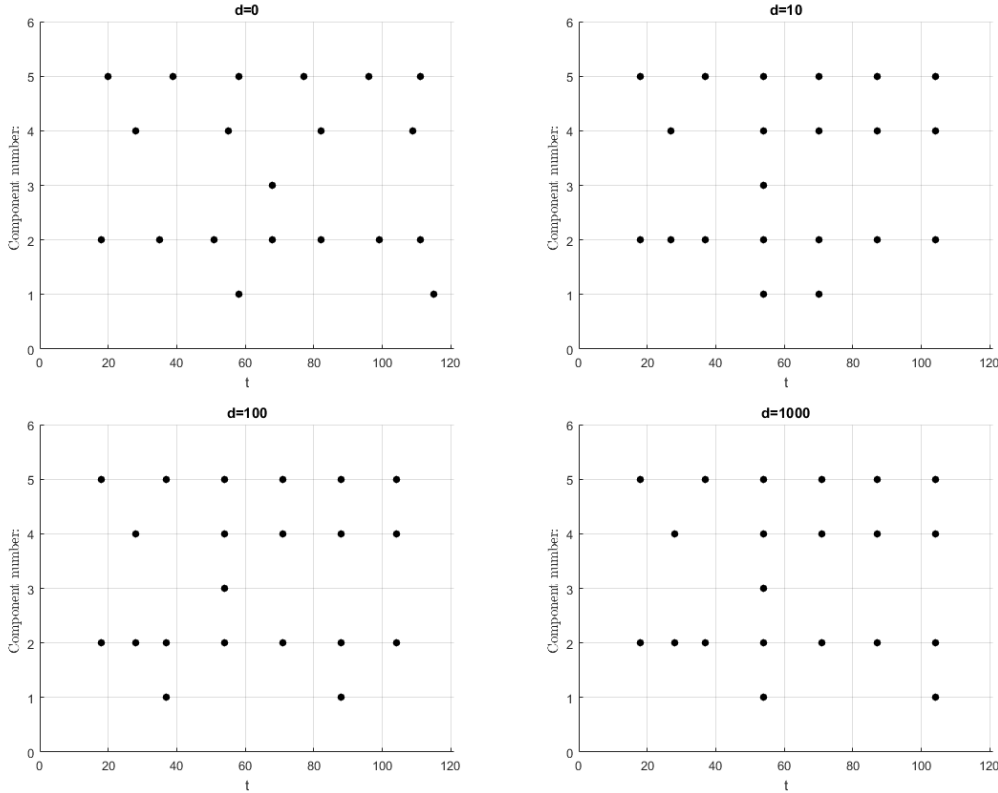


Figure 7: These graphs correspond to an implementation of the PMSPIC using time independent costs

Visible in these graphs is the network flow structure mentioned in section 3.5. The upper leftmost graph depicts a system in which the set-up cost is  $d := 0$ , meaning that the allowed nodes are chosen such that the flow leaves each node at a time near the end of each components specific lifetime  $T_i$ . The optimal value for this case was 320. When the set-up cost increases to 10, an extra occasion is added to the flow of the fourth component and we observe an overall shrinkage of the maintenance intervals for the first, second and fifth component. This new set-up cost lead to an optimal value of 397. The occasions are now also "lined up", reflecting that the flow is now allowed to pass only through a set of specific nodes.

A similar behaviour is reflected in the lower leftmost graph in which  $d := 100$ , except that the maintenance interval for the first component has now been expanded, leading to an optimal value of 1027. The fourth and final graph represents a case where the set-up cost is  $d := 1000$ , in which we observe that the maintenance interval for the first component has been moved again while the rest of the occasions remain the same as in the case for  $d := 100$ . This change resulted in an optimal value of 7327.

## 5 Discussion

### 5.1 A discussion of the time independent cases

In the first case with 500 components and 120 time steps, we see a decrease in the numbers of maintenance occasions in two of the cases when we increase the set-up cost  $d$ . When the set-up cost increases from 0 to 10 and 100 to 1000, respectively, the number of maintenance occasions drop from 71 to 15 and 15 to 11. These results were expected since the higher set-up cost makes it economically beneficial to schedule fewer occasions. The impact of the set-up cost increase from 10 to 100 was on the other hand not expected. The number of maintenance occasions is the same in both cases and maintenance is performed at almost exactly the same time steps. The objective value is increased by 1350, an increase that is owed solely to the increase in set-up cost. The cost to replace components stays the same. The explanation for this similar results is that the optimal solutions to both cases are not unique and are thus degenerate.

