The facial structure of and efficient solution methods for the opportunistic replacement problem

Magnus Önnheim

March 10, 2010

Abstract

We study a basic mathematical model for opportunistic maintenance planning. Two new classes of facets are derived, based on combinatorial implications and zero-half Chvátal-Gomory cuts, and it is indicated how these generalize to extended replacement models. Furthermore a dynamic programming scheme for solving the model is presented and shown to in some cases have superior performance to plain vanilla branch-and-cut solver.

Sammanfattning

Vi studerar en enkel matematisk modell för opportunistisk underhållsplanering. Två nya klasser av fasetter presenteras, baserade på kombinatoriska implikationer samt speciella Chvátal-Gomory olikheter. Vidare indikeras hur dessa fasetter kan generaliseras till utvidgade modeller. En dynamisk programmeringsalgoritm ges även för att lösa den givna modellen, och demonstreras i vissa fall ha överlägsen prestanda jämfört med en standard branch-and-cut-lösare.

Acknowledgments

I would like to thank my supervisor and examiner professor Michael Patriksson for his role in the making of this thesis. Furthermore, the rest of the optimization group at Chalmers should not go without mention, and I would in particular like to thank Ann-Brith Strömberg and Adam Wojchiechowski.

Contents

Introduction

The problem of planning maintenance is a continuously growing subject. As our society grows ever more technologically advanced, with more and more tasks being done by machines, there are simply many components that can fail. As a natural consequence, the costs of performing maintenance is growing, and will likely continue to do so for many years to come. Therefore, the importance of performing maintenance operations well seems hard to overestimate. It is estimated in a recent study (by Forum Vision Instandhaltung, Germany) that in the European Union maintenance costs in the manufacturing industry alone is as high as $2 trillion annually, while another source [RKM04] states that in the US, roughly one third of the cost of maintenance is wasted by inefficient management methods. This could mean that maintenance in the EU amounts to roughly twice the GDP of Italy, and the GDP of the Netherlands is wasted in inefficiency.

The scope of this thesis is to study the mathematical aspects of the concept of opportunistic maintenance planning. The idea is to use some mathematical model to decide whether or not to perform maintenance that is not strictly neccessary when an opportunity arises. That is, we anticipe a future need for maintenance, and ask the question whether or not the cost of replacing a part that is not yet broken outweighs the cost of having to perform maintenance again later. An example would be to replace all the tires of a car when it gets a flat, because we interpret the flat as a sign that all the tires are worn out.

This thesis is divided into five main chapters. Chapter 1 discusses some brief theoretical results, and serves mainly to fix notation and references for later chapters. For a more thorough introduction to combinatoral optimization we refer the reader to [NW88]. Chapters 2 through 4 contains the main body of this thesis, in that they discuss the facial structure of a maintenance polytope, and efficient algorithms for solving the replacement problem. Numerical results are then provided in chapter 5.

Chapter 1

Basic theory of combinatorial optimization

This chapter serves to collect some basic results of combinatorial optimization as well as to fix notation. Much of this text follows the treatment on the subject in [NW88].

1.1 Basic premises

By a combinatorial problem we shall understand a problem of the form

$$\begin{aligned} &\text{minimize } \mathbf{c^T x}, && \text{(1.1a)}\\ &\text{subject to } \quad A\mathbf{x} \leq \mathbf{b}, && \text{(1.1b)}\\ &\qquad\qquad \mathbf{x} \in \mathbf{X}, && \text{(1.1c)} \end{aligned}$$

where $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n,\ A \in \mathbb{R}^{m \times n},\ \mathbf{b} \in \mathbb{R}^m$, and $\mathbf{X}$ is some finite set. We will throughout assume that $\mathbf{X} = \{0,1\}^n$. Let us denote the feasible set of (1.1) by S. It is a well-known fact from linear programming theory that if an optimum to a linear programming problem exists, it can be taken as an extreme point of the feasible polyhedron. Furthermore, the representation theorem for convex polyhedra establishes that any extreme point of the convex hull of S is feasible in S, and that $\mathrm{conv}(S)$ is a bounded polytope. Thus, we can if we so wish in (1.1) replace S with $\mathrm{conv}(S)$, transforming the problem into a linear programming one. Let us write this problem as that to

$$\begin{aligned} &\text{minimize } \mathbf{c^T x}, && \text{(1.2a)}\\ &\text{subject to } A'\mathbf{x} \leq \mathbf{b}', && \text{(1.2b)} \end{aligned}$$

where $A',\ b'$ are such that $\mathrm{conv}(S) = \{\mathbf{x} \in R^n | A'\mathbf{x} \leq b'\}$. An approch to solving (1.1) could therefore be to solve (1.2). However, this would require a complete description of the matrix A', which in practice is generally undoable. Instead one is often

forced to work with approximations of $\mathrm{conv}(S)$. The canonical approximation would be to in (1.2) replace A' by A from (1.1). A tighter approximation may be obtained if one is able to somehow find some subset of rows of the matrix A'. Hopefully, as the approximation of $\mathrm{conv}(S)$ gets tighter, the solutions to the *relaxed* version of (1.2) can be used to guide us towards a solution of (1.1).

1.2 A brief primer of polyhedral theory

In this section we list some useful theorems for use in the rest of this thesis. We begin by defining the notion of a *facet* of a polyhedron P. To do this we first need a concept of faces and their dimensions.

Definition 1 (Face)**.** A face of a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ is a set $F \subseteq P$ of the form $F = \{\mathbf{x} \in P | \pi^T\mathbf{x} = \pi_0\}$, where (π, π_0) is such that $\pi^T\mathbf{x} \leq \pi_0$ for all $\mathbf{x} \in P$. We say that (π, π_0) *represents* the face F. It is called *proper* if $F \neq \emptyset$ and $F \neq P$.

Hence, a face is the intersection of P and a hyperplane described by a valid inequality for P. From this definition it can be seen that a face of a polyhedron is itself a polyhedron, and we can define a common notion of dimension.

Definition 2 (Polyhedral dimension)**.** A polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ is said to be of dimension k if the maximum number of affinely independent points of P is $k + 1$. The codimension of F is defined as $\mathrm{codim}(F) = n - \dim(F)$.

As the maximum number of affinely independent points of $\mathbb{R}^n$ is $n + 1$, it follows that no polyhedron in $\mathbb{R}^n$ can have dimension greater than n. If the dimension is exactly n we call the polyhedron *fulldimensional*. In the context of combinatorial optimization, fulldimensionality can be understood as there being no redundancies in the modelling. That is, it is not possible to immediately conclude from the constraints that some variable must be set to a fixed value. It can thus be interpreted as there being no hidden equality constraints in the matrix A of (1.1). It also follows from this definition that polyhedral dimension is the same as the vector space dimension of the tangent vectors of F.

We are now ready to define a facet of P.

Definition 3 (Facet)**.** A face of a polyhedron P of dimension k is called a facet if it is of dimension $k - 1$.

A facet represented by (π, π_0) is thus the tightest possible valid inequality for P, in the sense that it cannot be translated nor tilted into another valid inequality that is not implied by $\mathbf{x} \in P,\ \pi^T\mathbf{x} \leq \pi_0$. Hence, when searching for rows of the matrix A' in (1.2) it is desirable to find facets of $\mathrm{conv}\, S$. To be able check whether rows define facets, we need a less abstract characterization than is given in Definition 3, which the following theorem gives us.

Theorem 4 (characterization of facets)**.** *Let P be a fulldimensional polyhedron in $\mathbb{R}^n$ and let $F = \{\mathbf{x} \in P | \pi^T\mathbf{x} = \pi_0\}$ be a proper face of P. The following statements are equivalent:*

1. *F is a facet of P,*
2. *If $\lambda^T x = \lambda_0$ for all $\mathbf{x} \in F$, then $(\lambda, \lambda_0) = \alpha(\pi, \pi_0)$ for some $\alpha \in \mathbb{R}$,*
3. $\operatorname{codim} F = 1$.

Furthermore, the equivalence between 2 and 3 remains true even if the assumption on the fulldimensionality of P is dropped.

Proof. The equivalence between 1 and 2 is established in [NW88]. The equivalence between 1 and 3 follows immediately from the definition of a facet of a fulldimensional polyhedron, which completes the proof when P is fulldimensional. Let us now drop the assumption on P being fulldimensional, and show that $2 \Leftrightarrow 3$. Let $\mathbf{x}^0 \in F$. Then it holds that

$$\lambda^T(\mathbf{x} - \mathbf{x}^0) = 0, \quad \forall \mathbf{x} \in F \tag{1.3}$$

and hence also that

$$\lambda^T \mathbf{y} = 0, \quad \forall \mathbf{y} \in \operatorname{Span}(F - \mathbf{x}^0). \tag{1.4}$$

In other words, $\lambda \in \operatorname{Span}(F - \mathbf{x}^0)^\perp$. From basic linear algebra we now obtain

$$\dim(\lambda | \lambda^T x = \lambda^0, \forall \mathbf{x} \in F) + \dim(F) = \dim(\mathbb{R}^n), \tag{1.5}$$

from which the equivalence of 2 and 3 immediately follows. □

That is, to show that a valid inequality for P yields a facet, we check that the vector space dimension of the orthogonal complement of the tangent vectors of F has a low enough dimension.

1.3 Integral polyhedra

A polyhedron P is said to be *integral* if all of its extreme points are integral, and a matrix A such that any submatrix of A has determinant $-1, 0$ or 1 is called *totally unimodular* (TU). We have the following theorem, which can be viewed as a consequence of Cramer's rule.

Theorem 5. *If A is TU, then the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is integral for all $b \in \mathbb{Z}^n$.*

This theorem can be very useful, since if we know that a matrix in (1.1) is TU, we can immediately replace the condition that $x \in \{0, 1\}^n$ with $x \in [0, 1]^n$. Furthermore, some ways exist in which a matrix can checked for TU, which in many cases are based on network properties on the matrix. A treatment of this theory lies beyond the scope of this thesis and we refer the interested reader to [NW88].

1.4 Chvátal-Gomory rounding

Generating rows of A' in (1.2) is not a trivial problem. However some general methods to do this exist, and Chvátal-Gomory Rounding is such an approach that will prove to be fruitful in Chapter 3. We assume without loss of generality that in (1.1), $A \in \mathbb{Z}^{m \times n}$ and $\mathbf{b} \in \mathbb{Z}^m$. Then we may obtain a new valid inequality for S[1] by taking a non-negative linear combination of the rows of A.

Let $u_i,\ i \in \{1, \ldots, m\}$ be non-negative real numbers. Then

$$\sum_{j=1}^{n} \left(\sum_{i=1}^{m} u_i a_{ij} \right) x_j \leq \sum_{i=1}^{m} u_i b_i \tag{1.6}$$

is a valid inequality for S. We may round the left hand side down yielding

$$\sum_{j=1}^{n} \left\lfloor \sum_{i=1}^{m} u_i a_{ij} \right\rfloor x_j \leq \sum_{i=1}^{m} u_i b_i \tag{1.7}$$

as another valid inequality for S. Now the left hand side is an integer, as $x_j \in \{0, 1\}$, and hence we may round the right hand side down as well, yielding

$$\sum_{j=1}^{n} \left\lfloor \sum_{i=1}^{m} u_i a_{ij} \right\rfloor x_j \leq \left\lfloor \sum_{i=1}^{m} u_i b_i \right\rfloor \tag{1.8}$$

as a new valid inequality for S. The procedure above is called *Chvátal-Gomory Rounding* and the obtained inequality is called a Chvátal-Gomory (CG) inequality.

The procedure may of course be applied recursively, also using previously obtained Chvátal-Gomory inequalities as the basis for new CG inequalities. The class of CG inequalities can then be defined as any inequality obtained after a finite number of applications of the CG procedure, as well as any valid inequality that is implied by the CG inequalities. These implied inequalities are then said to be dominated by the CG inequalities.

Given the inequalities of (1.1), it follows that the inequalities (1.6) and (1.7) are redundant in the description of $\mathrm{conv}(S)$. The natural question arises whether (1.8) can be nonredundant. The following theorem establishes this and more.

Theorem 6 (Theorem 2.8 [NW88])**.** *Let $\pi^T x \leq \pi_0$ with $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ be a valid inequality for $S = P \cap \mathbb{Z}^n$ with $P = \{\mathbf{x} \in \mathbb{R}^n_+ | A\mathbf{x} \leq b, \mathbf{x} \leq 1\}$. Then $\pi^T \mathbf{x} \leq \pi_0$ is dominated by a Chvátal-Gomory inequality for S.*

This theorem tells us that, in principle, to solve any combinatorial problem, all we need to do is to find all the Chvátal-Gomory inequalities and solve the linear programming problem (1.2). Another theorem in [NW88] tells us that this generation of CG inequalities can be done in a finite, but likely intractable, number of steps. Furthermore, the linear programming problem (1.2) may also become too large to handle.

[1] That is, a row of A' in (1.2).

1.5 Automated facet generation

When hunting for facets of a polytope, it is very helpful to have a computer program capable of generating all facets for some small instance of the problem at hand. One way of doing this is complete enumeration coupled with Fourier-Motzkin elimination.

Given a combinatorial optimization problem, it is in principle easy to enumerate the feasible set, as it is finite. Suppose we have such an enumeration: $\{v_1, v_2, \ldots, v_N\}$. Then we may describe the convex hull of those points as the set of $\mathbf{x} \in [0,1]^n$ satisfying

$$\mathbf{x} = \sum_{i=1}^{N} \lambda_i v_i, \tag{1.9a}$$

$$1 = \sum_{i=1}^{N} \lambda_i, \tag{1.9b}$$

$$0 \leq \lambda_i, \quad i = 1, \ldots, N \tag{1.9c}$$

for some $\lambda_i,\ i = 1, \ldots, N$. The idea is to project out the variables λ_i, to yield a system of equations and inequalities containing only $\mathbf{x}$. This can be done by means of Fourier-Motzkin elimination. To project out λ_1, we write the system (1.9) as

$$\lambda_1 \geq A(\mathbf{x}, \lambda_j), \quad j \geq 2 \tag{1.10a}$$

$$\lambda_1 \leq B(\mathbf{x}, \lambda_j), \quad j \geq 2, \tag{1.10b}$$

where we can interpret the matrices A and B as setting variable dependent upper and lower bounds on the variable λ_1, respectively. It then follows that given $(\mathbf{x}, \lambda_i),\ i \in \{2, \ldots, N\}$, $(\mathbf{x}, \lambda)$ is feasible in (1.9) if and only if

$$\max A(\mathbf{x}, \lambda_j) \leq \lambda_1 \leq \min B(\mathbf{x}, \lambda_j). \tag{1.11}$$

Thus, there exists some λ_1 such that $(\mathbf{x}, \lambda)$ is feasible in (1.9) if and only if

$$\max A(\mathbf{x}, \lambda_j) \leq \min B(\mathbf{x}, \lambda_j). \tag{1.12}$$

Hence (1.12) is a system not containing λ_1 such that the solution set is equal to the solution set of (1.9) projected on $\lambda_1 = 0$. By repeating this we may project out all the variables $\lambda_i,\ i \in \{1, \ldots, N\}$ and obtain a facial representation of the polytope in question.

Unfortunately, this projection can be a very slow algorithm. Let n_A, n_B be the number of rows of A and B, respectively. Then the modelling of (1.12) with linear inequalities requires in the worst case scenario $n_A n_B$ rows. Hence the number of inequalities may grow very large, and using redundancy checks to remove inequalities is very important. Even so, this procedure is not applicable but for the smallest of problems. However, studies of the facets of small instances of a class of problems may lead to insights into more general structures that apply to all instances of that class.

Chapter 2

Basic replacement model

An instance of the Basic Replacement Problem (BRP) is defined as follows. Consider a set $\mathcal{N}$ of components, $N := |\mathcal{N}|$, and a set $\mathcal{T} = \{1, \ldots, T\}$, $T \in \mathbb{N}$ being the planning horizon. A component $i \in \mathcal{N}$ is assigned a life limit of T_i timesteps, and a cost of replacement c_{it}. There is also a fixed cost d_t associated with performing maintenance for any component at time t. The objective is to minimize the cost of a maintenance schedule, keeping the system running without failure during the planning period. This is encoded in a problem with the variables:

$$z_t = \begin{cases} 1, & \text{maintenance is performed at time } t, \\ 0, & \text{otherwise,} \end{cases} \qquad t \in \mathcal{T},$$

$$x_{it} = \begin{cases} 1, & \text{part } i \text{ is replaced at time } t, \\ 0, & \text{otherwise,} \end{cases} \qquad i \in \mathcal{N},\ t \in \mathcal{T},$$

yielding the BRP as

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} c_{it} x_{it} + \sum_{t \in \mathcal{T}} d_t z_t, , \tag{2.1a}$$

$$\text{subject to} \quad \sum_{t=l+1}^{l+T_i} x_{it} \geq 1, \qquad l = 0, \ldots, T - T_i,\ i \in \mathcal{N}, \tag{2.1b}$$

$$x_{it} \leq z_t, \qquad t \in \mathcal{T},\ i \in \mathcal{N}, \tag{2.1c}$$

$$x_{it}, z_t \in \{0, 1\}, \qquad t \in \mathcal{T}. \tag{2.1d}$$

The constraints are straightforward to understand. Constraint (2.1b) states that for all components and any window of time the size of the life of that component, it has to be replaced at least once. The constraint (2.1c) ensures that if we repair some component at a time t, a maintenance occasion has occured, and the maintenance cost d_t must be paid. Let us henceforth denote the set of feasible points of (2.1) by S. The convex hull of S will be called the replacement polytope.