One of the reasons this happens is because the set of components  $N$  is extremely large which may nullify the impact of the set-up cost. It only amounts to about 5% of the objective value for all the cases except for the one with  $d = 1000$ , where it is about 20%. In other words, the ration  $\frac{d_t}{c_{it}}$  is small enough to be neglected for  $d_t = 0, 10, 100$ . Another possible explanation for this is that the data, i.e, the replacement costs and component lifetimes, are unfortunately randomly generated using MatLabs rand function meaning that while the numbers are generated in such a way that a longer lifetime corresponds to a higher replacement cost and vice versa, they are still clustered around some expected value. If the tests were performed using data gathered from a real life situation then maybe the number of components and replacement costs would not affect the solutions in the same way.

In the case with the new constraint the objective values and the number of maintenance occasions stay the same. The only difference in the graphs is that some occasions have been moved a single time-step ahead or back, respectively, and the number of components replaced during some occasions differs only slightly. The explanation for this is that even without the new constraint, no maintenance was performed during the last 14 time-step, meaning that the constraint is unnecessary. These results were not shocking by any means but was, in fact, expected. It is unfortunate that the extra constraint did not result in any meaningful difference; but it does, however, tell us that the model behaves as it should when new constraints that do not affect the feasible set are added. If our data was taken from real world events, then this would probably not be the case, since the lifespans would become less varied.

In the case with 50 components, the behaviour of the graphs is similar to the case with 500 components. The difference is that the set-up cost has a higher impact. The number of maintenance occasions in this case are 55, 13, 9, 7 for  $d = 0, 10, 100, 1000$ , respectively, which is lower compared to 71, 15, 15, 11 in the 500 case. When  $d$  increase from 100 to 1000 the value of the objective increases from 4134 to 10574 which is an increase of 150%! If we compare that to the case with 500 components, where it only increased by 25%, it is safe to say that the impact of the set-up cost is in part dependent on the number of components in the system. This is by no means an unexpected result, however. If for example, the set-up cost is the same as replacing 30 components, then it is crucial to avoid that cost for a system with 50 components. For a system with 500 components, that cost is of no real concern if  $d$  is small, becoming more of a pain as  $d$  increases. In the case with  $d = 1000$  only 7 maintenance occasions were scheduled, which is the lowest possible since the component with the shortest life time would not last for 120 time steps if it were to be replaced only 6 times. This indicates the importance of avoiding a high set-up costs for a system with a small number of components. This also indicates that it may be more profitable in the long run to spend more on product development when it comes to systems for which the set-up cost is large. This would of course lead to a larger initial investment but also, in turn, result in a decrease in the maintenance cost. When the new constraint with the  $r$  value was added to the case with  $d = 100$ , it had a great impact. None of the graphs looks similar, not even when the difference of the  $r$ -value is only 2. The constraint definitely has a great impact on the objective, thus the value increases from 4134 to 4811, when  $r$  increases from 0 to 16. The increase is somewhat linear. The number of maintenance occasions first decrease by 1 and

then increase by 2, which is 10 for  $r = 14$  and  $r = 16$ . All of this is of course expected, since the constraint only benefits the buyer of the maintenance and makes it both harder and more expensive to perform it.

## 5.2 The cases with time dependent monotonous costs

In the cases with time dependent monotonously decreasing costs, maintenance is scheduled at 2 occasions for  $T = 10$ , and 4 occasions for  $T = 20$ . Maintenance is performed as late as possible, which reflects the fact that the cost to replace a component now decreases with time. All components are replaced during the very last scheduled occasions for the case of  $T = 20$ , another effect of the decreasing cost that had not been observed before but can reasonably be inferred from the data. Our findings now point towards that it is economically beneficial to perform maintenance as late as possible. Since the set-up cost is high compared to the costs of replacement, a large number of components are replaced at each occasion. An increase in the number of time steps would perhaps lead to the inverse of this phenomenon as both the set-up cost and replacement cost would tend to zero in the theoretical limit, leading to a schedule where components are replaced whenever the need arises. The data we used here is deterministic and independent of the data used in the time independent cases and is thus not representative of real life events.

In the same cases with the extra constraint, we can observe the impact of the prohibitive constraint, i.e, that we do not allow any maintenance to be performed after some predetermined time step. In the case with 10 time steps, the last maintenance occasion occurs at  $t = 6$  instead of  $t = 9$ . The number replaced components at the maintenance occasions are also different from before, but the total number of replaced components is still 13. This result was definitely expected since maintenance was done at  $t = 9$  before we added the constraint, which means that the occasion has to be moved to an earlier point in time so as to not violate the constraint. The solution in the case with the  $K$ -value does also have a higher objective value, since the new constraint forces a maintenance occasion to be scheduled at an earlier point in time, meaning that it will also be more expensive due to the fact that costs are now decreasing with time. Not being able to use these 3 last occasions, which are also the cheapest to perform maintenance on, can of course make the objective value higher. The case with  $T = 20$  is unfortunately the same, with or without the new constraint. This is for the same reasons mentioned earlier for the 500 component cases.

## 5.3 The case with the PMSPIC

The graphs and results in the test with the PMSPIC can unfortunately not be compared in any meaningful way to the tests done with the ORP. This is because the model that was first used to produce the results was incorrect, which was something that was found comparatively late in the course of the project. Since the PMSPIC is more complex than the ORP due to the inclusion of extra variables the solution time is also much longer for systems of many components, as a consequence. The data output from AMPL was also a lot more difficult to handle for the PMSPIC. That is why we have chosen to include a system with only 5 components in the test, a system which illustrates the network interpretation of the PMSPIC well.

## 5.4 Comparison of computation times

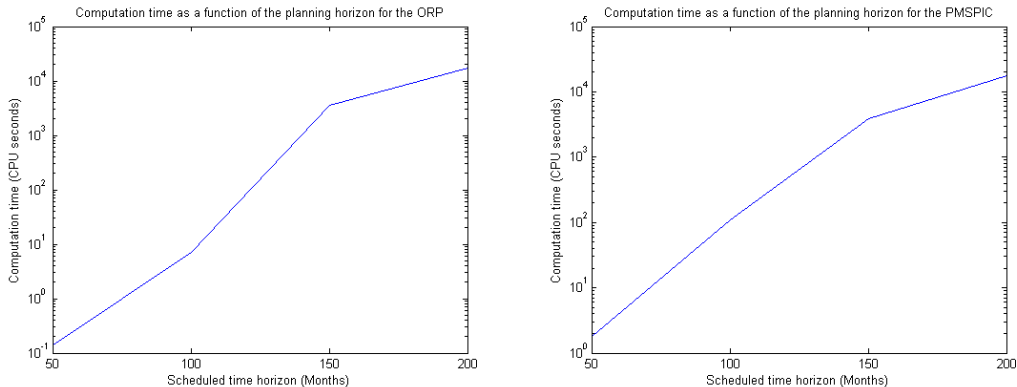


Figure 8: The logarithm of the computation times for both models as a function of the planning horizon  $T$ , where  $c_{it} = c_i$  and  $d_t = d$ .

The above graphs represent a comparison between the computation time for both the ORP and the PMSPIC. The computation time of the ORP, corresponding to the leftmost graph, increases almost linearly with time and we thus have a relation of the form  $\ln(y(T)) = kT + m$ , i.e.  $y(T) = Ce^{kT}$ , where  $C$  and  $k$  are non-negative constants. The proof given on the complexity of the ORP in section 3.2 is only applicable when all costs are time dependent. It is thus not possible to draw any general conclusions as to the complexity of the time independent ORP. It is, however, reasonable to conclude that it at least tends towards being non-polynomial in time.

As for the PMSPIC, illustrated in the rightmost graph, the situation seems to be similar. The PMSPIC seems to increase at a more steady pace, however. Note that these graphs were produced using only four time steps between 50 and 200 and are intended to be used for illustrative purposes only. There are still several time steps for which the computation time may vary quite heavily.

## 5.5 Conclusion and thoughts on future experiments

The main conclusion we can make is that the ratio  $\frac{d_t}{c_{it}}$  is what decides the impact of the set-up cost. The higher the ratio, the higher the impact and vice-versa. In a situation where time dependent costs are applicable this result is not always valid, however. The way in which this time-dependence manifests also carries a large influence over the optimal scheduling. In the case presented in section 4.1.5, the time-dependence stems from costs being monotonously decreasing. This specific choice is motivated by the fact that the ORP can be solved using a greedy algorithm when  $c_{i,t+1} \leq c_{it}$  and  $d_{t+1} \leq d_t$ , and also by the fact that the time dependent OPR is NP-hard as proven in section 3.2. This results in a problem that is NP-hard but also easier to solve due to the nature of the greedy algorithm.

When it comes to future experiments and additions, it would be interesting to analyse the two models and establish which is the absolute "best". It would also be interesting to re-run all tests once again with data gathered from real life events and analyse the difference in the behaviour of the models and results. Another thing could be to put more time into investigating the complexity class of the time independent version of both the ORP and the PMSPIC. The final goal would then be to produce a proof of the NP-hardness of both problems.