This model for the opportunistic replacement problem was first proposed by Dickman et al [DEW91] in 1991, and seems to have been largely forgotten since. However, in a recent case study [APS08] and a survey [PSW09], results were demonstrated that indicate that this model and variations thereof can have superior performance compared to simpler policies as can be found in for example [RKM04].

2.1 Properties of the replacement polytope

We begin this section by collecting some known properties of the replacement polytope that were proven in a recent article [PSW09]. The first property is that when the maintenance occasions are fixed, that is, for a subset $\mathcal{T}' \subset \mathcal{T}$, we let $z_t = 1,\ t \in \mathcal{T}',\ z_t = 0,\ t \in \mathcal{T}\backslash\mathcal{T}'$, the result is an integrality property.

Proposition 7. *The polyhedron defined by* (2.1b) *and*

$$x_{it} \leq 1, \qquad t \in \mathcal{T}', \tag{2.2a}$$
$$x_{it} \leq 0, \qquad t \in \mathcal{T}\backslash\mathcal{T}', \tag{2.2b}$$

for $i \in \mathcal{N}$ is TU.

This property allows us to relax the integrality requirements on the variables x_{it}. Furthermore, this shows that if $T_i = 1$ for some $i \in \mathcal{N}$, forcing $z_t = 1,\ t \in \mathcal{T}$, then BRP is very easy to solve. Hence, from now on we will assume that $T_i \geq 2,\ i \in \mathcal{N}$.

Finally, the following two propositions were also proved in the article [PSW09].

Proposition 8. *If $T_i \geq 2$, $i \in \mathcal{N}$, the replacement polytope is fulldimensional.*

Proposition 9. *If $T_i \geq 2,\ i \in \mathcal{N}$, the inequalities* (2.1b)-(2.1c) *define facets of the replacement polytope. The lower and upper bounds on x_{it}, z_t, respectively, define facets if $T_i \geq 3,\ i \in \mathcal{N}$.*

2.2 Properties of optimal solutions

In [PSW09], the following proposition was shown.

Proposition 10. *If $c_{it}, d_t,\ t \in \mathcal{T}$, are non-increasing in t, there exists an optimal solution such that if $z_t = 1$, then $t = \sum_i k_i T_i$ for some $k_i \in \mathbb{N}$.*

This proposition shows that some preprocessing variable reduction is possible. We elaborate on this result, showing that a more general construction is possible.

Proposition 11 (No maintenance until failure, 1)**.** *If $c_{it}, d_t,\ t \in \mathcal{T}$, are non-increasing in t, then there exists an optimal solution that satisfies*

$$z_t \leq \sum_{i \in \mathcal{N}} z_{t-T_i} \tag{2.3}$$

where we have defined auxiliary parameters

$$z_t = \begin{cases} 0, & t \leq -1, \\ 1, & t = 0. \end{cases} \tag{2.4}$$

Proof. Suppose that (x, z) is optimal in BRP not satisfying (2.3), and let $s = \min\{t | z_t > \sum_{i \in \mathcal{N}} z_{t-T_i}\}$. It follows that no part requires replacement at $t = s$. Hence the maintenance occasion may be postponed until the failure of some part. As all costs are non-decreasing, it follows that doing so will not increase the objective value. □

Yet another version of this proposition is available.

Proposition 12 (No maintenance without failure, 2)**.** *If $c_{it}, d_t,\ t \in \mathcal{T}$, are non-increasing in t, then there exists an optimal solution that satisfies*

$$z_t \leq \sum_{i \in \mathcal{N}} x_{i,t-T_i} \tag{2.5}$$

where we have defined auxiliary parameters

$$x_{it} = \begin{cases} 0, & t \leq -1 \\ 1, & t = 0 \end{cases} \qquad i \in \mathcal{N}. \tag{2.6}$$

This follows as in the proof of Proposition 11. The careful reader will notice that (2.3) is implied by (2.5) when $T_i, i \in \mathcal{N}$, are distinct. We define a modified replacement polytope using the inequalities (2.5):

$$S' = \{x \in S | \text{(2.5) holds for } t \in \mathcal{T},\ t > \max T_i\}. \tag{2.7}$$

The assumption that $t > \max T_i$ is purely technical to retain fulldimensionality of S', which is shown in Proposition 14. If we were to incorporate inequalities for $t < \max T_i$ this would force $x_{it} = 0$ at some times. This is of course practically useful, however it would make the following exposition more technical.

The above propositions can now be summarized as:

$$S' \subset S, \tag{2.8}$$

and

$$\min_{x \in S} \left\{ \sum_{i \in \mathcal{N}, t \in \mathcal{T}} c_{it} x_{it} + \sum_{t \in \mathcal{T}} d_t z_t \right\} = \min_{x \in S'} \left\{ \sum_{i \in \mathcal{N}, t \in \mathcal{T}} c_{it} x_{it} + \sum_{t \in \mathcal{T}} d_t z_t \right\} \tag{2.9}$$

for c_{it}, d_t non-increasing in t. We also obtain the following proposition, showing that (2.5) is a strong inequality.

Proposition 13. *If $T_i \geq 3, i \in \mathcal{N}$, $|\mathcal{N}| \geq 2$, and $T_i, i \in \mathcal{N}$, are distinct, then the face of* $\operatorname{conv} S'$ *defined by* (2.5) *defines a facet of* $\operatorname{conv} S'$.

Proof. We use Theorem 4. To do this, we should first prove that $\operatorname{conv} S'$ is fulldimensional. For now, we assume that this is true, and postpone a proof until Proposition 14. Let $s \in \mathcal{T},\ s > \max_i T_i$, and define $F_s = \{(x,z) \in \operatorname{conv} \mathcal{S}', z_s = \sum_i x_{s,i-T_i}\}$. Let λ, μ, ρ be such that

$$\sum_{i\in\mathcal{N},t\in\mathcal{T}} \lambda_{it}x_{it} + \sum_{t\in\mathcal{T}} \mu_t z_t = \rho, \quad \forall (x,z) \in F_s. \tag{2.10}$$

We wish to show that $\lambda_{it} = \alpha\chi_{t=s-T_i}$, $\mu_t = -\alpha\chi_{t=s}$ and $\rho = 0$. We begin by defining (x^A, z^A) as $x^A_{it} = 1 - \chi_{t=s-T_i,t=s}$, where χ_U denotes the indicator function of the set U, and $z^A_t = 1 - \chi_{t=s}$. It follows that $(x^A, z^A) \in F_s$. For $j \in \mathcal{N},\ \tau \in \mathcal{T}\backslash\{s, s-T_j\}$, define (x^B, z^B) as $x^B_{it} = x^A_{it} - \chi_{i=j,t=\tau}$, $z^B_t = z^A_t$. It follows that $(x^B, z^B) \in F_s$, and insertion relative to (x^A, z^A) into (2.10) yields $\lambda_{j\tau} = 0$. For $\tau \in \mathcal{T}\backslash\{s\}$, define (x^C, z^C) as $z^C_t = z^A_t - \chi_{t=\tau}$, $x^C_{it} = 0, t = \tau$, $x^C_{it} = x^A_{it}$ otherwise. It follows that $(x^C, z^C) \in F_s$, and insertion into (2.10) yields $\mu_\tau = 0$. For $j \in \mathcal{N}$, define (x^D, z^D) as $x^D_{it} = x^A_{it} + \chi_{i=j,t=s-T_j}$, $z^D_t = z^A_t + \chi_{t=s}$. It follows that $(x^D, z^D) \in F_s$, and insertion into (2.10) yields $\lambda_{j,s-T_j} + \mu_s = 0$. Hence $\lambda_{j,s-T_j} = \alpha$ and $\mu_s = -\alpha$ for some constant α. The proposition now follows since insertion of any of the above vectors into (2.10) yields $\rho = 0$. □

We now show that $\operatorname{conv} S'$ is fulldimensional.

Proposition 14. $\operatorname{conv} S'$ *is fulldimensional.*

Proof. The proof of Proposition 13 shows that $\operatorname{codim}(F_s) = 1$, by virtue of Theorem 4, and the proposition follows if $\operatorname{conv} S' \neq F_s$. Define for some $j \in \mathcal{N}$, (x,z) as $z_t = 1, t \in \mathcal{T}$, $x_{it} = 1 - \chi_{i=j,t=s-T_j}$. It follows that $(x,z) \in \operatorname{conv} S'$, but $(x,z) \notin F_s$. □

We wish to point out that Propositions 11 and 12 still hold if we replace sums by $\max$. However, the resulting inequality will then be nonlinear which may be handled through the use of disjunctive inequalities [NW88], or they may be used to implement custom branching rules for a branch-and-bound scheme. We also want to alert the reader to the obvious fact that if we in Propositions 11 and 12 enforce costs to be strictly decreasing, then inequalities (2.3) and (2.5) holds for *any* optimal solution.

2.3 Inducing separability

The causal nature of the problem begs the question if there is some subproblem structure that can be exploited in a dynamic programming scheme, or a relaxation scheme. The basic idea is that given that (x,z) is optimal in an instance of BRP with time horizon T, we expect it to yield near-optimal solutions for problem instances where the planning horizon is changed. To formalize this, consider an instance of BRP with, for simplicity, an odd planning horizon $T = 2T_0 + 1$. We may then Lagrangian relax all constraints that contain variables at T_0+1, with multipliers u_j, yielding the Lagrangian subproblem

$$\min \sum c_{it}x_{it} + d_t z_t + \sum_j u_j \sum_{i,t} (\lambda^j_{it}x_{it} + \mu^j_t z_t) \tag{2.11a}$$

$$\text{s.t.} \quad \sum_{t=l+1}^{l+T_i} x_{it} \geq 1, \qquad i \in \mathcal{N},\ T_0+1 \notin \{l+1,\ldots,l+T_i\}, \tag{2.11b}$$

$$z_t \geq x_{it}, \qquad i \in \mathcal{N},\ t \in \mathcal{T}\backslash\{T_0+1\}, \tag{2.11c}$$

$$x_{it}, z_t \in \{0,1\}, \qquad i \in \mathcal{N},\ t \in \mathcal{T}. \tag{2.11d}$$

It follows that this problem separates into two smaller BRP problems, as relaxing all constraints containing variables at T_0+1 yields a problem where no constraint contains variables to the right and the left of T_0+1 simultaneously. These subproblems can then hopefully be solved much faster than the original problem. It should also be clear that we may divide the problem into more parts, yielding a potentially large set of smaller subproblems. By finding good multipliers with some algorithm, such as subgradient optimization or bundle methods, one may be able to obtain stronger lower bounds on the integer optimum than is obtainable through the linear relaxation. This method also gives us an indication of what defines difficult instances of BRP; if lives are large, it follows that we need to relax more constraints above, which heuristically may lead to weaker bounds. This relaxation is also a very natural one, in that it partitions the problem into two parts, and modifies the costs near the splitting to emulate the full problem, making us expect a tight duality gap. However, due to time constraints, the author has not investigated the strength and computational efficiency of such relaxations.

2.4 Uniqueness of optimal solutions

The solution to BRP may be highly degenerate, especially if the costs are constant in time. As an example, consider a simple instance of BRP with only one component, $n = 1$, and constant costs. It follows that the optimal solutions are exactly those which replace the component exactly $\left\lfloor \frac{T}{T_1} \right\rfloor$ times. Particularly, if $T_1 = T$ there are T_1 distinct optimal solutions. However, appealing to our intuitive view of maintenance, not all these solutions should be treated as being optimal, in the sense that we should not repair parts before they break. In this section we construct a way of perturbing the objective function of constant cost BRP in a way that breaks degeneracy, and such that the modified objective function has decreasing costs. The results of Section 2.2 provide a partial result towards this end, in that if we discount costs by a small amount, yielding strictly decreasing costs, the optimal solution will satisfy a quite constrictive inequality, breaking some degeneracy.

Let $\mathcal{P}$ be either the polytope of BRP or its continuous relaxation for some instance. Define the set $\mathbf{X}$ as

$$\mathbf{X} = \operatorname*{argmin}_{x \in \mathcal{P}} \left(\sum_{i \in \mathcal{N}, t \in \mathcal{T}} c_i x_{it} + \sum_{t \in \mathcal{T}} d z_t \right). \tag{2.12}$$

Now, for $0 < \epsilon_{it}^x, \epsilon_t^z < \epsilon$, we define

$$\mathbf{X}_\epsilon = \underset{x\in\mathcal{P}}{\operatorname{argmin}} \left(\sum_{i\in\mathcal{N},t\in\mathcal{T}} c_i(1-\epsilon_{it}^x)x_{it} + \sum_{t\in\mathcal{T}} d(1-\epsilon_t^z)z_t \right). \tag{2.13}$$

The theory of linear programming yields that for ϵ small enough, $\mathbf{X}_\epsilon \subset \mathbf{X}$. It follows that for $\epsilon > 0$ small enough,

$$\mathbf{X}_\epsilon = \underset{(x^k,z^k),k\in 1,\ldots,p}{\operatorname{argmax}} \left\{ \sum_{i\in\mathcal{N},t\in\mathcal{T}} c_i\epsilon_{it}^x x_{it}^k + \sum_{t\in\mathcal{T}} d\epsilon_t^z z_t^k \right\}. \tag{2.14}$$

We now describe a strategy for choosing $\epsilon_{it}^x, \epsilon_t^z$ such that $\mathbf{X}_\epsilon$ is a singleton set. Suppose that ϵ has been chosen such that the above proposition holds. First we pick ϵ_t^z in an recursive manner: let $0 < \epsilon_T^z < \epsilon$. Define $\mathcal{K}_T = \{1,\ldots,p\}$ and let

$$M_T = \max_{k\in\mathcal{K}_T}\{z_T^k\}, \tag{2.15a}$$

$$m_T = \max_{k\in\mathcal{K}_T|z_T^k<M_T}\{z_T^k\}. \tag{2.15b}$$

If $m_T = -\infty$ it follows that $z_T^k = M_T, \forall k \in \mathcal{K}_T$, and we define $\mathcal{K}_{T-1} = \mathcal{K}_T$. Otherwise we wish to arrange so that

$$d\epsilon_T^z > \frac{1}{M_T - m_t}\left(\sum_{i\in\mathcal{N},t\in\mathcal{T}} c_i\epsilon_{it}^x + \sum_{t=1}^{T-1} d\epsilon_t^z \right), \tag{2.16}$$

which can be guaranteed by forcing an upper bound δ_T on all ϵ except ϵ_T^z. In (2.16) we require that $0 < \delta_T < \alpha d\epsilon_T^z(M_T - m_T)$, where α is the remaining number of undefined ϵ. It follows that $(x^k, z^k) \in \mathbf{X}_\epsilon, k \in \mathcal{K}_T$ only if $z_T^k = M_T$. Thus we define $\mathcal{K}_{T-1} = \{k \in \mathcal{K}_T | z_T^k = M_T\}$, yielding $\mathbf{X}_\epsilon \subset \mathcal{K}_{T-1}$, with a slight abuse of notation. Recursively define

$$M_t = \max_{k\in\mathcal{K}_t}\{z_t^k\}, \tag{2.17a}$$

$$m_t = \max_{k\in\mathcal{K}_t|z_t^k<M_t}\{z_t^k\}. \tag{2.17b}$$

By setting bounds on ϵ as in (2.16) we can guarantee that $(x^k, z^k) \in \mathbf{X}_\epsilon$ only if $z_t^k = M_t$, from which we may define $\mathbf{X}_\epsilon \subset \mathcal{K}_t = \{k \in \mathcal{K}_{t+1} | z_t^k = M_t\}$. It follows that $\emptyset \neq \mathcal{K}_1$ and $(x^i, z^i) \in \mathcal{K}_1$ if and only if $z_t^k = M_t,\ t \in \mathcal{T}$. By repeating this procedure for each $i \in \mathcal{N}$, we obtain a singleton set $\mathcal{K}$ such that

$$\emptyset \neq \mathbf{X}_\epsilon \subset \mathcal{K} = \{(x^k, z^k) | z_t^k = M_t, x_{it}^k = M_t^i\} \tag{2.18}$$

from which it follows that $\mathbf{X}_\epsilon$ is also a singleton set.

It should be noted that the above procedure of setting bounds on ϵ is not computationally implementable, due to finite precision and rapidly decreasing bounds. However, in practice it usually suffices to let $\epsilon_{it}^z = \epsilon_t^z = \delta \frac{t}{T}$ for some small δ to break alot of degeneracy, even though there is no guarantee that optima become unique.

The main advantage of using these discounts yielding less degeneracy is that it significantly speeds up the cutting plane methods that will be discussed later in this thesis.

Chapter 3

The facial structure of the replacement polytope

In this chapter we derive two large partially overlapping classes of facets of the replacement polytope. For the first class, a full description, as well as proofs of validity and the facet property of the entire class, will not be presented in this chapter, as such a presentation is near impossible to understand on its own. Instead we will start with a simple subclass of facets, and give a full proof of validity and the facet property for this subclass. Then we demonstrate extensions and generalizations of this subclass, building up the entire class step by step. Full proofs of each step are not given here, instead we concentrate on the simplest cases and appeal to the reader's pattern recognition as well as introduce a pictorial representation[1] of how to construct proofs. Throughout we implicitly assume that for all components $T_i \geq 2$.

We start by proving a simple lemma, which will be referred to throughout this section.

Lemma 15 (Killing Lemma)**.** *Let*

$$\sum_{i\in\mathcal{N},t\in\mathcal{T}} \lambda_{it}x_{it} + \sum_{t\in\mathcal{T}} \mu_t z_t \geq \rho \tag{3.1}$$

be a valid inequality for the replacement polytope. Let F be the face of the replacement polytope defined by (3.1)*, and assume that F is proper. Let*

$$\sum_{i\in\mathcal{N},t\in\mathcal{T}} \lambda'_{it}x_{it} + \sum_{t\in\mathcal{T}} \mu'_t z_t = \rho' \tag{3.2}$$

be an equation that is satisfied for all $(x,z) \in F$. Then if $\mu_s = 0$ and there exists an $(x,z) \in F$ such that $z_s = 0$, then $\mu'_s = 0$. Alternatively, if $\lambda_{js} = 0$ and there exists an $(x,z) \in F$ such that $z_s = 1,\ x_{js} = 0$, then $\lambda'_{js} = 0$.

[1]The saying "A picture speaks more than a thousands words" has rarely been more true.

Proof. We prove this for $\mu_s = 0$; the other case follows similarly. Let $(x, z) \in F$ be such that $z_s = 0$. Since $\mu_s = 0$, it follows that (x', z') satisfies (3.1), where $x' = x$ and $z'_t = z_t + \chi_{\{t=s\}}$, and hence $(x', z') \in F$. Insertion into (3.2) then yields the desired result. □

3.1 Combinatorial facets

This section deals with a class of facets of $\mathrm{conv}(S)$ which are obtained through various combinatorial implications. It is not clear if or how these constructions may generalize to more general maintenance models, and as such, we will not go into an excessive amount of detail for all the constructions.

3.1.1 Simple facets

We start by taking a simple example of a facet, having just enough structure to encompass the crucial construction. Suppose we have an instance of (2.1), and pick $p, q \in \mathcal{N}$ such that $T_q < T_p$, and, for $l \in \{0, \dots, T - T_p\}$ and $s \in \{1, \dots, l + T_p - T_q\}$, consider the inequality

$$\sum_{t=l+1}^{l+s-1} x_{pt} + \sum_{t \in \{l+s, l+s+T_q\}} z_t + \sum_{t=l+s+1}^{l+s+T_q-1} (x_{pt} + x_{qt}) + \sum_{t=l+s+T_q+1}^{l+T_p} x_{pt} \geq 2. \quad (3.3)$$

The inequality is illustrated in Figure 3.1. Each node represents a point of time, and the variable name under it represents the variables to be added to the inequality at that point of time. Black nodes are nodes at which z_t is present in the inequality.

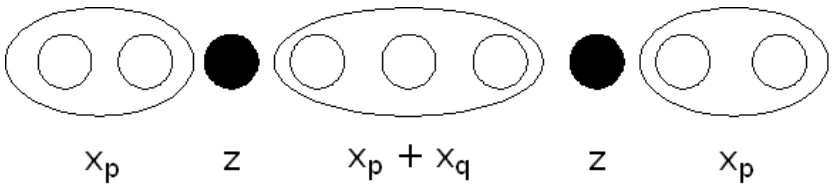

Figure 3.1: Pictorial representation of an inequality according to (3.3), with $T_p = 9$, $T_q = 4$, $s = 3$. Here the inequality represented is $x_{p1} + x_{p2} + z_3 + \sum_4^6 (x_{pt} + x_{qt}) + z_7 + x_{p8} + x_{p9} \geq 2$.

We also take this opportunity to fix some terminology. We call the component p enveloping component, whilst q is called a packing component, where we expect the etymology to be clear.

Proposition 16. (3.3) *is valid for* S.

Proof. The conclusion is trivial if $z_{l+s} = z_{l+s+T_q} = 1$; assume that $z_{l+s} = 0$. Then, by (2.1c), $x_{i,l+s} = 0$, and it follows from (2.1b) that $\sum_{t=l+s}^{l+s+T_q-1} x_{qt} \geq 1$. The remaining terms of the left hand side of (3.3) can, as $x_{p,l+s} = 0$, be written as

$$\sum_{t=l+1}^{l+T_p} x_{pt} + \left(z_{l+s+T_q} - x_{p,l+s+T_q}\right) - x_{p,l+s} \geq \sum_{t=l+1}^{t=l+T_p} x_{pt} + 0 + 0 \geq 1,$$

yielding that (3.3) is valid when $z_{l+s} = 0$. The case when $z_{l+s+T_q} = 0$ follows similarly. □

One might hope that this inequality is not only valid, but also strong, and the following proposition shows that this is indeed the case.

Proposition 17. (3.3) *defines a facet of* $\mathrm{conv}(S)$.

Proof. To avoid cluttering the proof with indices, we assume that $l = 0$; the general case follows similarly. We use the characterization of Theorem 4, which is applicable since $\mathrm{conv}(S)$ is fulldimensional. Let $F = \{(x, z) = \mathrm{conv}(S) | \text{(3.3) holds with equality}\}$. Let $\lambda,\ \mu,\ \rho$ be such that

$$\sum_{i \in \mathcal{N}, t \in \mathcal{T}} \lambda_{it} x_{it} + \sum_{t \in \mathcal{T}} \mu_t z_t = \rho \qquad \forall (x, z) \in F. \tag{3.4}$$

The first objective is to show that all coefficients λ_{it}, μ_t that are not included in (3.3) must be zero. To this end, define (x^A, z^A) as $z_t^A = 1,\ t \in \mathcal{T}$, and x_{it}^A maximally on F, that is $x_{it}^A = 0$ if $i = q,\ t \in \{s+1, \ldots, s+T_q-1\}$ or $i = p,\ t \in \{1, \ldots, s-1, s+1, \ldots, s+T_q-1, s+T_q+1, \ldots, T_p\}$, and $x_{it}^A = 1$ otherwise. It follows that $(x^A, z^A) \in F$. Varying x_{it}^A, $i \notin \{p, q\}$, and inserting into (3.4) yields $\lambda_{it} = 0$, $i \notin \{p, q\}$ or $i = p,\ t \in \{1, \ldots, s-1, s+1, \ldots, s+T_q-1, s+T_q+1, \ldots, T_p\}$ or $i = q,\ t \in \{s+1, \ldots, s+T_q-1\}$. Also by varying $z_t,\ t \notin \{s, s+T_q\}$ while varying x_{it} to keep it maximal on F we get that $\mu_t = 0,\ t \notin \{s, s+T_q\}$. Hence (3.4) can be written as

$$\sum_{t=l+1}^{l+s-1} \lambda_{pt} x_{pt} + \sum_{t \in \{l+s, l+s+T_q\}} \mu_t z_t + \sum_{t=l+s+1}^{l+s+T_q-1} (\lambda_{pt} x_{pt} + \lambda_{qt} x_{qt}) + \sum_{t=l+s+T_q+1}^{l+T_p} \lambda_{pt} x_{pt} = \rho. \tag{3.5}$$

By modifying (x^A, z^A) such that $z_s = 0$, $x_{pt} = 1$ for some $t \in \{1, \ldots, s-1, s+1, \ldots, s+T_q-1, s+T_q+1, \ldots, T_p\}$ we get that $\mu_s = \lambda_{pt} := \alpha$, $t \in \{1, \ldots, s-1, s+1, \ldots, s+T_q-1, s+T_q+1, \ldots, T_p\}$, and in the same way $\mu_{s+T_q} = \mu_s = \alpha$. In a similar manner, we can modify (x^A, z^A) in a way such that we get from (3.4) that $\lambda_{qt} = \mu_s = \alpha$. A final insertion of (x^A, z^A) into (3.4) yields $\rho = 2\alpha$. □

The rest of this section will be spent showing how generalizations of the facets (3.3) can be constructed. There are three basic extensions. In (3.3) the inequality contains $T_q - 1$ q-component variables, which can generalized to $rT_q - 1$, for integers $r \geq 1$. It

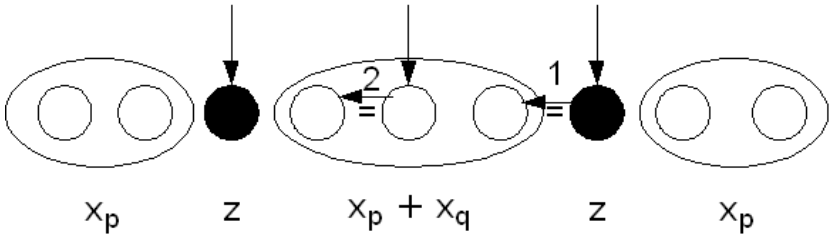

Figure 3.2: Pictorial representation of the jumping procedure, with $T_p = 9$, $T_q = 2$, $r = 2$.

is also possible to pack more than one component into a single enveloping component. Finally, we may also proceed recursively, taking an obtained inequality and packing it into an even larger component.

3.1.2 Simple extension

The first extension is fairly straightforward. Let the situation be as in (3.3), and let $r \geq 1$ be an integer, and let $s \in \{1, \ldots, T_p - rT_q\}$. Consider the inequality

$$\sum_{t=l+1}^{l+s-1} x_{pt} + \sum_{t \in \{l+s, l+s+T_q\}} z_t + \sum_{t=l+s+1}^{l+s+rT_q-1} (x_{pt} + x_{qt}) + \sum_{t=l+s+rT_q+1}^{l+T_p} x_{pt} \geq r + 1. \tag{3.6}$$

We can immediately see that with minor modifications the proof of Proposition 16 can be used to show that (3.6) is a valid inequality. Furthermore, the proof of Proposition 17 can easily be repeated to show that (3.6) is facet except for showing $\lambda_{qt} = \mu_s$, $t \in \{s+1, \ldots, s+rT_q-1\}$. This can however be shown by using what we will refer to as a jumping procedure, that we illustrate pictorially.

Figure 3.2 represents a sequence of feasible points on the face defined by (3.6). A vertical arrow indicates which relevant variables are set to 1, and a horizontal arrow represents a jump[2]. That is, the variable corresponding to the source of an arrow is set to 0, while the target variable is set to 1. The number above an arrow indicates in which order the jumps are performed. In figure 3.2 we jump through three points $(x^k, z^k),\ k = 1, 2, 3$. These are $x^1_{qt} = 1,\ t \in \{l+s, l+s+T_q, \ldots, l+s+rT_q\}$, $z^1_t = 1,\ t \in \{l+s, l+rT_q\}$, and $z^2_t = z^3_t = 1,\ t = l+s$, $x^2_{qt} = 1,\ t \in \{l+s, l+T_q, \ldots, l+(r-1)T_q, l+(r-1)T_q+T_q-1\}$, $x^3_{qt} = 1,\ t \in \{l+s, l+s+1, l+s+1+T_q, \ldots, l+s+1+(r-1)T_q\}$. An equality under a horizontal arrow shows what making the jump implies when inserting into (3.4). In Figure 3.2 the equality under arrow 1 means that $\mu_{l+s+rT_q} = \lambda_{q,l+s+rT_q-1}$, and under arrow 2 that $\lambda_{q,l+T_q} = \lambda_{q,l+T_q-1}$. Using this pictorial representation it follows that if we can find pictures connecting all points of time containing an x_q variable, we have shown that

[2]While, as usual, all other variables are defined maximally on the face.

$\lambda_{qt} = \lambda_{qt'}$ for $t, t' \in \{l+s+1, \ldots, l+s+rT_q-1\}$. For the situation depicted in Figure 3.2 we need only use the one depicted and its time reflection[3]. A more general situation is depicted in figure 3.3, and we appeal to the reader to understand that this example can be generalized to arbitrary instances.

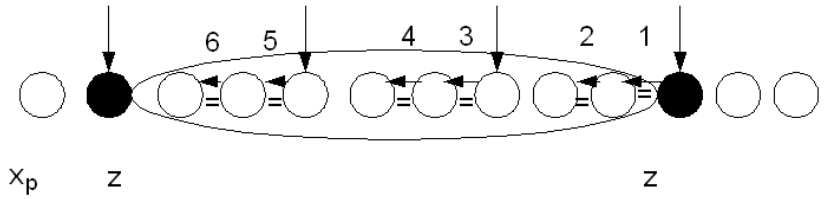

Figure 3.3: $T_q = 3,\ r = 3,\ T_p = 13$

3.1.3 Multiple packing components

The next, and significantly harder, step is to introduce more than one packing component into a single enveloping component. This is done as a partial superposition of constraints of the simpler form above, which from now on will be referred to as basic inequalities. Given m inequalities of the form (3.6), with the same $p,\ l$ such that the packing component and the window of time are the same, which are written as $\sum \lambda_{it}^{(k)} x_{it} + \sum \mu_t^{(k)} z_t \geq \rho^{(k)}$ for $k = 1, \ldots, m$, we define a new inequality by letting

$$\sum_{i,t} \lambda_{it} + \sum_{t} \mu_t z_t \geq \rho, \tag{3.7a}$$

$$\lambda_{it} = \sum_{k} \lambda_{it}^{(k)}, \qquad i \in \mathcal{N}\backslash\{p\},\ t \in \mathcal{T} \tag{3.7b}$$

$$\lambda_{pt} = \min_{k} \{\lambda_{it}^{(k)}\}, \qquad t \in \mathcal{T} \tag{3.7c}$$

$$\mu_t = \max_{k} \{\mu_t^{(k)}\}, \qquad t \in \mathcal{T} \tag{3.7d}$$

$$\rho = 1 + \sum_{k} \left(\rho^{(k)} - 1\right). \tag{3.7e}$$

Let $\tau_1^{(k)} = \min\{t | \mu_t^{(k)} = 1\}$ and $\tau_2^{(k)} = \max\{t | \mu_t^{(k)} = 1\}$. We will assume that we have ordered such that $\tau_1^{(1)} \leq \tau_1^{(2)} \leq \ldots \leq \tau_1^{(m)}$, where ties are broken with $\tau_2^{(1)} \leq \tau_2^{(2)} \leq \ldots \leq \tau_2^{(m)}$.

***Example* 1.** Let $T_1 = 3, T_2 = 4, T_3 = 8$. Define two inequalities of the form (3.3) by letting $l = 0$, $q^{(1)} = 1,\ s^{(1)} = 1$ and $q^{(2)} = 2,\ s^{(2)} = 4$. The valid inequalities become

[3]That is, the diagram that is obtained by flipping the diagram from left to right

$$z_1 + \sum_{t=2}^{3}(x_{1t} + x_{3t}) + z_4 + \sum_{t=5}^{8} x_{3t} \geq 2,$$
$$\sum_{t=1}^{3} x_{3t} + z_4 + \sum_{t=5}^{7}(x_{2t} + x_{3t}) + z_8 \geq 2.$$

Using (3.7) we obtain the inequality

$$z_1 + \sum_{t=2}^{3}(x_{1t} + x_{3t}) + z_4 + \sum_{t=5}^{7}(x_{2t} + x_{3t}) + z_8 \geq 3.$$

In this example we obtain $\tau_1^{(1)} = 1$, $\tau_2^{(1)} = 4$, $\tau_1^{(2)} = 4$, $\tau_2^{(2)} = 8$.

We now seek to find some conditions under which (3.7) is valid, and some conditions under which it defines a facet. We will from now use the shorthand (λ, μ, ρ) to denote an inequality of the form (3.7a).

Proposition 18. *An inequality of the form* (3.7) *such that if* $\tau_1^{(k)} = \tau_1^{(k')}$ *for some* $k \neq k'$*, then* $\tau_2^{(k)} \neq \tau_2^{(k'')}$ *for any* $k'' = 1, \ldots, m,\ k'' \neq k$*, is valid.*

Proof. We prove this using induction on the number of basic inequalities, m. Let $(x, z) \in S$. Proposition 16 and its extension yields the result for $m = 1$. Assume that the statement is true for any inequality with $m \leq L - 1$, and let (λ, μ, ρ) be such that $m = L$. Then the inequality (λ', μ', ρ') consisting of the first $L - 1$ basic inequalites is valid according to the induction assumption.