## 6 Appendix

### 6.1 The Traveling Salesman Problem

A salesman is required to visit  $n$  cities and must also return to the city from which he started. Travelling from city  $i$  to city  $j$  incurs a cost  $c_{ij}$ , where  $1 \leq i, j \leq n, i \neq j$ . A solution to this problem is referred to as a tour. The question is then: in which order should the salesman visit the cities to minimize the total cost of a tour? This is the definition of the Travelling Salesman Problem (TSP). The following formulation is presented as the one in [2, p. 62], known as the Dantzig-Fulkerson-Johnson formulation of the asymmetric TSP. Note that this formulation is valid only for the case of asymmetric travelling costs, i.e,  $c_{ij} \neq c_{ji}$ . Further formulations for both symmetric and asymmetric costs can be found in the same book.

Let  $x_{ij} \in \{0, 1\}$ , where  $x_{ij} = 1$  if city  $j$  is visited directly after city  $i$ , and 0 otherwise. Assume that we have a directed graph  $D = (V, A)$  and costs  $c_a, a \in A$ . Given a set of cities  $S \subseteq V$  we let  $\delta^+(S) := \{ij \in A \mid i \in S, j \notin S\}$  and  $\delta^-(S) := \{ij \in A \mid i \notin S, j \in S\}$  and define the Dantzig-Fulkerson-Johnson formulation of the asymmetric TSP as the following:

$$\text{minimize } \sum_{a \in A} x_a c_a \tag{1}$$

$$\text{subject to } \sum_{a \in \delta^+(i)} x_a = 1, \quad i \in V \tag{2}$$

$$\sum_{a \in \delta^-(i)} x_a = 1, \quad i \in V \tag{3}$$

$$\sum_{a \in \delta^+(S)} x_a \geq 1, \quad \emptyset \subset S \subset V \tag{4}$$

$$x_a \in \{0, 1\}, \quad a \in A \tag{5}$$

Constraints (2) and (3) ensures that each city is visited exactly once during a single tour. (4) makes sure that the solution is not partitioned into several subtours, i.e, that the tour only consists of a single closed sequence of cities and not several disjoint closed sequences. This is what is called a subtour elimination constraint.

### 6.2 The Set Covering Problem

Given a collection of elements  $N = \{1, \dots, n\}$ , as well as a collection of subsets  $S$  of  $N$ , we define a cover of  $N$  as a subcollection  $C \subseteq S$  such that the union of  $S$  is equal to  $N$ . The goal of the Set Covering Problem is to find a cover that uses the fewest amount of sets. Letting  $x_i$  be the  $i$ :th element of  $S$ , we define the set covering optimization problem as the following:

$$\text{minimize } \sum_{i \in S} x_i \tag{1}$$

$$\text{subject to } \sum_{i: j \in S} x_i \geq 1, \quad \forall j \in N \tag{2}$$

$$x_i \in \{0, 1\}, \quad i \in S \tag{3}$$

(2) ensures that every element of  $N$  is covered and (3) ensures that an element either lies in the set cover or does not lie in the set cover.

## References

- [1] Garey, Michael R., and David S. Johnson., 'Computers and Intractability: A Guide to the Theory of NP-Completeness', Anonymous Translator( San Francisco, Freeman, 1979).
- [2] Conforti, Michele, Gérard Cornuéjols, Giacomo Zambelli, et al. , 'Integer Programming', Anonymous Translator, Volume 271(2014th edn, Cham, Springer International Publishing, 2014).
- [3] Almgren, Torgny, Niclas Andréasson, Michael Patriksson, et al. 'The Opportunistic Replacement Problem: Theoretical Analyses and Numerical Tests', *Mathematical Methods of Operations Research*, vol. 76/no. 3, (2012), pp. 289-319.
- [4] Bangalore, Pramod, Chalmers University of Technology, Chalmers tekniska högskola, et al. 'Load and Risk Based Maintenance Management of Wind Turbines', vol. 4132/(2016).
- [5] Gustavsson, Emil, Michael Patriksson, Ann-Brith Strömberg, et al. 'Preventive Maintenance Scheduling of Multi-Component Systems with Interval Costs', (2014),
- [6] Ben-Daya, Mohammed, Uday Kumar, D. N. P. Murthy, et al. , 'Introduction to Maintenance Engineering: Modelling, Optimization and Management', Anonymous Translator(1st edn, Hoboken, John Wiley & Sons, Inc, 2016)