Suppose that the packing component of the basic inequality L is q. Then it holds by (2.1b) that $\sum_t \lambda_{qt}^{(L)} x_{qt} \geq \rho^{(k)} - 2$. If this is strict the conclusion is trivial. It is also trivial if (λ', μ', ρ') is strict, assume otherwise. It then follows from (2.1b) that $\sum_t \mu_t^{(L)} z_t = 2$, and from the assumptions on $\tau_j^{(k)},\ j = 1, 2$ it follows that $\sum_t (\mu_t - \mu_t') z_t \geq 1$. If this inequality is strict the conclusion is trivial, assume otherwise.

It remains to show that $\sum_t \left(\lambda_{pt} - \lambda_{pt}'\right) x_{pt} \geq 0$. This is however true, since the assumptions above imply $\sum_t \mu_t' z_t \geq 1$. As we assumed that (λ', μ', ρ') was satisfied with equality, $\sum_t \lambda_{pt}' x_{pt} = 0$ follows. □

It turns out that it is hard to characterize exactly when the inequalities of Proposition 18 define facets of the replacement polytope, and we will not treat this in any great detail. A partial result is given later in Proposition 21. Instead we move on, generating more valid inequalities.

3.1.4 Layered inequalities

Assume that we are given an inequality of the form in Proposition 18 (λ, μ, ρ) with packing component p at the window of time $\{l + 1, \ldots, l + T_p\}$. We also assume

that $\mu_{l+1} = 0$, for reasons that will become clear in Proposisition 19. Choose a new packing component p' such that $T_{p'} \geq T_p + 1$ and $l' \in \{l + T_p + 1 - T'_p, \ldots, l\}$. We define a new inequality (λ', μ', ρ') by

$$\sum_{i,t} \lambda'_{it} x_{it} + \sum_t \mu'_t z_t \geq \rho', \tag{3.8a}$$

$$\lambda'_{it} = \lambda_{it}, \qquad i \in \mathcal{N} \backslash \{p', p\},\ t \in \mathcal{T}, \tag{3.8b}$$
$$\lambda'_{pt} = \lambda_{pt} - \chi_{\{l+1\}}, \tag{3.8c}$$
$$\lambda'_{p't} = \chi_{\{l'+1,\ldots,l'+T_{p'}\} \backslash \{l+1, l+T_p+1\}}, \tag{3.8d}$$
$$\mu'_t = \mu_t + \chi_{\{l+1, l+T_p+1\}}, \tag{3.8e}$$
$$\rho' = \rho + 1. \tag{3.8f}$$

An illustrative example is shown in Figure 3.4.

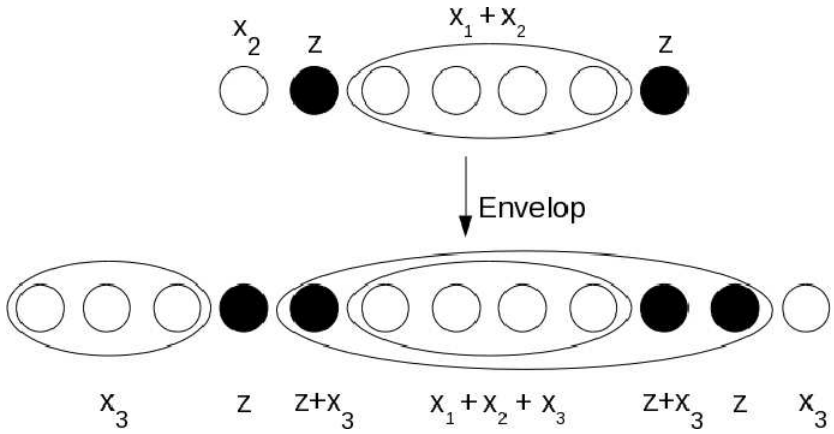

Figure 3.4: Example of how to envelop an existing contsraint into a larger one. Here $T_1 = 5,\ T_2 = 7,\ T_3 = 12$. Note that the packing inequality splits into two variants in the enveloped inequality. Looking at the nodes $4 - 10$, ignoring x_3, the original packed inequality is dominated by these coefficients. Looking at nodes $5-11$, ignoring x_3, there is a variant of the original packed inequality that is obtained by letting $l \rightarrow l + 1,\ s \rightarrow s - 1$.

Showing that this procedure defines new valid inequalities for S is similar to proving Proposition 16. This should perhaps not come as a surprise as the construction is recursive in nature, in the sense that we perform the same operation on the packed inequalities that we did with the original inequalities when constructing the simple packings.

Proposition 19. *The inequality defined by* (3.8) *is valid for* S.

Proof. We show that the inequality is valid for $S = S^0 \cup S^1$, where $S^\delta = \{(x, z) \in S | z_{l+1} = \delta\}$, $\delta \in \{0, 1\}$. When $z_{l+1} = 0$ the conclusion follows immediately from (λ, μ, ρ) being valid and the life constraint (2.1b) for p'. When $z_{l+1} = 1$, there is a packing inequality situated at $l + 1$, $s - 1$ that is dominated by the coefficients of (λ', μ', ρ'), as in Figure 3.4, and the conclusion follows for this case as well. □

The above proof indicates why we assumed that $\mu_l + 1 = 0$ above: when $z_{l+1} = 1$ we had to identify a new packing inequality to reach the conclusion, and this was done by letting $s \to s - 1$. This can not be done unless $\mu_{l+1} = 0$, or equivalently, $s \geq 2$ in the original packing inequality. This should also make it clear that could instead have assumed that $\mu_{l+T_q} = 0$. We also claim that Proposition 19 can be extended so that this new inequality can be embedded into yet another larger component p'' in the same fashion, if $\mu'_{l+1} = 0$.

It can also be shown that the embedding defines a facet of $\operatorname{conv} S$ if the embedded constraint does, under the simplifying assumption that the conditions of the Killing Lemma are satisfied.

Proposition 20. *Let (λ, μ, ρ) be a facet of $\operatorname{conv} S$ of one of the types discussed in this section, which is embedded into p' according to* (3.8) *yielding the inequality (λ', μ', ρ'). Then (λ', μ', ρ') defines a facet of $\operatorname{conv} S$, if the conditions of the Killing Lemma are satisfied.*

Proof. We only outline the proof. Let α, β, γ be as in Theorem 4. Proving that the inequality yields a facet can then be broken down into three steps:

1. eliminate all coefficients that are not present,

2. show that all embedded coefficients are equal, and

3. show that all enveloping coefficients are equal.

That all relevant coefficients are equal then follows as the embedded coefficients and the enveloping coefficients overlap.

The first step is established through the Killing Lemma. The second step follows by fixing $z_t = x_{p't} = x_{it} = 1$ on one of the links between the embedding and eveloping coefficients, and referring to the fact that the embedded constraint defined a facet. The last step follows by defining a feasible point where no embedded variables use the links between the embedding and the enveloping variables, from which it follows that the maintenance occasion for p' may be placed anywhere. □

We will refer to this as a *layered inequality*. That an inequality is referred to as being of the k:th layer indicates how many times we have packed inequalities into bigger components. The first layer inequalities are then simply (2.1b) and the simple facets of (3.3) are of the second layer. By enveloping an inequality of the k:th layer we then obtain an inequality of the $(k + 1)$:th layer.

3.1.5 Partial characterization of facial properties of multiple packings

It follows that all that is left to do to characterize which of the valid inequalities are facets amounts to determining what types of multiple packing components that are facets. The author has not been able to do this in full generality, but will here present a partial result.

Proposition 21. *In the notation of Proposition 18, assume that* $\tau_2^{(k)} = \tau_1^{(k+1)}$, $k \in 1, \dots, m-1\}$. *Then the inequality defined by* (3.7) *defines a facet of* $\operatorname{conv} S$.

Proof. We prove this in the same way that we proved this for the first extension with a jumping procedure. That is, we use Theorem 4, let F be the face of $\operatorname{conv} \mathcal{S}$ and let α, β, γ be such that

$$\sum_{i \in \mathcal{N}, t \in \mathcal{T}} \alpha_{it} x_{it} + \sum_{t \in \mathcal{T}} \beta_t z_t = \gamma, \quad \forall (x, z) \in F. \tag{3.9}$$

We claim that it is similar to other proofs already given to show that all coefficients not present in (3.7) are zero, and to show that $\alpha_{pt} = \beta_s$ otherwise. The crucial point is to show that the coefficients of the packed components are equal. This can however be accomplished using the same jumping procedure as before. Define a feasible solution on the face F (x^A, z^A) as $z_t^A = 1,\ t \in \mathcal{T}$, and $x_{it}^A = 1 - \lambda_{it},\ i \in \mathcal{N},\ t \in \mathcal{T}$. Then we proceed with the jumping procedure as illustrated in Figure 3.5 and its time reflection. □

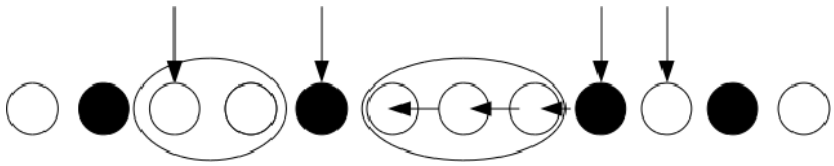

Figure 3.5: Illustration of the jumping procedure when using multiple packing components.

If we let $\tau_2^{(k)} < \tau_1^{(k+1)}$ it seems we still obtain facets in many cases if we require $T_i \geq 3,\ i \in \mathcal{N}$, and in some cases with $T_1 = 2$ as well. The issue is to show that all irrelevant coefficients have to be zero, and the author has not been able to find any reasonable characterization.

3.2 Chvátal-Gomory facets

In this section we outline a method of generating facets as first order Chvátal-Gomory inequalities. Numerical results based on automated facet generation suggest that many facets can be generated as simple Chvátal-Gomory inequalities using only $\frac{1}{2}$ as a multiplier from some subset of constraints, and we seek a classification of when such inequalities define facets of the replacement polytope. When using $u_i \in \{0, 1/2\}$ there

are two major benefits. The first is that the algebra is considerably simpler to work with than for general u_i. The other is that following an article by Caprara and Fischetti [CF93], in which such *zero-half cuts* were first discussed, modern general MIP solvers such as CPLEX have built-in features that generate zero-half cuts heuristically.

As we will only use multipliers $\frac{1}{2}$ it follows that we must use an odd number of inequalities (2.1b) for the Chvátal-Gomory procedure to yield anything non-trivial. Furthermore, for anything non-trivial to occur, the constraints must mix together before rounding. For two inequalities from (2.1b) this may happen in two ways. Either the inequalities come from the same component and correspond to values of l such that they overlap in time, or we have constraints from different components that overlap in time, for which we may add multiples of (2.1c) for the different components, transforming $\frac{1}{2}(x_{it} + x_{jt}) \geq z_t$. Hence our basic construction for Chvátal-Gomory inequalities will be to pick an odd number of inequalities (2.1b) that overlap in time, and mixing them together using (2.1c). The inequalities picked from (2.1b) will henceforth be called *constituent inequalities*.

3.2.1 The basic construction

We start by treating the case where all the constituent inequalities come from different components. That is, we have picked an odd number $k \geq 3$ of inequalities spanning over sets $\mathcal{T}_i,\ i \in 1, \ldots, k$, i.e. $\mathcal{T}_i = \{l_i + 1, \ldots, l_i + T_i\}$. To simplify, we assume that $k = n$ and $T_1 \leq T_2 \leq \ldots \leq T_k$. The assumption that $k = n$ might affect the application of the Killing Lemma negatively, however we disregard that as cases where it has any significance seem to be pathological. Furthermore, this will never cause any obtained inequality to be invalid.

We assume that for any $i \in \{1, \ldots, k\}$ there exist $j_1, j_2 \in \{1, \ldots, k\}\backslash\{i\}$, $j_1 \neq j_2$ such that $\mathcal{T}_i \cap \mathcal{T}_{j_1} \neq \emptyset$ and $\mathcal{T}_i \cap \mathcal{T}_{j_2} \neq \emptyset$. If $\mathcal{T}_i \cap \mathcal{T}_j \neq \emptyset$ let $t_{ij} \in \mathcal{T}_i \cap \mathcal{T}_j$, and pick a distinct subset $\mathcal{IJ}$, $|\mathcal{IJ}| = k$, of these. Now we are ready to execute the Chvátal-Gomory procedure. By a linear combination of the above inequalities we get a new valid inequality

$$\frac{1}{2}\left(\sum_{i=1}^{k}\sum_{t\in\mathcal{T}_i} x_{it} + \sum_{(i,j)\in\mathcal{IJ}} \left(2z_{t_{ij}} - x_{it_{ij}} - x_{jt_{ij}}\right)\right) \geq \frac{k}{2} \tag{3.10}$$

By first rounding the left hand side and then the right hand side we obtain, as k is odd,

$$\sum_{i=1}^{k}\sum_{t\in\mathcal{T}_i\backslash\{t_{ij},(i,j)\in\mathcal{IJ}\}} x_{it} + \sum_{(i,j)\in\mathcal{IJ}} z_{t_{ij}} \geq \frac{k+1}{2} \tag{3.11}$$

as a valid inequality.

***Example* 2.** We consider the simplest possible case, $k = 3$, $T_1 = 3$, $T_2 = 4$, $T_3 = 5$, $l_i = 0,\ i = 1, 2, 3$, $T = 5$, and the procedure is illustrated in figure 3.6. The valid inequality here becomes

$$(z_1 + x_{31}) + (z_2 + x_{22}) + (x_{13} + x_{23} + x_{33}) + z_4 + x_{35} \geq 2.$$

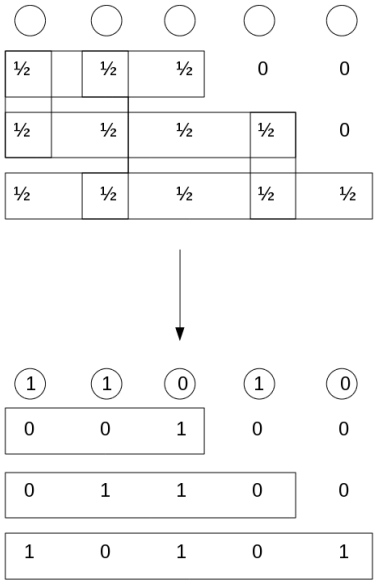

Figure 3.6: Example of a Chvátal-Gomory inequality. Horizontal rectangles indicate the constituent inequalities, while vertical groupings indicate the choice of $\mathcal{IJ}$. Numbers indicate the coefficients of the inequality. Coefficients inside circles indicate the coefficient for z_t at that time, and the number at row i column t is the coefficient for x_{it}.

Now we search for conditions such that the inequalities obtained define facets of the replacement polytope. To this end we define a graph $\mathcal{G} = (V, E)$, and state conditions in terms of this graph. Let $V = \{1, \ldots, k\}$, and let $e_{ij} \in E$ iff $(i, j) \in \mathcal{IJ}$. The interpretation of this graph is that nodes correspond to constituent inequalities, while an edge between two nodes state that the two constituent inequalities corresponding to those nodes have been mixed, and can thus be satisfied by letting $z_{t_{ij}} = x_{i,t_{ij}} = x_{jt_{ij}} = 1$. Note that this is satisfied by using only one variable of the left hand side of (3.11), due to the mixing. Furthermore we define $\mathcal{G}_i = (V_i, E_i)$ to be restricted graph when deleting node i.

We also define a notion of crossing for reasons that will become apparent in Proposition 23.

Definition 22. An edge e_{ij} of $\mathcal{G}$ is said to *cross* a node k if $\min_{t \in \mathcal{T}_k}\{|t - t_{ij}|\} \leq 1$.

***Example* 3** (*Example 2 continued*)**.** The graph corresponding to Example 2 is shown in Figure 3.7.

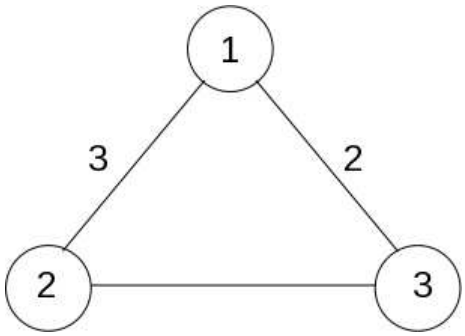

Figure 3.7: The graph corresponding to example 2. Numbers next to edges indicate which non-trivial nodes it crosses.

Now we are ready to state and prove the following proposition.

Proposition 23. *The inequality* (3.11) *defines a facet of the replacement polytope if for all $i \in \{1, \ldots, k\}$ we can partition $\mathcal{G}_i$ into connected subgraphs $\mathcal{G}_i^j = (V_i^j, E_i^j)$, $|V_i^j| = 2$ such that*

$$\hat{E}_i = E_i \backslash \left(\cup_{j=1}^{\frac{2}{k-1}} E_i^j \right) \tag{3.12}$$

does not contain any edges crossing i, and the conditions of the Killing Lemma hold.

Before moving on to the proof, we interpret the formulation of the proposition. An edge in the graph $\mathcal{G}$ indicate that both the constituent inequalities it links share a common variable. Furthermore, due to the conditions on how the graph was constructed, it is possible to satisfy both the constituent contraints by letting exactly one variable that is contained in the associated inequality be 1. A partitioning as in the proposition can

then be interpreted as satisfying the constituent constraints in pairs, using one variable from the inequality in each pair. By the requirements on the crossings, we have a large degree of freedom in how we choose to satisfy the last remaining constituent inequality. This allows us to find many affinely independent points satisfying our inequality with equality, which leads to the conclusion that it is a facet.

Proof of Proposition 23. The definitions that go into it make the proposition neat to prove. We use Theorem 4, and let F be the face defined by (3.11). Let (λ, μ, ρ) be such that

$$\sum_{i\in\mathcal{N},t\in T} \lambda_{it}x_{it} + \sum_{t\in\mathcal{T}} \mu_t z_t = \rho, \quad \forall (x,z) \in F. \tag{3.13}$$

The Killing Lemma removes all coefficients not part of (3.11), and the above equation reduces to

$$\sum_{i=1}^{n} \sum_{t\in\mathcal{T}_i\setminus\{t_{ij}|(i,j)\in\mathcal{IJ}\}} \lambda_{it}x_{it} + \sum_{(i,j)\in\mathcal{IJ}} z_{t_{ij}} = \rho \tag{3.14}$$

Note that the conditions imply that $\mathcal{G}$ is connected. Hence it suffices to show that $\lambda_{it} = \mu_s$ for $t \in \mathcal{T}_i\setminus\{t_{ij}|(i,j) \in \mathcal{IJ}\}$ and $s \in \{t_{ij}|(i,j) \in \mathcal{IJ}\}$. Then the coefficient inequalities will propagate through the graph, as by construction two nodes sharing an edge has at least one coefficient in common. Inspired by this we consider the constituent inequality m, and pick a partition of $\mathcal{G}_m$ corresponding to the assumptions. Define (x^m, z^m) as $z^m_t = 1$, $t \in \mathcal{T}\setminus(\cup\mathcal{T}_i)$ and $z^m_t = 1$ for $t = t_{ij}$ and some $(i,j) \in E\setminus\hat{E}_i$, $z^m_t = 0$ otherwise. Also, for $i \in \mathcal{N}\setminus\{m\}$ let $x^m_{it} = z^m_t$, $t \in \mathcal{T}\setminus\mathcal{T}_i$, $x^m_{it} = 1$ for $t = t_{ij}$ and some $(i,j) \in E\setminus\hat{E}_i$, $x^m_{it} = 0$ otherwise. Finally let $x^m_{mt} = z^m_t$ for $t \in \mathcal{T}\setminus\mathcal{T}_i$, $x^m_{mt} = 0$ otherwise. We can now see that (x^m, z^m) is almost feasible in BRP, as the only constraint it does not satisfy is (2.1b) for $\mathcal{T}_m$. Furthermore, plugging (x^m, z^m) into the left hand side of (3.11) yields $LHS(x^m, z^m) = \frac{k-1}{2}$. That is, to be feasible in BRP and satisfy (3.13) we must add a maintenance occasion for m at exactly one point in $\mathcal{T}_i$. As all edges crossing m by assumption are in E^j_m for some j[4], it follows that we may pick any maintenance occasion within $\mathcal{T}_i$, from which insertion into (3.14) yields $\lambda_{it} = \mu_s$ for $t \in T_i\setminus\{t_{ij}|(i,j) \in \mathcal{IJ}\}$ and $s \in \{t_{ij}|(i,j) \in \mathcal{IJ}\}$. □

The natural structure of the above proof is somewhat obscured by the details, and we demonstrate with a simple example.

***Example* 4** (Example 2 continued)**.** In this example it is fairly easy to see that the Killing Lemma holds, and in the graph 3.7 we easily see that the conditions of Proposition 23 hold. Let us show that all coefficients of constituent inequality 2 are equal, that is, that $\mu_1 = \lambda_{22} = \lambda_{23} = \mu_4$. The partitioning of V_2 is trivial, and we see that we retain only the edge e_{13}, which corresponds to setting $z^2_2 = x^2_{12} = x^2_{32} = 1$ in the proof above. We then see that the constituent inequalities 1 and 3 are satisfied in Figure 3.6. We may now perform maintenance of component 2 at any point in the constituent inequality 2 at the cost of increasing the right hand side of (3.11), as we have already

[4] And hence the value of the corresponding z_t equals 1.

set $z_2 = 1$, which after insertion into (3.14) yields $\mu_1 = \lambda_{22} = \lambda_{23} = \mu_4$. Similarly we may for constituent inequality 1 show that $\mu_1 = \mu_2 = \lambda_{13}$, and we see that both constituent inequalities share μ_1 as a coefficient, yielding $\mu_2 = \lambda_{13} = \mu_1 = \lambda_{22} = \lambda_{23} = \mu_4$. The incorporation of the coefficients for constituent inequality 3 follows analogously.

We also want to alert the reader to that the assumption on the Killing Lemma seems to be very weak, and it mainly serves to avoid the unpleasantness of handling the details that become neccessary to lift it. The cases where the conditions of the Killing Lemma do not hold are typically of form where some component life is $T_i = 2$, which can be viewed as a pathological case.

3.2.2 Extensions

We now discuss two extensions of the above proposition. Firstly, Proposition 23 was stated completely in terms of the associated graph, and from now on we will talk about zero-half cuts in terms of their graphs. The assumptions on how the inequalities were generated (in terms of t_{ij}) forced the associated graph to be simple (i.e. two nodes are joined by at most one edge). This restriction can in fact be lifted by only requiring that t_{ij} are distinct, and the proof goes through with some technical modifications.

Secondly, we can extend the above results to inequalities where the constituent inequalities do not have distinct components. For each component $i \in \mathcal{N}$ pick a number k_i of constraints at l^i_j, $j = 1, \ldots, k_i$ such that

$$l^i_2 \leq l^i_1 + T_i - 1, \tag{3.15a}$$

$$l^i_{j+2} = l^i_j + T_i, \qquad j = 2, \ldots, k_i \tag{3.15b}$$

and $\sum_i k_i$ is odd. Now we wish to pick inequalities from (2.1c) following certain rules, such that we link together the constraints picked above. For every $i \in \mathcal{N}$ such that $k_i \geq 2$, we pick inequalities $z_{l^i_2} - x_{i,l^i_2} \geq 0$ and $z_{l_{k_i-1}+1} - x_{i,l_{k_i}+1}$ from (2.1c), and require that we pick other inequalities from (2.1c) for the same times, with different components. For $i \in \mathcal{N}$ such that $k_i = 1$ the same rules apply as before.

***Example* 5.** In Figure 3.8 we demonstrate an inequality of the above mentioned type.

We now define a graph for this inequality as before, with the addition that we also let there be an edge between two nodes if the constituent inequalities come from the same component and overlap. In total, it now follows that two nodes of the inequality graph share an edge if and only if it is possible to satisfy the constituent inequalities by only using one variable of the generated inequality, if we disregard the crossings. This was the essential property that facilitated Proposition 23, and it still goes through with some modifications.

3.2.3 Lifting crossings

In Proposition 23, we had to assume some structure on the crossings, otherwise the associated inequalities would not define facets of the replacement polytope. We indicate

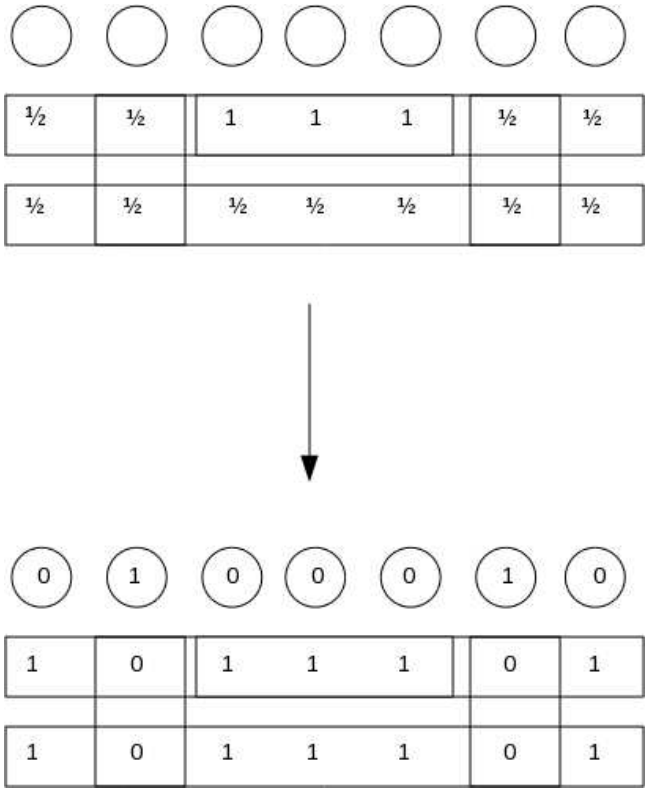

Figure 3.8: A CG inequality generating without distinct constituent inequalities. Note that this inequality is also of the simple combinatorial type.

how some of these crossed inequalities can be lifted to facets. Consider an inequality as in Proposition 23, except that for one node in the graph, there is one edge crossing that node which can not be included in the partition. Let the crossed node come from component $i \in \mathcal{N}$, and the crossing occur at $t \in \mathcal{T}_i$. It follows that the coefficients for x_{it} and z_t are 1 in this inequality. We wish to show that a modified inequality, where we let the coefficient of x_{it} be 0, is valid. Then the proof of Proposition 23 will go through for this new inequality, as the only hindrance was the crossing.

Denote the original inequality by $\pi^T(x,z) \geq \pi_0$, and the modified by $\pi'^T(x,z) \geq \pi'_0$. It is immediately clear that the primed inequality is valid when $z_t = 0$, since then $x_{it} = 0$, and the primed and unprimed inequality are equivalent. We must show that primed inequality is valid when $z_t = 1$. The result follows if we can show that if $z_t = 1$ and $\pi^T(x,z) = \pi_0$, then $x_{it} = 0$. But if $\pi^T(x,z) = \pi_0$, then (x,z) corresponds to a partitioning of the graph V into subgraphs of size 2 and one subgraph of size 1. By assumption no such partitiong exists that contains the edge corresponding to $z_t = 1$ and $x_{it} = 1$ and the result follows.

The careful reader may have noticed that the graphs representing the inequalities above share a striking similarity to *odd cycle inequalities without chords* for set covering/packing polyhedra [Pad73]. In fact, if we were to perform the above construction for a set covering instance, proving similar propositions, we would get exactly the odd cycle inequalities. This should perhaps not come as a surprise, as the constraints (2.1b) of the replacement problem are of the set covering type. However, using only these to try and generate odd cycle inequalities is futile, due to the total unimodularity of this submatrix from Proposition 7. What we are doing above can then be viewed as a perturbed version of the odd cycle inequalities that occur due the nature of constraints (2.1c). This observation begs the question of whether other cutting planes for set covering polytopes can be perturbed in such a way as to be applicable for the replacement problem.

We conclude this section by noting that the characterizations of when zero-half cuts become facets, or even valid, seem badly adapted for inclusion in a computer program. The characterizations are indirect and abstract, and the author has not been able to formulate even a good heuristic algorithm for generation[5]. Hence the separation problem of finding the most violated zero-half cut may very well be hard to solve. Instead we take the results to mean that zero-half cuts can be a powerful tool for BRP, and as such an efficient solver will likely benefit from an aggressive heuristic zero-half cut generation. However the abstract characterization is a double-edged sword; this is also what makes it likely that the methodology of identifying facets may be generalized to other replacement models.

3.3 Extended models

It was previously mentioned that some of the facets derived were of the odd cycle inequality type, that could be utilized since the problem had a set covering substructure.

[5]The facets of this section seem to behave nicely when the associated graphs are simple. However, preliminary numerical results indicate that graphs corresponding to useful inequalities are very far from being simple.

We may then ask ourselves whether the work done generalizes to more realistic models for opportunistic maintenance, if they possess a set covering substructure. In this section we give an example of such a generalization, and argue that many of the results above generalize to other models as well.

3.3.1 Deterministic individual components

Underlying BRP is the implicit assumption that for each component that needs replacement, there is an infinite pool of identical component individuals to choose from. In a more realistic setting this may not be the case. Let us consider the set $\mathcal{N}$ as a set of tasks that must be performed, and let us define a set $R_i,\ i \in \mathcal{N}$ of components that can perform that task. We can then model an opportunistic maintenance problem with the variables

$$x_{it}^r = \begin{cases} 1, & \text{individual } r \text{ of component } i \text{ is replaced at } t \\ 0, & \text{otherwise} \end{cases} \tag{3.16a}$$

$$z_t = \begin{cases} 1, & \text{maintenance performed at time } t \\ 0, & \text{otherwise} \end{cases} \tag{3.16b}$$

where $i \in \mathcal{N}$, $t \in \mathcal{T}$ and $r \in \{1, \ldots, n_i\} = R_i$, and define the *Uncapacitated Individual Replacement Problem* (UIRP) as

$$\min \sum_{i \in \mathcal{N}, t \in \mathcal{T}} \sum_{r \in \mathcal{R}_i} c_{it}^r x_{it}^r + \sum_{t \in \mathcal{T}} d_t z_t \tag{3.17a}$$

$$\text{s.t.} \quad \sum_{r \in R_i} \sum_{t=\min(l - T_i^r + 1, 1)}^{l} x_{it}^r \geq 1, \qquad i \in \mathcal{N}, l \in \{S_i, \ldots, T\}, \tag{3.17b}$$

$$\sum_{r \in R_i} \sum_{t=1}^{S_i} x_{it} \geq 1, \qquad i \in \mathcal{N}, \tag{3.17c}$$

$$x_{it}^r \leq z_t \qquad i \in \mathcal{N}, t \in \mathcal{T}, r \in R_i \tag{3.17d}$$

$$z_t, x_{it}^r \in \{0, 1\} \qquad i \in \mathcal{N}, t \in \mathcal{T}, r \in R_i \tag{3.17e}$$

where T_{it}^r is the lifetime of indidual r of component i at time t and S_i is the remaining life of component i at the start of the planning horizon. The structure is very similar to that of BRP: the objective is to minimize the cost of keeping the system running over the planning period. The canonical example of such a model would be a repair versus replace scenario. That is, given some component failure, we have the option of replacing the component by a brand new one, or to repair it. Repairs are probably less costly, but the next component failure may occur earlier. In this case $|R_i| = 2$.

3.3.2 Varying lives

The assumption in BRP that all components have lives that are constant may not be satisfied in several real-world applications. An example where this is not the case could be a cooling or heating system that carries a different load depending on the weather conditions. We therefore modify the BRP by considering each component as having a maximum total usage capacity L_i, and associating with each unit of time a usage load u_{it}. We assume that a component breaks if its total usage since its last replacement exceeds L_i, and to keep the system running we must replace components before they break. We may model this by defining

$$T_i(l) = \max \left\{ t \mid \sum_{l+1}^{l+t} u_{it} \leq L_i,\ l + t \leq T \right\},\ l \in \mathcal{T} \tag{3.18}$$

yielding the Varying Lifelength Replacement Problem (VLRP) as

$$\min_{(x,z)} \quad \sum_{i \in \mathcal{N}, t \in \mathcal{T}} c_{it} x_{it} + \sum_{t \in \mathcal{T}} d_t z_t \tag{3.19a}$$

$$\text{s.t.} \quad \sum_{l+1}^{l+T_i(l)} x_{it} \geq 1, \qquad i \in \mathcal{N}, 0 \leq l \leq T - T_i(l), \tag{3.19b}$$

$$x_{it} \leq z_t, \qquad i \in \mathcal{N}, t \in \mathcal{T}, \tag{3.19c}$$

$$x_{it}, z_t \in \{0, 1\}, \qquad i \in \mathcal{N}, t \in \mathcal{T}. \tag{3.19d}$$

3.3.3 Conjectures for general models

We can see in both the above proposed models that they contain a set covering substructure, coupled with mixing constraints. Although a detailed inspection of previous constructions will not be given, the author is confident that many previous results hold for these models as well. In particular, Proposition 23, which is quite abstract to its nature, should fairly easily generalize to these models as well. Furthermore, results on decreasing cost instances and the breaking of degeneracy should also generalize. This leads to writing out two conjectures for replacement models with set covering substructure.

Metaproposition 24. *If costs are non-increasing, maintenance may be postponed until the failure of some component.*

This metaproposition states that the fundamental property that facilitates the shortest path solver of Section 4.2 is satisfied. However it is not entirely clear how the *á priori* reductions on the number of possibly optimal maintenance decisions that are given for the basic model may extend to other models.

Metaproposition 25. *Zero-half cuts generated as in Section 3.2 will often define facets.*

It is outside the scope of this thesis to define general conditions under which these metapropositions become actual propositions, or even to define what is meant by a

general replacement model. However we expect that the reader is able to follow the Chapters 2 and 3 for UIRP and VLRP, and perform the modifications that are neccessary.

Chapter 4

Algorithms

We propose a polynomial time separation algorithm, as well as a shortest path based algorithm for solving BRP.

4.1 Constraint generation using graphs

The resulting linear programming problem that is obtained by adding all the facets obtained in Chapter 3 to the problem (1.2) can be seen to have a very large number of constraints, and an explicit treatment of all facets soon becomes intractable. However, some of the inequalitites have sufficient structure to allow for a constraint generation approach. This is done by constructing several graphs of a reasonable size, which will allow the separation problem to be solved as shortest path problems in these graphs. The algorithm will also in some cases produce valid inequalities that are not facets. Not all the inequalities derived in Chapter 3 will be generated, and what follows below is an algorithm for solving the restricted separation problem of generating some of the layered facets of Section 3.1.4 in polynomial time.

We will start by restricting the problem to generating facets of the second layer. We will also assume that $\min T_i \geq 3$, and that $T_1 < T_2 < \ldots < T_n$, which are simplifying conditions that we expect the reader to understand how they can be lifted. Let us fix a packing component p, and construct a weighted digraph on pT nodes:

$$V = \{(s,k) | s \in \mathcal{T},\ k \in \{1,\ldots,p-1\}\}, \tag{4.1a}$$
$$E = \{e_{(s,k),(s',k')}\}, \tag{4.1b}$$

where $e_{(s,k),(s',k')} \in E$ if $s - s' \geq 1$, $s, s' \in \mathcal{T}$, and

$$\begin{cases} s - s' = mT_{k'}, & k = p,\ k' \leq p-1,\ m \geq 1, \\ s \leq s', & k' = p. \end{cases} \tag{4.2}$$

Given a solution (x^{LP}, z^{LP}) to an LP-relaxed version of (2.1), the weights of the edges $w_{(s,k),(s',k')}$ are defined as

$$\sum_{t=s+1}^{s'-1} (x^{LP}_{k't} + x^{LP}_{pt}) + z^{LP}_{s'} - m, \qquad s - s' = mT'_k,\ k, k' \leq p-1,\ m \geq 1 \tag{4.3a}$$

$$\sum_{t=s+1}^{s'-1} (x^{LP}_{k't} + x^{LP}_{pt}) + z^{LP}_{s} + z^{LP}_{s'} - x^{LP}_{ps} - m, \quad s - s' = mT_{k'},\ k = p,\ k' \leq p-1,\ m \geq 1 \tag{4.3b}$$

$$\sum_{t=s+1}^{s'} x^{LP}_{pt}, \qquad s \leq s',\ k' = p. \tag{4.3c}$$

We claim that a path $\mathcal{P}$ through this graph from $(l+1, p)$ to $(l+T_p, p)$ for some l corresponds to a second layer inequality $g(x, z) \geq r$ with the packing component p of the type in Proposition 21. Furthermore, $g(x^{LP}, z^{LP}) - r = L(\mathcal{P}) - 1$ where $L(\mathcal{P})$ is the weighted length of that path. Hence if $L(\mathcal{P}) < 1$ we have identified an inequality that is violated in the LP-relaxed solution. This can be understood if we interpret (4.3) properly. An inequality of the type we are discussing lives on the times $t \in \{l+1, l+T_p\}$, for some l. The edges represent feasible ways in which we can 'fill' this timeframe with coefficients in a way corresponding to a valid inequality. The weights (4.3a)-(4.3b) respresent the addition of a packed inequality: (4.3a) adds a packing such that $\tau_2^{(k)} = \tau_1^{(k+1)}$, while (4.3b) yields a packing where the τ:s are distinct. The exact expressions for how this is done are derived from (3.7).

To prove these statements we begin by defining a mapping $f : X \to \mathbb{R}^{nT+T}$, where X is the space of paths in (V, E). The inequality corresponding to the path $\mathcal{P}$ will then be given as $f(\mathcal{P})^T(x, z, 1) \geq 1$.

***Example* 6.** We give an example of a path through a graph such as above. Let $p = 3$, and $T_1 = 4,\ T_2 = 6,\ T_3 = 11$. Figure 4.1 shows an inequality that is generated as a path. The resulting inequality becomes

$$z_1 + z_5 + z_{11} + \sum_{t=2}^{4}(x_{1t} + x_{3t}) + \sum_{t=6}^{10}(x_{2t} + x_{3t}) \geq 3 \tag{4.4}$$

Definition 26. The mapping $f : X \to \mathbb{R}^{nT+1}$ is defined as $f(\mathcal{P}) = \sum_{e \in \mathcal{P}} f(e)$, and $f(e_{(s,k),(s',k')})$ is defined as in (4.3) where x^{LP}_{it}, z^{LP}_{it} are replaced by the corresponding unit vectors, and constants m are replaced with the unit vector $(0, 0, \ldots, 0, m)^T$.

We can now prove the following proposition.

Proposition 27. *If $\mathcal{P}$ is a path in (V, E) from $(l+1, p)$ to $(l+T_p, p)$ for some $l \in \{0, \ldots, T - T_p\}$, then*

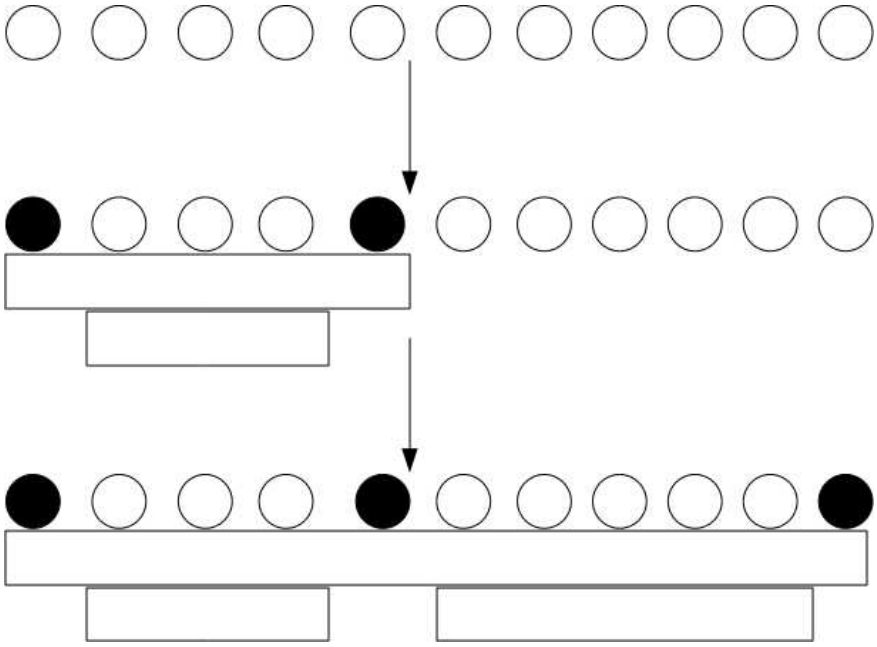

Figure 4.1: Pictorial representation of the a path through a graph according to (4.3), with $T_3 = 11,\ T_1 = 4,\ T_2 = 6$. The first row shows the initial (empty) situation, the second rows the coefficients that are added after traversing the first edge, and so on.

$$f(\mathcal{P})^T (x, z, 1) \geq 1 \tag{4.5}$$

is a valid inequality of the type (3.7) *with the properties of Proposition 21 of the replacement polytope.*

Proof. We prove this using induction over $|\mathcal{P}|$, where $|\mathcal{P}|$ is the number of nodes in the path. If $|\mathcal{P}| = 2$, the statement is trivial, since then clearly (4.5) is an inequality of the form (2.1b). If $|\mathcal{P}| = 3$ and the last node of $\mathcal{P}$ is $(l + T_p, p)$, the statement is also trivial. Now assume that the statement is true for all $\mathcal{Q}$ such that $|\mathcal{Q}| \leq u$, $u \geq 2$, and all $\mathcal{Q}$ such that $|\mathcal{Q}| = u+1$ where the last node of $\mathcal{Q}$ is $(l+T_p, p)$, and let $|\mathcal{P}| = u+1$. Then, $(s, k) \in \mathcal{P}$ for some $s \in \{l + 1, \ldots, l + T_p\},\ k \leq p - 1$, or (4.5) is trivially of the form (2.1b). Let s be the last such node. Then, removing the node (s, k) from $\mathcal{P}$ and collapsing its inward and outward edge[1] to an edge to $(l + T_p, p)$ yields a path of the form in the induction assumption, which yields a valid inequality of the type (3.7). Furthermore $\mathcal{P}$ can then be viewed as this valid inequality with the addition of (s, k) defining a new basic inequality. Hence, by (3.7), it follows that $f(\mathcal{P})^T(x, z, 1) \geq 1$ is also a valid inequality of the replacement polytope of type (3.7). The final case where $|\mathcal{P}| = u + 2$ with the last node being $(l + T_p, p)$ is treated similarly. □

It is also clear from Definition 26 and (4.3) that $f(\mathcal{P})^T(x^{LP}, z^{LP}, 1) = L(\mathcal{P})$.

From the preceding discussion it now follows that the restricted separation problem can be solved by n many-to-many shortest path problems, and hence that this restricted problem can be solved in polynomial time. However, as noted above, not all inequalities of Chapter 3 are generated in this fashion. The approach of viewing the separation problem as shortest problems through graphs may be extended to higher layer facets: recursively we solve the separation problem for the k-th layer and use the solution to update the weigths for layer $k + 1$. We also point out that the graph can be made significantly smaller than is described above: we may replace the pT nodes with only $3T$ nodes, however such a description becomes more involved. The crucial observation is that we need only keep track of whether the last node had $k = p$ or not, however the weights must be defined differently. The issue that is harder to solve is generating inequalities where packing components overlap, and there is currently no known efficient algorithm for generating the zero-half cuts.

4.2 Dynamic search algorithms

In this section we propose and discuss a shortest path based algorithm in Markov state space for computing solutions to BRP, under natural assumptions on the costs c_{it} and d_t.

4.2.1 Non-increasing costs

We consider instances of BRP where c_{it} and d_t are decreasing in t for all $\in \mathcal{N}$. Consider the weighted digraph $\mathcal{D} = (V, E, W)$ where

[1] with a trivial modification if the node has no outward edge

$$V = \{(t, \tau_1, \ldots, \tau_n), \tau_i \in \{0, \ldots, T_i - 1\} \cup \{(0, T_1, \ldots, T_n), (T + 1)\}, \quad (4.6)$$

i.e. a node in V represents a possible state of the maintenance system, in the sense that it tracks to what point time the system has survived during a maintenance schedule, and the remaining lives of each component, as well as auxiliary initial and final states for $t = 0, T + 1$. We define the edges of this digraph as $e \in E$ if e goes from a node s^1 to a node s^2 such that $t^2 - t^1 \leq \min_i T_i$ and $t^2 - t^1 = \tau_i^1$ for some $i \in \mathcal{N}$ such that $\tau_i \neq 0$, or $t^2 - t_1 = \min_i T_i$. We also require that

$$\tau_i^2 = \begin{cases} \tau_i^1 - (t^2 - t^1), & \text{if } \tau_i^1 - (t^2 - t^1) \geq 0, \\ T_i - (t^2 - t^1), & \text{othwerwise.} \end{cases} \quad (4.7)$$

These edges can interpreted as linking together two maintenance states if there exists a maintenance decision at t_1 such that we can advance time to t_2 without system failure. Furthermore the remaining life of the components at the new state will evolve according to 4.7.

We also define extra edges $e \in E$ from s^1 to s^2 if $t^2 = T+1$ and $t^2 - t^1 \leq \min_i T_i$. It follows that the outward degree of each vertex in $\mathcal{D}$ is less than n. The weigths of the edges $w \in W$ from states at t^1 to states at t^2 are defined as

$$w = d_{t^2} + \sum_j c_{jt^1}, \quad (4.8)$$

where the sum runs over all indices j such that $\tau_j - (t^2 - t^1) < 0$. The weights of edges connected to states where $t = 0$ is d_t, and edges reaching $t = T + 1$ have weight $\sum_j c_{jt}$, with j running as before. One should think of this graph as describing states of the maintenance and connecting states if there exists a maintenance decision at t^1 such that we reach the state at t^2, while keeping the system running until t^2 whilst satisfying (2.3). The weights are given as the costs to perform those maintenance decisions. Let X be the space of paths in $\mathcal{D}$ from $(t, T_1, \ldots, T_n)$ to $(T+1)$, and the define the mapping $f : X \to \{0, 1\}^{nT+T}$, as $f(\mathcal{P}) = \sum_{e \in \mathcal{P}} g(e)$, where $g : E \to \{0, 1\}^{nT+T}$ by letting

$$(g(e))_{(i-1)+t} = \begin{cases} 1, & t = t^1,\ \tau_i^1 - (t^2 - t^1) < 0, \\ 0, & \text{otherwise} \end{cases} \qquad i \in \mathcal{N}, t \in \mathcal{T}, \quad (4.9)$$

$$(g(e))_{nT+t} = \begin{cases} 1, & t = t^2 \leq T + 1, \\ 0, & \text{otherwise.} \end{cases} \qquad t \in \mathcal{T}. \quad (4.10)$$

Note that the definition of g yields that the range of f really is $\{0, 1\}^{nT+T}$, due to the structure of $\mathcal{D}$. We will from now understand the binary string $y \in \{0, 1\}^{nT+T}$ as corresponding to (x, z) in BRP through $x_{it} = y_{(i-1)T+t},\ z_t = y_{nT+T}$ for $i \in \mathcal{N}, t \in \mathcal{T}$, and simply denote $f(\mathcal{P}) = (x^P, z^P)$. Also note that $w(x^P, z^P) = (c, d)^T (x^P, z^P)$, where w is the pathlength function.

Proposition 28. *(x^P, z^P) is feasible in BRP, and satisfies* (2.3)*.*

Proof. We need to show that (x^P, z^P) satisfies (2.1b) and (2.1c), as the rest is trivial. We begin with (2.1c). Let j, s be such that $x^P_{js} = 1$, and we must show that $z^P_s = 1$. But then, according to the definition of g, there must exist an $e \in \mathcal{P}$ leaving a state with $t_e = s$, and as $\mathcal{P}$ is a path neither starting nor ending in a state with $t = s$ there exists an edge entering this state, e'. But then, from the definition of g, it follows that $z^P_s = 1$.

Now let $j \in \mathcal{N}$ and $l \in \{0, \dots, T - T_j\}$. We wish to show that $\sum_{t=l+1}^{l+T_j} x^P_{jt} \geq 1$. By definition, E only contains an edge from $t^1 \geq 1$ to t^2 if $t^2 - t^1 \leq \min_i T_i$. Hence the path $\mathcal{P}$ passes through a state $v \in V$ with $t_v \in \{l+1, \dots, l+T_i\}$, and from the definition of E it follows that for some edge to such a state, it must hold that $\tau_j - (t^2 - t^1) < 0$, as $\tau_j < T_j$. The definition of g then yields that for some $t \in \{l+1, \dots, l+T_j\}$, $x^P_{jt} = 1$. □

Proposition 29. *Let (x^*, z^*) be an optimal solution to BRP with non-increasing costs. Then there exists a path $\mathcal{P} \in X$ such that $w(x^P, z^P) = (c, d)^T(x^*, z^*)$*

Proof. As the costs are non-increasing we may assume that (x^*, z^*) satisfies (2.5), by Proposition 12. We prove the proposition by explicitly constructing the path $\mathcal{P}$. Let $t_1 < t_2 < \ldots < t_m$ be the times where $z^*_t = 1$, and let $t_0 = 0$. It follows that if we can find $\mathcal{P} = \{e_1, \dots, e_{m+1}\}$ such that it yields (x^*, z^*), the statement is true. By (2.3) it follows that $t_1 = \min_i T_i$, and we let $e_1 = (0, T_1, \dots, T_n) \to (T_1, 0, T_2 - T_1, \dots, T_n - T_1) \in E$. Now assume that for $j = 2, \dots, m$, the edges $e_1, \dots, e_{j-1}$ have been defined such that $\sum_{i=1}^{j-1} g(e)$ agrees with (x^*, z^*) for all times $t < t_j$, and look for an edge from the state at t_{j-1} to some state at t_j. By (2.3) and the defining properties of E, such an edge exists, and it only remains to show that $x^P_{it_{j-1}} = x^*_{it_{j-1}}$ for $i \in \mathcal{N}$ for some such edge. To this end, define $s_i = \max(t | x^*_{it} = 1, t < t_{j-1})$ for $i \in \mathcal{N}$. Trivially, $s_i \in \{t_0, t_1, \dots, t_{j-2}\}$. Equation (2.1b) now yields that $t_{j-1} \in \{s_i + 1, \dots, s_i + T_i\}$, and decreasing costs then imply that

$$x^*_{it_{j-1}} = \begin{cases} 1, & t_j \notin \{s_i + 1, \dots, s_i + T_i\} \\ 0, & \text{otherwise} \end{cases} \tag{4.11}$$

However this statement is the same as that defining g, and it follows that the edges $e_1, \dots, e_m$ can be chosen in the desired way. All that remains is to conclude that $e_{m+1} = (t_m, \tau_1, \dots, \tau_n) \to (T + 1)$, which completes the construction.

□

What we have managed to show is now that BRP with decreasing costs can be solved as shortest path problem in the weighted digraph $\mathcal{D}$.

4.2.2 Complexity analysis

The natural question arises how difficult it is to solve the shortest path problem defined above. We assume that the reader is familiar with the most common shortest path solvers and the standard notation[2]. A well known worst case runtime for Dijkstra's

[2]Otherwise, a primer can be found on Wikipedia

algorithm is $\mathcal{O}(|V|log|V| + |E|)$, and the number of expanded nodes is $\mathcal{O}(|V| + |E|)$. For the above problem we have $|V| = \mathcal{O}(T\Pi_i T_i)$ and $|E| \leq n|V|$, and it follows for large instances of BRP, the runtime may grow very large. However in practice we expect the runtime to be much smaller, as the digraph $\mathcal{D}$ is not connected, and it suffices to use estimates of the size of the connected component of the source state. Consider instances of BRP where we fix n, T, and are allowed to vary $T_i,\ i \in \mathcal{N}$. As we increase T_i, it follows that $|V|$ grows very rapidly, and in the extreme case $T_i = T,\ i \in \mathcal{N}$, the size of the state graph is $|V| = T^{n+1}$. However the connected component of the initial node has $|V_0| = 3$, that is the intitial node, the goal node and an intermediary node at $(T, 0, \ldots, 0)$. Continuing with this line of reasoning, we heuristically expect the number of expanded nodes to depend not on T directly, but rather on $\frac{T}{\min_i T_i}$. We partially formalize this in a proposition.

Proposition 30. *Let an instance P of BRP with constant costs be given by $n, T, T_i, c_i, d, i \in \mathcal{N}$. Define the k-discretized, $k \in \mathbb{Z}^+$, modification of this as P^k with $T' = kT$, $n' = n$, $T'_i = kT_i, c'_i = c_i, d' = d,\ i \in \mathcal{N}$. Then the associated shortest path problems for P and P^k are equivalent.*

Proof. We prove this by showing that the connected component of the initial node in both problems induce the same digraph. Without loss of generality, we assume that the least common divisor $LCD(T, T_1, \ldots, T_n) = 1$. Let us denote the connected components of the initial nodes of the digraphs as $\mathcal{D}_c,\ \mathcal{D}_c^k$. We begin by mapping all nodes of $\mathcal{D}_c$ into $\mathcal{D}_c^k$ by letting $(t, \tau_1, \ldots, \tau_n) \mapsto (kt, k\tau_1, \ldots, k\tau_n)$. It then suffices to show that this mapping is surjective. However it follows from the definition of the edges that for any node in $\mathcal{D}_c^k$, $v = (t', \tau'_1, \ldots, \tau'_n)$ we have that $t' \equiv 0 \bmod k$, $\tau'_i \equiv 0 \bmod k$, and the proposition follows. □

The proposition shows that in a sense the time discretization of a given problem does not make the problem harder to solve, something which is certainly not true for a general MIP.

4.2.3 Improvements

To have any chance of efficiently solving large problems, it is not possible to use Dijkstra's algorithm, but rather the guided A*-algorithm [WiA], which functions in exactly the same way as Dijkstra's algorithm, but on a graph with modified edge weights

$$w'_{ij} = w_{ij} - h(i) + h(j), \tag{4.12}$$

where $h(i)$ denotes an optimistic heuristic of the shortest path distance from node i to the goal node. Hence, where Dijkstra ranks nodes according the distance from the source to those nodes, A* ranks them according to distance from source to node *and* an estimate of distance from node to goal. We then also need to supply the algorithm with a heuristic for computing the remaining distance to the goal state, and we will in this section discuss different ways to do this. There are essentially four different approaches: simple estimates, LP-relaxation estimates, strengthened LP-relaxation estimates, and partitioning heuristics. We begin by discussing LP-relaxation estimates, and then derive simple estimates from this, as well as some strengthened LP-relaxations.

Consider a state in the digraph $\mathcal{D}_c$, $v = (t, \tau_1, \ldots, \tau_n)$. By construction the partial solution of BRP that corresponds to a path leading to this state keeps the maintenance system working until time t. It follows that an admissible heuristic for computing the remaining path length is given by solving a modified version of the LP-relaxation with the variables corresponding to the current path fixed. A strenghtened LP-relaxation can then be obtained by adding some cutting planes to the subproblem, or by running a branch & cut scheme with a few nodes. It should be noted that the A*-algorithm then reduces to a branch & cut algorithm for the entire problem, with an adapted branching rule. We expect this choice of a heuristic to be a good general heuristic for a wide range of problems, as it yields a balanced tradeoff between heuristic accuracy and heuristic computation time. We may also choose to improve this relaxation by for example requiring that the total number of maintenance occasions/component replacements should be integral.

For smaller problems, where the number of expanded nodes may not be a concern, we will instead want a faster, but perhaps less accurate heuristic. This can be obtained by computing lower bounds on the optimal value of the LP-relaxed subproblem above. We can easily do this by computing lower bounds on the number of replacements of each part, as well as a lower bound on the number of maintenance occcasions, which can be obtained directly as linear combinations of (2.1b). We get that

$$\sum_{t=s}^{T} x_{it} \geq 1 + \lfloor \frac{T - s - \tau_i + 1}{T_i} \rfloor \tag{4.13}$$

$$\sum_{t=s+1}^{T} z_t \geq \lfloor \frac{T - s + 1 - \min_i\{\tau_i | \tau > 0\}}{\min_i T_i} \rfloor. \tag{4.14}$$

We now reach the most involved heuristic, which we expect to have to use for very large problems. This is motivated by the intuitive observation that for large T, i.e. $\frac{T}{\min_i T_i}$ is large, the solution to BRP should be periodical[3], and the separability property discussed in Chapter 2. That is, we wish to use the optimal values for smaller instances as an estimate of the optimal value of larger problems. Consider instances of BRP where $n,\ T_i,\ i \in \mathcal{N}$ are fixed, and we are allowed to let T vary. Let $f^*(T)$ be the optimal value of the problem with planning horizon T.

Proposition 31. *For fixed $n, T_i, i \in \mathcal{N}$, if c_{it}, d_t are decreasing in t, then f^* is superadditive, that is*

$$f^*(T_1 + T_2) \geq f^*(T_1) + f^*(T_2). \tag{4.15}$$

Furthermore,

$$f^*(T_1 + T_2) \leq f^*(T_1) + f^*(T_2) + d + \sum_{i \in \mathcal{N}} c_i. \tag{4.16}$$

[3]Or rather, there exists a near-optimal solution that is periodical

Proof. Let (x^*, z^*) be an optimal solution to BRP with planning horizon T. Define (x^j, z^j), $j = 1, 2$ as

$$x^1_{it} = x^*_{it}, \qquad t \in \{1, \dots, T_1\}, \tag{4.17a}$$
$$z^1_t = z^*_t, \qquad t \in \{1, \dots, T_1\}, \tag{4.17b}$$
$$x^2_{i,t} = x^*_{i,t+T_1}, \qquad t \in \{1, \dots, T_2\}, \tag{4.17c}$$
$$z^2_t = z^*_{t+T_1}, \qquad t \in \{1, \dots, T_2\}. \tag{4.17d}$$

It follows that (x^j, z^j) are feasible in BRP with planning horizons T_j, respectively, and consequently $f(x^j, z^j) \geq f^*(T_j)$, $j = 1, 2$. Finally

$$f^*(T_1 + T_2) = f(x^1, z^1) + f(x^2, z^2) \geq f^*(T_1) + f^*(T_2). \tag{4.18}$$

Conversely if we let (x^j, z^j), $j = 1, 2$, be optimal solutions to the problems with planning horizons T_j, $j = 1, 2$, respectively, we can define a feasible solution to the problem with planning horizon $T_1 + T_2$ using (4.17), and adding $x_{iT_1} = z_{T_1} = 1$ for $i \in \mathcal{N}$, finishing the proof.

□

We also point out that the same result holds for the LP-relaxed optima. We wish to alert the reader to that the main usage of the partitioning heuristic is not to produce good quality solutions to the BRP, but to provide stronger lower bounds than are obtainable by linear relaxation methods. Also note that the partitioning is very similar to the Lagrangian relaxation with multipliers 0 that was discussed in Chapter 2. Finally we note that there exists in a sense a dual version of the partitiong, where we instead of partitiong the planning horizon partition the components into subgroups. It is then easily seen that the inequalities corresponding to Proposition 31 for component partitionings hold when the inequalities are reversed, which may then be used to construct a primal heuristic.

4.2.4 Heuristics

For large instances, the method above may not be able to terminate within a reasonable timeframe, and we may need to use some heuristics to speed things up. A basis for this comes from Proposition 31, which loosely speaking is an indication that the linear relaxation gap grows asymptotically linearly for a fixed instance when increasing T. A further motivation of this is that for very large T we expect the solution to both BRP and its linear relaxation to be periodical. This yields that the heuristic estimates of remaining cost are less sharp for states at early times, and that this should be linearly related. We may then motivate the heuristic to introduce a memory horizon for the A*-algorithm, that is, we delete all the open states that are some fixed distance away from the latest expanded state. That is, if we have found a state at t with an estimated cost of f_t, and there are states at $s < t$ with the same score[4], we guess that the state

[4]Note that it cannot be smaller as t was an expanded node

at s is worse than the one at t, and that we are fooled into believing that it is good merely because the heuristic bound is bad. Note that for this heuristic to perform well, the remaining distance heuristic has to behave monotonically. It is controlled through the use of 2 parameters, MEM_PURGE_FREQ and MEM_HORIZON, which control how often the memory is purged, and how far back to purge, respectively. It is natural to estimate that a natural time-scale for the problem at hand is given by its minimal life limit, and hence MEM_HORIZON is defined in terms of multiples of this.

Another heuristic speedup is to limit the number of open nodes at any given instant of time. That is, we define a parameter MAX_NODES_AT_T, which defines how many open states we allow that are at the same t. If a new state is added to the open set with the same t, we simply remove the state at t with the worst score. This heuristic will then work well if the heuristic is monotone for states at t, in the sense that we assume that the ordering of heuristic scores are a good estimate of the ordering of actual scores. This heuristic has the added benefit of keeping the memory usage of the algorithm low.

Chapter 5

Numerical results

In this chapter we present some basic numerical tests on implementations of the theoretical results obtained in this thesis. This chapter is not meant as as extensive numerical study, but should rather be read as a guideline to when the theory is applicable. The reason for this is mainly that, as the problem at hand has not previously been extensively studied, there exist no resource for benchmark problems. Furthermore the author does not have access to large amounts of data from real world industry applications.

There are two main investigations that we wish to perform in this chapter. These are a study of the power of the facet-generating shortest path based separation algorithm of Section 4.1, and the dynamic programming scheme of Section 4.2. For the separation algorithm, we are mainly interested in how much the linear programming relaxation gap of BRP can be reduced, and for which instances it performs well.

The dynamic programming scheme is granted a bit more attention, as there are more things of interest to test. The tests we perform include:

1. Performance of the exact solver.
2. Dependence on choice of heuristic.
3. Solution quality and performance when using heuristic speedups.
4. Comparisons with commercial grade MIP solvers.

The reference MIP solver that is used is CPLEX 12.1. Algorithms implemented by the author are written in MATLAB R2008b, with some parts written in C, being called with MATLAB's MEX-interface. All numerical experiments are performed on a dedicated Linux desktop workstation.

5.1 Separation problem

It is interesting to study how much the constraint generation scheme is able to reduce the gap of the continuous relaxation in relation to the optimal integer solution. The first test we perform is to solve many instances of a medium-sized problem, so that

the true optimum is obtainable. The instances are chosen such that the number of components n and the horizon T is fixed. The component costs c_i are fixed random numbers $c_i \in U(1, 2)$, and the fixed cost d is varied.

We test using the simplest version of the separation problem of Section 4.1. The implemented algorithm version is capable of generating facets of the second layer only. The authors experience indicate that gap reduction is essentially dependent on one factor, which is the minimal life of the components. In Figures 5.1 and 5.2, we graph the relaxation gaps before and after the cutting plane algorithm for problems with short-lived components are longlived components, respectively.

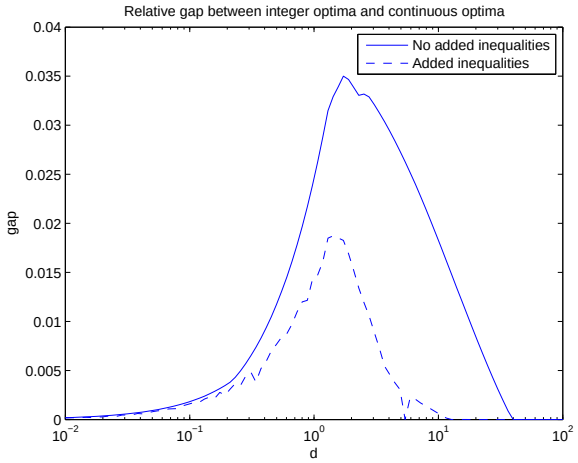

Figure 5.1: Relative gaps between integer and continuous optima, varying d, problem 1. This problem contains several components with short lives, as far down as $T_i = 3$.

We note that the facet generation performs very well on this problem.

We also give an instance for which this approach performs poorly. The instance is constructed such that $\min T_i$ is larger; such a problem is problem 2, and the results are depicted in Figure 5.2.

The poor performance can be explained somewhat. What the given subclass of facets does is to superposition slightly more than, for simplicity, two inequalities from (2.1b), yielding for both fractional and integer solutions an inequality that is probably not satisfied strictly. However in fractional solutions, the mixing with (2.1c) allows us to exchange for example $x_{21} = 1/2,\ x_{22} = 1/2$ with $z_2 = 1/2$ in the inequality. Heuristically, it follows that the more x-variable content we exchange with z-variable content, the more violation we obtain. For this subclass of facets, we maximally substitute 4 x-variables for 2 z variables. It, again heuristically, follows that the amount of violation of constraints is determined by the typical size of the values of the variables in the fractional solution. Now, finally, we expect that the typical size of the variables in the optimal fractional solution is $(\min_i T_i)^{-1}$, for obvious reasons, and hence that the subclass of facets generated above is stronger for problem with shorter lives than for larger lives.

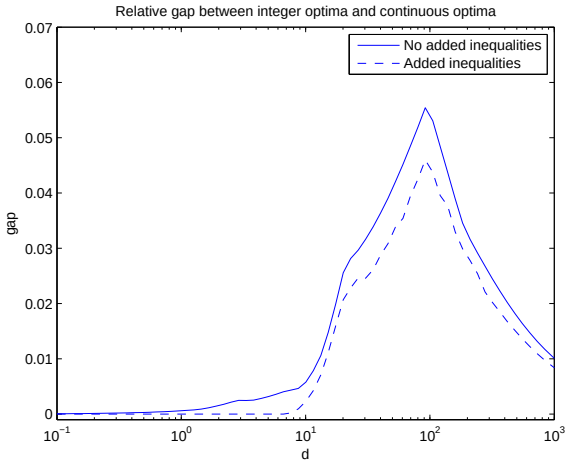

Figure 5.2: Relative gaps between integer and continuous optima, varying d, problem 2. This instance contains components with lives spanning from $T_i = 13$ to $T_i = 40$.

If we now consider the zero-half cuts of Section 3.2, they were allowed to use many more mixing constraints from (2.1c). As the lives of the components increase, causing the violation gain from one mixing to decrease, the number of possible mixings increase as well. Therefore we expect the zero-half cuts to be more robust for problems with longer lives. Unfortunately, it seems to be much harder to add facets with a large number of mixings as they are very time-consuming to identify, which is why this has not yet been implemented.

5.2 Dynamic search algorithms

We also wish to test the performance of the proposed A*-algorithm of Section 4.2 for different heuristics, as well as comparing to commercial general MIP solvers. Again, implementations are written in MATLAB with linear programs solved in CPLEX 12.1, which is also the MIP solver that is used for comparison. As a cutting plane generation has not yet been implemented to work within this algorithm, we take advantage of CPLEX's built-in zero-half cut generation to emulate the results of section 3.2. These implementation details may have a significant impact on timing results. MATLAB's scripting language, being uncompiled, is typically very slow for applications containing many loops. Hence the A*-algorithm is severely penalized in realtime comparisons[1].

The tests are performed on the problems of Table 5.2.1. The relevant data that is measured is the computation time in real time, and the number of nodes that are processed for each algorithm. The algorithms that are tested are denoted CPLEX, Simple,

[1]It can be mentioned that in the shortest path algorithm for the separation problem, early versions were coded in MATLAB. When switching to C, a speedup factor of roughly 50-100 was obtained.

Problem	n	T	$\min T_i$	$\max T_i$
1	10	50	2	11
2	4	100	9	22
3	2	500	15	23
4	2	13	3	5

LP and LPCUT, where the names of the last three indicate the heuristic estimate of remaining distance that is used.

5.2.1 Problem testbed

Throughout this section we will variations of six problems as a benchmark. Two problems, HPT and LPT, are real-world instances obtained from Volvo Aero, while problems 1 through 4 are invented by the author to be fairly illustrative examples of different classes of problems. For the four invented problems, basic data is given in Table 5.2.1.

The problems of Table 5.2.1 were chosen to represent different classes of instances. Problem 1 is a dense problem, in the sense that it contains components with similar lives, which makes optimal solution contain many maintenance occasions. Problem 2 is intented to be a more realistic problem. Problem 3 is chosen to investigate the effects of a long planning horizon, while problem 4 is a very simple problem intended to investigate whether there exists some overhead when running the algorithms.

5.2.2 Time dependence of relaxation gaps

It was hypothesised earlier that the gaps of the heuristic estimates for fixed instances grew as the planning horizon grew. We test this hypothesis by computing both integer and continous optima for some problems when varying the planning horizon. To test this we let CPLEX solve modifications of problem 2, for $T = 25, \ldots, 125$. The result is shown in Figure 5.3, from which we can see that for this problem, the hypothesis that both the integral and the continuous optimas grow approximately linearly is fairly accurate. Furthermore the gap seems to follow the same pattern. This result provides strong support for the memory horizon heuristic for the A*-algorithm.

Simple tests

In this section we let the exact A*-algorithm solve the testbed of instances in Table 5.2.1. The results are presented in Table 5.1

We see that in real time comparisons, CPLEX seems to win on most instances. However it should be noted that real time may not the most relevant basis of comparison, as this should be favoring CPLEX with its efficient implementation as opposed to the fairly slow implementation of the A*-algorithm in MATLAB. It should also be noted that CPLEX is parallellised, and uses 2 processors, while the A*-algorithm does not. More interesting is the number of expanded nodes, that should be compared with the number of branching nodes of CPLEX, where the A*-algorithm seems to perform significantly better than CPLEX, especially for long-horizon instances. However

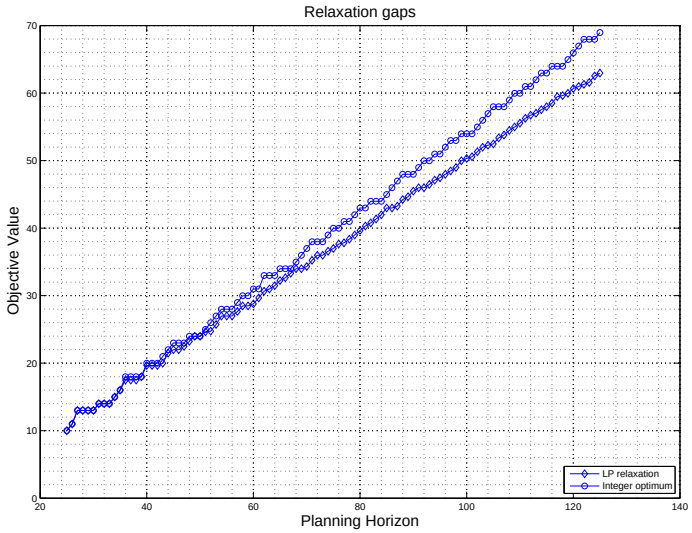

Figure 5.3: Integer and continuous optima for problem 2 when varying the planning horizon T. Both objective values appear to increase approximately linearly.

	CPLEX		Simple		LP		LPCUT	
	Time(s)	Nodes	Time(s)	Iterations	Time(s)	Iterations	Time(s)	Iterations
1	0.94	187	578	46457	873	18484	30	146
2	1.70	503	3.5	2995	6.6	81	14	51
3	$*$	326947	7.2	10384	530	3683	2327	3543
4	0.04	0	0.007	8	0.4	6	0.4	6

Table 5.1: Timing results when solving a testbed of instances. If the runtime exceeds one hour, the algorithm is considered to have failed, and this is marked by a $*$. Otherwise the problem is solved to optimality.

	Simple			LP			LPCUT		
	Time(s)	Iter.	Obj.Val.	Time(s)	Iter.	Obj.Val.	Time(s)	Iterations	Obj.Val.
1	2	1329	131	12.4	58	131	9.4	125	131
2	0.28	225	54	4.0	30	54	11.8	47	54
3	0.52	603	120	35.3	106	120	106	146	120

Table 5.2: Performance of the memory horizon heuristic on the testbed of problems. All variants obtain the true optimum.

for dense instances, that is instances with a large number of components and many maintenance occasions, such as problem 1, CPLEX still performs better. This is not surprising, as many maintenance occasions implies a large number of edges in the optimal solution in the shortest path formulation, while a large number of components significantly increases the number of edges, both increasing the computational time. However, the strong LP version still performs fairly well on this problem, which indicates the importance of choosing a proper remaining distance heuristic.

In conclusion, the A*-algorithm can beat a general MIP solver for some instances, however there is not a distinct advantage.

Heuristics

We test the heuristic version of the A*-algorithm with different parameter settings, and study the effects on solution time and quality. For the SIMPLE heuristic we use the parameters MEM_PURGE_FREQ $= 200$ and for the LP and LPCUT
MEM_PURGE_FREQ $= 20$. For all algorithms MEM_HORIZON = 2. In this initial test we use MAX_NODES_AT_T $= \infty$.

We can see in Table 5.2 that for all problems the optimal solution is obtained. Futhermore the solution times are significantly decreased. The effect is most dramatic for the Simple heuristic on problem 1. This is easily understood, as this problem is very dense, which makes the Simple heuristic very bad. This causes the long running time for the exact solution. However the simple heuristic is still consistent on this problem, in the monotone sense, making the memory horizon heuristic powerful. It is also interesting to note that we no longer neccessarily expand less nodes with the LPCUT solver.

We also study the effects when doubling the planning horizon to investigate the scaling properties of the heuristic.

The known true optima for the problems in Table 5.3 are $265, 113, 242$, as obtained by CPLEX[2]. We can then see for these fairly large problems, with planning horizons of up to $T = 1000$, the objective value is either optimal or very close to optimal, with the deviation being at most 1.5%, and with very short runtimes in comparison to a traditional branch & cut. It should be noted that the worst performance is obtained on problem 1 containing a component with life $T_i = 2$. This should not come as a

[2]Obtaining the optimum for problem 2 with CPLEX took over 36 hours.

	Simple			LP			LPCUT		
	Time(s)	Iter.	Obj.Val.	Time(s)	Iter.	Obj.Val.	Time(s)	Iter.	Obj.Val.
1	3.4	2547	268	37	293	266	114	149	270
2	1.1	800	113	108	205	113	263	274	113
3	1.1	1219	242	977	425	243	1592	392	242

Table 5.3: Performance of memory horizon heuristic on the testbed of problems with doubled horizons.

	CPLEX		LP		LP heuristic		
	Time(s)	Nodes	Time(s)	Iterations	Time(s)	Iterations	Gap
HPT	1770	66797	807	1031	161	206	0
LPT	53	220	53	47	53	47	0

Table 5.4: Performance on two real world problems for CPLEX and the LP A*-algorithm.

surprise as the memory horizon then becomes very short[3], and demonstrates that the generic setting $MEM\_HOR = 2$ is actually fairly robust even in this pathological case.

5.2.3 Real world instances

We try our algorithms on real world data obtained from Volvo Aero Corporation. The data is however to be considered as secret, and we only present numbers on the magnitude of the problems. There are two problem sets, HPT and LPT, that stem from maintenance criteria on moddule of a high- and low pressure turbine of an aircraft engine. The problems contain safety critical components that must not break, and are as such assigned a deterministic life limit, making the BRP applicable. For the HPT problem there are 9 components, and the LPT problem has 10 components, both with a planning horizon $T = 200$. We only test the LP version of the exact A*-algorithm, as it from the previous result seem to be the most robust, as well as being the most comparable to CPLEX in terms of the number of nodes it expands.

A graph of the solution progress for the algorithms above on HPT is presented in figure 5.4. Note that this graph plots the number of solved LP-relaxations for the A*-algorithm, which is larger than the number of expanded nodes. Furthermore, we only allow CPLEX to use one thread. It is interesting to note that the number of nodes is cut down significantly for CPLEX compared to the parallell processing solution. This is an indication that for this type of problems, parallell processing MIP solvers may not have a significant advantage over singlethreaded solvers. However, it is likely that the performance increase for parallell processing in the A*-algorithm is likely to be approximately linear in the number of processors. We can see that on the LPT,

[3]Much shorter than the longest life of $T_i = 11$.

there is not much difference in the solution process, and they are in fact tied in real time. However for HPT they are miles apart, where CPLEX expands 65 times as many nodes, and is also significantly beaten in the real time comparison. It was also tested to let CPLEX solve the HPT problem without allowing it generate any cutting planes, where it failed to produce an optimal solution after running for 24 hours.

We also try the full-force heuristic on both HPT and LPT. We use Simple on HPT and LPT, with parameters MEM_PURGE_FREQ $= 200$, MEM_HOR $= 3$, MAX_NODES_AT_T $= 20$, settings that are to be considered as generic. For these settings we vary T. The problems considered will be too large to prove optimality. Firstly we study the solution times for the Simple version algorithm with full heuristics, by letting it solve variants of HPT with planning horizons $100 \leq T \leq 1100$. The result is graphed in Figure 5.5. We see that the solution time is approximately linear in T, as is to be expected.

If the heuristic is to be used to solve a subproblem in, say, a stochastic programming setting, what we wish to obtain is most likely the optimal maintenance decision today, and future maintenance scheduling is less interesting. In this setting, a heuristic that prematurely cuts off maintenance decisions today may perform badly. Hence we also report solution times when not using the memory horizon heuristic, which are graphed in figure 5.6 for HPT with $50 \leq T \leq 650$.

We also investigate the solution quality obtained. The true optima are computed by CPLEX, and are compared to the heuristic optima in Figure 5.7 for $100 \leq T \leq 190$. In fact the true optimum is obtained for all tested values of T.

Figure 5.4: Solution progress on the HPT problem, displaying the lower bound the optimum. The horizontal line is the true integer optimum

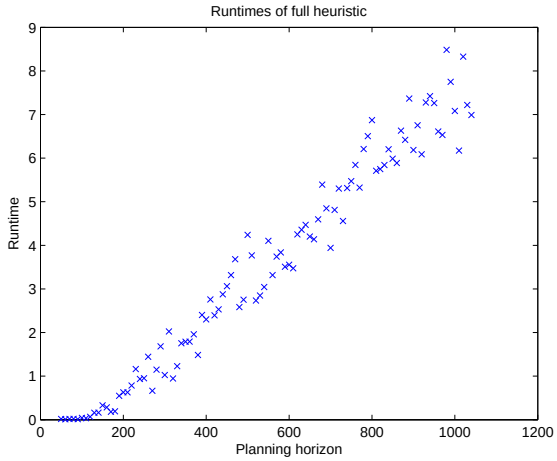

Figure 5.5: Time dependence of full-force heuristic.

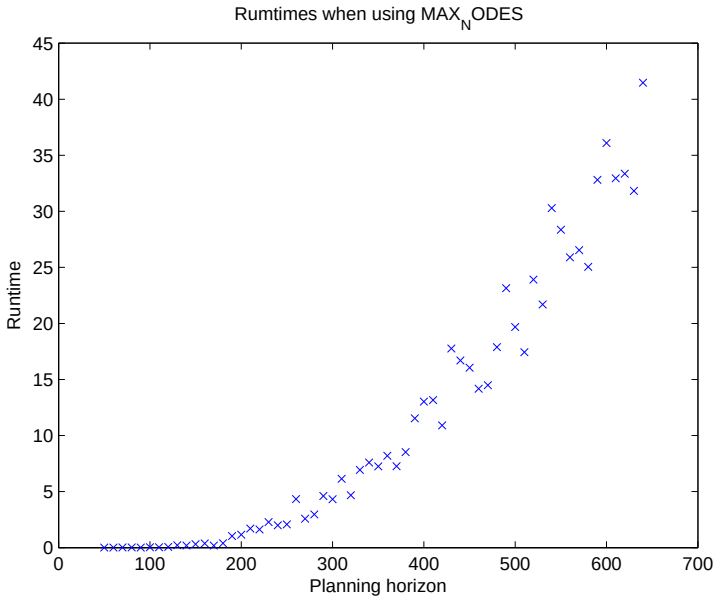

Figure 5.6: Time dependence of heuristic with MEM_HOR = ∞

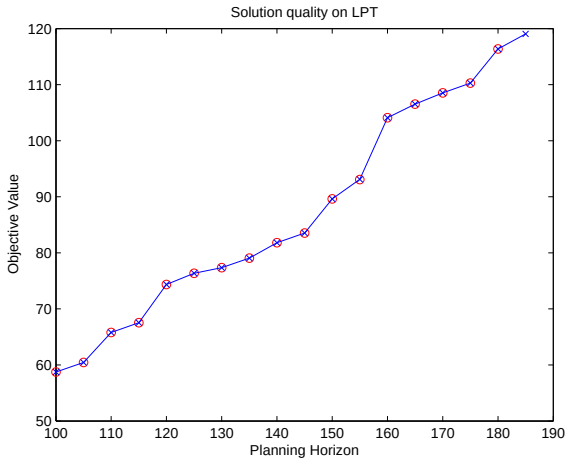

Figure 5.7: Solution quality obtained on HPT for various planning horizons. Circles are true optimas, while the line displays heuristically obtained optimas.

Chapter 6

Final Remarks

The work done in this thesis show that binary linear programming models for opportunistic maintenance planning are likely to have sufficient substructure to make *á priori* conclusions about the facial structure. In particular, they have a set covering substructure, which may be used as a guiding principle for generation of strong cutting planes. It may also be possible to find combinatorial implications that generate cutting planes. However these cutting planes can be hard to generate, which may be a direction for future research.

Furthermore we have demonstrated that natural dynamic programming formulations can have a significant advantage over branch-and-cut approaches, if there exists some way to *á priori* reduce the number of maintenance decisions that stand any chance of being optimal.

The biggest remaining question is however if our approaches extend to a stochastic programming setting.

Bibliography

[AEP05] N. ANDRÉASSON, A. EVGRAFOV, AND M. PATRIKSSON, *An Introduction to Continuous Optimization*, Studentlitteratur, Lund, 2005.

[NW88] G.L. NEMHAUSER AND L.A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.

[CF93] A.CAPRARA, M. FISCHETTI,$\{0, 1/2\}$-*Chvátal-Gomory cuts*, Mathematical Programming, Volume 74, Number 3 (1996).

[RKM04] R.K. MOBLEY, *Maintenance Fundamentals*, Elsevier, Amsterdam, second ed., 2004.

[DEW91] B. DICKMAN, S. EPSTEIN AND Y. WILAMOWSKY, *A mized integer liear programming formulation for the multi-component deterministic opportunistic replacement*, The Journal of Operational Research Society of India, 28 (1991),pp. 165-175.

[PSW09] T.ALMGREN, N. ANDRÉASSON, M PATRIKSSON, A. STRÖMBERG, A. WOJCIECHOWSKI, *The replacement problem: A polyhedral and complexity analysis.*, Preprint - Department of Mathematical Sciences, Chalmers University of Technology and Göteborg University 2009.

[APS08] T. ALMGREN, N.ANDRÉASSON, D. ANEVSKI, M.PATRIKSSON, A. STRÖMBERG, J. SVENSSON, *Optimization of opportunistic maintenance activities: A case study in the aircraft industry.*, Preprint - Department of Mathematical Sciences, Chalmers University of Technology and Göteborg University; 2008:45

[Pad73] M.PADBERG, *On the facial structure of set packing polyhedra*, Mathematical Programming, Volume 5, Number 1 (1973), pp. 199-215.

[WiA] http://en.wikipedia.org/wiki/A-star, Accessed March 10, 2010.